SparseAttn for Video Generation

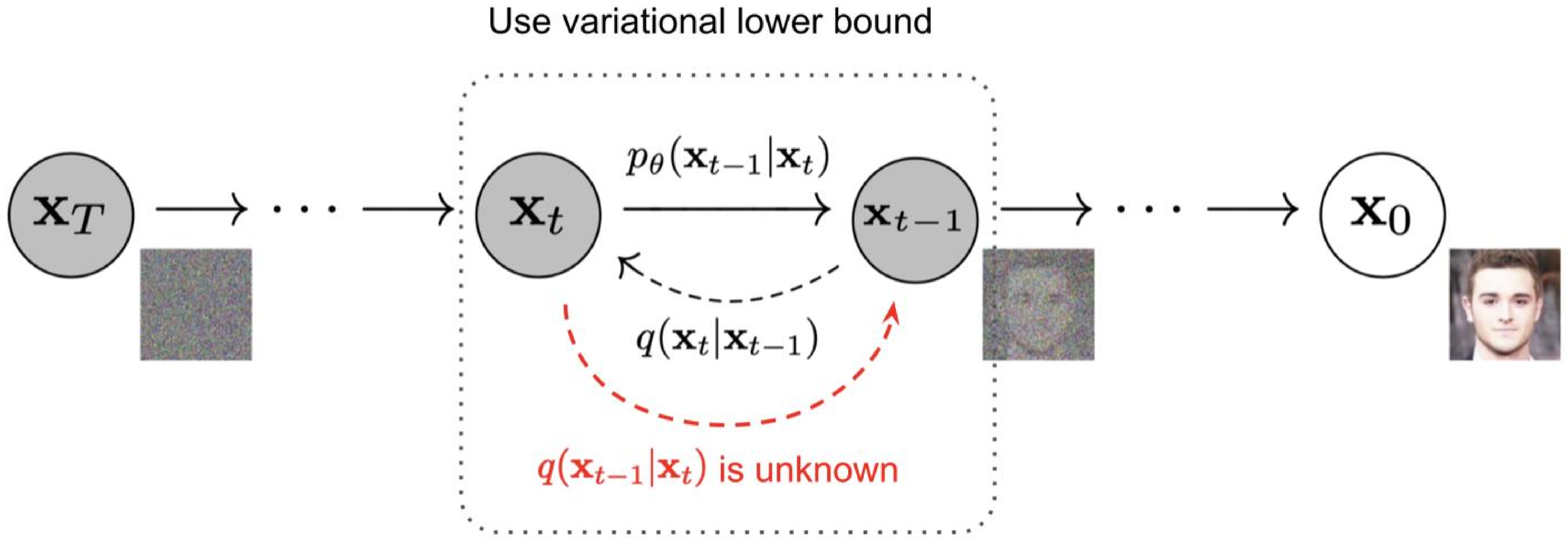
2025.04.11

Yulin Sun

1. Background

Review the diffusion process and 3D attn in t2v/i2v models

Diffusion Process & DiT



Diffusion Process

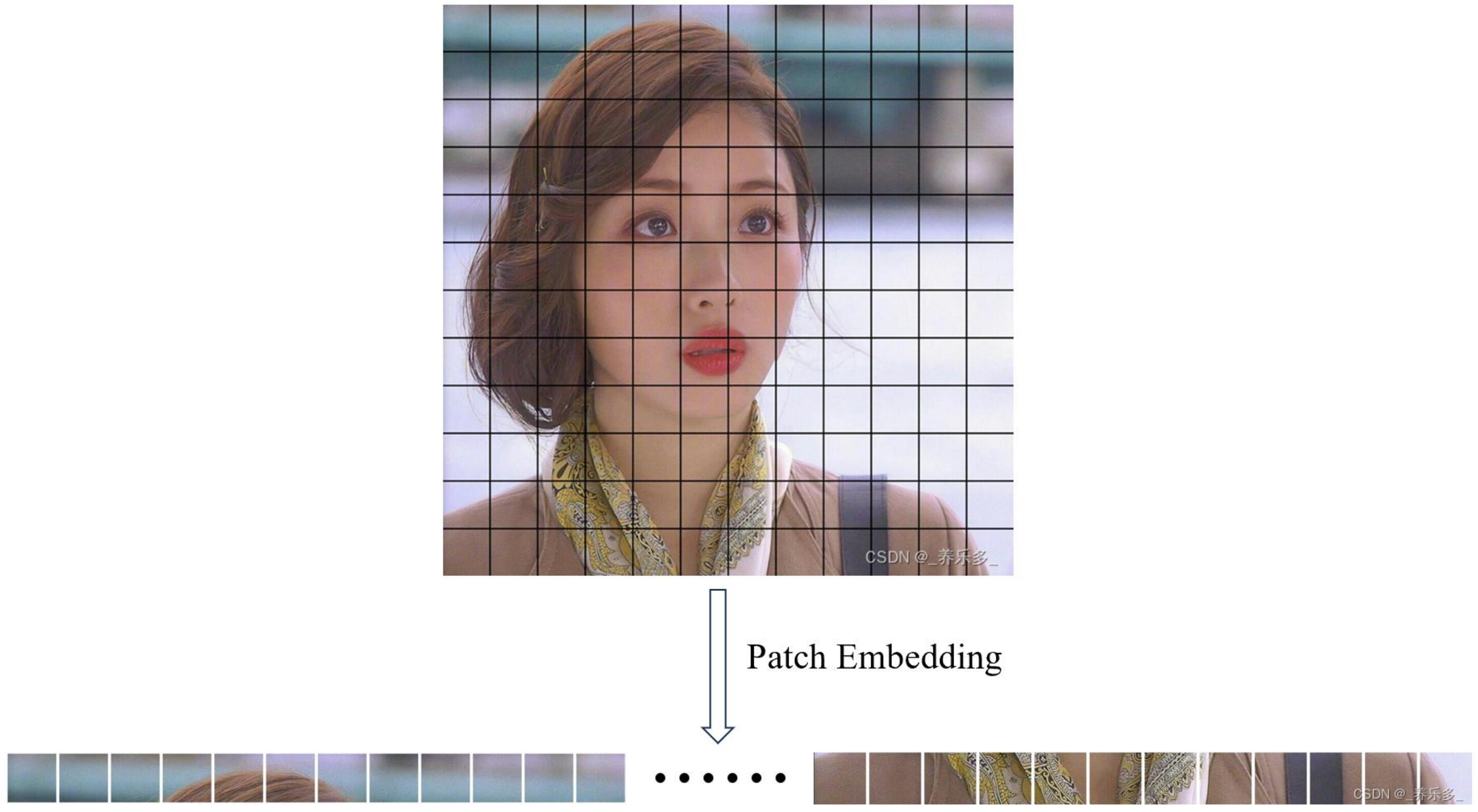
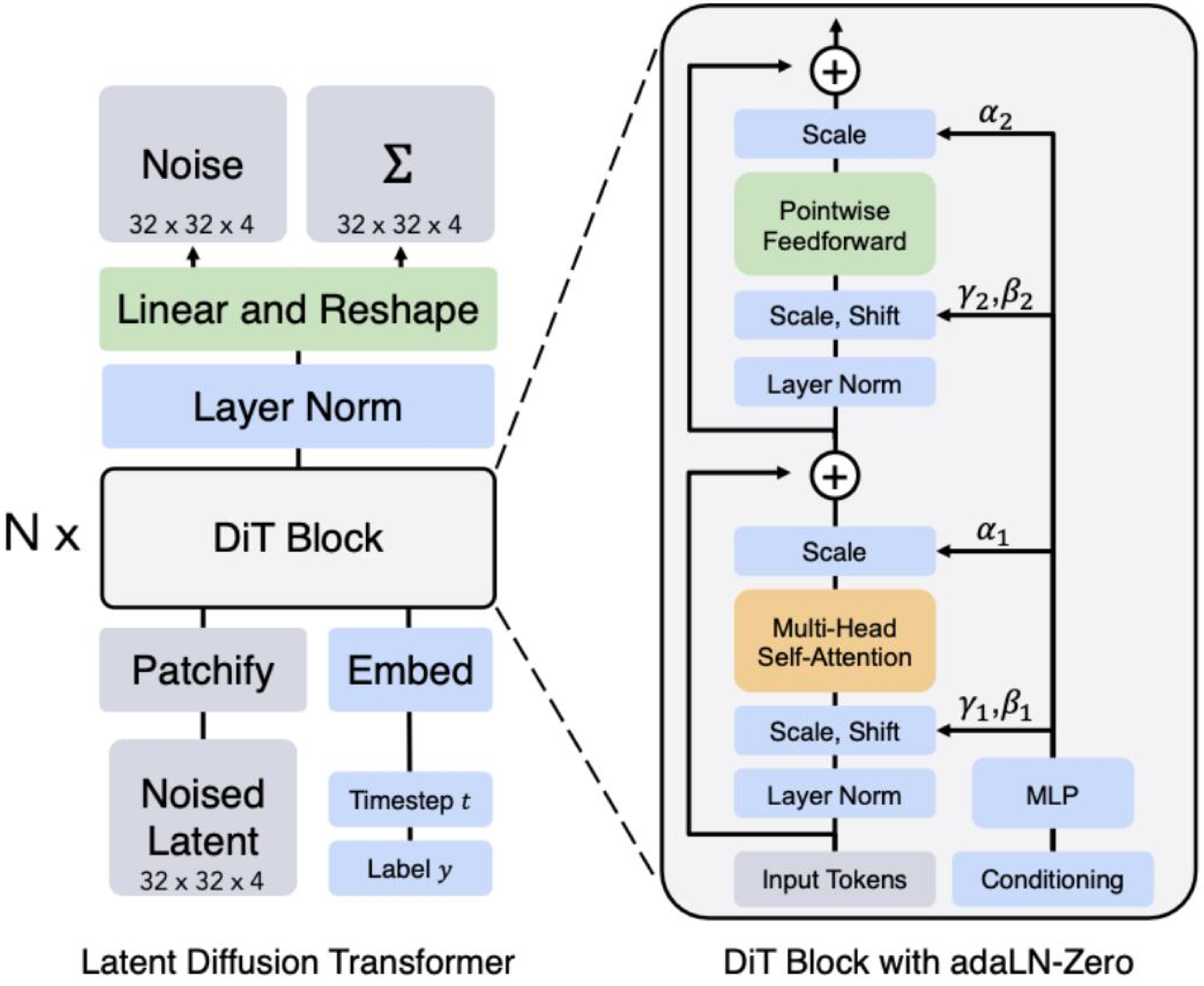


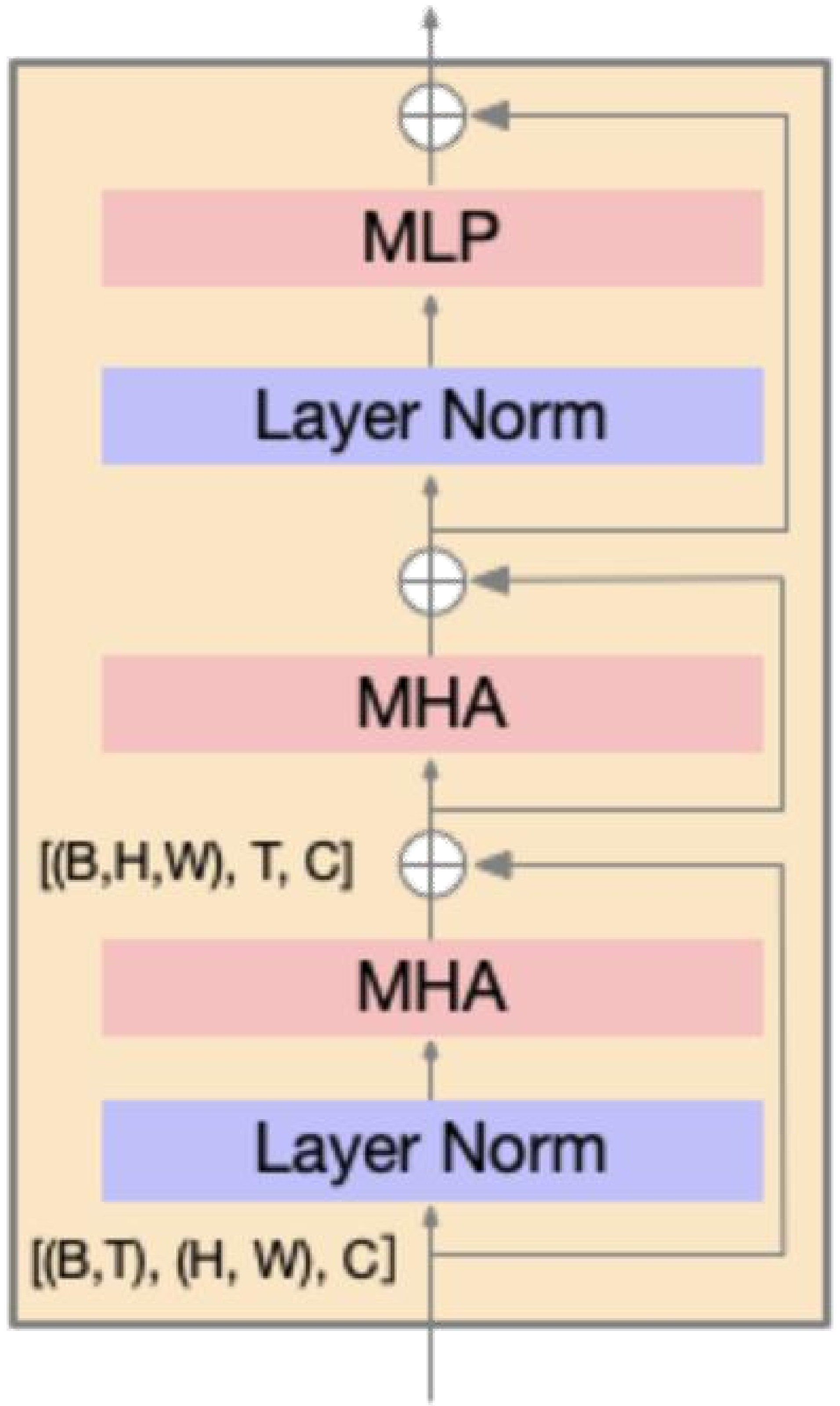
Image Patchify -> Tokens



DiT Structure

2D & 3D Attn in video generation

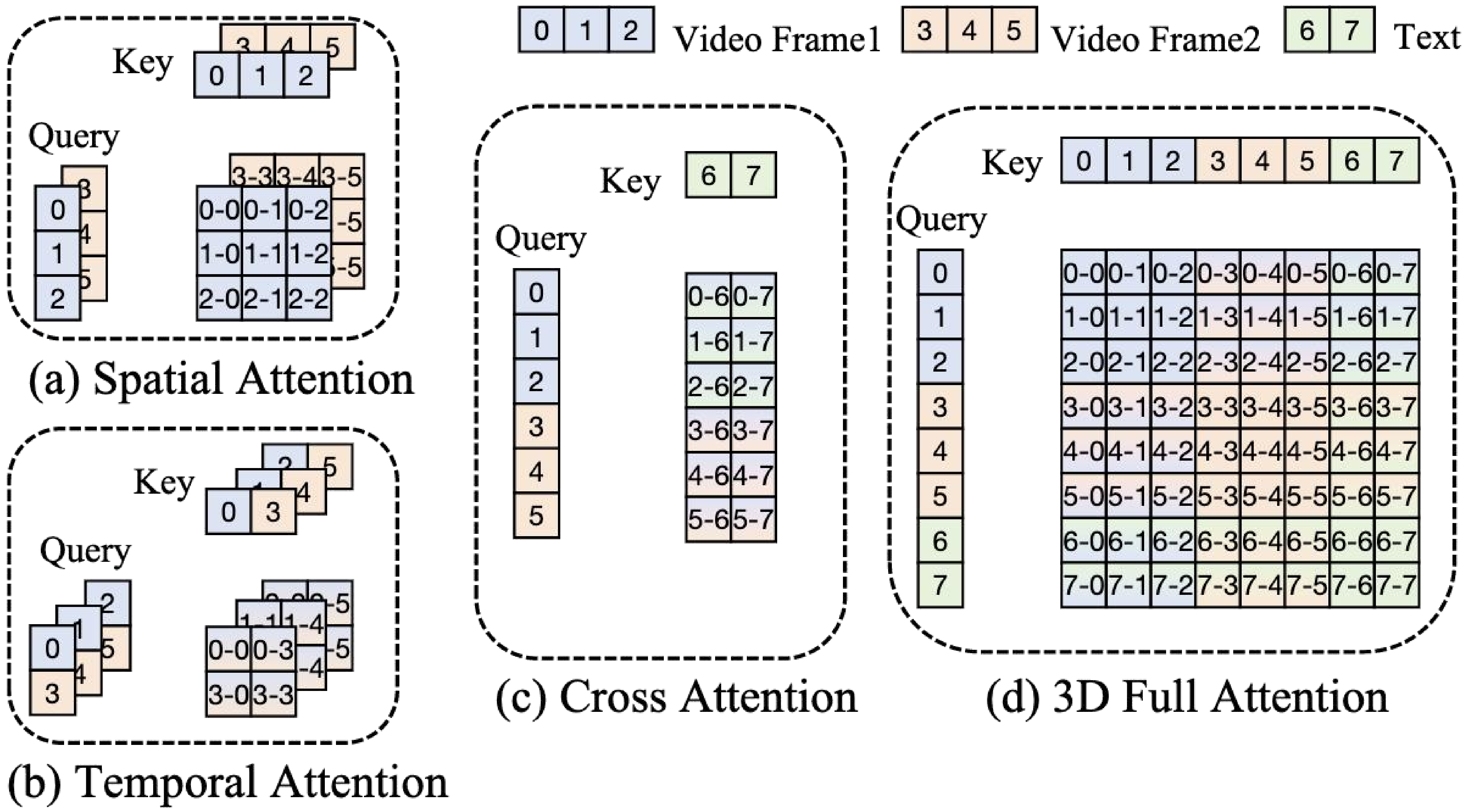
- 2D attn:
  - After encoder: hidden_states shapes (B, T, H, W, C)
  - Spatial Attn:
    - Reshape to (B*T, H*W, C)
    - Seq_len = H*W, only tokens with the same T do attn
  - Temporal Attn:
    - Reshape to (B*H*W, T, C)
    - Seq_len = T, only tokens with the same H*W do attn



STDiT (Sequential)

2D & 3D Attn in video generation

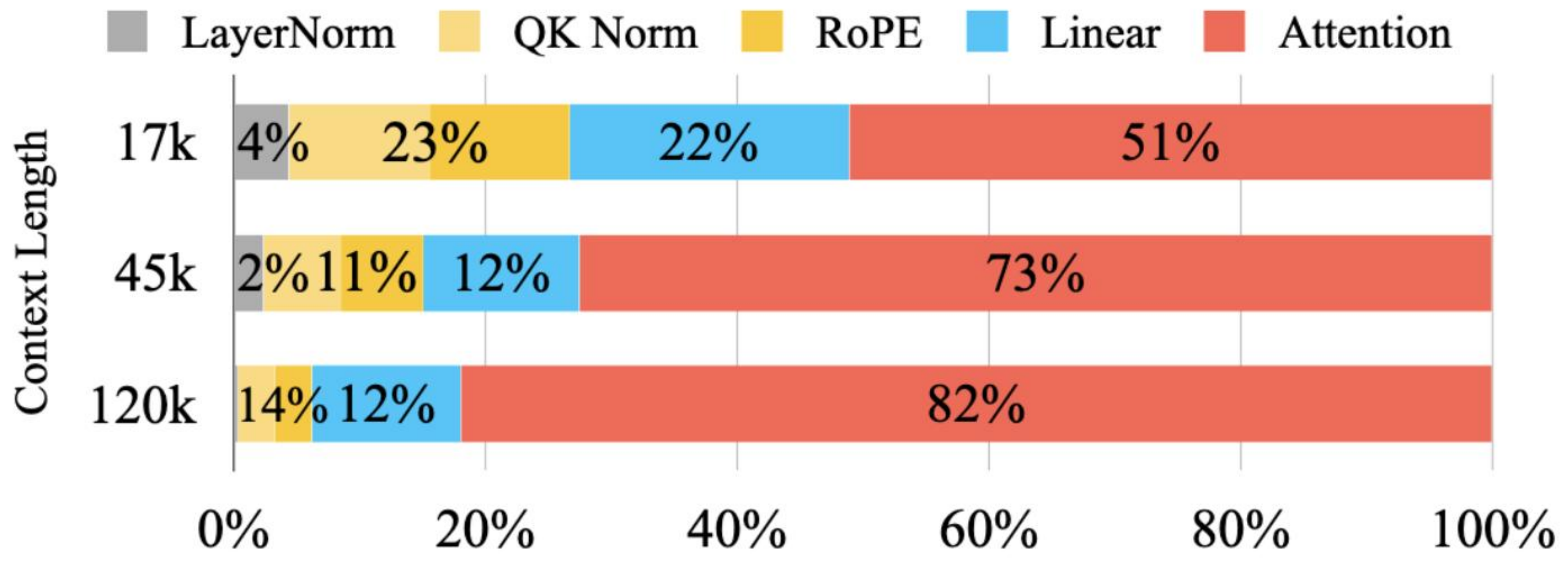
- 3D full attn:
  - After encoder: hidden_states shapes (B, T, H, W, C)
  - Reshape to (B, T*H*W, C)
  - Seq_len = T*H*W
- Capture all the influence between tokens
- i.e. 2D doesn’t have attn between (frame 0, token 2) and (frame 1, token 0)

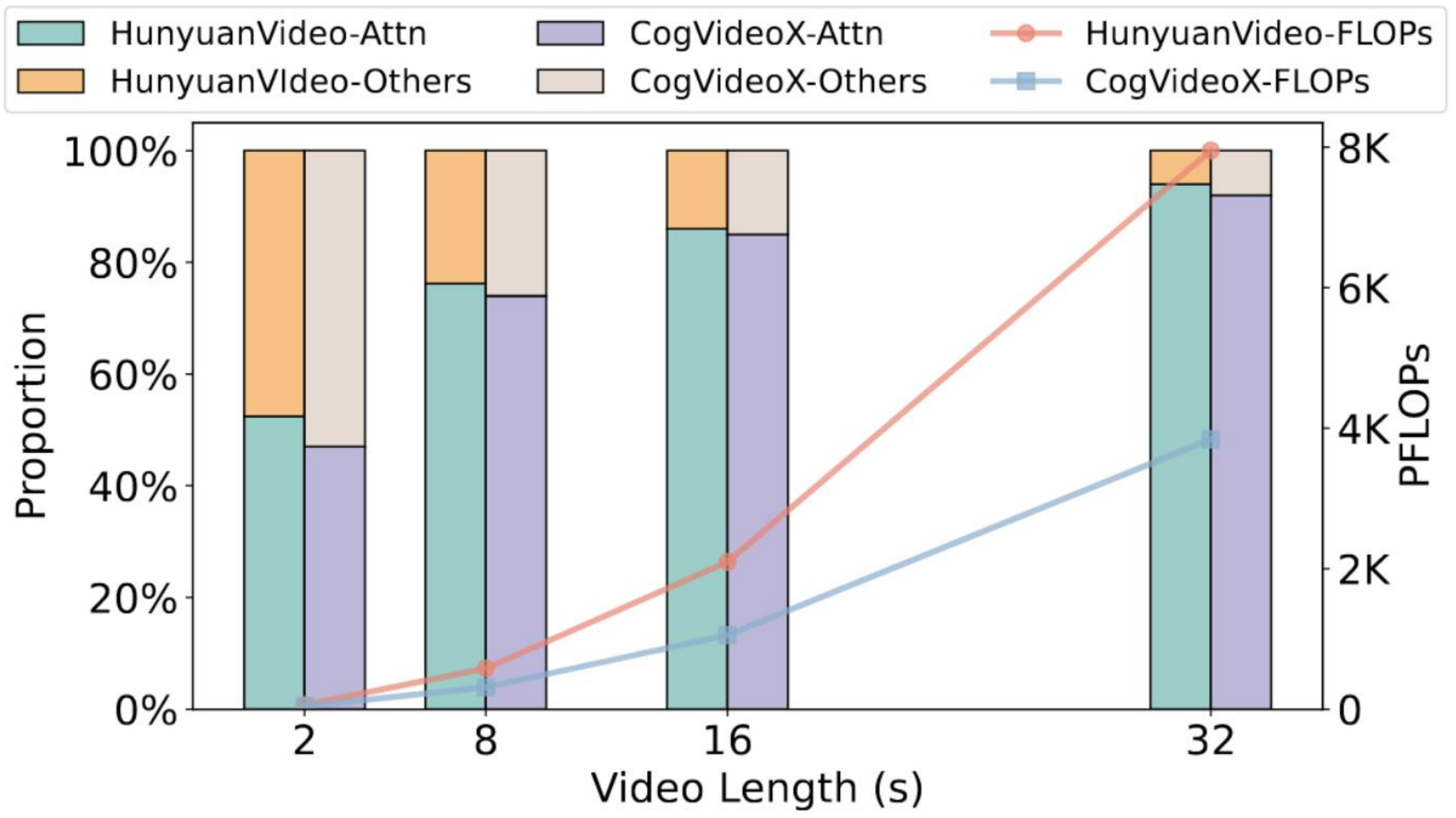


2D attn compare with 3D attn

Bottleneck in 3D Attention

- Long seqlen (e.g. 119054 for a 5s video in hunyuan) with $\mathcal{O}(n^2)$ complexity
- High computation rate in total inference time
- Solution:
  - Reduce redundant computation in 3D attention
  - Leverage spatial and temporal similarity

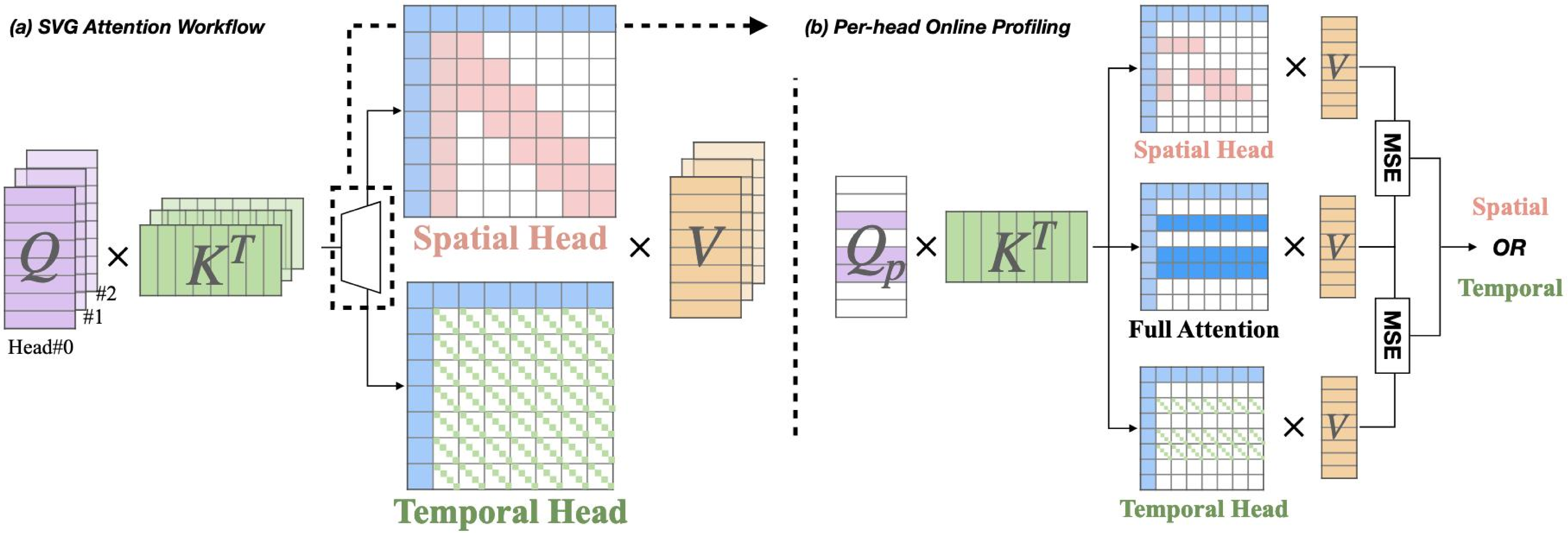




2. Sparse VideoGen (aka SVG)

Distinguish spatial or temporal pattern in attn map

Overview

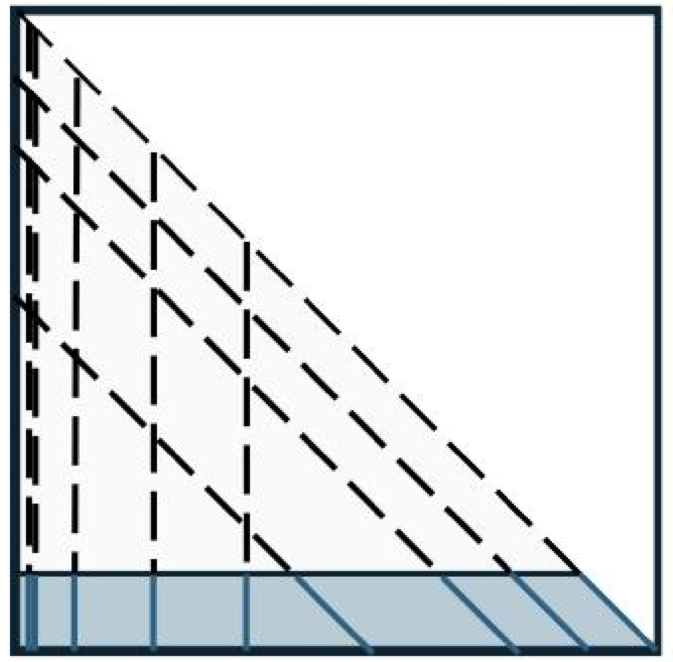


- Contrib. 1: Identify two types of attention heads (spatial & temporal)
- Contrib. 2: Online profiling strategy for sparsity identification (0.02% overhead cmp attn)
- Contrib. 3: Hardware-efficient layout transformation

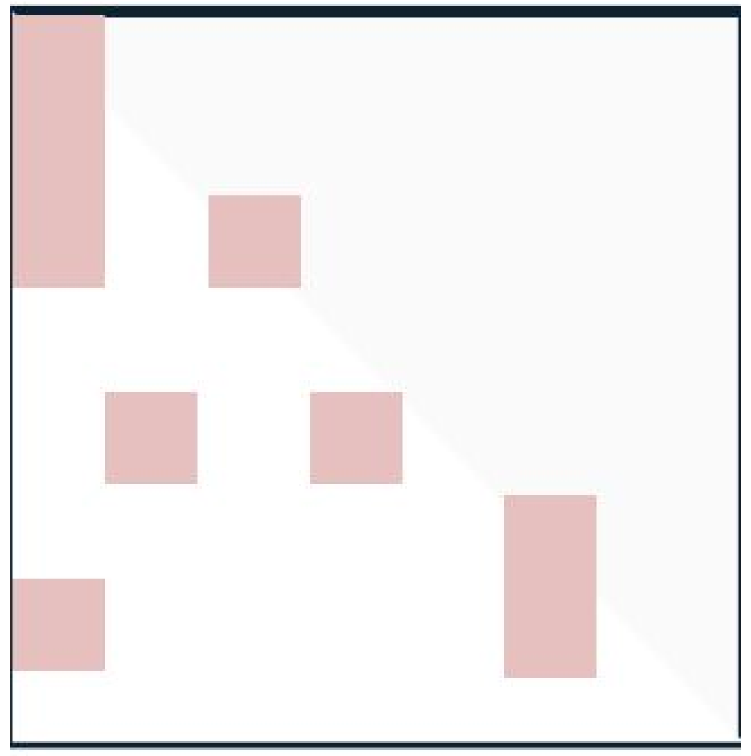
Contribution 1

AttnMap Pattern —— Spatial & Temporal Heads

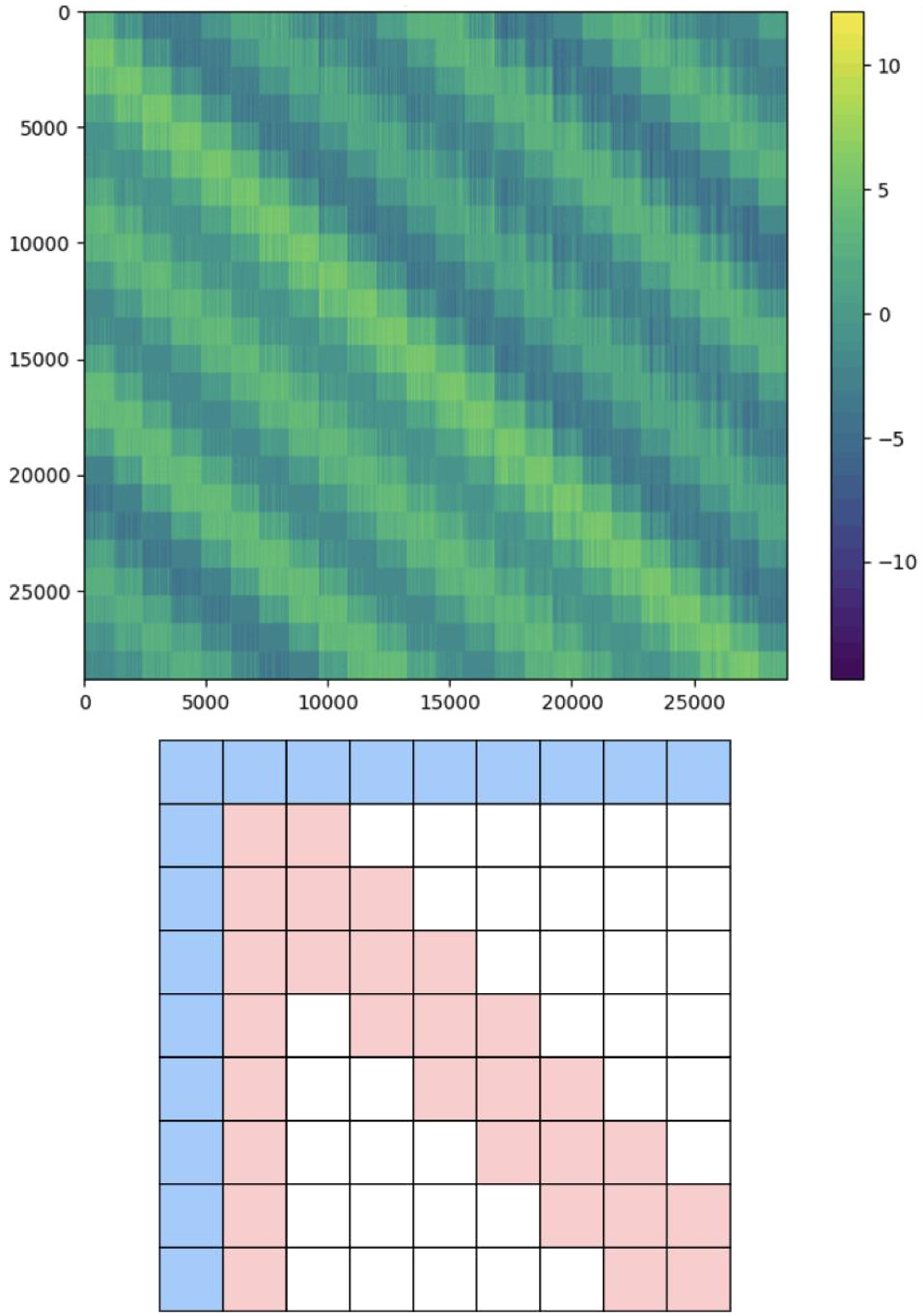
A shape



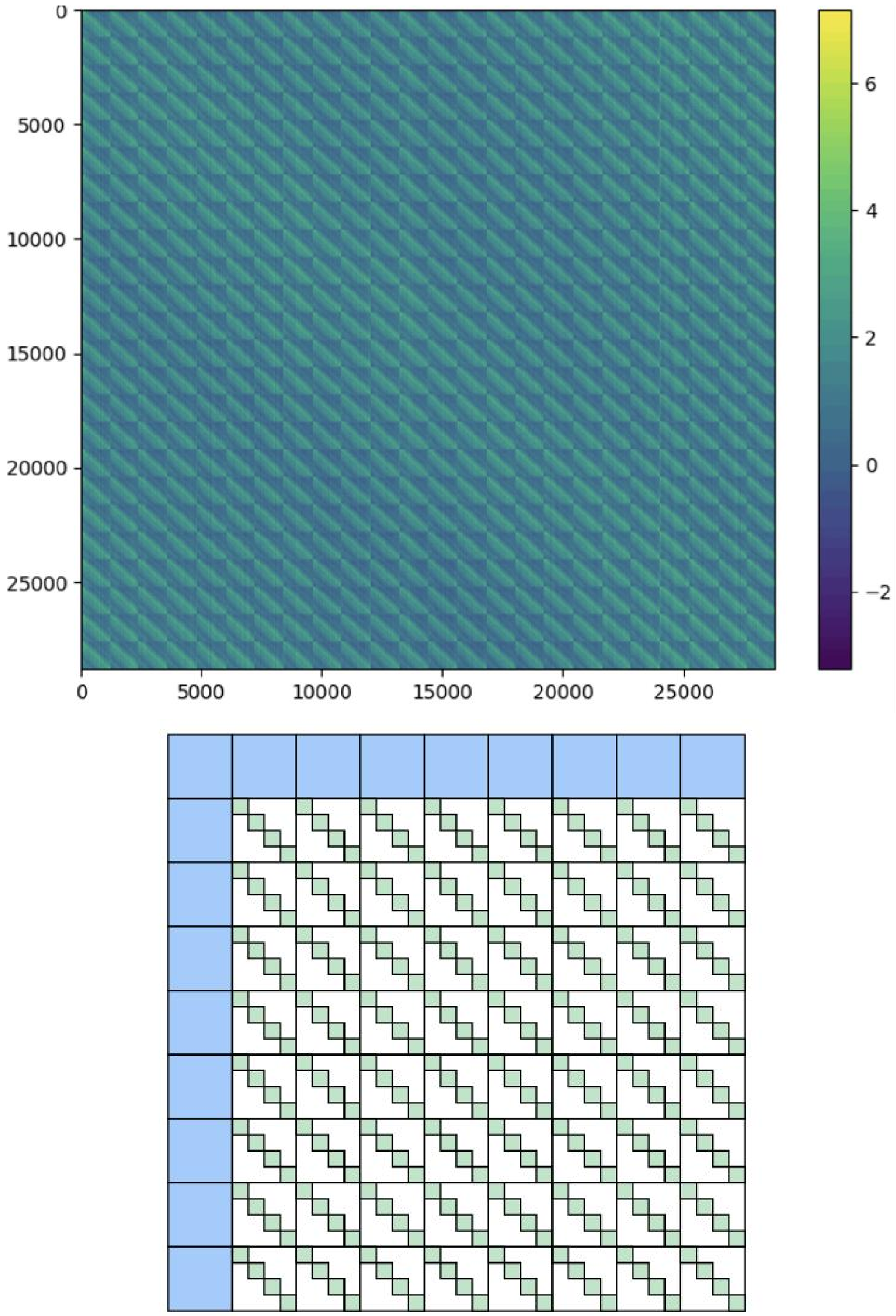
Vertical - Slash



Block



Spatial Sparse Pattern

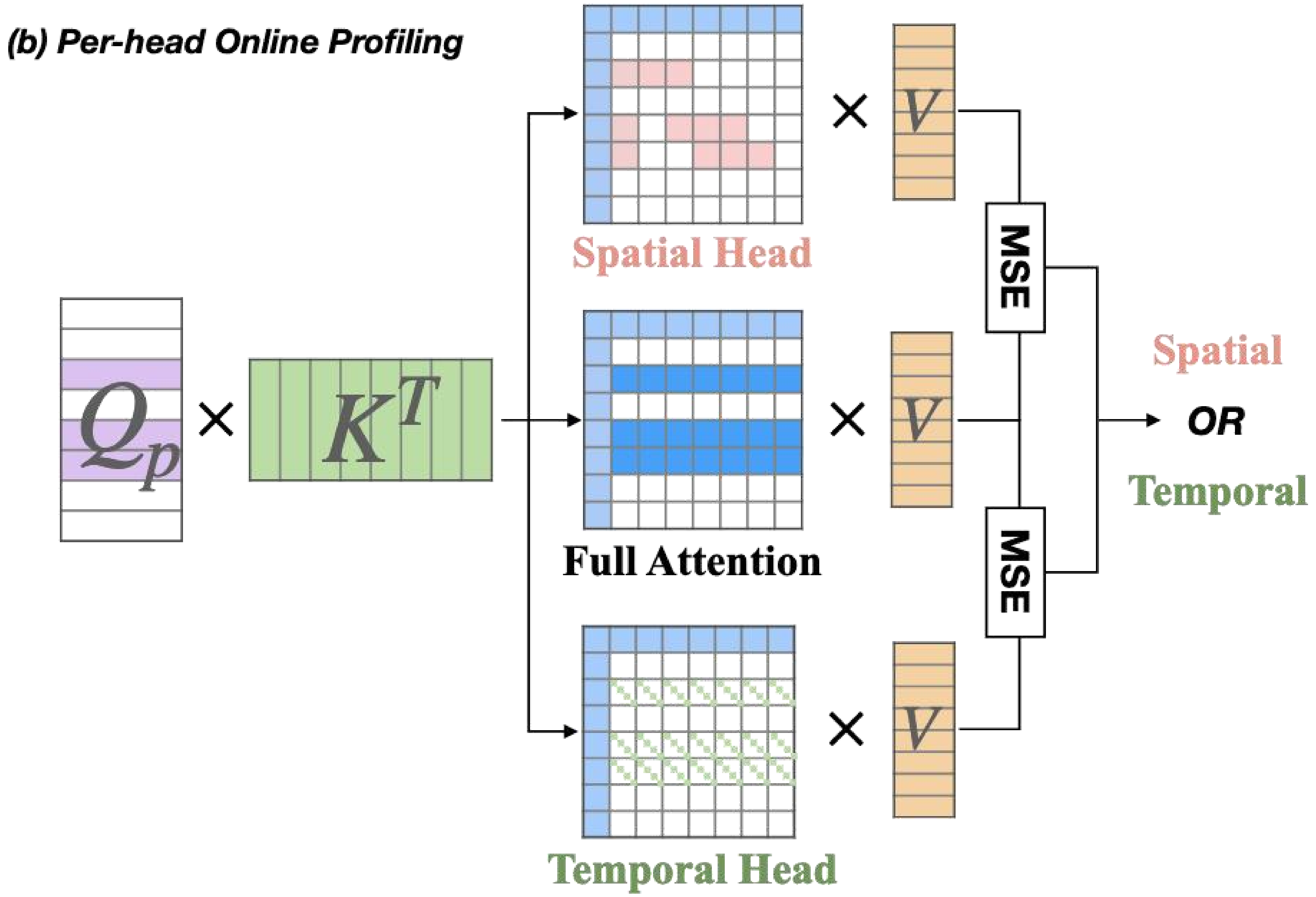


Temporal Sparse Pattern

Contribution 2

Sampling Method for Online Profiling Patterns

- Sparse patterns in the same attn head are variant in different prompts or layers —— need online profiling
- Metric: MSE
- Set width of diagonal manually
- Uniformly sample few rows (querys) (32 / 119054) for profiling

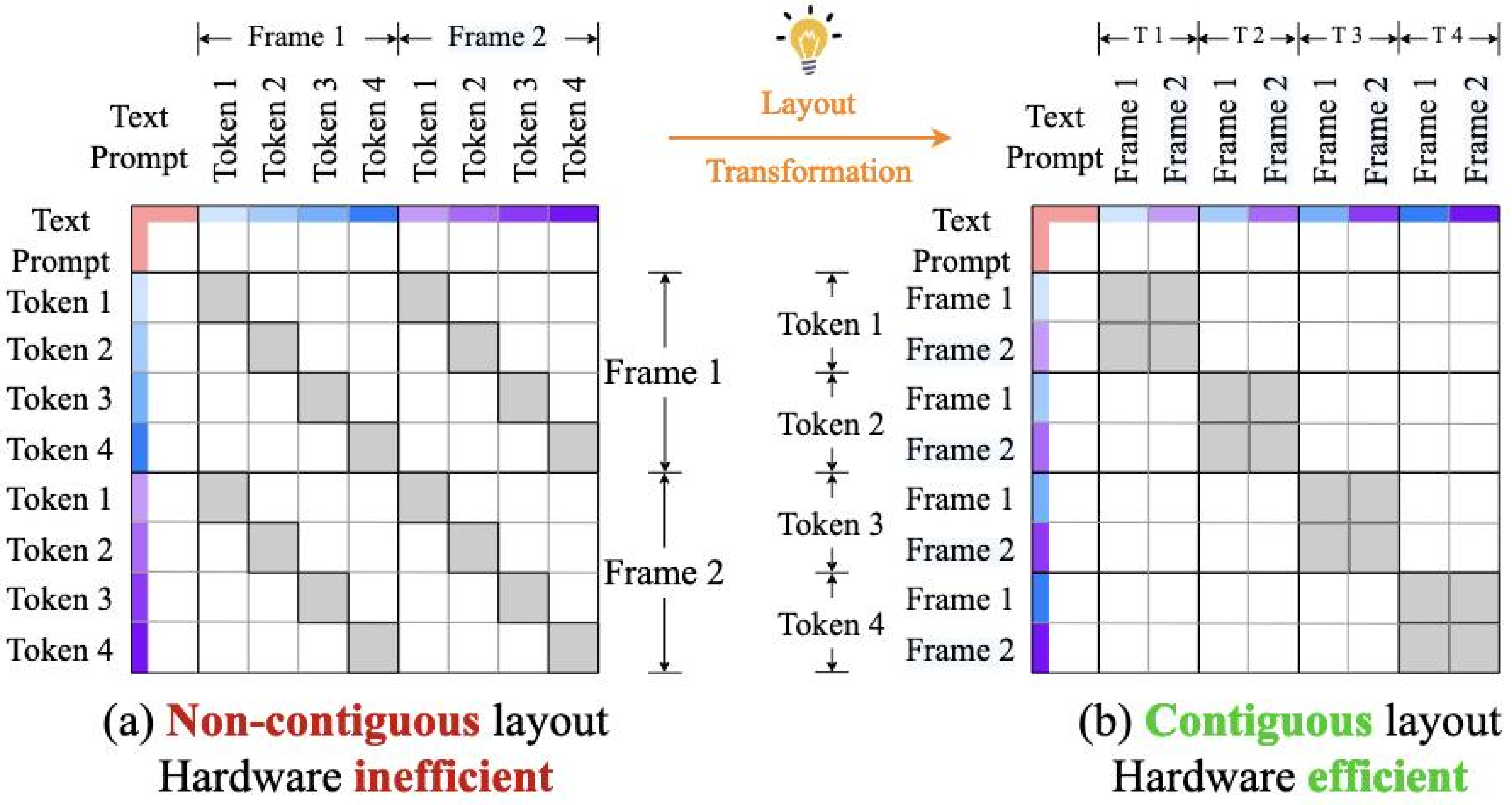


Use MSE to decide spatial/temporal pattern

Contribution 3

Online Layout Transformation for Hardware Efficiency

- In temporal heads
  - Stride: #tokens_in_a_frame
  - Non-contiguous
- Reshape Q & K before attn
  - spatial first -> temporal first
  - Little overhead

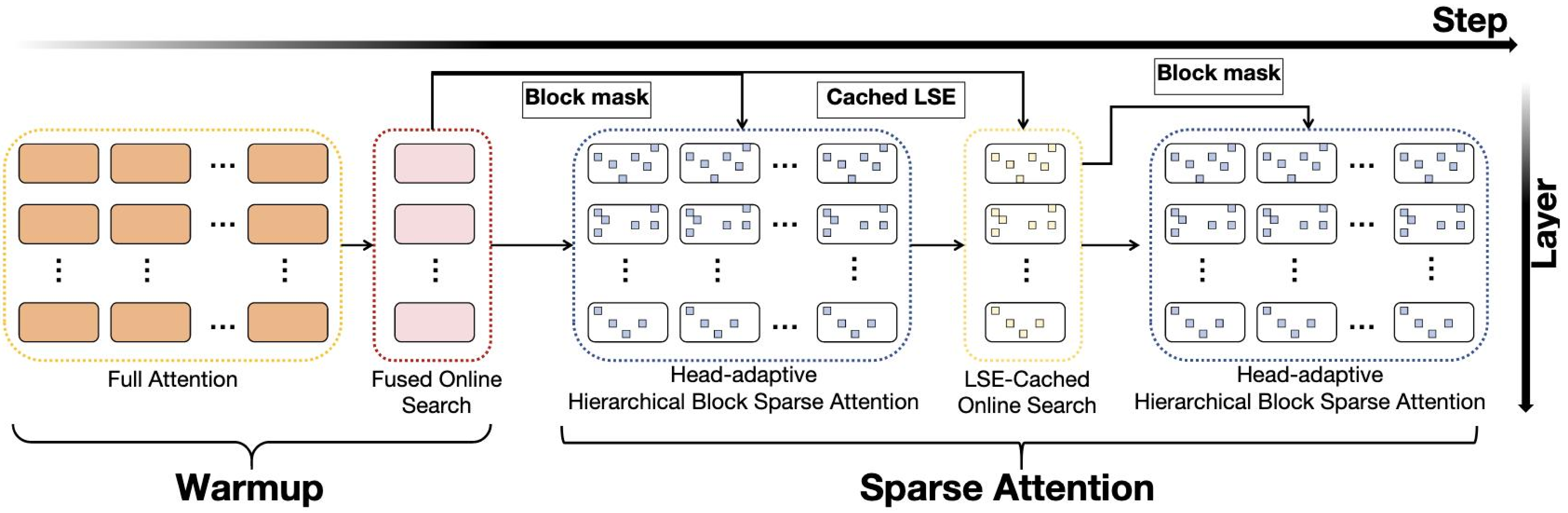


Order: spatial first -> temporal first

3. Adaptive Sparse Attention (aka Ada Spa)

Blockified attn mask and LSE-cached method for profiling

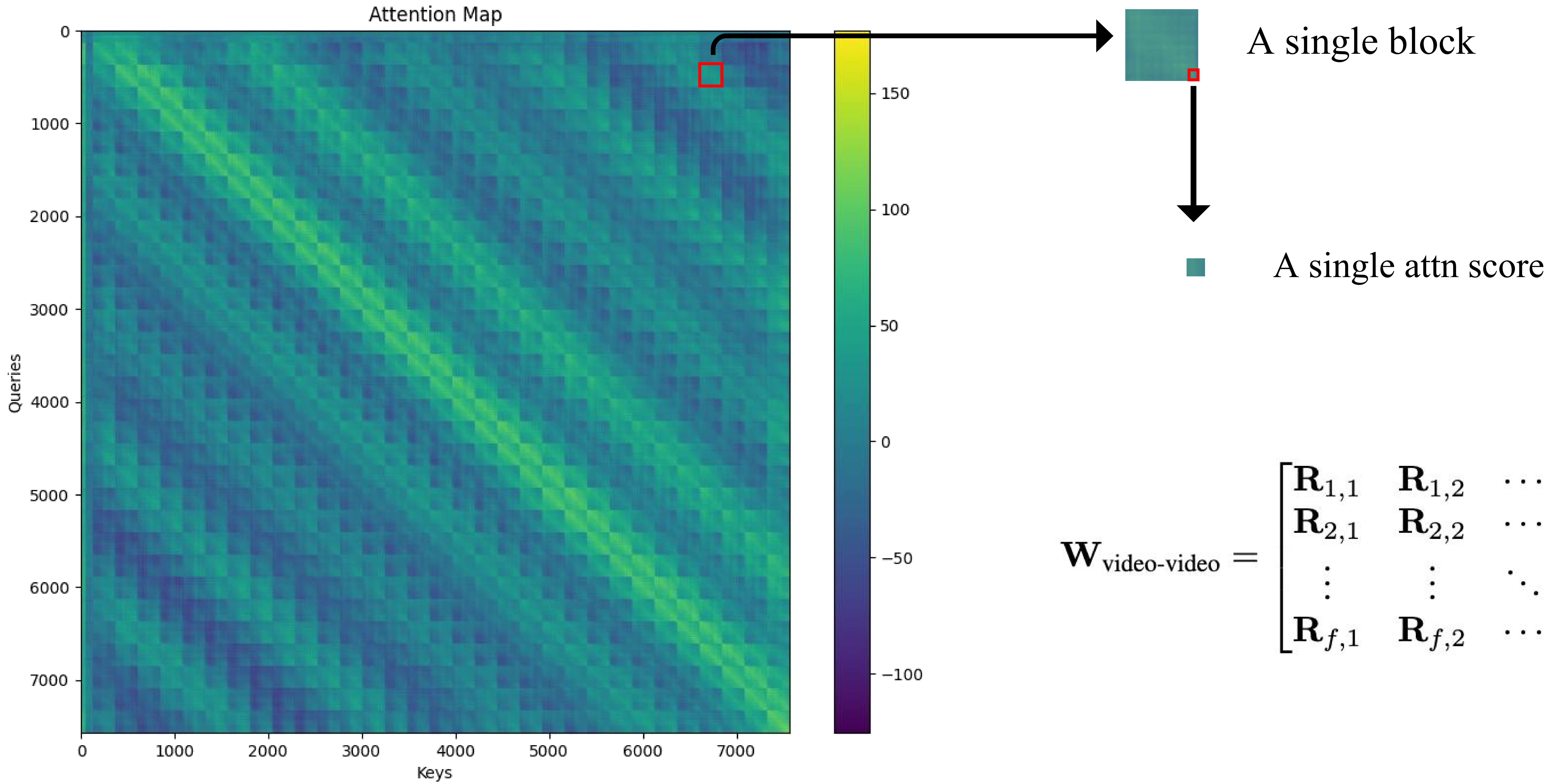
Overview



- Contrib. 1: Identify blockified pattern exhibits stronger expressive capability
- Contrib. 2: Cache LSE for sparsity index online searching
- Contrib. 3: Use different sparsity for different heads (adaptive during timesteps)

Contribution 1

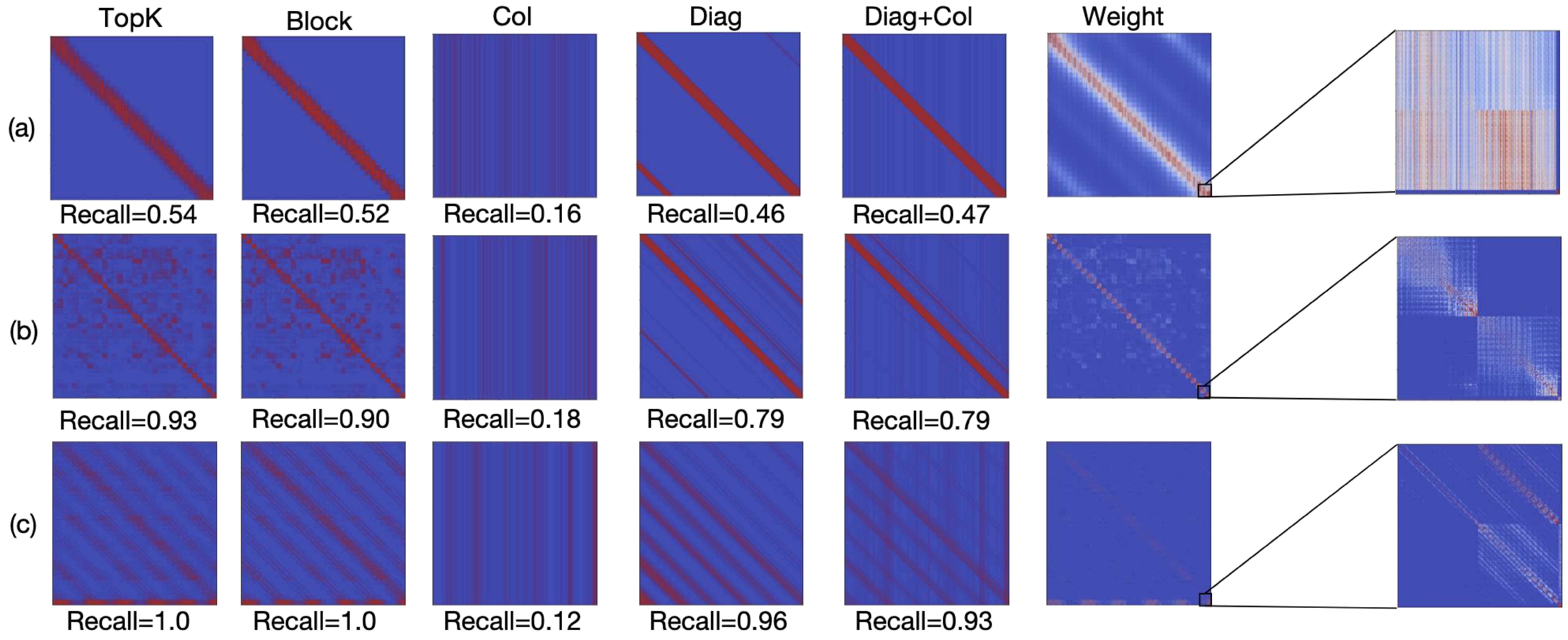
Blockified sparse pattern exceeds its contiguous counterpart



$$\mathbf{W}_{\text{video-video}} = \begin{bmatrix} \mathbf{R}_{1,1} & \mathbf{R}_{1,2} & \cdots & \mathbf{R}_{1,f} \\ \mathbf{R}_{2,1} & \mathbf{R}_{2,2} & \cdots & \mathbf{R}_{2,f} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{f,1} & \mathbf{R}_{f,2} & \cdots & \mathbf{R}_{f,f} \end{bmatrix}$$

Contribution 1

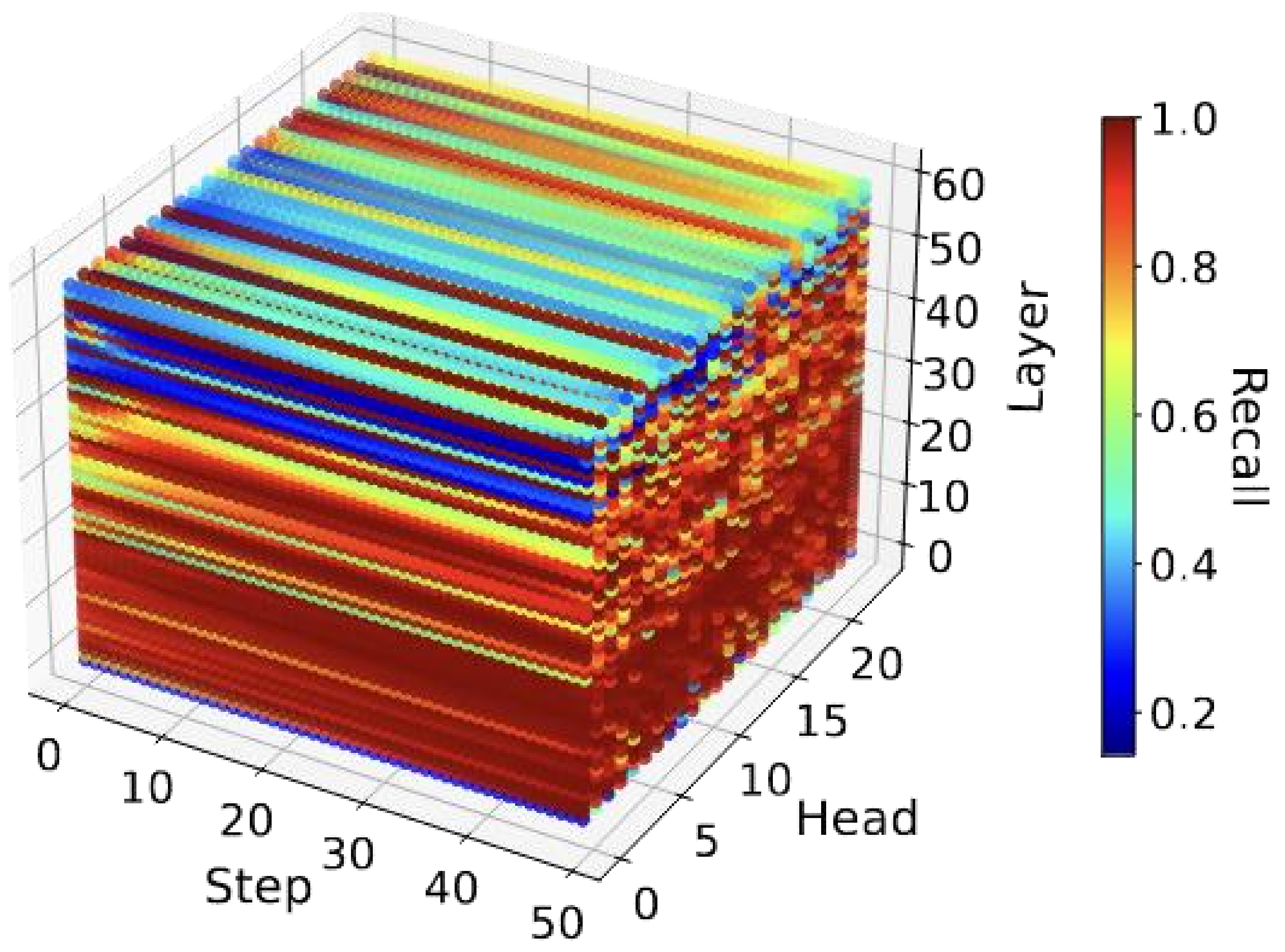
Blockified sparse pattern exceeds its contiguous counterpart



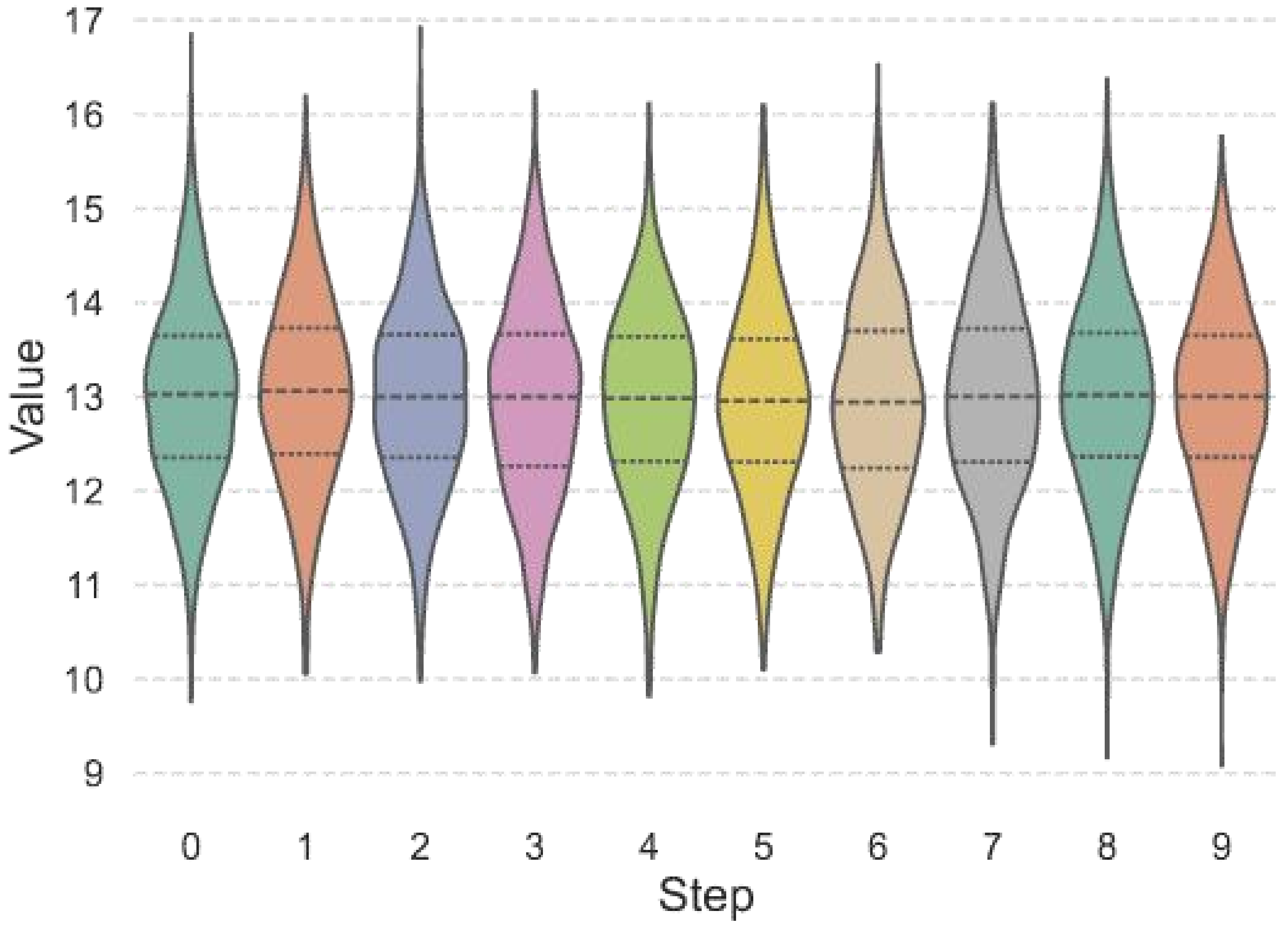
- **Recall**: sum(selected attn scores) / sum(attn scores in the attn map)
- Blockified pattern can **achieve better recall & cover more global pattern**

Contribution 2

Slow change of sparse patterns & LSE across timesteps



Recall in a single inference, using topK blocks

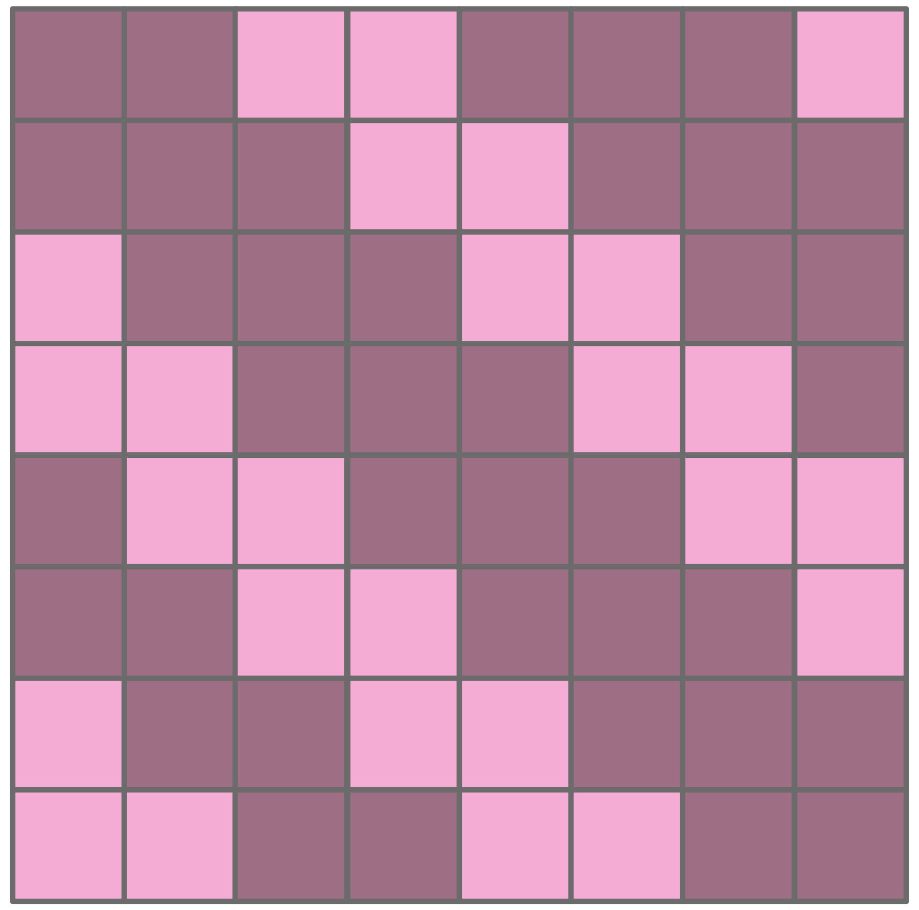


LSE distribution across timesteps

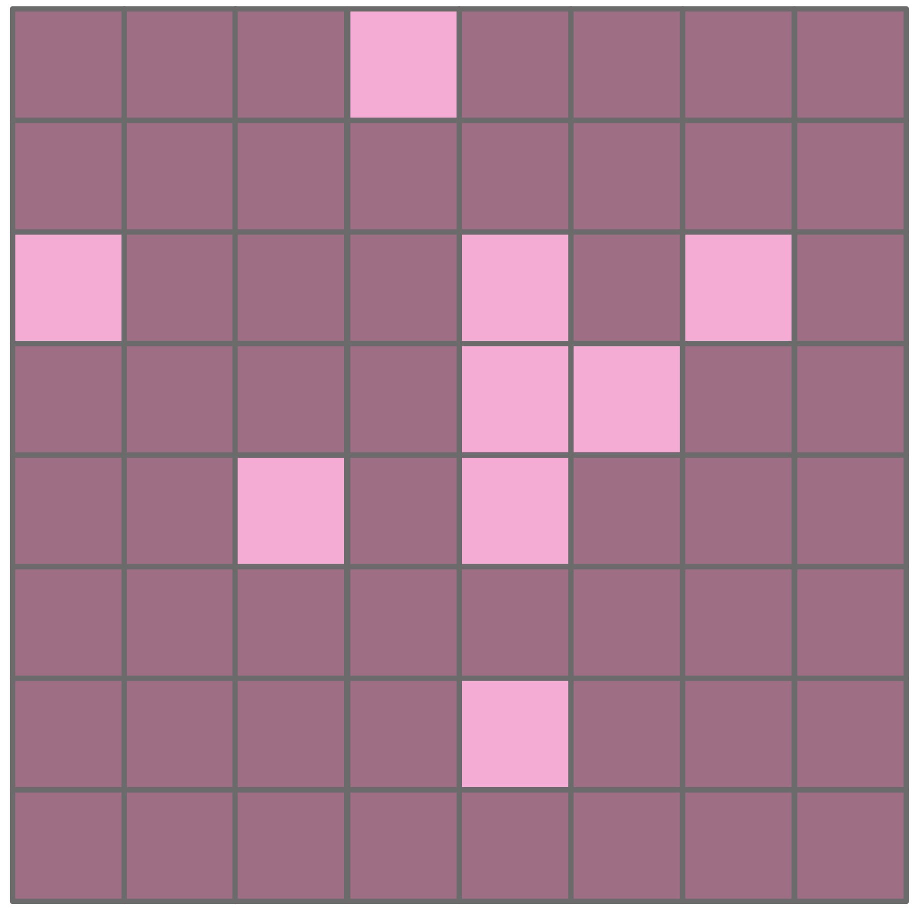
Contribution 3

Adaptive sparsity between heads & timesteps

- Different heads hold different sparsity characteristics
- A single uniform sparsity —— suboptimal Recall
- Distinct sparsity level for each head —— kernel load imbalance
- Adaptive sparsity: increase(decrease) the sparsity with the high(low) Recall

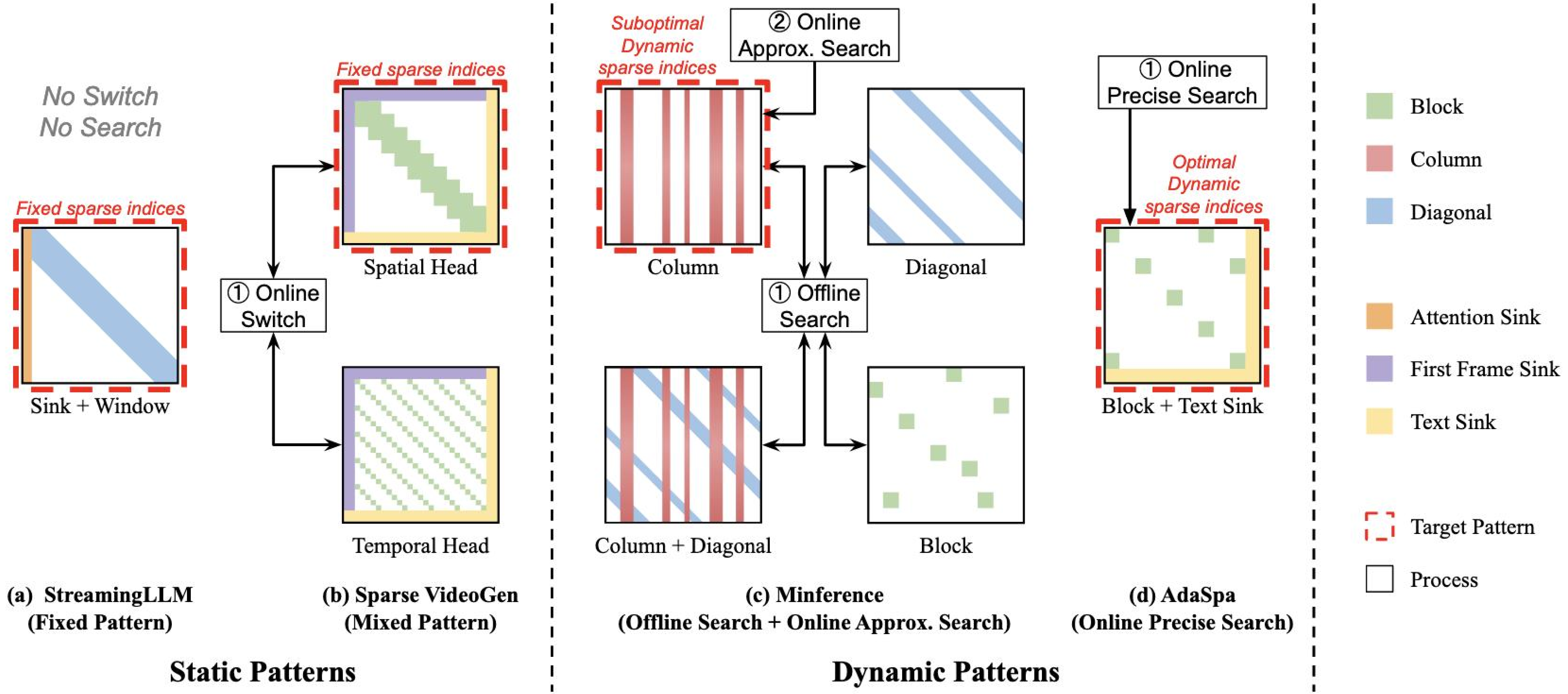


Low Sparsity



High Sparsity

Comparison with other methods



Thanks！