

VideoComposer: Compositional Video Synthesis with Motion Controllability

Xiang Wang^{1*} Hangjie Yuan^{1*} Shiwei Zhang^{1*} Dayou Chen^{1*} Jiuniu Wang¹
 Yingya Zhang¹ Yujun Shen² Deli Zhao¹ Jingren Zhou¹

¹Alibaba Group ²Ant Group

{xiaolao.wx, yuanhangjie.yhj, zhangjin.zsw}@alibaba-inc.com
 {dayou.cdy, wangjiuniu.wjn, yingya.zyy, jingren.zhou}@alibaba-inc.com
 {shenyujun0302, zhaodeli}@gmail.com

Abstract

The pursuit of controllability as a higher standard of visual content creation has yielded remarkable progress in customizable image synthesis. However, achieving **controllable video synthesis** remains challenging due to the large variation of temporal dynamics and the requirement of cross-frame temporal consistency. Based on the paradigm of compositional generation, this work presents **VideoComposer** that allows users to flexibly compose a video with textual conditions, spatial conditions, and more importantly temporal conditions. Specifically, considering the characteristic of video data, we introduce the motion vector from compressed videos as an explicit control signal to provide guidance regarding temporal dynamics. In addition, we develop a Spatio-Temporal Condition encoder (STC-encoder) that serves as a unified interface to effectively incorporate the spatial and temporal relations of sequential inputs, with which the model could make better use of temporal conditions and hence achieve higher inter-frame consistency. Extensive experimental results suggest that VideoComposer is able to control the spatial and temporal patterns simultaneously within a synthesized video in various forms, such as text description, sketch sequence, reference video, or even simply hand-crafted motions. The code and models will be publicly available at <https://videocomposer.github.io>.

1 Introduction

Driven by the advances in computation, data scaling and architectural design, current visual generative models, especially diffusion-based models, have made remarkable strides in automating content creation, empowering designers to generate realistic images or videos from a textual prompt as input [22, 44, 49]. These approaches typically train a powerful diffusion model [44] conditioned by text [21] on large-scale video-text and image-text datasets [3, 47], reaching unprecedented levels of fidelity and diversity. However, despite this impressive progress, a significant challenge remains in the limited controllability of the synthesis system, which impedes its practical applications.

*Note condition 31
diff control*

Most existing methods typically achieve controllable generation mainly by introducing new conditions, such as segmentation maps [44, 59], inpainting masks [66] or sketches [34, 72], in addition to texts. Expanding upon this idea, Composer [26] proposes a new generative paradigm centered on the concept of **compositionality**, which is capable of composing an image with various input conditions, leading to remarkable flexibility. However, Composer primarily focuses on considering

*Equal contribution.

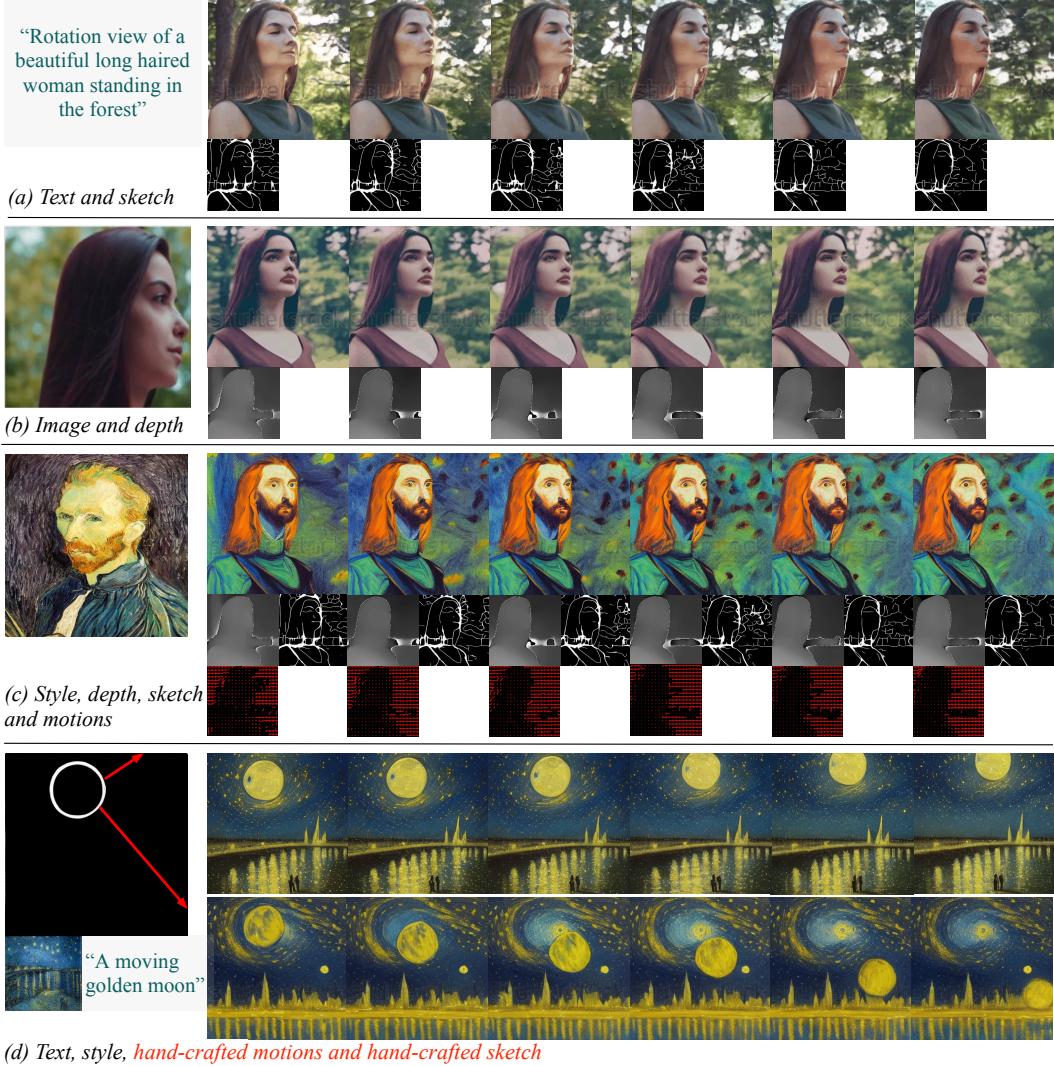


Figure 1: **Compositional video synthesis.** **(a-c)** VideoComposer is capable of generating videos that adhere to textual, spatial and temporal conditions or their subsets; **(d)** VideoComposer can synthesize videos conforming to expected motion patterns (red stroke) and shape patterns (white stroke) derived from two simple strokes.

multi-level conditions within the spatial dimension, hence it may encounter difficulties when comes to video generation due to the inherent properties of video data. This challenge arises from the complex temporal structure of videos, which exhibits a large variation of temporal dynamics while simultaneously maintaining temporal continuity among different frames. Therefore, incorporating suitable temporal conditions with spatial clues to facilitate controllable video synthesis becomes significantly essential.

Above observations motivate the proposed VideoComposer, which equips video synthesis with improved controllability in both spatial and temporal perception. For this purpose, we decompose a video into three kinds of representative factors, *i.e.*, textual condition, spatial conditions and the crucial temporal conditions, and then train a latent diffusion model to recompose the input video conditioned by them. In particular, we introduce the video-specific *motion vector* as a kind of temporal guidance during video synthesis to explicitly capture the inter-frame dynamics, thereby providing direct control over the internal motions. To ensure temporal consistency, we additionally present a unified STC-encoder that captures the spatio-temporal relations within sequential input utilizing cross-frame attention mechanisms, leading to an enhanced cross-frame consistency of the output videos. Moreover, STC-encoder serves as an interface that allows for efficient and unified utilization of the control signals from various condition sequences. As a result, VideoComposer is

capable of flexibly composing a video with diverse conditions while simultaneously maintaining the synthesis quality, as shown in Fig. 1. Notably, we can even control the motion patterns with simple hand-crafted motions, such as an arrow indicating the moon’s trajectory in Fig. 1d, a feat that is nearly impossible with current methods. Finally, we demonstrate the efficacy of VideoComposer through extensive qualitative and quantitative results, and achieve exceptional creativity in the various downstream generative tasks.

2 Related work

Image synthesis with diffusion models. Recently, research efforts on image synthesis have shifted from utilizing GANs [17], VAEs [29], and flow models [13] to diffusion models [21, 50, 54, 73] due to more stable training, enhanced sample quality, and increased flexibility in a conditional generation. Regarding image generation, notable works such as DALL-E 2 [42] and GLIDE [36] employ diffusion models for text-to-image generation by conducting the diffusion process in pixel space, guided by CLIP [40] or classifier-free approaches. Imagen [46] introduces generic large language models, *i.e.*, T5 [41], improving sample fidelity. The pioneering work LDMs [44] uses an autoencoder [14] to reduce pixel-level redundancy, making LDMs computationally efficient. Regarding image editing, pix2pix-zero [38] and prompt-to-prompt editing [19] follow instructional texts by manipulating cross-attention maps. Imagic [27] interpolates between an optimized embedding and the target embedding derived from text instructions to manipulate images. DiffEdit [11] introduces automatically generated masks to assist text-driven image editing. To enable conditional synthesis with flexible input, ControlNet [72] and T2I-Adapter [34] incorporate a specific spatial condition into the model, providing more fine-grained control. One milestone, Composer [26], trains a multi-condition diffusion model that broadly expands the control space and displays remarkable results. Nonetheless, this compositionality has not yet been proven effective in video synthesis, and VideoComposer aims to fill this gap.

Video synthesis with diffusion models. Recent research has demonstrated the potential of employing diffusion models for video synthesis [6, 18, 23, 28, 32, 68]. Notably, Imagen Video [22] and Make-A-Video [49] both model the video distribution in pixel space, which limits their applicability due to high computational demands. In contrast, MagicVideo [75] models the video distribution in the latent space, following the paradigm of LDMs [44], significantly reducing computational overhead. With the goal of editing videos guided by texts, Video2P2P [30] and vid2vid-zero [60] manipulate the cross-attention map, while Dreamix [33] proposes an image-video mixed fine-tuning strategy. However, their generation or editing processes solely rely on text-based instructions [40, 41]. A subsequent work, Gen-1 [15], integrates depth maps alongside texts using cross-attention mechanisms to provide structural guidance. Both MCDiff [9] and LaMD [25] target motion-guided video generation; the former focuses on generating human action videos and encodes the dynamics by tracking the keypoints and reference points, while the latter employs a learnable motion latent to improve quality. Nevertheless, incorporating the guidance from efficient motion vectors or incorporating multiple guiding conditions within a single model is seldom explored in the general video synthesis field.

Motion modeling. Motion cues play a crucial role in video understanding fields, such as action recognition [2, 5, 7, 39, 55, 57, 58], action detection [10, 62, 70, 74], human video generation [35, 37, 61], *etc.* Pioneering works [2, 7, 35, 39, 57, 61] usually leverage hand-crafted dense optical flow [69] to embed motion information or design various temporal structures to encode long-range temporal representations. Due to the high computational demands of optical flow extraction, several attempts in compressed video recognition [8, 48, 63, 71] have begun to utilize more efficient motion vectors as an alternative to represent motions and have shown promising performance. In contrast to these works, we delve into the role of motions in video synthesis and demonstrate that motion vectors can enhance temporal controllability through a well-designed architecture.

3 VideoComposer

In this section, we will comprehensively present VideoComposer to showcase how it can enhance the controllability of video synthesis and enable the creation of highly customized videos. Firstly, we in brief introduce Video Latent Diffusion Models (VLDMs) upon which VideoComposer is designed, given their impressive success in various generative tasks. Subsequently, we delve into the details of VideoComposer’s architecture, including the composable conditions and unified Spatio-Temporal

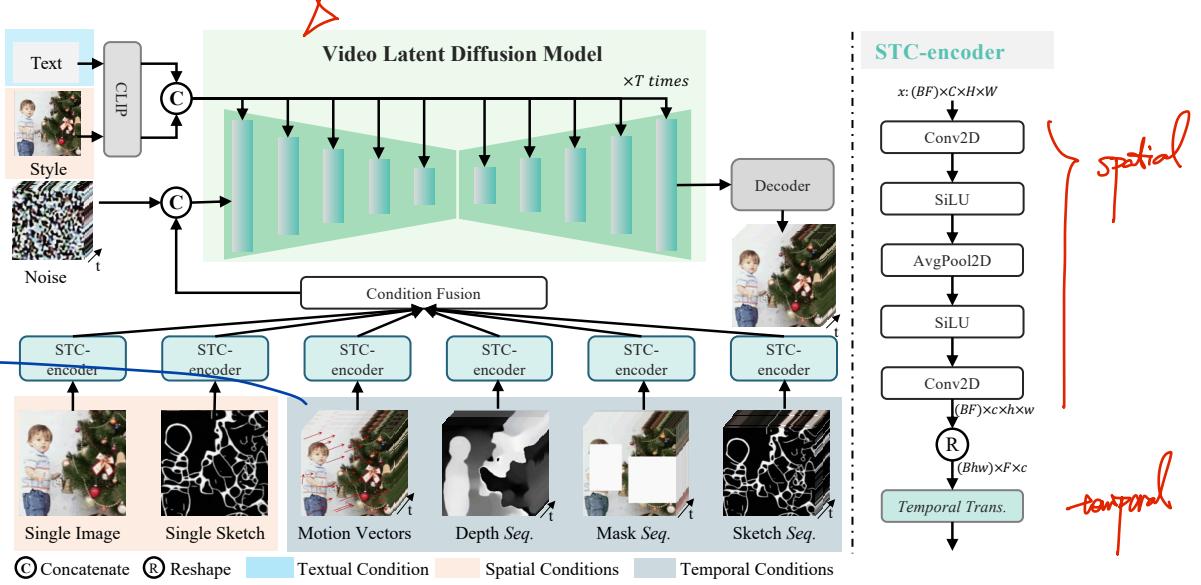


Figure 2: **Overall architecture** of VideoComposer. First, a video is decomposed into three types of conditions, including textual condition, spatial conditions and temporal conditions. Then, we feed these conditions into the unified STC-encoder or the CLIP model to embed control signals. Finally, the resulting conditions are leveraged to jointly guide VLDMs for denoising.

Condition encoder (STC-encoder) as illustrated in Fig. 2. Finally, the concrete implementations, including the training and inference processes, will be analyzed.

3.1 Preliminaries

Compared to images, processing video requires substantial computational resources. Intuitively, adapting image diffusion models that process in the pixel space [36, 42] to the video domain impedes the scaling of VideoComposer to web-scale data. Consequently, we adopt a variant of LDMs that operate in the latent space, where local fidelity could be maintained to preserve the visual manifold.

Perceptual video compression. To efficiently process video data, we follow LDMs by introducing a pre-trained encoder [14] to project a given video $x \in \mathbb{R}^{F \times H \times W \times 3}$ into a latent representation $z = \mathcal{E}(x)$, where $z \in \mathbb{R}^{F \times h \times w \times c}$. Subsequently, a decoder \mathcal{D} is adopted to map the latent representations back to the pixel space $\bar{x} = \mathcal{D}(z)$. We set $H/h = W/w = 8$ for rapid processing.

Diffusion models in the latent space. To learn the actual video distribution $\mathbb{P}(x)$, diffusion models [21, 50] learn to denoise a normally-distributed noise, aiming to recover realistic visual content. This process simulates the reverse process of a Markov Chain of length T . T is set to 1000 by default. To perform the reverse process on the latent, it injects noise to z to obtain a noise-corrupted latent z_t following [44]. Subsequently, we apply a denoising function $\epsilon_\theta(\cdot, \cdot, t)$ on z_t and selected conditions c , where $t \in \{1, \dots, T\}$. The optimized objective can be formulated as:

$$\mathcal{L}_{VLDM} = \mathbb{E}_{\mathcal{E}(x), \epsilon \in \mathcal{N}(0, 1), c, t} [\|\epsilon - \epsilon_\theta(z_t, c, t)\|_2^2] \quad (1)$$

To exploit the inductive bias of locality and temporal inductive bias of sequentiality during denoising, we instantiate $\epsilon_\theta(\cdot, \cdot, t)$ as a 3D UNet augmented with temporal convolution and cross-attention mechanism following [1, 23, 45].

3.2 VideoComposer

Videos as composable conditions. We decompose videos into three distinct types of conditions, *i.e.*, textual conditions, spatial conditions and crucially temporal conditions, which can jointly determine the spatial and temporal patterns in videos. Notably, VideoComposer is a generic compositional framework. Therefore, more customized conditions can be incorporated into VideoComposer depending on the downstream application and are not limited to the decompositions listed above.



Textual condition. Textual descriptions provide an intuitive indication of videos in terms of coarse-grained visual content and motions. In our implementation, we employ the widely used pre-trained text encoder from OpenCLIP² ViT-H/14 to obtain semantic embeddings of text descriptions.

Spatial conditions. To achieve fine-grained spatial control and diverse stylization, we apply three spatial conditions to provide structural and stylistic guidance: *i) Single image*. Video is made up of consecutive images, and a single image usually reveals the content and structure of this video. We select the first frame of a given video as a spatial condition to perform image-to-video generation. *ii) Single sketch*. We extract sketch of the first video frame using PiDiNet [51] as the second spatial condition and encourage VideoComposer to synthesize temporal-consistent video according to the structure and texture within the single sketch. *iii) Style*. To further transfer the style from one image to the synthesized video, we choose the image embedding as the stylistic guidance, following [4, 26]. We apply a pre-trained image encoder from OpenCLIP ViT-H/14 to extract the stylistic representation.

Temporal conditions. To accomplish finer control along the temporal dimension, we introduce four temporal conditions: *i) Motion vector*. Motion vector as a video-specific element is represented as two-dimension vectors, *i.e.*, horizontal and vertical orientations. It explicitly encodes the pixel-wise movements between two adjacent frames, as visualized by red arrows in Fig. 3. Due to the natural properties of motion vector, we treat this condition as a motion control signal for temporal-smooth synthesis. Following [48, 63], we extract motion vectors in standard MPEG-4 format from compressed videos. *ii) Depth sequence*. To introduce depth information, we utilize the pre-trained model from [43] to extract depth maps of video frames. *iii) Mask sequence*. To facilitate video regional editing and inpainting, we manually add masks. We introduce tube masks [16, 53] to mask out videos and enforce the model to predict the masked regions based on observable information. *iv) Sketch sequence*. Compared with the single sketch, sketch sequence can provide more control details and thus achieve precisely customized synthesis.



Figure 3: Examples of motion vectors.

STC-encoder. Sequential conditions contain rich and complex space-time dependencies, posing challenges for controllable guidance. In order to enhance the temporal awareness of input conditions, we design a Spatio-Temporal Condition encoder (STC-encoder) to incorporate the space-time relations, as shown in Fig. 2. Specifically, a light-weight spatial architecture consisting of two 2D convolutions and an average pooling layer is first applied to the input sequences, aiming to extract local spatial information. Subsequently, the resulting condition sequence is fed into a temporal Transformer layer [56] for temporal modeling. In this way, STC-encoder facilitates the explicit embedding of temporal cues, allowing for a unified condition interface for diverse inputs, thereby enhancing inter-frame consistency. It is worth noting that we repeat the spatial conditions of a single image and single sketch along the temporal dimension to ensure their consistency with temporal conditions, hence facilitating the condition fusion process.

After processing the conditions by STC-encoder, the final condition sequences are all in an identical spatial shape to z_t and then fused by element-wise addition. Finally, we concatenate the merged condition sequence with z_t along the channel dimension as control signals. For textual and stylistic conditions organized as a sequence of embeddings, we utilize the cross-attention mechanism to inject textual and stylistic guidance.

3.3 Training and inference

Two-stage training strategy. Although VideoComposer can initialize with the pre-training of LDMs [44], which mitigates the training difficulty to some extent, the model still struggles in learning to simultaneously handle temporal dynamics and synthesize video content from multiple compositions. To address this issue, we leverage a two-stage training strategy to optimize VideoComposer. Specifically, the first stage targets pre-training the model to specialize in temporal modeling through text-to-video generation. In the second stage, we optimize VideoComposer to excel in video synthesis controlled by the diverse conditions through compositional training.

²https://github.com/mlfoundations/open_clip

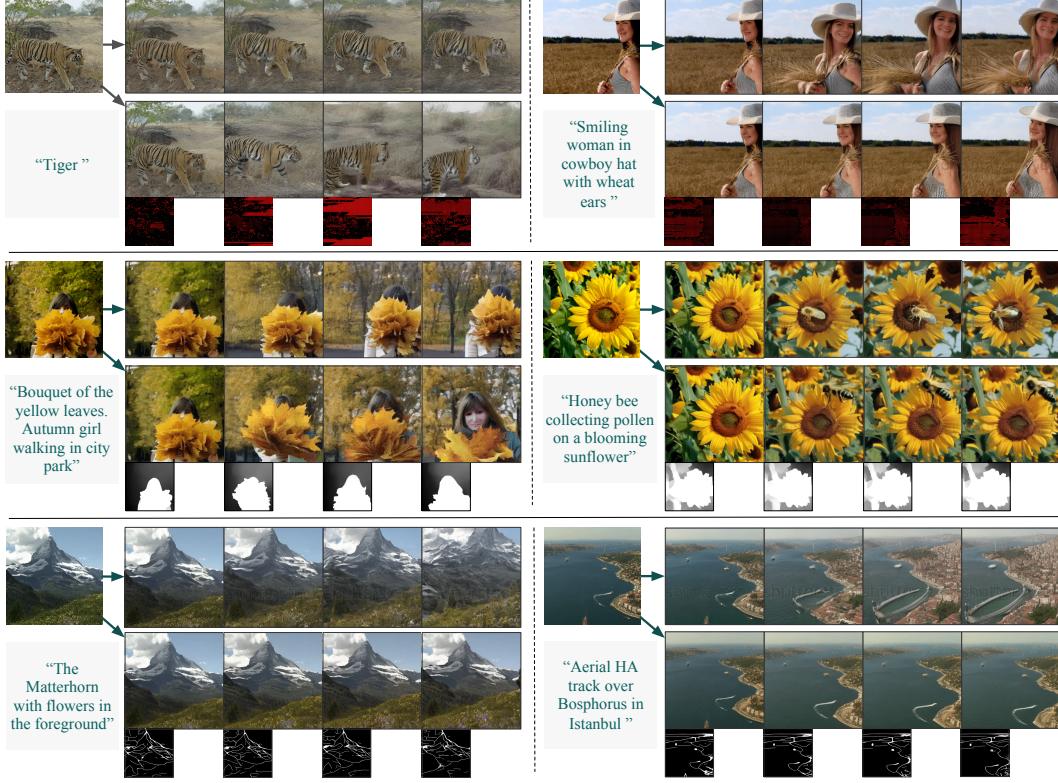


Figure 4: **Compositional image-to-video generation.** We showcase six examples, each displaying two generated videos. The upper video is generated using a given single frame as the spatial condition and a textual condition describing the scene. The lower video is generated by incorporating an additional sequence of temporal conditions to facilitate finer control over the temporally evolving structure.

Inference. During inference, DDIM [73] is employed to enhance the sample quality and improve inference efficiency. We incorporate classifier-free guidance [20] to ensure that the generative results adhere to specified conditions. The generative process can be formalized as:

$$\hat{\epsilon}_\theta(z_t, c, t) = \epsilon_\theta(z_t, c_1, t) + \omega(\epsilon_\theta(z_t, c_2, t) - \epsilon_\theta(z_t, c_1, t)) \quad (2)$$

where ω is the guidance scale; c_1 and c_2 are two sets of conditions. This guidance mechanism extrapolates between two condition sets, placing emphasis on the elements in $(c_2 \setminus c_1)$ and empowering flexible application. For instance, in text-driven video inpainting, c_2 represents the expected caption and a masked video, while c_1 is an empty caption and the same masked video.

4 Experiments

4.1 Experimental setup

Datasets. To optimize VideoComposer, we leverage two widely recognized and publicly accessible datasets: WebVid10M [3] and LAION-400M [47]. WebVid10M [3] is a large-scale benchmark scrapped from the web that contains 10.3M video-caption pairs. LAION-400M [47] is an image-caption paired dataset, filtered using CLIP [40].

Evaluation metrics. We utilize two metrics to evaluate VideoComposer: *i*) To evaluate video continuity, we follow Gen-1 [15] to compute the average CLIP cosine similarity of two consecutive frames, serving as a **frame consistency metric**; *ii*) To evaluate motion controllability, we adopt end-point-error [52, 67] as a **motion control metric**, which measures the Euclidean distance between the predicted and the ground truth optical flow for each pixel.

4.2 Composable video generation with versatile conditions

In this section, we demonstrate the ability of VideoComposer to tackle various tasks in a controllable and versatile manner, leveraging its inherent compositionality. It's important to note that the conditions

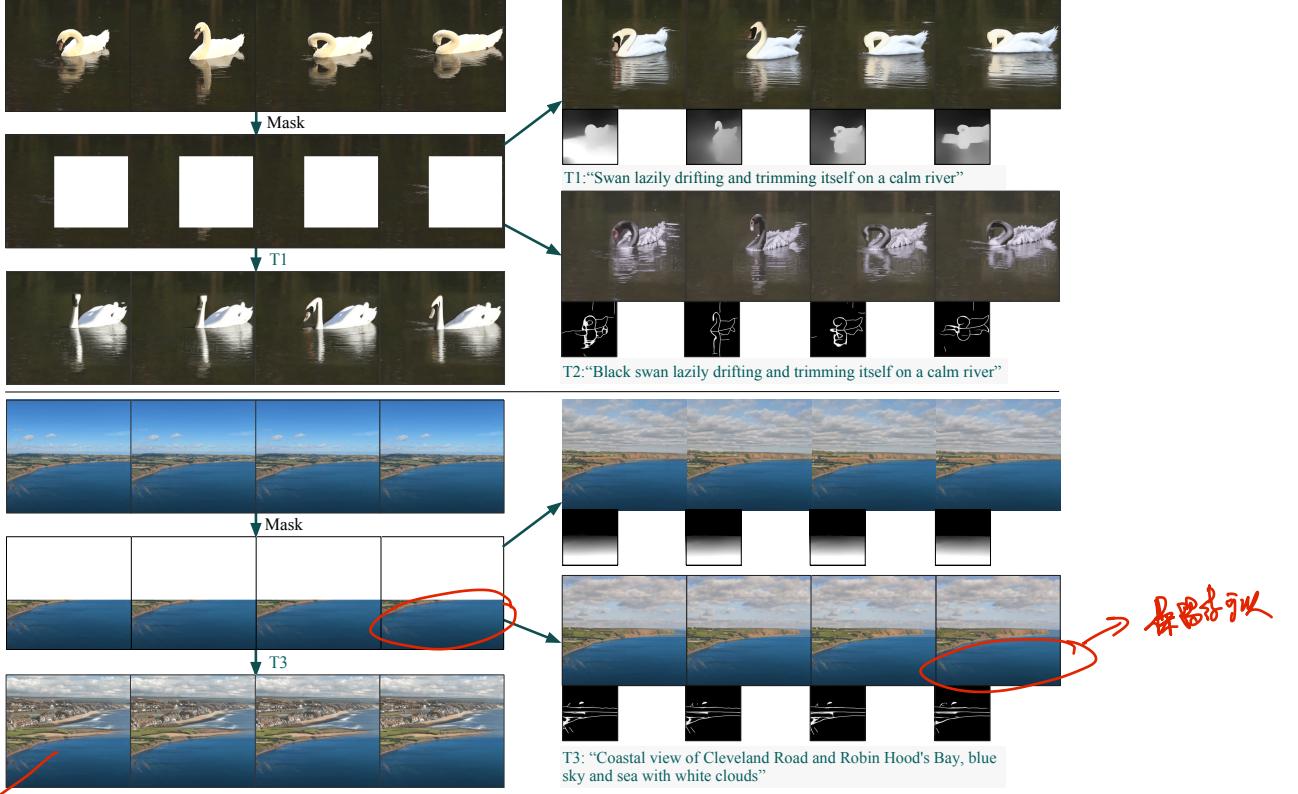


Figure 5: **Compositional video inpainting.** By manually adding masks to videos, VideoComposer can perform video inpainting, facilitating the restoration of the corrupted parts according to textual instructions. Furthermore, by incorporating temporal conditions specifying the visual structure, VideoComposer can perform customized inpainting that conforms to the prescribed structure.

employed in these examples are customizable to specific requirements. We also provide additional results in the supplementary material for further reference.

Compositional Image-to-video generation. Compositional training with a single image endows VideoComposer with the ability of animating static images. In Fig. 4, we present six examples to demonstrate this ability. VideoComposer is capable of synthesizing videos conformed to texts and the initial frame. To further obtain enhanced control over the structure, we can incorporate additional temporal conditions. We observe resultant videos consistently adhere to the given conditions.

Compositional video inpainting. Jointly training with masked video endows the model with the ability of filling the masked regions with prescribed content, as shown in Fig. 5. VideoComposer can replenish the mask-corrupted regions based on textual descriptions. By further incorporating temporal conditions, *i.e.*, depth maps and sketches, we obtain more advanced control over the structure.

Compositional sketch-to-video generation. Compositional training with single sketch empowers VideoComposer with the ability of animating static sketches, as illustrated in Fig. 6. We observe that VideoComposer synthesizes videos conforming to texts and the initial sketch. Furthermore, we observe that the inclusion of mask and style guidance can facilitate structure and style control.

Table 1: **Evaluating the motion controllability.** “Text” and “MV” represent the utilization of text and motion vectors as conditions for generation.

Method	Text	MV	Motion control ↓
w/o STC-encoder	✓		4.03
w/o STC-encoder	✓	✓	2.67
VideoComposer	✓	✓	2.18

4.3 Experimental results of motion control

Quantitative evaluation. To validate superior motion controllability, we utilize the motion control metric. We randomly select 1000 caption-video pairs and synthesize corresponding videos. The results are presented in Tab. 1. We observe that the inclusion of motion vectors as a condition reduce

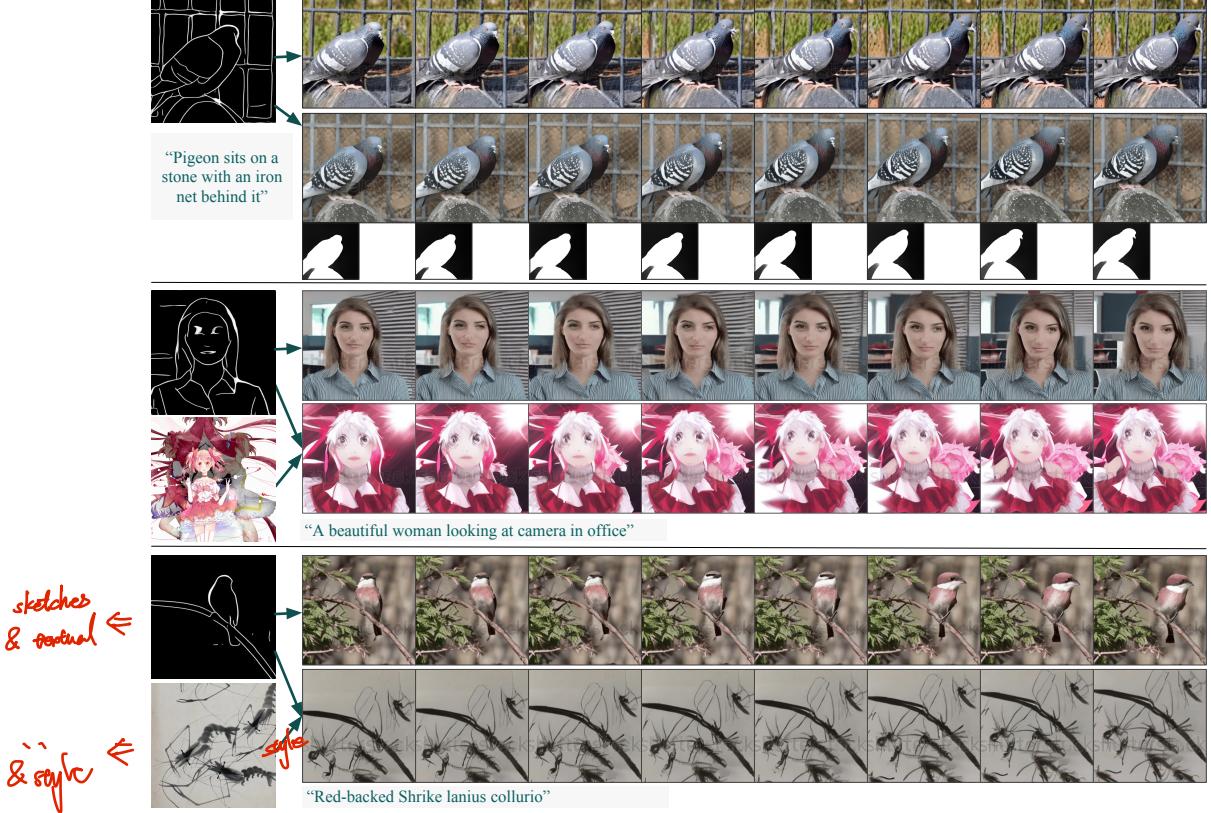


Figure 6: **Compositional sketch-to-video generation.** In the first example, the upper video is generated using text and a single sketch as the conditions, while the lower is generated by using an additional mask sequence for finer control over the temporal patterns. For the last two examples, the **upper video is generated using a single sketch and a textual condition**, while the lower is generated with an additional style from a specified image.

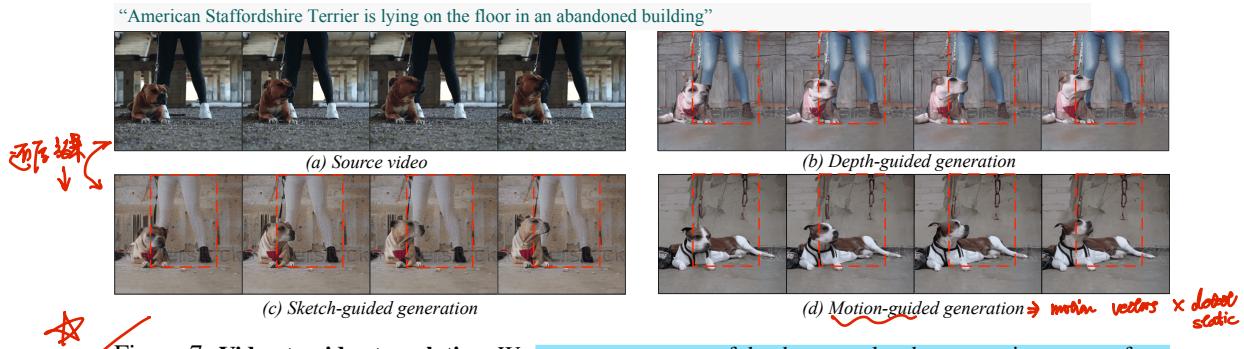


Figure 7: **Video-to-video translation.** We extract a sequence of depth maps, sketches or motion vectors from the **source video**, along with textual descriptions, to perform the translation. By utilizing motion vectors, we achieve **static-background removal**.

the motion control error, indicating an enhancement of motion controllability. The incorporation of STC-encoder further advances the motion controllability.

Motion vectors prioritizing moving visual cues. Thanks to the **nature of motion vectors**, which encode inter-frame variation, static regions within an image are inherently omitted. This prioritization of moving regions facilitates motion control during synthesis. In Fig. 7, we present results of video-to-video translation to substantiate such superiority. We observe that motion vectors exclude the static background, i.e., human legs, a feat that other temporal conditions such as depth maps and sketches cannot accomplish. This advantage lays the foundation for a broader range of applications.

Versatile motion control with motion vectors. Motion vectors, easily derived from hand-crafted strokes, enable more versatile motion control. In Fig. 8, we present visualization comparing CogVideo [24] and VideoComposer. While CogVideo is limited to insufficient text-guided motion

foot2video

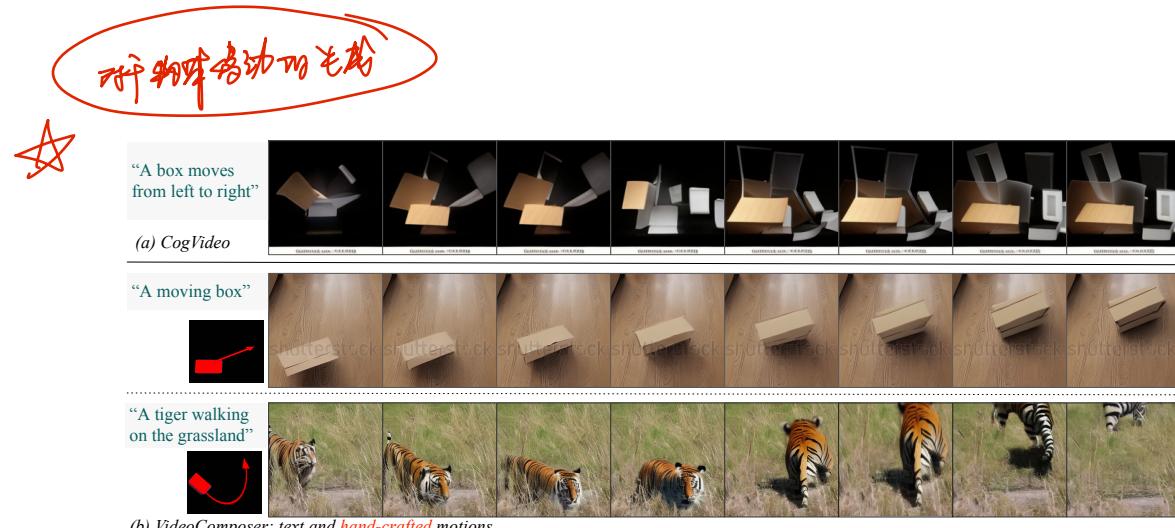


Figure 8: **Versatile motion control using hand-crafted motions.** (a) Limited motion control using CogVideo [24]. (b) Fine-grained and flexible motion control, empowered by VideoComposer.

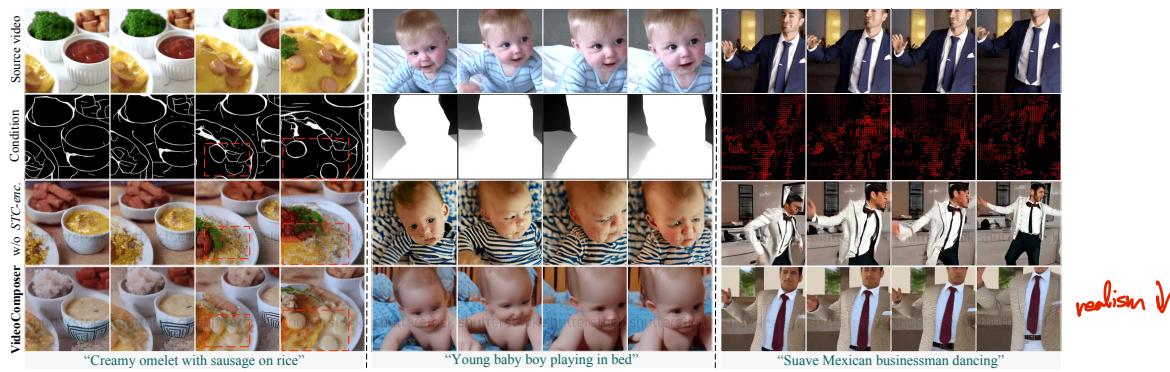


Figure 9: **Qualitative ablation study.** We present three representative examples. The last two rows of videos display generated videos conditioned on a textual condition and one additional temporal condition (*i.e.*, sketches, depth maps or motion vectors). Regions exhibiting deficiencies or fidelity are emphasized within red boxes.

control, VideoComposer expands this functionality by additionally leveraging motion vectors derived from hand-crafted strokes to facilitate more flexible and precise motion control.

4.4 Ablation study

In this subsection, we conduct qualitative and quantitative analysis on VideoComposer, aiming to demonstrate the effectiveness of incorporating STC-encoder.

Quantitative analysis. In Tab. 2, we present the frame consistency metric computed on 1000 test videos. We observe that incorporating STC-encoder augments the frame consistency, which we attribute to its temporal modeling capacity. This observation holds for various temporal conditions such as sketches, depth maps and motion vectors.

Qualitative analysis. In Fig. 9, we exemplify the usefulness of STC-encoder. We observe that in the first example, videos generated by VideoComposer without STC-encoder generally adhere to the sketches but omit certain detailed information, such as several round-shaped ingredients. For the left two examples, VideoComposer without STC-encoder generates videos that are structurally inconsistent with conditions. We can also spot the noticeable defects in terms of human faces and poses. Thus, all the above examples can validate the effectiveness of STC-encoder.

Table 2: **Quantitative ablation study of STC-encoder.** “Conditions” denotes the conditions utilized for generation.

Method	Conditions	Frame consistency \uparrow
w/o STC-encoder	Text and sketch sequence	0.910
VideoComposer	Text and sketch sequence	0.923
w/o STC-encoder	Text and depth sequence	0.922
VideoComposer	Text and depth sequence	0.928
w/o STC-encoder	Text and motion vectors	0.915
VideoComposer	Text and motion vectors	0.927

5 Conclusion

In this paper, we present **VideoComposer**, which aims to explore the compositionality within the realm of video synthesis, striving to obtain a flexible and controllable synthesis system. In particular, we explore the use of temporal conditions for videos, specifically motion vectors, as powerful control signals to provide guidance in terms of temporal dynamics. An STC-encoder is further designed as a unified interface to aggregate the spatial and temporal dependencies of the sequential inputs for inter-frame consistency. Our experiments, which involve the combination of various conditions to augment controllability, underscore the pivotal role of our design choices and reveal the impressive creativity of the proposed **VideoComposer**.

References

- [1] Text to video synthesis in modelscope. 2023. URL <https://modelscope.cn/models/damo/text-to-video-synthesis/summary>.
- [2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6836–6846, 2021.
- [3] Max Bain, Arsha Nagrani, Gülcin Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pages 1728–1738, 2021.
- [4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.
- [5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.
- [6] Andreas 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*, pages 22563–22575, 2023.
- [7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.
- [8] Jiawei Chen and Chiu Man Ho. Mm-vit: Multi-modal video transformer for compressed video action recognition. In *WACV*, pages 1910–1921, 2022.
- [9] Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023.
- [10] Feng Cheng and Gedas Bertasius. Tallformer: Temporal action localization with a long-memory transformer. In *ECCV*, pages 503–521, 2022.
- [11] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.
- [12] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, pages 16344–16359, 2022.
- [13] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.
- [14] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021.
- [15] Patrick Esser, Johnathan Chiu, Parmida Atighchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *arXiv preprint arXiv:2302.03011*, 2023.
- [16] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. In *NeurIPS*, pages 35946–35958, 2022.
- [17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, pages 139–144, 2020.

- [18] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *arXiv preprint arXiv:2205.11495*, 2022.
- [19] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.
- [20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, pages 6840–6851, 2020.
- [22] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [23] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.
- [24] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.
- [25] Yaosi Hu, Zhenzhong Chen, and Chong Luo. Lamd: Latent motion diffusion for video generation. *arXiv preprint arXiv:2304.11603*, 2023.
- [26] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023.
- [27] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022.
- [28] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023.
- [29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [30] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. *arXiv preprint arXiv:2303.04761*, 2023.
- [31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [32] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *CVPR*, 2023.
- [33] Eyal Molad, Eliah Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023.
- [34] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.
- [35] Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *CVPR*, pages 18444–18455, 2023.
- [36] 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. *arXiv preprint arXiv:2112.10741*, 2021.
- [37] Katsunori Ohnishi, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Hierarchical video generation from orthogonal information: Optical flow and texture. In *AAAI*, volume 32, 2018.
- [38] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *arXiv preprint arXiv:2302.03027*, 2023.

- [39] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5533–5541, 2017.
- [40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.
- [41] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, pages 5485–5551, 2020.
- [42] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [43] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1623–1637, 2020.
- [44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.
- [45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.
- [46] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, pages 36479–36494, 2022.
- [47] 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 preprint arXiv:2111.02114*, 2021.
- [48] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *CVPR*, pages 1268–1277, 2019.
- [49] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.
- [50] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICCV*, pages 2256–2265, 2015.
- [51] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *ICCV*, pages 5117–5127, 2021.
- [52] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020.
- [53] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022.
- [54] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *NeurIPS*, pages 11287–11302, 2021.
- [55] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, pages 1510–1517, 2017.
- [56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.
- [57] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016.
- [58] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021.
- [59] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022.

- [60] Wen Wang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023.
- [61] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. G3an: Disentangling appearance and motion for video generation. In *CVPR*, pages 5264–5273, 2020.
- [62] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, pages 3164–3172, 2015.
- [63] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *CVPR*, pages 6026–6035, 2018.
- [64] 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 preprint arXiv:2104.14806*, 2021.
- [65] 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*, pages 720–736. Springer, 2022.
- [66] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. *arXiv preprint arXiv:2212.05034*, 2022.
- [67] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, pages 8121–8130, 2022.
- [68] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022.
- [69] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223, 2007.
- [70] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, pages 7094–7103, 2019.
- [71] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, pages 2718–2726, 2016.
- [72] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.
- [73] Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*, 2022.
- [74] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2914–2923, 2017.
- [75] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.



Figure A10: **Compositional sketch sequence-to-video generation.** We showcase five examples, each displaying a video generated from a sequence of sketches and a textual description. The final example additionally incorporates a style condition.

Appendix

In this Appendix, we first elaborate on more implementation details (Appendix A) and present more experimental results (Appendix B). Next, we provide a section of discussion (Appendix C) on the limitations and potential societal impact of VideoComposer.

A More implementation details

Pre-training details. We adopt AdamW [31] as the default optimizer with a learning rate set to 5×10^{-5} . In total, VideoComposer is pre-trained for 400k steps, with the first and second stage being pre-trained for 132k steps and 268k steps, respectively. In terms of two-stage pre-training, we allocate one fourth of GPUs to perform image pre-training, while the rest of the GPUs are dedicated to video pre-training. We use center crop and randomly sample video frames to compose the video input whose $F = 16$, $H = 256$ and $W = 256$. During the second stage pre-training, we adhere to [26], using a probability of 0.1 to keep all conditions, a probability of 0.1 to discard all conditions, and an independent probability of 0.5 to keep or discard a specific condition. Regarding the use of

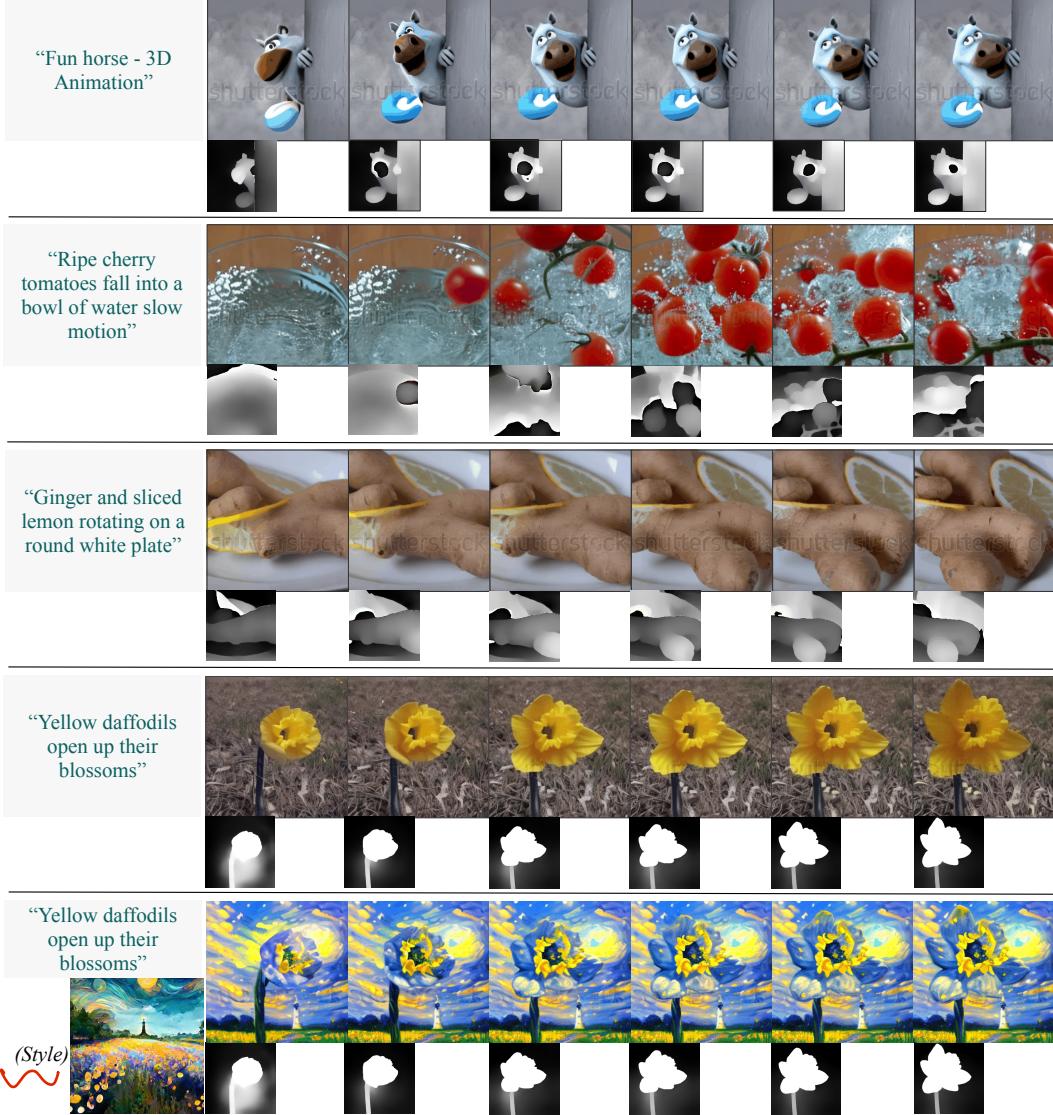


Figure A11: **Compositional depth sequence-to-video generation.** We showcase five examples, each displaying a video generated from a sequence of depth maps and a textual description. The final example additionally incorporates a style condition.

WebVid10M [3], we sample frames from videos using various strides to ensure frame rate equal to 4, aiming to maintain a consistent frame rate.

The structure of 3D UNet as $\epsilon_\theta(\cdot, \cdot, t)$. To leverage the benefits of LDMs pre-trained on web-scale image data, *i.e.*, Stable Diffusion³, we extend the 2D UNet to a 3D UNet by introducing temporal modeling layers. Specifically, within a single UNet block, we employ four essential building blocks: spatial convolution, temporal convolution, spatial transformer and temporal transformer. The spatial blocks are inherited from LDMs, while temporal processing blocks are newly introduced. Regarding temporal convolution, we stack four convolutions with $1 \times 1 \times 3$ kernel, ensuring the temporal receptive field is ample for capturing temporal dependencies; regarding temporal transformer, we stack one Transformer layer and accelerate its inference using flash attention [12].

vivit

³<https://github.com/Stability-AI/stablediffusion>

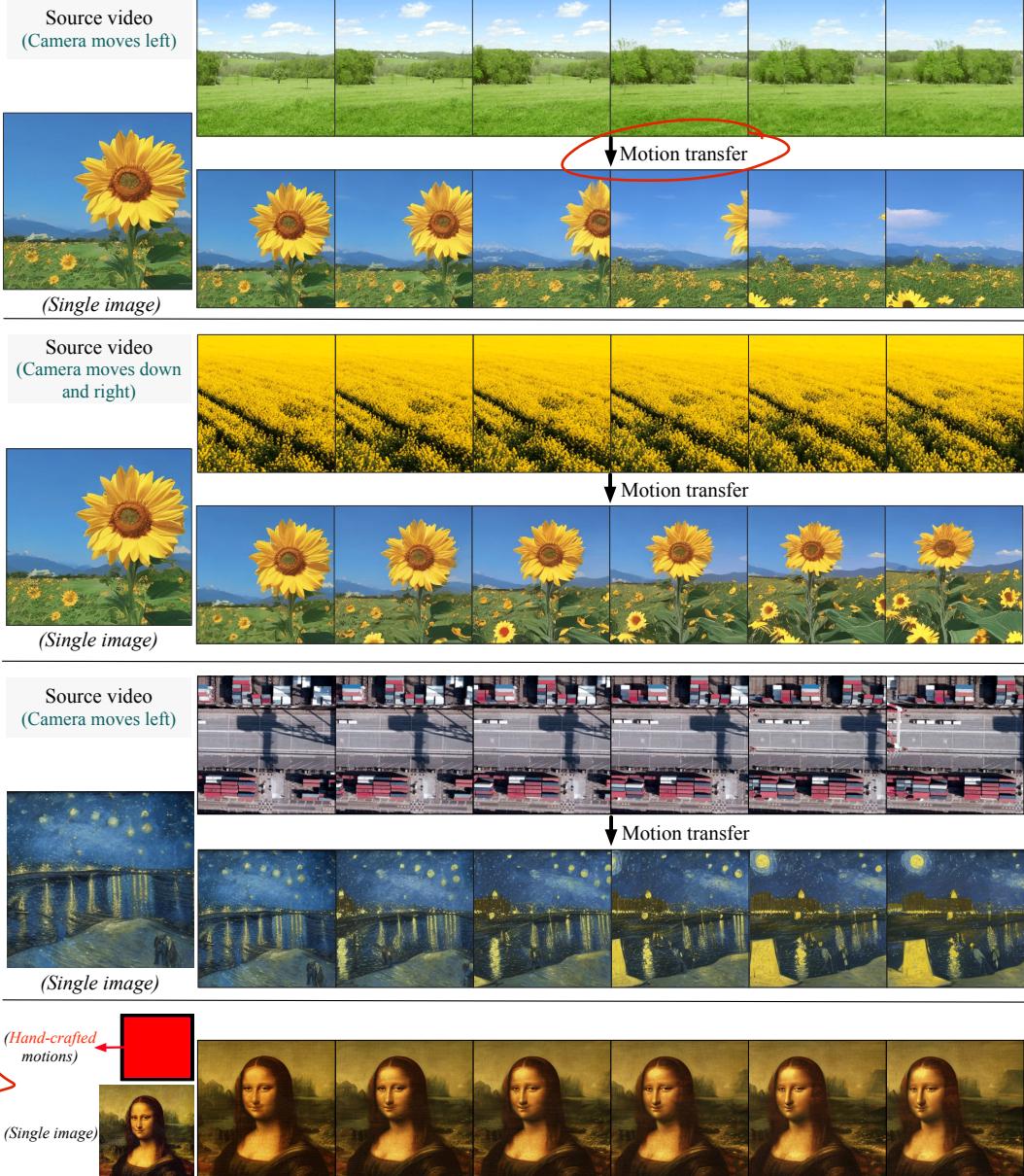


Figure A12: Motion transfer. We showcase four examples, each displaying a video generated from a single image and motions. In the first three examples, we transfer the motion patterns in a source video to the generated video by extracting and utilizing motion vectors. The final example incorporates hand-crafted motions instead.

B More experimental results

In this subsection, we aim to provide additional experiments that complement the findings presented in the main paper and showcase more versatile controlling cases.

Compositional sketch sequence-to-video generation. Compositional training with sketch sequences enables VideoComposer to possess the ability of generation videos adhering to sketch sequences. This generation paradigm lays more emphasis on the structure control, which differs from compositional sketch-to-video generation and can be viewed as video-to-video translation. In Fig. A10, we exemplify this capacity. We observe videos' fidelity to the provided conditions, including texts, sketches and style.

Compositional depth sequence-to-video generation. Conducting compositional training with depth sequences allows VideoComposer to effectively generate videos in accordance with depth sequences.

Table A3: Text-to-video generation performance on MSR-VTT.

Method	Zero-shot	FVD ↓	CLIPSIM ↑
GODIVA [64]	No	-	0.2402
Nüwa [65]	No	-	0.2439
CogVideo (Chinese) [24]	Yes	-	0.2614
CogVideo (English) [24]	Yes	1294	0.2631
MagicVideo [75]	Yes	1290	-
Make-A-Video [49]	Yes	-	0.3049
Video LDM [6]	Yes	-	0.2929
Text-to-video pre-training (First stage)	Yes	803	0.2876
VideoComposer	Yes	580	0.2932

In Fig. A11, we illustrate this capability. Videos generated with VideoComposer faithfully adhere to the given conditions, including text prompts, depth maps, and style.

Motion transfer. Incorporating motion vectors as a composition of videos enables motion transferability. In Fig. A12, we conduct experiments to demonstrate such capability. Through utilizing hand-crafted motion vectors or motion vectors extracted from off-the-shelf source videos, we can transfer the motion patterns to synthesized videos.

Text-to-video generation performance. Although VideoComposer is not specifically tailored for text-to-video generation, its versatility allows VideoComposer to perform the traditional text-to-video generation task effectively. In Tab. A3, we follow the evaluation settings in Video LDM [6] to adopt Fréchet Video Distance (FVD) and CLIP Similarity (CLIPSIM) as evaluation metrics and present the quantitative results of text-to-video generation on MSR-VTT dataset compared to other existing methods. The results in the table demonstrate that VideoComposer achieves competitive performance compared to state-of-the-art text-to-video approaches. In addition, VideoComposer outperforms our first-stage text-to-video pre-training, demonstrating that VideoComposer can achieve compositional generation without sacrificing its capability of text-to-video generation. In the future, we aim to advance VideoComposer by leveraging stronger text-to-video models, enabling more flexible and controllable video synthesis.

C Discussion

Limitations. Due to the absence of a publicly available large-scale and high-quality dataset, we have developed VideoComposer using the watermarked WebVid10M dataset. As a result, the synthesized videos contain watermarks, which affect the generation quality and lead to less visually appealing results. Furthermore, in order to reduce the training cost, the resolution of the generated videos is limited to 256×256 . Consequently, some delicate details might not be sufficiently clear. In the future, we plan to utilize super-resolution models to expand the resolution of the generated videos to improve the visual quality.

Potential societal impact. VideoComposer, as a generic video synthesis technology, possesses the potential to revolutionize the content creation industry, offering unprecedented flexibility and creativity, and hence, promising significant commercial advantages. Traditional content creation processes are labor- and cost-intensive. VideoComposer could alleviate these burdens by enabling designers to manipulate subjects, styles, and scenes through instructions spanning human-written text, and styles and subjects sourced from other images. Moreover, VideoComposer could potentially revolutionize education industry by creating unique and customized video scenarios for teaching complex concepts.

However, it's necessary to note that VideoComposer also represents a dual-use technology with inherent risks to society. As with prior generative foundation models, such as Imagen Video [22] and Make-A-Video [49], VideoComposer inherits the implicit knowledge embedded within the pre-trained model (*i.e.*, StableDiffusion) and the pre-trained dataset (*i.e.*, WebVid and LAION). Potential issues include but not limited to the propagation of social biases (such as gender and racial bias) and the creation of offensive content.

Given that VideoComposer is a research-oriented project aimed at investigating compositionality in diffusion-based video synthesis, our primary focus lies in scientific exploration and proof of

concept. If VideoComposer is deployed beyond the scope of research, we strongly recommend several precautionary measures to ensure its responsible and ethical use: **(i)** Rigorous evaluation and oversight of the deployment context should be conducted; **(ii)** Necessary filtering of prompts and generated content should be implemented to prevent misuse.