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

Supplementary Material

A. Adjacency Preservation

Vanilla Mamba2 cannot be scaled to process huge images and video tokens well due to its long-range decay and the well-known adjacency preservation issue (see Sec. 1 of the main paper), causing distorted and inconsistent videos. Loss comparisons in Fig. 11 of the main paper and ablative videos (see Sec. B) validate the effectiveness of RMS and TESA. Mamba models scan image and video tokens into a sequence, where the ***minimum distance*** between originally adjacent tokens in k layers reflects their most precise correlation. For an $H \times W \times T$ token tensor, we compute ***its average***, d_k, among adjacent tokens in a $2 \times 2 \times 2$ cube and plot d_k in Fig. 1 for $H = W = T = 32$. RMS achieves the same d_k as Zigzag while being much more efficient and scalable (see Table 2 of the main paper). RMS and TESA thoroughly address the adjacency preservation issue.

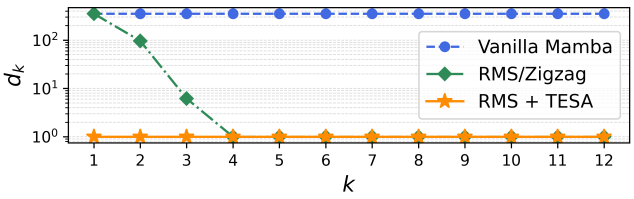

Figure 1. Average minimum distance between adjacent tokens.

B. Visual Examples

We provide visual examples that include:

- **Video Demos.** 17-second and 68-second videos generated by LinGen (see Fig. 2).
- **Comparisons with existing video generation works.** Our baselines are typical open-source models (see Fig. 3), including T2V-Turbo-v2 [16], CogVideoX-5B [31], and OpenSora v1.2 [34], state-of-the-art accessible commercial models (see Fig. 4), including Kling [13], Runway Gen3 [21], and LumaLabs [17], and minute-length video generation trials (see Fig. 5), including Loong [29] and PA-VDM [30]. 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. 6 and Fig. 7).

C. Comparisons with Prior Works

In this section, we first supplement VBench results reported in Sec. C.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. C.2 to demonstrate high quality of videos generated by LinGen.

C.1. Automatic Metrics: VBench Results

We provide a more complete VBench-Long leaderboard in Table 1. We also evaluate LinGen on the standard VBench and compare it with other models on this leaderboard in Table 2. 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 [19] and compare it with other models on the VBench-Custom leaderboard in Table 3.

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 4, when we evaluate our model at 256p and 512p resolutions on VBench-Custom, they obtain similar scores. However, 512p-generated videos have a much higher win rate than 256p-generated videos in human evaluation of video quality.

C.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. 4 and Fig. 8, respectively. Fig. 8 shows that LinGen outperforms typical open-source video generative models by a large margin.

D. 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. 6 and Fig. 7. This indicates that all of them contribute effectively to the consistency and high quality of the videos generated.

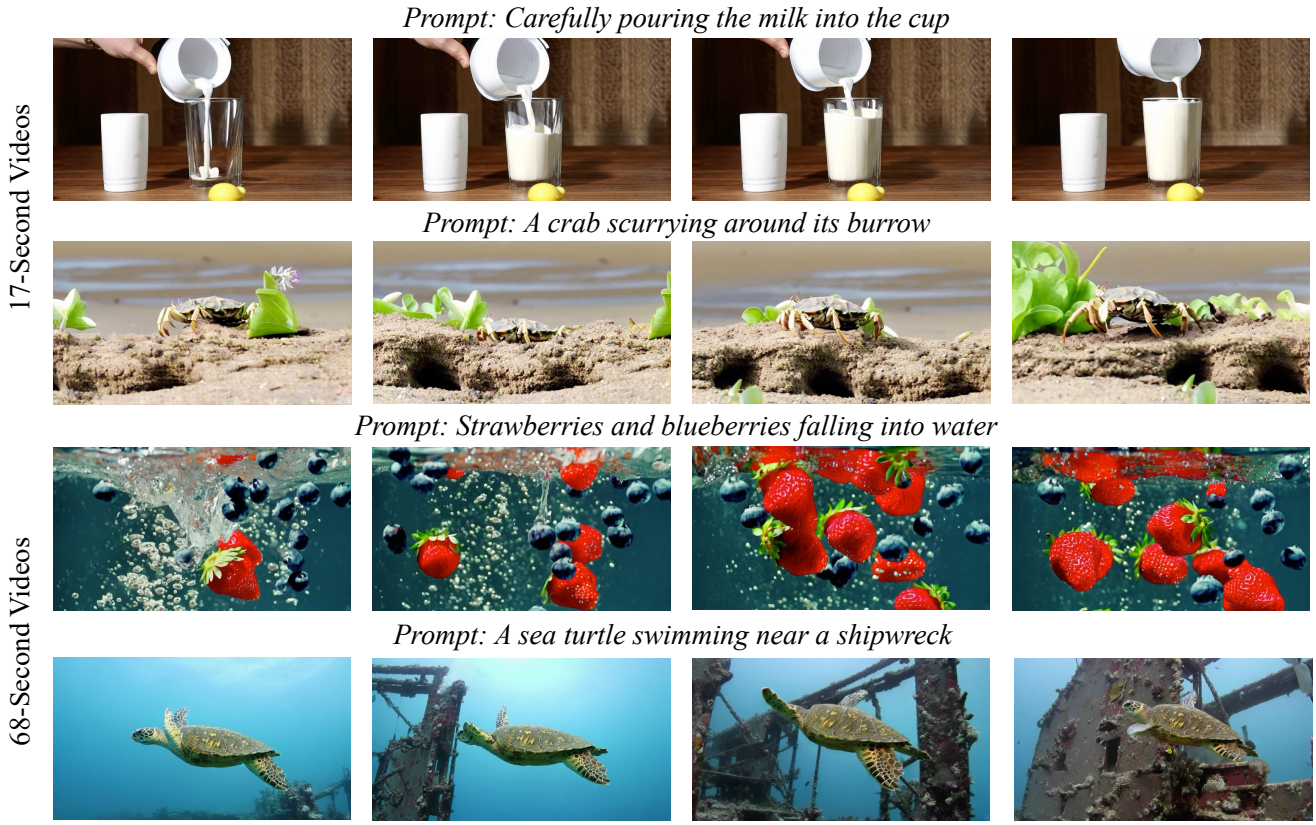

Figure 2. Examples of 17-second and 68-second videos generated by LinGen.

E. Model Implementation Details

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

E.1. Backbone Details

LinGen learns a spatiotemporally compressed latent space using a Temporal AutoEncoder (TAE), designed similarly to the one in a prior work [19]. The TAE achieves a temporal compression rate of 8× and a spatial compression rate of 8×8, followed by a 2×2×1 patchification. LinGen uses a factorized learnable positional embedding [7] to enable arbitrary video size and length. We employ RMSNorm [32] and SwiGLU [22] in LinGen, with adaptive layer normalization conditioned on the time step [18].

After completing architectural design exploration depicted in Fig. 9, we employ 32 layers with 20 heads in each, with the dimension of embedding vectors being 2560.

E.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 [9, 10]. Mamba [8], 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 [5] 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 [23]. It improves stability. As mentioned in our main paper, the FLOPs cost of a bidirectional Mamba2 module is given by

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

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. $O(Nd)$ includes the FLOPs cost of 1D convolution and the SSM block in Mamba2:

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

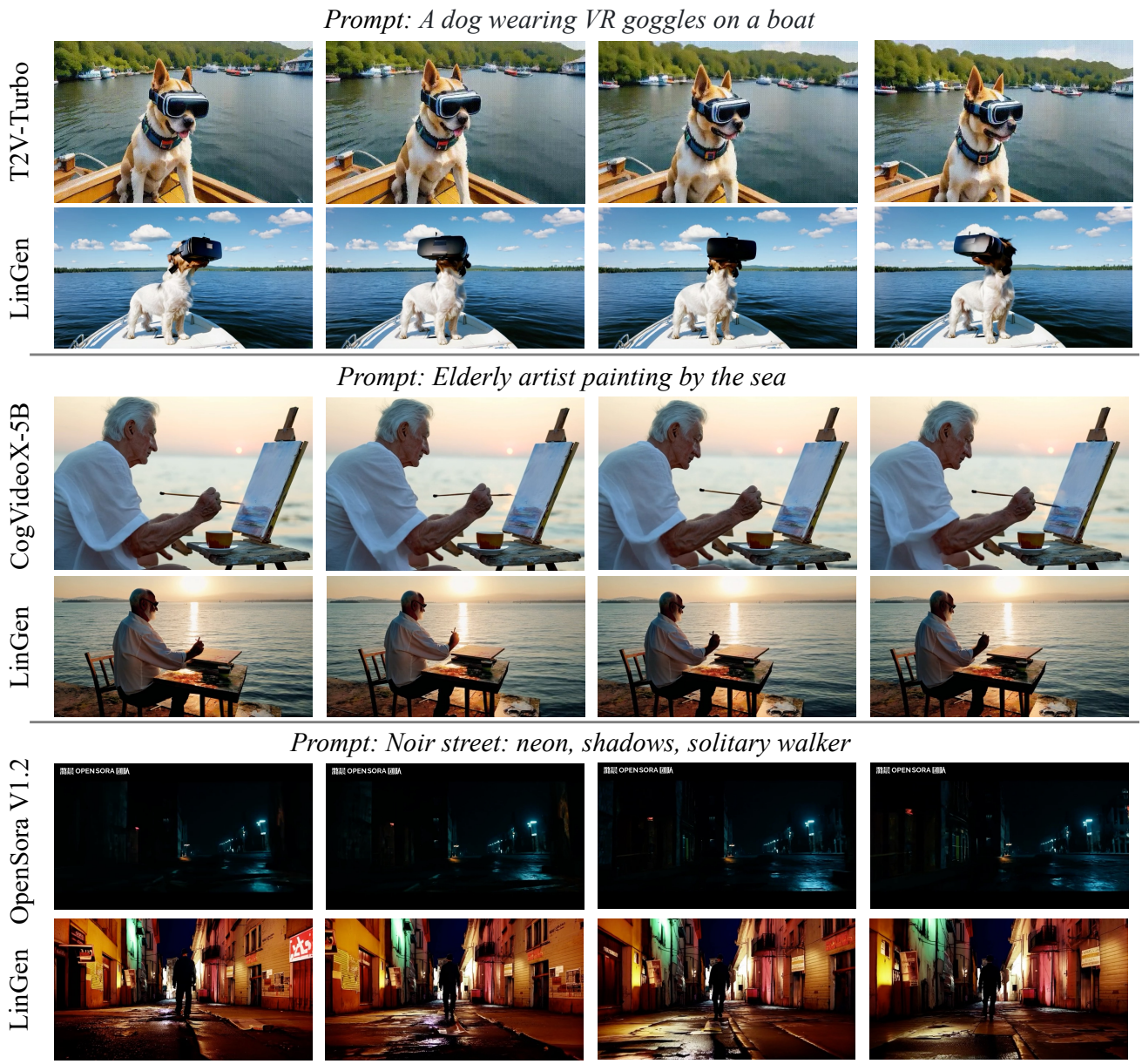

Figure 3. Comparisons with typical open-source video generative models.

$$C_{\text{SSM}} = 4ENd_s d + 2ENd \tag{3}$$

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 [6] and xFormers [15], (2) supports much larger hidden state sizes with lower latency, and (3) has better support for tensor parallelism for upscaling of the model [26].

Although Mamba2 compromises expressive power due to the simplification of the decay matrix in an SSM [5], 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 [5].

E.3. Training Recipe Details

In this section, we introduce our progressive training recipe in Sec. E.3.1. Then, we discuss our text-to-image and text-to-video hybrid training setting in Sec. E.3.2. We describe

Figure 4. 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 [21]	97.10%	96.62%	98.61%	99.23%	60.14%	**63.34%**	**66.82%**	**84.11%**	**82.32%**	256
Kling [13]	**98.33%**	97.60%	99.30%	**99.40%**	46.94%	61.21%	65.62%	83.39%	81.85%	313
CogVideoX-5B [31]	96.23%	96.52%	98.66%	96.92%	**70.97%**	61.98%	62.90%	82.75%	81.61%	48
Mochi-1 [25]	96.99%	97.28%	99.40%	99.02%	61.85%	56.94%	60.64%	82.64%	80.13%	163
OpenSora V1.2 [34]	96.75%	**97.61%**	**99.53%**	98.50%	42.39%	56.85%	63.34%	81.35%	79.76%	408
Mira [12]	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 [21]	87.81%	53.64%	96.40%	80.90%	65.09%	**54.57%**	24.31%	**24.71%**	26.69%	75.17%
Kling [13]	87.24%	**68.05%**	93.40%	89.90%	**73.03%**	50.86%	19.62%	24.17%	26.42%	75.68%
CogVideoX-5B [31]	85.23%	62.11%	**99.40%**	82.81%	66.35%	53.20%	**24.91%**	25.38%	**27.59%**	**77.04%**
Mochi-1 [25]	86.51%	50.47%	94.60%	79.73%	69.24%	36.99%	20.33%	23.65%	25.15%	70.08%
OpenSora V1.2 [34]	82.22%	51.83%	91.20%	**90.08%**	68.56%	42.44%	23.95%	24.54%	26.85%	73.39%
Mira [12]	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 1. 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.

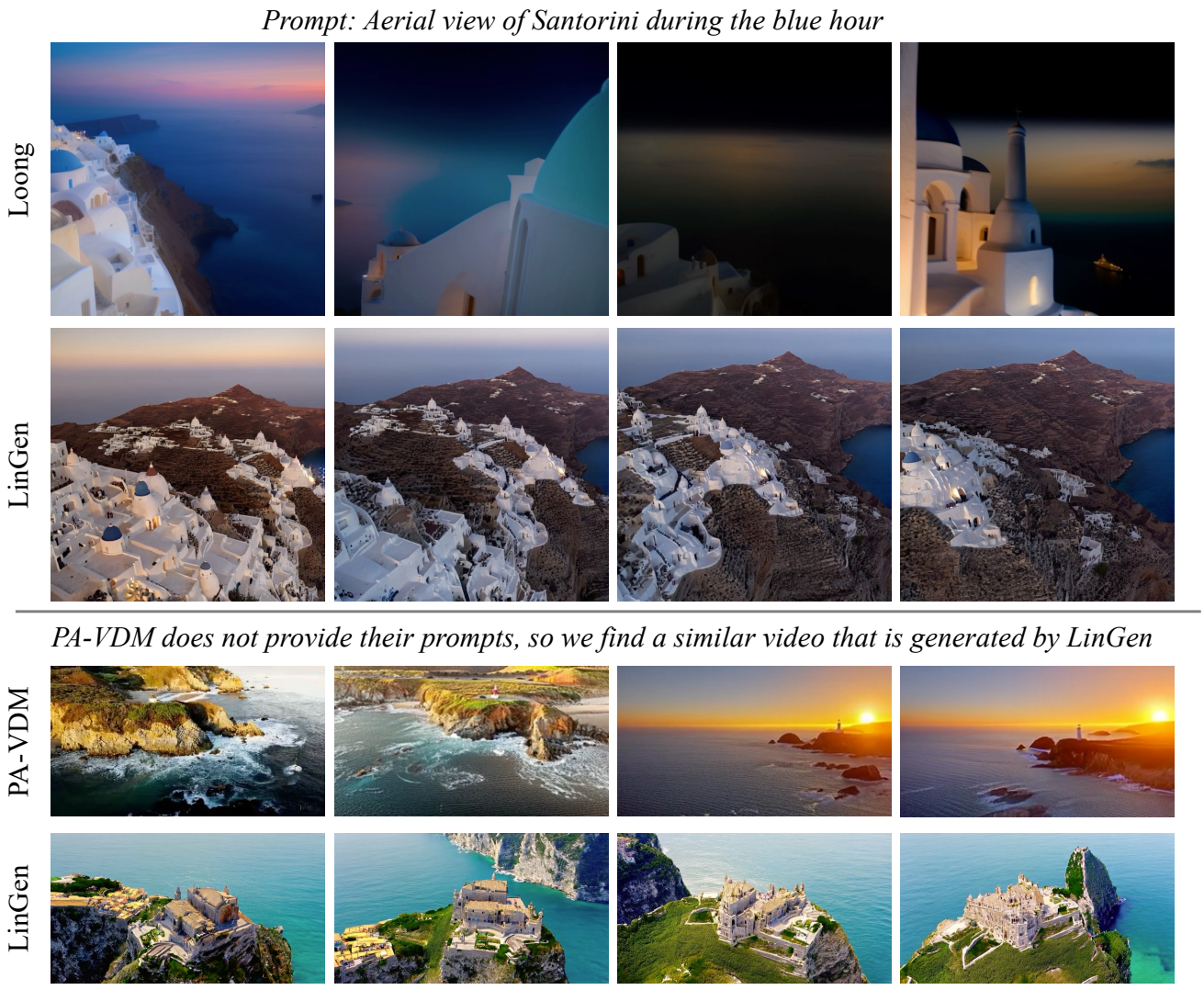

Figure 5. Comparisons with existing trials on generating minute-length videos.

the details of our training datasets and quality-tuning design in Sec. E.3.3.

E.3.1 Progressive Training Recipe

We use a progressive recipe to pre-train our LinGen-4B model. As shown in Table 5, 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.

E.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 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.

E.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 [24] 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

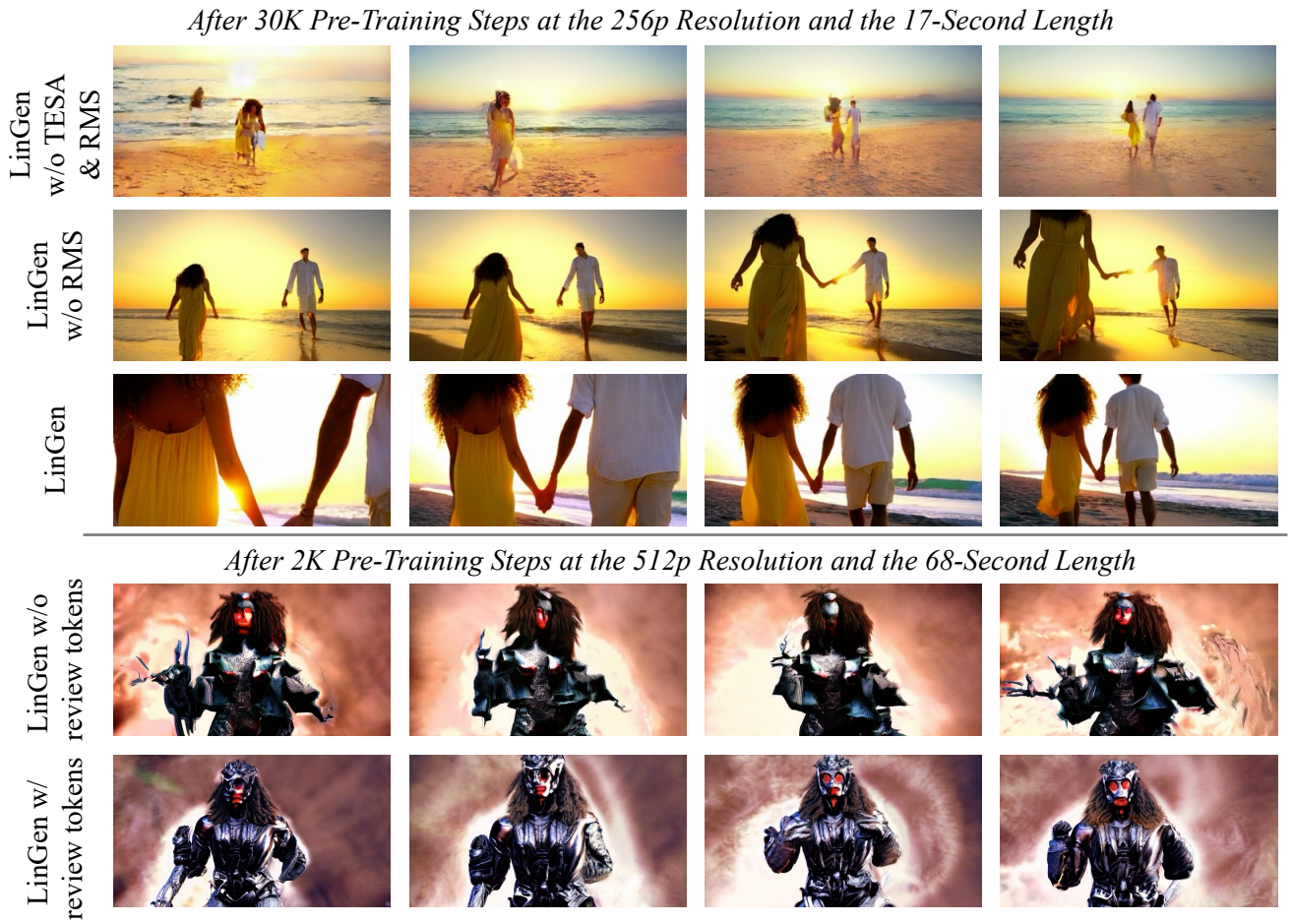

Figure 6. Visual examples of ablation experiments on the TESA block, RMS, and review tokens.

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 [20] 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 [4, 19]. We first filter videos via automatic metrics, including aesthetic score and motion score. Then, we balance the concepts in the set of videos, manually identify cinematic videos, and manually caption them.

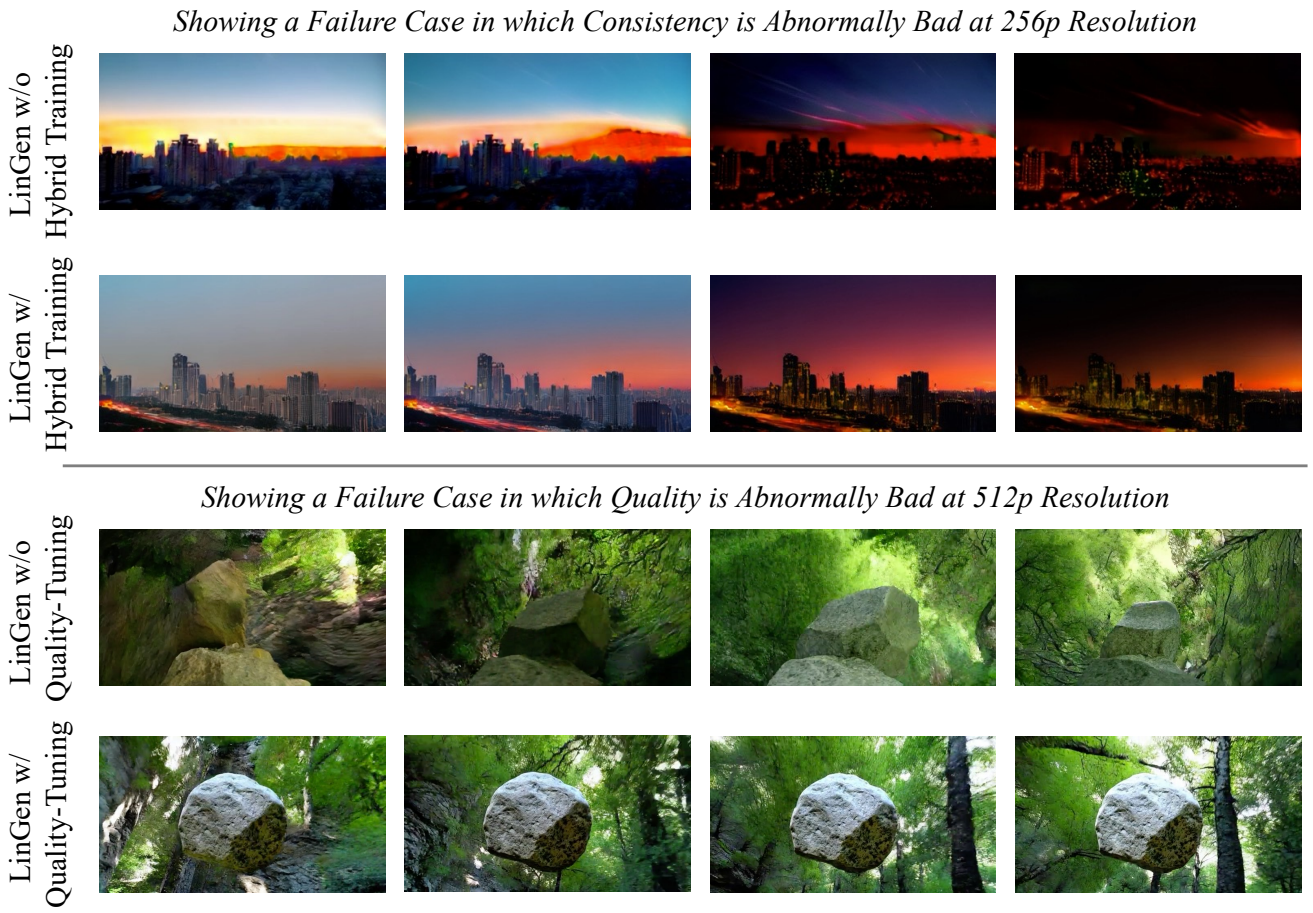

Figure 7. Visual examples of ablation experiments on hybrid training and quality-tuning.

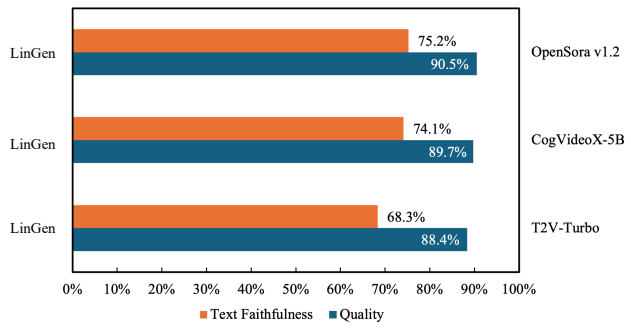

Figure 8. Win rates of human evaluation of quality and text-video alignment of videos generated by LinGen and typical open-source video generative models.

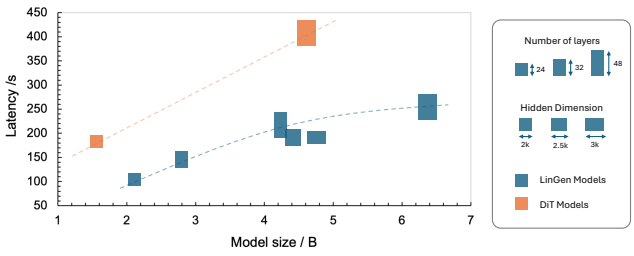

Figure 9. Latency of generating 512p 17s videos with different model designs. The latency of LinGen models scales more slowly with model size than self-attention-based standard DiT models. Note that we perform 100 inference steps to measure average latency. This is different from the default setting 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 [16]	95.50%	96.71%	97.35%	97.07%	**90.00%**	62.61%	**71.78%**	**85.13%**	**83.52%**	16
Runway Gen-3 [21]	97.10%	96.62%	98.61%	99.23%	60.14%	63.34%	66.82%	84.11%	82.32%	256
LaVie-2 [28]	97.90%	**98.45%**	98.76%	98.42%	31.11%	**67.62%**	70.39%	83.24%	81.75%	61
Pika-1.0 [14]	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 [34]	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 [16]	**95.33%**	61.49%	96.20%	92.53%	43.32%	**56.40%**	24.17%	**27.06%**	**28.26%**	**77.12%**
Runway Gen-3 [21]	87.81%	53.64%	96.40%	80.90%	65.09%	54.57%	24.31%	24.71%	26.69%	75.17%
LaVie-2 [28]	97.52%	**64.88%**	96.40%	91.65%	38.68%	49.59%	25.09%	25.24%	27.39%	75.76%
Pika-1.0 [14]	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 [34]	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 2. 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 [21]	**97.61%**	97.61%	**99.58%**	**66.96%**	63.58%	18.89%	78.79%
Pika [14]	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 [33]	95.53%	98.02%	98.24%	57.35%	59.75%	44.44%	77.50%
LaVie-Interpolation [28]	92.00%	97.33%	97.82%	54.00%	59.78%	46.11%	75.86%
LaVie [28]	91.41%	97.47%	96.38%	54.94%	61.90%	49.72%	75.75%
ModelScope [27]	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 [11]	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 3. **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 4. 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 5. The pre-training recipe of LVGen. The model was trained on Nvidia H100 GPUs.

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. 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. 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. 8

[4] 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. 6

[5] 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, 3

[6] 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. 3

[7] 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. 2

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

[9] 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

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

[11] 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

[12] 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

[13] Kling AI. Kling AI: Next-generation AI creative studio. https://klingai.com/, 2024. 1, 4

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

[15] 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. 3

[16] 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. 1, 8

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

[18] 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. 2

[19] 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, 6

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

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

[22] Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. 2

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

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

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

[26] Roger Waleffe et al. An empirical study of Mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024. 3

[27] 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. 8

[28] 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. 8

[29] 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. 1

[30] 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. 1

[31] 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. 1, 4

[32] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[33] 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

[34] 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, 4, 8