
InfinityStar: Unified Spacetime AutoRegressive Modeling for Visual Generation

1 Spacetime Autoregressive Modeling

Spacetime RoPE. We introduce spacetime rotary position embeddings (Spacetime RoPE) tailored for InfinityStar. This is achieved by decomposing original rotary embeddings[4] into four components: scale, time, height, and width. As shown in Fig. 1, the scale ID serves as a counter of scales up to now. The temporal ID remains zero for tokens in the image pyramid. For tokens in video pyramids, it increments as the frame grows. Distinct IDs are assigned to height and width components based on the token’s position in the image or video. Spacetime RoPE enhances the modeling of complex positional information for tokens hidden in image and video pyramids.

Spacetime Autoregressive Transformer with Bitwise Self-Correction. To alleviate the train-test discrepancies of teacher-forcing training, we adopt bitwise self-correction mechanism proposed by Infinity[3]. Specifically, during training, some of the input tokens are randomly flipped to simulate the prediction error in the inference phase. Besides, the target labels are also recomputed to match the perturbed inputs. Moreover, when predicting the token distribution, the traditional index-wise classifier is replaced by a bitwise classifier. The bitwise classifier predicts d bits instead of 2^d indices, significantly reducing the memory costs and optimization difficulties. Algorithm 1 shows the detailed procedure of Spacetime Pyramid with Bitwise Self-Correction.

Algorithm 1 Spacetime Pyramid Encoding with BSC

Require: raw feature \mathbf{F} , scale schedule number K , clip number N
 image pyramid scale schedule: $(1, h_1, w_1), \dots, (1, h_K, w_K)$,
 clip pyramid scale schedule: $(T, h_1, w_1), \dots, (T, h_K, w_K)$

```

 $\mathbf{R}_{queue} \leftarrow []$   $\triangleright$  multi-scale bit labels
 $\tilde{\mathbf{F}}_{queue} \leftarrow []$   $\triangleright$  inputs for transformer
for  $c = 1, 2, \dots, N$  do  $\triangleright$  iter-clips iteration
   $t_{start} = 1 + (c - 1) * T$ 
   $\mathbf{F}_c \leftarrow$  raw features from time  $t_{start}$  to  $t_{start} + t_c$ 
  for  $k = 1, 2, \dots, K$  do  $\triangleright$  intra-clip multi-scale iteration
     $\mathbf{R}_k = \text{quant}(\text{down}(\mathbf{F}_c - \mathbf{F}_{c,k-1}^{flip}, (t_k, h_k, w_k)))$ 
    Queue_Push( $\mathbf{R}_{queue}, \mathbf{R}_k$ )
     $\mathbf{R}_k^{flip} = \text{Random\_Flip}(\mathbf{R}_k, p)$ 
     $\mathbf{F}_{c,k}^{flip} = \sum_{i=1}^k \text{up}(\mathbf{R}_i^{flip}, (h, w))$ 
     $\tilde{\mathbf{F}}_{c,k} = \text{down}(\mathbf{F}_{c,k}^{flip}, (t_{k+1}, h_{k+1}, w_{k+1}))$ 
    Queue_Push( $\tilde{\mathbf{F}}_{queue}, \tilde{\mathbf{F}}_{c,k}$ )
  end for
end for
Ensure:  $\mathbf{R}_{queue}, \tilde{\mathbf{F}}_{queue}$ 

```

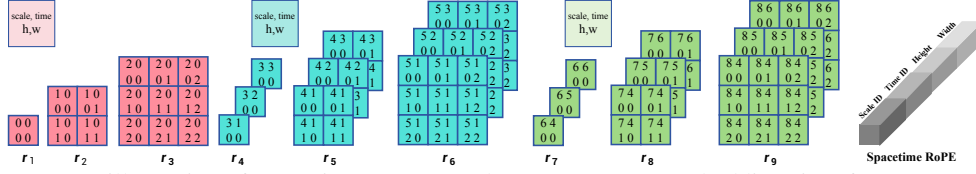


Figure 1: An illustration of Spacetime RoPE. We decompose rotary embeddings into four components, *i.e.*, scale, time, height, and width components. Spacetime RoPE enhances the modeling of complex positional information while supporting extrapolation.

	Filtered Quantity	Source	Caption	Filter
Pretrain Data	130M	Realistic Data <ul style="list-style-type: none"> • COYO • Object365 • OpenImages 	Long Caption: <ul style="list-style-type: none"> • InternVL2 • max length=512 Short Caption: <ul style="list-style-type: none"> • Blip2 	watermark < 0.8 image size > 320 Aesthetic > 4.5
High-quality Data	70M	Realistic Data <ul style="list-style-type: none"> • filtered from pretrained data • buy from suppliers Synthetic Data (5M, 7%) <ul style="list-style-type: none"> • download from internet text rendering data (50%), artistic images (35%), general data (15%) 	Long Caption: <ul style="list-style-type: none"> • InternVL2 • max length=512 Short Caption: <ul style="list-style-type: none"> • Blip2 • Human Annotate 	Round1 <ul style="list-style-type: none"> • aesthetic Score • low quality Score • white background • 56.7% left Round2 <ul style="list-style-type: none"> • grouped figure • watermark • 98.6% left

Figure 2: image data curation pipeline.

	Filtered Quantity	Source	Caption	Filter
Pretrain Data	13M	Realistic Data <ul style="list-style-type: none"> • Panda70M • Mira • InternVid • Pexels 	Caption: <ul style="list-style-type: none"> • Tarsier 2 • max length=512 	OCR <= 0.02 shorter edge > 480 Aesthetic > 4.3 duration ≥ 5s motion score > 0.3 fps ≥ 16
High-quality Data	3M	Realistic Data <ul style="list-style-type: none"> • buy from suppliers • filtered from pretrained data 	Caption: <ul style="list-style-type: none"> • Tarsier 2 • max length=512 	OCR <= 0.01 shorter edge > 720 Aesthetic > 4.5 duration ≥ 5s motion score > 0.5 fps ≥ 16

Figure 3: video data curation pipeline.

2 Infrastructure and Data

Infrastructure Optimization. Compared to diffusion models, visual autoregressive methods possess around 2.5× longer training sequences. This feature poses crucial pressure on hardware and algorithms when scaling models and increasing resolutions. In this work, we adopt advanced parallelism methods for scalable and efficient training.

Firstly, we utilize FlexAttention to implement various attention mechanisms. With our proposed Spacetime Sparse Attention, we achieve more than a 2× acceleration in training speed. Secondly, we adopt fully sharded data parallelism (FSDP) [6] to partition parameters, gradients, and optimizer states across GPU ranks. Thirdly, we adopt a fine-grained activation checkpointing strategy to reduce the overhead of vRAM and data transfer, making the parallelization more efficient. Last but not least, sequence parallelism further partitions long sequences into multiple chunks and then exploits ring self-attention for each chunk, making it feasible to train 720p videos with 200K sequence length.

Visual Captioning. Detailed visual captioning is crucial for enabling the model to accurately generate images and videos. For images, we use InternVL2.0[2] to produce dense descriptions for each sample. For video clips, we obtain overall video descriptions using Tarsier2[5]. Notably, Tarsier2 inherently captures camera motion types (e.g., zoom, pan right), eliminating the need for a separate prediction model. This simplifies the pipeline and improves efficiency.

Data Pipeline. Obtaining a high-quality image and video dataset requires a complex processing pipeline. In Fig.2 and Fig.3, we outline the data sources used, as well as the filters and thresholds applied during image and video preprocessing. Specifically for video, we follow video processing

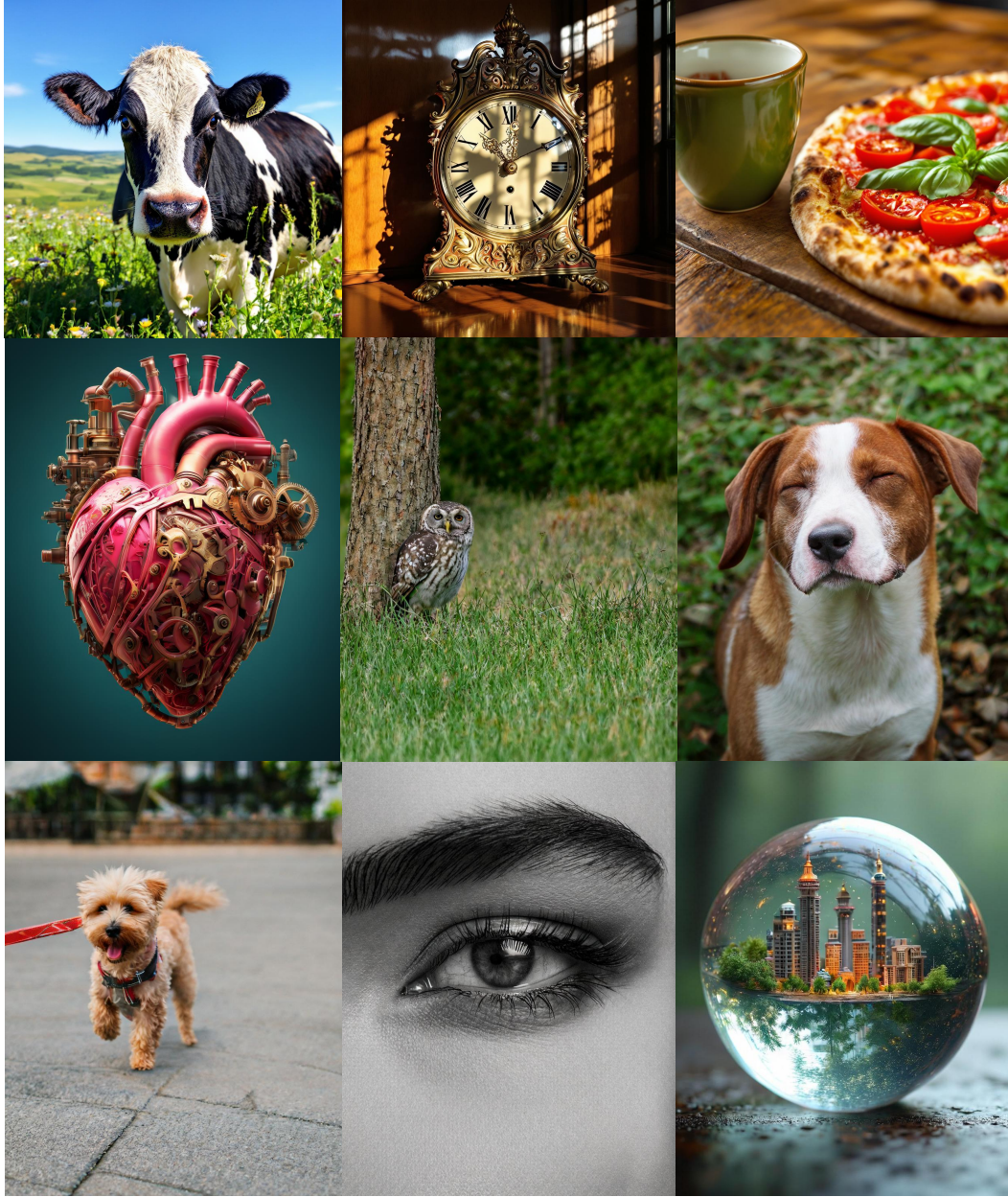


Figure 4: Text to image examples.

pipelines[1] to preprocess videos into high-quality training clips through OCR filtering, video clip extraction, visual aesthetic filtering, and motion filtering, etc.

3 More Qualitative Results

We have uploaded additional video examples in MP4 format, showcasing videos generated by our model. You can view these examples by opening the *[index.html](#)* file in your web browser.

3.1 Text-to-Image Generation.

Fig.4 shows more generated images from our InfinityStar-T2I model. Our model is capable of generating high-resolution images filled with vivid and intricate details.

3.2 Text-to-Video Generation.

We show 5-second 720p videos generated by our models in *index.html*. These cases include fast motion activities such as horse riding, running, and motorcycling, as well as close-up shots and dynamic movements of both people and animals, along with natural landscapes. Our model is capable of generating high-resolution, dynamic videos that meet industrial-grade quality standards.

3.3 Zero-shot Generation

Image to Video. Although trained exclusively on text-to-video data, InfinityStar can generate videos conditioned on an input image without any fine-tuning. Fig.5 illustrates qualitative results on the image-to-video task. The synthesized videos exhibit strong temporal coherence with the reference image—a critical requirement for this task—while faithfully capturing the semantic nuances of the accompanying text with high visual fidelity. Please refer to *index.html* for more I2V samples.



Figure 5: Zero-shot image to video examples. InfinityStar can generate videos following an input image without fine-tuning. The synthesized videos exhibit strong temporal and semantic coherence.

Video Extrapolation. Analogous to I2V, InfinityStar can naturally extrapolate videos by feeding the reference sequence as historical image-clips. Thanks to our spacetime RoPE, it is able to generate outputs that exceed the lengths seen during training. As mentioned above, more sample videos can be found in *index.html*.

3.4 Video Reconstruction.

Figure 6 illustrates a comparison between the reconstructed videos generated by different tokenizers and the original video. The discrete tokenizer trained from scratch (middle row) exhibits inferior reconstruction quality. In contrast, the tokenizer incorporating knowledge inheritance (top row) demonstrates a substantial improvement in visual fidelity, particularly in the preservation of intricate details such as human faces and complex textures.



Figure 6: Comparison between the reconstruction quality of different video tokenizers. The tokenizer incorporating knowledge inheritance (top row) demonstrates a substantial improvement compared to one trained from scratch (middle row).

References

- [1] S. Chen, C. Ge, Y. Zhang, Y. Zhang, F. Zhu, H. Yang, H. Hao, H. Wu, Z. Lai, Y. Hu, et al. Goku: Flow based video generative foundation models. *arXiv preprint arXiv:2502.04896*, 2025. [3](#)
- [2] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, Z. Muyan, Q. Zhang, X. Zhu, L. Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. [2](#)
- [3] 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](#)
- [4] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. [1](#)
- [5] L. Yuan, J. Wang, H. Sun, Y. Zhang, and Y. Lin. Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding. *arXiv preprint arXiv:2501.07888*, 2025. [2](#)
- [6] Y. Zhao, A. Gu, R. Varma, L. Luo, C.-C. Huang, M. Xu, L. Wright, H. Shojanazeri, M. Ott, S. Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023. [2](#)