

Real-Time Video Generation with **Pyramid Attention Broadcast**

Outline

- **Background:** Why is video DiT **so slow**?
- **Insight:** Observe **redundancy** in attention!
- **Method:** Pyramid Attention Broadcast with **algorithm-system co-design for real-time video generation!**
- **Evaluation:** Speedups and qualitative and quantitative results.
- **Conclusion:** Summary and future works.

Background

Why video DiT are so slow?

- Model size is small: 720M, 1.3B
- Token num is **much much larger**: A video of 10s 720p, token num $\sim 1\text{M}$

Time for generating a 10s 720p video: **6 min**

However, 10s video is just the beginning. What about video of 1min or even 1 hour?

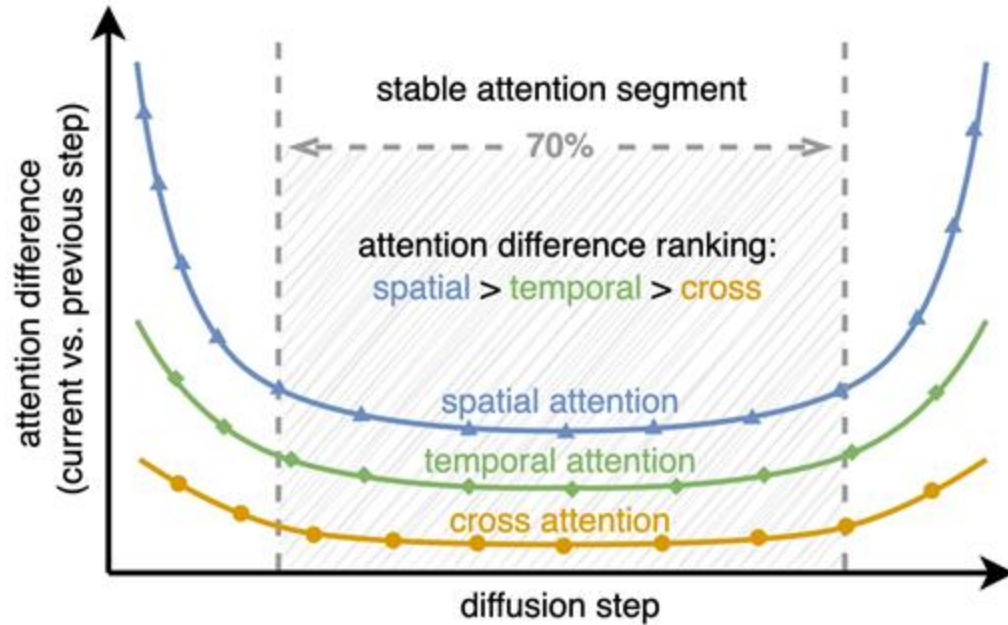
But speedup really matters to video applications. We need speed up!

Insight

But how to speed up?

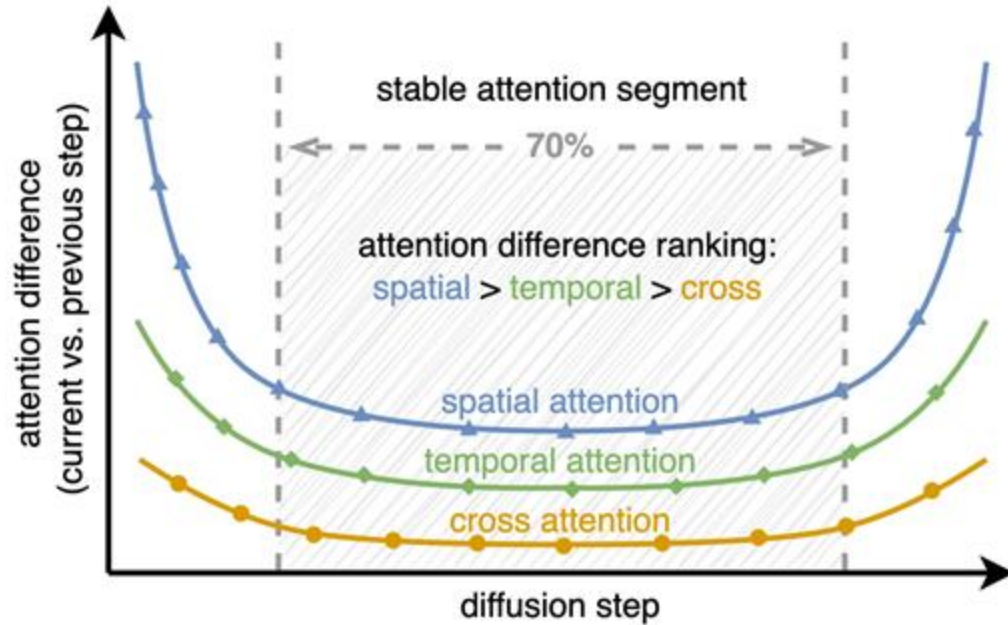
We have two key observation in video DiTs!

Insight



1. Attention differences across time steps exhibit a U-shaped pattern, with **the middle 70% of steps are very stable** with minor differences.

Insight



2. Within the stable middle segment, the differences varies among attention types: **spatial > temporal > cross**.

Insight

But how to speed up?

We have two key observation in video DiTs!

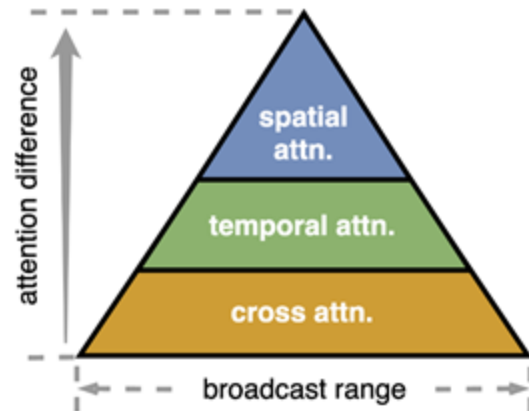
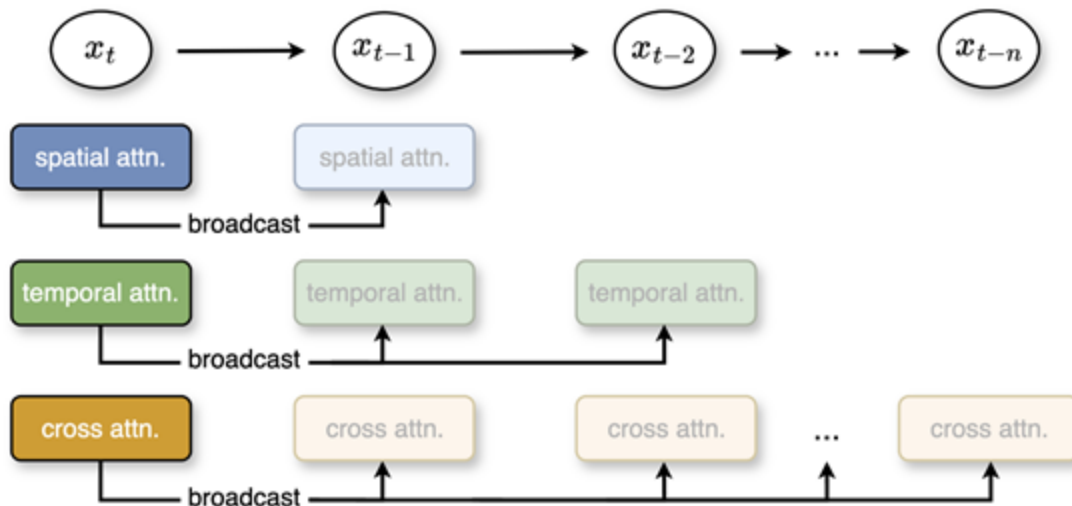
1. Attention differences across time steps exhibit a U-shaped pattern, with **the middle 70% of steps are very stable** with minor differences.
2. Within the stable middle segment, the differences varies among attention types: **spatial > temporal > cross.**

Problem: how to take advantage of these observations?

Method

Since there are redundancy in attention, we can reuse outputs by **broadcasting attention outputs to subsequent several steps!**

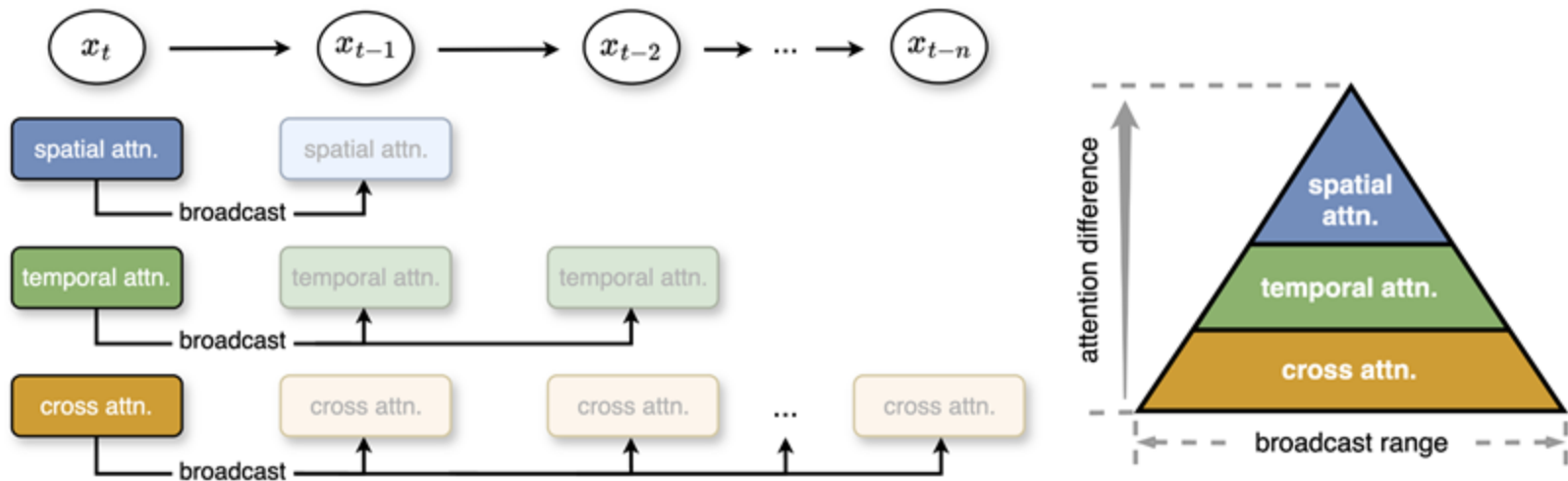
The broadcasted steps can save all computation of attention!



Method

But the difference of each attention varies a lot! How to achieve best efficiency while keeping generation quality?

Set different broadcast range for each attention! The more attention changes, the shorter its broadcast range is.



Method

What to broadcast: attention outputs or attention scores?

- **Attention outputs is better choice!**
- Both outputs and scores are similar:
 - Attn scores are similar because no much change for attn. And it is a common choice.
 - But why attention outputs are similar? Although pixel-level content has changed, their attention aggregated results are still similar!
- And attention output broadcast is way more efficient
 - Can skip qkvo project, pos emb, layernorm...

Method

To achieve real-time generation, attention broadcast is not enough.

We need to introduce sequence parallel.

Sequence parallel means splitting sequence over multiple GPUs to decrease workload on each GPU and reduce latency.

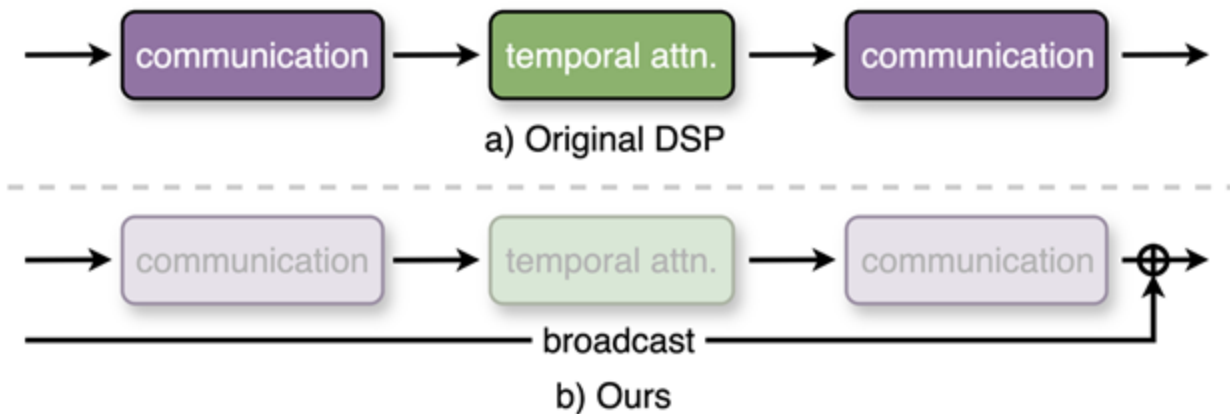
But current sequence parallel will introduce **a lot of extra communication time**, especially when scaling to multiple GPUs.

Method

Sequence parallel need to have communication for temporal attention block.

But if temporal attention is broadcasted, **both computation and communication can be eliminated!**

It brings **super scaling ability** when extending to multiple GPUs!



Evaluation

w/o PAB



Avg. FPS: 2.0

Progress: 0/5

00:00

w/ PAB



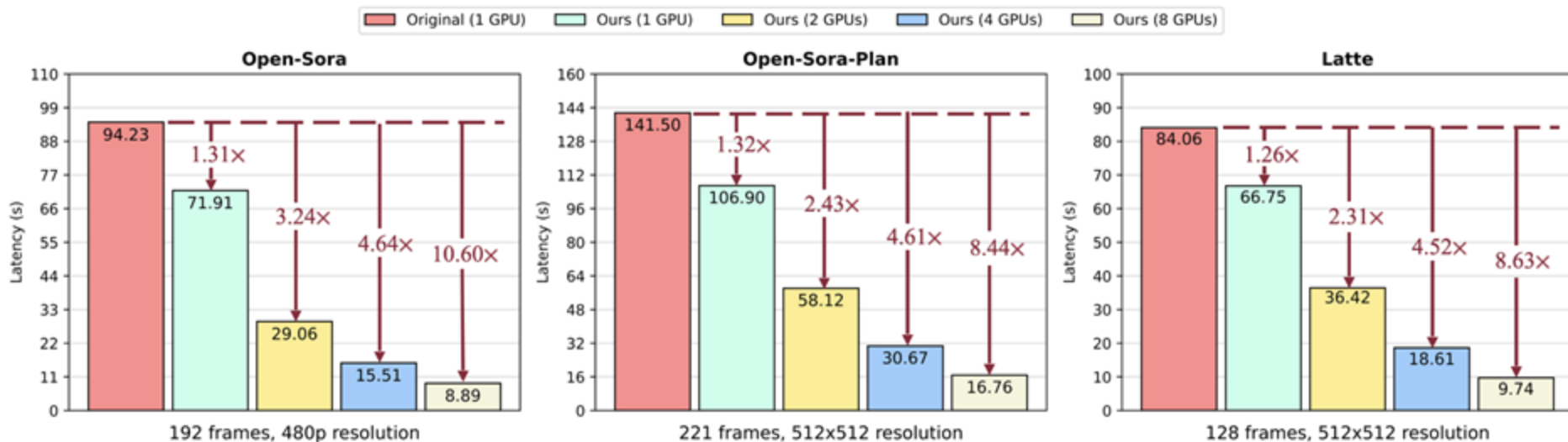
Avg. FPS: 20.2

Progress: 0/5

```
(cai) → OpenDiT git:(eval) x bash scripts/opensora/sample.sh
[06/26/24 04:40:01] INFO    colossalai - colossalai - INFO:
                          /data/xuanlei/miniconda3/envs/cai/lib/python3.9/site-packages/colossalai/initialize.py:67
                          launch
                          INFO    colossalai - colossalai - INFO:
                          Distributed environment is initialized, world size: 1
[2024-06-26 04:40:19] Prompt: time lapse of the rising sun over a tree in an open rural landscape, with clouds in the blue sky beautifully playing with the rays of light
100%|████████████████████| 30/30 [00:46<00:00, 1.57s/it]
[2024-06-26 04:41:07] Prompt: snow falling over multiple houses and trees on winter landscape against night sky. christmas as festivity and celebration concept.
0%|████████████████████| 0/30 [00:00<?, ?it/s]
```

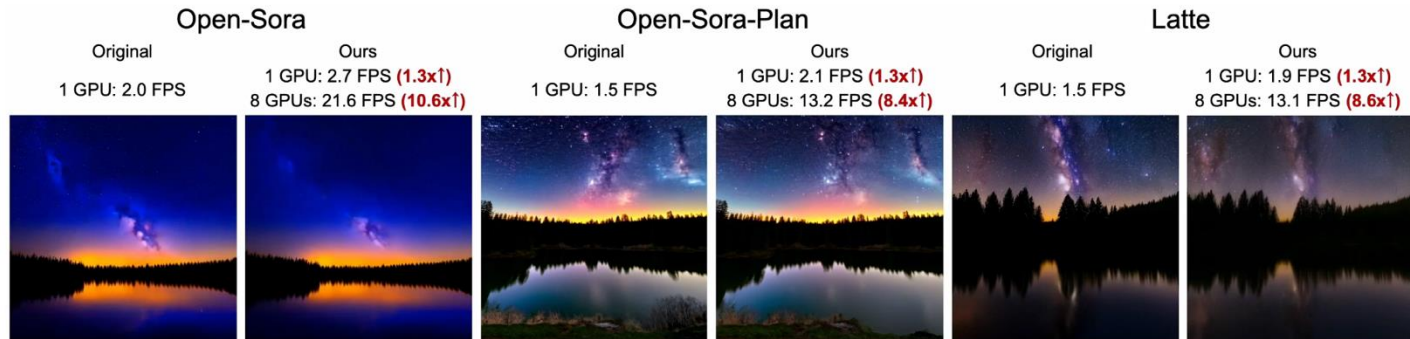
```
(cai) → OpenDiT git:(eval) x bash scripts/opensora/sample_pab.sh
[06/26/24 04:16:53] INFO    colossalai - colossalai - INFO:
                          /data/xuanlei/miniconda3/envs/cai/lib/python3.9/site-packages/colossalai/initialize.py:67
                          launch
                          INFO    colossalai - colossalai - INFO:
                          Distributed environment is initialized, world size: 8
Init Pyramid Attention Broadcast
[2024-06-26 04:17:13] Prompt: time lapse of the rising sun over a tree in an open rural landscape, with clouds in the blue sky beautifully playing with the rays of light
100%|████████████████████| 30/30 [00:21<00:00, 1.42it/s]
[2024-06-26 04:17:35] Prompt: snow falling over multiple houses and trees on winter landscape against night sky. christmas as festivity and celebration concept.
0%|████████████████████| 0/30 [00:00<?, ?it/s]
```

Evaluation



- For 1 GPU, 1.26 to 1.32x speedup.
- For 8 GPU, 8.44 to 10.60x speedup.
- Stable speedup for different models and noise schedulers!

Evaluation



Prompt: A serene night scene in a forested area. The first frame shows a tranquil lake reflecting the star-filled sky above. The second frame reveals a beautiful sunset, casting a warm glow over the landscape. The third frame showcases the night sky, filled with stars and a vibrant Milky Way galaxy. The video is a time-lapse, capturing the transition from day to night, with the lake and forest serving as a constant backdrop. The style of the video is naturalistic, emphasizing the beauty of the night sky and the peacefulness of the forest.



Prompt: White smoke on black background. simply drop it in and change its blending mode to screen or add..



Prompt: Summer landscape on a mountain lake. small rustic wooden pier on the water waves. morning and sunlight through the clouds waves, in the background of the mountain in the fog.

Evaluation



Prompt: Korean popular dish, samgyopsal, is being baked on a stone plate with kimchi. close-up, macro shot.



Prompt: Slow pan upward of blazing oak fire in an indoor fireplace.



Prompt: Snow falling over multiple houses and trees on winter landscape against night sky. christmas festivity and celebration concept.

Evaluation

Model	LPIPS (↓)	SSIM (↑)
Open-Sora	0.1740	0.8031
Open-Sora-Plan	0.2161	0.7102
Latte	0.3017	0.6427

Evaluation

Limitation:

1. More memory usage because of cache
2. Less speedup for very high quality generation (35%→~20%)



Conclusion

Key insight: attention redundancy

Pyramid Attention Broadcast:

- Broadcast attention outputs to alleviate redundancy.
- Set different broadcast strategies for each attention for best efficiency.

Sequence Parallel: super scaling ability combined with broadcast

Speedup: up to 21.6 FPS with 8 gpus with 10.6x acceleration for 480p, ~10 FPS for 720p

Conclusion

Future works:

- More fine-grained broadcast strategy
- Explore redundancy for mlp and diffusion steps
- Combined with distillation models

Thanks!