Flash Attention from First Principles

Zihan Zheng, Apr 9, 2026

What I wish to cover / expected outcome

- Forward Pass & Backward Pass
  - A "good" schema
  - Precise AI of Flash Attention
  - Able to write FA in Triton without referencing anything
- All contents will be written on the board. The slides are my cheat sheets.

The Unfair Die

- An unfair die with faces $x_1 = 1$, $x_2 = 2$, …, $x_6 = 6$ and observed counts $c(x_i)$:

x_i	1	2	3	4	5	6
$c(x_i)$	200	300	100	150	150	100

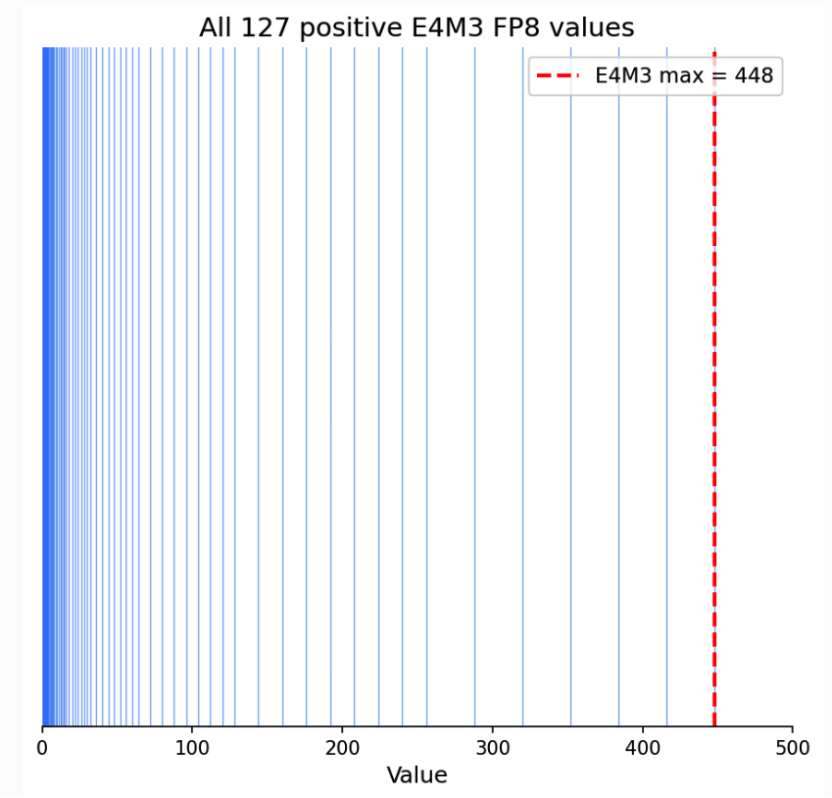
Question: What's the expected value E[X]?

$E[X] = \Sigma_i \, x_i \cdot P(X=x_i)$

wait … $P(X=x_i)$? $P(X= x_1)= (200) / (1000) = 0.2$

But what if … we live in a FP8 world?

Overflow!



Overflow? We Only Need a Ratio!

$$P(x_1) = \frac{c(x_1)}{\sum_j c(x_j)} = \frac{c(x_1)/m}{\sum_j c(x_j)/m}$$

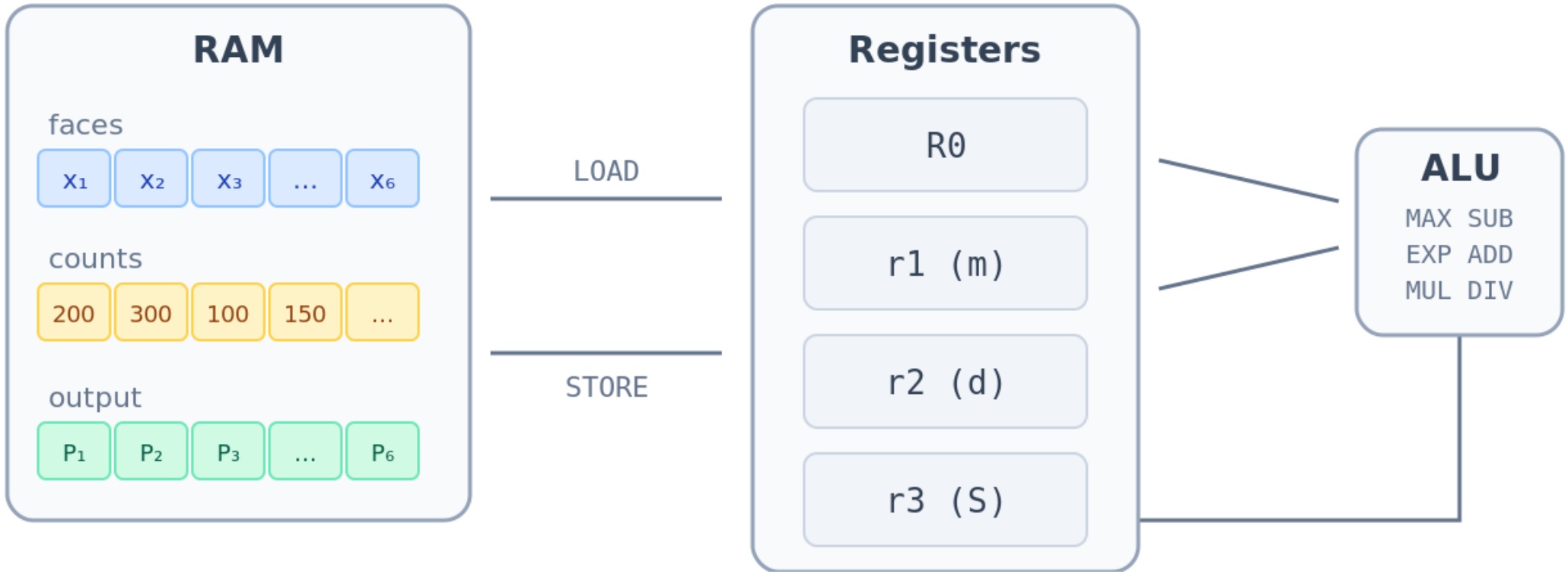
where $m = \max_j c(x_j)$.

$$\frac{200}{1000} = \frac{200/300}{\sum_j c(x_j)/300} = \frac{200/300}{200/300 + 300/300 + 100/300 + 150/300 + 150/300 + 100/300}$$

Every term ≤ 1. The sum ≈ 3.33. No overflow.

Think Like an Architect: Where Does Data Live?

1. Find $m = \max_j c(x_j)$
2. Compute denominator: $d = \sum_j c(x_j)/m$
3. For each i: $P(X = x_i) = \frac{c(x_i)/m}{d}$



Eyes on the Prize: Streaming Reduction

1. Find $m = \max_j c(x_j)$
2. Compute denominator: $d = \sum_j c(x_j)/m$
3. For each i: $P(X = x_i) = \frac{c(x_i)/m}{d}$

$$E[X] = \sum_i x_i \, P(X = x_i) = \sum_i x_i \frac{c(x_i)/m}{\sum_j c(x_j)/m}$$

$$= \frac{\sum_i x_i \, c(x_i)/m}{\sum_j c(x_j)/m}$$

- The numerator and denominator are **independent sums over the same index.**
- Each time we load one $c(x_i)$,we update both sums simultaneously.
- $P(x_i)$ is never computed. It never touches RAM.

Rethinking m = max $c(x_i)$

- Why do we need m?
- What are our choices for m?
  - max $c(x_i)$— actual max (best precision, costs one pass)
  - 448 — FP8 dynamic range max (loses precision, saves a pass)
  - Precomputed global statistics (chicken and egg?)
- Can we trade compute for less data movement?
  - Recall: activation checkpointing trades compute for memory
  - RAM and compute are a **trade-off decision**
- "max" is (conceptually) a streaming reduction itself!

Rethinking m = max $c(x_i)$, continued

With global max m:

$$d_i = \sum_{j=1}^{i} \frac{c(x_j)}{m} \qquad n_i = \sum_{j=1}^{i} \frac{x_j\, c(x_j)}{m} \qquad E_i[X] = \frac{n_i}{d_i}$$

At step $i+1$:

$$d_{i+1} = d_i + \frac{c(x_{i+1})}{m} \qquad n_{i+1} = n_i + \frac{x_{i+1}\, c(x_{i+1})}{m} \qquad E_{i+1}[X] = \frac{n_{i+1}}{d_{i+1}}$$

At step i:

$$d_i = \sum_{j=1}^{i} \frac{c(x_j)}{m_i} \qquad n_i = \sum_{j=1}^{i} \frac{x_j\, c(x_j)}{m_i} \qquad E_i[X] = \frac{n_i}{d_i}$$

At step $i+1$, with $m_{i+1} = \max(m_i,\, c(x_{i+1}))$:

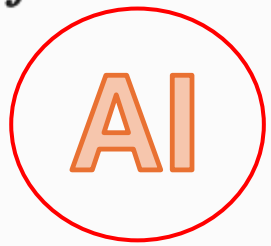
$$d_{i+1} = d_i \cdot \frac{m_i}{m_{i+1}} + \frac{c(x_{i+1})}{m_{i+1}} \qquad n_{i+1} = n_i \cdot \frac{m_i}{m_{i+1}} + \frac{x_{i+1}\, c(x_{i+1})}{m_{i+1}} \qquad E_{i+1}[X] = \frac{n_{i+1}}{d_{i+1}}$$

Flash Attention (1/3)

Proudly generated by Opus 4.6(1M) with max thinking effort:

$$E[X] = \frac{\sum_i x_i\, c(x_i)/m}{\sum_j c(x_j)/m} \quad \longleftrightarrow \quad \mathrm{softmax}(qK^T)\cdot V = \frac{\sum_i v_i\, e^{q\cdot k_i^T}/m}{\sum_j e^{q\cdot k_j^T}/m}$$

$$x_i \to v_i \qquad c(x_i) \to e^{q\cdot k_i^T} \qquad m \to \max_j\, q\cdot k_j^T$$



Generated by me:

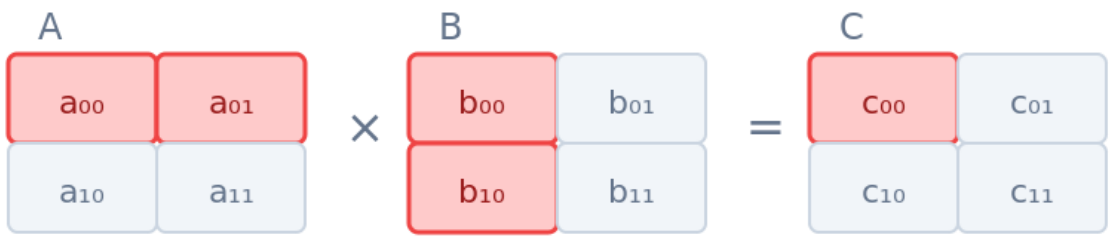
$$E[X] = \frac{\sum_i x_i\, c(x_i)/m}{\sum_j c(x_j)/m} \quad \longleftrightarrow \quad \mathrm{softmax}(qK^T)\cdot V = \frac{\sum_i v_i\, e^{q\cdot k_i^T - m}}{\sum_j e^{q\cdot k_j^T - m}}$$

$$x_i \to v_i \qquad c(x_i)/m \to e^{q\cdot k_i^T - m} \qquad m \to \max_j\, q\cdot k_j^T$$

Elements are tiles!

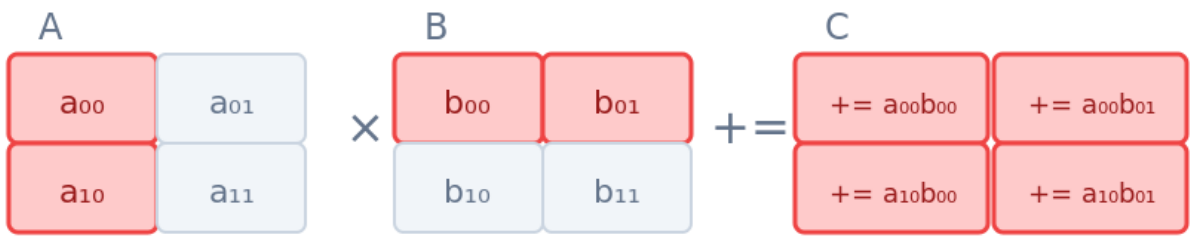
Loop ordering?

Inner Product



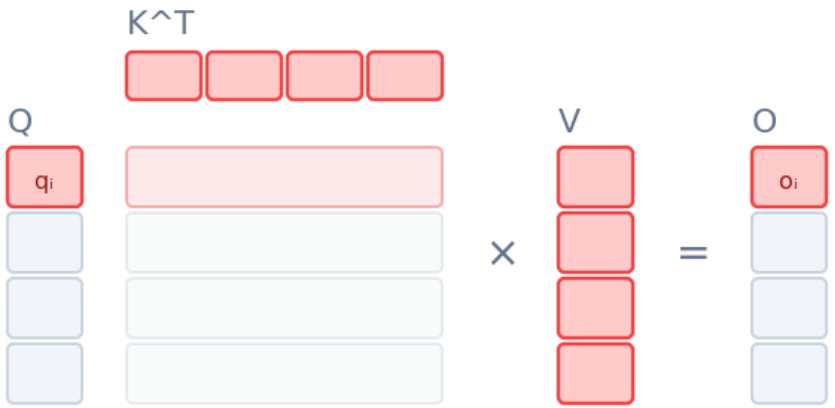
Load row of A + column of B → one element of C

Outer Product



Load column of A + row of B → all elements of C updated

Inner Product Style



Fix one Q row, stream all K, V → one output row

Outer Product Style

Fix one K, V row, update all Q rows → all outputs +=

Backward Pass: Overview

$$O = PV = \mathrm{Softmax}(S)\,V = \mathrm{Softmax}(QK^T)\,V.$$

Dimensions: O: $M \times d$, S: $M \times N$, V: $N \times d$.

$$O_{ij} = \sum_k P_{ik}\,V_{kj}.$$

M: $\mathrm{len}(Q)$. N: $\mathrm{len}(K)$. Q: $M \times d$.

Loss function: L: scalar. Want: $\frac{\partial L}{\partial Q}$ ($M \times d$), $\frac{\partial L}{\partial K}$ ($N \times d$), $\frac{\partial L}{\partial V}$ ($N \times d$).

$\frac{\partial L}{\partial O}$: $M \times d$, known.

$\mathrm{Attn} \to O \to \mathrm{FFN} \to \mathrm{Attn}$

$O = PV$.

$$\frac{\partial L}{\partial V} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial V} \quad \text{4D tensor.}$$

Trick 1: the δ_{lj} notation.

Let's look at one element:

$$\frac{\partial L}{\partial V_{ij}} = \sum_k \sum_l \frac{\partial L}{\partial O_{kl}} \frac{\partial O_{kl}}{\partial V_{ij}}$$

$$\frac{\partial V_{ml}}{\partial V_{ij}} = \delta_{im}\,\delta_{lj}$$

$$= \sum_k \sum_l \frac{\partial L}{\partial O_{kl}} \frac{\partial \sum_m P_{km} V_{ml}}{\partial V_{ij}}$$

$$= \sum_m \sum_k \sum_l \frac{\partial L}{\partial O_{kl}} \frac{\partial P_{km} V_{ml}}{\partial V_{ij}}$$

$$= \sum_{m,k,l} \frac{\partial L}{\partial O_{kl}} P_{km} \frac{\partial V_{ml}}{\partial V_{ij}} = \sum_m \sum_k \sum_l \frac{\partial L}{\partial O_{kl}} P_{km}\,\delta_{im}\,\delta_{lj}$$

$$= \sum_k \frac{\partial L}{\partial O_{kj}} P_{ki} = \sum_k P^T_{ik} \frac{\partial L}{\partial O_{kj}} = \left(P^T \frac{\partial L}{\partial O} \right)_{ij}$$

Similarly:

$$\frac{\partial L}{\partial P_{ij}} = \sum_k \sum_l \frac{\partial L}{\partial O_{kl}} \frac{\partial O_{kl}}{\partial P_{ij}} = \sum_m \sum_k \sum_l \frac{\partial L}{\partial O_{kl}} \frac{\partial \sum_m P_{km} V_{ml}}{\partial P_{ij}}$$

$$= \sum_m \sum_k \sum_l \frac{\partial L}{\partial O_{kl}} V_{ml}\, \delta_{ki}\, \delta_{mj} = \sum_l \frac{\partial L}{\partial O_{il}} V^T_{jl}$$

$$= \sum_l \frac{\partial L}{\partial O_{il}} V^T_{lj} = \left(\frac{\partial L}{\partial O} V^T \right)_{ij}.$$

$$\boxed{\frac{\partial L}{\partial P} = \frac{\partial L}{\partial O} V^T. \qquad \frac{\partial L}{\partial V} = P^T \frac{\partial L}{\partial O}. \qquad \frac{\partial L}{\partial Q} = \qquad \frac{\partial L}{\partial K} =}$$

The Final Boss!

$$\frac{\partial L}{\partial S_{ij}} = \sum_{k,l} \frac{\partial L}{\partial P_{kl}} \frac{\partial P_{kl}}{\partial S_{ij}} \delta_{ik} = \sum_{l} \frac{\partial L}{\partial P_{il}} \frac{\partial P_{il}}{\partial S_{ij}}$$

$$\frac{\partial P_{kl}}{\partial S_{ij}} \delta_{ik}$$

$$\frac{\partial P_{il}}{\partial S_{ij}} = \frac{\partial}{\partial S_{ij}} \frac{e^{S_{il}}}{\sum_k e^{S_{ik}}}$$

$$l = j: \quad \frac{e^{S_{ij}}(\Sigma) - e^{S_{ij}} e^{S_{ij}}}{(\Sigma)^2} = P_{ij} - P_{ij}^2$$

$$l \neq j: \quad \frac{0 - e^{S_{il}} e^{S_{ij}}}{(\Sigma)^2} = -P_{il}\, P_{ij}$$

$$= \delta_{lj} \left(\underbrace{P_{il} P_{ij}}_{P_{ij}} + P_{ij} - P_{ij}^2 \right) - P_{il}\, P_{ij}$$

$$= \delta_{lj}\, P_{ij} - P_{il}\, P_{ij} = P_{ij}(\delta_{jl} - P_{il}).$$

$$\frac{\partial L}{\partial S_{ij}} = \sum_{l} \frac{\partial L}{\partial P_{il}} \frac{\partial P_{il}}{\partial S_{ij}} = \sum_{l} \left(\frac{\partial L}{\partial O} V^T \right)_{il} P_{ij}(\delta_{jl} - P_{il})$$

$$= \sum_{k} \sum_{l} \frac{\partial L}{\partial O_{ik}} V_{lk}\, P_{ij}(\delta_{jl} - P_{il}).$$

$$= P_{ij} \sum_{k} \frac{\partial L}{\partial O_{ik}} \sum_{l} V_{lk}(\delta_{jl} - P_{il}) = P_{ij} \sum_{k} \frac{\partial L}{\partial O_{ik}} \left(V_{jk} - \sum_{l} P_{il} V_{lk} \right)$$

$$= P_{ij} \sum_{k} \frac{\partial L}{\partial O_{ik}} \left(\underbrace{V_{jk}}_{\widetilde{V}} - \underbrace{\sum_{l} P_{il} V_{lk}}_{O_{ik}} \right)$$

$$\boxed{= P_{ij} \left(\frac{\partial L}{\partial P_{ij}} - \sum_{k} \frac{\partial L}{\partial O_{ik}} O_{ik} \right)}$$

P & *S* not materialized: needs recomputation.

$$\frac{\partial L}{\partial P} = \frac{\partial L}{\partial O} V^T \qquad O_{ik} : \text{known} \qquad \sum_{k} \frac{\partial L}{\partial O_{ik}} O_{ik} : \text{propagated}$$

The Dilemma!

$$\frac{\partial L}{\partial Q} = \left(\frac{\partial L}{\partial S}\right) K \qquad \frac{\partial L}{\partial K} = \left(\frac{\partial L}{\partial S}\right)^T Q. \qquad \frac{\partial L}{\partial V} = P^T \left(\frac{\partial L}{\partial O}\right)$$

$$\boxed{\frac{\partial L}{\partial S_{ij}} = P_{ij}\left(\frac{\partial L}{\partial P_{ij}} - \sum_k \frac{\partial L}{\partial O_{ik}} O_{ik}\right)}$$

- Into steps: do we compute a row of S, or a column of S?
- Who will be inner product, who will be outer product?

Things we are not covering ...

- Causal masking: skipping tiles above the diagonal.
- Block-sparse attention
- Dropout

Key Takeaway:

- Zoom out:

$$E[X] = \sum_i x_i \, P(X = x_i) = \sum_i x_i \frac{c(x_i)/m}{\sum_j c(x_j)/m}$$

$$= \frac{\sum_i x_i \, c(x_i)/m}{\sum_j c(x_j)/m}$$

- Zoom in:

At step i:

$$d_i = \sum_{j=1}^{i} \frac{c(x_j)}{m_i} \qquad n_i = \sum_{j=1}^{i} \frac{x_j \, c(x_j)}{m_i} \qquad E_i[X] = \frac{n_i}{d_i}$$

At step $i+1$, with $m_{i+1} = \max(m_i, \, c(x_{i+1}))$:

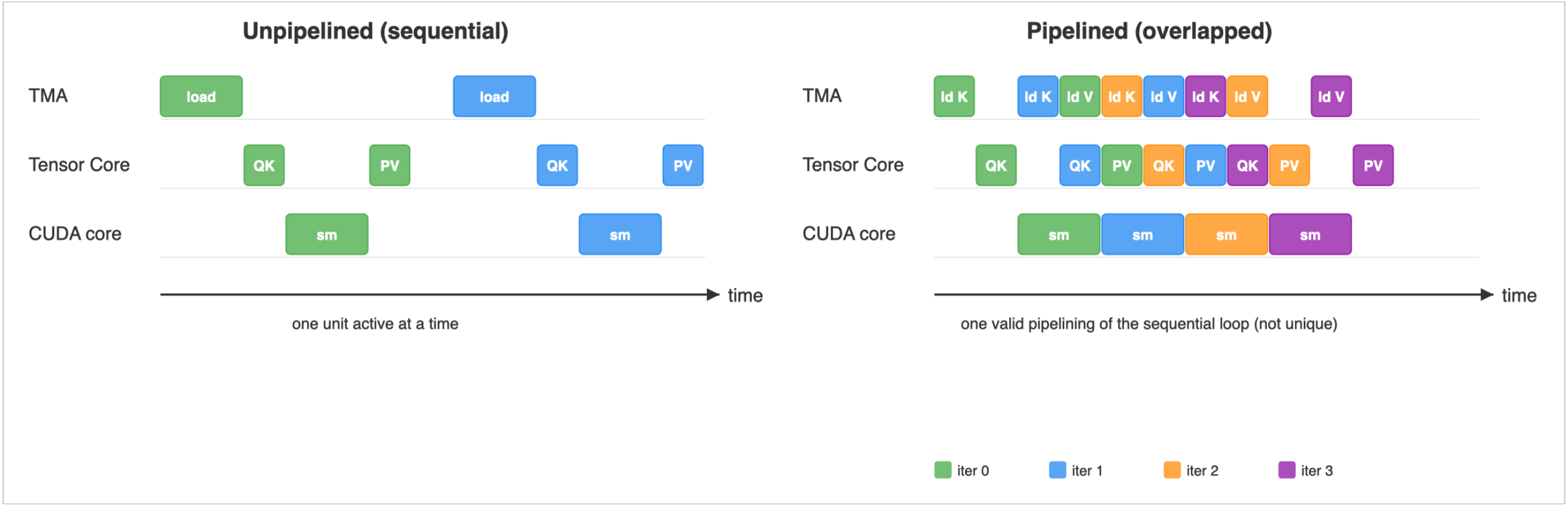
$$d_{i+1} = d_i \cdot \frac{m_i}{m_{i+1}} + \frac{c(x_{i+1})}{m_{i+1}} \qquad n_{i+1} = n_i \cdot \frac{m_i}{m_{i+1}} + \frac{x_{i+1} \, c(x_{i+1})}{m_{i+1}} \qquad E_{i+1}[X] = \frac{n_{i+1}}{d_{i+1}}$$

$$= P_{ij} \sum_k \frac{\partial L}{\partial O_{ik}} \left(\underbrace{V_{jk}}_{\widetilde{V}} - \underbrace{\sum_l P_{il} V_{lk}}_{O_{ik}} \right)$$

$$\boxed{= P_{ij} \left(\frac{\partial L}{\partial P_{ij}} - \sum_k \frac{\partial L}{\partial O_{ik}} O_{ik} \right)}$$

- Abstraction: Elements are tiles.
- Pattern: Inner product vs. outer product
- FA3: Instructions are tasks.

FA3


Unpipelined (sequential)
TMA
load
load
Tensor Core
QK
PV
QK
PV
CUDA core
sm
sm
time
one unit active at a time
Pipelined (overlapped)
TMA
ld K
ld K
ld V
ld K
ld V
ld K
ld V
ld V
Tensor Core
QK
QK
PV
QK
PV
QK
PV
PV
CUDA core
sm
sm
sm
sm
time
one valid pipelining of the sequential loop (not unique)
iter 0
iter 1
iter 2
iter 3

What's the most relevant paper to FA3?