CS11-711 Advanced NLP

Long-Context Models

Graham Neubig

Carnegie Mellon University
Language Technologies Institute

https://phontron.com/class/anlp-fall2024/

How Long are Sequences?

- One sentence: ~20 tokens
- One document: 100-10k tokens
- One book: 50k-300k tokens
- One video: 1.5k-1M tokens (~300/sec)
- One codebase: 20k-1B tokens
- One genome: 3B nucleotides

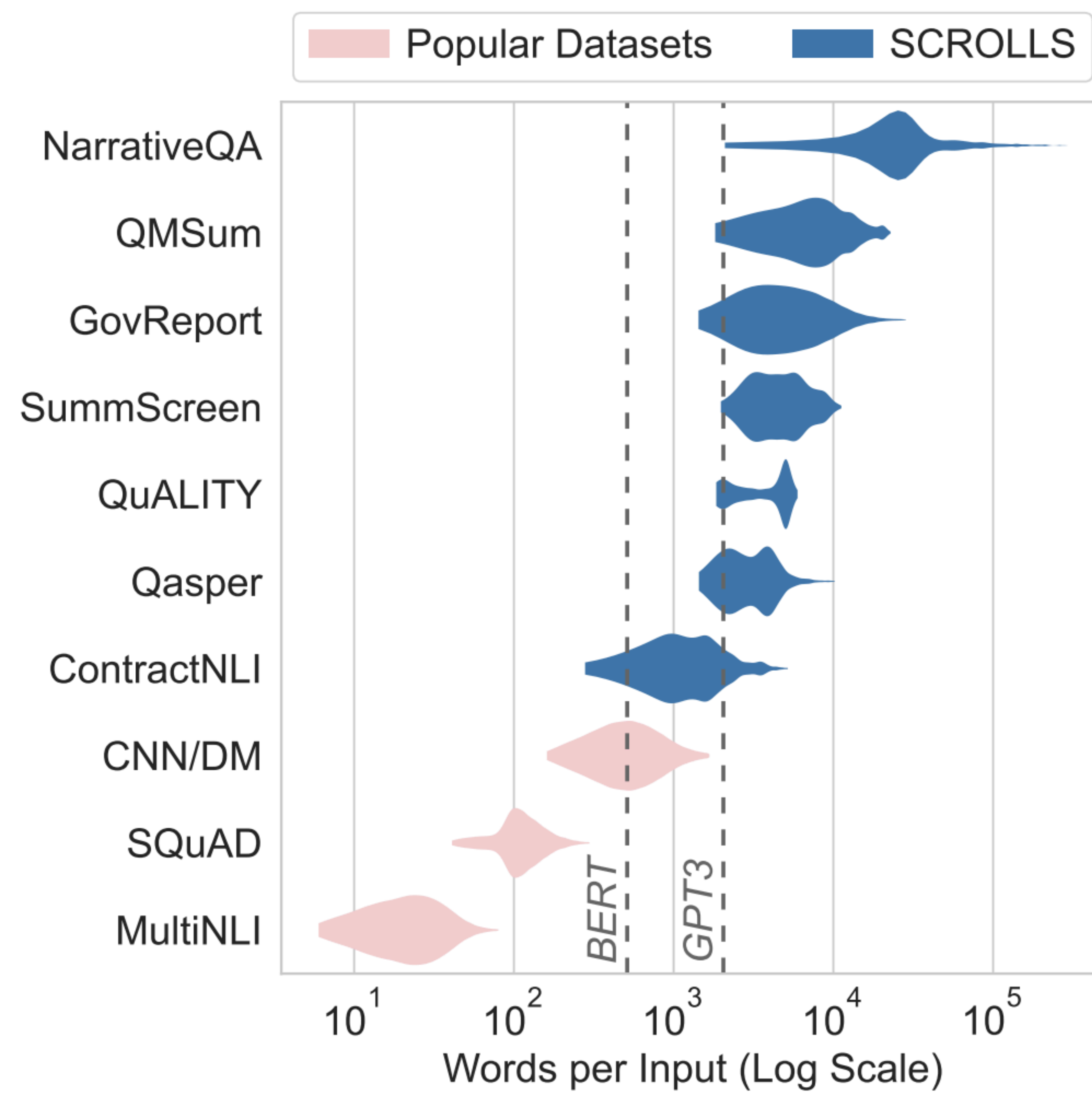
Why is Modeling Long Sequences Hard?

- **Memory Complexity:** Transformer models scale quadratically in memory
- **Compute Complexity:** Transformer models scale quadratically in computation
- **Training:** Data is lacking, training signal is weak, training on long sequences is costly

Long-context Use Cases and Evaluation

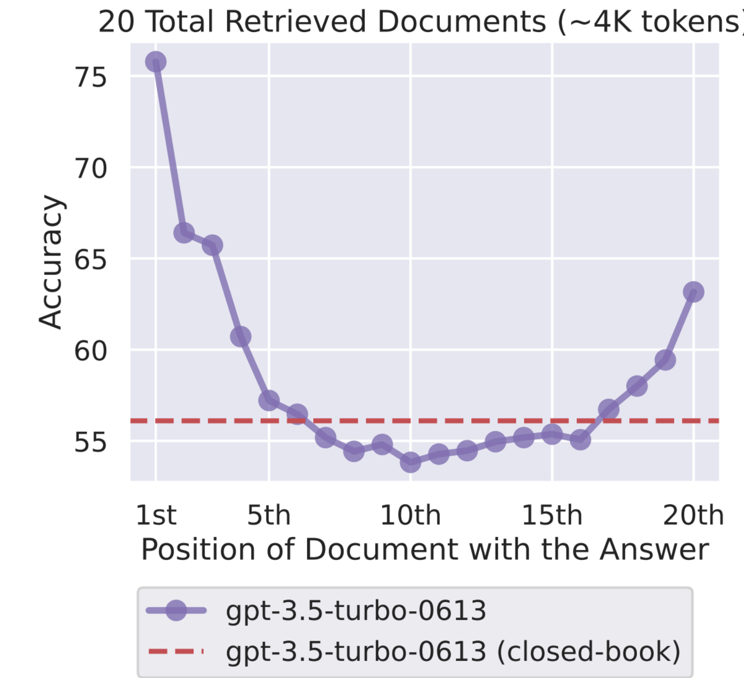
Benchmarks for Long-context Models

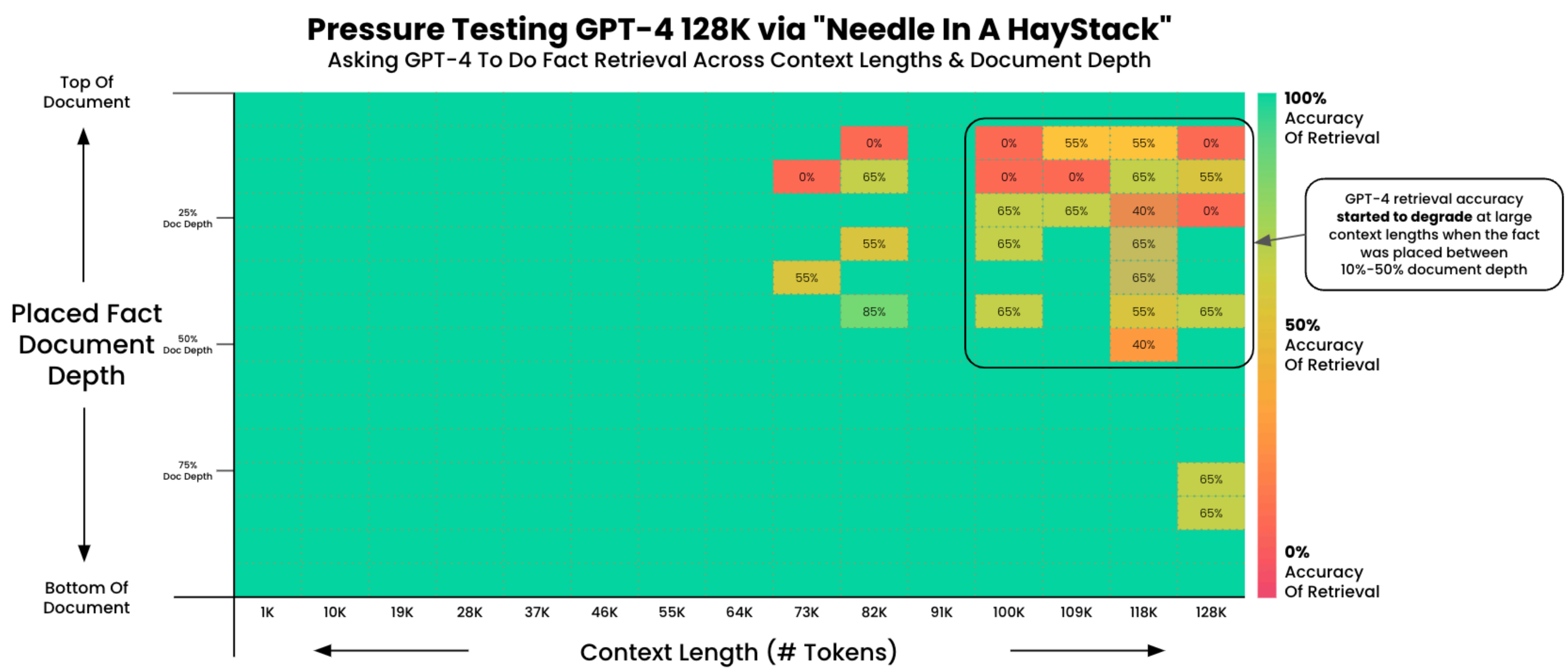
- **Long Range Arena:** Composite benchmark containing mostly non-NLP tasks (Tay et al. 2020)
- **SCROLLS:** Benchmark containing long-context summarization, QA, etc. (Shaham et al. 2022)



Targeted Analysis Tools

- “lost-in-the-middle” (Liu et al. 2023) demonstrates that models pay less attention to things in middle context
- “needle in a haystack” tests (Kamradt 2023) test across document length/position
- RULER (Hsieh et al. 2024) compiles a number of different NIAH tasks

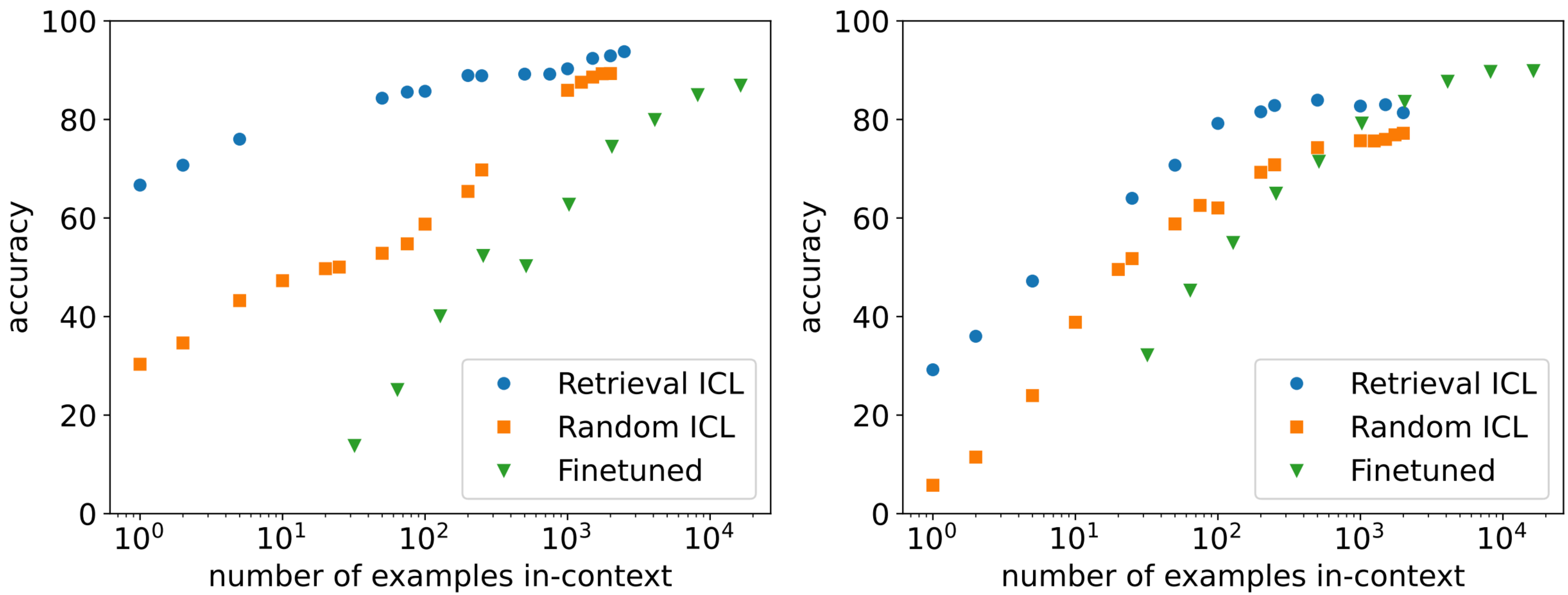




Long-context In-context Learning

(Bertsch et al. 2024)

- Can we provide lots of examples to long-context models and improve accuracy through ICL?

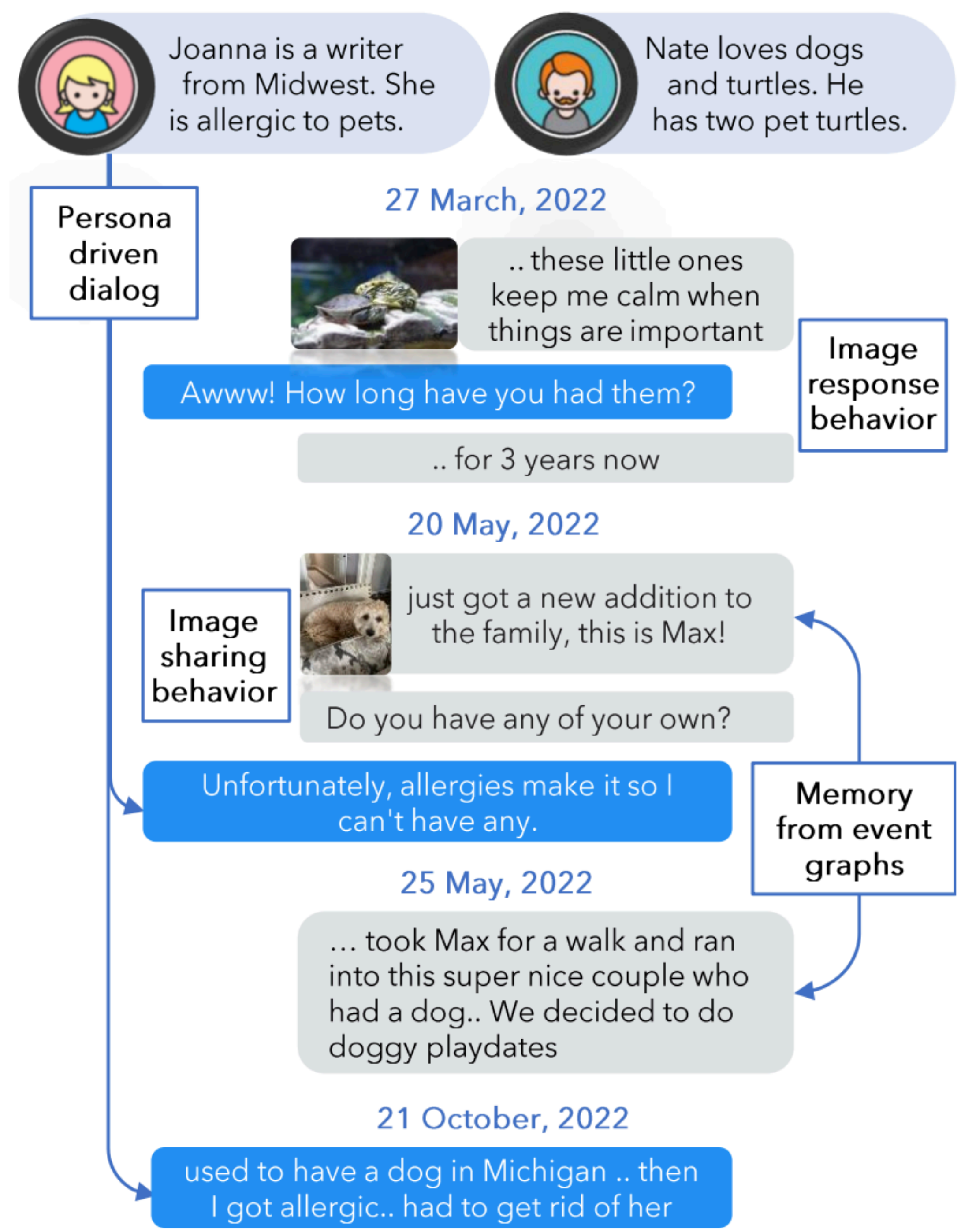


(a) Clinic-150 (b) Trecfine

- When many in-context examples are provided, it can be better than fine-tuning!

Long-context Dialog

- Chatbots that maintain long-term conversational context
- e.g. Locomo corpus (Maharana et al. 2024)
- Evaluate w/ question answering, summarization, response generation



Tackling Complexity:
Memory-efficient Computation

Vanilla Attention Complexity

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

$$\text{Attention}(Q, K, V) = AV$$

Time: $O(bs^2d)$ for QK^T
(but fast on GPU)

Memory: $\boxed{O(bs^2)}$ for all ops

Time: $O(bs^2d)$ for AV
(but fast on GPU)

Memory: $O(bsd)$

b: batch size, s: sequence length, d: dimension

Multi-head Attention Complexity

- Multi-head attention splits attention heads
- No effect on time complexity, but effect on memory

Time: $O(bs^2d)$ for QK^T
(but fast on GPU)

Time: $O(bs^2d)$ for AV
(but fast on GPU)

Memory: $\boxed{O(bs^2h)}$ for all ops

Memory: $O(bsd)$

b: batch size, s: sequence length, d: dimension, h: heads

Memory-efficient Computation

(Jang 2019, Rabe and Staats 2021)

- Insight: you don't need to materialize s^2 attention
- Calculate softmax numerator times values, and softmax denominator left-to-right

*softmax numerator * V*

$$V^* = \exp\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Memory: *O(bsd)*

softmax denominator

$$S^* = \mathrm{sum}\left(\exp\left(\frac{QK^T}{\sqrt{d_k}}\right)\right)$$

Memory: *O(bsh)*

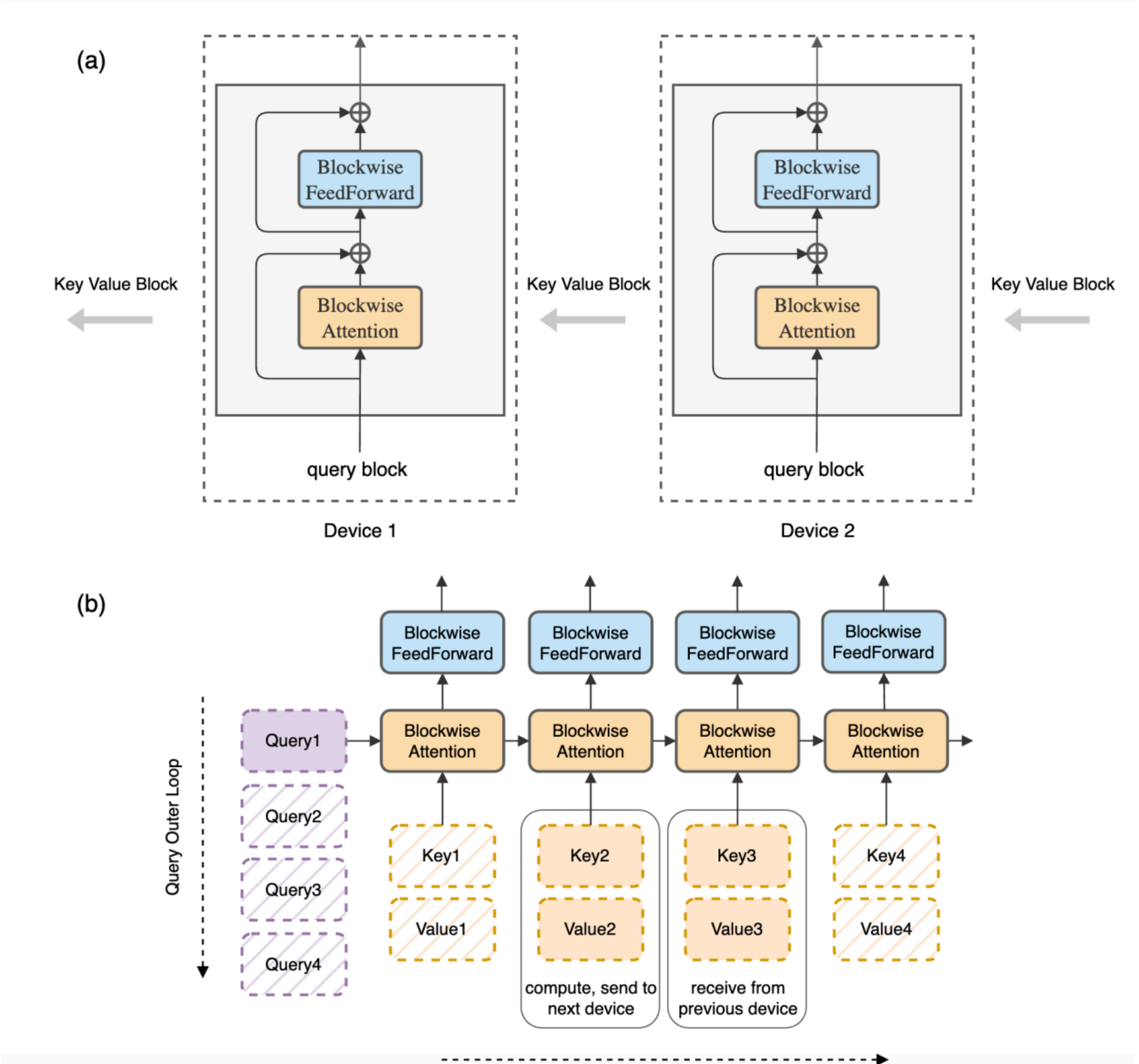
attention

$$\mathrm{Attention}(Q, K, V) = V^*/S^*$$

Memory: *O(bsd)*

Ring Attention (Liu et al. 2023)

- Further distribute storage/ incremental computation across multiple devices



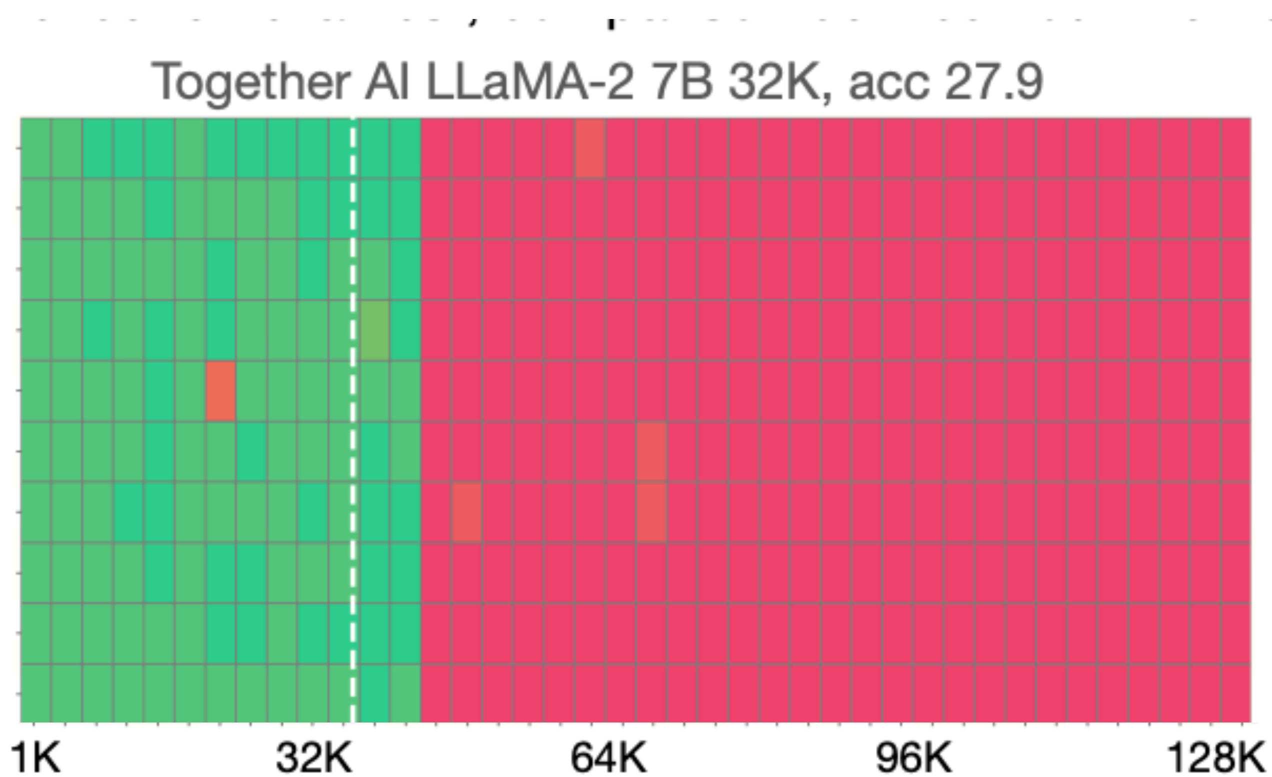
Extrapolation of Short-Context Models

Trained Models Fail to Extrapolate

- Most transformer models are trained on shorter sequences (4k)
  - If a document is longer than the limit, truncate or chunk
- This poses problems for positional encodings:
  - **Learned absolute encodings:** impossible to extrapolate
  - **Fixed absolute encodings:** move models out of distribution, very bad
  - **Relative encodings:** should extrapolate better in theory, but not really in practice

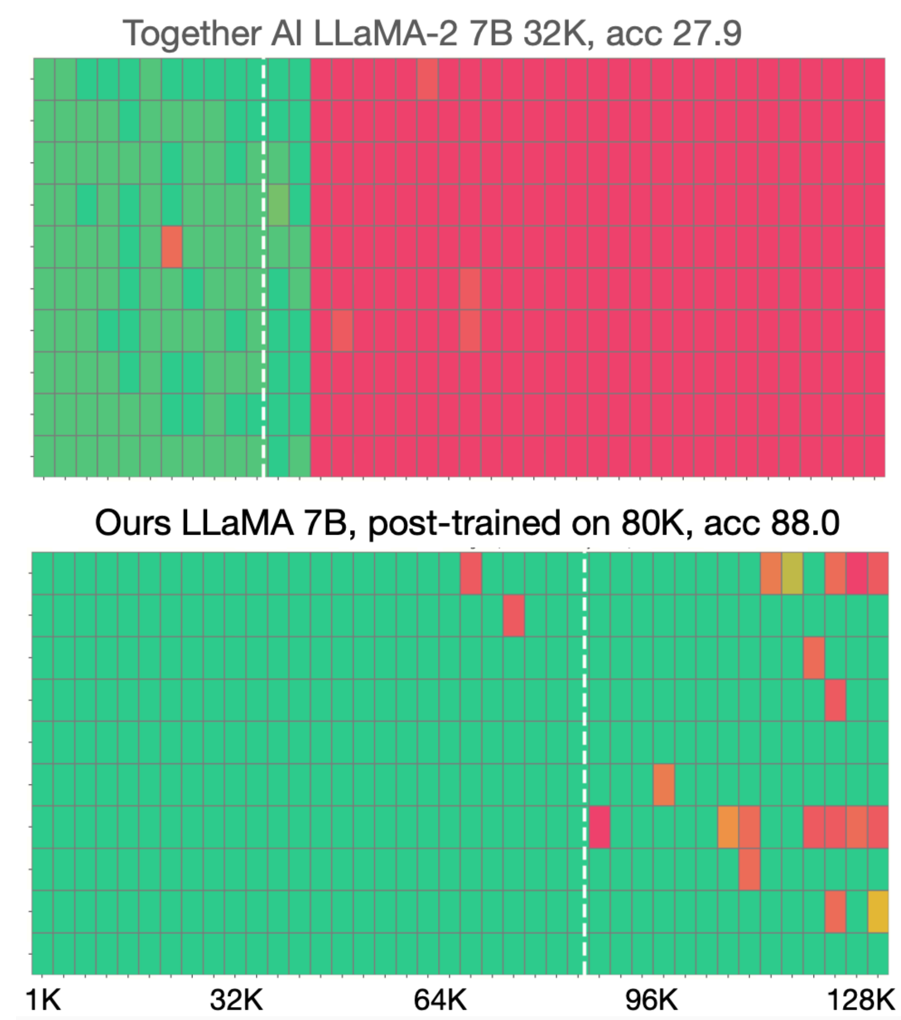
An Example of Failed Extrapolation (Fu et al. 2024)

- Llama-2 w/ 32k context (RoPE) can answer questions about sequences up to about 40k, but not beyond



Training w/ Long Context
(Fu et al. 2024)

- *Simple solution:* continually train on longer documents
- *Problem:* there aren't many long documents
  - *Solution:* upsample the longer documents
- *Problem:* upsampling favors certain domains such as books and GitHub
  - *Solution:* maintain domain mixture, but upsample long docs in each domain



RoPE Scaling

(see Lu et al. 2024)

- RoPE has a parameter adjusting the period

$$\mathbf{R}(\boldsymbol{\theta}, i) = \begin{pmatrix} \cos i\theta_1 & -\sin i\theta_1 & \cdots & 0 & 0 \\ \sin i\theta_1 & \cos i\theta_1 & \cdots & 0 & 0 \\ \vdots & & & & \\ 0 & 0 & \cdots & \cos i\theta_{\frac{d_k}{2}} & -\sin i\theta_{\frac{d_k}{2}} \\ 0 & 0 & \cdots & \sin i\theta_{\frac{d_k}{2}} & \cos i\theta_{\frac{d_k}{2}} \end{pmatrix}$$

- typically $\theta_j = b^{-\frac{2j}{d_k}}$ with b=10000

- **Position interpolation:** Multiply θ by a constant scaling factor (e.g. C_{short}/C_{long})

- **Neural tangent kernel:** Scale low-frequency components, but maintain high-frequency components

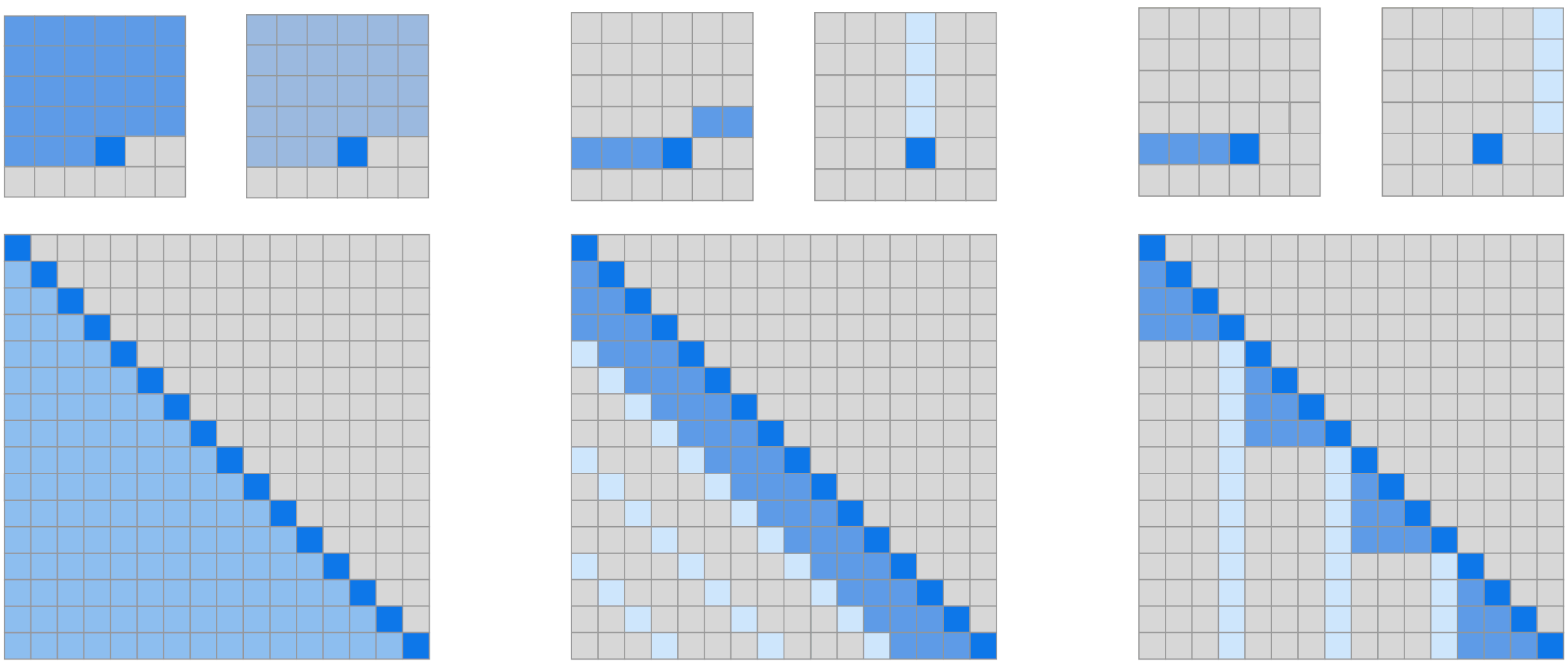
Tackling Complexity:
Alternative Transformer Architectures

Tackling Transformer Complexity

- Sparse Attention
- Sliding Window Attention
- Compression
- Low-rank Approximation

Sparse Transformers
(Child et al. 2019)

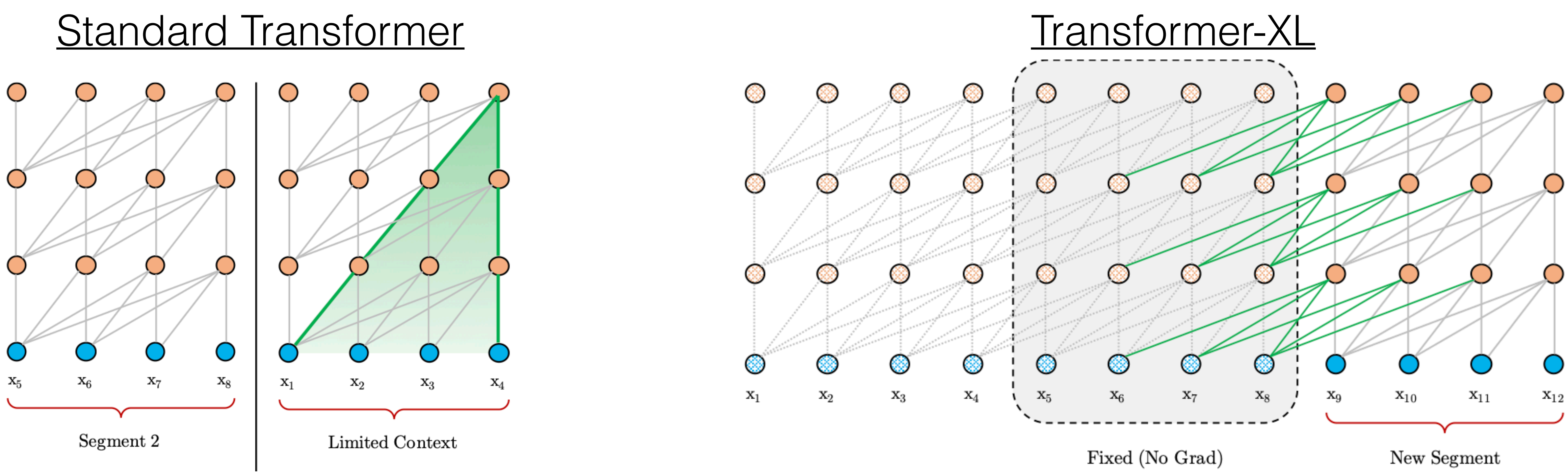
- Add "stride", only attending to every *n* previous states



(a) Transformer (b) Sparse Transformer (strided) (c) Sparse Transformer (fixed)

Truncated BPTT+Transformer

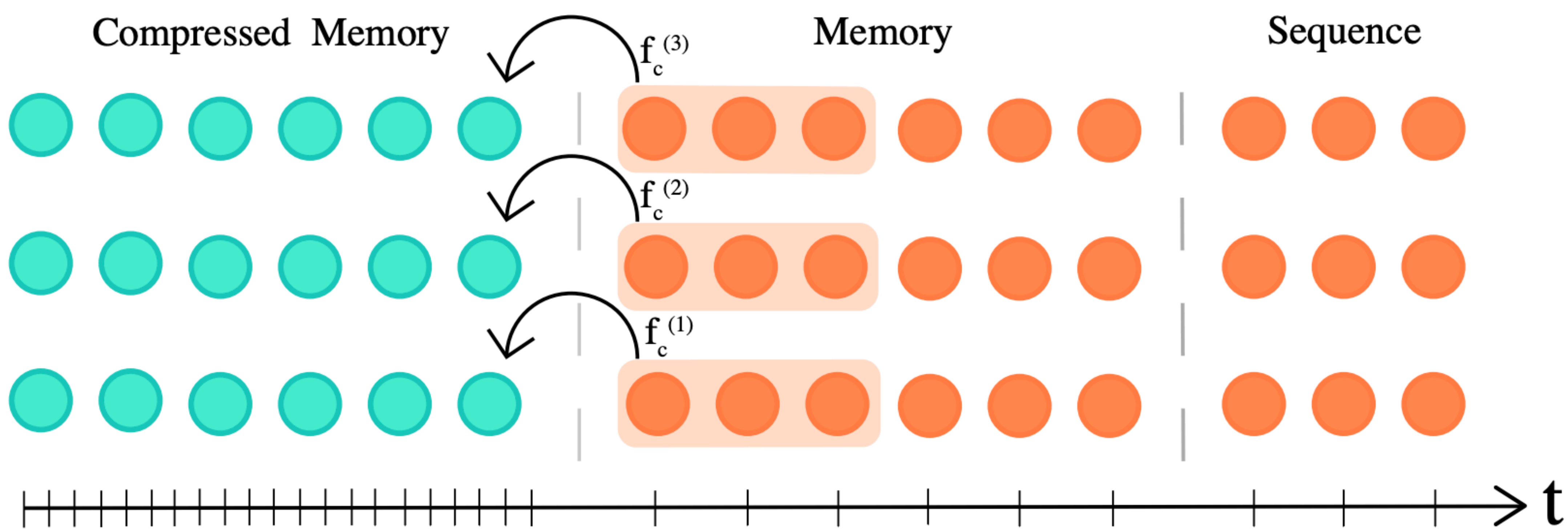
- Transformer-XL (Dai et al. 2019) attends to fixed **vectors** from the previous sentence



- Like truncated backprop through time for RNNs; can use previous states, but not backprop into them
- See also Mistral's (Jiang et al. 2023) sliding window attention

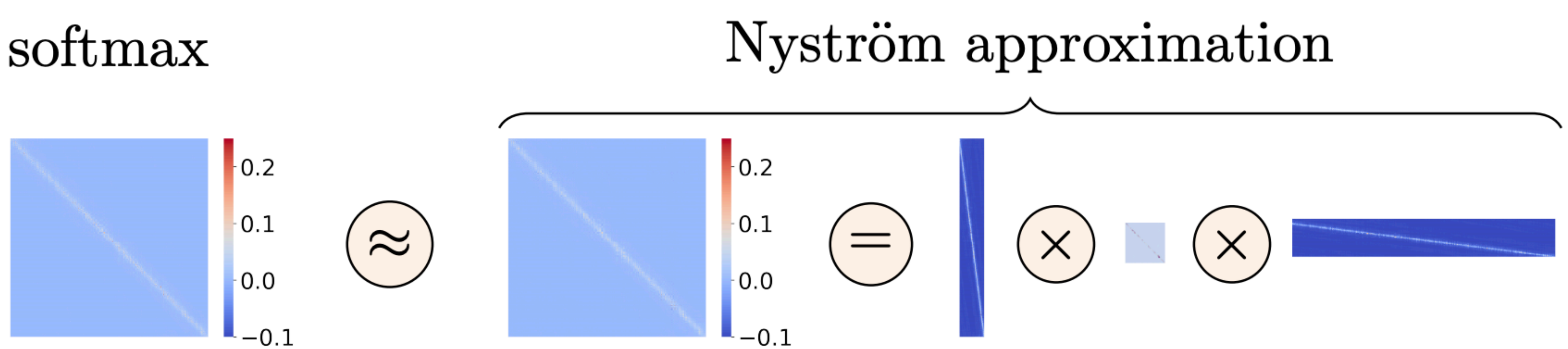
Compressing Previous States

- Add a "strided" compression step over previous states (Rae et al. 2019)



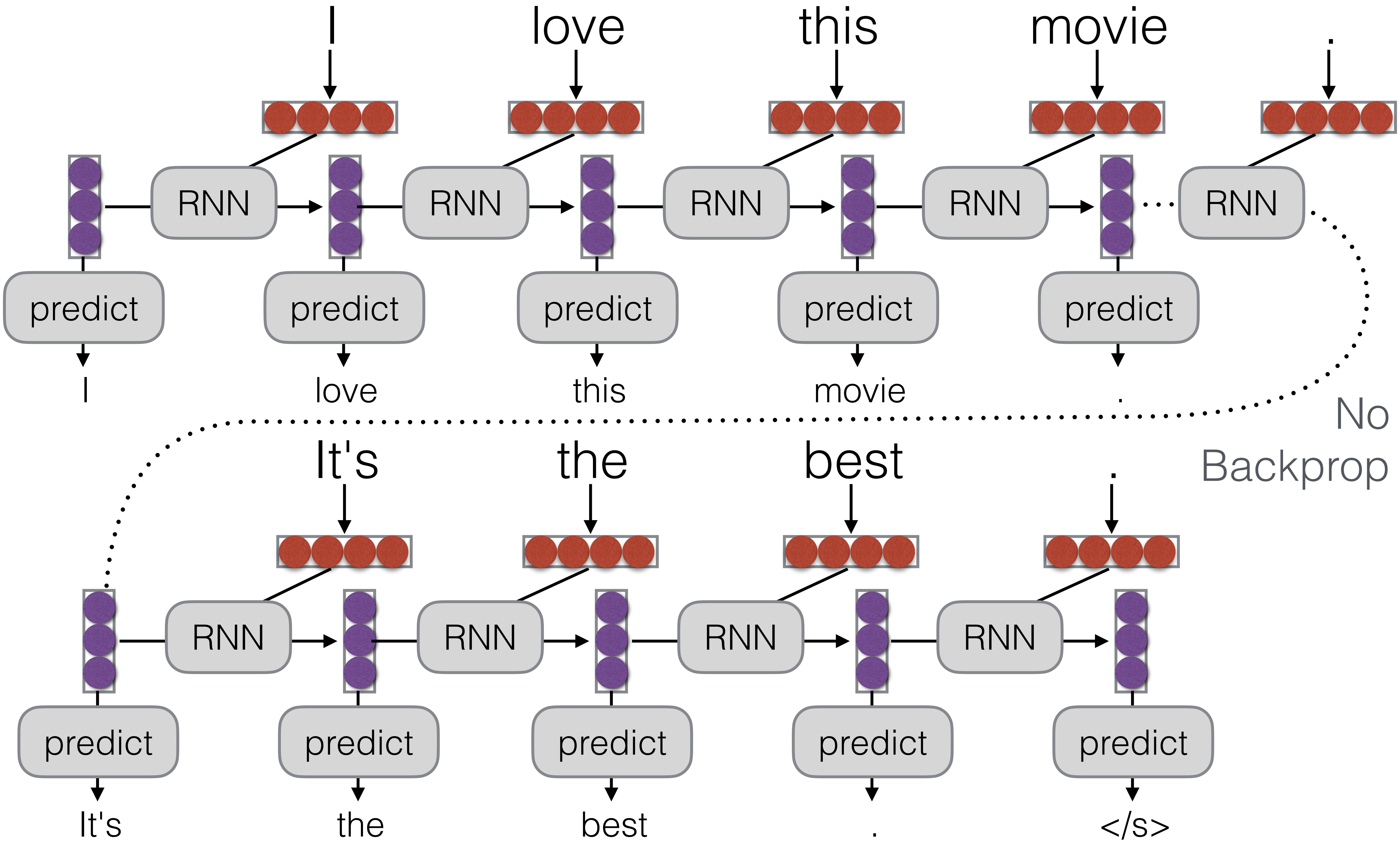
Low-rank Approximation

- Calculating the attention matrix is expensive, can it be predicted with a low-rank matrix?
- **Linformer:** Add low-rank linear projections into model (Wang et al. 2020)
- **Nystromformer:** Approximate using the Nystrom method, sampling "landmark" points (Xiong et al. 2021)



Tackling Complexity: Non-attentional Models

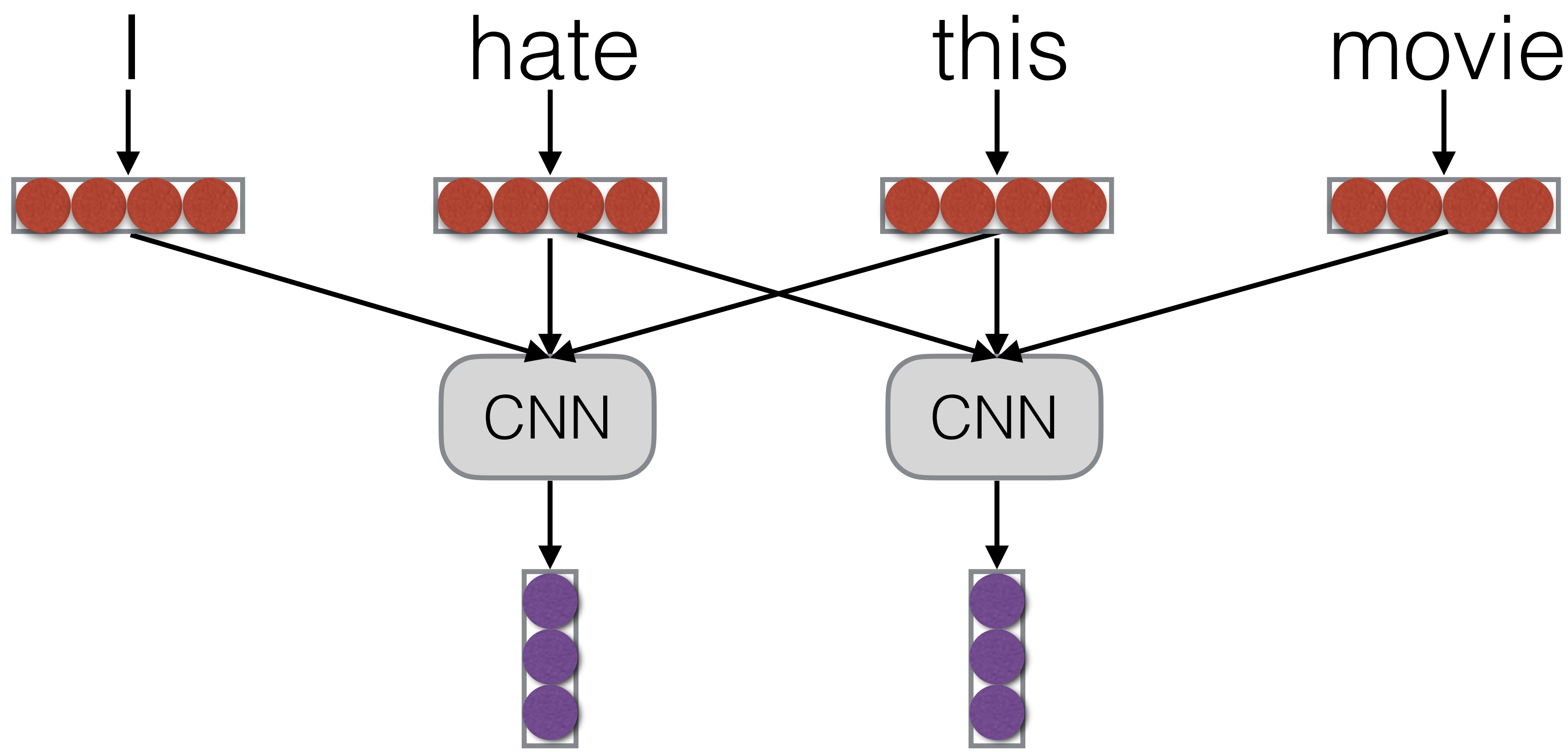
Reminder: RNNs



- Each RNN step depends on the previous - slow!

Convolution

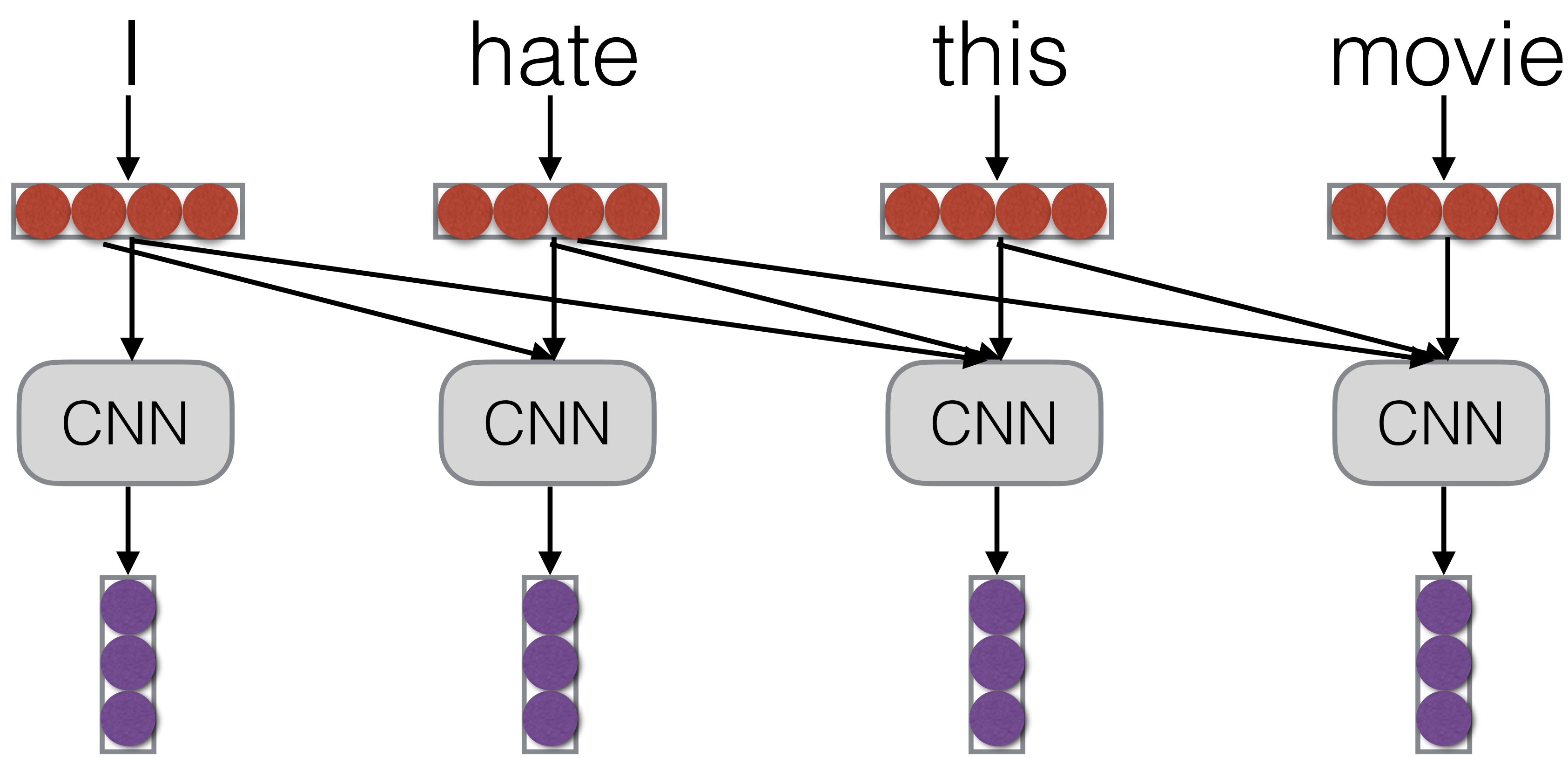
- Calculate based on local context



$$h_t = \mathrm{f}(W[x_{t-1}; x_t; x_{t+1}])$$

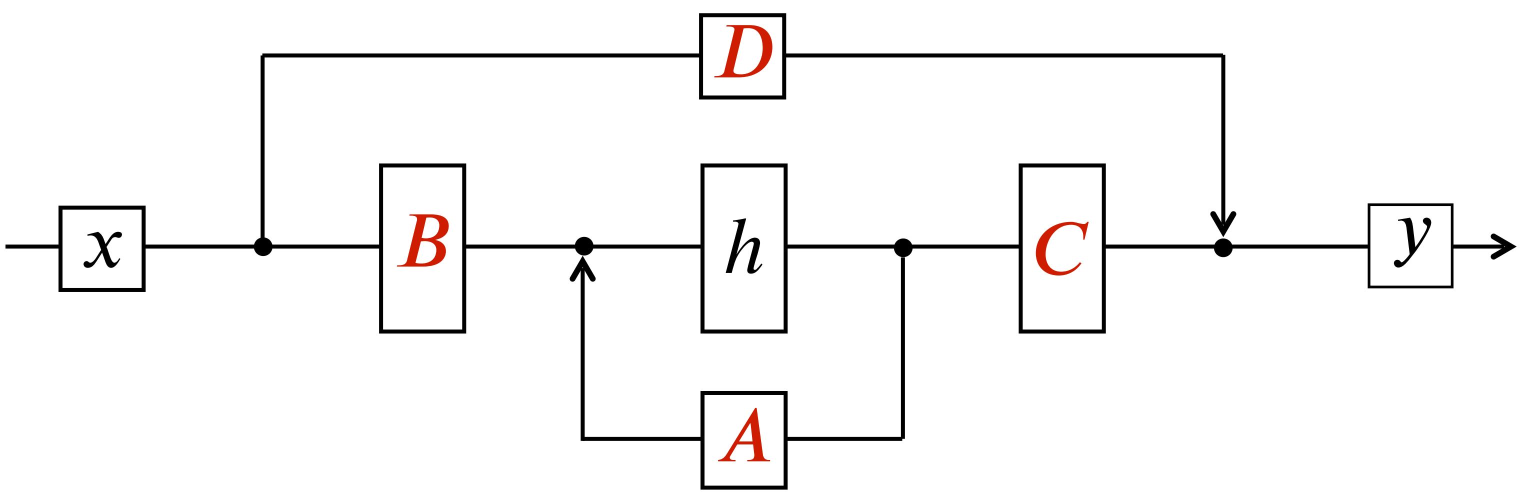
Convolution for Auto-regressive Models

- Functionally identical, just consider previous context



Structured State Space Models (Gu et al. 2021)

- Models that take a form like the following



$$h'(t) = \boldsymbol{A}h(t) + \boldsymbol{B}x(t)$$
$$y(t) = \boldsymbol{C}h(t) + \boldsymbol{D}x(t)$$

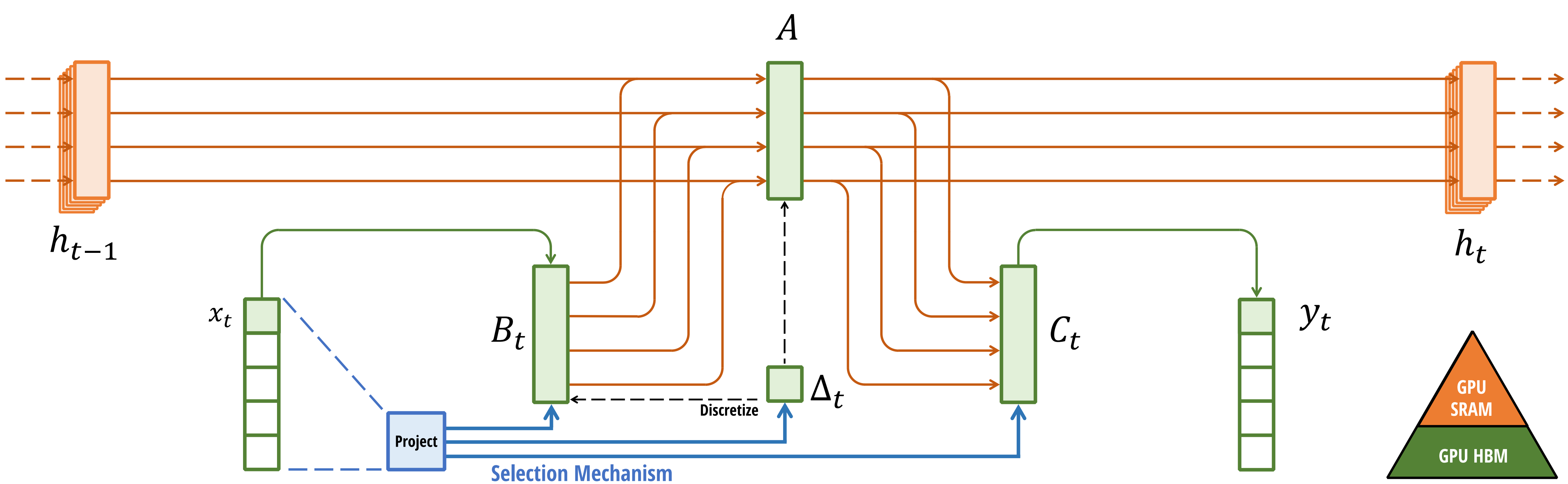
- Because there are no non-linearities, the current h/x as a function of previous states can be calculated in advance

Slide Credit: Albert Gu

Selective State Space Models - Mamba

(Gu and Dao 2023)

- To improve modeling power of state space models, condition parameters on current input



- Use efficient parts of GPU memory to handle expanded state

Questions?