Mind the Time: Temporally-Controlled Multi-Event Video Generation

Supplementary Material

We highly encourage the readers to check out our project page for video results of baselines and MinT.

A. Details on Rotary Position Embedding

A.1. Derivation of RoPE

We detail the derivation conducted in Sec. 3.1 of the main paper. Our derivation mostly follows [31, 38, 45] and only provides an intuitive motivation for our method. We refer readers to their papers for more rigorous results.

Given a query vector $\boldsymbol{q}_n = \left[q_n^{(0)}, \cdots, q_n^{(d-1)}\right] \in \mathbb{R}^d$ at index n and a key vector $\boldsymbol{k}_m = \left[k_m^{(0)}, \cdots, k_m^{(d-1)}\right] \in \mathbb{R}^d$ at index m, to apply RoPE, it first groups every two elements in them, and make them complex numbers as:

$$\begin{aligned} \bar{\boldsymbol{q}}_n &= \left[\bar{q}_n^{(0)}, \cdots, \bar{q}_n^{(d/2-1)}\right], \ \ \bar{q}_n^{(l)} = q_n^{(2l)} + i q_n^{(2l+1)}, \\ \bar{\boldsymbol{k}}_m &= \left[\bar{k}_m^{(0)}, \cdots, \bar{k}_m^{(d/2-1)}\right], \ \ \bar{k}_m^{(l)} = k_m^{(2l)} + i k_m^{(2l+1)}. \end{aligned} \tag{1}$$

Then, RoPE rotates each complex number by an angle θ_l, which is achieved as element-wise multiplication:

$$\tilde{\boldsymbol{q}}_n = \bar{\boldsymbol{q}}_n \odot e^{in\boldsymbol{\theta}}, \ \ \tilde{\boldsymbol{k}}_m = \bar{\boldsymbol{k}}_m \odot e^{im\boldsymbol{\theta}}, \tag{2}$$

where $\boldsymbol{\theta}$ is determined by the position l of each element in a vector. We follow prior works [25, 45] to use:

$$\boldsymbol{\theta} = \left[\theta_0, \cdots, \theta_{d/2-1}\right], \ \ \theta_l = 10000^{-2l/d}. \tag{3}$$

Eq. (3) indicates that each θ_l is a fixed value, and thus the rotation results in Eq. (2) is only decided by the vectors' index n and m. This is why in the main paper, we only consider a single θ_{base} instead of $\boldsymbol{\theta}$ for different elements.

We can now calculate the attention between $\tilde{\boldsymbol{q}}_n$ and $\tilde{\boldsymbol{k}}_m$:

$$\begin{aligned} A_{n,m} &= \text{Re}\left[\langle \tilde{\boldsymbol{q}}_n, \tilde{\boldsymbol{k}}_m \rangle\right] \\ &= \text{Re}\left[(\bar{\boldsymbol{q}}_n e^{in\boldsymbol{\theta}}) \cdot (\bar{\boldsymbol{k}}_m e^{im\boldsymbol{\theta}})^*\right] \\ &= \text{Re}\left[\sum_{l=0}^{d/2-1} (\bar{q}_n^{(l)} e^{in\theta_l})(\bar{k}_m^{(l)*} e^{-im\theta_l})\right] \\ &= \text{Re}\left[\sum_{l=0}^{d/2-1} \bar{q}_n^{(l)} \bar{k}_m^{(l)*} e^{i(n-m)\theta_l}\right] \\ &= \sum_{l=0}^{d/2-1} \left(q_n^{(2l)} k_m^{(2l)} + q_n^{(2l+1)} k_m^{(2l+1)}\right) \cos\left((n-m)\theta_l\right) + \\ &\qquad \left(q_n^{(2l)} k_m^{(2l+1)} - q_n^{(2l+1)} k_m^{(2l)}\right) \sin\left((n-m)\theta_l\right). \end{aligned} \tag{4}$$

Since we are interested in the bias introduced by RoPE in attention, we assume all queries $\boldsymbol{q}_n$ and all keys $\boldsymbol{k}_m$ are the same, so that their attention values without RoPE is the same. Empirically, we find that query and key vectors indeed have similar values in our DiT due to the use of Layer Normalization [5]. Thanks to the periodic property of $\sin(\cdot)$ and $\cos(\cdot)$, from Eq. (4), we have $A_{n,m} = A_{m,n}$, i.e., , the attention bias between $\boldsymbol{q}_n$ and $\boldsymbol{k}_m$ is only affected by the absolute distance between the two vectors, $|n-m|$.

The original RoPE paper [45] proves that the upper bound of $A_{n,m}$ decays monotonically with the distance $|n-m|$ until around 40. Since the RoPE used in the temporal cross-attention layer only encodes vectors using the temporal frame index, and our video DiT is trained on video tokens with up to around 50 frames, we roughly preserve the monotonicity of RoPE. As we will see in Appendix A.3, while there are some fluctuations of $A_{n,m}$ in the long range, the long-term decay makes their values significantly low.

A.2. Proof of the Property of ReRoPE

In Sec. 3.2 of the main paper, we propose to rescale all events to a fixed length L. For a timestamp t lying in the n-th event, we transform it as:

$$\begin{aligned} \tilde{t} = \frac{(t - t_n^{\text{start}})L}{t_n^{\text{end}} - t_n^{\text{start}}} + (n-1)L, \ \ \text{s.t. } t_n^{\text{start}} \le t \le t_n^{\text{end}}, \\ \tilde{t}_n^{\text{mid}} = L/2 + (n-1)L. \end{aligned} \tag{5}$$

After transformation, the distance between a video token in the n-th event and the middle timestamps of this event is:

$$\left|\tilde{t} - \tilde{t}_n^{\text{mid}}\right| = \left|\frac{t - t_n^{\text{start}}}{t_n^{\text{end}} - t_n^{\text{start}}} - \frac{1}{2}\right| L. \tag{6}$$

Next, we prove that it satisfies the three desired properties of the temporal cross-attention:

(i) *For video tokens within the time span of an event, they should attend the most to the text embedding of this event.*

Proof For $t_n^{\text{start}} \le t \le t_n^{\text{end}}$, we have:

$$-\frac{1}{2} \le \left(\frac{t - t_n^{\text{start}}}{t_n^{\text{end}} - t_n^{\text{start}}} - \frac{1}{2}\right) \le \frac{1}{2}, \tag{7}$$

thus, $\left|\tilde{t} - \tilde{t}_n^{\text{mid}}\right| \le L/2$. For any m-th event with $m \ne n$, its distance to this video token is:

$$\left|\tilde{t} - \tilde{t}_m^{\text{mid}}\right| = \left|\left(\frac{t - t_n^{\text{start}}}{t_n^{\text{end}} - t_n^{\text{start}}} - \frac{1}{2}\right) + (n-m)\right| L. \tag{8}$$

Since $|n-m| \ge 1$, we get:

$$\left|\left(\frac{t - t_n^{\text{start}}}{t_n^{\text{end}} - t_n^{\text{start}}} - \frac{1}{2}\right) + (n-m)\right| \ge \frac{1}{2}. \tag{9}$$

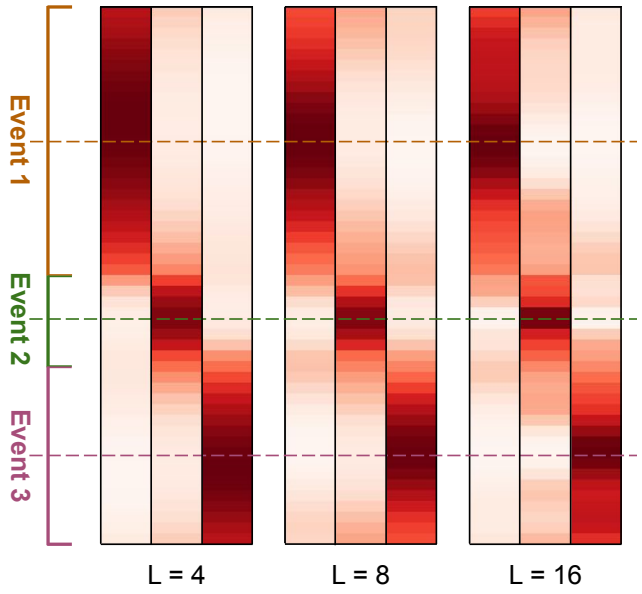

Figure 1. **Comparison of ReRoPE with different rescaling length L.** We use the same random vector for video tokens and text embeddings to only visualize the bias introduced by positional encoding. We visualize the case where videos have a temporal dimension of 50, and there are three temporal captions.

Therefore, we have:

$$\left|\tilde{t} - \tilde{t}_m^{\text{mid}}\right| \geq L/2 \geq \left|\tilde{t} - \tilde{t}_n^{\text{mid}}\right|, \ \ \forall\, m \neq n. \tag{10}$$

Since RoPE attention decays monotonically with the distance, we reach the property.

(ii) *For an event, the attention weight should peak with the video token at the midpoint of its time span, and then decrease towards the boundary of the event.*

Proof When a video token is at the midpoint of an event, we have $\tilde{t} - \tilde{t}_n^{\text{mid}} = 0$. Thus, the attention weight will be the highest. In addition, Eq. (6) increases when t goes from t_n^{mid} to t_n^{start} or t_n^{end}, leading to a decreased weight.

(iii) *The video token at the transition point between two events should attend equally to their text embeddings.*

Proof For $t = t_n^{\text{start}}$ or t_n^{end}, we always have the distance $\left|\tilde{t} - \tilde{t}_n^{\text{mid}}\right| = L/2$. Thus, the attention value is the same for video tokens at event borders. This is only possible in ReRoPE as we rescale all events to have the same length.

A.3. Visualizations of ReRoPE

In Sec. 4.5 of the main paper, we show that using different rescaling length L in ReRoPE leads to similar results. Fig. 1 visualizes the cross-attention map using $L = 4$, 8, and 16. The three attention maps are indeed similar, which explains why the performances are close. We also notice that with a higher L, the attention map of each event becomes more concentrated. It would be an interesting direction to study its effect in depth, which we leave for future work.

B. Detailed Experimental Setup

In this section, we provide full details on the datasets, baselines, evaluation settings, and the training and inference implementation details of our model.

B.1. Training Data

Before this work, there are mainly two types of video datasets that annotate open-set event captions and their precise timestamps. One such field is dense video captioning [20, 23, 58]. However, these datasets are limited in scale (usually fewer than 10k videos), which makes it impossible to fine-tune a large-scale video generator. Another field is video chaptering [54]. However, the temporal captions here are high-level chapter segmentation, where each annotated event is usually longer than one minute. This is too long for current video diffusion models to be trained on.

Since our model requires large-scale and fine-grained video event annotations, we manually source videos from existing datasets [9, 53] and annotate them, resulting in around 200k videos. To condition the model on scene cuts, we run TransNetV2 [44] to detect scene boundaries on annotated videos with a confidence threshold of 0.5.

Fig. 2 present some basic statistics of our dataset. While our training videos have varying lengths, the number of events per video and the average event length are similar, which makes model training easier.

Data processing. The training dataset contains videos of different lengths, resolutions, and aspect ratios. Following common practice [39, 57], we use data bucketing, which groups videos into a fixed set of sizes. Overall, we sample videos up to 512 resolution, and 10s during training. We pad to or subsample 4 temporal captions for batch training.

B.2. Evaluation Datasets

HoldOut. We randomly sample 2k videos from our training data as a holdout testing set. The prompts here are in-distribution with a minimum gap to training data.

StoryBench [7] consists of videos collected from DiDeMo [4], Oops [12], and UVO [51] datasets. It annotates each video with a background caption and one or more temporal captions similar to our format. We treat their background caption as the global caption in our setting, showing our model's generalization to out-of-distribution prompts. We filter out videos with only a single event, leading to around 3k testing samples.

VBench [21] is a comprehensive benchmark that tests different aspects of a video generation model. It has 16 evaluation dimensions, each with a carefully collected list of text prompts. Since we are interested in the dynamics of generated videos, we choose the *Dynamic Degree* dimension, which provides 72 prompts. Following the official evaluation protocol, we run each model to generate 5 videos using each prompt with 5 random seeds.

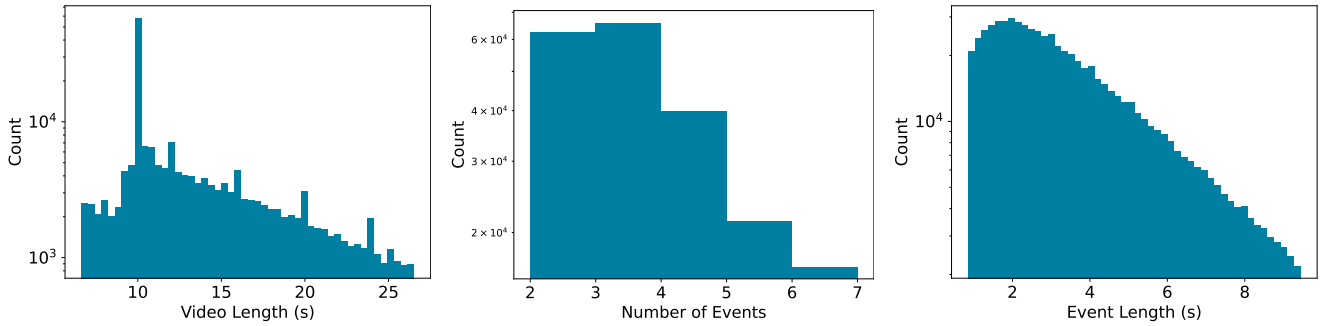

Figure 2. **Basic statistics of our training dataset.** We show the distribution of video length, the number of events per video, and the length of individual events. Most videos contain 2 to 4 events, and most events are under 5s.

B.3. Baselines

We only compare to methods that can generate smoothly connected events and have released their code.

MEVG [35] is the state-of-the-art multi-event video generation method. Given a sequence of event prompts, it generates the first video clip using the first event prompt. Then, to generate the next event, it runs DDIM inversion [43] to obtain the inverted noise latent of the previous clip, which is used to initialize the current noise latent. Then, when denoising the current latent, it also introduces several losses to enforce latent at adjacent frames to be similar. Original MEVG builds upon LVDM [15] and VideoCrafter [8] which are outdated. For a fair comparison, we re-implement it based on our base model. As far as we know, there is no prior work on inverting a rectified flow model, so we follow DDIM inversion to implement RF inversion which achieves similar results. To handle both global and temporal captions, we generate the first clip by concatenating the global caption and the first temporal caption. We keep other losses and hyper-parameters the same as in MEVG[1].

AutoReg. We fine-tune our base model to support initial frame conditioned video generation. The method is similar to MEVG, where we generate one event based on its own caption and the last frame of the previous clip.

Concat is a naive baseline that simply concatenates the global caption and all temporal captions to form a long prompt, and generates a video from it.

Remark. Since both MEVG and AutoReg are autoregressive methods, they can only generate fixed-length videos for each event. To enable comparison, we simply assume that the testing events all have the same duration when computing metrics. For Concat, it cannot separate the generation of different events. We thus assume all events are uniformly distributed in the generated video.

B.4. Evaluation Metrics

We identify three key aspects in multi-event text-to-video generation: visual quality, event text alignment, and event transition smoothness. We report common metrics such as FID [18], FVD [47] for visual quality, and per-frame CLIP-score [17, 41] for text alignment. We have tried more advanced metrics such as X-CLIP-score [34], but found it to perform similarly as CLIP-score.

It is well-known that traditional automatic metrics are not aligned with human perceptions. Recent works show that fine-tuning multi-modal LLMs on human feedback data can lead to more human-aligned video quality assessment metrics [14]. We take the state-of-the-art method VideoScore which outputs five scores for a video. We use the *visual quality* and *dynamic degree* output for visual quality, the *text-to-video alignment* output for text alignment, and the *temporal consistency* output for event transition smoothness. We further run TransNetV2 [44] to compute the average number of cuts in generated videos to measure event transition smoothness.

For visual quality and event transition smoothness, we compute relevant metrics on the entire video. We have also computed the visual quality of each event, and found it to be positively correlated with video-level results. For text alignment, since we care about event generation, we take the start and end timestamps of each event, crop out a sub-clip from the generated video, and compute metrics between this sub-clip and the corresponding event prompt.

B.5. Implementation Details

Base model. Our base text-to-video generator adopts the latent Diffusion Transformer framework [37]. It leverages a MAGVIT-v2 [56] as the autoencoder and a deep cascade of DiT blocks as the denoising backbone. The autoencoder is similar to the one in CogVideoX [55], which downsamples the spatial dimensions by 8× and the temporal dimension by 4×. Our backbone has 32 DiT blocks. Each block is similar to the one in Open-Sora [25], which consists of a 3D self-attention layer running on all video tokens, a cross-attention layer between video tokens and T5 text embeddings [42] of the input prompt, and an MLP. We do not use absolute positional encoding on video tokens. Instead, we apply RoPE in self-attention, which is factorized into spatial and temporal axes, similar to [25]. Finally, we use FlashAttention [11] in both self-attention and cross-attention.

[1] MEVG did not release the code at the time of paper submission. We obtain the official code from authors through private email communication.

Figure 3. **Qualitative comparisons of T2V.**

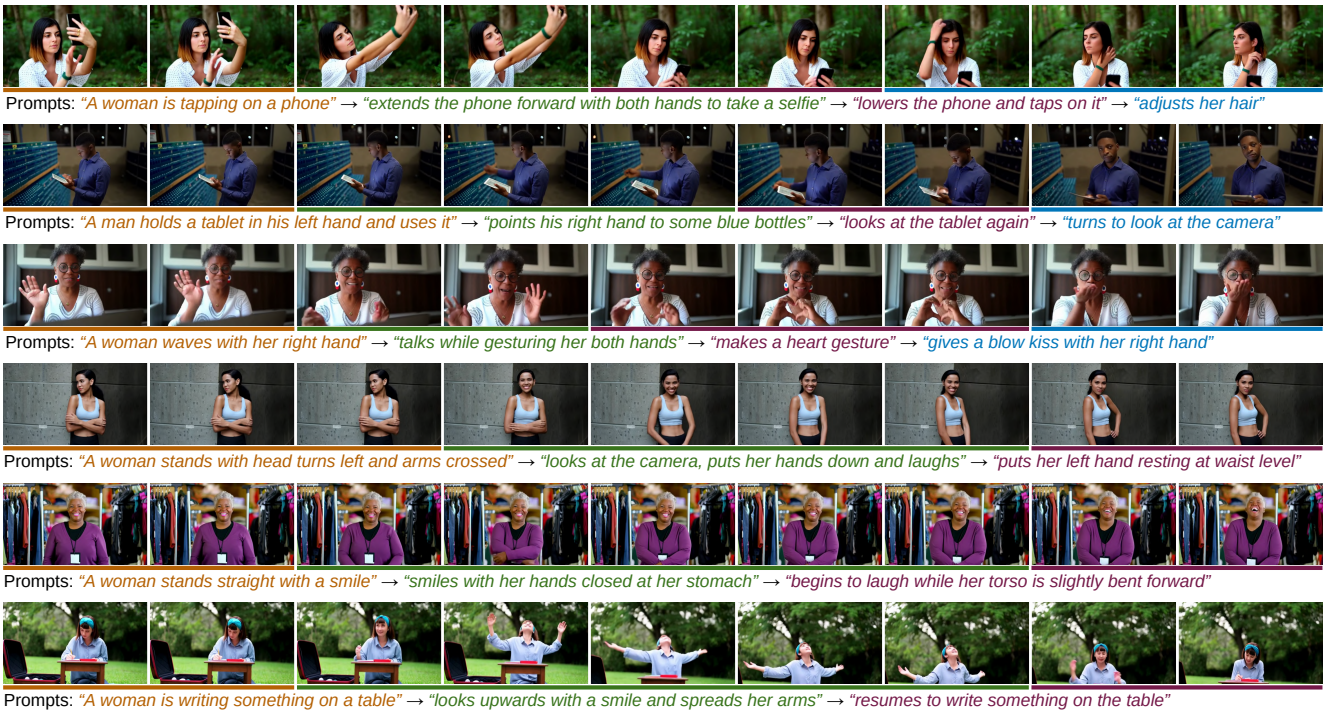

Figure 4. **More T2V results from MinT.** Please see our project page for more results.

The base model adopts the rectified flow training objective [29, 30]. We follow Stable Diffusion 3 [13] to choose the sampling parameters for the diffusion process.

MinT model. We fine-tune MinT from the base model to enable temporal caption control. We copy weights from the original cross-attention layer to initialize our added temporal cross-attention layer to accelerate convergence, since both layers take in the same text modality. Following prior works [26], we introduce a scaling factor that is initialized as 0, and we pass it through a $\tanh(\cdot)$ activation to multiply with the temporal cross-attention layer output. Such a design has been shown to stabilize model training.

Training. We use AdamW [33] to fine-tune the entire model with a batch size of 512 for 15k steps. We use a low learning rate of 1×10^{-5} for the pre-trained weights, and a higher one of 1×10^{-4} for the added weights. Both learning rates are linearly warmed up in the first 1k steps and stay constant. A gradient clipping of 0.05 is applied to stabilize training. To apply classifier-free guidance (CFG) [19], we randomly drop the text embedding of global and temporal captions (i.e., setting them as zeros) with a probability of 10%. Notice that when dropping the temporal captions, we drop all events together and also set the event timestamps to zeros. We implement our model using PyTorch [36] and conduct training on NVIDIA A100 GPUs.

Inference. We use the rectified flow sampler [30] with 256 sampling steps and a classifier-free guidance [19] scale of 8 to generate videos. We also use interval guidance [24] in CFG to mitigate the oversaturation issue, which only applies CFG between [25, 100] sampling steps. We have tried using separate CFG for global and temporal captions similar to in [6], but did not find it to improve results.

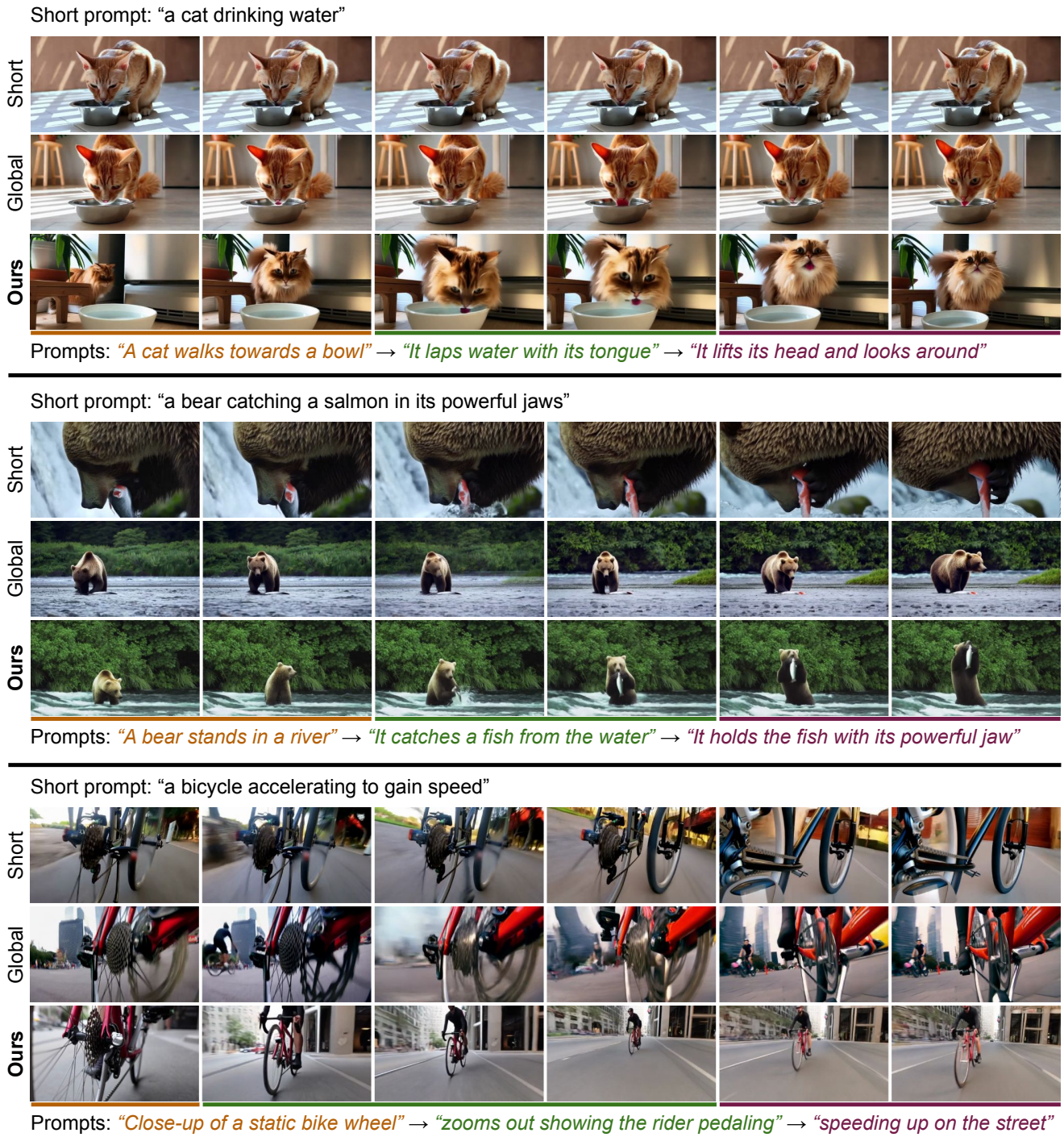

Figure 5. **Prompt enhancement results on VBench.** We can generate more interesting videos from a simple prompt. This highlights the flexible dynamics control ability brought by the temporal captions. Please see our project page for video results.

C. More Results

C.1. More Qualitative Results on T2V

Fig. 3 presents more qualitative comparisons with baselines. Concat only generates the woman writing on a paper while ignoring the subsequent events. AutoReg is able to synthesize a smooth transition between the first and the second event, but it fails to generate the third event. This is because conditioning on generated frames leads to video stagnation and results in frozen frames. MEVG generates each event well, but they are connected with abrupt shot transitions and completely different subjects. This is due to the free-form event captions we use, which change the subjects frequently. As a result, the inversion technique in MEVG

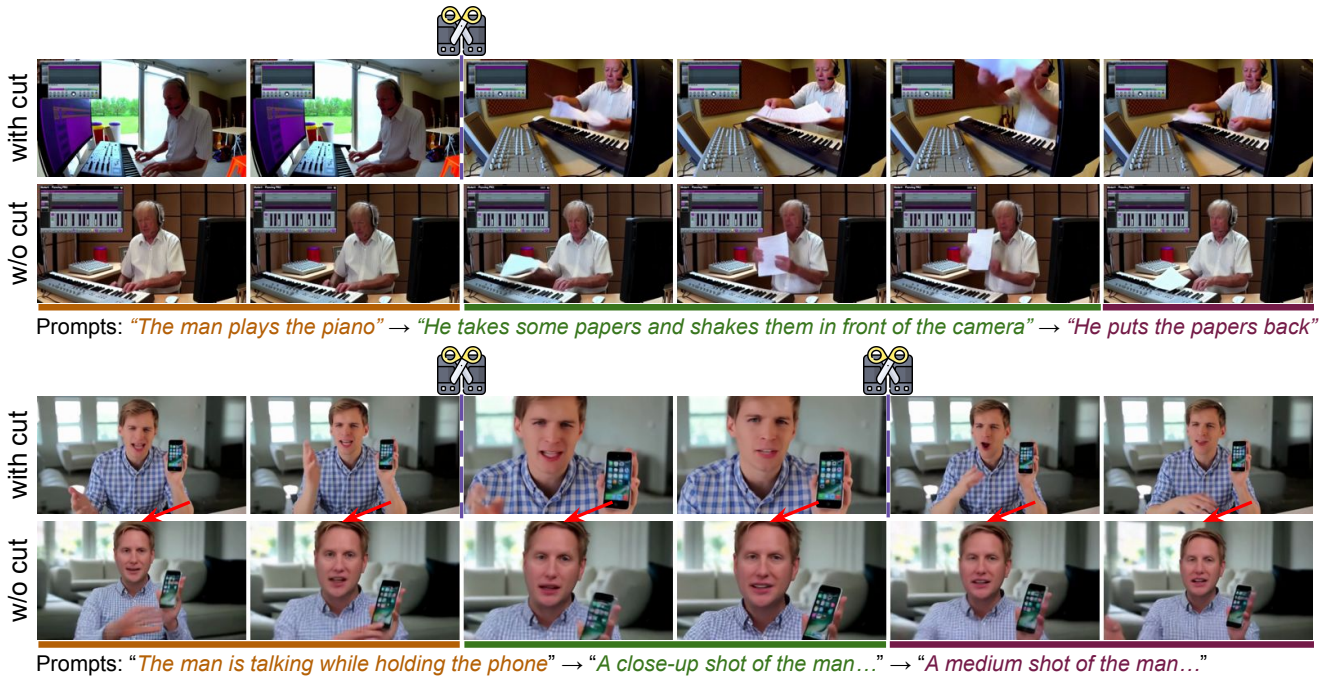

Figure 6. **Generated videos with and without scene cut input.** In each example, the first row is generated by inputting the scene cut at the illustrated timestamps, while the second row is by zeroing out the scene cut input. When using the scene cut, the model is able to generate a shot transition at desired timestamps, while keeping the subject consistent. In the second example, the model generates smooth zoom-in and zoom-out effects when zeroing out scene cuts. Please see our project page for more results.

cannot preserve the subjects well. So far, there is no inversion method designed for rectified flow models. Overall, MinT is the only method that successfully generates all events with smooth transitions and consistent entities.

We show more qualitative results of MinT in Fig. 4. Human-related subjects are known to be challenging in visual generation tasks. Yet, the results demonstrate our flexible control of human action sequences and time lengths.

C.2. Prompt Enhancement

Our prompt enhancer is built upon GPT-4 [3] and can extend a short prompt to a detailed global caption and multiple temporal captions with reasonable event timestamps. We provide the instruction we used on our project page. It is inspired by recent works [32, 35] and uses in-context examples from our dataset for better performance.

We show more prompt enhancement results using VBench prompts in Fig. 5. Thanks to the powerful LLM, our prompt enhancer can extend a short prompt to reasonable sequential events, covering rich object motion and camera movement. MinT can then generate more interesting and "eventful" videos from the extended prompt. This highlights the unique capability of our method, opening up a new direction towards more user-friendly video generation.

C.3. Scene Cut Conditioning

As shown in the ablation, removing scene cut conditioning leads to undesired shot transitions in generated videos. A closer inspection reveals that the generation of cuts is sensitive to the text prompt of an event. When it contains a description of the camera shot (e.g., "a close-up view of"), it is more likely to introduce a cut. In contrast, explicitly conditioning on scene cuts frees us from this issue.

We show some qualitative scene cut control results in Fig. 6. MinT is able to generate shot transitions at desired timestamps, while preserving subject identities. When zeroing out the scene cut input, we can get cut-free videos which validates our design. Finally, we show that our model can switch between sudden camera changes or gradual zoom-in and zoom-out effects, enabling fine-grained control.

An interesting direction is to learn different types of scene transitions such as jump cut, dissolve, and wipe. Since our goal is to retain training data instead of learning fancy transition control, we leave this for future work.

C.4. Event Time Span Control

MinT supports fine-grained control of event time span. To show this, we take a sample from our dataset and offset the start and end timestamps of all events by a specific value. Fig. 7 presents the results, where each video generates events following its new timing. In addition, we can roughly keep the appearance of the main subject and background unchanged. MinT is the first video generator in the literature that achieves this control ability. We view it as an important step towards a practical content generation tool.

Figure 7. **Generated videos with different event time spans.** In each example, we offset the start and end timestamps of all events by a specific number of seconds. Results show that MinT enables fine-grained event timing control while keeping the subjects' appearances to be roughly the same. This capability is very useful for controllable video generation. Please see our project page for more results.

Method	FID ↓	FVD ↓	CLIP-score ↑
Task: T2V (a.k.a. story generation in [7])			
Phenaki	273.41	998.19	0.210
Ours	**40.87**	**484.44**	**0.284**
Task: I2V (a.k.a. story continuation in [7])			
Phenaki	240.21	674.5	0.219
Ours	**21.85**	**314.59**	**0.273**

Table 1. **Comparison with Phenaki on StoryBench.** We compare with the zero-shot variant Phenaki-Gen-ZS in their paper [7] since our model is not fine-tuned on StoryBench. We clearly outperform Phenaki across all metrics in both tasks.

C.5. StoryBench Comparison with Phenaki

The original StoryBench paper [7] proposed a baseline for their dataset, which runs Phenaki [48] to generate events in an autoregressive way. However, they conducted evaluation on a much lower resolution (160×96), and neither their code nor pre-trained weights were released, making a direct comparison hard. We still compare with them in Tab. 1 for completeness. We only report metrics that both papers evaluate, which cover visual quality (FID, FVD) and text alignment (CLIP-score). MinT significantly outperforms Phenaki across all metrics in both T2V and I2V tasks. This demonstrates the effectiveness of fine-tuning from a large-scale pre-trained video model.

C.6. Comparison with SOTA Video Generators

To show that sequential event generation is a common failure case of even SOTA video generators, we present more results in Fig. 10 and Fig. 11. One surprising observation we had is that, when using prompts following the official guideline of these models (e.g., using the LLM provided by CogVideoX to enhance prompts), the model only generates the first event and ignores all subsequent ones. Only if we directly concatenate event captions without specifying global properties such as camera motion, background description, and detailed subject attributes (i.e., directly use prompts like "A person first do A, then do B, and finally do C"), the model starts to generate some events transitions.[2] One possible cause is that in the training data of these models, videos with sequential events are never annotated with such detailed global properties. However, since we do not have access to their training details, we can't figure out the true reason behind it. Therefore, we just use naively concatenated prompts to generate all results. The prompts we used for these models can be found on our project page. Notably, this workaround prevents us from using detailed captions to control the scene and subjects, which greatly affects the controllability of these models.

Still, when prompted with a text that contains multiple events, these models have three common failure modes:

1. Only generates partial events, and completely ignores the remaining ones. For example, in the third example in Fig. 10, all models miss the "blow kiss" action;
2. Generates events in the wrong order or "merge" multiple events. For example, in the last example in Fig. 10, Kling 1.5 generates the man with his hand under his mouth at the beginning of the video. Yet, this should happen last;
3. Bind wrong actions or properties to subjects. For example, in the first example in Fig. 11, Gen-3 Alpha generates a woman coming into the frame instead of a man.

Remark. There might be other ways to fix this issue without using temporally-grounded captions as in MinT. For example, one may fine-tune the model on video datasets annotated with detailed sequential event information [52]. Still, this will not allow precise control over the start and end times of events, which is a unique capability of our model.

[2]The detailed prompt does not exceed the maximum input text length of these models, so context length is not the reason here.

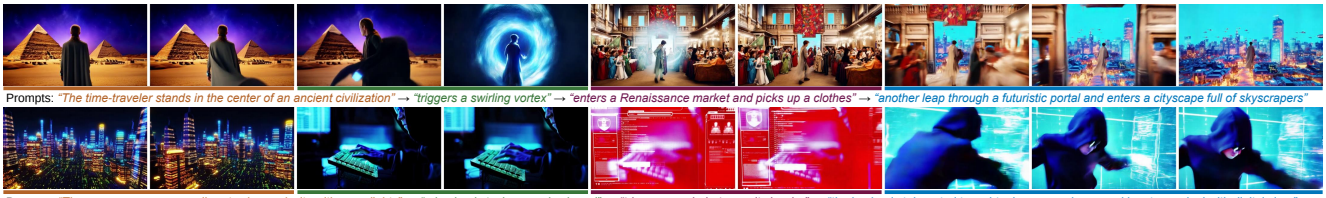

Figure 8. **Generated videos with extreme dynamics.** We prompt MinT to generate scene cuts at event boundaries, leading to explicit scene changes and large dynamics.

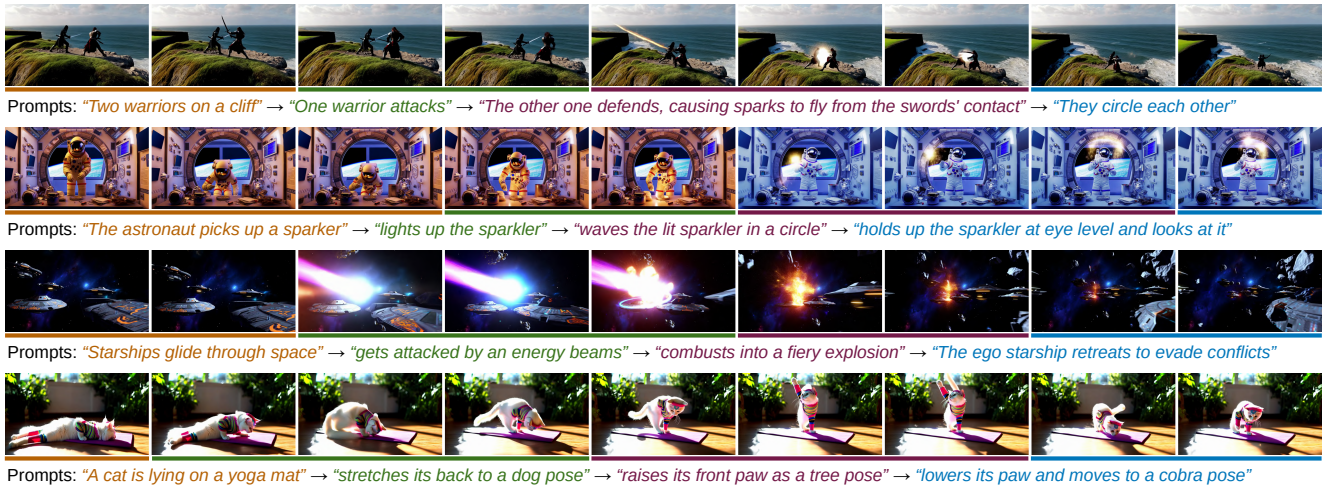

Figure 9. **Generated videos with out-of-distribution prompts.** After fine-tuning, MinT still possesses the base model's ability to generate novel concepts. Please see our project page for more results.

Quantitative comparison with Kling. Running Kling on all test prompts will incur unreasonable API costs (∼$5k). Therefore, we ran it on the 200 prompts used in our user study, and conducted a user study with 20 participants per prompt similar to our main experiment. Due to a weaker base model, MinT achieves a lower Visual Quality (31.55% win rate). Nevertheless, MinT clearly outperforms Kling in all three event-related metrics (73.18% in Text Alignment, 69.93% in Event Timing, and 68.27% in Event Transition).

C.7. Generating Videos with Extreme Dynamics

We prompt MinT to generate videos with extreme dynamics. Thanks to the scene cut conditioning, we enable explicit scene changes in generated videos as shown in Fig. 8.

C.8. Out-of-Distribution Prompts

MinT is fine-tuned on temporal caption videos that mostly describe human-centric events. In the paper, we have shown some non-human results such as animals and traffics. Here, we show that our model still possesses the ability to generate novel concepts and their combinations, which is an important property of large-scale pre-trained video generators. As shown in Fig. 9, MinT generates out-of-distribution characters such as warriors and astronaut, scenes such as starships in the space, and non-existing events such as a cat doing yoga. This proves that our model does not forget the rich pre-training knowledge in the base model.

D. Limitations and Future Works

MinT is fine-tuned from a pre-trained text-to-video diffusion model, and thus we are bounded by the capacity of the base model. For example, it is challenging to generate human hands or scenes involving complex physics.

When generating an event involving multiple subjects, MinT may fail to associate attributes and actions to the correct subject. Similar to the *temporal binding* problem we try to address in this paper, we believe this issue can be solved with *spatial binding*. For example, by grounding subjects with bounding boxes and attribute labels [26, 27, 50].

Finally, MinT sometimes fails to associate entities specified in the global caption and temporal captions. Such association requires complex reasoning of the text conditioning, and may be resolved by simply scaling up the training data.

Please refer to our project page for video examples and detailed analysis of these failure cases.

Future works. It is interesting to enhance our model with recent progress in training-free long video generation techniques [16, 40, 49]. Another direction is to combine MinT with video personalization methods [10, 22, 28, 32] to enable both fine-grained control within a shot and subject consistency across shots for minute-long video creation.

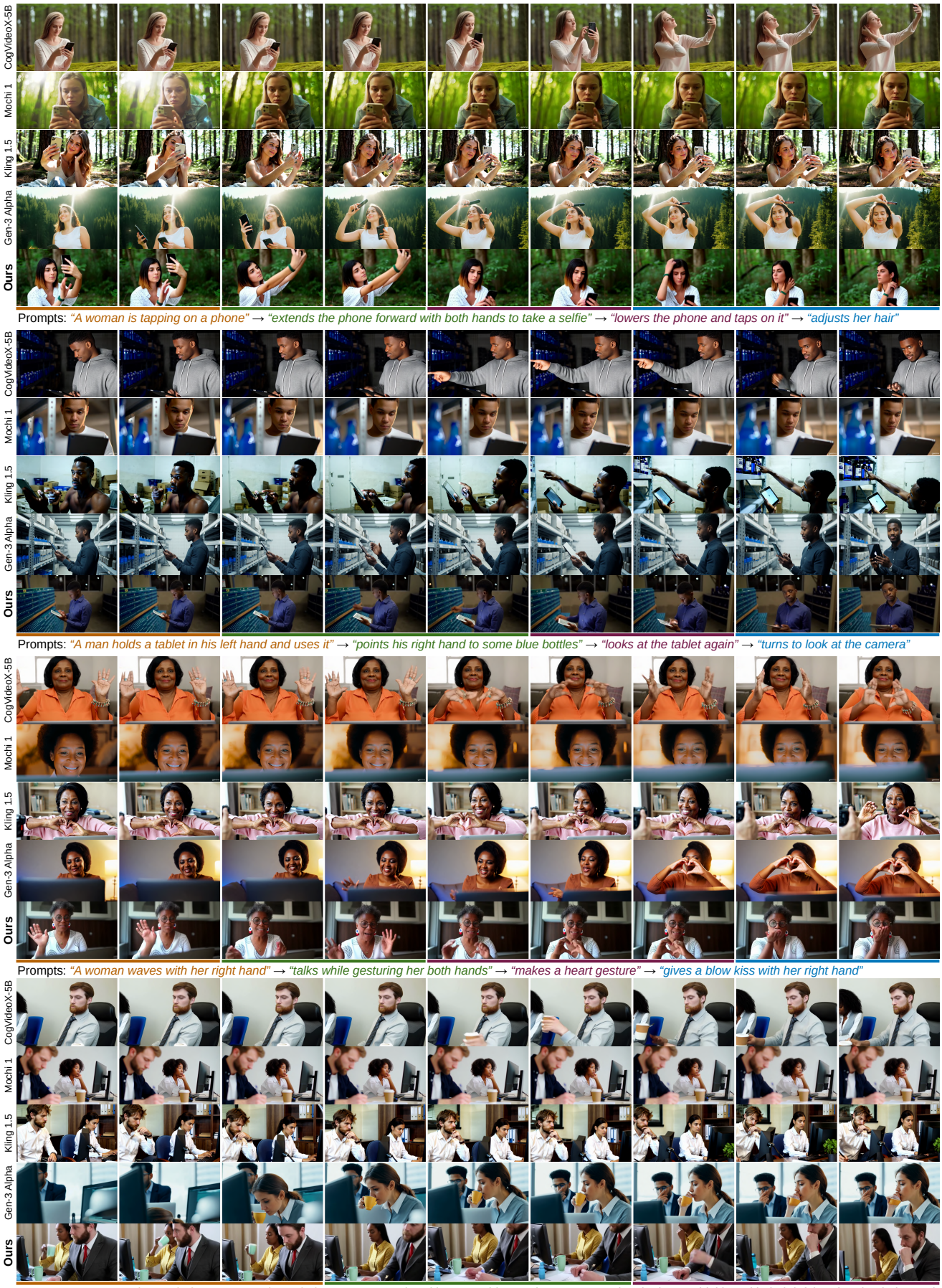

Figure 10. **More comparisons with SOTA video generators.** We run SOTA open-source models CogVideoX [55] and Mochi [46], and commercial models Kling 1.5 [2] and Gen-3 Alpha [1] using their online APIs. Please see our project page for video results.

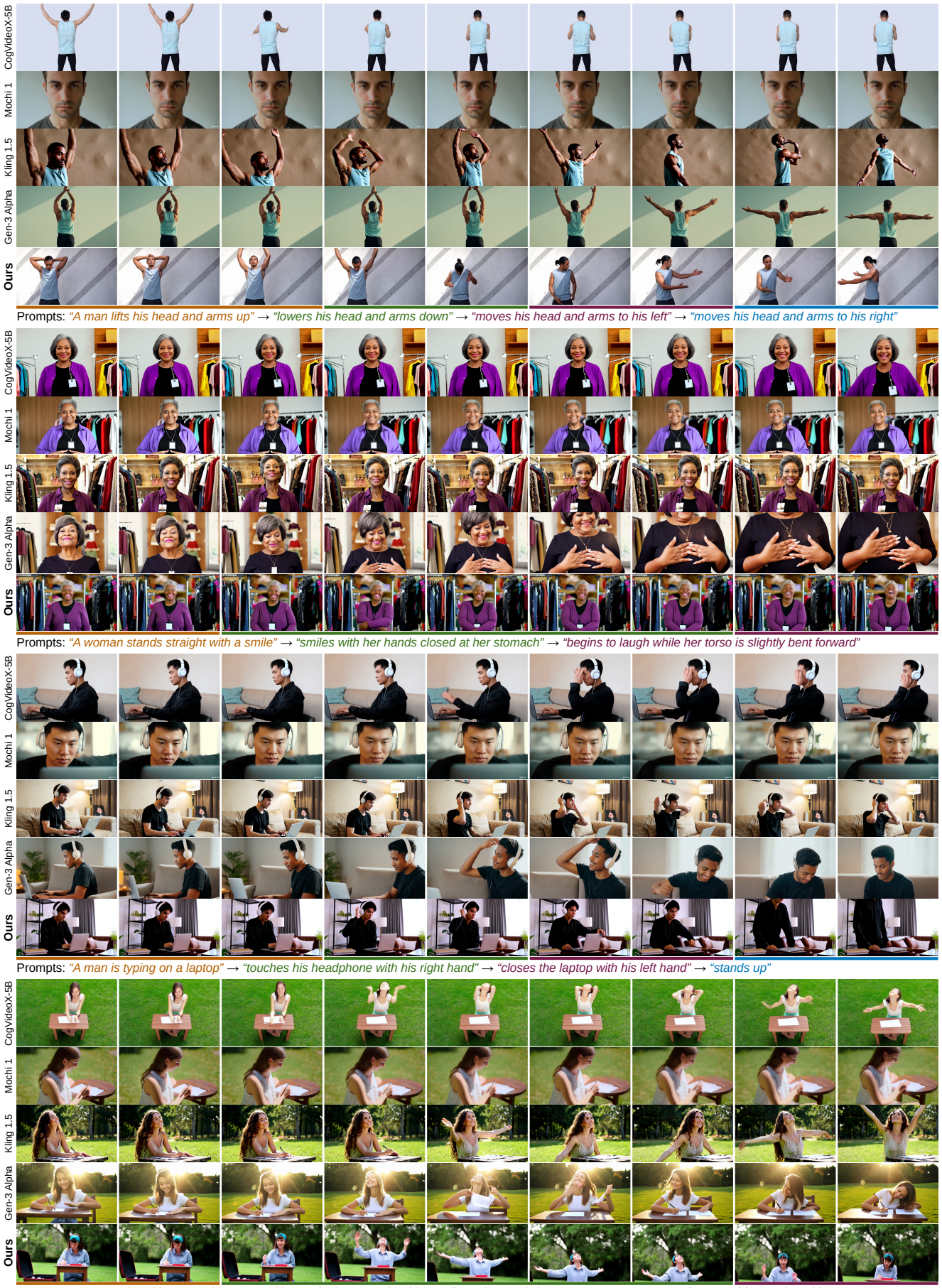

Figure 11. **More comparisons with SOTA video generators.** We run SOTA open-source models CogVideoX [55] and Mochi [46], and commercial models Kling 1.5 [2] and Gen-3 Alpha [1] using their online APIs. Please see our project page for video results.

References

[1] Gen-3 Alpha. https://runwayml.com/research/introducing-gen-3-alpha, 2024. Accessed: 2024-10-24. 9, 10

[2] Kling1.5. https://klingai.com/, 2024. Accessed: 2024-10-24. 9, 10

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[4] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 2

[5] Jimmy Lei Ba. Layer Normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 4

[7] Emanuele Bugliarello, H Hernan Moraldo, Ruben Villegas, Mohammad Babaeizadeh, Mohammad Taghi Saffar, Han Zhang, Dumitru Erhan, Vittorio Ferrari, Pieter-Jan Kindermans, and Paul Voigtlaender. StoryBench: a multifaceted benchmark for continuous story visualization. *NeurIPS*, 2024. 2, 7

[8] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 3

[9] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70M: Captioning 70m videos with multiple cross-modality teachers. In *CVPR*, 2024. 2

[10] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Yuwei Fang, Kwot Sin Lee, Ivan Skorokhodov, Kfir Aberman, Jun-Yan Zhu, Ming-Hsuan Yang, and Sergey Tulyakov. Multi-subject open-set personalization in video generation. *arXiv preprint arXiv:2501.06187*, 2025. 8

[11] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 2022. 3

[12] Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *CVPR*, 2020. 2

[13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 4

[14] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *EMNLP*, 2024. 3

[15] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022. 3

[16] Roberto Henschel, Levon Khachatryan, Daniil Hayrapetyan, Hayk Poghosyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. StreamingT2V: Consistent, dynamic, and extendable long video generation from text. *arXiv preprint arXiv:2403.14773*, 2024. 8

[17] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 3

[18] 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. *NeurIPS*, 2017. 3

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[20] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. *arXiv preprint arXiv:2011.11760*, 2020. 2

[21] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024. 2

[22] Yuming Jiang, Tianxing Wu, Shuai Yang, Chenyang Si, Dahua Lin, Yu Qiao, Chen Change Loy, and Ziwei Liu. VideoBooth: Diffusion-based video generation with image prompts. In *CVPR*, 2024. 8

[23] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 2

[24] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024. 4

[25] PKU-Yuan Lab and Tuzhan AI etc. Open-Sora-Plan, 2024. 1, 3

[26] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *CVPR*, 2023. 4, 8

[27] Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. Llm-grounded video diffusion models. In *ICLR*, 2024. 8

[28] Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. VideoDirectorGPT: Consistent multi-scene video generation via llm-guided planning. In *COLM*, 2024. 8

[29] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 4

[30] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 4

[31] Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of rope-based extrapolation. In *ICLR*, 2024. 1

[32] Fuchen Long, Zhaofan Qiu, Ting Yao, and Tao Mei. VideoStudio: Generating consistent-content and multi-scene videos. In *ECCV*, 2024. 6, 8

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 4

[34] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *ECCV*, 2022. 3

[35] Gyeongrok Oh, Jaehwan Jeong, Sieun Kim, Wonmin Byeon, Jinkyu Kim, Sungwoong Kim, and Sangpil Kim. MEVG: Multi-event video generation with text-to-video models. In *ECCV*, 2024. 3, 6

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 4

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. *ICCV*, 2023. 3

[38] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *ICLR*, 2024. 1

[39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2

[40] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. FreeNoise: Tuning-free longer video diffusion via noise rescheduling. In *ICLR*, 2024. 8

[41] 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*, 2021. 3

[42] 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. *JMLR*, 2020. 3

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3

[44] Tomáš Souček and Jakub Lokoč. TransNet V2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020. 2, 3

[45] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 1

[46] Genmo Team. Mochi, 2024. 9, 10

[47] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 3

[48] 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 descriptions. In *ICLR*, 2022. 7

[49] Fu-Yun 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 preprint arXiv:2305.18264*, 2023. 8

[50] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024. 8

[51] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021. 2

[52] Tianwei Xiong, Yuqing Wang, Daquan Zhou, Zhijie Lin, Jiashi Feng, and Xihui Liu. Lvd-2m: A long-take video dataset with temporally dense captions. *NeurIPS*, 2024. 7

[53] 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*, 2022. 2

[54] Antoine Yang, Arsha Nagrani, Ivan Laptev, Josef Sivic, and Cordelia Schmid. VidChapters-7M: Video chapters at scale. *NeurIPS*, 2023. 2

[55] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3, 9, 10

[56] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 3

[57] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-Sora: Democratizing efficient video production for all, 2024. 2

[58] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2