InfinityStar: Unified Spacetime AutoRegressive Modeling for Visual Generation

Jinlai Liu[*], **Jian Han**[*], **Bin Yan**[*] **Hui Wu,** **Fengda Zhu,** **Xing Wang**

Yi Jiang, **Bingyue Peng,** **Zehuan Yuan**[†]

ByteDance

{liujinlai.licio,hanjian.thu123,bin.yan,wuhui.321,fengdazhu}@bytedance.com,
{xing.wang,jiangyi.enjoy,bingyue.peng,yuanzehuan}@bytedance.com,

Codes and models: https://github.com/FoundationVision/InfinityStar

Abstract

We introduce InfinityStar, a unified spacetime autoregressive framework for high-resolution image and dynamic video synthesis. Building on the recent success of autoregressive modeling in both vision and language, our purely discrete approach jointly captures spatial and temporal dependencies within a single architecture. This unified design naturally supports a variety of generation tasks such as text-to-image, text-to-video, image-to-video, and long-duration video synthesis via straightforward temporal autoregression. Through extensive experiments, InfinityStar scores 83.74 on VBench, outperforming all autoregressive models by large margins, even surpassing diffusion competitors like HunyuanVideo. Without extra optimizations, our model generates a 5s, 720p video approximately 10× faster than leading diffusion-based methods. To our knowledge, InfinityStar is the first discrete autoregressive video generator capable of producing industrial-level 720p videos. We release all code and models to foster further research in efficient, high-quality video generation.

1 Introduction

Visual synthesis has witnessed remarkable progress in recent years, largely propelled by the scaling of Transformer architectures. In particular, video generation has attracted growing interest from both academia and industry, owing to its wide-ranging applications in content creation, world simulation, etc. At present, diffusion models[3, 18, 17, 28, 8, 39] lead the field by iteratively denoising latent representations to produce high-fidelity clips. Concurrently, autoregressive models[16, 30, 9] have been explored for their potential to unify image and video generation and to generalize over longer time horizons.

Despite their successes, each paradigm exhibits critical shortcomings. Video diffusion models excel at synthesizing fixed-length frame sequences by exploiting bidirectional attention, yet they incur substantial computational cost due to tens or even hundreds of sequential denoising steps, and they struggle to extend seamlessly to video extrapolation. Autoregressive methods based on next-token prediction, while inherently capable of streaming generation, often fall short in visual fidelity and suffer from prohibitive latency due to tens of thousands of inference steps.

These observations motivate the need for a generation framework that simultaneously possess high visual quality, efficiency and temporal generalization. Recently, Visual AutoRegressive modeling (VAR)[25] redefined image generation as a coarse-to-fine next-scale prediction. Its follow-up work, Infinity [13] further introduces bitwise modeling and scales up the vocabulary size, achieving

[*]Equal contribution. [†]Corresponding author: yuanzehuan@bytedance.com

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

comparable performance to diffusion models while offering significant advantages in inference speed. Inspired by the success of VAR [25] and Infinity [13], we present InfinityStar, a Spacetime Pyramid Modeling for unified text-to-image, text-to-video, zero-shot image-to-video, and zero-shot video extrapolation. This framework models a video as an image pyramid and multiple clip pyramids, not only naturally inheriting the text-to-image capabilities but also decoupling static appearance from dynamic motions in videos. Furthermore, we introduce several key improvements. First, we improve discrete reconstruction quality by leveraging knowledge inheritance from a continuous video tokenizer. Second, we introduce Stochastic Quantizer Depth during training of the tokenizer to alleviate the imbalanced information distribution across scales. Third, we propose Semantic Scales Repetition, which refines the predictions of earlier semantic scales in a video, significantly enhancing fine-grained details and complex motions of the generated videos.

We train InfinityStar on large-scale video corpora to support up to 720p resolution and variable durations. On the VBench benchmark[38], InfinityStar establishes a new state-of-the-art among autoregressive video models, even surpassing industry-leading HunyuanVideo[17] (83.74 v.s 83.24). Besides, InfinityStar shows a great advantage in terms of speed. Using visual tokenizers of the same compression rate, InfinityStar achieves a $10\times$ reduction in inference latency relative to leading diffusion models.

In summary, the main contributions of our work are as follows:

1. We propose InfinityStar, a novel spacetime pyramid modeling framework that unifies diverse visual generation tasks, demonstrating superior flexibility and versatility.
2. InfinityStar is the first discrete autoregressive model capable of generating high-quality videos, outperforming existing autoregressive text-to-video models and matching the performance of leading diffusion models.
3. Compared to the inefficiency of existing autoregressive models and diffusion models, InfinityStar significantly accelerates high-quality video generation.

2 Related Work

2.1 Video Diffusion Models

Diffusion models excel at generating high-fidelity data by gradually denoising random noise and can naturally extend to video generation. Early attempts [2, 5, 36] are built on U-Net architectures, demonstrating the feasibility of this approach but falling short in producing sharp, temporally coherent frames due to limited model capacity. The advent of Diffusion Transformers (DiT [21]) marked a turning point. SORA [3] harnessed DiT's scaling ability to process spatio-temporal patches at scale, dramatically improving both video consistency and generation quality. Inspired by SORA's success, industry efforts [34, 17, 28] have further advanced the field, pushing video generation to new heights. Although video diffusion models deliver outstanding quality, their slow generation speed hinders the production of high-resolution, long-duration videos.

2.2 Video AutoRegressive Models

Another class of methods [30, 9, 16] employs autoregressive models for video generation. Inspired by the success of LLMs, these works predict video tokens in specific orders using an autoregressive Transformer. For example, Emu3 [30] performs next-token prediction along both spatial and temporal axes, while NOVA [9] first predicts spatial tokens set-by-set and subsequently proceeds frame-by-frame in the temporal dimension. Although achieving preliminary progress, they require hundreds to thousands of inference steps, resulting in prohibitively low generation efficiency. In contrast, recent advances in next-scale prediction [25, 13] have demonstrated state-of-the-art performance in image synthesis, offering both improved quality and markedly faster inference. In this work, we extend the next-scale prediction paradigm to the unified tasks of image and video generation.

2.3 Discrete Video Tokenizers

For a long time, discrete [33, 35] and continuous [17, 34, 28] video tokenizers have been developed independently. Although some works [1, 29] provide both discrete and continuous tokenizers, the

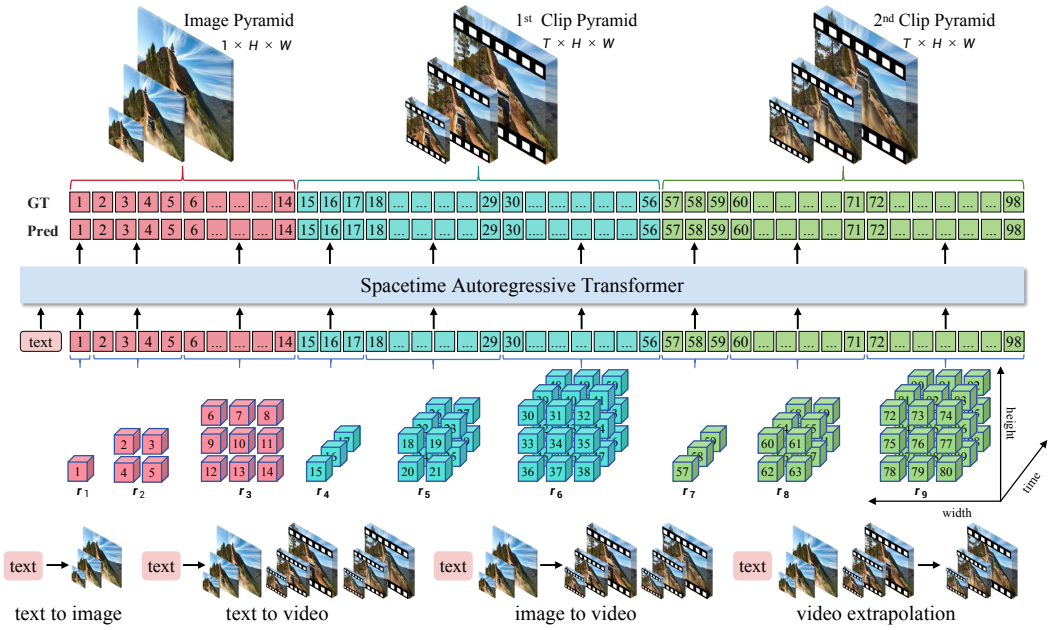


Figure 1: **Spacetime pyramid modeling of InfinityStar.** Built with an unified autoregressive pipeline, InfinityStar is capable of performing text-to-image, text-to-video, image-to-video, video extrapolation tasks all in one model.

network configurations are usually not aligned. For example, Cosmos [1] chooses 6 and 16 as latent dimensions in its discrete and continuous variants respectively. This misalignment hinders the knowledge reuse between two types of tokenizers. As a result, most mainstream discrete video tokenizers are either trained from scratch [1] or starting from a pretrained discrete image tokenizer [35, 29]. However, these training strategies have the following drawbacks. First, training from scratch is inefficient and converges slowly. Second, weights pretrained on static images are not optimal for video reconstruction. To alleviate these deficiencies, we propose a new training strategy, which inherits the architecture and knowledge of a trained continuous video tokenizer. Experiments show that this strategy significantly boosts the convergence of discrete video tokenizers.

3 InfinityStar Architecture

3.1 Preliminaries

Infinity for Image Generation. Infinity [13] decomposes an image into a sequence of hierarchical token blocks using a visual tokenizer and models the relationship between tokens by a visual autoregressive Transformer (VAR Transformer). To cover images of various sizes, Infinity pre-defines a list of token block sizes $\{(h_1, w_1), ...(h_K, w_K)\}$, called scale schedule. The size (h_i, w_i) in scale schedule grows as i increases, forming a pyramid-like structure, which we refer as **image pyramid** in later discussion. Next we introduce the training and inference procedure of Infinity.

In the first training stage, a visual tokenizer learns to reconstruct the raw image and compress it into a sequence of discrete tokens, which can be modeled by the VAR Transformer in the next stage. Specifically, the tokenizer first encodes the raw images into compact latents, then transforms latents into K discrete residual token blocks $(r_1, r_2, ..., r_K)$ using a bitwise multi-scale residual quantizer [13]. Each token block r_i consists of $h_i \times w_i$ discrete tokens of d-dim with vocabulary size of 2^d. Then in the second stage, a VAR Transformer is trained to predict next residual token block r_k conditioned on text embedding $\psi(t)$ and former tokens blocks $r_{<k}$. Formally, in each step, VAR Transformer predicts a conditional probability $p(r_k|r_{<k}, \psi(t))$. During the inference, Infinity generates an image by running the VAR Transformer K times autoregressively, merging the predicted tokens and running the tokenizer decoder once.

3.2 Spacetime Pyramid Modeling for Unified Generation

Extending the spatial-only next-scale prediction paradigm of Infinity [13] to video generation presents a primary challenge: *how to incorporate the temporal dimension*. The straightforward strategies are either letting time grows uniformly, *i.e.*, from $(1, 1, 1)$ to (T, H, W), or keeping time constant, *i.e.*, from $(T, 1, 1)$ to (T, H, W). We empirically found that letting time grow uniformly produces flickering videos. As for the constant time pyramid, we refer to it as the **pseudo-spacetime pyramid**. Despite its conceptual simplicity, it suffers from two fundamental limitations. First, the treatment of videos differs markedly from that of images, preventing a text-to-video (T2V) model from effectively leveraging the knowledge learned by a text-to-image (T2I) model and complicating its extension to tasks such as image-to-video (I2V). Second, because appearance and motion in videos are coupled in this design, the model faces significant difficulty in accurately fitting both aspects.

To overcome these challenges, we propose a novel **spacetime pyramid modeling** framework as shown in Fig.1. Each video is decomposed into sequential clips $\{c_1, c_2, \cdots, c_N\}$. We regard the first frame as c_1 (*i.e.*, $T = 1$) to encode video main static appearance cues specifically and other clips share an equal duration $T > 1$. Each clip is modeled as a 3D volume pyramid similar as Infinity [13]. In particular, for each clip, there are K scales with each represented as a residual token block r_k of (T, h_k, w_k) dimension. *It is worth noting that all scales in the pyramid are extended only in spatial dimension instead of time*. Mathematically, the clip tokens are generated auto-regressively across scales as:

$$p(r_1^1, \ldots, r_K^1) = \prod_{k=1}^{K} p(r_k^1 \mid r_1^1, \ldots, r_{k-1}^1, \psi(t)), \tag{1}$$

For inter-clip predictions, clips are generated sequentially conditioned on prior clip predictions and the text input in an autoregressive manner. In this way, we could generate infinitely long videos theoretically. Formally, the autoregressive likelihood of the whole video can be expressed as:

$$p(r_1^1, \ldots, r_K^N) = \prod_{c=1}^{N} \prod_{k=1}^{K} p(r_k^c \mid r_1^1, \ldots, r_{k-1}^c, \psi(t)), \tag{2}$$

3.3 Visual Tokenizer

Training video tokenizers faces greater challenges than training image tokenizers. First, training tokenizers on videos of tens of frames is much computationally heavier than training on static images. Therefore, training a video tokenizer from scratch is extremely time-consuming and suffers from slow convergence. Second, the scale schedule in videos leads to more imbalanced information distribution, where most information is concentrated in the last few scales. This brings great difficulties to the optimization of VAR Transformer. To solve these challenges, we introduce two techniques, knowledge inheritance from continuous video tokenizer and stochastic quantizer depth.

Knowledge Inheritance from Continuous Video Tokenizer. Instead of designing and training a discrete video tokenizer from scratch, we inherit the architecture and weights of a trained continuous video tokenizer, *i.e.* video VAE. Specifically, we first insert a parameter-free quantizer between the pre-trained VAE encoder and the decoder. The quantizer is based on binary spherical quantization [40], being similar to that of Infinity [13] but with new spacetime pyramid scale schedule. This does not introduce any new parameter like codebook in VQ [26] and well retains knowledge of the original VAE. As shown in Fig.5, the discrete video tokenizer reconstructs videos decently, even without any fine-tuning. To further improve the reconstruction quality, we fine-tune the new tokenizer jointly on images and videos like previous works [29, 1]. During the fine-tuning, the KL loss of the original VAE is replaced with the commitment loss plus the entropy penalty [40]. As shown in Fig.5, with the help of knowledge of continuous video VAE, the convergence accelerates dramatically.

Stochastic Quantizer Depth. When tokenizing videos using the spacetime pyramid schedule, the information distribution on different scales gets extremely imbalanced. Specifically, there are only a few tokens in the early scales, while there are tens of thousands of tokens in the last scales. Thus the tokenizer tends to reconstruct videos solely relying on tokens from the last few scales and not to learn useful representation in early scales as shown in Fig.6 (left). However, this imbalanced distribution is difficult to model using VAR Transformer because the dependence between the latter token blocks and the former ones is weak. To alleviate this problem, we propose a regularization called stochastic quantizer depth. During training, each one of the last N scales has a probability p of being discarded.

In this way, there are 2^N possible scale schedules during training. This requires the tokenizer to reduce the reliance on last scales and store more information in tokens of early scales. As in Fig.6 (right), with the help of this regularization, the reconstruction results of early scales become much clearer. This balanced information distribution makes the training of VAR Transformer easier.

3.4 Spacetime Autoregressive Transformer

To accommodate the newly introduced temporal dimension, enhance the quality of generated videos, and alleviate the substantial computational overhead associated with a large number of tokens, we propose the following modifications to the VAR Transformer: Spacetime RoPE, Semantic Scale Repetition, and Spacetime Sparse Attention. We put Spacetime RoPE in the appendix.

Semantic Scale Repetition. With carefully crafted positional encodings, InfinityStar can already generate videos of acceptable quality. However, we observe that the structural coherence and motion dynamics in these outputs remain suboptimal. As shown in Figure 6, the overall layout and the placement of foreground objects are determined by the early scales of the clip pyramid—what we term the "semantic scales." This observation motivates us to enhance generation fidelity at these semantic scales. To this end, we introduce a simple yet effective technique called semantic scale repetition. Concretely, if a clip pyramid comprises K scale tuples, we repeat the first K_s tuples N times, thereby reinforcing the semantic-level representations. In this way, every earlier residual r_k undergoes multiple rounds of refinement, improving the generation quality of semantics and the performance in complex scenarios with large motion. Given that the tokens at these early scales account for only a small fraction of the total token count, the additional computational overhead incurred by repeating them is negligible.

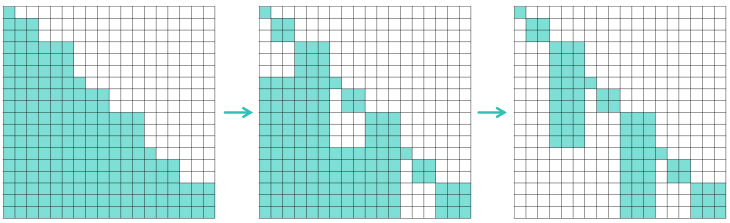


Figure 2: Illustration of three causal attention variants. We plot three pyramids on the scale size = (1,2,3) for visualization simplicity. From left to right, VAR block-wise causal mask with full history, Switti block-wise non-causal mask with full history, and spacetime sparse attention.

Spacetime Sparse Attention. Autoregressive video generation faces significant challenges due to the high computational costs of long context. As on the left of Fig.2, Infinity [13] employs a block-wise causal mask for single pyramid modeling. Switti [27] verifies that conditioning next-scale predictions solely on inputs from preceding scales is sufficient, resulting in a sparser attention mask as on the middle of Fig.2. For long video generation, it's necessary to attend history tokens to achieve temporal consistency. However, attending full history leads to an explosively long sequence. Considering each clip corresponds to 5s, which is sufficient to maintain temporal consistency, here we only attend to the last scale of the preceding clip. Finally, we obtain a highly sparse attention as show in Fig.2 (right). Our spacetime sparse attention drastically reduces attention computational overhead during both training and inference, all while delivering better performance.

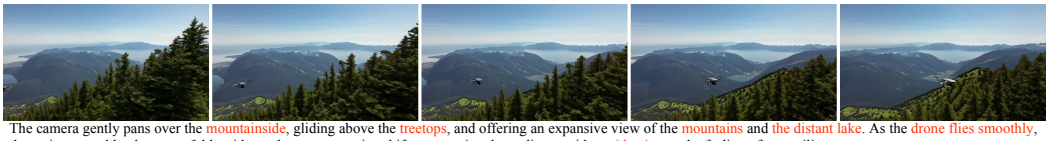
4 Experiment

4.1 Implementation

Datasets. The training data of InfinityStar includes text-to-image data and text-to-video data. We curated 130 pretraining and 70M high-quality text-to-image data. They are either from open-sourced data like COYO[4], OpenImages[19] or bought from external suppliers. To balance the data distribution and improve overall aesthetics, we also involve 5M high-quality synthetic data. In terms of text-to-video data, we curated around 16M video data. All videos are longer than 5 seconds. Among them 13M videos are under 336×192 resolution used for pre-training. They are mainly from Panda-70M[7], Mira[15], and other internal video-text pairs. Apart from those 192p videos, we also curated 3M 480p and 50K 720p high-quality videos for fine-tuning.

A woman with shoulder-length brown hair is seen talking to someone off-screen to the right. She is wearing a dark-colored top and a necklace. The background is blurred, but it appears to be an indoor setting with some indistinct objects and a window. The woman slightly moves her head while speaking.

A man wearing a green Adidas shirt and a blue cap is holding a sketchbook with both hands. He is standing outdoors in front of a building with a blue facade and some greenery. The man is flipping through the pages of the sketchbook. He points to different parts of the drawings while talking.



The camera gently pans over the mountainside, gliding above the treetops, and offering an expansive view of the mountains and the distant lake. As the drone flies smoothly, the entire natural landscape unfolds without abrupt perspective shifts, presenting the audience with a wide view and a feeling of tranquility.

Figure 3: Text to image and text to video examples.

Model and Training. After inserting the patchify and unpathify layers between WAN 2.1 VAE's encoder and decoder, we obtain a video tokenizer with a compression rate of $4 \times 16 \times 16$ and a latent dimension of 64. BSQ quantization is adopted with a vocabulary of size 2^{64}. In contrast to using a vocabulary of size 2^{64} for all scales, we use a vocabulary of size 2^{16} for the former small scales and 2^{64} for the latter large scales. We empirically find that it boosts convergence and has a negligible impact on the reconstruction quality. Starting with the pretrained weights of WAN 2.1 VAE, the discrete tokenizer is fine-tuned jointly on images of 256×256, 512×512, 768×768 and videos of $256 \times 256 \times 81$ for 30K iterations. The learning rate is $1e^{-4}$.

InfinityStar is trained progressively with four stages. The first stage is T2I pre-training. Then the pretrained weights of T2I are loaded as initialization of T2V. The latter three stages are training with videos of 192p, 360p, 720p, respectively. Each time we increase the training resolution, we preserve scale schedule of lower resolutions and append several larger scales, which enables better inheritance. The global batch size for 192p is 2048 and that of 360p and 720p is 1024. The learning rate for 192p is $2e^{-4}$. Then we decay it to $1e^{-4}$ for 360p and 720p. We train the model on videos of 192p, 360p, 720p for 50K, 8K, 3K iterations, respectively. Specifically, each clip pyramid is composed of 80 frames at 16 fps, and Semantic Scales Repetition repeats the first $K_s = 7$ scales $N = 6$ times. Details about infrastructure optimizations are presented in the appendix.

4.2 Text-to-Image Generation

The upper part of Fig.3 shows generated images from our InfinityStar-T2I model, showcasing InfinityStar's strength in generating high-fidelity and photo-realistic images across various categories and image styles. We also carry out the quantitative evaluation on the GenEval[12] and DPG[14] benchmarks. As in Tab.1, InfinityStar achieves the best overall score of 0.79 on the GenEval bench with a prompt rewriter. It's worth noting that InfinityStar exceeds Infinity by 6% on overall score. We attribute the significant improvement to the larger model size and the architectural innovations. On the DPG bench, InfinityStar reaches an overall score of 86.55, surpassing Infinity by 3.09%. These

Table 1: Evaluation on the GenEval [12] and DPG [14] benchmark. † result is with prompt rewriting.

Methods	# Params	GenEval↑				DPG↑		
		Two Obj.	Position	Color Attri.	**Overall**	Global	Relation	**Overall**
Diffusion Models								
SDXL [22]	2.6B	0.74	0.15	0.23	0.55	83.27	86.76	74.7
PixArt-Sigma [6]	0.6B	0.62	0.14	0.27	0.55	86.89	86.59	80.5
SD3 (d=38) [10]	8B	0.89	0.34	0.47	0.71	-	-	-
SANA-1.0 [31]	1.6B	-	-	-	0.66	-	-	84.8
FLUX-dev [20]	12B	-	-	-	0.67	-	-	84.0
FLUX-schnell [20]	12B	-	-	-	0.71	-	-	84.8
AutoRegressive Models								
LlamaGen [23]	0.8B	0.34	0.07	0.04	0.32			65.2
Chameleon [24]	7B	-	-	-	0.39	-	-	-
Show-o [32]	1.3B	0.80	0.31	0.50	0.68	-	-	67.5
Emu3 [30]	8B	0.81†	0.49†	0.45†	0.66†	-	-	81.6
Infinity [13]	2B	0.85†	0.49†	0.57†	0.73†	**93.11**	90.76	83.46
InfinityStar-T2I	8B	**0.90**†	**0.62**†	**0.67**†	**0.79**†	91.68	**91.87**	**86.55**

quantitative results demonstrate InfinityStar's strong capabilities of image generation following users' prompts.

4.3 Text-to-Video Generation

In the lower part of Fig.3, we present the generated videos of InfinityStar regarding user prompts. The generated videos successfully capture the semantic information in user prompts while maintaining high aesthetics and visual quality. Especially for the second example in Fig.3, the generated video accurately restores the delicate movements of the characters flipping through sketchbooks, talking while pointing to different parts of the drawings. In Tab.2, we compare InfinityStar with leading diffusion and autoregressive approaches on VBench—a comprehensive video benchmark spanning 16 evaluation dimensions. Our model achieves an overall score of 83.74, outperforming all open-source autoregressive baselines by a substantial margin. Moreover, InfinityStar surpasses diffusion-based competitors such as OpenSora[41], CogVideoX[34], and HunyuanVideo[17]. These results demonstrate that, through its novel spacetime autoregressive design, InfinityStar not only pushes the capabilities of discrete autoregressive video models but also attains performance on par with—and in some cases superior to—state-of-the-art diffusion methods.

Figure 4: Zero-shot video extrapolation examples.

Zero-shot Generation. Although trained exclusively on T2V data, InfinityStar can generate videos conditioned on an image or a video as historical without any fine-tuning. Fig.4 shows video extrapolation results. The synthesized videos exhibit strong temporal coherence with the reference while faithfully capturing the semantic nuances of texts. Zero-shot I2V samples are presented in the appendix.

4.4 Ablation Study

Visual Tokenizer. As shown in Fig.5 and Tab.3, loading weights of continuous video tokenizer significantly speeds up the convergence and achieves the best reconstruction results. As shown in Fig.6, stochastic quantizer depth largely improves the reconstruction quality of early scales. In terms of generation, using VAE with SQD leads to a notable improvement in VBench scores (81.28 *v.s.* 81.07 as shown in Tab.4). Moreover, we observe that SQD contributes to faster convergence during the video generation training.

Table 2: Evaluation on the VBench benchmark. † result is with prompt rewriting.

Models	# Params	Human Action	Scene	Multiple Objects	Appear. Style	Quality Score	Semantic Score	**Overall**
Diffusion Models								
AnimateDiff-V2	1.5B	92.60	50.19	36.88	22.42	82.90	69.75	80.27
VideoCrafter-2.0[5]	1.5B	95.00	**55.29**	40.66	**25.13**	82.20	73.42	80.44
OpenSora V1.2[41]	1.1B	85.80	42.47	58.41	23.89	80.71	73.30	79.23
Show-1[37]	6B	95.60	47.03	45.47	23.06	80.42	72.98	78.93
Gen-3 [11]	-	96.40	54.57	53.64	24.31	84.11	75.17	82.32
CogVideoX-5B[34]	5B	**99.40**	53.20	62.11	24.91	82.75	77.04	81.61
HunyuanVideo[17]	13B	94.40	53.88	68.55	19.80	85.09	75.82	83.24
Wan 2.1[28]	14B	98.80	53.67	**81.44**	21.13	**85.64**	**80.95**	**84.70**
AutoRegressive Models								
Nova[9]†	0.6B	95.20	54.06	77.52	20.92	80.39	79.05	80.12
Emu3[30]	8B	77.71	37.11	44.64	20.92	84.09	68.43	80.96
InfinityStar†	8B	96.43	52.08	78.66	21.81	84.73	79.78	83.74

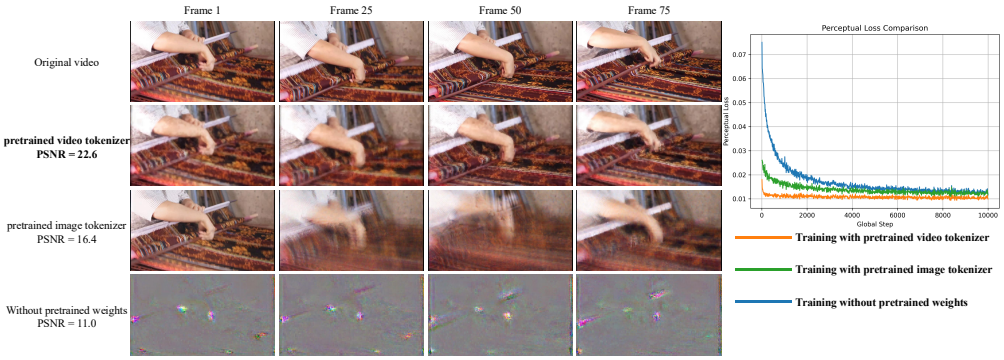


Figure 5: Influence of pretrained weights on reconstruction and convergence. The left sub-figure shows the reconstructed frames using different pretrained weights `without finetuning`. Loading weights of continuous video tokenizer achieves the best results. The right sub-figure shows that training with pretrained video tokenizer converges significantly faster than the other two strategies.

Table 3: Reconstruction metrics on an internal high-motion video benchmark (480p 81 frames).

Pretrained Weights	PSNR(↑)	SSIM(↑)	LPIPS(↓)
Continuous Video VAE	**33.37**	**0.94**	**0.065**
Image VAE	29.10	0.90	0.123
None	30.04	0.90	0.124

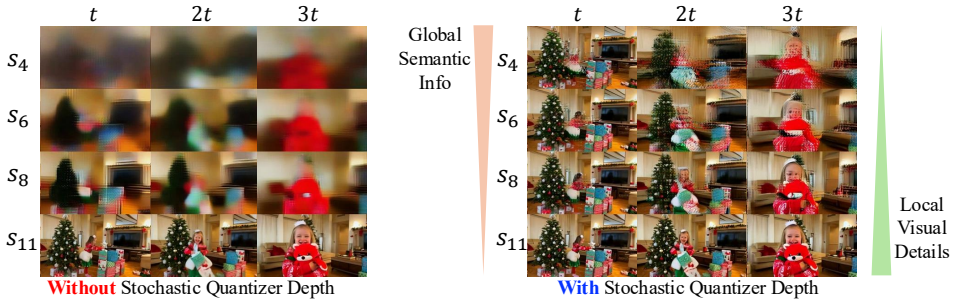


Figure 6: The influence of stochastic quantizer depth. Sub-figure (s_i, nt) represents the reconstructed frame nt using all tokens from the image pyramid plus tokens of first i scales in the clip pyramid. SQD significantly improves the reconstruction quality of early scales. Besides, the earlier scales correspond to global semantics, while the later ones are responsible for local visual details.

Pseudo-Spacetime Pyramid *v.s.* Spacetime Pyramid. As illustrated in Fig.7, videos generated by the pseudo-spacetime pyramid lack visual details and deliver simpler motion. In contrast, spacetime pyramid generates videos with richer details and higher motion. Besides, spacetime pyramid improves VBench's overall score from 80.30 to 81.28 as illustrated in Tab.4. These experiments support the

Figure 7: Comparison between Pseudo-Spacetime Pyramid and Spacetime Pyramid. Spacetime Pyramid could generate videos with richer details and higher motion.

hypothesis that spacetime pyramid could decouple appearance and temporal information. The image pyramid corresponds to the appearance information and clip pyramids focus on subsequent motions. This decoupling makes it easier to learn video motions. In addition to advances in performance, spacetime pyramid could unify T2I, T2V, I2V tasks into one framework.

Semantic Scale Repetition. In Fig.6, we can observe that the earlier scales correspond to semantic information, while the later ones are responsible for high-frequency details. Here we compare the generation results with and without semantic scale repetition. As shown in Fig.8, semantic scale repetition is highly effective in improving the structure stability and motion quality. The quantitative results further confirm the significant gains. As shown in Tab.4, semantic scale repetition improves VBench's overall score from 75.72 to 81.28.

Spacetime Sparse Attention. In Tab.4 and Tab.5, we compare different attention mechanisms. Spacetime sparse attention shows superior performance to full attention in the Vbench total score (81.28 *v.s.* 80.77), while showing a significant advantage in saving computation and GPU VRAM. SSA reaches 1.5× speedup when generating 192p 161 frames. The efficiency advantage becomes larger as the resolution and duration grow. For 480p 161 frames, full attention fails due to OOM while SSA completes it within 44.7s using 63GB VRAM. We hypothesize that SSA produces better results than full attention because it reduces exposure bias. Full attention is more susceptible to accumulated errors. The reason we do not condition on smaller scales of the preceding clip is that it misses the former clips' visual details and brings visual inconsistency between clips. Although it reaches 1.1× , 1.5× speedup for 192p and 480p 161 frames, we observe a significant performance drop in Vbench from 81.28 to 80.75 as shown in Tab.4. Therefore, the proposed spacetime sparse attention strikes a better balance between computational efficiency and visual quality.

4.5 Inferency Latency

As shown in Tab.6, we report the end-to-end inference latency measured on a single GPU, including both the text encoder and VAE decoder. Wan-2.1[28] and Nova[9] were evaluated using their default GitHub configurations. Even without employing stronger compression, InfinityStar achieves a 32× speedup over Wan-2.1. Furthermore, despite its larger model size, InfinityStar delivers a 6× speedup compared to Nova. These results highlight our model's significant efficiency advantage over both diffusion and autoregressive approaches.

5 Conclusion and Limitation

We introduce InfinityStar, a unified spacetime autoregressive framework capable of synthesizing high-resolution images and dynamic, high-motion videos. By seamlessly integrating spatial and

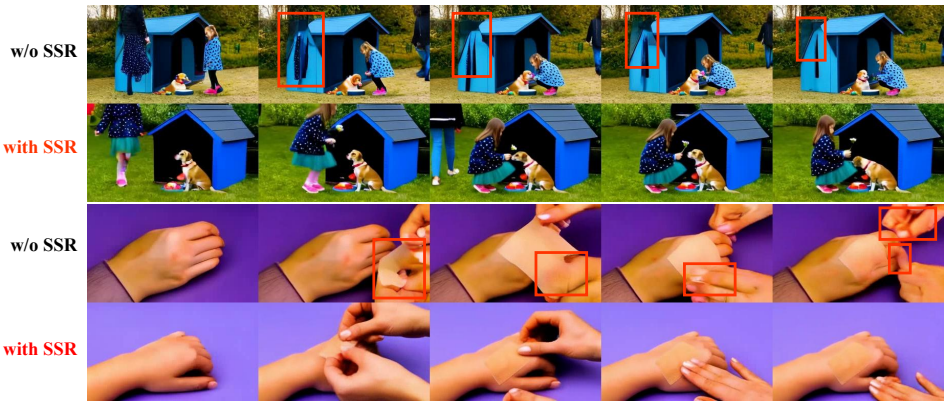


Figure 8: Semantic Scale Repetition (SSR) greatly improves structure stability and motion quality.

Table 4: Comprehensive ablation studies. Experiment with 1M 192p training data, $batch\_size = 40$, and 30K iterations. We evaluate the results on the Vbench benchmark.

Vbench	total score	quality score	semantic score
InfinityStar (Our Model)	**81.28**	**81.56**	80.16
Attend to former clip's largest scale			
Ablation by removing/replacing core components			
w/o Semantic Scale Repetition(SSR)	75.72	76.73	71.68
w/o Spacetime Pyramid (using Pseudo-Spacetime)	80.30	80.81	78.28
w/o Stochastic Quantizer Depth(SQD)	81.07	81.21	**80.54**
Comparison of different Attention Mechanism variants			
Full Attention	80.77	81.15	79.23
Attend to former clip's 3rd largest scale	80.86	81.26	79.26
Attend to former clip's 6th largest scale	80.75	80.98	79.80

Table 5: Computational efficiency comparison of attention mechanisms on a single GPU.

	(192p 65 frames)	(192p 161 frames)	(480p 161 frames)
Full Attention	8.6s / 40.8GB	24.3s / 57GB	OOM
Attend to former clip's largest scale	7.7s / 38.5GB	16.7s / 40GB	44.7s / 63 GB
Attend to former clip's 3rd largest scale	7.4s / 38.2GB	15.8s / 39GB	34.5s / 58 GB
Attend to former clip's 6th largest scale	7.3s / 37.9GB	15.2s / 38GB	30.5s / 55GB

Table 6: Computational efficiency comparison.

Method	Model	# Parameters	Durations(s)	Frames	Resolution	Time(s)	Speedup
Diffusion	Wan 2.1[28]	14B	5	81	720p	1864	1
AR	Nova[9]	0.6B	5	81	480p	354	5
AR	InfinityStar	8B	5	81	720p	58	32

temporal prediction within a purely discrete architecture, InfinityStar supports diverse generation tasks while maintaining both state-of-the-art quality and exceptional efficiency. Our extensive evaluation demonstrates that InfinityStar outperforms prior autoregressive video models and rivals leading diffusion-based approaches, producing 5s of 720p video in one-tenth the inference time. As the first discrete autoregressive model to deliver industrial-grade 720p video synthesis, we anticipate that InfinityStar will catalyze future research on rapid, long video generation.

However, due to limited computational resources, we did not scale our model training and parameter size to the level of leading diffusion models, which bottlenecks current performance. Furthermore, our inference pipeline has not yet been fully optimized, indicating room for future improvement.

References

[1] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 2, 3, 4

[2] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[3] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh. Video generation models as world simulators. *OpenAI*, 2024. 1, 2

[4] M. Byeon, B. Park, H. Kim, S. Lee, W. Baek, and S. Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 5

[5] H. Chen, M. Xia, Y. He, Y. Zhang, X. Cun, S. Yang, J. Xing, Y. Liu, Q. Chen, X. Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 2, 8

[6] J. Chen, C. Ge, E. Xie, Y. Wu, L. Yao, X. Ren, Z. Wang, P. Luo, H. Lu, and Z. Li. Pixart-sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 7

[7] T.-S. Chen, A. Siarohin, W. Menapace, E. Deyneka, H.-w. Chao, B. E. Jeon, Y. Fang, H.-Y. Lee, J. Ren, M.-H. Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024. 5

[8] G. DeepMind. Veo 3. *https://deepmind.google/technologies/veo/veo-3/*, 2025.05. 1

[9] H. Deng, T. Pan, H. Diao, Z. Luo, Y. Cui, H. Lu, S. Shan, Y. Qi, and X. Wang. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*, 2024. 1, 2, 8, 9, 10

[10] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 7

[11] A. Germanidis. Introducing gen-3 alpha: A new frontier for video generation. *Runway*, 2024. 8

[12] D. Ghosh, H. Hajishirzi, and L. Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36, 2024. 6, 7

[13] J. Han, J. Liu, Y. Jiang, B. Yan, Y. Zhang, Z. Yuan, B. Peng, and X. Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*, 2024. 1, 2, 3, 4, 5, 7

[14] X. Hu, R. Wang, Y. Fang, B. Fu, P. Cheng, and G. Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024. 6, 7

[15] X. Ju, Y. Gao, Z. Zhang, Z. Yuan, X. Wang, A. Zeng, Y. Xiong, Q. Xu, and Y. Shan. Miradata: A large-scale video dataset with long durations and structured captions, 2024. 5

[16] D. Kondratyuk, L. Yu, X. Gu, J. Lezama, J. Huang, G. Schindler, R. Hornung, V. Birodkar, J. Yan, M.-C. Chiu, K. Somandepalli, H. Akbari, Y. Alon, Y. Cheng, J. Dillon, A. Gupta, M. Hahn, A. Hauth, D. Hendon, A. Martinez, D. Minnen, M. Sirotenko, K. Sohn, X. Yang, H. Adam, M.-H. Yang, I. Essa, H. Wang, D. A. Ross, B. Seybold, and L. Jiang. Videopoet: A large language model for zero-shot video generation, 2024. 1, 2

[17] W. Kong, Q. Tian, Z. Zhang, R. Min, Z. Dai, J. Zhou, J. Xiong, X. Li, B. Wu, J. Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 1, 2, 7, 8

[18] Kuaishou. Kling ai. *https://klingai.kuaishou.com/*, 2024.06. 1

[19] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020. 5

[20] B. F. Labs. Flux. https://blackforestlabs.ai/announcing-black-forest-labs/, 2024. 7

[21] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2

[22] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 7

[23] P. Sun, Y. Jiang, S. Chen, S. Zhang, B. Peng, P. Luo, and Z. Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 7

[24] C. Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 7

[25] K. Tian, Y. Jiang, Z. Yuan, B. Peng, and L. Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024. 1, 2

[26] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 4

[27] A. Voronov, D. Kuznedelev, M. Khoroshikh, V. Khrulkov, and D. Baranchuk. Switti: Designing scale-wise transformers for text-to-image synthesis. *arXiv preprint arXiv:2412.01819*, 2024. 5

[28] A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 2, 8, 9, 10

[29] J. Wang, Y. Jiang, Z. Yuan, B. Peng, Z. Wu, and Y.-G. Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *Advances in Neural Information Processing Systems*, 37:28281–28295, 2024. 2, 3, 4

[30] X. Wang, X. Zhang, Z. Luo, Q. Sun, Y. Cui, J. Wang, F. Zhang, Y. Wang, Z. Li, Q. Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 1, 2, 7, 8

[31] E. Xie, J. Chen, J. Chen, H. Cai, H. Tang, Y. Lin, Z. Zhang, M. Li, L. Zhu, Y. Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 7

[32] J. Xie, W. Mao, Z. Bai, D. J. Zhang, W. Wang, K. Q. Lin, Y. Gu, Z. Chen, Z. Yang, and M. Z. Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 7

[33] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 2

[34] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2, 7, 8

[35] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, A. Gupta, X. Gu, A. G. Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 2, 3

[36] Y. Zeng, G. Wei, J. Zheng, J. Zou, Y. Wei, Y. Zhang, and H. Li. Make pixels dance: High-dynamic video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8850–8860, 2024. 2

[37] D. J. Zhang, J. Z. Wu, J.-W. Liu, R. Zhao, L. Ran, Y. Gu, D. Gao, and M. Z. Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision*, pages 1–15, 2024. 8

[38] F. Zhang, S. Tian, Z. Huang, Y. Qiao, and Z. Liu. Evaluation agent: Efficient and promptable evaluation framework for visual generative models. *arXiv preprint arXiv:2412.09645*, 2024. 2

[39] Y. Zhang, H. Yang, Y. Zhang, Y. Hu, F. Zhu, C. Lin, X. Mei, Y. Jiang, B. Peng, and Z. Yuan. Waver: Wave your way to lifelike video generation. *arXiv preprint arXiv:2508.15761*, 2025. 1

[40] Y. Zhao, Y. Xiong, and P. Krähenbühl. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*, 2024. 4

[41] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024. 7, 8

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: This paper claims the contributions that we are the first discrete autoregressive model to achieve high-quality 720p video generation with the fastest generation speed in the abstract and introduction

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our limitations in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our proof, such as the autoregressive likelihood formula, has been validated and verified through experimental comparisons, demonstrating the effectiveness of the proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We will open-source our model and code, and use publicly available data to guide people on how to reproduce our results, as training large-scale video generation models is extremely resource-intensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Currently, the code needs to be cleaned, and the data needs to be maintained as an open-source subset to ensure the community can follow along. We will soon open-source the code along with documentation for some of the data sources.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have outlined the multi-stages of our training, including the training steps and relevant training steps, and learning rate for each stage.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For both our benchmark and ablation study evaluations, we run multiple trials and report the average results to ensure the statistical significance of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We use internal clusters to train our models. In terms of estimation, the four-stage video generation process consumes 5,000, 40,000, 30,000, and 30,000 GPU hours respectively. The video VAE requires 2,000 GPU hours. The ablation study has a total cost of 10,000 GPU hours, and the evaluation consumes 1,000 GPU hours.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have clear filters in place for privacy, inappropriate and harmful content for both image and video data.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Visual generation has a broad impact on society. On one hand, video generation models significantly improve production efficiency. On the other hand, we closely monitor the potential societal risks related to the misuse of information by models trained on the data we use. We have filtered the harmful and copyrighted data in our training dataset

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: First, our training data will undergo strict filtering, and we will ensure rigorous verification before open-sourcing to check for any obvious issues like harmful visual content. After open-sourcing, we will provide a model that includes a default safeguards model to filter any unsuitable generated content.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In the paper, we specify the sources of the open-source data used and provide proper citations. Regarding the code, we will acknowledge the relevant contributors on our GitHub repository when we make it publicly available.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: In the appendix, we will provide more visual samples with details. All these materials will be anonymized. The code is our proprietary property, and the data collection is compliant.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We use LLM for writing and formatting

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.