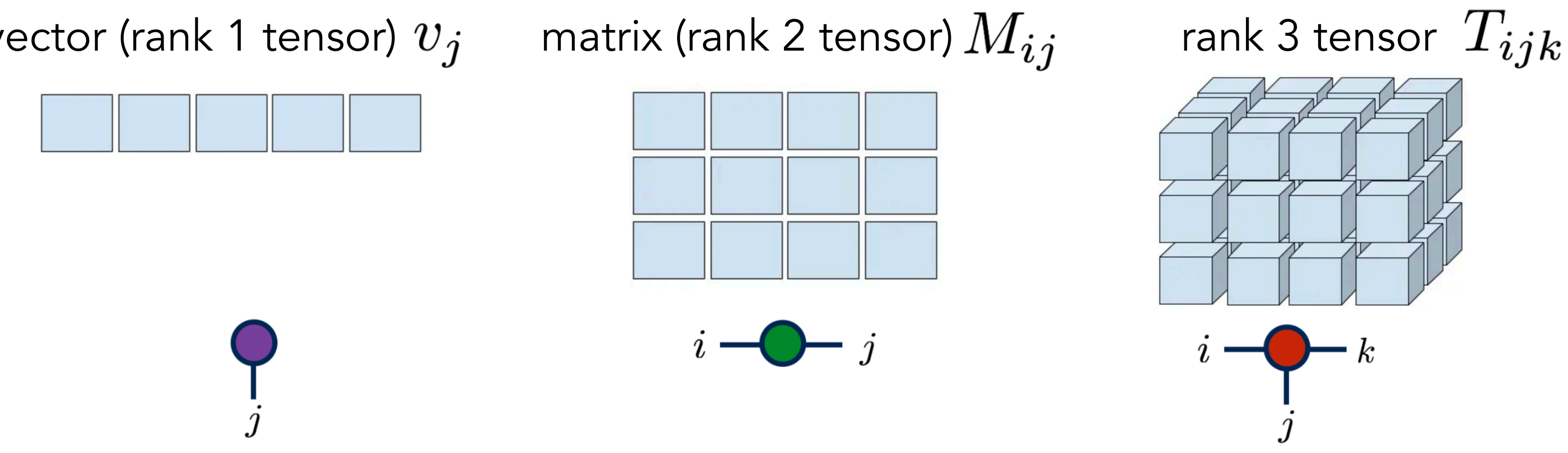
Quantum-inspired tensor network methods for construction and transformation of multivariate functions

Ryan Anselm
Center for Computational Quantum Physics, Flatiron Institute, New York

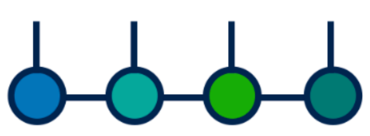
Background

- A tensor of rank **r** is a multilinear map that can be represented with an **r**-dimensional array of numbers.

vector (rank 1 tensor) v_j matrix (rank 2 tensor) M_{ij} rank 3 tensor T_{ijk}



- Tensor networks are factorizations of very large tensors into networks of smaller tensors. If a low-rank factorization exists, it is *exponentially* more memory efficient than the original tensor.

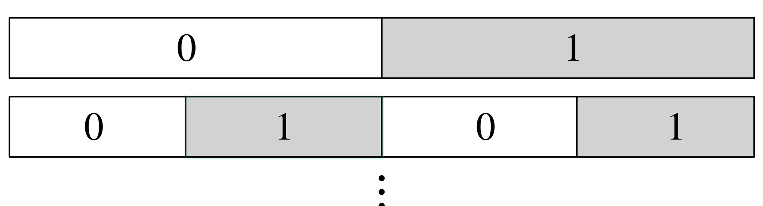


$$= \sum_{\alpha_1,\alpha_2,\alpha_3} A^{s_1}_{\alpha_1} B^{s_2}_{\alpha_1,\alpha_2} C^{s_3}_{\alpha_2,\alpha_3} D^{s_4}_{\alpha_3}$$

- A **quantized tensor network (QTN)** representation encodes a function of continuous variable(s) $f(x)$ on a bounded domain to n bits of precision using a tensor network $T_{s_1,s_2,\ldots,s_{n-1},s_n}$:

$$f(x) = T_{s_1,s_2,\ldots,s_{n-1},s_n} \qquad x = \frac{s_1}{2^1} + \frac{s_2}{2^2} + \ldots + \frac{s_{n-1}}{2^{n-1}} + \frac{s_n}{2^n}$$

$$x \in \left\{ \frac{0}{2^n}, \frac{1}{2^n}, \ldots, \frac{2^n-2}{2^n}, \frac{2^n-1}{2^n} \right\} \qquad \vec{s} \in \{0,1\}^n$$

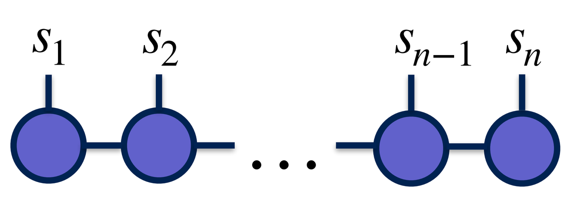


- QTN format takes advantage of **separation of scales,** present in many physical systems of interest

Example: $f(x) = e^{ax}$ has a directly constructible low-rank quantized tensor network form.

$$e^{ax} = e^{a(s_1/2 + s_2/2^2 + \ldots + s_n/2^n)} = (e^{a/2})^{s_1} \cdot (e^{a/2^2})^{s_2} \cdot \ldots \cdot (e^{a/2^{n-1}})^{s_{n-1}} \cdot (e^{a/2^n})^{s_n}$$

$$\equiv \begin{bmatrix} 1 \\ e^{a/2} \end{bmatrix}_{s_1} \cdot \begin{bmatrix} 1 \\ e^{a/2^2} \end{bmatrix}_{s_2} \cdot \ldots \cdot \begin{bmatrix} 1 \\ e^{a/2^{n-1}} \end{bmatrix}_{s_{n-1}} \cdot \begin{bmatrix} 1 \\ e^{a/2^n} \end{bmatrix}_{s_n}$$



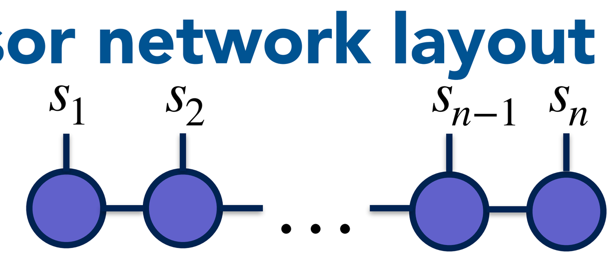
- Can directly construct exponentials, trigonometric functions, polynomials, and sums and products of these functions

$$\cos(ax) = \frac{e^{iax} + e^{-iax}}{2} \qquad \sin(ax) = \frac{e^{iax} - e^{-iax}}{2i}$$

This project uses and contributes to the development of **ITensorNumericalAnalysis.jl**, a part of the **ITensors** ecosystem developed at CCQ.

Automatic determination of tensor network layout

- The most well-studied layout for QTNs is a **matrix product state (MPS)**:

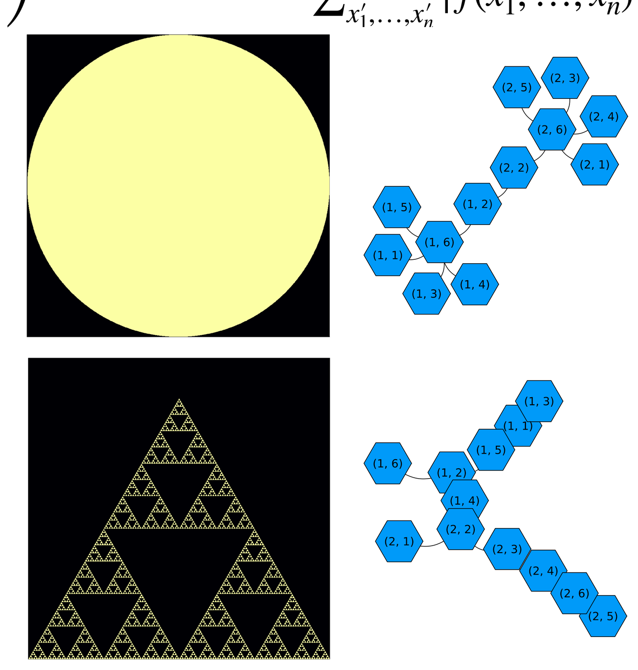


- However, numerical experiments indicate other layouts can outperform an MPS

Mutual Information (MI):

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} P(X=x, Y=y) \log\left(\frac{P(X=x,Y=y)}{P(X=x)P(Y=y)}\right) \qquad P(x_1,\ldots,x_n) = \frac{|f(x_1,\ldots,x_n)|^2}{\sum_{x'_1,\ldots,x'_n} |f(x'_1,\ldots,x'_n)|^2}$$

- Mutual information $I(x_i; x_j)$ gives a *measure of correlation* between two input variables x_i and x_j.
- We explore determining tensor network layout by finding a *maximum spanning tree* using pairwise MI values as edge weights of a graph.
- Should balance MI with *low degree* and *scale locality* to obtain memory-efficient layouts.

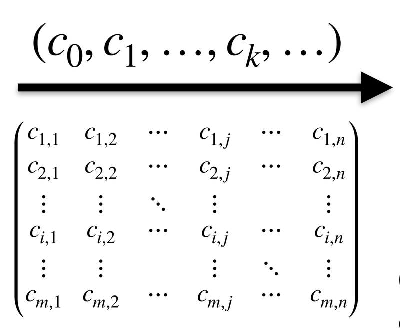


Above: Optimal spanning tree layouts for the above 2D functions according to MI heuristic. Two high degree nodes are found for the circle, while the Sierpiński triangle forms a line along each dimension.

Interpolative Methods

Fourier and Chebyshev Interpolation

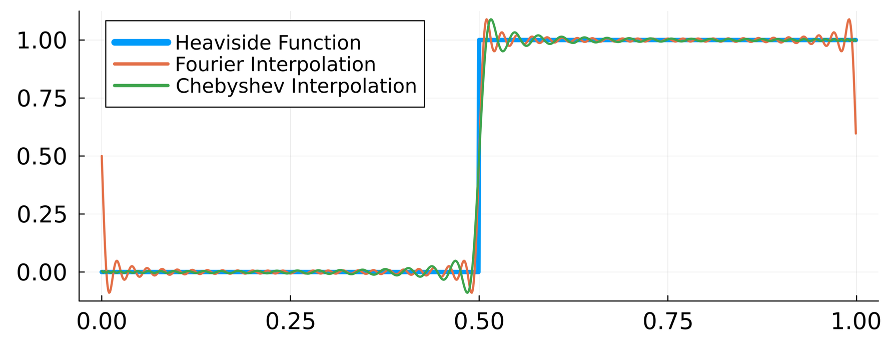
1. Find the Fourier/Chebyshev coefficients $\{c_i\}_{i\in\mathbb{N}}$ for $f(\mathbf{x})$ (1D or 2D)
2. Truncate the coefficients vector/matrix to desired precision.

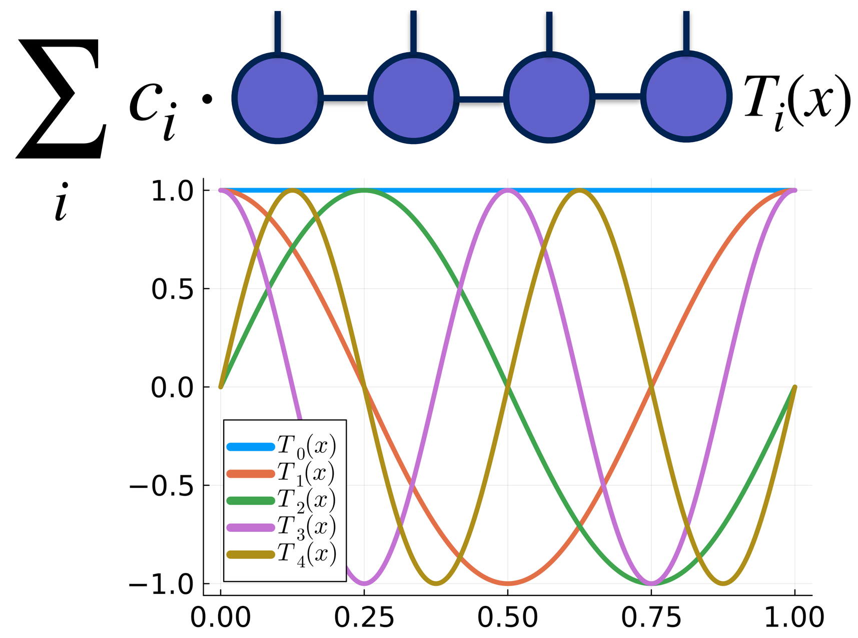


3. Construct approximation of $f(\mathbf{x})$ as weighted sum of QTN basis functions (sines/cosines/polynomials*)

(*use recursive Clenshaw evaluation for efficient construction of truncated Chebyshev expansions)

- Fourier/Chebyshev interpolation struggles with **non-smooth functions** e.g. image data or step functions.

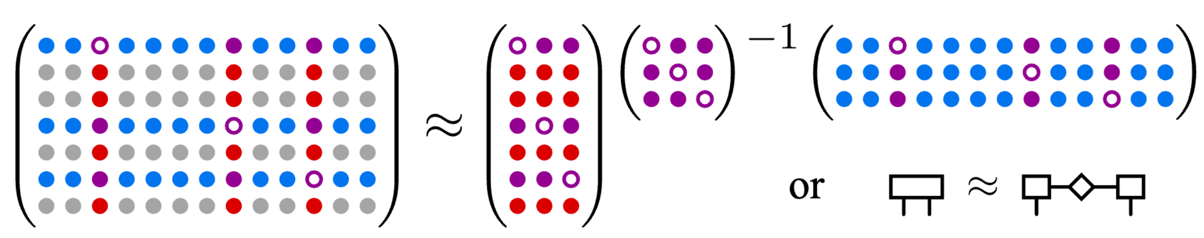




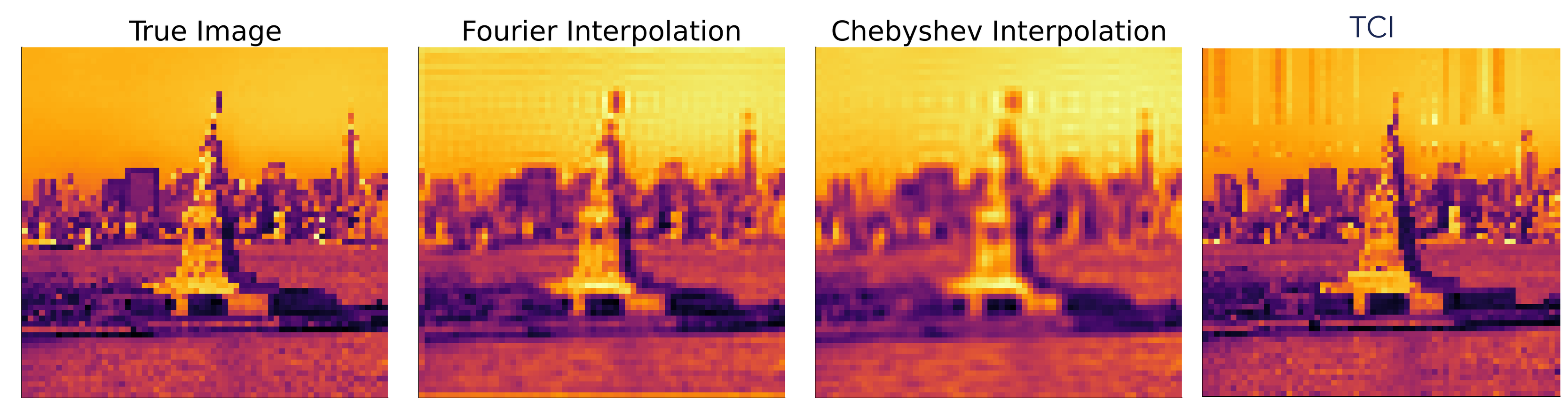
Above: Fourier basis functions

Tensor Cross Interpolation (TCI)

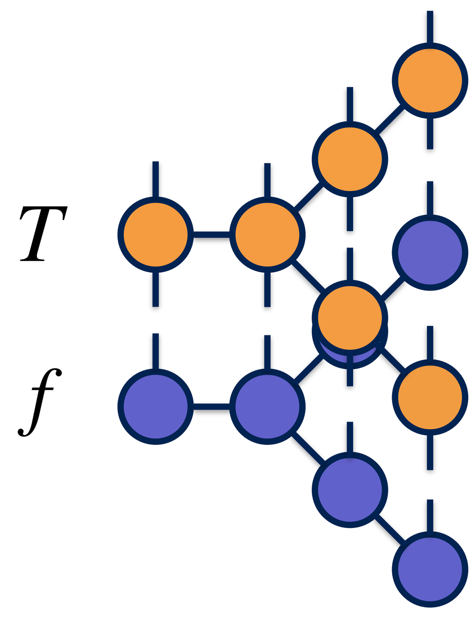
- **TCI** is a tensor network learning algorithm that adaptively interpolates a desired function/tensor using matrix cross interpolation.



Above: TCI approximates a large matrix by interpolating on a selected submatrix defined by pivots (empty circles).



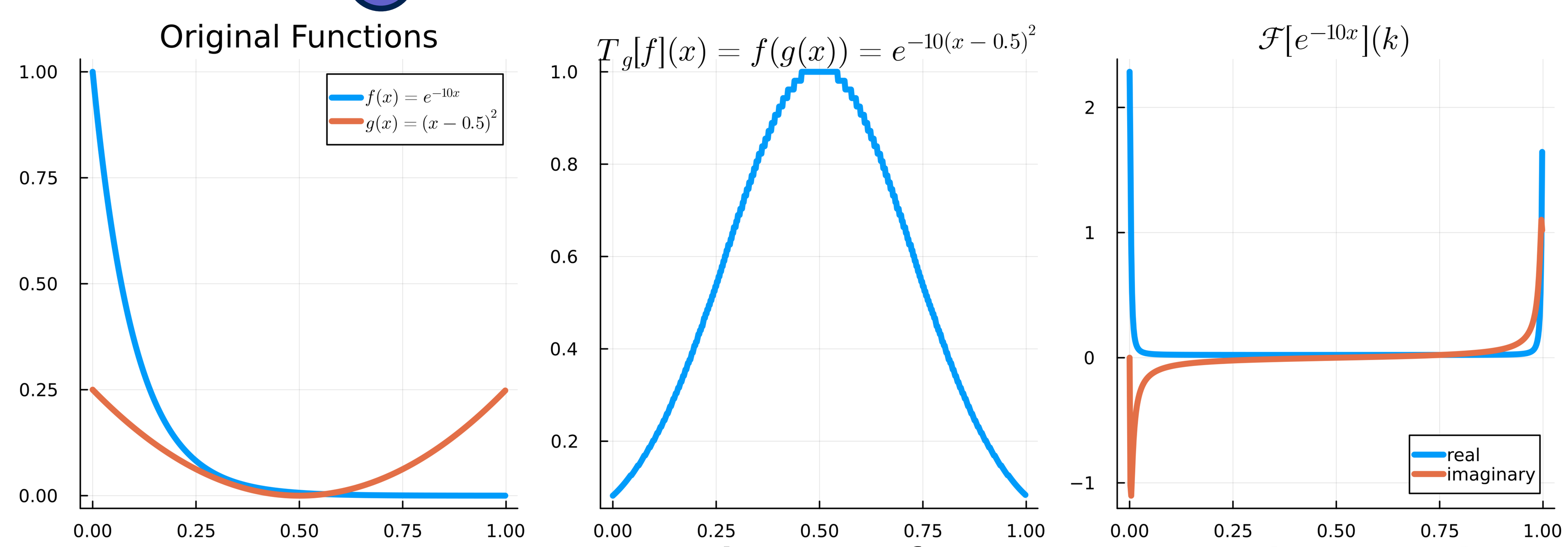
Functionals in quantized tensor network format



Can also construct **functionals** in QTN format that preserve network layout, enabling native transformation of QTN functions to other QTN functions.

Koopman (composition) operator $T = K_g$ $\quad K_g[f](x) = f(g(x))$

Fourier operator $T = \mathcal{F}$ $\quad \mathcal{F}[f](k) = \sum_{x=0}^{2^n} f(x) e^{-2\pi i k x/2^n}$



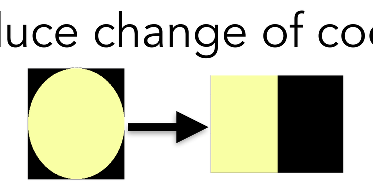
$f(x) = e^{-10x}$, $g(x) = (x-0.5)^2$ Application of Koopman operator Application of Fourier operator

Conclusions & Future Research

Through the development of the **ITensorNumericalAnalysis.jl** library**,** we expand the toolbox of techniques for applying quantized tensor networks to scientific domains that require memory-efficient representation and manipulation of multivariate functions.

Future research desiderata include:

- Rigorous criteria & algorithms for optimizing tensor network layout via mutual information
- An improved TCI variant that dynamically determines tensor network layout
- Implementations of QTN operators that induce change of coordinate system e.g. Polar to Cartesian: $(r,\theta) \to (r\cos\theta, r\sin\theta)$



CONTACT :
ranselm@flatironinstitute.org

Acknowledgements: Ryan Levy (CCQ), Joseph Tindall (CCQ), Miles Stoudenmire (CCQ)