ICML
International Conference
On Machine Learning

FlashBlock: Attention Caching for Efficient Long-Context Block Diffusion

Zhuokun Chen[1] Jianfei Cai[1] Bohan Zhuang[2]

[1] Monash University [2] Zhejiang University

Limitations of Autoregressive Long Context Generation

Autoregressive diffusion models combine autoregressive modeling with diffusion-based generation for long sequence generation.

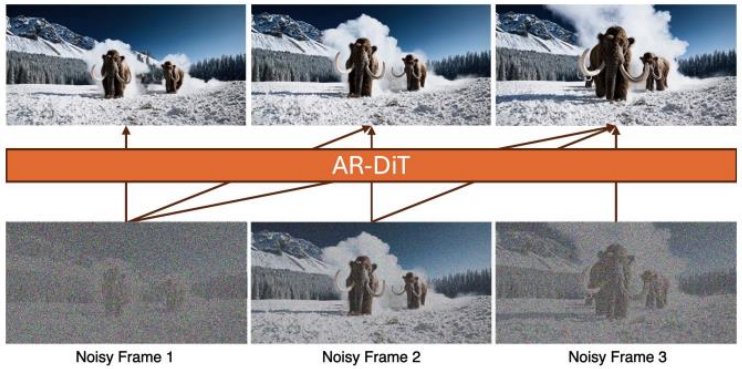
Given a sequence of blocks:

$$x_{1:N} = (x_1, x_2, \ldots, x_N)$$

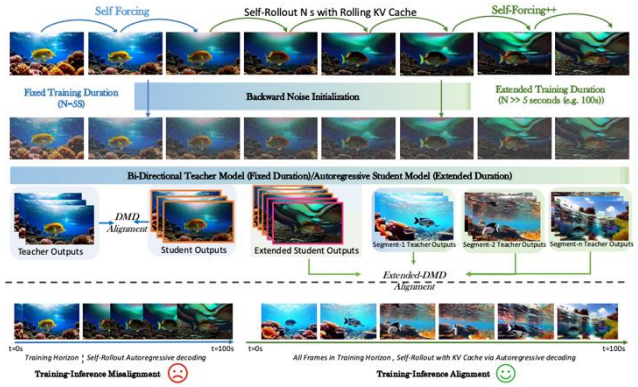
The generation follows the chain rule:

$$p(x_{1:N}) = \prod_{i=1}^{N} p(x_i \mid x_{<i})$$

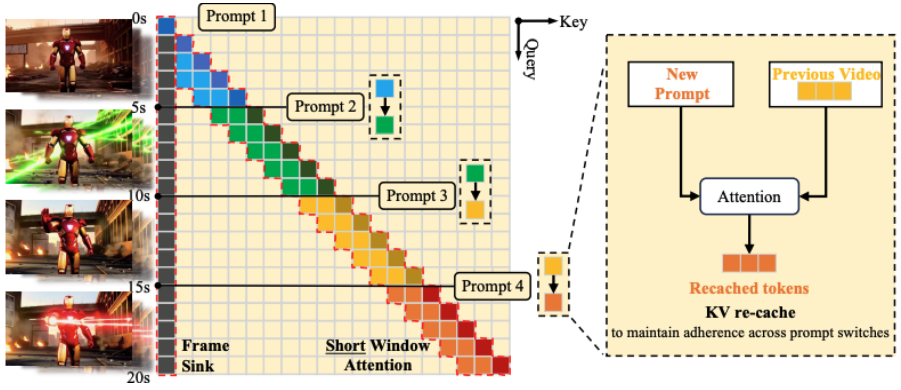
Each block distribution is modeled using a diffusion process.



Causvid



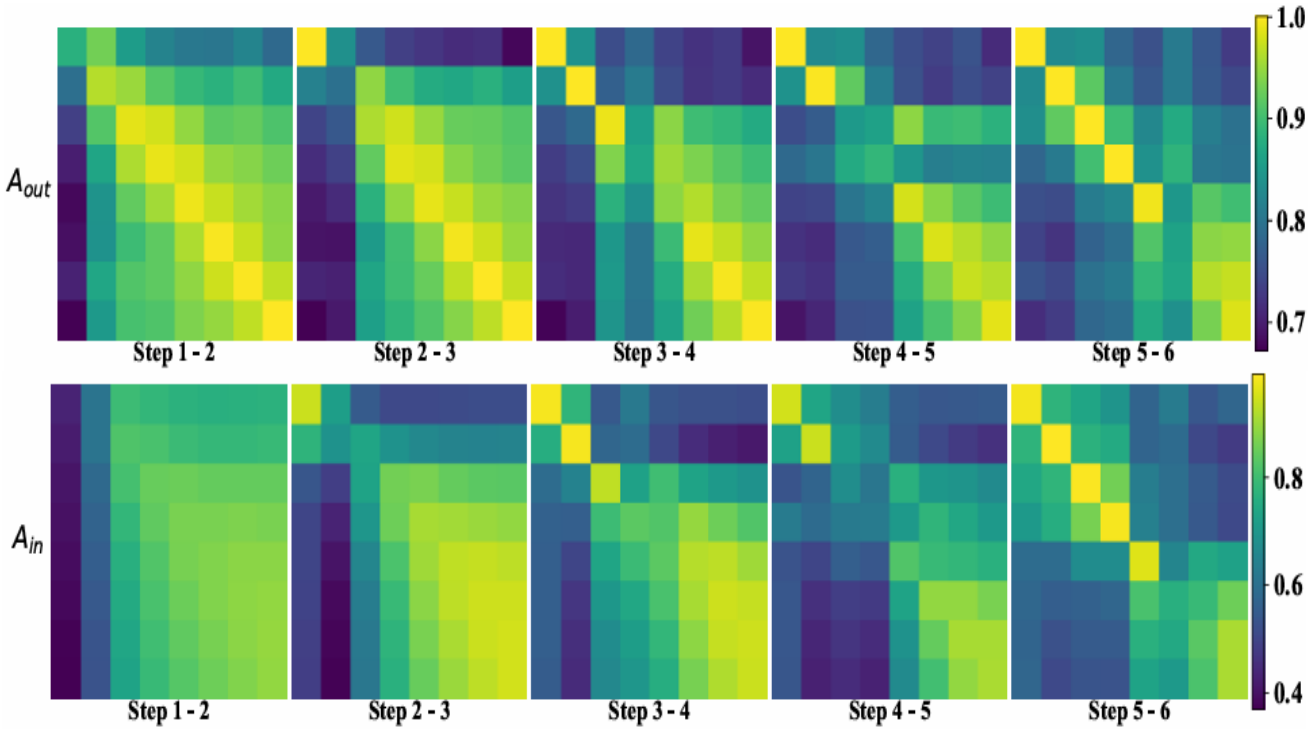
Self-Forcing



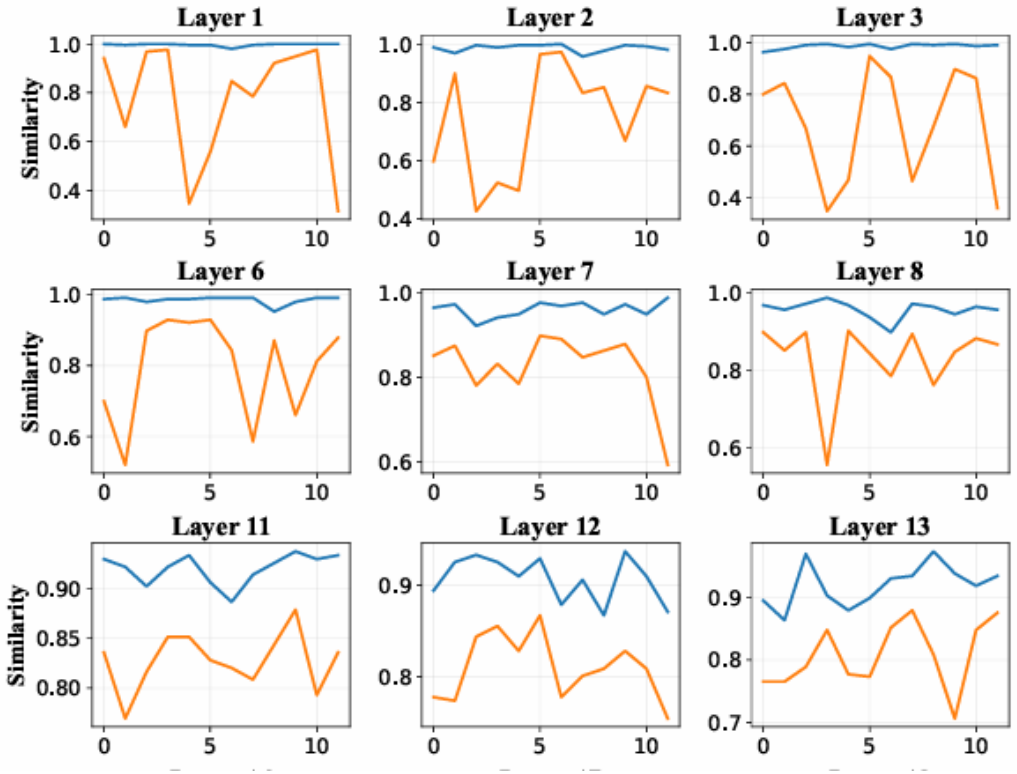
LongLive

However, in autoregressive diffusion models, long contexts still require **repeated attention** over a growing KV cache.

Motivation



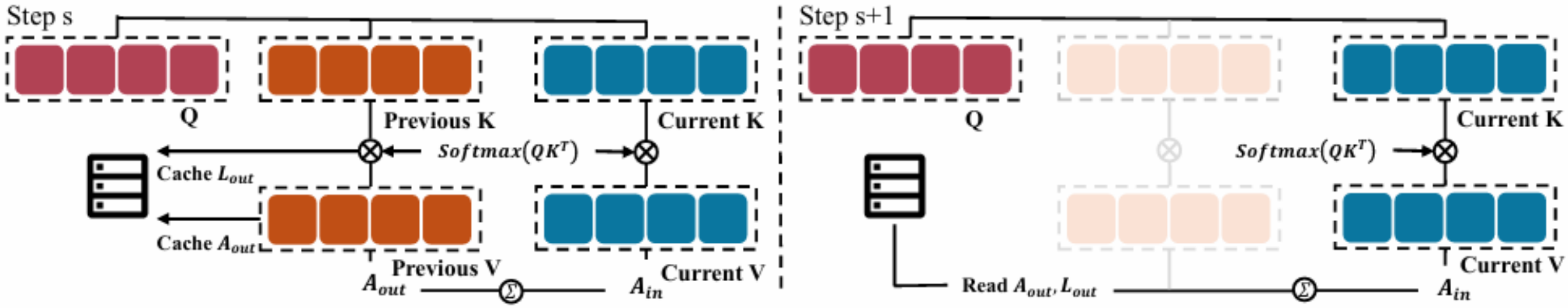
Visualization in block dLLM



Visualization in autoregressive video generation

- Key observation: **block-external attention is more stable across diffusion steps**

Method



- Split attention into two parts:

$$Z_{i,\text{in}} = \sum_{j \in \mathcal{J}_{\text{in}}} e^{s_{ij}}, \qquad U_{i,\text{in}} = \sum_{j \in \mathcal{J}_{\text{in}}} e^{s_{ij}} v_j,$$

$$Z_{i,\text{out}} = \sum_{j \in \mathcal{J}_{\text{out}}} e^{s_{ij}}, \qquad U_{i,\text{out}} = \sum_{j \in \mathcal{J}_{\text{out}}} e^{s_{ij}} v_j.$$

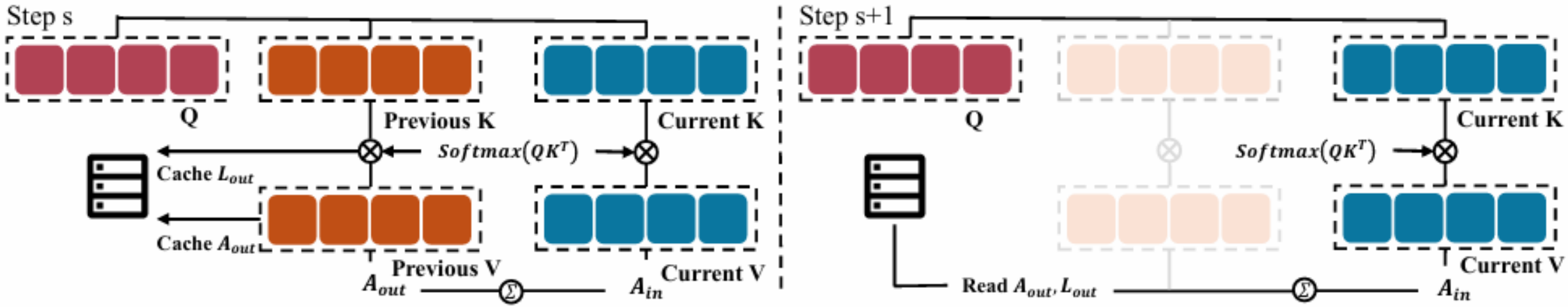
$$a_i = \frac{U_{i,\text{out}} + U_{i,\text{in}}}{Z_{i,\text{out}} + Z_{i,\text{in}}}.$$

- **Cache** stable A_{out} in step s and **reuse** in step $s+1$:

$$A^s_{\text{out}} = \frac{U^s_{\text{out}}}{Z^s_{\text{out}}}, \qquad L^s_{\text{out}} = \log Z^s_{\text{out}}.$$

$$A^{s+1}_{\text{in}} = \frac{U^{s+1}_{\text{in}}}{Z^{s+1}_{\text{in}}}, \qquad L^{s+1}_{\text{in}} = \log Z^{s+1}_{\text{in}}.$$

Method



- Recompute only changing A_{in}:

$$A_{\text{in}}^{s+1} = \frac{U_{\text{in}}^{s+1}}{Z_{\text{in}}^{s+1}}, \quad L_{\text{in}}^{s+1} = \log Z_{\text{in}}^{s+1}.$$

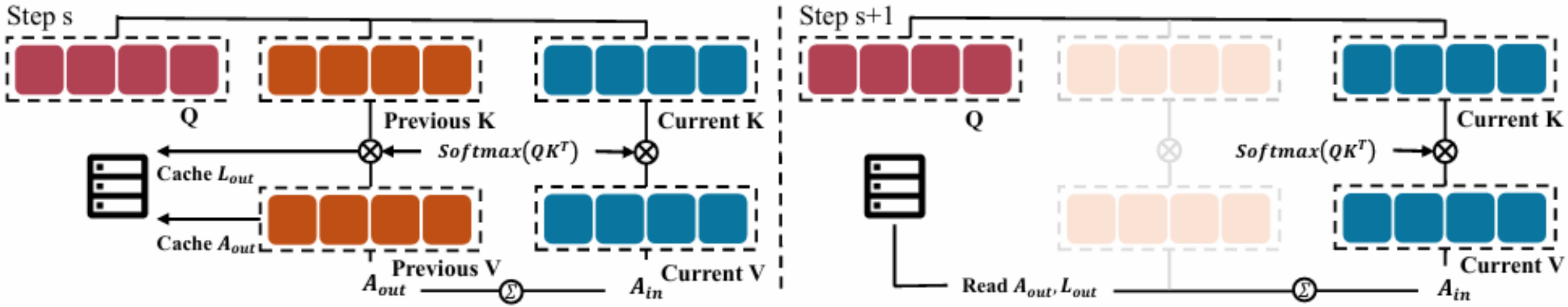
- Combine both parts:

$$A_{\text{full}}^{s+1} = \frac{e^{L_{\text{out}}^{s} - m} A_{\text{out}}^{s} + e^{L_{\text{in}}^{s+1} - m} A_{\text{in}}^{s+1}}{e^{L_{\text{out}}^{s} - m} + e^{L_{\text{in}}^{s+1} - m}}.$$

- Reuse-aware distillation:

$$\mathcal{L}_{\text{reuse}} = \text{KL}(p_{\text{teacher}} \,\|\, p_{\text{reuse}}) \quad \mathcal{L}_{\text{reg}} = \text{KL}\left(p_{\text{teacher}} \,\|\, p_{\text{dense}}^{\text{student}}\right) \quad \mathcal{L} = \mathcal{L}_{\text{reuse}} + \lambda \mathcal{L}_{\text{reg}}$$

Method



- Recompute only changing A_{in}:

$$A_{\text{in}}^{s+1} = \frac{U_{\text{in}}^{s+1}}{Z_{\text{in}}^{s+1}}, \quad L_{\text{in}}^{s+1} = \log Z_{\text{in}}^{s+1}.$$

- Combine both parts:

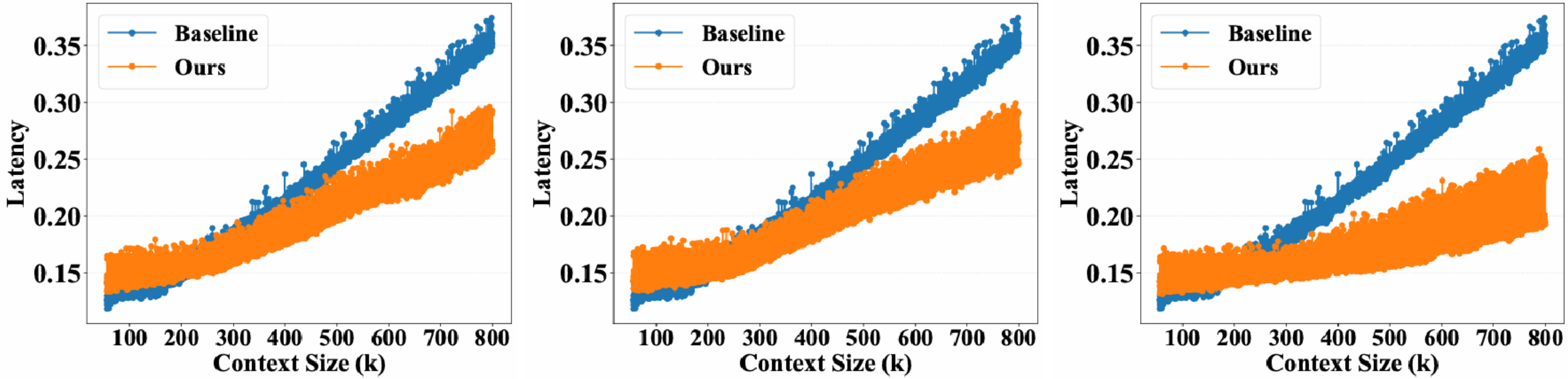
$$A_{\text{full}}^{s+1} = \frac{e^{L_{\text{out}}^{s} - m} A_{\text{out}}^{s} + e^{L_{\text{in}}^{s+1} - m} A_{\text{in}}^{s+1}}{e^{L_{\text{out}}^{s} - m} + e^{L_{\text{in}}^{s+1} - m}}.$$

- Reuse-aware distillation:

$$\mathcal{L}_{\text{reuse}} = \text{KL}(p_{\text{teacher}} \,\|\, p_{\text{reuse}}) \quad \mathcal{L}_{\text{reg}} = \text{KL}\left(p_{\text{teacher}} \,\|\, p_{\text{dense}}^{\text{student}}\right) \quad \mathcal{L} = \mathcal{L}_{\text{reuse}} + \lambda \mathcal{L}_{\text{reg}}$$

Performance on dLLM

Method	Block Size	TPS	Mathematical Reasoning			Coding			
			GSM8K	MATH500	AIME	MBPP	HumanEval	LiveCodeBench-V2	LiveBench
Trado-8B-Thinking	4	312	93.25	86.00	33.33	25.60	50.61	36.79	32.03
Trado-8B-Thinking + Ours	4	451	93.12	85.80	33.33	33.60	51.22	35.64	31.25
Trado-8B-Thinking	8	532	91.74	82.00	26.67	29.00	54.27	27.01	30.47
Trado-8B-Thinking + Ours	8	674	90.22	81.80	26.67	32.00	53.66	26.42	29.34



- FlashBlock consistently improves decoding throughput while maintaining comparable reasoning and coding performance.
- The acceleration becomes **more pronounced under long-context generation**.

Experiment on video generation

- **Performance on VBench**

Method	Density	E2E Lat. (s) ↓	Attn. (s) ↓	HF	CR	CT	PH	CS
LongLive-1.3B	100%	93.23	23.02	**0.8188**	**0.1821**	0.2796	0.4273	0.4195
SpargeAttn	30%	92.34	22.04	0.5993	0.1821	0.2717	0.4019	0.4770
SpargeAttn	50%	92.78	22.56	0.7609	0.1808	0.2837	0.4257	0.4080
Ours	55%	**84.64**	**14.43**	0.7861	0.1818	**0.3026**	**0.4231**	**0.4310**

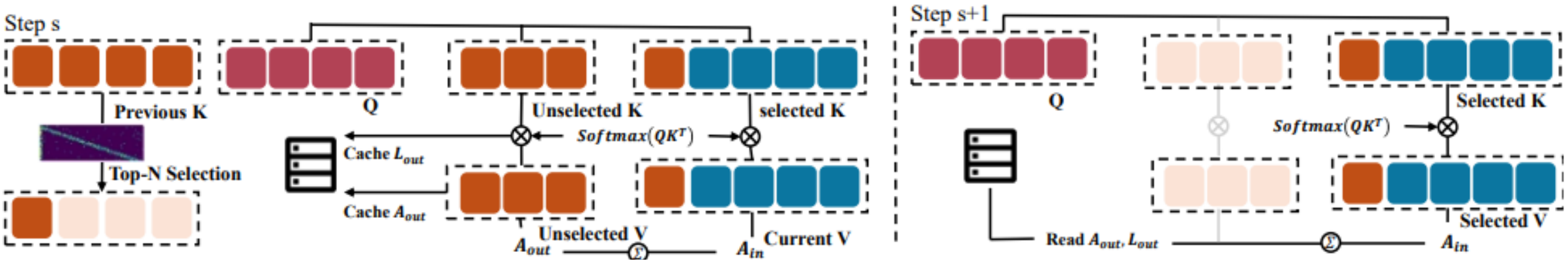
- **Comparison with attention reuse method**

Table 9. Comparison with PAB on five representative VBench subsets. Lower is better for latency metrics, and higher is better for VBench scores.

Method	Dynamic Attr.	Human Identity	Motion Rat.	Composition	Material	E2E Lat. (s) ↓	Attn. (s) ↓
PAB (step 2)	0.2088	0.7376	0.4310	0.1802	0.3714	70.93	17.72
PAB (steps 2–3)	0.2198	0.6986	**0.4483**	**0.1832**	0.3747	66.73	13.24
PAB (steps 2–4)	0.2857	0.6876	0.3966	0.1814	0.3906	**64.23**	**11.02**
Dense	**0.3583**	**0.7663**	0.4195	0.1821	0.5674	73.28	19.89
`FlashBlock`	0.3475	0.7593	0.4310	0.1818	**0.6138**	66.07	12.46

- FlashBlock reduces attention latency by up to **1.6×** with negligible impact on video quality.
- The method preserves fidelity, controllability, and temporal consistency across VBench metrics.
- Compared with existing attention reuse approaches, FlashBlock achieves a better quality–efficiency trade-off.

Combined with sparse attention method



Method	GSM8K		MATH500		HumanEval	
	Acc.	Δ	Acc.	Δ	Pass@1	Δ
Trado-4B-Instruct	91.21	–	70.80	–	45.12	–
SparseD ($d = 20\%$)	34.72	+7.96	39.40	+7.40	23.78	+9.76
SparseD + Ours ($d = 20\%$)	42.68		46.80		33.54	
SparseD ($d = 30\%$)	68.61	+3.64	59.20	+5.40	29.88	+6.71
SparseD + Ours ($d = 30\%$)	72.25		64.60		36.59	
SparseD ($d = 40\%$)	84.61	+2.65	66.20	+3.20	39.02	+5.49
SparseD + Ours ($d = 40\%$)	87.26		69.40		44.51	

Table 3. L1 distance between sparse attention outputs and full attention for the first attention layer on a randomly sampled input. Results are reported under different attention density ratios.

Method / Density	50%	40%	30%	20%	10%
SparseD	0.0014	0.0015	0.0022	0.0028	0.0031
SparseD + Ours	**0.0005**	**0.0005**	**0.0006**	**0.0008**	**0.0008**

- FlashBlock is orthogonal to sparse attention and can be **seamlessly combined** with existing sparsification strategies.
- It recovers part of the **discarded attention information** and significantly improves accuracy under aggressive sparsity.

Ablation Study

τ	Distill.	AIME	MATH500	HumanEval	MBPP
–	–	33.33	86.00	50.61	25.60
2	×	26.67	80.40	48.17	31.40
2	✓	**33.33**	**85.80**	**51.22**	**33.60**
3	×	20.00	78.40	44.51	30.20
3	✓	**30.00**	**85.00**	**50.61**	**31.60**
4	×	6.67	48.40	37.20	24.20
4	✓	**13.33**	**66.20**	**42.07**	**28.60**

- $\tau = 2$ achieves near-optimal performance with minimal quality degradation.
- The distilled model **consistently** improves downstream performance across benchmarks.

Visualization

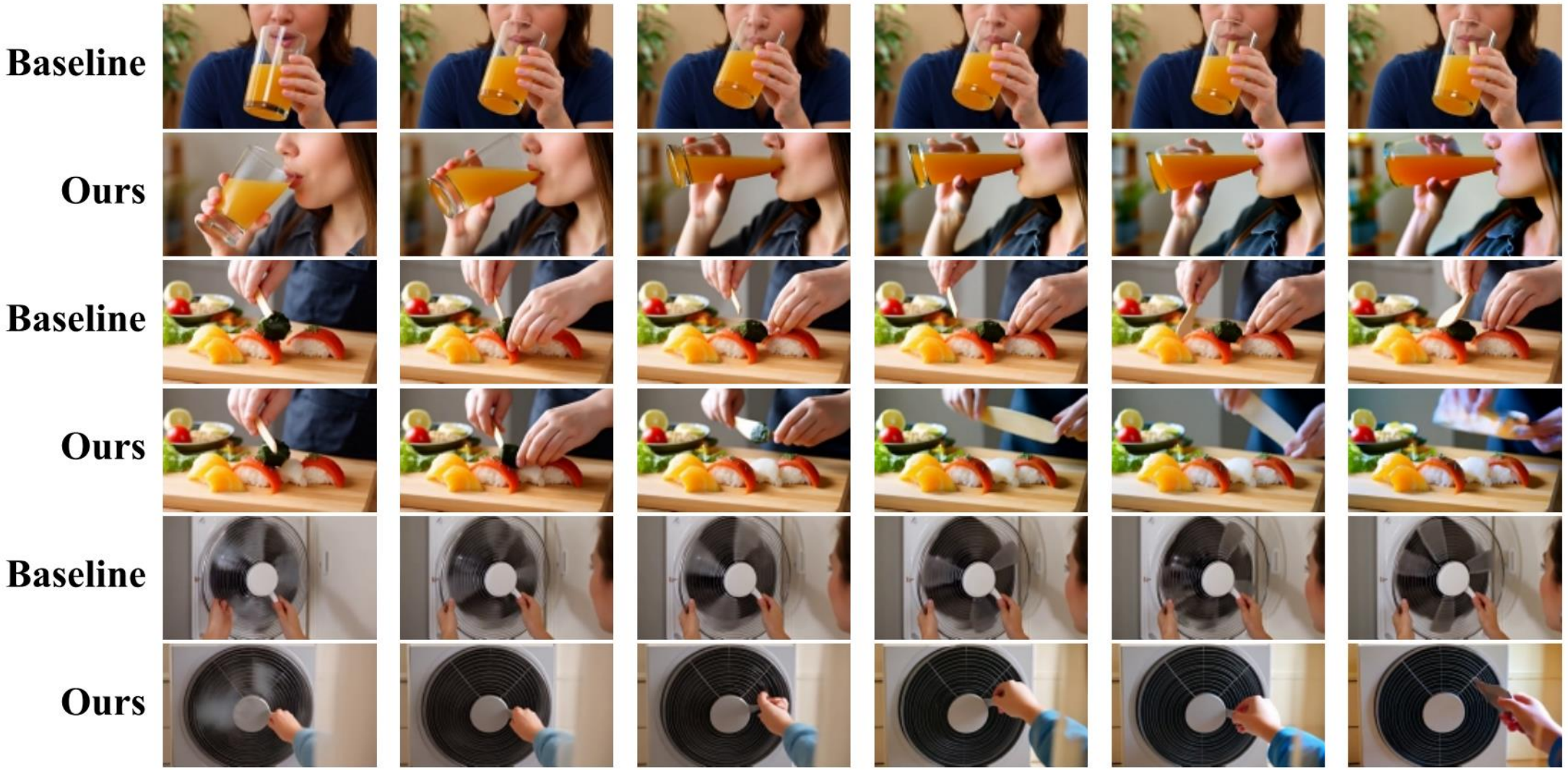


Figure 7. **Qualitative examples on Motion Rationality.** We visualize three representative video examples selected from the Motion Rationality dimension of VBench using the LongLive-1.3B model. For each example, we show uniformly sampled frames. Rows are organized in pairs, where the upper row corresponds to the dense baseline and the lower row corresponds to our accelerated method.

Visualization

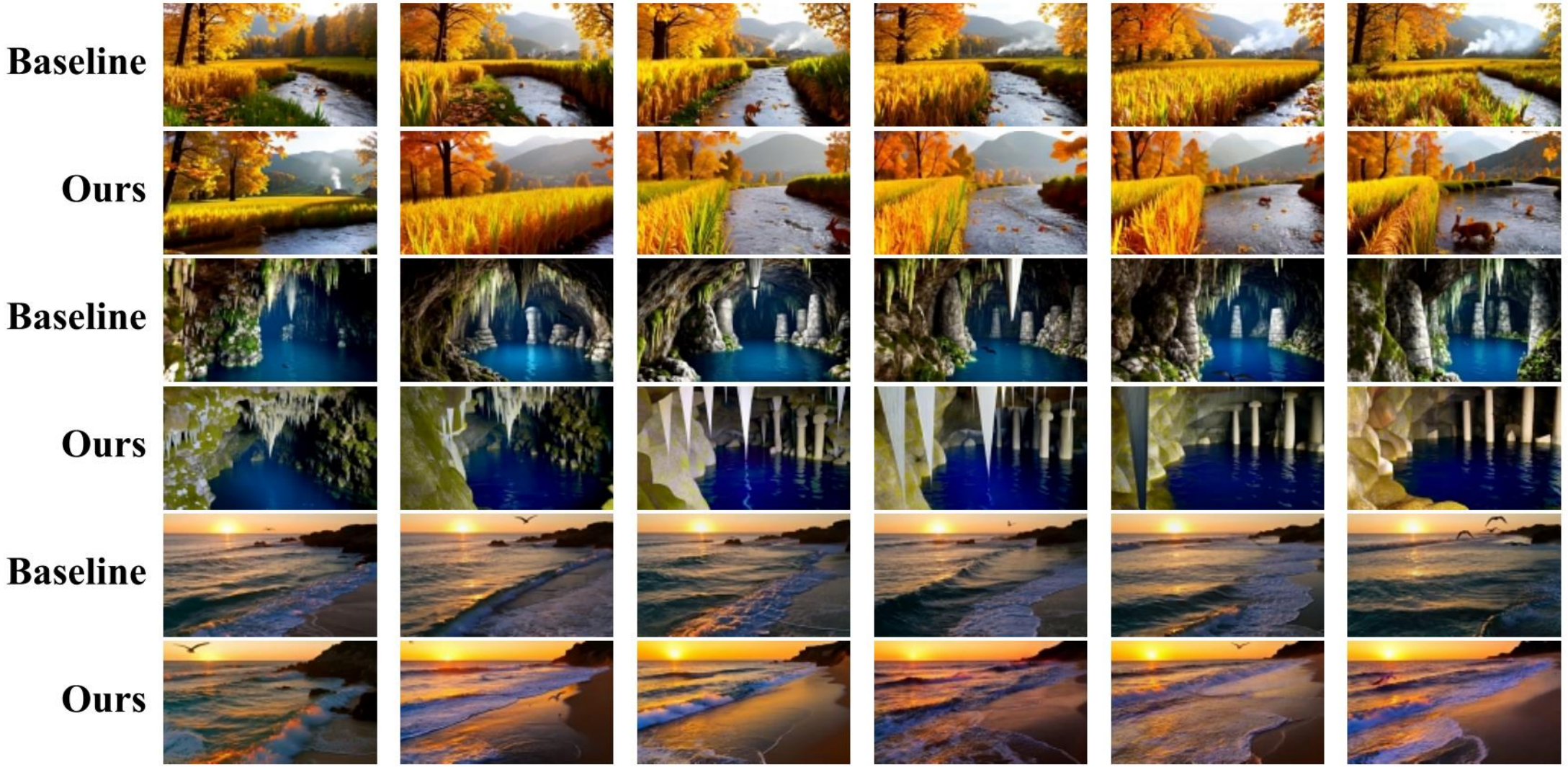


Figure 10. **Qualitative examples on Complex Landscape.** We visualize three representative video examples selected from the Complex Landscape dimension of VBench using the LongLive-1.3B model. For each example, we show uniformly sampled frames. Rows are organized in pairs, where the upper row corresponds to the dense baseline and the lower row corresponds to our accelerated method.

Conclusion

- Flashblock: attention reuse for block diffusion
- Efficient attention acceleration for long contexxt
- Future: system optimization with sparse attention methods

Paper: https://arxiv.org/abs/2602.05305

Project page: https://caesarhhh.github.io/FlashBlock/

Thanks for listening!