LinGen: Towards High-Resolution Minute-Length Text-to-Video Generation with Linear Computational Complexity

Hongjie Wang[1,2], Chih-Yao Ma[2], Yen-Cheng Liu[2], Ji Hou[2], Tao Xu[2], Jialiang Wang[2], Felix Juefei-Xu[2], Yaqiao Luo[2], Peizhao Zhang[2], Tingbo Hou[2], Peter Vajda[2], Niraj K. Jha[1], Xiaoliang Dai[2]
[1]Princeton University, [2]Meta

Abstract

*Text-to-video generation enhances content creation but is highly computationally intensive: The computational cost of Diffusion Transformers (DiTs) scales quadratically in the number of pixels. This makes minute-length video generation extremely expensive, limiting most existing models to generating videos of only 10-20 seconds length. We propose a **Lin**ear-complexity text-to-video **Gen**eration (**LinGen**) framework whose cost scales linearly in the number of pixels. For the first time, LinGen enables high-resolution minute-length video generation on a single GPU without compromising quality. It replaces the computationally-dominant and quadratic-complexity block, self-attention, with a linear-complexity block called **MATE**, which consists of an **MA**-branch and a **TE**-branch. The MA-branch targets short-to-long-range correlations, combining a bidirectional Mamba2 block with our token rearrangement method, Rotary Major Scan, and our review tokens developed for long video generation. The TE-branch is a novel TEmporal Swin Attention block that focuses on temporal correlations between adjacent tokens and medium-range tokens. The MATE block addresses the adjacency preservation issue of Mamba and improves the consistency of generated videos significantly. Experimental results show that LinGen outperforms DiT (with a 75.6% win rate) in video quality with up to 15× (11.5×) FLOPs (latency) reduction. Furthermore, both automatic metrics and human evaluation demonstrate our LinGen-4B yields comparable video quality to state-of-the-art models (with a 50.5%, 52.1%, 49.1% win rate with respect to Gen-3, LumaLabs, and Kling, respectively). This paves the way to hour-length movie generation and real-time interactive video generation. We provide 68s video generation results and more examples in our project website: https://lingen.github.io/.*

1. Introduction

Diffusion Models (DMs) [16, 54] have exhibited superior performance on various generative tasks, including image generation [5, 41, 46, 48], image editing [4, 22, 50, 70], 3D shape generation [34, 58], and video generation [1, 11, 42, 75]. Among them, high-resolution text-to-video generation is widely regarded as one of the most challenging tasks due to two key factors: (1) the immense complexity of predicting the values of hundreds of millions of pixels and (2) the human eye's acute sensitivity to inconsistencies across frames. Sora [1] and Movie Gen [42] achieve highly consistent video generation by scaling Diffusion Transformers (DiTs) [40] to tens of billions of parameters. However, the computational cost of DiTs scales quadratically in the resolution and length of generated videos, making it extremely expensive to generate long videos and limiting the raw video length of most existing models to 10-20 seconds.

Numerous existing studies have focused on improving the efficiency of video generation. This can be categorized into two approaches: (1) sampling distillation [28, 63], which reduces the number of sampling steps, and (2) efficient architectural designs that lower the computational cost of each sampling step, which includes factorized attention [2, 62] and State Space Models (SSMs) [10, 37]. However, they either retain quadratic complexity or are restricted to generating low-resolution, short videos. It is challenging to perform high-resolution long video generation solely based on the linear-complexity SSMs like Mamba [12], due to its ***adjacency preservation issue*** [10]. Mamba was originally designed for language tasks, where the inputs are natively sequences. When it is adapted to the vision modality, rearranging 2D (images) or 3D (videos) tensors into a 1D sequence becomes a necessity. This rearrangement causes spatially and temporally adjacent tokens to become distant in the sequence. This significantly hurts the quality of generated images and videos [19] due to the inherent decay when Mamba calculates long-range correlations [12]. Although more sophisticated rearrangement methods [15, 19, 45] could alleviate this issue, they can hardly ensure consistency across frames when scaled to high-resolution long video generation.

To address the above challenge, we propose a **Linear-complexity text-to-video Generation (LinGen)** frame-

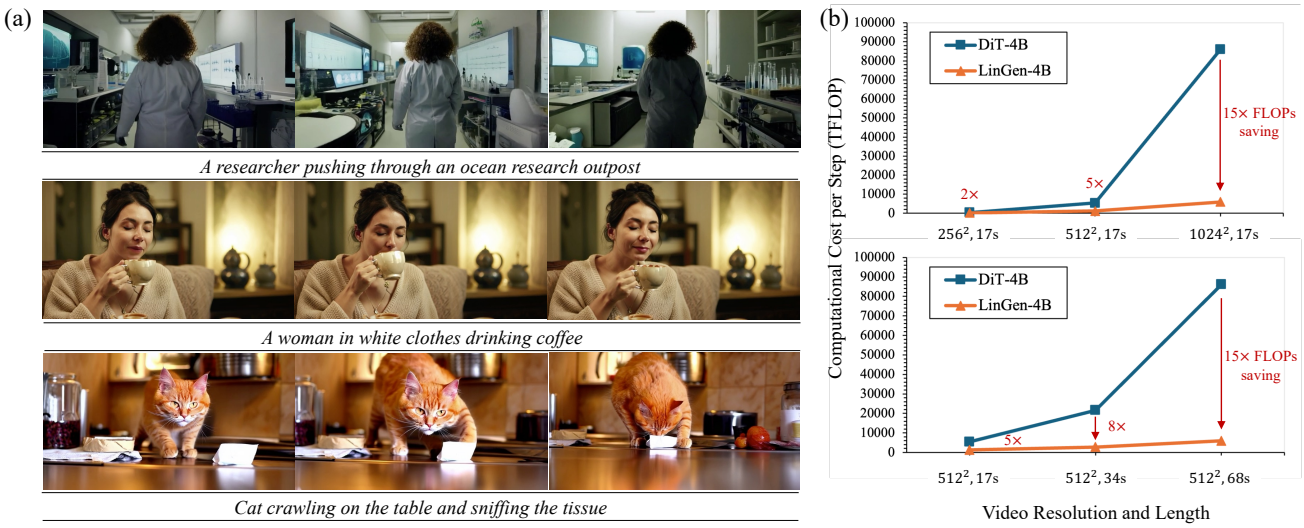

Figure 1. **LinGen generates photorealistic high-resolution long videos with linear computational complexity.** (a) High-quality videos generated using our LinGen model. (b) The computational cost scaling curves across different video resolutions and lengths. LinGen achieves 15× speed-up compared to the standard DiT when generating 68s-length videos at 512p resolution.

work that scales linearly in the number of pixels in generated videos. To the best of our knowledge, LinGen is **the first to enable photorealistic high-resolution minute-length video generation at a high frame rate on a single GPU without video extension, super-resolution, or compromising quality**. It not only addresses the aforementioned adjacency preservation issue, but also comprehensively enhances the short-, medium-, and long-range correlations while maintaining linear complexity. LinGen replaces the self-attention layers in DiTs with our proposed linear-complexity MATE blocks. Each MATE block is composed of an MA-branch and a TE-branch. The MA-branch consists of a bidirectional Mamba2 [6] (a transformer-format SSM variant) block equipped with our proposed Rotary-Major Scan (RMS) and review tokens. RMS rearranges 3D token tensors in the latent space before they enter the bidirectional Mamba2 block, enhancing short-range correlations. To alleviate the inherent long-range correlation decay of SSMs, review tokens provide an overview of the processed token sequences to the hidden state of Mamba2 blocks at the start of sequence processing, to calibrate long-range correlations. The TE-branch is a novel TEmporal Swin Attention (TESA) block. It computes correlations among short-range spatially adjacent and medium-range temporally adjacent tokens, focusing on addressing the adjacency preservation issue and improving video consistency. Note that LinGen is orthogonal to sampling distillation and can potentially be combined with it to further boost its efficiency. Our contributions can be summarized as follows.

- We propose LinGen, a text-to-video generation framework that enables photorealistic minute-length video generation with linear computational complexity.
- To comprehensively cover short-, medium-, and long-range correlations, we compose our proposed self-attention replacement block, MATE, with an MA-branch, including a bidirectional Mamba2 block equipped with our RMS and review tokens, and a TE-branch that includes a novel TESA block.
- We establish the superiority of the proposed LinGen framework by comparing it to our self-attention baseline, DiT-4B, and other existing video generation models via human evaluations and automatic evaluation metrics. Experimental results indicate LinGen generates photorealistic high-quality videos while achieving linear scaling and up to 15× speed-up when generating minute-length videos at 16 fps (see Fig. 1).

2. Related Work

High-Quality Video Generation. Sora [1] was the first work to successfully produce high-resolution videos with exceptional consistency. It learns an encoded latent space and deploys a large-scale DiT embedded in it. Runway Gen3 [47], LumaLabs [33], and Kling [24] are subsequent works capable of generating highly consistent, high-resolution videos with high frame rates. MovieGen [42] generates photorealistic and highly consistent videos with all implementation details revealed. However, it scales the DiT to 30 billion parameters. Its quadratic complexity makes generating minute-length videos very difficult. Several open-source models [3, 62, 75] also aim to generate high-quality videos. However, the quality of their outputs still notably lags behind that of the aforementioned models. An alternative to DMs for video generation is the use of transformer-based language models, which autoregressively generate video tokens [25, 38, 59, 69, 72].

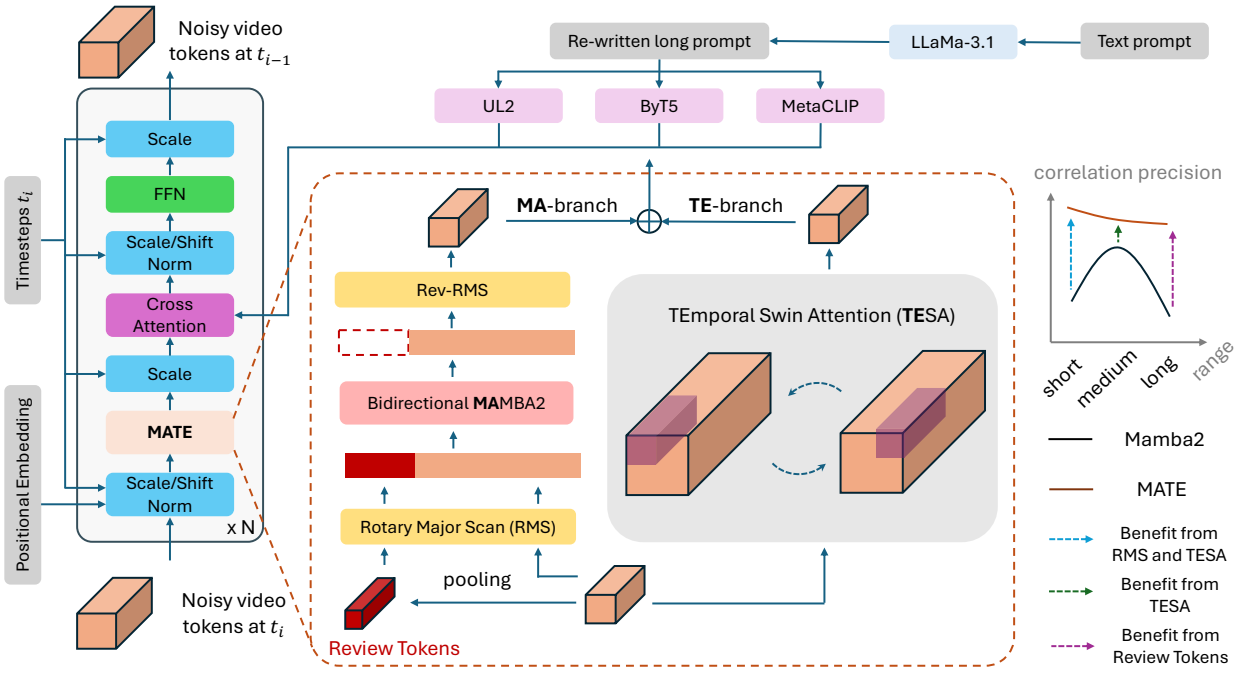

Figure 2. **Overview of the LinGen denoising module.** LinGen replaces self-attention layers with a MATE block, which inherits linear complexity from its two branches: **MA-branch** and **TE-branch**. The **MA-branch** consists of a bidirectional Mamba2 block, RMS, and review tokens to cover short-to-long-range correlations. The **TE-branch** is a TEmporal Swin Attention block that addresses the adjacency preservation issue and improves the consistency of generated videos significantly.

While these models are well-suited to multimodal conditioning tasks, the quality of their generated videos generally falls short of that achieved by DM-based models.

Efficient Video Generation. The high computational cost of DM-based video generation has prompted various research efforts to address this challenge. Most of them are inspired by efficient DM-based image generation works [23, 35, 36, 61] and can be divided into two types: (1) **Sampling distillation** to reduce the required number of sampling steps to generate high-quality videos. VideoLCM [63] uses Consistency Distillation [55] to enable satisfactory video generation in four steps. T2V-Turbo [28] integrates reward feedback into the distillation process to further improve video quality. (2) **Efficient denoising architecture design** to reduce the cost of each sampling step. Many existing works [2, 17, 53, 62, 64] employ factorized spatial and temporal attention to reduce the computational cost of calculating global attention across the entire 3D video token tensor. They still maintain quadratic complexity. Matten [10] and DiM [37] replace some self-attention layers with bidirectional Mamba blocks. However, they either need to maintain some global self-attention layers (thus have quadratic complexity) or can only generate low-resolution short videos. On the contrary, LinGen solves the adjacency preservation issue well and manages to generate high-quality minute-length videos.

Minute-Length Video Generation. Some existing works [65, 66] have conducted early explorations into generating minute-length videos. However, their generated videos have various limitations, including low frame rates, low resolution, and reduced quality due to the extension-based generation pattern.

3. Methodology

The computational cost of self-attention scales quadratically with the number of tokens in the sequence, creating a bottleneck for DiT-based video generative models due to the extensive length of the encoded video token sequence [32, 64]. Such a quadratic complexity makes generating high-resolution minute-length videos extremely expensive. Therefore, we propose LinGen, a text-to-video generation framework that produces photorealistic videos with linear complexity, enabling high-resolution minute-length video generation at a low cost.

3.1. Overview

LinGen uses a Temporal AutoEncoder design that is similar to a prior work [42]. In the latent space, LinGen denoises tokens using Flow Matching [30] and the linear-quadratic t-schedule [42]. The denoising module of LinGen is shown in Fig. 2. We provide more implementation details in the Supplementary Material (Supp. Mat.) section. The cross-attention layer conditions on text embeddings projected by three encoders: UL2 [56], ByT5 [68], and MetaCLIP [67]. They take long prompts re-written

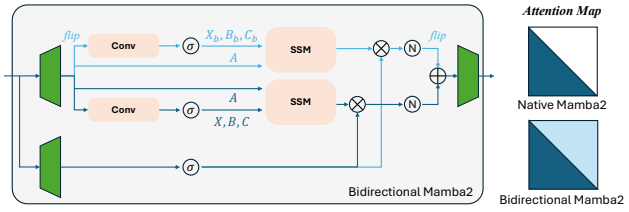

Figure 3. **The bidirectional Mamba2 module.** Native Mamba2 only generates the lower triangular part of the attention map due to its causal characteristic. Thus, we deploy bidirectional Mamba2 to obtain the complete attention map for vision tasks.

by LLaMa-3.1 [9] as input. Most importantly, LinGen replaces the self-attention layer of vanilla DiTs with our proposed MATE block, achieving linear computational complexity. MATE is composed of two branches: **MA-branch** and **TE-branch**. The **MA-branch** incorporates a bidirectional Mamba2 block, RMS, to enhance short-range correlations, and review tokens to calibrate long-range correlations (see Sec. 3.2). The **TE-branch** is a novel TESA block, focusing on correlations among short-range spatially adjacent and medium-range temporally adjacent tokens (see Sec. 3.3). As opposed to Mamba, MATE addresses its adjacency preservation issue and comprehensively enhances short-, medium-, and long-range correlations while maintaining linear complexity in the number of tokens. We describe these components in detail in the following sections and introduce our training recipe in Sec. 3.4.

3.2. MA-Branch: Targets Short-to-Long Range

Bidirectional Mamba2. Mamba2 [6] unifies SSMs and masked efficient attention by proposing a special SSM with an attention format (*i.e.*, Structured State Space Duality). Compared to Mamba, Mamba2 is more hardware-friendly. Thus, we deploy the bidirectional version of Mamba2 in LinGen to obtain the complete correlation map, as shown in Fig. 3. The number of FLoating Point Operations (FLOPs) of this block is given by

$$C_{\text{bimamba}} = (6 + \frac{2}{d_h})END^2 + 4Nd_sd + O(Nd), \quad (1)$$

where E is the expansion factor, d is the dimension of token embedding vectors, N is the number of tokens, d_s is the hidden state size, and d_h is the head dimension of Mamba2, whose default value is 64. We provide the complete expression for C_{bimamba} in Supp. Mat. This format shows that C_{bimamba} scales linearly in N. The linear complexity of Mamba and Mamba2 makes them highly suitable for video generation, where latent space sequences often contain tens or even hundreds of thousands of tokens. However, videos generated by the native Mamba model exhibit high inconsistency, primarily due to the adjacency preservation issue when rearranging 3D tensor tokens into a sequence [10, 19]. Previous works have attempted to address this problem by mixing Mamba layers with global attention layers [10], thus compromising linear complexity. On the contrary, we equip Mamba2 with RMS and review tokens to build the MA-branch and develop the TE-branch with TESA, enhancing control over continuous spatial and temporal neighbors and calibrating long-range correlations while maintaining linear complexity.

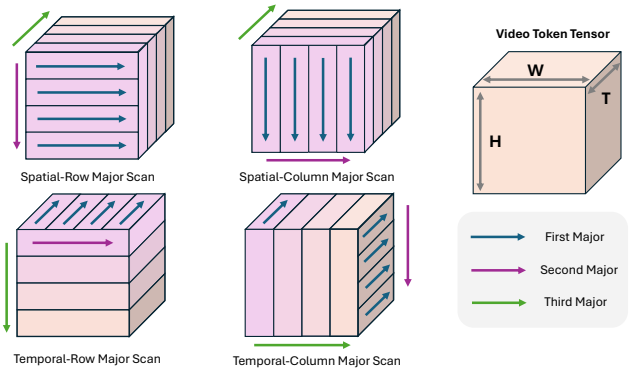

Figure 4. **Rotary-Major Scan (RMS).** We apply different scan schedules across layers to preserve adjacency along various dimensions. Note that scan is bidirectional in practice, but for clarity, only one direction is illustrated for each scan schedule.

Rotary-Major Scan. Assume the token tensor shape in the latent space is $H \times W$. Adjacent tokens in the same column are separated at a distance of H in the default row-major scan. Taking into account that Mamba-calculated correlation precision decays as the distance increases, the failure of adjacency preservation leads to distortion in generated images. Zigzag scan [19] was proposed to alleviate this issue, but it causes significant latency increment when rearranging huge 3D tensors for video generation (see Sec. 4.5).

Thus, we propose RMS, which causes negligible extra latency when targeting large 3D video token tensors. It rearranges the 3D tensor that represents the latent video into a 1D sequence in four different ways in different layers, including spatial-row major, spatial-column major, temporal-row major, and temporal-column major, as shown in Fig. 4. We employ these different scan methods in different layers in an alternating fashion. Assuming the token tensor shape in the latent space is $T \times H \times W$, the index of token $T[t][y][x]$ in the re-arranged 1D sequence in the l-th layer is given by

$$n_l = \begin{cases} t \cdot (H \cdot W) + y \cdot W + x, & \text{if } l \mod 4 = 0 \\ t \cdot (H \cdot W) + x \cdot H + y, & \text{if } l \mod 4 = 1 \\ y \cdot (T \cdot W) + x \cdot T + t, & \text{if } l \mod 4 = 2 \\ x \cdot (T \cdot H) + y \cdot T + t, & \text{if } l \mod 4 = 3 \end{cases}$$

Note that the scan in each layer is bidirectional; hence, a flipped sequence $n_{l,flip} = T \cdot H \cdot W - n_l$ always exists simultaneously. RMS can be implemented with just a few lines of code to reshape the token tensor, making it highly hardware-friendly for processing large tensors. Ablation experiments (see Sec. 4.5) show that RMS achieves similar performance to the Zigzag scan in video generation while

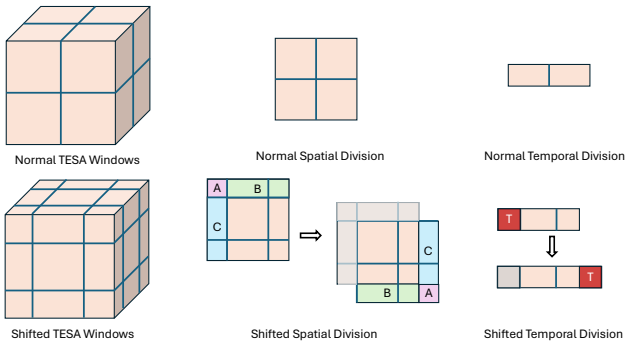

Figure 5. **TEmporal Swin Attention (TESA).** We divide the token tensor into small windows and calculate self-attention within each window. The windows are alternately shifted across layers to cross the boundaries of local windows. The window size remains fixed across different resolutions, hence maintaining linear complexity.

significantly reducing additional latency.

Review Tokens. To enhance the overall understanding of generated videos and improve text-video alignment in long video generation, we add review tokens when processing extremely long sequences. Specifically, we append an average-pooled version of the token tensor to the beginning of the sequence (and its flipped version) expanded by RMS, allowing Mamba2 to incorporate an overview of the sequence into its hidden state before sequence processing begins. This does not introduce any extra parameters, although it incurs extra FLOPs that equal

$$C_{\text{RT}} = \frac{1}{p_t \cdot p_x \cdot p_y} \cdot C_{\text{bimamba}}, \tag{2}$$

where p_t, p_y, p_x are the average pooling range along the temporal, height, and width dimensions of the video token tensor, respectively. As this equation shows, C_{RT} also scales linearly in the number of tokens N, following the behavior of C_{bimamba}. In practice, we set $\{p_t, p_y, p_x\} = \{8, 4, 4\}$. Thus, the extra cost of review tokens is marginal.

3.3. TE-Branch: TEmporal Swin Attention

Besides the MA-branch, to further address the adjacency preservation issue and enhance video consistency, we propose TEmporal Swin Attention (TESA) to build the TE-branch, which gathers short-range information along the spatial dimension and medium-range information along the temporal dimension, as shown in Fig. 5. It is inspired by a prior window attention work [31], divides the 3D video token tensor into multiple windows, and calculates attention between tokens within the same window. Assuming the window size is $T_w \times S_w \times S_w$ and the video token tensor size is $T \times H \times W$, the FLOPs of TESA is given by

$$C_{\text{TESA}} = (8N_w d^2 + 4N_w^2 d) \cdot \left\lceil \frac{T}{T_w} \right\rceil \cdot \left\lceil \frac{H}{S_w} \right\rceil \cdot \left\lceil \frac{W}{S_w} \right\rceil \tag{3}$$

where $N_w = T_w \cdot S_w \cdot S_w$ and d is the dimension of token embedding vectors. This equation indicates that C_{TESA} scales linearly in $N = T \cdot H \cdot W$. Its spatial window size $S_w \times S_w$ is very small (we set it to 4×4 in practice), because we mainly use the MA-branch of MATE to deal with spatial correlations and TESA focuses on adjacent correlations along the spatial dimension. Benefiting from such a small spatial window size, TESA incurs negligible extra latency (see Sec. 4.5). As indicated in Fig. 5, the window range of TESA shifts alternatingly in different layers. The self-attention computation in the shifted windows crosses the boundaries of the previous windows, establishing connections among them and enlarging the receptive field.

3.4. Training Recipe

Progressive Training. We use a progressive recipe (check details in Supp. Mat.) to pre-train our LinGen-4B model. We first pre-train our model on the text-to-image task at a 256p resolution, followed by text-to-video pre-training at progressively higher resolutions (256p to 512p) and longer video lengths (17s to 34s and then 68s).

Text-to-Image and Text-to-Video Hybrid Training. In the text-to-video pre-training stages, we incorporate text-image pairs into the pre-training dataset and perform text-to-image and text-to-video joint training in practice. We find such a hybrid training improves consistency of generated videos in some failure cases.

Quality Tuning. Similar to the observation in prior works [5, 11], we find the quality of generated videos can be greatly enhanced by fine-tuning the model on a small set of high-quality videos. We select 3K high-quality videos from our pre-training dataset and fine-tune our model on them.

4. Experiments

In this section, we begin by describing the experimental settings in Sec. 4.1. We then illustrate the efficiency superiority of LinGen in Sec. 4.2. Next, we benchmark LinGen against state-of-the-art models in Sec. 4.3. In addition, we demonstrate rapid adaptation of LinGen to longer sequences in Sec. 4.4. Finally, in Sec. 4.5, we report on ablation studies that validate the effectiveness of individual modules and techniques incorporated into LinGen.

4.1. Experimental Settings

Models. (1) LinGen-4B. We build the denoising module of this model following the setting described in Sec. 3. We employ 32 layers with 20 heads in each, with the dimension of embedding vectors being 2560. (2) DiT-4B. We replace MATE blocks in LinGen-4B with global self-attention layers to build a standard DiT. Our DiT-4B has 32 layers with 24 heads in each, with the dimension of embedding vectors being 3072. (3) State-of-the-art models. We compare LinGen to state-of-the-art accessible commercial text-to-video generative models, including Runaway Gen3 [47], Kling [24], and LumaLabs [33], and a typical open-source

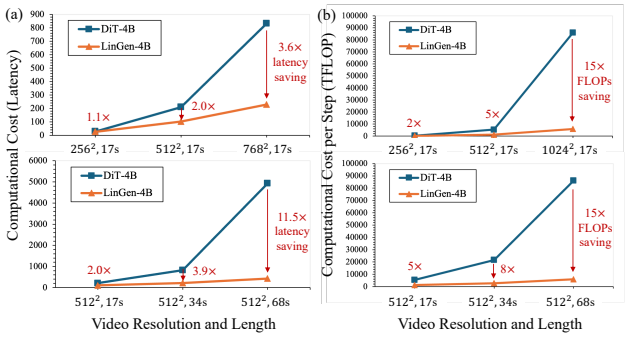

Figure 6. **Computational cost comparison between DiT-4B and LinGen-4B.** (a) Latency. (b) FLOPs. The cost of LinGen scales significantly slower with both video length and video resolution than DiT. Latency is measured on a single H100 GPU.

model, OpenSora [75]. We provide comparisons to more open-source models [3, 28, 62, 64, 71] in Supp. Mat. Note that most of these open-source models can only generate short videos containing less than 100 raw frames.

Datasets. We use 300M licensed ShutterStock [52] text-image pairs and 24M licensed ShutterStock text-video pairs to pre-train our models. We select 3K videos from the ShutterStock and RawFilm [43] video dataset to fine-tune our models. More details are provided in Supp. Mat.

4.2. Efficiency: Linear Computational Complexity

We compare the efficiency of DiT-4B and our proposed LinGen-4B in terms of FLOPs cost and latency. We show the results in Fig. 6. In terms of FLOPs, LinGen-4B achieves 5×, 8×, and 15× speed-up relative to DiT-4B when generating 512p videos of 17s, 34s, and 68s length, respectively. In terms of latency, LinGen-4B achieves 2.0× and 3.6× speed-up relative to DiT-4B when generating 512p and 768p 17s videos on a single H100, respectively. LinGen-4B achieves 2.0×, 3.9×, and 11.5× latency speed-up compared to DiT-4B when generating 512p videos of 17s, 34s, and 68s length, respectively. These results indicate that the cost of LinGen scales linearly in the number of pixels in generated videos, thus demonstrating huge efficiency and scalability superiority of LinGen.

4.3. Comparing Quality to State-of-the-Art Models

We evaluate the performance of our proposed LinGen-4B model and other text-to-video models in three ways: (1) Exhibit visual examples for eyeballing comparison, as shown in Fig. 7. We provide more examples in Supp. Mat. (2) Use human evaluation to perform A/B comparison and calculate win rates. (3) Use automatic quantitative metrics to compare LinGen with more existing text-to-video models. We use a standard video evaluation benchmark, VBench [20], to evaluate video quality and text-video faithfulness. VBench comprehensively evaluates text-to-video models using 16 disentangled dimensions. Each dimension is tailored to specific prompts and evaluation methods.

Human Evaluation Results. We compare the quality and text-faithfulness of videos generated by DiT-4B and LinGen-4B at 256p after being trained for 40K steps with a batch size of 1024; results are shown in Fig. 8. This indicates that **LinGen-4B outperforms DiT-4B in both video quality and text-video alignment**, while achieving linear complexity and significant speed-up. We speculate that, while both models are transferred from the text-to-image generation task, LinGen exhibits a superior ability to adapt to longer token sequences (see Sec. 4.4). Consequently, LinGen learns text-to-video generation more efficiently than DiT, resulting in improved performance. Fig. 9 incidates that LinGen has comparable performance to state-of-the-art commerical video generative models.

Automatic Quantitative Results. Given that the shortest video from LinGen is 17s long, significantly surpassing most models on the VBench-standard leaderboard, we evaluate LinGen against models on VBench-Long instead, as shown in Table 1. It shows that **LinGen outperforms Kling in terms of video quality and has similar overall performance to both Gen-3 and Kling, while achieving linear complexity and enabling more than one thousand raw frames generation on a single GPU.** LinGen outperforms OpenSora significantly. We provide the complete leaderboard and evaluation results on VBench-standard and VBench-Custom in Supp. Mat.

4.4. Adaptation to Longer Token Sequences

LinGen adapts to longer sequences of latent tokens more quickly than DiT. This could benefit from the strong adaptation ability of Mamba models to longer sequences, which has also been observed in language tasks [44]. We observe this phenomenon in the loss curves when transferring the model trained on 256p video generation to 512p generation in progressive training, as shown in Fig. 10 (a). We further conduct a human evaluation on the checkpoints at an early stage of 512p 17s video generation pre-training and 512p 34s video generation pre-training, as shown in Fig. 10 (b). The results validate our observation that LinGen adapts more quickly to longer sequences of latent tokens than DiT, which means better scalability for video generation at higher resolutions and longer lengths.

4.5. Ablation Experiments

For performance, we conduct ablation experiments on the 256p 17s video generation task in two ways: (1) Comparing loss curves. The prior work [42] has observed that the loss curve correlates well with visual quality evaluated by humans. Thus, we compare the loss curves under different training settings to validate their effectiveness, as shown in Fig. 11. (2) Performing human evaluations. We select corresponding checkpoints after 30K pre-training steps and perform A/B quality comparison between the default set-

Figure 7. Visual examples of videos generated from different models. LinGen-4B generates videos that have similar quality to state-of-the-art commercial video generative models, including Gen-3, LumaLabs, and Kling, while achieving linear complexity and significant speed-up relative to the standard DiT architecture.

Model	Subject Consist.	BG. Consis.	Temp. Flick.	Motion Smooth.	Aesthe. Quality	Imag. Quality	Dyna. Degree	**Quality Score**	**Total Score**	**Max. Raw Frames**
Runway Gen-3 [47]	97.10%	96.62%	98.61%	99.23%	60.14%	**63.34%**	**66.82%**	**84.11%**	**82.32%**	256
Kling [24]	**98.33%**	97.60%	99.30%	**99.40%**	46.94%	61.21%	65.62%	83.39%	81.85%	313
OpenSora V1.2 [75]	96.75%	**97.61%**	**99.53%**	98.50%	42.39%	56.85%	63.34%	81.35%	79.76%	408
LinGen	98.30%	97.60%	99.26%	98.58%	**63.67%**	60.55%	63.36%	83.77%	81.76%	**1088**

Model	Object Class	Multiple Objects	Human Action	Color	Spatial Relatio.	Scene	Appear. Style	Temp. Style	Overall Consist.	**Semantic Score**
Runway Gen-3 [47]	87.81%	53.64%	96.40%	80.90%	65.09%	**54.57%**	**24.31%**	**24.71%**	26.69%	75.17%
Kling [24]	87.24%	**68.05%**	93.40%	89.90%	**73.03%**	50.86%	19.62%	24.17%	26.42%	**75.68%**
OpenSora V1.2 [75]	82.22%	51.83%	91.20%	**90.08%**	68.56%	42.44%	23.95%	24.54%	**26.85%**	73.39%
LinGen	**90.98%**	55.15%	**97.50%**	83.95%	58.15%	53.51%	21.08%	24.29%	26.32%	73.73%

Table 1. Automatic evaluation of LinGen on VBench-Long. **Quality Score** measures the quality of generated videos and **Semantic Score** measures text-video alignment. **Total Score** is their weighted sum. Higher values indicate better performance for all these metrics. LinGen is comparable to state-of-the-art commercial models (*i.e.*, Gen-3 and Kling) and outperforms the typical open-source model (*i.e.*, OpenSora) significantly. LinGen not only achieves a much higher maximum number of raw frames but also does so on a single GPU.

ting of LinGen and the changed setting of LinGen. The win rates are shown in Fig. 12. We provide more visual examples in Supp. Mat. **For efficiency**, we measure 512p 17s video generation latency of LinGen under different settings, as shown in Table 2. Fig. 12 validates the effectiveness of review tokens, hybrid training, and quality tuning, and Table 2 shows review tokens incur marginal extra latency.

TESA Block. Table 2 shows that the TESA block only incurs marginal latency, while Fig. 11 indicates that it contributes effectively to the quality of generated videos. As

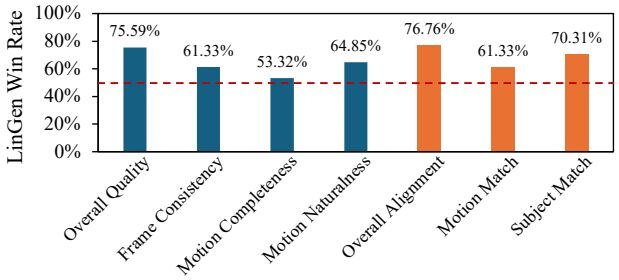

Figure 8. Human evaluation on the quality and text-video alignment of videos generated by DiT-4B and LinGen-4B. LinGen outperforms DiT due to it faster adapation to longer token sequences.

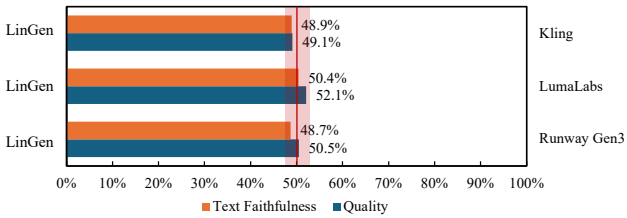

Figure 9. Win rates of human evaluation on the quality and text-video alignment of videos generated by LinGen and state-of-the-art video generative models. LinGen has comparable performance to them, given that the variance of human evaluation is 3%.

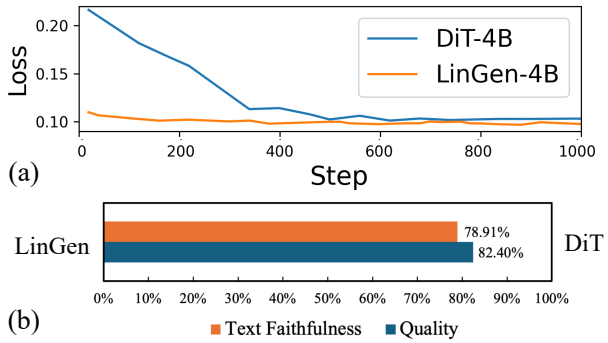

Figure 10. LinGen adapts much faster to the new task than DiT. (a) Loss curves when transferring the model trained on 256p video generation to 512p. (b) Win rates of human evaluation on quality and text-video faithfulness comparison between LinGen-4B and DiT-4B. Checkpoints are selected after 1K pre-training steps.

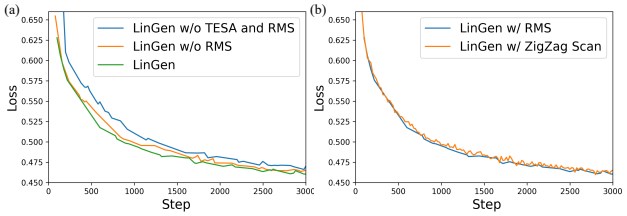

Figure 11. Loss curves of 256p text-to-video pre-training under different settings. (a) Ablation on the TESA block and RMS. (b) Ablation on different scan methods.

expected, TESA is efficient due to its small window size, while being very effective due to its addressing of the adjacency preservation issue and enhancing medium-range temporal correlation calculation.

Rotary Major Scan. Fig. 11 (a) shows that RMS is effective in improving video quality by mitigating the adjacency preservation issue, while causing negligible extra latency, as indicated by Table 2. On the contrary, existing scan methods, such as Zigzag scan, incur a significant latency increment when operating on huge 3D video token tensors. In addition, we find the loss curve of LinGen w/ Zigzag scan is almost the same as that of LinGen w/ RMS, as shown in Fig. 11 (b), indicating RMS achieves similar performance to Zigzag scan with a much lower extra latency.

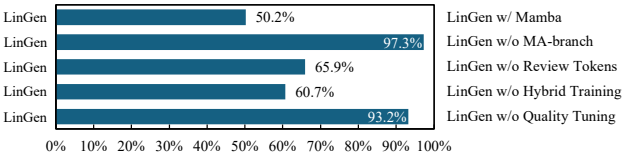

Figure 12. Win rates of human evaluation on quality comparison between the LinGen default setting and corresponding variants.

Model	Latency/s
LinGen (default setting)	102
LinGen w/o TESA	94 (-8)
LinGen w/o RMS	99 (-3)
LinGen w/ Zigzag	144 (+42)
LinGen w/ Mamba	127 (+25)
LinGen w/o MA-branch	65 (-37)
LinGen w/o review tokens	98 (-4)

Table 2. Latency of the LinGen default setting and variant settings when generating 512p 17s videos.

Mamba and Mamba2. Compared to Mamba, Mamba2 is more efficient and hardware-friendly [6]. Table 2 validates this, showing that LinGen w/ Mamba2 is 25% faster than LinGen w/ Mamba. Fig. 12 shows that LinGen w/ Mamba2 achieves almost the same video quality as LinGen w/ Mamba. In addition, although giving up the whole MA-branch brings significant speed-up, it severely impacts the quality of generated videos, as shown in Table 2 and Fig. 12, proving the necessity of including the MA-branch.

5. Conclusion

In the paper, we proposed LinGen, a linear-complexity text-to-video generation framework that enables high-resolution minute-length video generation on a single GPU. It replaces self-attention layers in DiTs with our novel MATE block, which inherits linear complexity from its two branches: MA-branch and TE-branch. Compared to the native Mamba block, MATE addresses its adjacency preservation issue and comprehensively enhances short-, medium-, and long-range correlations, improving the consistency and fidelity of generated videos significantly. Our experimental results show that LinGen achieves linear complexity and up to 11.5× speed-up in terms of latency, while maintaining the high quality of generated videos. LinGen presents a linear-complexity self-attention replacement, paving the way for broader adoption of this framework to hour-length video generation and real-time interactive video generation.

Acknowledgment

This work was supported in part by a Meta summer internship and in part by NSF under Grant No. CCF-2203399.

References

[1] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. https://openai.com/research/video-generation-models-as-world-simulators, 2024. 1, 2, 8

[2] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. VideoCrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 1, 3, 8

[3] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. VideoCrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024. 2, 6, 8

[4] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. DiffEdit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 1

[5] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. EMU: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 1, 5

[6] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024. 2, 4, 8

[7] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 2

[8] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M. Alabdulmohsin, et al. Patch n'Pack: NaViT, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The LLaMa 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 4

[10] Yu Gao, Jiancheng Huang, Xiaopeng Sun, Zequn Jie, Yujie Zhong, and Lin Ma. Matten: Video generation with Mamba-attention. *arXiv preprint arXiv:2405.03025*, 2024. 1, 3, 4

[11] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. EMU Video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. 1, 5

[12] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1, 2

[13] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent memory with optimal polynomial projections. *Advances in Neural Information Processing Systems*, 33:1474–1487, 2020. 2

[14] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 2

[15] Haoyang He, Yuhu Bai, Jiangning Zhang, Qingdong He, Hongxu Chen, Zhenye Gan, Chengjie Wang, Xiangtai Li, Guanzhong Tian, and Lei Xie. MambaAD: Exploring state space models for multi-class unsupervised anomaly detection. *arXiv preprint arXiv:2404.06564*, 2024. 1

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1

[17] 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. 3

[18] 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. 8

[19] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Bjorn Ommer. ZigMa: Zigzag Mamba diffusion model. *arXiv preprint arXiv:2403.13802*, 2024. 1, 4

[20] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 6

[21] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. MiraData: A large-scale video dataset with long durations and structured captions. *arXiv preprint arXiv:2407.06358*, 2024. 4

[22] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 1

[23] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. BK-SDM: A lightweight, fast, and cheap version of stable diffusion. *arXiv preprint arXiv:2305.15798*, 2023. 3

[24] Kling AI. Kling AI: Next-generation AI creative studio. https://klingai.com/, 2024. 2, 5, 7, 1, 4

[25] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. VideoPoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023. 2

[26] Pika Labs. Pika labs. https://www.pika.art/, 2024. 8

[27] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xFormers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022. 2

[28] Jiachen Li, Weixi Feng, Tsu-Jui Fu, Xinyi Wang, Sugato Basu, Wenhu Chen, and William Yang Wang. T2V-Turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. *arXiv preprint arXiv:2405.18750*, 2024. 1, 3, 6

[29] Jiachen Li, Qian Long, Jian Zheng, Xiaofeng Gao, Robinson Piramuthu, Wenhu Chen, and William Yang Wang. T2V-Turbo-v2: Enhancing video generation model post-training through data, reward, and conditional guidance design. *arXiv preprint arXiv:2410.05677*, 2024. 8

[30] Yaron Lipman, Ricky T.Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 5

[32] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. VDT: General-purpose video diffusion transformers via mask modeling. *arXiv preprint arXiv:2305.13311*, 2023. 3

[33] Luma Labs. Dream machine. https://lumalabs.ai/dream-machine, 2024. 2, 5, 1

[34] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3D point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 1

[35] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent Consistency Models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 3

[36] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023. 3

[37] Shentong Mo and Yapeng Tian. Scaling diffusion Mamba with bidirectional SSMs for efficient image and video generation. *arXiv preprint arXiv:2405.15881*, 2024. 1, 3

[38] Charlie Nash, Joao Carreira, Jacob Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models. *arXiv preprint arXiv:2203.09494*, 2022. 2

[39] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. 1

[40] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1

[41] 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. 1

[42] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie Gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 1, 2, 3, 6, 5

[43] RawFilm, Inc. RawFilm: 8k cinematic royalty-free stock footage. https://raw.film/, 2024. 6, 5

[44] Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. SAMBA: Simple hybrid state space models for efficient unlimited context language modeling. *arXiv preprint arXiv:2406.07522*, 2024. 6

[45] Yulin Ren, Xin Li, Mengxi Guo, Bingchen Li, Shijie Zhao, and Zhibo Chen. MambaCSR: Dual-interleaved scanning for compressed image super-resolution with SSMs. *arXiv preprint arXiv:2408.11758*, 2024. 1

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1

[47] Runway ML. Introducing Gen-3 alpha. https://runwayml.com/research/introducing-gen-3-alpha, 2024. 2, 5, 7, 1, 4, 8

[48] 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. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1

[49] Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. 1

[50] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu Edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024. 1

[51] Sam Shleifer, Jason Weston, and Myle Ott. NormFormer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*, 2021. 2

[52] Shutterstock, Inc. Shutterstock: Stock photos, royalty-free images, graphics, vectors, videos, and music. https://www.shutterstock.com/. 6, 5

[53] 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. 3

[54] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1

[55] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 3

[56] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, et al. UL2: Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022. 3

[57] Genmo Team. Mochi 1. https://github.com/genmoai/models, 2024. 4

[58] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. LION: Latent point diffusion models for 3D shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022. 1

[59] 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 *International Conference on Learning Representations*, 2022. 2

[60] Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. An empirical study of Mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024. 2

[61] Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K. Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16080–16089, 2024. 3

[62] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. ModelScope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 1, 2, 3, 6, 8

[63] Xiang Wang, Shiwei Zhang, Han Zhang, Yu Liu, Yingya Zhang, Changxin Gao, and Nong Sang. VideoLCM: Video latent consistency model. *arXiv preprint arXiv:2312.09109*, 2023. 1, 3

[64] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. LAVIE: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 3, 6, 8

[65] Yuqing Wang, Tianwei Xiong, Daquan Zhou, Zhijie Lin, Yang Zhao, Bingyi Kang, Jiashi Feng, and Xihui Liu. Loong: Generating minute-level long videos with autoregressive language models. *arXiv preprint arXiv:2410.02757*, 2024. 3, 1

[66] Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. *arXiv preprint arXiv:2410.08151*, 2024. 3, 1

[67] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023. 3

[68] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022. 3

[69] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 2

[70] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 1

[71] 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. 6, 1, 4

[72] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. MagViT: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023. 2

[73] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[74] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision*, pages 1–15, 2024. 8

[75] 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. https://github.com/hpcaitech/Open-Sora, 2024. 1, 2, 6, 7, 4, 8

LinGen: Towards High-Resolution Minute-Length Text-to-Video Generation with Linear Computational Complexity

Supplementary Material

A. Visual Examples

We have posted visual examples of videos generated using our LinGen model and other baseline models at a local website. You can open the `index.html` file with your website browser to investigate them. This local website is based on an online website template [39]. These visual examples include:

- **Video Demos.** 17-second and 68-second videos generated by LinGen (see Fig. 13).
- **Comparisons with existing video generation works.** Our baselines are typical open-source models (see Fig. 14), including T2V-Turbo [28], CogVideoX-5B [71], and OpenSora v1.2 [75], state-of-the-art accessible commercial models (see Fig. 15), including Kling [24], Runway Gen3 [47], and LumaLabs [33], and minute-length video generation trials (see Fig. 16), including Loong [65] and PA-VDM [66]. Note that PA-VDM has not yet released the code and prompts. Thus, we selected one LinGen-generated video similar to their demo video for reference.
- **Ablation experiments.** Video comparisons to validate the effectiveness of modules and techniques deployed in LinGen, including TEmporal Swin Attention (TESA), Rotary-Major Scan (RMS), review tokens, hybrid training, and quality-tuning (see Fig. 17 and Fig. 18).

B. Comparisons with Prior Works

In this section, we first supplement VBench results reported in Sec. B.1 in order to compare with more models and discuss the limitations of VBench. Then, we present visual examples of the generated videos to provide comparisons with prior works and include additional human evaluation results in Sec. B.2 to demonstrate high quality of videos generated by LinGen.

B.1. Automatic Metrics: VBench Results

We provide a more complete VBench-Long leaderboard in Table 3. We also evaluate LinGen on the standard VBench and compare it with other models on this leaderboard in Table 4. Note that most models on this leaderboard can only generate very short videos (usually shorter than 5 seconds). VBench also provides the option to perform evaluations with customized prompts, although only some of the quality metrics are supported. We evaluate LinGen with Movie Gen Bench prompts [42] and compare it with other models on the VBench-Custom leaderboard in Table 5.

The VBench results do not perfectly align with human preference. We find that Kling is more preferred in human evaluation than Runway Gen-3, but it obtains a lower VBench score. To further illustrate this point, as shown in Table 6, when we evaluate our model at 256p and 512p resolutions on VBench-Custom, they obtained similar scores. However, 512p-generated videos have a much higher win rate than 256p-generated videos in human evaluation of video quality.

B.2. Visual Examples and Human Evaluation

Given that the VBench results do not perfectly align with human preference, we provide more visual examples and human evaluation results to demonstrate the high quality of videos generated by LinGen in Fig. 15 and Fig. 19, respectively. Fig. 19 shows that LinGen outperforms typical open-source video generative models by a large margin.

C. More Ablation Experiments

We provide more visual examples of ablation experiments on the TESA block, RMS, review tokens, hybrid training, and quality-tuning in Fig. 17 and Fig. 18. This indicates that all of them contribute effectively to the consistency and high quality of the videos generated.

D. Model Implementation Details

In this section, we first provide more details of our model backbone in Sec. D.1. Then, we compare Mamba and Mamba2 and present their technical details in Sec. D.2. Finally, we give the details of our training recipe in Sec. D.3.

D.1. Backbone Details

LinGen learns a spatiotemporally compressed latent space using a Temporal AutoEncoder (TAE), designed similarly to the one in a prior work [42]. The TAE achieves a temporal compression rate of $8\times$ and a spatial compression rate of 8×8, followed by a $2\times2\times1$ patchification. LinGen uses a factorized learnable positional embedding [8] to enable arbitrary video size and length. We employ RMSNorm [73] and SwiGLU [49] in LinGen, with adaptive layer normalization conditioned on the time step [40].

After completing architectural design exploration depicted in Fig. 20, we employ 32 layers with 20 heads in each, with the dimension of embedding vectors being 2560.

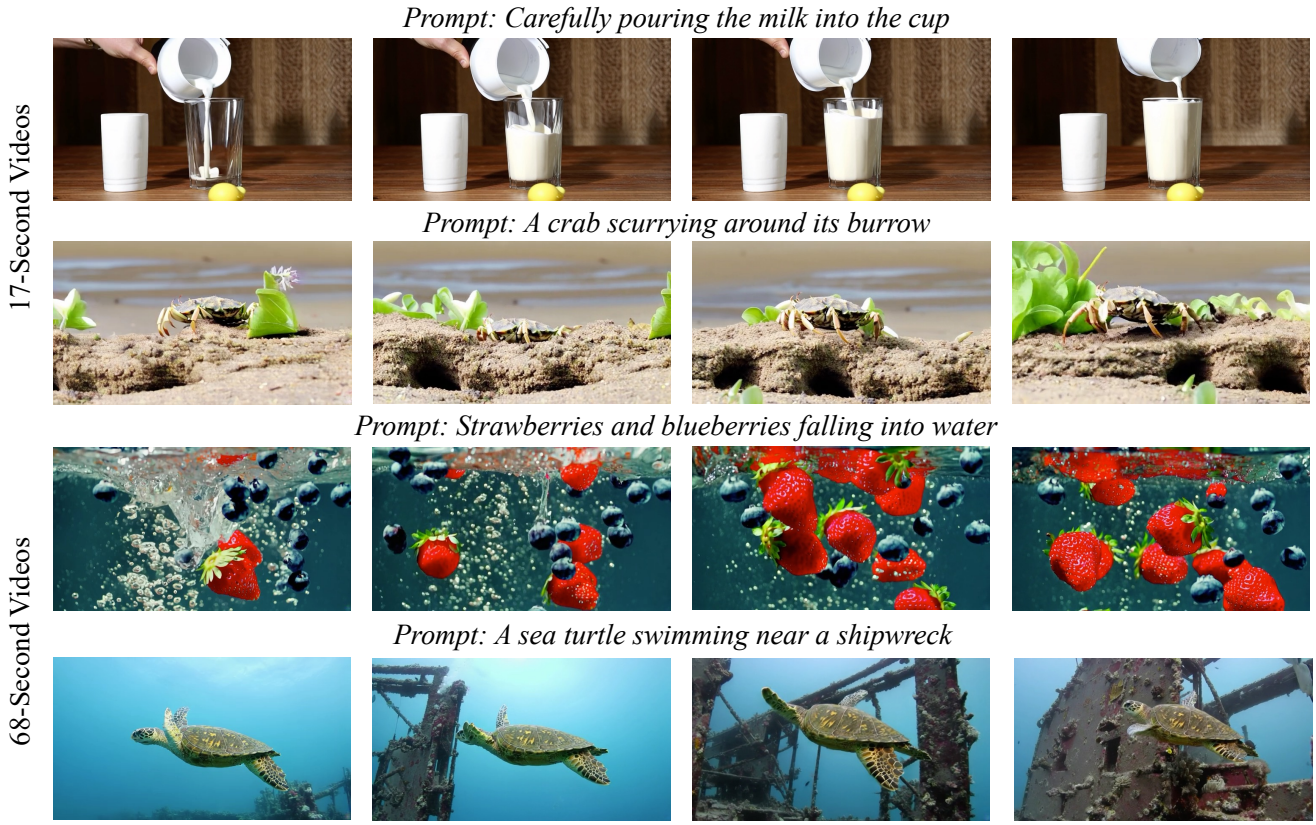

Figure 13. Examples of 17-second and 68-second videos generated by LinGen.

D.2. Mamba and Mamba2

SSMs have gained popularity in the field of natural language processing due to their high efficiency and strong performance in handling long sequences [13, 14]. Mamba [12], as a variant of SSM, enhances efficiency significantly by incorporating dynamic parameters into the SSM structure and developing algorithms optimized for better hardware compatibility. Based on this, Mamba2 [6] unifies SSMs and masks efficient attention by proposing a special SSM with an attention format (*i.e.*, Structured State Space Duality). Mamba2 removes sequential linear projections that are used in Mamba and produces SSM parameters A, B, C in parallel. The normalization layer in Mamba2 is the same as that in [51]. It improves stability. As mentioned in our main paper, the FLOPs cost of a bidirectional Mamba2 module is given by

$$C_{\text{bimamba}} = (6 + \frac{2}{d_h})END^2 + 4Nd_sd + O(Nd), \quad (4)$$

where E is the expansion factor, d is the dimension of token embedding vectors, N is the number of tokens, d_s is the hidden state size, and d_h is the head dimension of Mamba2, whose default value is 64. $O(Nd)$ includes the FLOPs cost of 1D convolution and the SSM block in Mamba2:

$$C_{\text{conv}} = 2EK(N + K - 1)d \quad (5)$$

$$C_{\text{SSM}} = 4ENd_sd + 2ENd \quad (6)$$

where K is the kernel size of 1D convolution. The above FLOPs should be doubled when the module is bidirectional.

Compared to Mamba, Mamba2 (1) has an attention format and thus benefits from existing efficient attention kernels, such as FlashAttention [7] and xFormers [27], (2) supports much larger hidden state sizes with lower latency, and (3) has better support for tensor parallelism for upscaling of the model [60].

Although Mamba2 compromises expressive power due to the simplification of the decay matrix in an SSM [6], it compensates for this using a much larger hidden state size. We set the hidden state size to 16 and 128 in LinGen w/ Mamba and LinGen w/ Mamba2, respectively, for both quality comparison and latency measurement, following their default values in the original design [6].

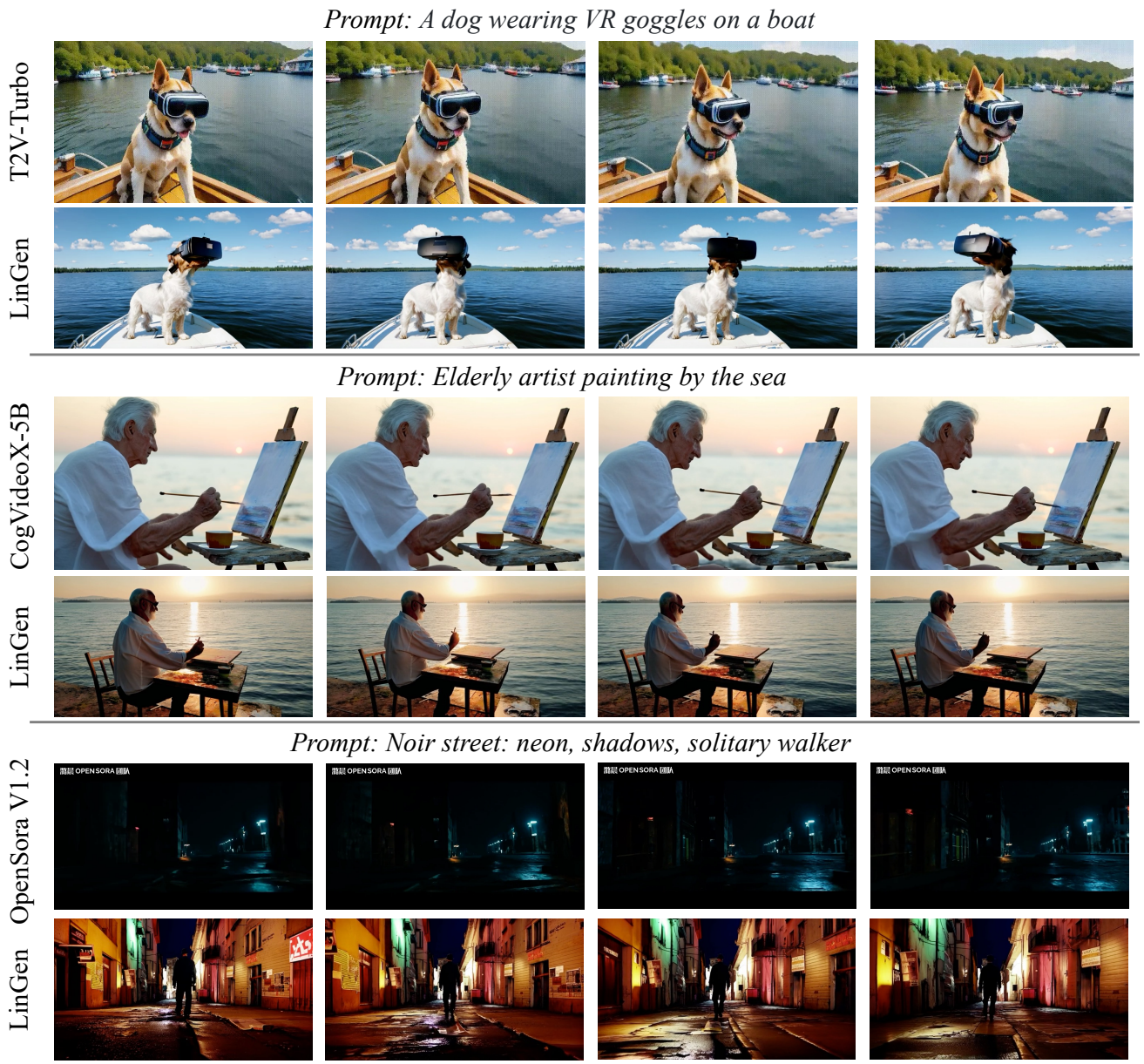

Figure 14. Comparisons with typical open-source video generative models.

D.3. Training Recipe Details

In this Section, we introduce our progressive training recipe in Sec. D.3.1. Then, we discuss our text-to-image and text-to-video hybrid training setting in Sec. D.3.2. We describe the details of our training datasets and quality-tuning design in Sec. D.3.3.

D.3.1 Progressive Training Recipe

We use a progressive recipe to pre-train our LinGen-4B model. As shown in Table 7, we first pre-train our model on the text-to-image task at a 256p resolution, followed by text-to-video pre-training at progressively higher resolutions and longer video lengths. In this progressive training schedule, the token sequence length in the latent space gradually increases.

D.3.2 Hybrid Training

In the text-to-video pre-training stages, we incorporate text-image pairs into the pre-training dataset and perform text-to-image and text-to-video joint training in practice. The sampling ratio of text-image pairs to text-video pairs is

Figure 15. Comparisons with state-of-the-art accessible commercial models.

Model	Subject Consist.	BG. Consis.	Temp. Flick.	Motion Smooth.	Aesthe. Quality	Imag. Quality	Dyna. Degree	**Quality Score**	**Total Score**	**Max. Raw Frames**
Runway Gen-3 [47]	97.10%	96.62%	98.61%	99.23%	60.14%	**63.34%**	**66.82%**	**84.11%**	**82.32%**	256
Kling [24]	**98.33%**	97.60%	99.30%	**99.40%**	46.94%	61.21%	65.62%	83.39%	81.85%	313
CogVideoX-5B [71]	96.23%	96.52%	98.66%	96.92%	**70.97%**	61.98%	62.90%	82.75%	81.61%	48
Mochi-1 [57]	96.99%	97.28%	99.40%	99.02%	61.85%	56.94%	60.64%	82.64%	80.13%	163
OpenSora V1.2 [75]	96.75%	**97.61%**	**99.53%**	98.50%	42.39%	56.85%	63.34%	81.35%	79.76%	408
Mira [21]	96.23%	96.92%	98.29%	97.54%	60.33%	42.51%	60.16%	78.78%	71.87%	60
LinGen	98.30%	97.60%	99.26%	98.58%	63.67%	60.55%	63.36%	83.77%	81.76%	**1088**

Model	Object Class	Multiple Objects	Human Action	Color	Spatial Relatio.	Scene	Appear. Style	Temp. Style	Overall Consist.	**Semantic Score**
Runway Gen-3 [47]	87.81%	53.64%	96.40%	80.90%	65.09%	**54.57%**	24.31%	**24.71%**	26.69%	75.17%
Kling [24]	87.24%	**68.05%**	93.40%	89.90%	**73.03%**	50.86%	19.62%	24.17%	26.42%	75.68%
CogVideoX-5B [71]	85.23%	62.11%	**99.40%**	82.81%	66.35%	53.20%	**24.91%**	25.38%	**27.59%**	**77.04%**
Mochi-1 [57]	86.51%	50.47%	94.60%	79.73%	69.24%	36.99%	20.33%	23.65%	25.15%	70.08%
OpenSora V1.2 [75]	82.22%	51.83%	91.20%	**90.08%**	68.56%	42.44%	23.95%	24.54%	26.85%	73.39%
Mira [21]	52.06%	12.52%	63.80%	42.24%	27.83%	16.34%	21.89%	18.77%	18.72%	44.21%
LinGen	**90.98%**	55.15%	97.50%	83.95%	58.15%	53.51%	21.08%	24.29%	26.32%	73.73%

Table 3. A more complete **VBench-Long** leaderboard. **Quality Score** measures the quality of generated videos and **Semantic Score** measures text-video alignment. **Total Score** represents their weighted sum. Higher values indicate better performance for all these metrics. LinGen can be seen to be comparable to state-of-the-art commercial models (*i.e.*, Gen-3 and Kling) and significantly outperform typical open-source models.

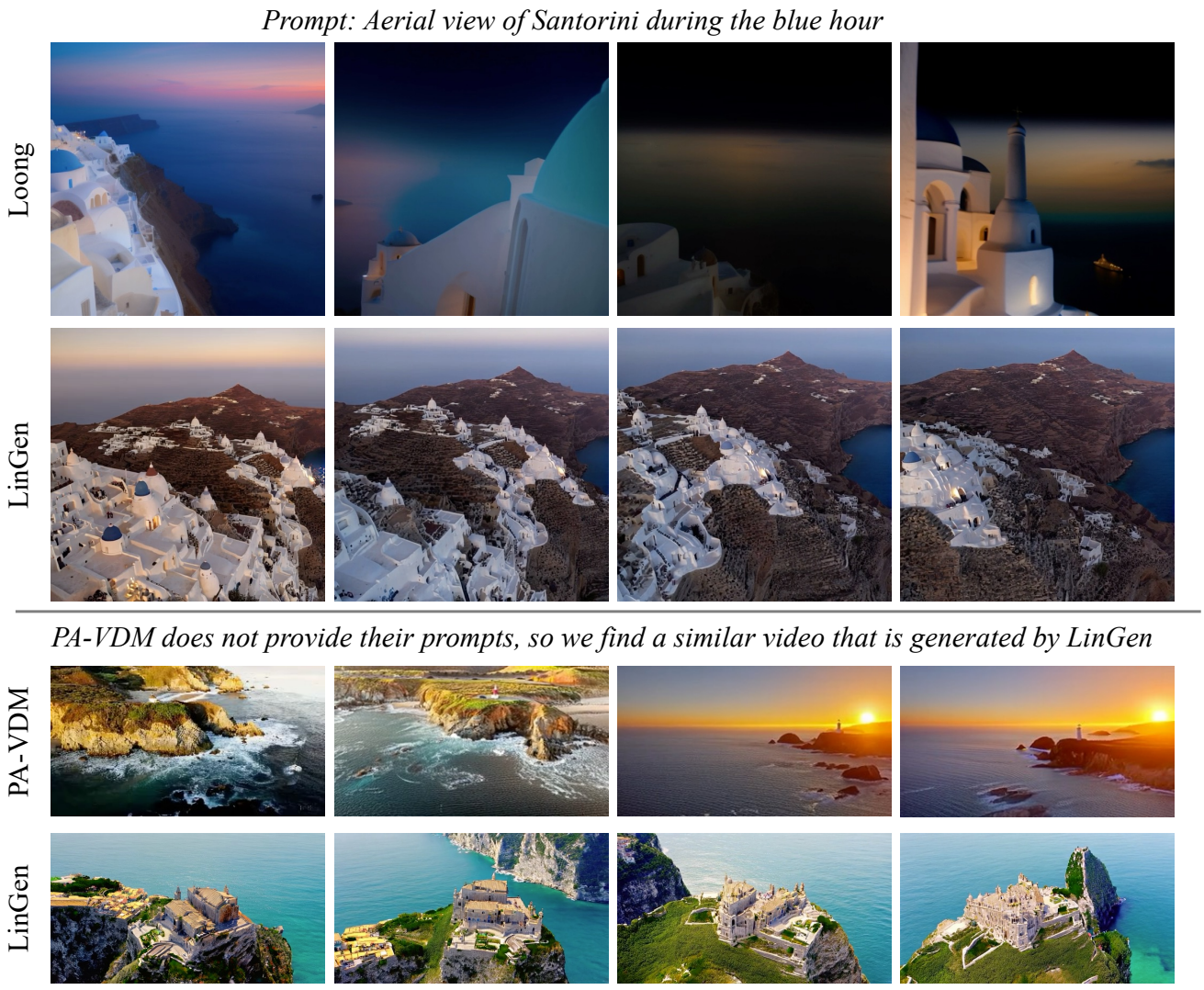

Figure 16. Comparisons with existing trials on generating minute-length videos.

1:100, which is very small, preventing this hybrid setting from reducing the motion of generated videos. We find such a hybrid training setting not only maintains the model's ability to generate images but also improves consistency of generated videos in some failure cases.

D.3.3 Quality Tuning and Datasets

We use a progressive training schedule to train our DiT-4B and LinGen-4B models. (1) Text-to-image pre-training at 256p resolution. We use the licensed ShutterStock [52] image dataset, which includes 300M text-image pairs, to train our models. (2) Text-to-video pre-training at 256p and 512p resolutions to generate 17s videos. We use the licensed ShutterStock video dataset, which includes 24M text-video pairs, to train our models. (3) Text-to-video pre-training at 512p resolution to generate 34s and 68s videos. We select 2.5M videos that are longer than 30 seconds from the licensed ShutterStock video dataset to train our models. (4) Text-to-video pre-training at 512p resolution to generate 68s videos. We select 145K videos that are longer than 60s from the licensed ShutterStock video dataset to train our models. (5) Text-to-video quality tuning at 512p resolution. For the 17s video generation, we select 3K videos with extremely high quality and good motions from the ShutterStock and RawFilm [43] video dataset to fine-tune our model. For 68s video generation, we select 300 minute-length videos with high quality and good motions from the ShutterStock video dataset to fine-tune our model.

The way that we select high-quality videos is similar to that in prior works [5, 42]. We first filter videos via automatic metrics, including aesthetic score and motion score.

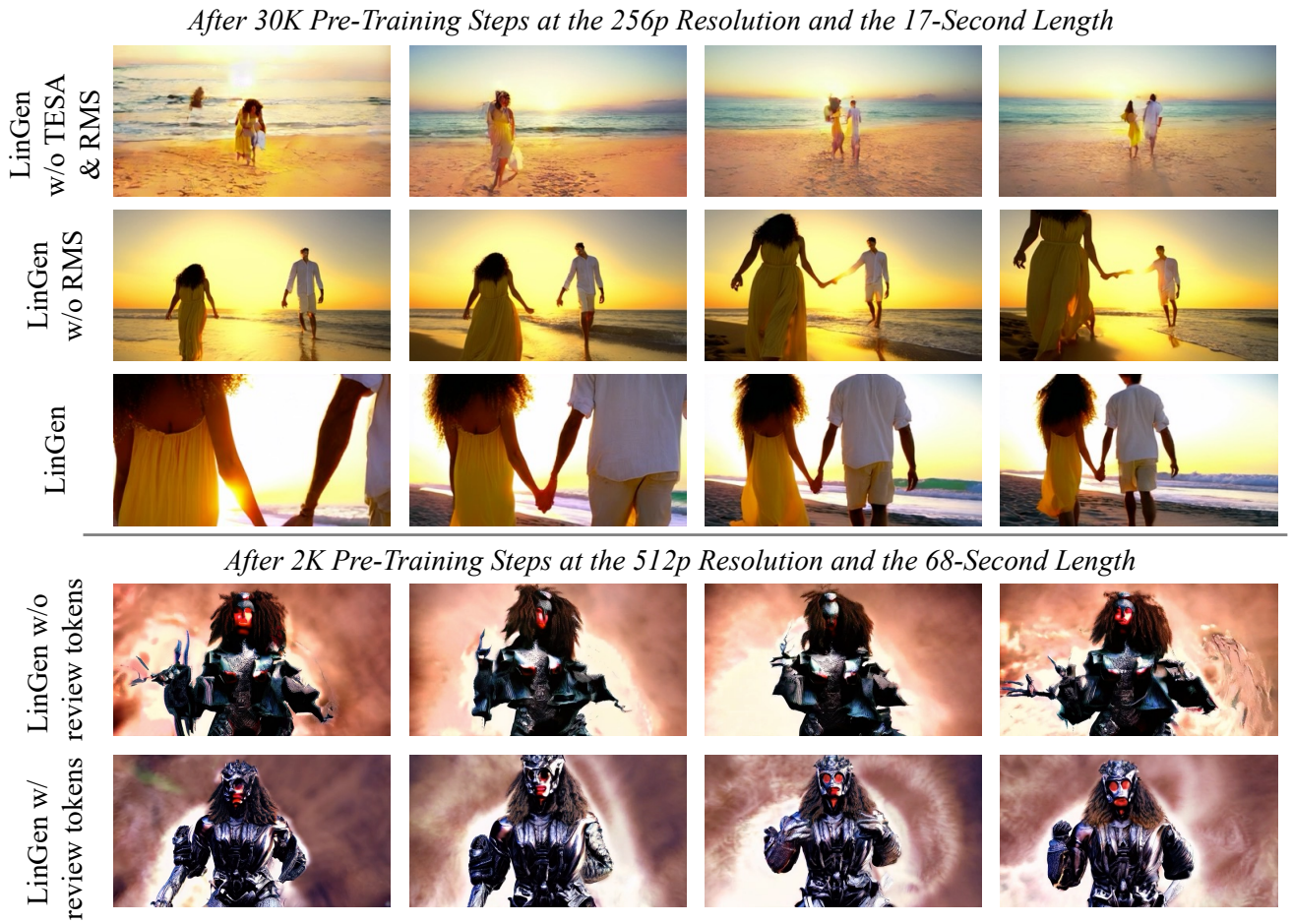

Figure 17. Visual examples of ablation experiments on the TESA block, RMS, and review tokens.

Then, we balance the concepts in the set of videos, manually identify cinematic videos, and manually caption the videos.

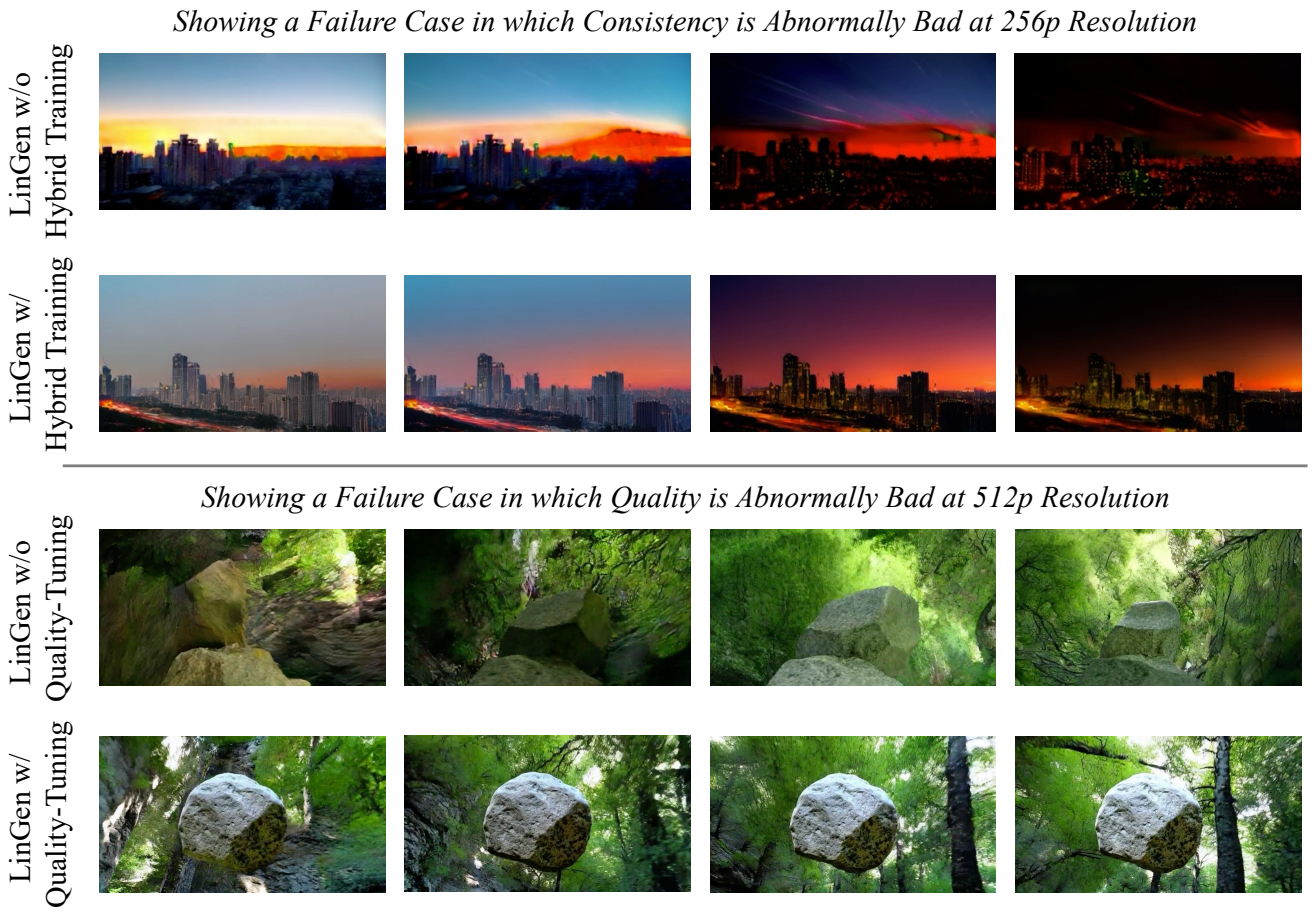

Figure 18. Visual examples of ablation experiments on hybrid training and quality-tuning.

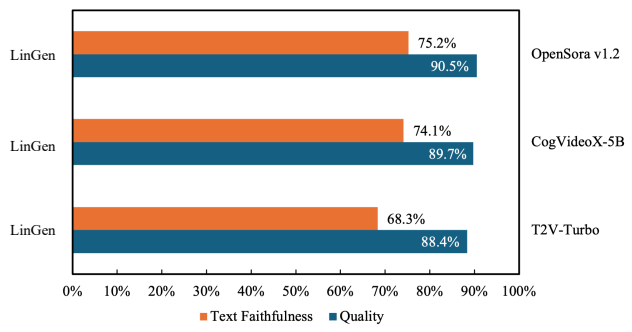

Figure 19. Win rates of human evaluation of quality and text-video alignment of videos generated by LinGen and typical open-source video generative models.

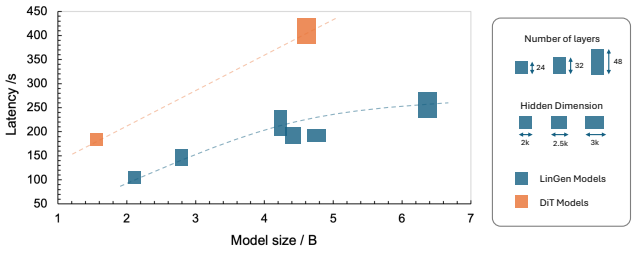

Figure 20. Latency of generating 512p 17s videos with different model designs. The latency of LinGen models scales more slowly with model size than self-attention-based standard DiT models. Note that we perform 100 inference steps to measure average latency. This is different from the default setting of 50 steps employed in our main paper.

Model	Subject Consist.	BG. Consis.	Temp. Flick.	Motion Smooth.	Aesthe. Quality	Imag. Quality	Dyna. Degree	**Quality Score**	**Total Score**	**Max. Raw Frames**
T2V-Turbo-v2 [29]	95.50%	96.71%	97.35%	97.07%	**90.00%**	62.61%	**71.78%**	**85.13%**	**83.52%**	16
Runway Gen-3 [47]	97.10%	96.62%	98.61%	99.23%	60.14%	63.34%	66.82%	84.11%	82.32%	256
LaVie-2 [64]	97.90%	**98.45%**	98.76%	98.42%	31.11%	**67.62%**	70.39%	83.24%	81.75%	61
Pika-1.0 [26]	96.94%	97.36%	**99.74%**	**99.50%**	47.50%	62.04%	61.87%	82.92%	80.69%	72
VideoCrafter-2.0 [3]	96.85%	98.22%	98.41%	97.73%	42.50%	63.13%	67.22%	82.20%	80.44%	16
OpenSora V1.2 [75]	96.75%	97.61%	99.53%	98.50%	42.39%	56.85%	63.34%	81.35%	79.76%	408
LinGen	**98.30%**	97.60%	99.26%	98.58%	63.67%	60.55%	63.36%	83.77%	81.76%	**1088**

Model	Object Class	Multiple Objects	Human Action	Color	Spatial Relatio.	Scene	Appear. Style	Temp. Style	Overall Consist.	**Semantic Score**
T2V-Turbo-v2 [29]	**95.33%**	61.49%	96.20%	92.53%	43.32%	**56.40%**	24.17%	**27.06%**	**28.26%**	**77.12%**
Runway Gen-3 [47]	87.81%	53.64%	96.40%	80.90%	65.09%	54.57%	24.31%	24.71%	26.69%	75.17%
LaVie-2 [64]	97.52%	**64.88%**	96.40%	91.65%	38.68%	49.59%	25.09%	25.24%	27.39%	75.76%
Pika-1.0 [26]	88.72%	43.08%	86.20%	90.57%	61.03%	49.83%	22.26%	24.22%	25.94%	71.77%
VideoCrafter-2.0 [3]	92.55%	40.66%	95.00%	**92.92%**	35.86%	55.29%	**25.13%**	25.84%	28.23%	73.42%
OpenSora V1.2 [75]	82.22%	51.83%	91.20%	90.08%	**68.56%**	42.44%	23.95%	24.54%	26.85%	73.39%
LinGen	90.98%	55.15%	**97.50%**	83.95%	58.15%	53.51%	21.08%	24.29%	26.32%	73.73%

Table 4. Automatic evaluation of LinGen on **VBench-standard**. **Quality Score** measures the quality of generated videos and **Semantic Score** measures text-video alignment. **Total Score** represents their weighted sum. Higher values indicate better performance for all these metrics.

Model	Subject Consistency	Background Consistency	Motion Smoothness	Aesthetic Quality	Imaging Quality	Dynamic Degree	**Quality Score**
Sora [1]	94.96%	95.84%	98.93%	60.30%	57.70%	**69.30%**	**79.69%**
Runway Gen-2 [47]	**97.61%**	97.61%	**99.58%**	**66.96%**	63.58%	18.89%	78.79%
Pika [26]	96.76%	**98.95%**	99.51%	63.15%	54.73%	37.22%	78.26%
VideoCrafter-1.0 [2]	95.10%	98.04%	95.67%	62.67%	61.99%	55.00%	78.14%
Show-1 [74]	95.53%	98.02%	98.24%	57.35%	59.75%	44.44%	77.50%
LaVie-Interpolation [64]	92.00%	97.33%	97.82%	54.00%	59.78%	46.11%	75.86%
LaVie [64]	91.41%	97.47%	96.38%	54.94%	61.90%	49.72%	75.75%
ModelScope [62]	89.87%	95.29%	95.79%	52.06%	58.57%	66.39%	74.91%
VideoCrafter-0.9 [2]	86.24%	92.88%	91.79%	44.41%	57.22%	89.72%	71.53%
CogVideo [18]	92.19%	96.20%	96.47%	38.18%	41.03%	42.22%	68.14%
LinGen	94.00%	96.08%	98.82%	57.86%	**67.39%**	44.92%	78.59%

Table 5. **VBench-Custom** results based on customized prompts. **Quality Score** represents the weighted sum of these supported metrics.

Model	Subject Consistency	Background Consistency	Motion Smoothness	Aesthetic Quality	Imaging Quality	Dynamic Degree	**Quality Score**	**Human Eval. Win Rate**
LinGen @ 512p	94.00%	96.08%	98.82%	57.86%	67.39%	44.92%	78.59%	88.4%
LinGen @ 256p	93.61%	96.55%	98.84%	48.83%	53.92%	66.98%	78.19%	11.6%

Table 6. VBench-Custom results of LinGen at different resolutions. Higher-resolution videos obtain a much higher win rate in human evaluation but only obtain a slightly higher VBench quality score. This indicates that VBench does not perfectly align with human preference.

Stage	# Steps	Batch size	GPU days
256p text-to-image	118k	8192	1189
256p text-to-video, 17s	125k	1024	1919
512p text-to-video, 17s	32k	512	2598
512p text-to-video, 34s	14k	512	2392
512p text-to-video, 68s	6k	256	1307

Table 7. The pre-training recipe of LVGen. The model was trained on Nvidia H100 GPUs.