SAND2023-12786C

# Modeling spatio-temporal processes in climate models via functional tensor networks

Cosmin Safta[1], Alex Gorodetsky[3], John D. Jakeman[2], Khachik Sargsyan[1]

[1]Sandia National Laboratories, Livermore, CA

[2]Sandia National Laboratories, Albuquerque, NM

[3]University of Michigan, MI

SIAM CSE (2023)
Feb 23 - March 3, 2023

SAND2023-XXXX

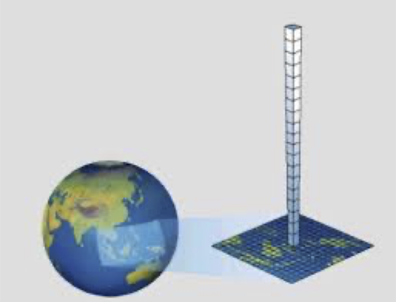
# Acknowledgement

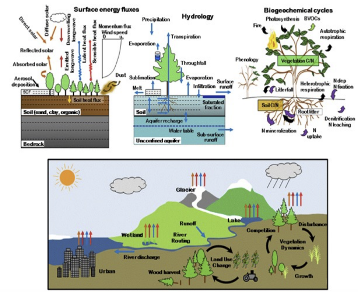
This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) Program through the FASTMath Institute.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This presentation describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the presentation do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

# Outline

# Energy Exascale Earth System Model (E3SM) Land Component






- The Land Model (ELM) Component of the Energy Exascale Earth System Model (E3SM) is increasingly complex with many processes
  - Large ensembles are needed for uncertainty quantification are not computationally infeasible
  - Focus on surrogate models that exploit model structure to increase the efficiency of sensitivity analysis and model calibration studies

# Cheaper Surrogates are Necessary to Replace Expensive Computational Models for UQ Assessments

## Requirements:

- expressivity with a limited number of parameters
- once constructed surrogate models need to be computationaly cheap – analyses often requiring $O(10^6)$ evaluations with limited computational resources

## Functional Approximations:

- tensor-product basis approximations

$$f(\boldsymbol{\lambda}) = \sum_{i_1}^{N_1}\sum_{i_2}^{N_2}\ldots\sum_{i_d}^{N_d}\phi_1^{(i_1)}(\lambda_1;\boldsymbol{\theta})\phi_2^{(i_2)}(\lambda_2;\boldsymbol{\theta})\ldots\phi_d^{(i_d)}(\lambda_d;\boldsymbol{\theta})$$

  - the curse of dimensionality $O(N^d)$ typically limits the polynomial order/no. of functions
  - ...this places limits on the surrogate model capacity to adapt to non-linear behavior
- Instead focus on *low-rank functional tensor network* models

# Cheaper Surrogates are Necessary to Replace Expensive Computational Models for UQ Assessments

## Requirements:

- expressivity with a limited number of parameters
- once constructed surrogate models need to be computationaly cheap – analyses often requiring $O(10^6)$ evaluations with limited computational resources

## Functional Approximations:

- tensor-product basis approximations

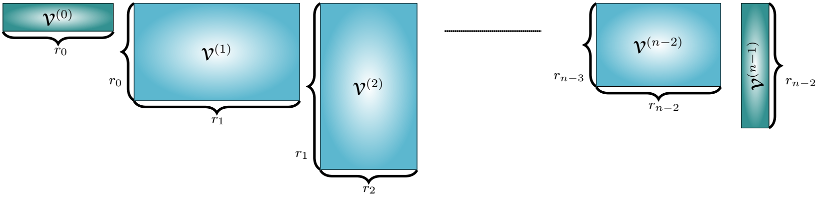
$$f(\boldsymbol{\lambda}) = \sum_{i_1}^{N_1} \sum_{i_2}^{N_2} \dots \sum_{i_d}^{N_d} \phi_1^{(i_1)}(\lambda_1; \boldsymbol{\theta}) \phi_2^{(i_2)}(\lambda_2; \boldsymbol{\theta}) \dots \phi_d^{(i_d)}(\lambda_d; \boldsymbol{\theta})$$

  - the curse of dimensionality $O(N^d)$ typically limits the polynomial order/no. of functions
  - ...this places limits on the surrogate model capacity to adapt to non-linear behavior

- Instead focus on *low-rank functional tensor network* models

# Functional Tensor-Train Models

Analogous to tensor-train models [Oseledets, 2013]: approximate multivariate functions instead of multidimensional arrays

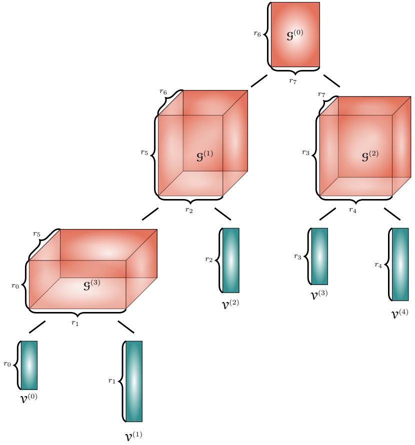



$$\mathcal{V}^{(k)}(\lambda_k;\boldsymbol{\theta}_k) = \begin{bmatrix} f_{11}^{(k)}(\lambda_k;\boldsymbol{\theta}_{11}^{(k)}) & f_{12}^{(k)}(\lambda_k;\boldsymbol{\theta}_{12}^{(k)}) & \cdots & f_{1r_k}^{(k)}(\lambda_k;\boldsymbol{\theta}_{1r_k}^{(k)}) \\ f_{21}^{(k)}(\lambda_k;\boldsymbol{\theta}_{21}^{(k)}) & f_{22}^{(k)}(\lambda_k;\boldsymbol{\theta}_{22}^{(k)}) & \cdots & f_{2r_k}^{(k)}(\lambda_k;\boldsymbol{\theta}_{2r_k}^{(k)}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{r_{k-1}1}^{(k)}(\lambda_k;\boldsymbol{\theta}_{r_{k-1}1}^{(k)}) & f_{r_{k-1}2}^{(k)}(\lambda_k;\boldsymbol{\theta}_{r_{k-1}2}^{(k)}) & \cdots & f_{r_{k-1}r_k}^{(k)}(\lambda_k;\boldsymbol{\theta}_{r_{k-1}r_k}^{(k)}) \end{bmatrix}$$

- Model evaluation/gradient computation consists of a sequence of matrix-vector multiplications
    - A.A. Gorodetsky, J.D. Jakeman. “Gradient-based optimization for regression in the functional tensor-train format,” J. of Comp. Phys. 374 (2018): 1219-1238.

# Tensor Models can have Arbitrary Network Structure

- Increased flexibility to represent model structure
- Example: a hierarchical Tucker format for a 5-dimensional model




- $\mathcal{V}^{(k)}$ represent tensor cores constructed with univariate functions in $\lambda_k$.
- $\mathcal{G}^{(i)}$ represent tensor cores with scalar elements.

# Functional Tensor Networks – Definitions

A tensor contraction is a binary operation on two tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times \ldots \times I_{d_A}}$ and $\mathcal{B} \in \mathbb{R}^{J_1 \times \ldots \times J_{d_B}}$ yielding a tensor $\mathcal{C}$.

$$\mathcal{C} = \mathcal{A} \,{}_{\Gamma}\!\times_{\Upsilon} \mathcal{B}$$

- The operation is parameterized by two index sets, $\Gamma = \{\gamma_1, \ldots, \gamma_\ell\}$ and $\Upsilon = \{\eta_1, \ldots, \eta_\ell\}$, satisfying three conditions:
  1. $1 \le \gamma_k \le d_A$ for each $\gamma_k \in \Gamma$
  2. $1 \le \eta_k \le d_B$ for each $\eta_k \in \Upsilon$
  3. $I_{\gamma_k} = J_{\eta_k}$ for $k = 1, \ldots, \ell$
- After permuting the modes so that the contracting dimensions are first

$$c_{j_1,\ldots,j_{d_A-\ell},k_1,\ldots,k_{d_B-\ell}} = \sum_{\gamma_1=1}^{I_{\gamma_1}} \cdots \sum_{\gamma_\ell=1}^{I_{\gamma_\ell}} \tilde{a}_{\gamma_1,\ldots,\gamma_\ell,j_1,\ldots,j_{d_A-\ell}} \tilde{b}_{\gamma_1,\ldots,\gamma_\ell,k_1,\ldots,k_{d_B-\ell}},$$

  with $\mathcal{C}$ having order $d_A + d_B - 2\ell$.

Example: Matrix-Matrix multiplication

$$c_{j,k} = \sum_{\gamma_1=1}^{I_\gamma} \tilde{a}_{\gamma,j} b_{\gamma,k}$$

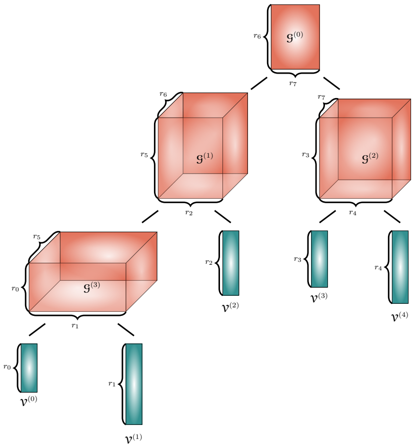
# Functional Tensor Networks – Definitions

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \;{}_{\vec{i}}\times_{\vec{j}}\; \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

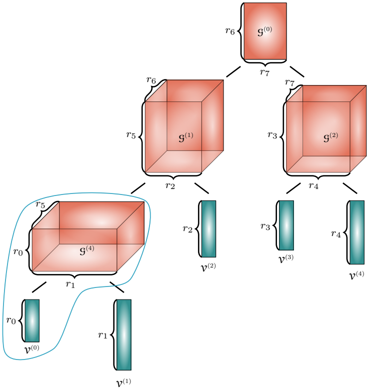
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \; {}_{\vec{i}}\times_{\vec{j}} \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

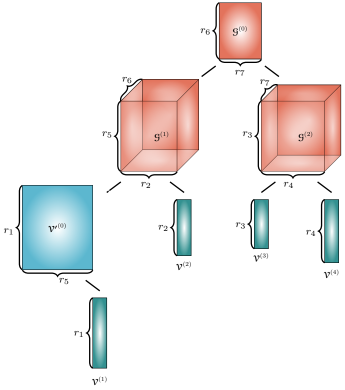
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \;{}_{\vec{i}}\!\times_{\vec{j}} \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

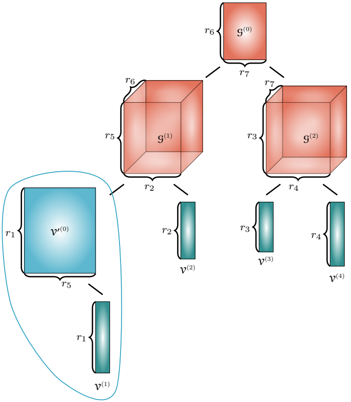
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \,{}_{\vec{i}}\!\times_{\vec{j}}\, \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

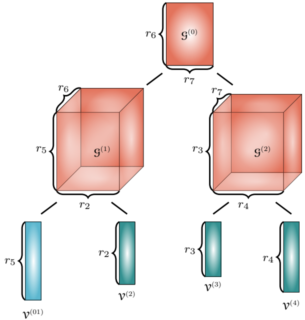
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
    - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \;{}_{\vec{i}}\!\times_{\vec{j}} \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

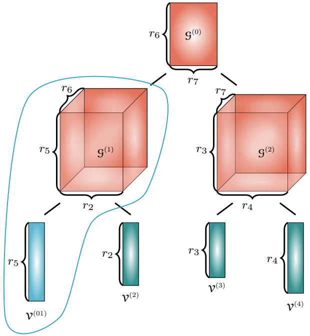
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
    - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \;{}_{\vec{i}}\times_{\vec{j}}\, \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

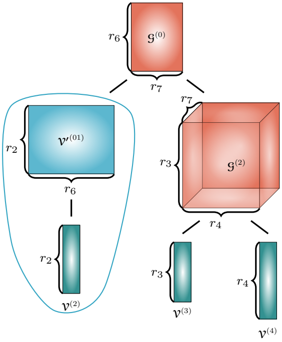
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \ {}_{\vec{i}}\times_{\vec{j}} \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \dots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \dots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

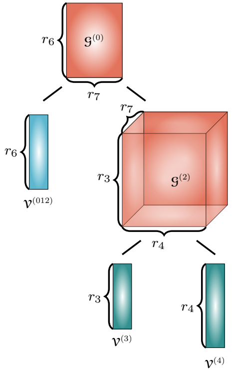
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \;{}_{\vec{i}}\times_{\vec{j}}\; \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

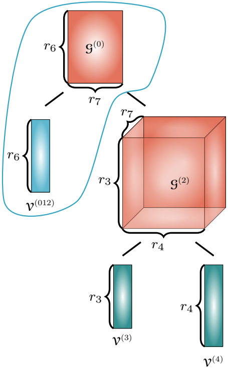
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \;_{\vec{i}}\times_{\vec{j}} \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

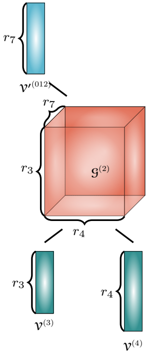
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \,{}_{\vec{i}}\!\times_{\vec{j}} \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

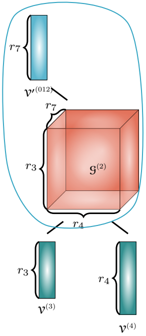
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
    - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \;{}_{\vec{i}}\!\times_{\vec{j}} \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

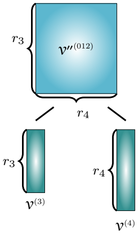
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \;{}_{\vec{i}}\times_{\vec{j}}\, \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

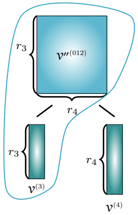
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

$$\mathcal{V}^{(i)} \; {}_{\vec{i}}\times_{\vec{j}} \; \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

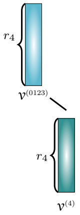
# Functional Tensor Networks – Topologies

A tensor network is a connected graph

$$\mathcal{TN} = (V, E)$$

- each vertex $\mathcal{V}^{(i)} \in V$ is a tensor of order $d^{(i)}$
- the set of edges $E$ denote contractions
  - An edge $E^{(ij)}$ from vertex $\mathcal{V}^{(i)}$ to vertex $\mathcal{V}^{(j)}$ is a pair of multi-indices $E^{(ij)} = \{\vec{i}, \vec{j}\}$ and denotes the contraction

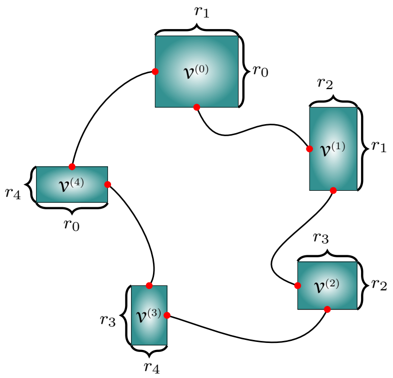
$$\mathcal{V}^{(i)} \ {}_{\vec{i}}\times_{\vec{j}} \mathcal{V}^{(j)}.$$




Here, $V = \{\mathcal{V}^{(0)}, \mathcal{V}^{(1)}, \ldots, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \ldots\}$

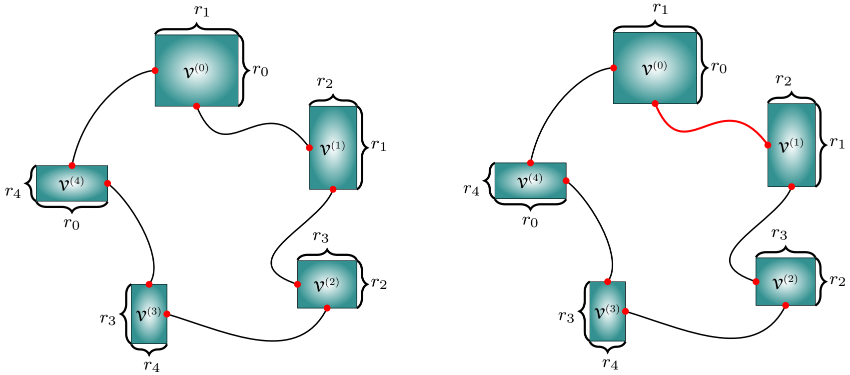
*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

# Functional Tensor Networks – Other Topologies



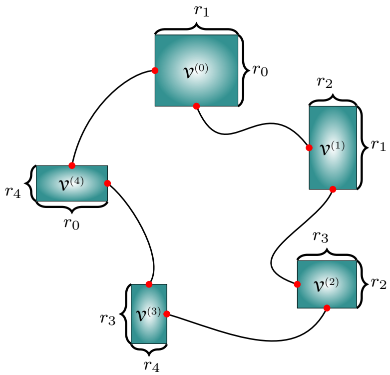

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

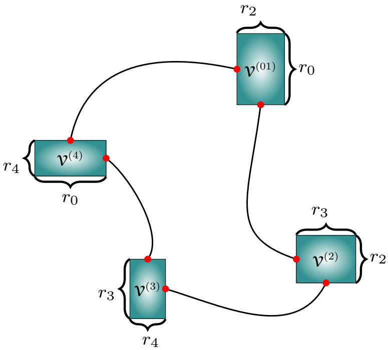
# Functional Tensor Networks – Other Topologies




*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

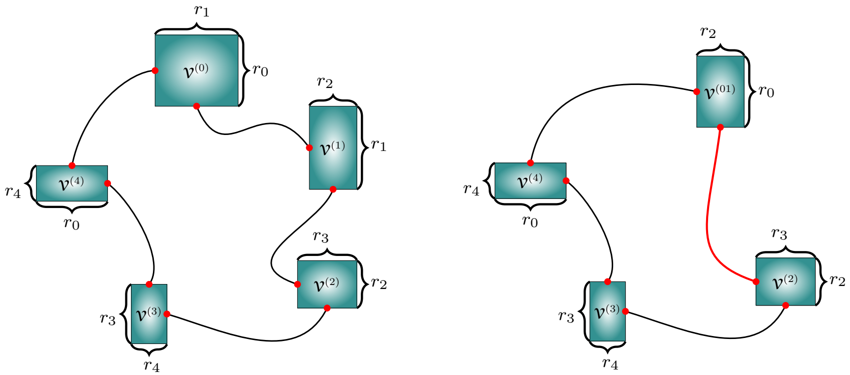
# Functional Tensor Networks – Other Topologies



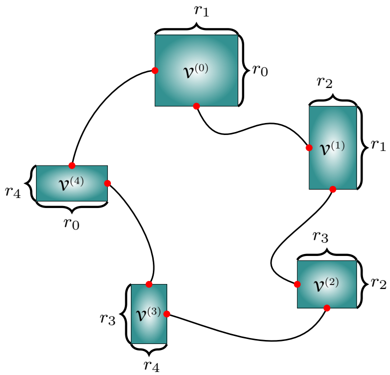

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

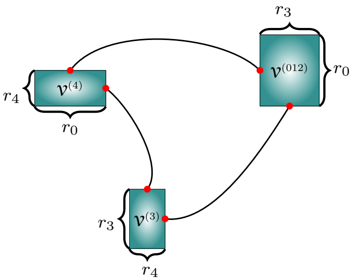
# Functional Tensor Networks – Other Topologies




*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

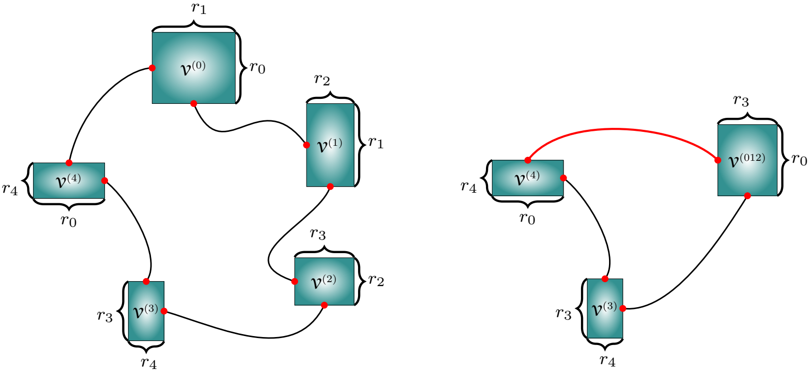
# Functional Tensor Networks – Other Topologies



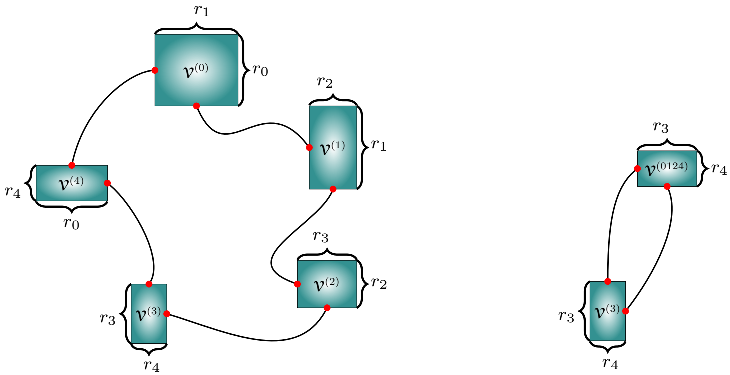

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

# Functional Tensor Networks – Other Topologies



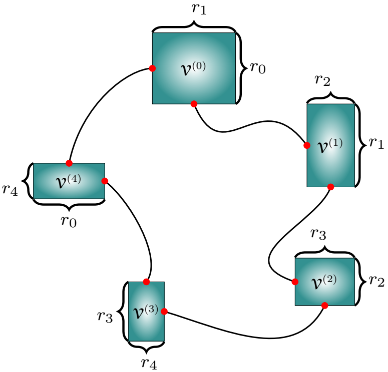

*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

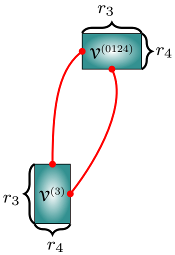
# Functional Tensor Networks – Other Topologies




*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

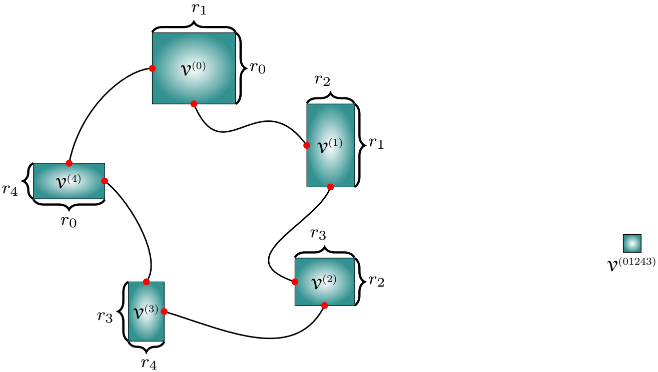
# Functional Tensor Networks – Other Topologies




*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

# Functional Tensor Networks – Other Topologies




*Full tensor network contraction consists of a set of recursive pairwise contractions until one vertex is left*

# Functional Representations – Univariate Functions

Linear Representations (e.g. polynomial chaos expansions)

$$f_{ij}^{(k)}(\lambda_k(\xi_k);\boldsymbol{\theta}_{ij}^{(k)}) = \sum_{l=0}^{p_k} \theta_{ijl}^{(k)} \Psi_l^{(k)}(\xi_k)$$

Non-Linear Representations (e.g. radial basis functions)

$$f_k^{(ij)}(\lambda_k;\boldsymbol{\theta}_k^{(ij)}) = \sum_{l=0}^{p_k} \theta_{k,l,1}^{(ij)} \exp(-\theta_{k,l,2}^{(ij)}(\lambda_k - \theta_{k,l,3}^{(ij)})^2)$$

# Functional Representations – Univariate Functions

Linear Representations (e.g. polynomial chaos expansions)

$$f_{ij}^{(k)}(\lambda_k(\xi_k);\boldsymbol{\theta}_{ij}^{(k)}) = \sum_{l=0}^{p_k} \theta_{ijl}^{(k)} \Psi_l^{(k)}(\xi_k)$$

Non-Linear Representations (e.g. radial basis functions)

$$f_k^{(ij)}(\lambda_k;\boldsymbol{\theta}_k^{(ij)}) = \sum_{l=0}^{p_k} \theta_{k,l,1}^{(ij)} \exp(-\theta_{k,l,2}^{(ij)}(\lambda_k - \theta_{k,l,3}^{(ij)})^2)$$

# Functional Tensor Networks - Evaluate Moments and Conditional Statistics

Each tensor core consists of scalars or univariate functions therefore contractions and integrals commute

## Expectation

$$\mathbb{E}[\mathcal{TN}] = (\mathbb{E}[\mathcal{V}], E)$$

where $\mathbb{E}[\mathcal{V}] \triangleq \{\mathbb{E}_{\lambda_0}[\mathcal{V}^{(0)}], \mathbb{E}_{\lambda_1}[\mathcal{V}^{(1)}], \ldots\}$

- For univariate functions given by polynomial chaos expansions, the elements of a 2D tensor $\mathbb{E}_{\lambda_k}[\mathcal{V}^{(k)}]$ are given by

$$\mathbb{E}_{\lambda_k}[\mathcal{V}^{(k)}(\lambda_k; \boldsymbol{\theta}_k)] = \begin{bmatrix} \theta^{(k)}_{110} & \theta^{(k)}_{120} & \cdots & \theta^{(k)}_{1\,r_k 0} \\ \theta^{(k)}_{210} & \theta^{(k)}_{220} & \cdots & \theta^{(k)}_{2\,r_k 0} \\ \vdots & \vdots & \ddots & \vdots \\ \theta^{(k)}_{r_{k-1}10} & \theta^{(k)}_{r_{k-1}20} & \cdots & \theta^{(k)}_{r_{k-1} r_k 0} \end{bmatrix}$$

- Conditional expectations $\mathbb{E}_i[\mathcal{TN}]$ require marginalization over subset $i$ of the set of tensor cores, e.g.

$$\mathbb{E}_1[\mathcal{V}] \triangleq \{\mathcal{V}^{(0)}, \mathbb{E}_{\lambda_1}[\mathcal{V}^{(1)}], \mathcal{V}^{(2)}, \ldots\}$$

R. Ballester-Ripoll *et al*, “Sobol tensor trains for global sensitivity analysis”, Reliability Engineering & System Safety 183 (2019): 311-322.

# Functional Tensor Networks - Evaluate Moments and Conditional Statistics

Variance

$$\mathbb{V}\mathrm{ar}[\mathcal{T}\mathcal{N}] = \mathbb{E}[(\mathcal{T}\mathcal{N})^2] - \mathbb{E}[\mathcal{T}\mathcal{N}]^2$$

The first term can be written as

$$\mathbb{E}[(\mathcal{T}\mathcal{N})^2] = \left(\mathbb{E}[\tilde{\mathcal{V}}], E\right)$$

where $\mathbb{E}[\tilde{\mathcal{V}}] \triangleq \{\mathbb{E}_{\lambda_0}[\mathcal{V}^{(0)} \otimes \mathcal{V}^{(0)}], \mathbb{E}_{\lambda_1}[\mathcal{V}^{(1)} \otimes \mathcal{V}^{(1)}], \ldots\}$

- For univariate functions given by polynomial chaos expansions, the elements of a 2D tensor $\mathbb{E}_{\lambda_k}[\mathcal{V}^{(k)} \otimes \mathcal{V}^{(k)}]$ are given by

$$\sum_{l=0}^{p_k} \theta^{(k)}_{i_1 j_1 l} \theta^{(k)}_{i_2 j_2 l} \langle \Psi^{(k)}_l (\xi_k)^2 \rangle$$

# Functional Tensor Networks - Sobol Indices

## Law of Total Variance

$$\mathbb{V}\mathrm{ar}[\mathcal{TN}] = \mathbb{V}\mathrm{ar}_i[\mathbb{E}_{\setminus i}[\mathcal{TN}]] + \mathbb{E}_i[\mathbb{V}\mathrm{ar}_{\setminus i}[\mathcal{TN}]]$$

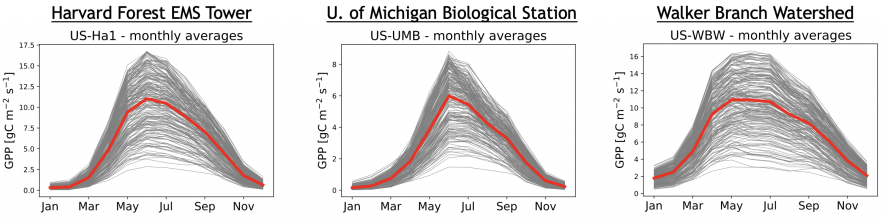
after normalization

$$1 = \underbrace{\frac{\mathbb{V}\mathrm{ar}_i[\mathbb{E}_{\setminus i}[\mathcal{TN}]]}{\mathbb{V}\mathrm{ar}[\mathcal{TN}]}}_{S_i} + \underbrace{\frac{\mathbb{E}_i[\mathbb{V}\mathrm{ar}_{\setminus i}[\mathcal{TN}]]}{\mathbb{V}\mathrm{ar}[\mathcal{TN}]}}_{S^T_{\setminus i}}$$

- First order $S_i$ and total order $S_i^T = 1 - S_{\setminus i}$ are computed using tensor network algebra described on previous slides.
- Joint sensitivity indices are evaluated through a similar approach

$$S_{ij} = \frac{\mathbb{V}\mathrm{ar}_{i,j}[\mathbb{E}_{\setminus i,j}[\mathcal{TN}]]}{\mathbb{V}\mathrm{ar}[\mathcal{TN}]} - S_i - S_j$$

# ELM Data – Simulations Corresponding to Select Observation sites




- *200 runs* corresponding to uniformly randomly sampled parameters over a *10D* parameter space
  - 160 training runs/40 validations runs
  - 8-fold cross validation over 160 training runs

# Functional Tensor Network Models – Training

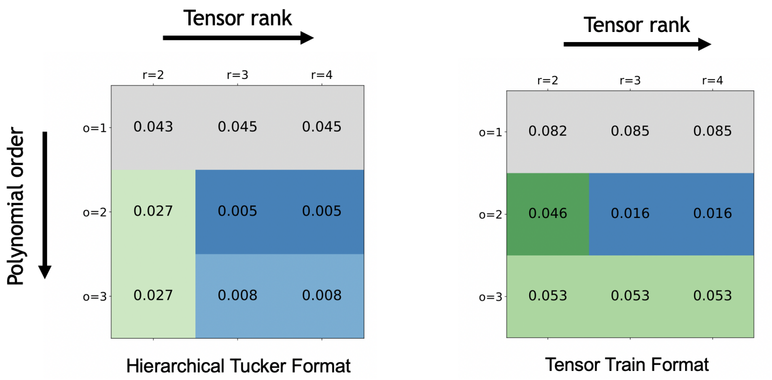
- Data split into 160 training runs / 40 validations runs
- Non-linear least squares with 8-fold cross validation over the training runs
- Univariate functions represented as polynomial expansions based on Legendre polynomials
  - Cross-validation to pick optimum regularization parameter, tensor rank, and polynomial order

$$\theta^* = \arg\min_{\theta} \left( \frac{1}{2} \sum_{i=1}^{N} \left( f(\lambda^{(i)}; \theta) - y^{(i)} \right)^2 + \alpha ||\theta||_2^2 \right)$$

  - Quality of fit assessed via mean-squared error (MSE)

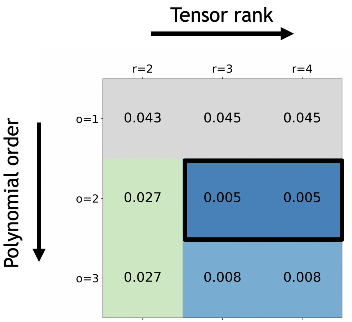
$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left( f(\lambda^{(i)}; \theta^*) - y^{(i)} \right)^2$$

# ELM Fit Results – FTN Models (in Hierarchical Tucker Format)

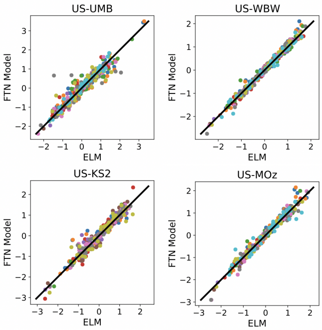



Site US-Ha1/June: Validation mean-squared error for Hierarchical Tucker models compared to Tensor Train models

# ELM Fit Results – FTN Models (in Hierarchical Tucker Format)




Site US-Ha1/June: Validation MSE




Validation data centered and normalized by the monthly standard deviation

# ELM Results: Variance-based GSA

Main Effect Sobol Index

$$S_i = \frac{Var[\mathbb{E}(f(\boldsymbol{\lambda}|\lambda_i)]}{Var[f(\boldsymbol{\lambda})]}$$

Total Effect Sobol Index

$$S_i^T = 1 - \frac{Var[\mathbb{E}(f(\boldsymbol{\lambda}|\lambda_{-i})]}{Var[f(\boldsymbol{\lambda})]}$$

| Parameter | March | | June | | September | | October | |
|---|---|---|---|---|---|---|---|---|
| | $S_i$ | $S_i^T$ | $S_i$ | $S_i^T$ | $S_i$ | $S_i^T$ | $S_i$ | $S_i^T$ |
| flnr | 0.70 | 0.72 | 0.80 | 0.83 | 0.84 | 0.86 | 0.76 | 0.77 |
| mbbopt | 0.01 | 0.02 | 0.09 | 0.13 | 0.04 | 0.06 | 0.02 | 0.02 |
| vcmaxse | 0.13 | 0.15 | 0.02 | 0.02 | 0 | 0 | 0.02 | 0.02 |
| dayl_scaling | 0.06 | 0.07 | 0 | 0 | 0.04 | 0.05 | 0.14 | 0.14 |

- fnlr (fraction of N in RuBisCO – CO2 conversion process)
- mbbopt (stomatal conductance slope – net CO2 flux)
- vcmaxse (entropy for photosynthetic parameters)
- dayl_scaling (day length scaling parameter)

# Closure

- Extended functional tensor train models to accommodate generic tensor network configurations
  - Expanded flexibility in capturing the structure of the original model
  - Efficient gradient computations through tensor network contractions
  - Alex Gorodetsky, CS, John Jakeman (2021)
    https://tinyurl.com/2p92thbn
- Functional tensor network models constructed via ridge regression are in good agreement with validation data for the driver application
  - Global Sensitivity Analysis results match subject matter expertise given the training runs available for this study
- Next steps: account for spatio-temporal dependencies and model calibration in a Bayesian setting.