Direct interpolative construction of quantized tensor trains

Tensor4All Meeting

April 22, 2024

Michael Lindsey

UC Berkeley

What is an MPS / TT?

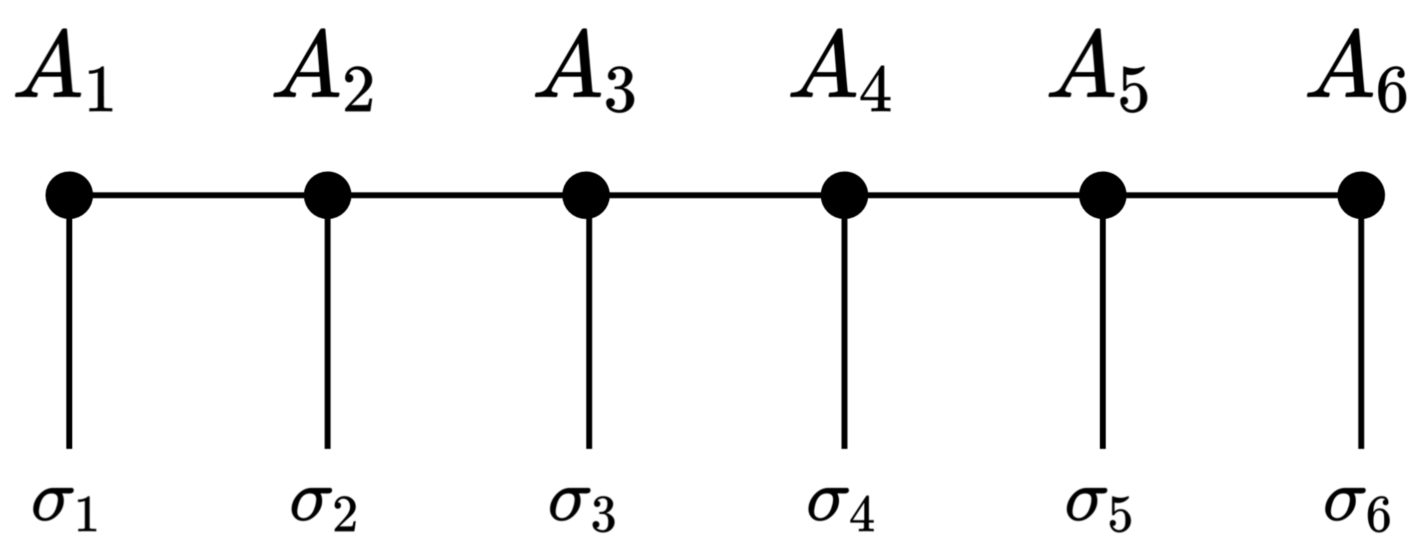
Consider a tensor $T \in (\mathbb{R}^2)^K \simeq \mathbb{R}^2 \times \cdots \times \mathbb{R}^2$

It is a matrix product state (**MPS**) / tensor train (**TT**) if it can be written:

$$T(\sigma_1, \ldots, \sigma_K) = \sum_{\alpha_1 \in [r_1], \ldots, \alpha_{K-1} \in [r_{K-1}]} A_1^{1,\alpha_1}(\sigma_1) A_2^{\alpha_1,\alpha_2}(\sigma_2) \cdots A_{d-1}^{\alpha_{d-2},\alpha_{d-1}}(\sigma_{K-1}) A_K^{\alpha_{K-1},1}(\sigma_K)$$

in terms of **tensor cores** $A_k \in \mathbb{R}^{2 \times r_{k-1} \times r_k}$

$r_1, \ldots, r_{K-1}$ are called the **bond dimensions / TT ranks**



What can be done with MPS / TT?

- **Basic primitives**
  - **Entrywise addition** (ranks grow additively)
  - **Entrywise multiplication** (ranks grow multiplicatively)
  - **MPO-MPS multiplication** (ranks grow multiplicatively)
  - **Optimal compression** of a single rank (cubic cost in rank)
- **Major algorithms**
  - **DMRG-style algorithms** (based on alternating block updates) for eigenvalue problems, linear least squares, and more
  - **TDVP** (time-dependent variational principle) for real/imaginary-time evolution
  - **TCI** (tensor cross interpolation) to construct TT from entry queries
- **References**
  - **Key historical references:** Fannes et al (1992), Klümper et al (1992), White (1992), Perez-Garcia et al (2007), Oseledets and Tyrtyshnikov (2009), Oseledets and Tyrtyshnikov (2010), Oseledets (2011)
  - Very helpful resource: **tensornetwork.org**

What is a QTT?

Consider a function:

$$f\,:\,[0,1]\to\mathbb{R}$$

Identify variable x with binary decimal expansion

$$x=\sum_{k=1}^{\infty}2^{-k}\sigma_k=0.\sigma_1\sigma_2\sigma_3...$$

$$x\leftrightarrow(\sigma_1,\ldots,\sigma_K)$$

Then we can identify f with a tensor T

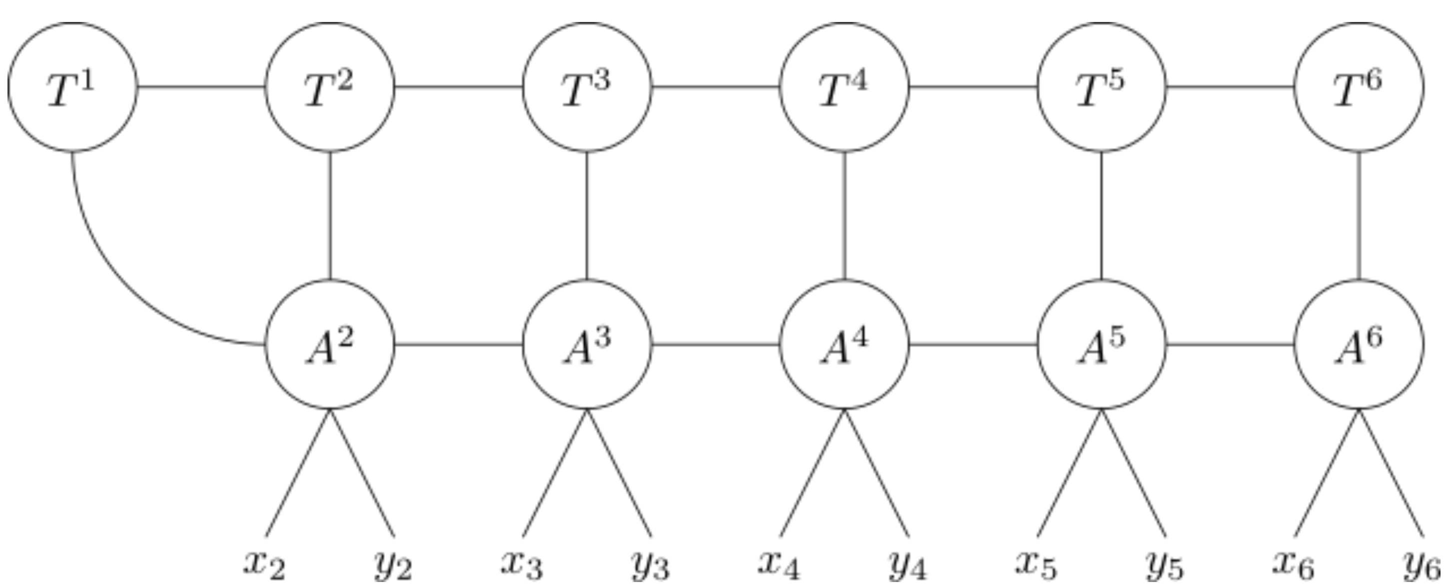
$$f(x)=T(\sigma_1,\ldots,\sigma_K)$$

A quantized tensor train (**QTT**) is a representation of such a tensor T as an MPS / TT [Khoromskij (2011)]

Why QTT?

- **Access to MPS / TT toolbox**
  - **DMRG**-type solvers
  - **TDVP**-type time evolution
- **Hidden structure may be revealed**
  - *What structure, and is it structure that cannot be revealed by other means?*

- **QTT-specific algorithms**
  - **Convolution**
    - Kazeev et al (2013)
    - See diagram below
  - **Discrete Fourier transform**
    - Dolgov et al (2012)
    - Chen et al (2023)



Can also do fast matvecs of QTTO times dense vectors

[Corona et al (2017)]

pmkielstra.github.io/Adding-Subtracting-and-Quantized-Tensor-Trains/

What is known about QTT compression?

- **Exponentials have rank 1:**
  - $\exp(\alpha x) = \exp\left(\alpha \sum_{k=1}^{K} 2^{-k}\sigma_k\right) = \prod_{k=1}^{K} e^{\alpha\, 2^{-k}\sigma_k}$
- **Degree-*N* polynomials have rank *N***
  - Explicit construction of cores [Oseledets (2013)]
- **Techniques for bounding QTT ranks:**
  - Approximate a function as a sum of Fourier modes [Dolgov et al (2012)]
  - Approximate with a polynomial [Shi and Townsend (2021)]
- **Questionable talking point:**
  - *If the QTT ranks are bounded, QTT offers exponential speedup over grid-based discretization*
- It is actually nontrivial to establish that the storage cost of QTTs for "smooth" functions is **not worse** than the cost of storing a grid / basis representation
  - But we will see that this is true, and in fact QTTs can flexibly represent more complicated functions that are tricky to represent "classically"

Part I: Analysis of QTT compression

M.L., *Multiscale interpolative construction of quantized tensor trains*, arXiv:2311.12554.

What is unknown about QTT compression?

- QTT ranks tend to decay asymptotically with depth. **Why?**
- The QTT ranks of a Gaussian is bounded independent of the width. **Why?**
  - Does ***not*** follow from Fourier series / polynomial approximation results
  - Similarly, other functions with sharp peaks have low QTT ranks
- The QTT ranks of an Ω-bandlimited function are $O\left(\sqrt{\Omega}\right)$, ***not*** $O(\Omega)$ as suggested by Fourier series approximation. **Why?**
- Although an explicit construction for the QTT cores of a polynomial is known, it is not stable because it involves coefficients in the monomial basis. **Can we achieve a stable construction?**
- Can we derive algorithms that **reveal the rank automatically** even if it is not understood *a priori*?

Unfolding matrices

- For any bond $m = 1, \ldots, K-1$, can view T as a matrix via $T(\sigma_{1:K}) = T(\sigma_{1:m}, \sigma_{m+1:K})$
  - This is called the *m*-th unfolding matrix of T
  - TT ranks are controlled by these ranks, cf. [Oseledets (2011)]
- Then if we can decompose $T(\sigma_{1:K}) \approx \sum_\alpha T_{\mathrm{L}}^{\alpha}(\sigma_{1:m})\, T_{\mathrm{R}}^{\alpha}(\sigma_{m+1:K})$, where we control the number of terms in the sum, we have control over the QTT ranks
- Later we will describe constructive algorithms for building the QTT….

Interpolative point of view

Split argument into big piece $x_{\leq m}$ and small piece $x_{>m} \in [0, 2^{-m}]$

$$f(x) = f(x_{\leq m} + x_{>m})$$

$$x_{\leq m} := \sum_{k=1}^{m} 2^{-k}\sigma_k, \quad x_{>m} := \sum_{k=m+1}^{K} 2^{-k}\sigma_k$$

Define function $[0,1] \to \mathbb{R}$ on reference interval:

$$v \mapsto f(u + 2^{-m}v)$$

Insert interpolative decomposition:

$$f(u + 2^{-m}v) \approx \sum_{\alpha} f(u + 2^{-m}c^{\alpha})P^{\alpha}(v)$$

Take c^{α} to be Chebyshev-Lobatto nodes on $[0,1]$ and P^{α} to be corresponding Lagrange interpolating functions

Therefore:

$$T(\sigma_{1:K}) = f(x) \approx \sum_{\alpha} \underbrace{f(x_{\leq m} + 2^{-m}c^{\alpha})}_{=:T_{\mathrm{L}}^{\alpha}(\sigma_{1:m})} \underbrace{P^{\alpha}(2^{m}x_{>m})}_{=:T_{\mathrm{R}}^{\alpha}(\sigma_{m+1:K})}$$

Rank of ***m***-th unfolding matrix is bounded by the number of terms in this sum

Decaying rank bounds

- Standard error bounds for Chebyshev interpolation (cf. Trefethen's book) can be applied under various assumptions on the smoothness of f
- **Importantly**, the interpolation gets ***easier*** as we go deeper into the QTT!
  - When you zoom in, things get smoother
- Most striking conclusion in the case where f is Ω-bandlimited
  - The *m*-th unfolding matrix rank is bounded via interpolation by $\sim 2^{-m}\,\Omega$
  - Meanwhile the *m*-th unfolding matrix rank is trivially bounded by 2^m (# of rows)

Theorem (M.L.), *stylized*: For an Ω-bandlimited function, the ε-ranks of the unfolding matrices an are uniformly bounded by $O\left(\sqrt{\Omega} + \log(1/\varepsilon)\right)$

- Thus the QTT storage complexity is ***not worse*** than grid representation

Part II: Direct construction of QTTs

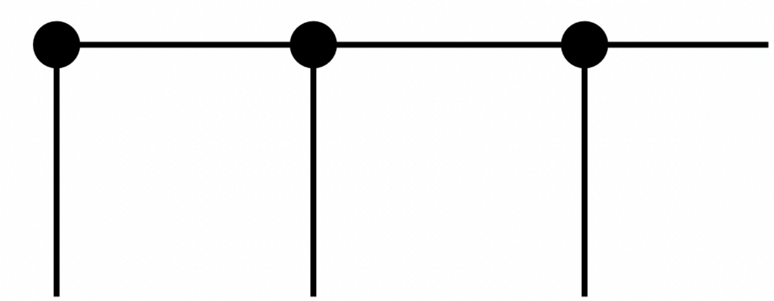
M.L., *Multiscale interpolative construction of quantized tensor trains*, arXiv:2311.12554.

Direct construction

Suppose we have constructed a tensor

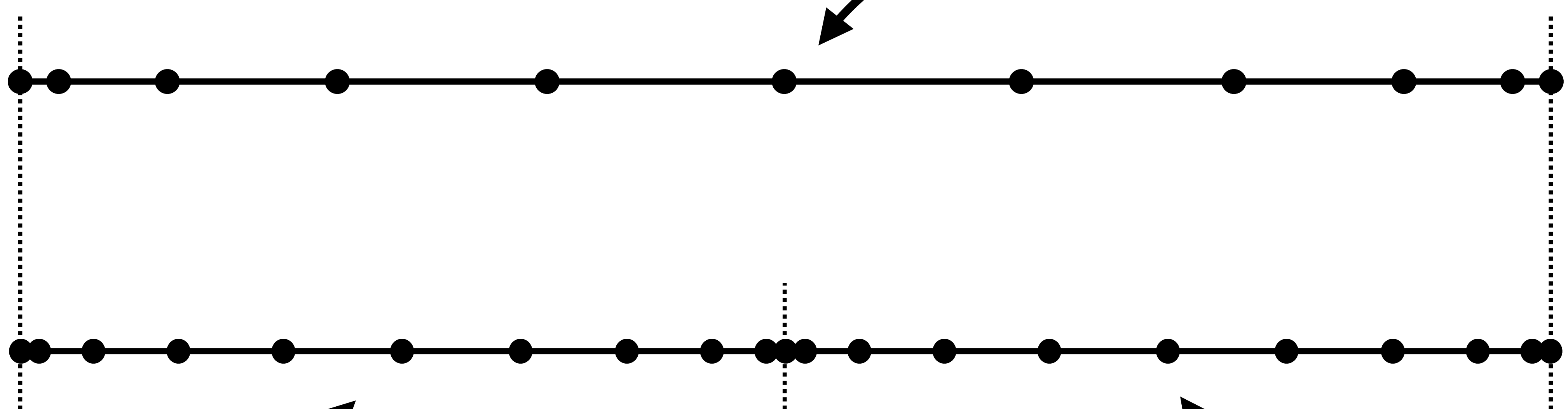
$$S^{\alpha}_{\leq m}(\sigma_{1:m}) \approx f\left(\sum_{k=1}^{m} 2^{-k}\sigma_k + 2^{-m}c^{\alpha}\right)$$

$S_{\leq 3}$



How to get next tensor $S^{\alpha}_{\leq m+1}(\sigma_{1:m+1})$?

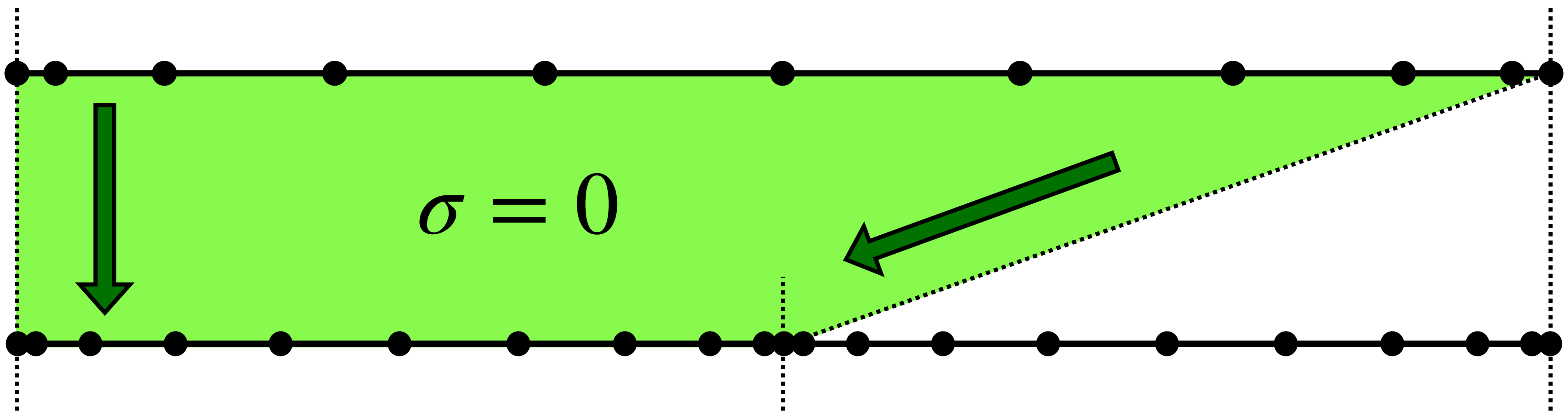
have represented function on this grid $\left\{x_{\leq m} + 2^{-m}c^{\alpha}\right\}_{\alpha=0}^{N}$



want to represent functions on the two grids $\left\{x_{\leq m} + 2^{-(m+1)}\sigma_{m+1} + 2^{-(m+1)}c^{\beta}\right\}_{\beta=0}^{N}$, for each $\sigma_{m+1} \in \{0,1\}$

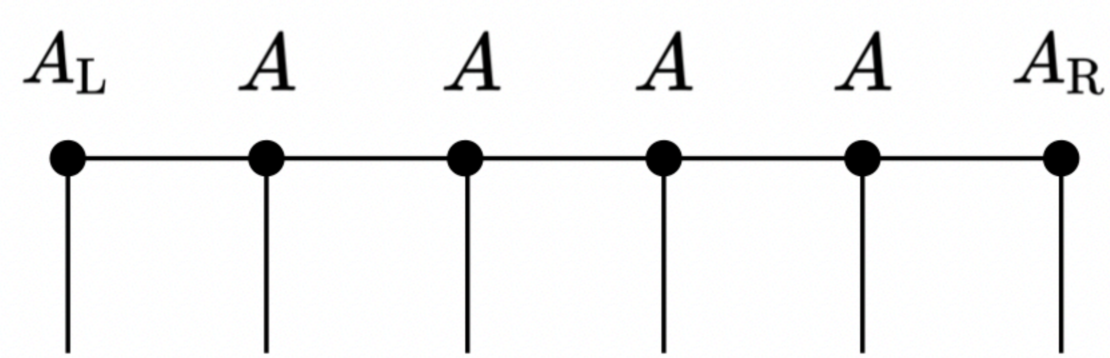
Direct construction

Construct a tensor core $A^{\alpha\beta}(\sigma)$ which interpolates wide grid to left or right narrow grid, depending on whether $\sigma = 0$ or 1



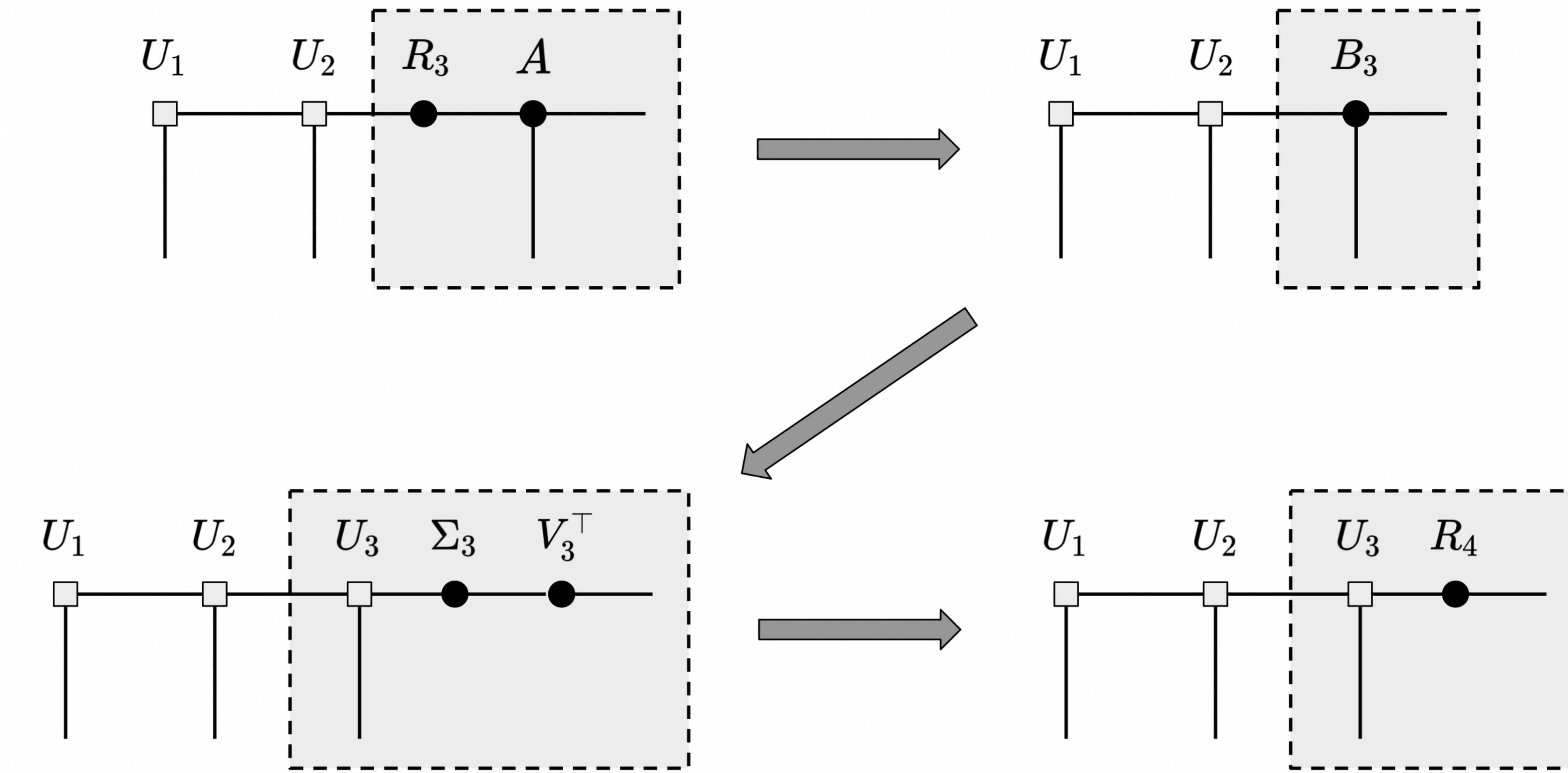
$$A^{\alpha\beta}(\sigma) := P^{\alpha}\left(\frac{\sigma + c^{\beta}}{2}\right)$$

Construct full QTT by repeatedly attaching this core (with boundary conditions):



Rank-revealing construction

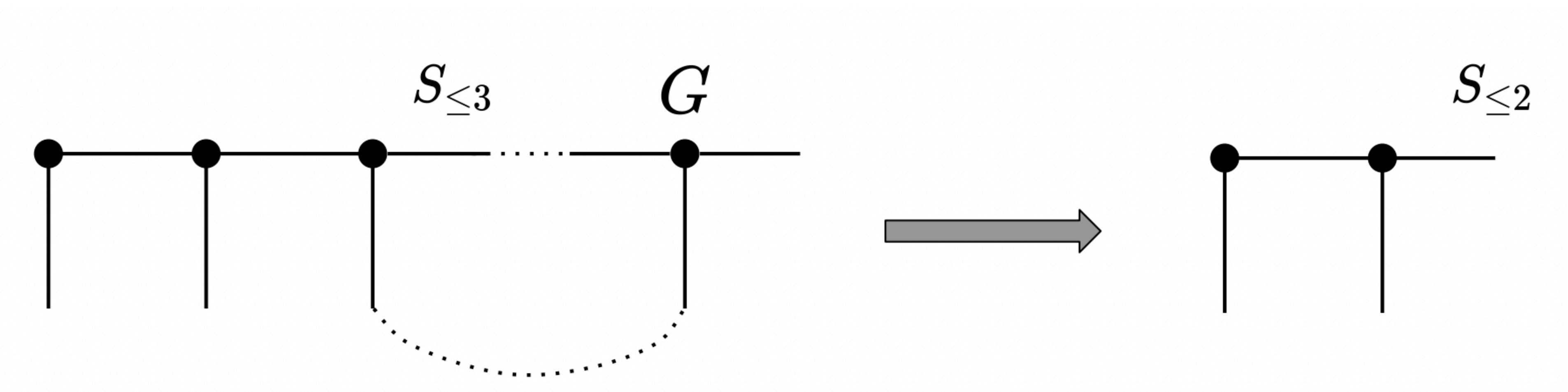
Can modify the construction to reveal the rank on the fly as we attach cores



Usually this is dangerous! But we are protected by the fact that the tail of *A* cores act as a Chebyshev interpolator, which can only amplify entrywise errors by the Lebesgue constant of the interpolation scheme (cf. Trefethen's book). See preprint for rigorous statement.

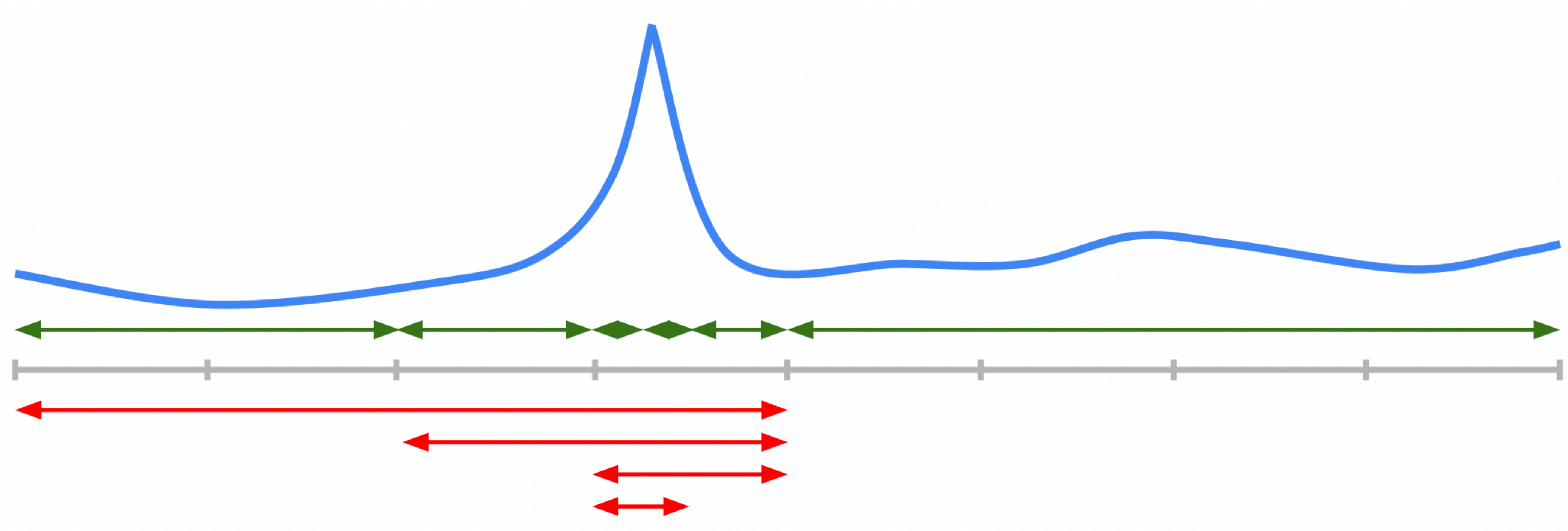
Improvements and extensions

- Can replace dense interpolating cores A with sparse approximations in the large N limit (where N is the size of the interpolating grid), cf. [Boyd (1992)]
  - Reduces the cost of rank-revealing algorithm to $O(Nr^2)$, where r is revealed rank
- Extensions to multivariate case
  - Different conventions (interleaved / serial ordering) are considered
  - Ye (speaking later in this session) et al have considered many options in practice
- Can "invert" the construction (recover interpolating grid values from QTT) by attaching a particular core G which is a generalized inverse of A



Multiresolution construction

Suppose that we can construct nested dyadic intervals (pictured in red) on which interpolation is "dangerous" (due to poor quantitative smoothness)



- For the complementary intervals (green) at each level, suppose that *N*-point interpolation is accurate
- Then we can construct an accurate QTT of rank $\boldsymbol{N + q}$, where q is the maximum number of dangerous subintervals at each level (here $\boldsymbol{q = 1}$)
  - "Cellular automaton" type construction: bide your time until you land in a safe subinterval, then interpolate all the way down

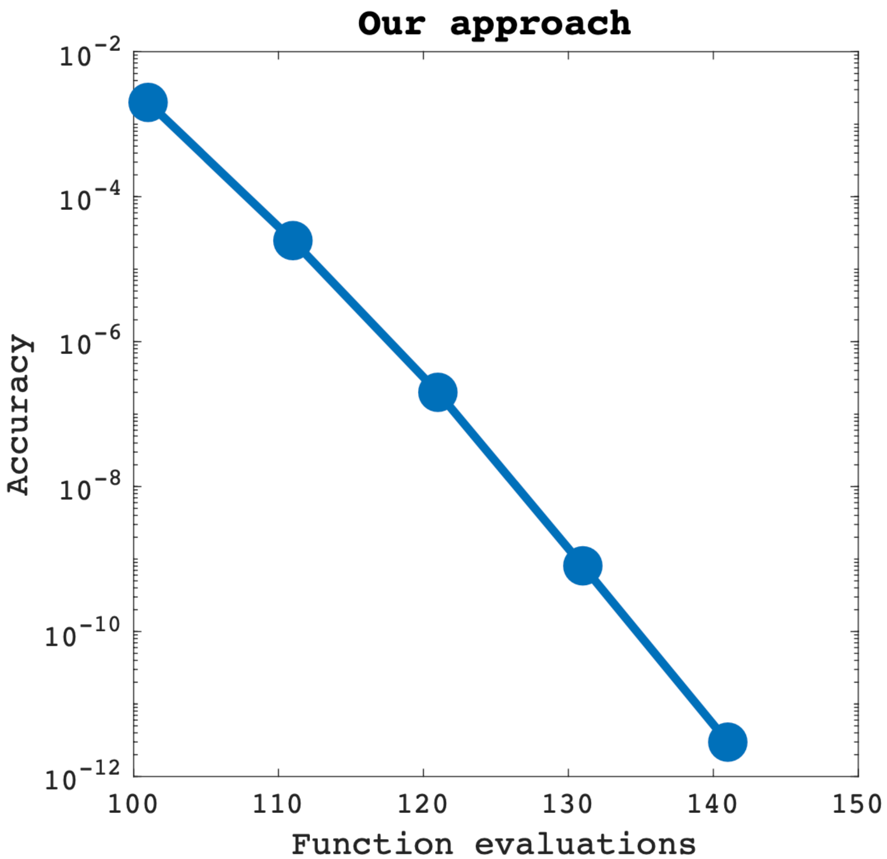
One simple demonstration

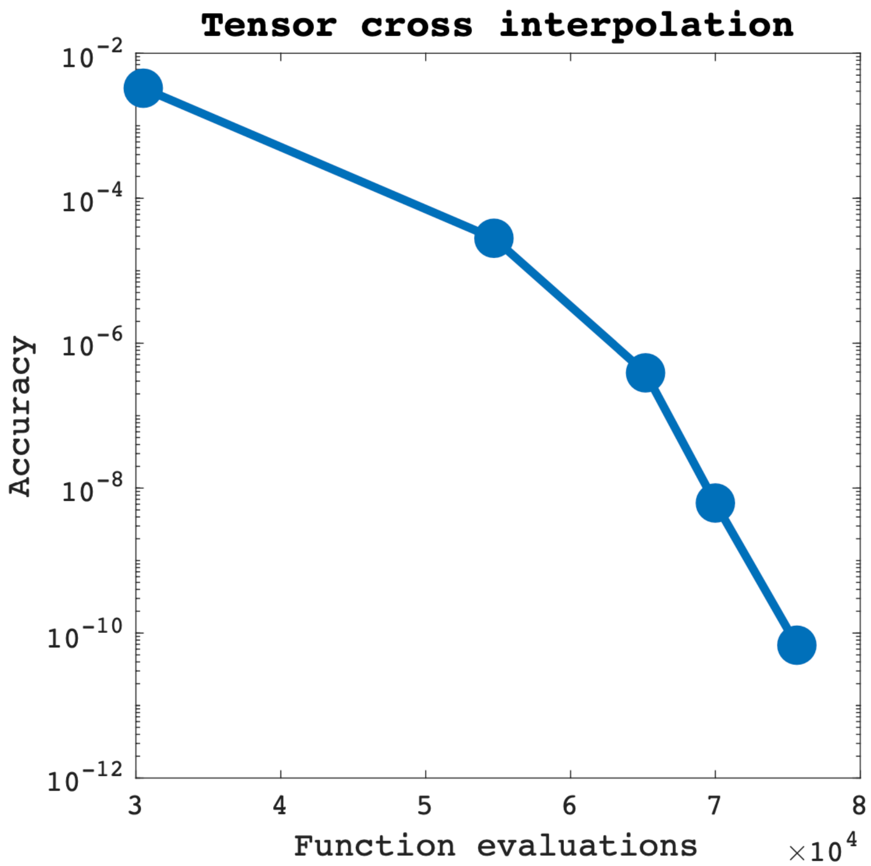
Random Fourier series

$$f(x) = \sum_{j=1}^{J} [a_j \cos(2\pi j x) + b_j \sin(2\pi j x)]$$

Compare to TCI

($J = 25$)





Stable results where TCI fails

($N = 2J$)

J	200	300	400	500	600	1000	2000
Error	9.8×10^{-11}	1.1×10^{-10}	8.4×10^{-11}	1.3×10^{-10}	1.8×10^{-10}	2.2×10^{-10}	3.5×10^{-10}

Further demonstrations

- See preprint for further demonstrations!
    - Sparse cores, construction inversion, multivariate cases, multiresolution construction (validated to be sharp for Gaussians, etc.)

Part II: MPO compression of the DFT

Jielun Chen and **M.L.**, *Direct interpolative construction of the discrete Fourier transform as a matrix product operator*, arXiv:2404.03182.

The DFT as a quantized operator

Consider the **discrete Fourier transform**:

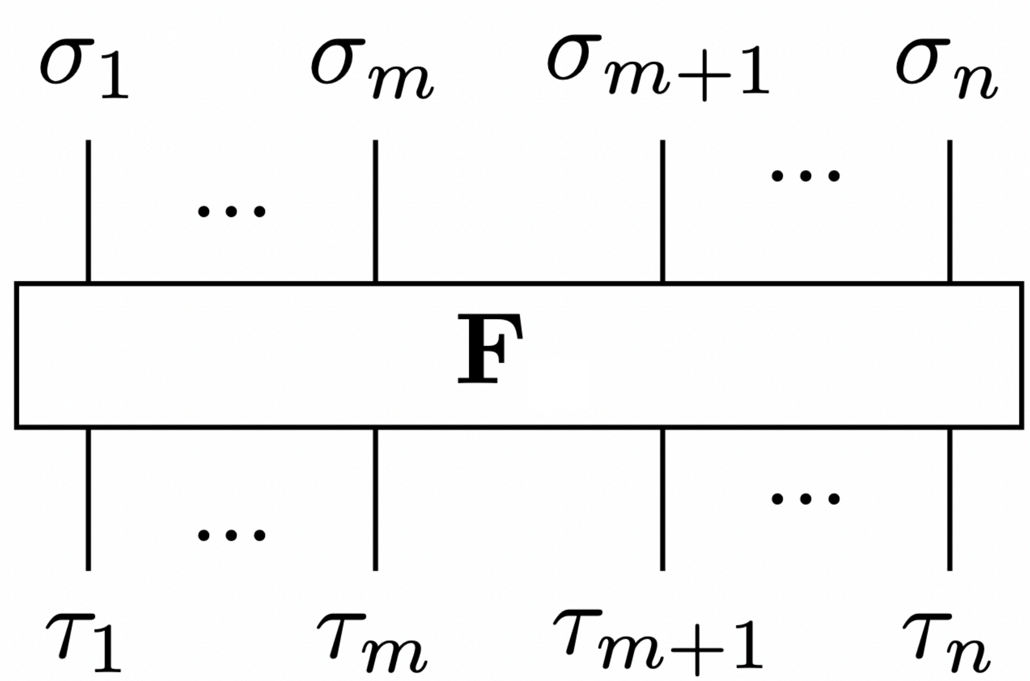
$\mathbf{F}_{s,t} = e^{-\frac{2\pi i s t}{N}}$	$s, t \in \{0, 1, \ldots, N-1\}$, where $N = 2^n$

Identify indices with binary expansions (bit-reversing the column index!)

$$s = \sum_{k=1}^{n} 2^{n-k} \sigma_k, \quad t = \sum_{k=1}^{n} 2^{k-1} \tau_k$$

Then we can identify **F** with a tensor:

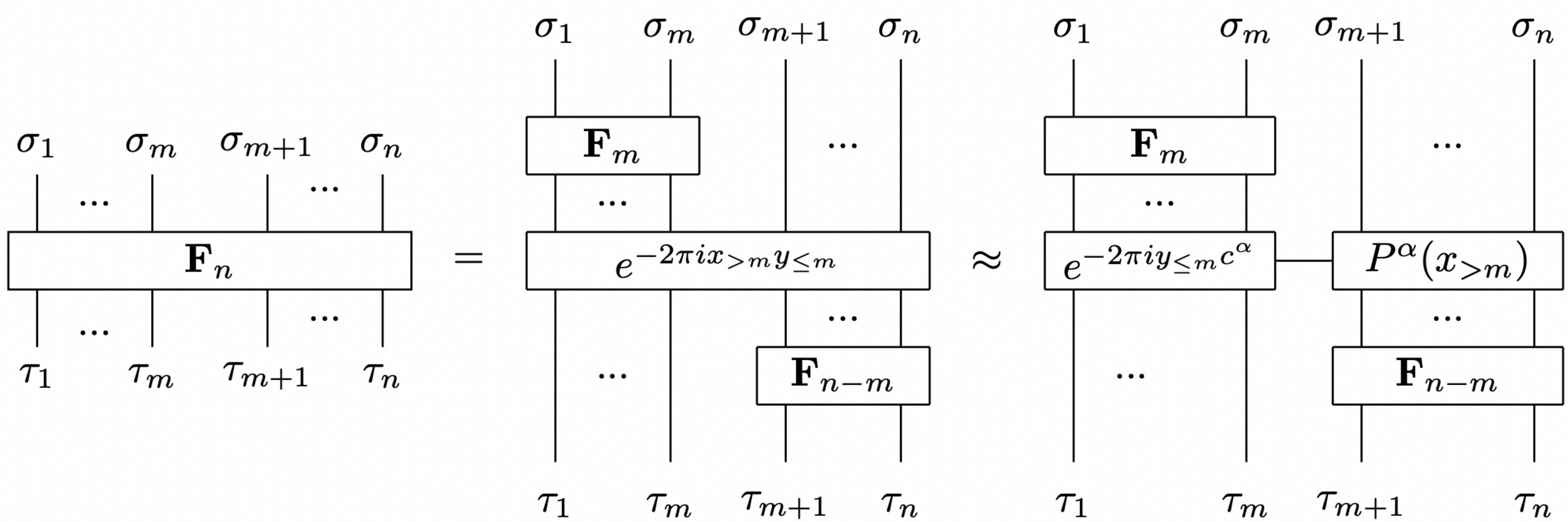
$$\mathbf{F}_{s,t} = F(\sigma_{1:n}, \tau_{1:n})$$
$$= e^{-\pi i \sum_{k,l=1}^{n} 2^{-k} 2^{l} \sigma_k \tau_l}$$



Previous analysis of DFT as MPO

- "Superfast Fourier transform" [Dolgov et al (2012)]
  - Applies DFT by interleaving MPO-MPS products with MPS compressions
- "QFT has low entanglement" [Chen et al (2023)]
  - Takes point of view of quantum circuit representation of the quantum Fourier transform (QFT)
  - Proved that the MPO for the DFT is **actually low-rank**
- **Explicit and efficient construction** of MPO as DFT still lacking!

New interpolative rank bound



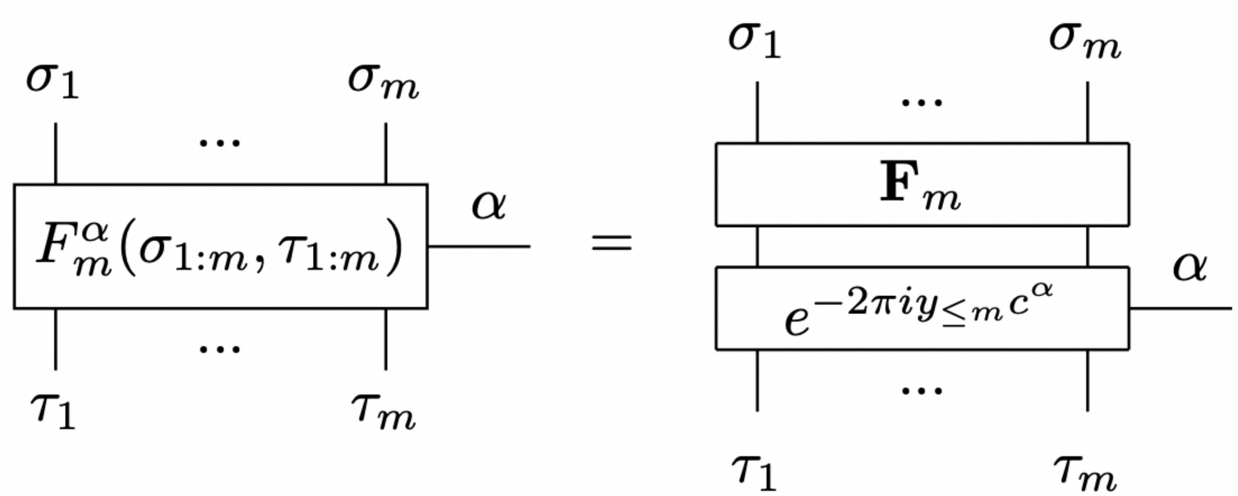
$$x_{>m} = \sum_{k=m+1}^{n} 2^{m-k}\sigma_k \in [0,1], \quad y_{\leq m} = \sum_{l=1}^{m} 2^{l-m-1}\tau_l \in [0,1]$$

Proposition 1. *There exists a rank-$(K+1)$ approximation of the m-th unfolding matrix of F whose entrywise error is bounded uniformly by*

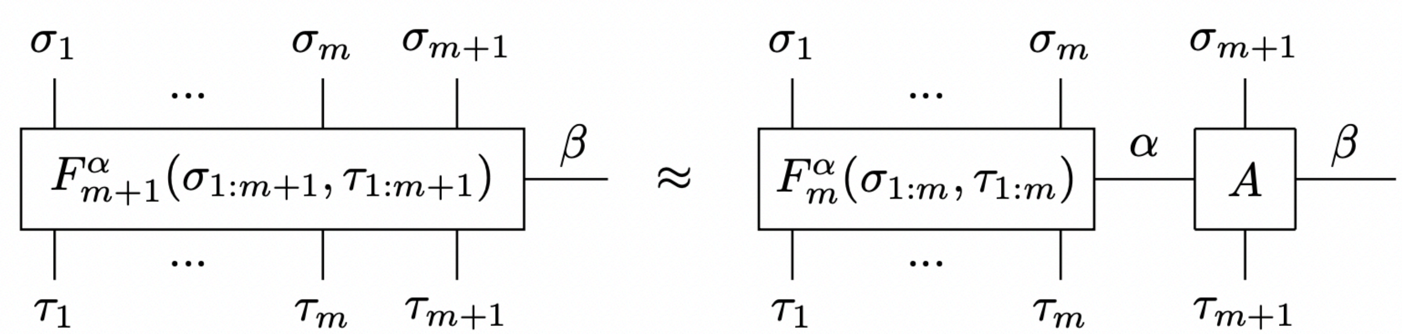
$$\frac{4\left(\frac{\pi}{2}\right)^{K+1} e^K K^{-K}}{K-\frac{\pi}{2}}.$$

Explicit MPO construction

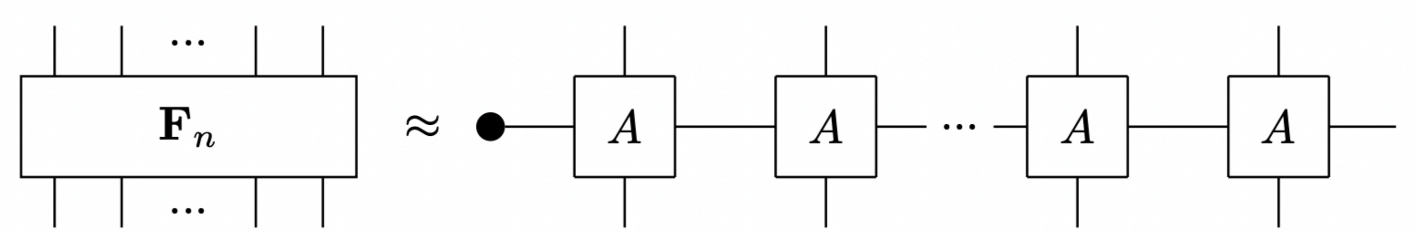
$$F_m^\alpha(\sigma_{1:m}, \tau_{1:m}) = e^{-\pi i \sum_{k,l=1}^m 2^{-k} 2^l \sigma_k \tau_l} e^{-2\pi i y_{\leq m} c^\alpha}$$



How to get next tensor $F_{m+1}^\alpha(\sigma_{1:m+1}, \tau_{1:m+1})$?



$$A^{\alpha\beta}(\sigma, \tau) = P^\alpha\left(\frac{\sigma + c^\beta}{2}\right) e^{-\pi i (\sigma + c^\beta)\tau}$$



Additional comments

- Error bound (cf. preprint) for explicit MPO construction nearly matches bounds for unfolding matrix ranks
- Connection between QTT and the complementary low rank (CLR) properties are noted
- Also consider connections to the approximate quantum Fourier transform (**AQFT**)
    - The AQFT simply leaves out long-range gates from QFT which contribute small phases
    - It turns out that the AQFT can be recovered **exactly** in our framework using a different (piecewise constant) interpolation scheme

Conclusions

- **Interpolation** is the right framework for leveraging smoothness to understand the detailed structure of QTT ranks, theoretically and practically
- It also helps us understand how to construct the DFT directly as an MPO
- Functions that can be represented efficiently with a tree-structured multiresolution grid are low-rank QTTs
  - So are their Fourier transforms
- We can go back and forth between (multires) grids and QTTs (zipping and unzipping)
- **Questions:**
  - Are there any interesting / useful functions which **cannot** be treated sharply with this analysis?
  - Can we get end-to-end QTT advantage over classical methods on a well-defined numerical analysis problem? What would this mean?

All aboard this actual quantized tensor train

Thank you for your attention!

arXiv:2311.12554

arXiv:2404.03182