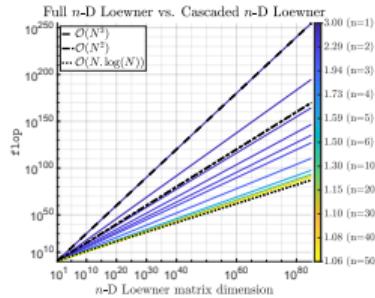
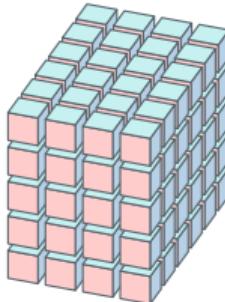


# The Loewner framework, the Kolmogorov superposition theorem, and the curse of dimensionality

... "(au delà de) l'ombre de Loewner", Marcel Berger.

C. Poussot-Vassal, in coll. with A.C. Antoulas [Rice Univ.], I.V. Goșea [MPI] and P. Vuillemin [ONERA]  
Workshop on data-driven control and analysis of dynamical systems

<https://arxiv.org/abs/2405.00495> (in SIAM Review - Research Spotlight)



# Forewords

## Starting (motivating) examples - Borehole function

$$\mathbf{H}({}^1x, \dots, {}^8x) = \mathbf{H}(r_w, r, T_u, H_u, T_l, H_l, L, K_w) = \frac{2\pi T_u (H_u - H_l)}{\ln\left(\frac{r}{r_w}\right) \left(1 + \frac{2LT_u}{\ln(r/r_w)r_w^2 K_w}\right) + \frac{T_u}{T_l}}$$

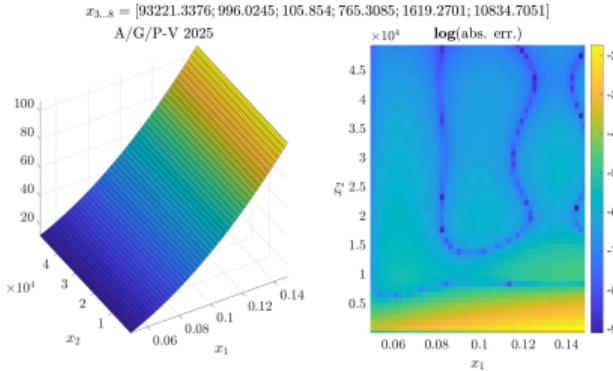


$$\begin{matrix} {}^1x \\ [r_w, \overline{r_w}] \end{matrix} \quad \times \quad \cdots \quad \times \quad \begin{matrix} {}^8x \\ [\underline{K_w}, \overline{K_w}] \end{matrix}$$

$\mathbf{tab}_8 \in \mathbb{C}^{8 \times 8 \times \dots \times 8}$

$\approx 130 \text{ Mo ('real')}$

#	Alg.	Parameters	Dim.	CPU [s]	RMSE	min err.	max err.
30	A1	1e-09,1	1.02e+04	19.3	0.00455	2e-09	0.061
	A2	1e-15,2	1.02e+04	39.1	0.00456	2.93e-09	0.0611



# Forewords

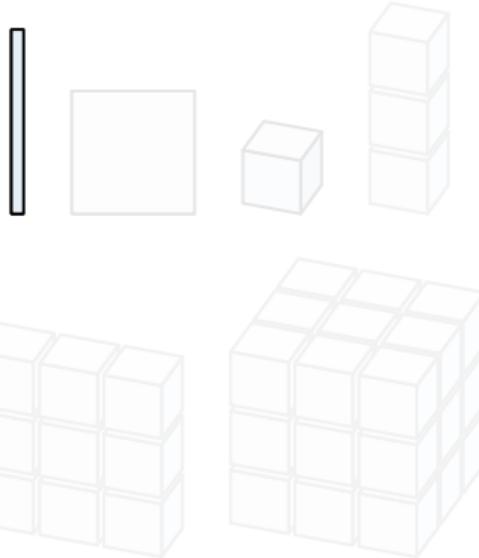
## Data (and tensors)

Column / Row data

$$\left. \begin{array}{c} {}^1\lambda_{j_1}, {}^1\mu_{i_1} \end{array} \right\} \xrightarrow{\mathbf{H}({}^1x)} \left\{ \begin{array}{c} \mathbf{w}_{j_1}, \mathbf{v}_{i_1} \end{array} \right.$$

${}^1x$	
${}^1\lambda_{1,\dots,k_1}$	$\mathbf{W}_{k_1}$
${}^1\mu_{1,\dots,q_1}$	$\mathbf{V}_{q_1}$

Tensors (1-D) tab<sub>1</sub>



# Forewords

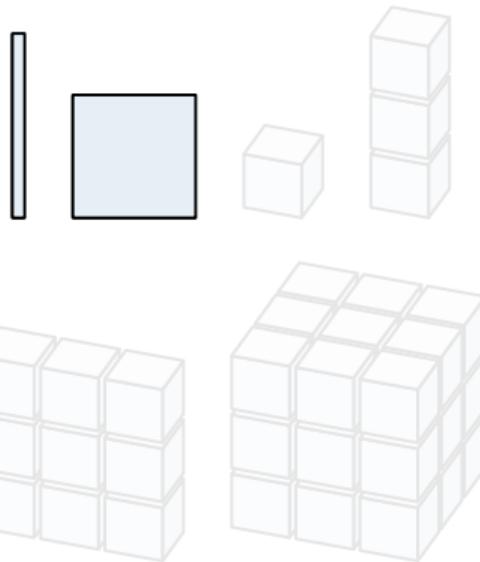
## Data (and tensors)

Column / Row data

$$\left. \begin{array}{l} {}^1\lambda_{j_1}, {}^1\mu_{i_1} \\ {}^2\lambda_{j_2}, {}^2\mu_{i_2} \end{array} \right\} \xrightarrow{\mathbf{H}({}^1x, {}^2x)} \left\{ \begin{array}{l} \mathbf{w}_{j_1, j_2}, \mathbf{v}_{i_1, i_2} \end{array} \right.$$

${}^1x$	${}^2x$	
${}^1\lambda_{1, \dots, k_1}$	$\mathbf{W}_{k_1, k_2}$	$\phi_{cr}$
${}^1\mu_{1, \dots, q_1}$	$\phi_{rc}$	$\mathbf{V}_{q_1, q_2}$

Tensors (2-D) tab<sub>2</sub>



# Forewords

## Data (and tensors)

### Column / Row data

$$\left. \begin{array}{l} {}^1\lambda_{j_1}, {}^1\mu_{i_1} \\ {}^2\lambda_{j_2}, {}^2\mu_{i_2} \\ {}^3\lambda_{j_3}, {}^3\mu_{i_3} \end{array} \right\} \xrightarrow{\mathbf{H}({}^1x, {}^2x, {}^3x)} \left\{ \begin{array}{l} \mathbf{w}_{j_1, j_2, j_3}, \mathbf{v}_{i_1, i_2, i_3} \end{array} \right.$$

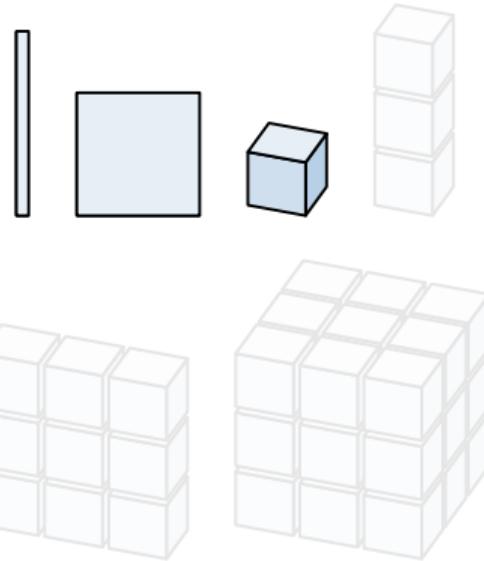
$${}^3x = {}^3\lambda_{1, \dots, k_3}$$

${}^1x$	${}^2x$	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$
${}^1\lambda_{1, \dots, k_1}$	$\mathbf{W}_{k_1, k_2, k_3}$	$\phi_{crc}$	
${}^1\mu_{1, \dots, q_1}$	$\phi_{rcc}$	$\phi_{rrc}$	

$${}^3x = {}^3\mu_{1, \dots, q_3}$$

${}^1x$	${}^2x$	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$
${}^1\lambda_{1, \dots, k_1}$	$\phi_{crr}$	$\phi_{crr}$	
${}^1\mu_{1, \dots, q_1}$	$\phi_{rcr}$	$\mathbf{V}_{q_1, q_2, q_3}$	

### Tensors (3-D) $\text{tab}_3$



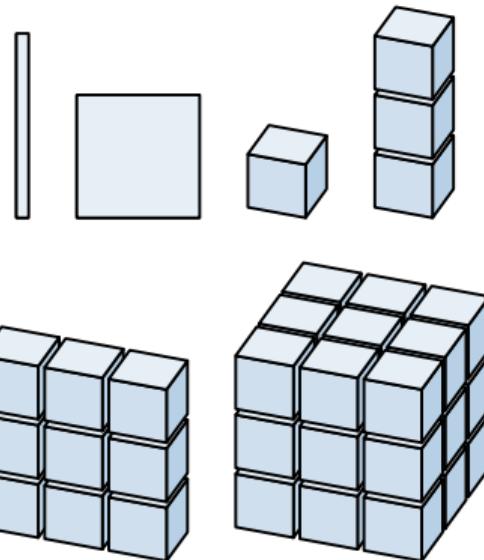
# Forewords

## Data (and tensors)

Column / Row data

$$\left. \begin{array}{l} {}^1\lambda_{j_1}, {}^1\mu_{i_1} \\ {}^2\lambda_{j_2}, {}^2\mu_{i_2} \\ {}^3\lambda_{j_3}, {}^3\mu_{i_3} \\ \vdots \\ {}^n\lambda_{j_n}, {}^n\mu_{i_n} \end{array} \right\} \xrightarrow{\mathbf{H}(^{1x, \dots, n}x)} \left\{ \mathbf{w}_{j_1, \dots, j_n}, \mathbf{v}_{i_1, \dots, i_n} \right.$$

Tensors ( $n$ -D)  $\text{tab}_n$



# Forewords

Contributions claim & trajectory of the presentation

## List of contributions

- ▶  $n$ -D tensor data to  $n$ -D Loewner matrix  $\mathbb{L}_n$
- ▶  $n$ -variable transfer functions
- ▶ Taming the curse of dimensionality
  - » in computation effort (flop)
  - » in storage needs (Bytes)
  - » in accuracy
- ▶  $n$ -variable decoupling
  - » KST formulation for rational functions
  - » connection with KAN
- ▶ Comparison with MLP, KAN, AAA



---

 A.C. Antoulas, I-V. Gosea, C. P-V and P. Vuillemin, "On the Loewner framework, the Kolmogorov superposition theorem, and the curse of dimensionality", SIAM Review, 2025.

 A.C. Antoulas, I-V. Gosea and C. P-V., "Tensor-based multivariate function approximation: methods benchmarking and comparison", <https://arxiv.org/abs/2506.04791>.

# Multi-variate data, function & Loewner matrix

1-D case

$$\begin{cases} P_c^{(1)} &:= \left\{ \left( {}^1\lambda_{j_1}; \mathbf{w}_{j_1} \right), j_1 = 1, \dots, k_1 \right\} \\ P_r^{(1)} &:= \left\{ \left( {}^1\mu_{i_1}; \mathbf{v}_{i_1} \right), i_1 = 1, \dots, q_1 \right\} \end{cases}$$

## Loewner matrix

$$\mathbb{L}_1 \in \mathbb{C}^{q_1 \times k_1}$$

$$(\mathbb{L}_1)_{i_1, j_1} = \frac{\mathbf{v}_{i_1} - \mathbf{w}_{j_1}}{}^1\mu_{i_1} - {}^1\lambda_{j_1}$$

## Lagrangian form

$$\mathbf{g}({}^1x) = \frac{\sum_{j_1=1}^{k_1} \frac{c_{j_1} \mathbf{w}_{j_1}}{{}^1x - {}^1\lambda_{j_1}}}{\sum_{j_1=1}^{k_1} \frac{c_{j_1}}{{}^1x - {}^1\lambda_{j_1}}}$$

## Null space

$$\text{span } (\mathbf{c}_1) = \mathcal{N}(\mathbb{L}_1)$$

$$\mathbf{c}_1 = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{k_1} \end{bmatrix} \in \mathbb{C}^{k_1}$$

# Multi-variate data, function & Loewner matrix

1-D case (example)

Data generated from  $\mathbf{H}(^1x) = \mathbf{H}(s) = (s^2 + 4)/(s + 1)$  of complexity (2)

$$\left. \begin{array}{rcl} {}^1\lambda_{j_1} & = & [1, 3, 5] \\ {}^1\mu_{i_1} & = & [2, 4, 6, 8] \end{array} \right\} \xrightarrow{\mathbf{H}} \left\{ \begin{array}{rcl} \mathbf{w}_{j_1} & = & [5/2, 13/4, 29/6] \\ \mathbf{v}_{i_1} & = & [8/3, 4, 40/7, 68/9] \end{array} \right.$$

## Loewner matrix

$$\mathbb{L}_1 = \begin{bmatrix} \frac{1}{6} & \frac{7}{12} & \frac{13}{18} \\ \frac{1}{2} & \frac{3}{4} & \frac{5}{6} \\ \frac{9}{14} & \frac{23}{28} & \frac{37}{42} \\ \frac{13}{18} & \frac{31}{36} & \frac{49}{54} \end{bmatrix}$$

## Null space

$$\mathbf{c}_1 = \begin{bmatrix} \frac{1}{3} \\ -\frac{4}{3} \\ 1 \end{bmatrix}$$

## Lagrangian form

$$\mathbf{g}(s) = \frac{\frac{5}{6(s-1)} - \frac{13}{3(s-3)} + \frac{29}{6(s-5)}}{\frac{1}{3(s-1)} - \frac{4}{3(s-3)} + \frac{1}{s-5}} = \mathbf{H}(s)$$

# Multi-variate data, function & Loewner matrix

2-D case

$$\begin{cases} P_c^{(2)} &:= \left\{ (\textcolor{brown}{1}\lambda_{j_1}, \textcolor{brown}{2}\lambda_{j_2}; \mathbf{w}_{j_1, j_2}), \ j_1 = 1, \dots, k_1 \quad j_2 = 1, \dots, k_2 \right\} \\ P_r^{(2)} &:= \left\{ (\textcolor{violet}{1}\mu_{i_1}, \textcolor{violet}{2}\mu_{i_2}; \mathbf{v}_{i_1, i_2}), \ i_1 = 1, \dots, q_1 \quad i_2 = 1, \dots, q_2 \right\} \end{cases}$$

## Loewner matrix

$$\mathbb{L}_2 \in \mathbb{C}^{q_1 q_2 \times k_1 k_2}$$

$$\ell_{j_1, j_2}^{i_1, i_2} = \frac{\mathbf{v}_{i_1, i_2} - \mathbf{w}_{j_1, j_2}}{(\textcolor{violet}{1}\mu_{i_1} - \textcolor{brown}{1}\lambda_{j_1})(\textcolor{violet}{2}\mu_{i_2} - \textcolor{brown}{2}\lambda_{j_2})}$$

## Lagrangian form

$$\mathbf{g}(\textcolor{violet}{1}x, \textcolor{brown}{2}x) = \frac{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \frac{c_{j_1, j_2} \mathbf{w}_{j_1, j_2}}{(\textcolor{violet}{1}x - \textcolor{brown}{1}\lambda_{j_1})(\textcolor{brown}{2}x - \textcolor{brown}{2}\lambda_{j_2})}}{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \frac{c_{j_1, j_2}}{(\textcolor{violet}{1}x - \textcolor{brown}{1}\lambda_{j_1})(\textcolor{brown}{2}x - \textcolor{brown}{2}\lambda_{j_2})}}$$

## Null space

$$\text{span } (\mathbf{c}_2) = \mathcal{N}(\mathbb{L}_2)$$

$$\mathbf{c}_2 = \begin{bmatrix} c_{1,1} \\ \vdots \\ \hline c_{1,k_2} \\ \vdots \\ \hline c_{k_1,1} \\ \vdots \\ c_{k_1,k_2} \end{bmatrix} \in \mathbb{C}^{k_1 k_2}$$

# Multi-variate data, function & Loewner matrix

## 2-D case (example)

Data generated from  $\mathbf{H}(^1x, ^2x) = \mathbf{H}(s, t) = (s^2t)/(s - t + 1)$  of complexity (2, 1)

$$\left. \begin{array}{l} {}^1\lambda_{j_1} = [1, 3, 5] \\ {}^1\mu_{i_1} = [0, 2, 4] \\ {}^2\lambda_{j_2} = [-1, -3] \\ {}^2\mu_{i_2} = [-2, -4] \end{array} \right\} \xrightarrow{\mathbf{H}} \left[ \begin{array}{ccc|cc} -\frac{1}{3} & -\frac{3}{5} & -\frac{1}{2} & -\frac{2}{3} \\ -\frac{9}{5} & -\frac{27}{7} & -3 & -\frac{9}{2} \\ -\frac{25}{7} & -\frac{25}{3} & -\frac{25}{4} & -10 \\ \hline 0 & 0 & 0 & 0 \\ -1 & -2 & -\frac{8}{5} & -\frac{16}{7} \\ -\frac{8}{3} & -6 & -\frac{32}{7} & -\frac{64}{9} \end{array} \right]$$

**Loewner matrix**

$$\mathbb{L}_2 = \left[ \begin{array}{cc|cc|cc} \frac{1}{3} & -\frac{3}{5} & \frac{3}{5} & -\frac{9}{7} & \frac{5}{7} & -\frac{5}{3} \\ \frac{1}{9} & \frac{3}{5} & \frac{1}{5} & \frac{9}{7} & \frac{5}{21} & \frac{5}{3} \\ \hline \frac{19}{15} & -1 & \frac{1}{5} & -\frac{79}{35} & \frac{23}{35} & -\frac{101}{45} \\ \hline \frac{41}{63} & \frac{59}{35} & -\frac{17}{105} & \frac{11}{7} & \frac{1}{7} & \frac{127}{63} \\ \frac{89}{63} & -\frac{139}{105} & \frac{97}{35} & -\frac{5}{7} & -1 & -\frac{79}{21} \\ \hline \frac{61}{81} & \frac{293}{135} & \frac{239}{135} & \frac{205}{63} & -\frac{223}{189} & \frac{11}{9} \end{array} \right]$$

**Null space**

$$\mathbf{c}_2 = \left[ \begin{array}{c} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{array} \right]$$

# Multi-variate data, function & Loewner matrix

## 2-D case (example)

Data generated from  $\mathbf{H}(^1x, ^2x) = \mathbf{H}(s, t) = (s^2t)/(s - t + 1)$  of complexity (2, 1)

$$\left. \begin{array}{lcl} {}^1\lambda_{j_1} & = & [1, 3, 5] \\ {}^1\mu_{i_1} & = & [0, 2, 4] \\ {}^2\lambda_{j_2} & = & [-1, -3] \\ {}^2\mu_{i_2} & = & [-2, -4] \end{array} \right\} \xrightarrow{\mathbf{H}} \left[ \begin{array}{ccc|cc} -\frac{1}{3} & -\frac{3}{5} & -\frac{1}{2} & -\frac{2}{3} \\ -\frac{9}{5} & -\frac{27}{7} & -3 & -\frac{9}{2} \\ -\frac{25}{7} & -\frac{25}{3} & -\frac{25}{4} & -10 \\ \hline 0 & 0 & 0 & 0 \\ -1 & -2 & -\frac{8}{5} & -\frac{16}{7} \\ -\frac{8}{3} & -6 & -\frac{32}{7} & -\frac{64}{9} \end{array} \right]$$

### Lagrangian form

$$\mathbf{g}(s, t) = -\frac{\frac{1}{9(s-1)(t+1)} - \frac{1}{3(s-1)(t+3)} - \frac{2}{(s-3)(t+1)} + \frac{6}{(s-3)(t+3)} + \frac{25}{9(s-5)(t+1)} - \frac{25}{3(s-5)(t+3)}}{\frac{1}{3(s-1)(t+1)} - \frac{5}{9(s-1)(t+3)} - \frac{10}{9(s-3)(t+1)} + \frac{14}{9(s-3)(t+3)} + \frac{7}{9(s-5)(t+1)} - \frac{1}{(s-5)(t+3)}} = \mathbf{H}(s, t)$$

# Multi-variate data, function & Loewner matrix

*n-D case*

$$\begin{cases} P_c^{(n)} := \left\{ (^1\lambda_{j_1}, ^2\lambda_{j_2}, \dots, ^n\lambda_{j_n}; \mathbf{w}_{j_1, j_2, \dots, j_n}), \quad j_l = 1, \dots, k_l, \quad l = 1, \dots, n \right\} \\ P_r^{(n)} := \left\{ (^1\mu_{i_1}, ^2\mu_{i_2}, \dots, ^n\mu_{i_n}; \mathbf{v}_{i_1, i_2, \dots, i_n}), \quad i_l = 1, \dots, q_l, \quad l = 1, \dots, n \right\} \end{cases}$$

## Loewner matrix

$$\mathbb{L}_n \in \mathbb{C}^{q_1 q_2 \cdots q_n \times k_1 k_2 \cdots k_n}$$

$$\ell_{j_1, j_2, \dots, j_n}^{i_1, i_2, \dots, i_n} = \frac{\mathbf{v}_{i_1, i_2, \dots, i_n} - \mathbf{w}_{j_1, j_2, \dots, j_n}}{(^1\mu_{i_1} - ^1\lambda_{j_1}) \cdots (^n\mu_{i_n} - ^n\lambda_{j_n})}$$

## Lagrangian form

$$\mathbf{g}(^1x, \dots, ^nx) = \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \frac{c_{j_1, \dots, j_n} \mathbf{w}_{j_1, \dots, j_n}}{(^1x - ^1\lambda_{j_1}) \cdots (^n x - ^n\lambda_{j_n})}}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \frac{c_{j_1, \dots, j_n}}{(^1x - ^1\lambda_{j_1}) \cdots (^n x - ^n\lambda_{j_n})}}$$

## Null space

$$\text{span } (\mathbf{c}_n) = \mathcal{N}(\mathbb{L}_n)$$

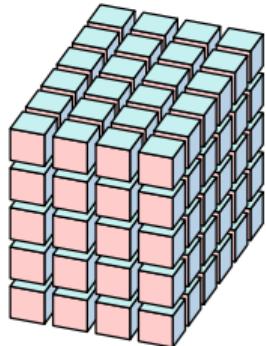
$$\mathbf{c}_n = \begin{bmatrix} c_{1, \dots, 1} \\ \vdots \\ \frac{c_{1, \dots, k_n}}{\vdots} \\ c_{k_1, \dots, 1} \\ \vdots \\ c_{k_1, \dots, k_n} \end{bmatrix} \in \mathbb{C}^{k_1 \cdots k_n}$$

# Taming the curse of dimensionality

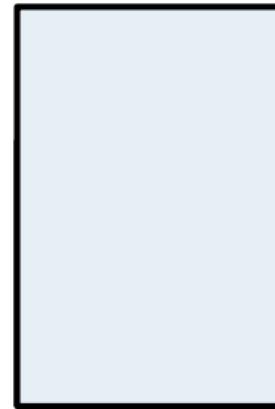
*n*-variable Loewner matrix operator

$$\begin{aligned} \mathbb{C}^{k_1} \times \mathbb{C}^{q_1} \times \dots \times \mathbb{C}^{k_n} \times \mathbb{C}^{q_n} \times \mathbb{C}^{(k_1+q_1) \times \dots \times (k_n+q_n)} &\longrightarrow \mathbb{C}^{Q \times K} \\ (\textcolor{brown}{^1}\lambda_{j_1}, \textcolor{pink}{^1}\mu_{i_1}, \dots, \textcolor{brown}{^n}\lambda_{j_n}, \textcolor{pink}{^n}\mu_{i_n}, \mathbf{tab}_n) &\longmapsto \mathbb{L}_n \end{aligned}$$

*n*-D tensor  $\mathbf{tab}_n$

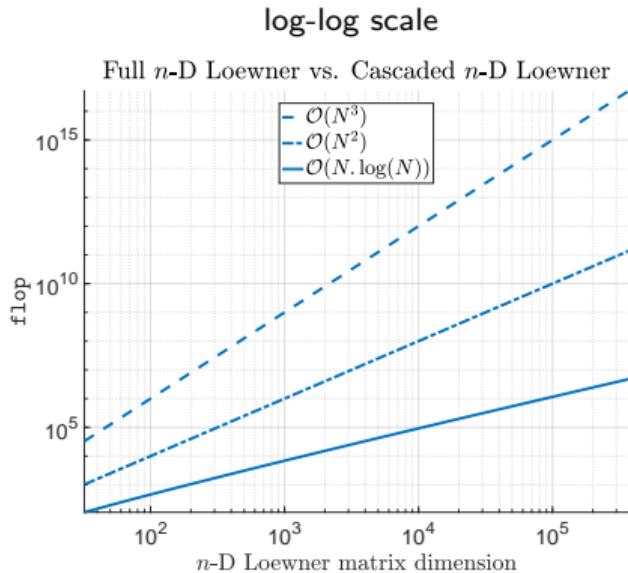


matrix  $\mathbb{L}_n$



# Taming the curse of dimensionality

Null space flop and memory issues



Let (rows)  $Q = q_1 q_2 \dots q_n$  and (columns)  
 $K = k_1 k_2 \dots k_n$

$$\mathbb{L}_n \in \mathbb{C}^{Q \times K}$$

## Computational issue

Note that  $Q \times K$  matrix SVD flop estimation is

- ▶  $QK^2$  (if  $Q > K$ )
- ▶  $N^3$  (if  $Q = K = N$ )

## Storage issue

Note that  $Q \times K$  matrix storage estimation is

- ▶ in real double
- ▶ in complex double

# Taming the curse of dimensionality

2-D case (example cont'd, to get the idea)

Data generated from  $\mathbf{H}(^1x, ^2x) = \mathbf{H}(s, t) = (s^2t)/(s - t + 1)$  of complexity (2, 1)

${}^2x$	${}^2\lambda_1 = -1$	${}^2\lambda_2 = -3$	${}^2\mu_1 = -2$	${}^2\mu_2 = -4$
${}^1x$	$h_{1,1} = -\frac{1}{3}$	$h_{1,2} = -\frac{3}{5}$	$h_{1,3} = -\frac{1}{2}$	$h_{1,4} = -\frac{2}{3}$
${}^1\lambda_1 = 1$	$h_{2,1} = -\frac{9}{5}$	$h_{2,2} = -\frac{27}{7}$	$h_{2,3} = -3$	$h_{2,4} = -\frac{9}{2}$
${}^1\lambda_2 = 3$	$h_{3,1} = -\frac{25}{7}$	$h_{3,2} = -\frac{25}{3}$	$h_{3,3} = -\frac{25}{4}$	$h_{3,4} = -10$
${}^1\lambda_3 = 5$	$h_{4,1} = 0$	$h_{4,2} = 0$	$h_{4,3} = 0$	$h_{4,4} = 0$
${}^1\mu_1 = 0$	$h_{5,1} = -1$	$h_{5,2} = -2$	$h_{5,3} = -\frac{8}{5}$	$h_{5,4} = -\frac{16}{7}$
${}^1\mu_2 = 2$	$h_{6,1} = -\frac{8}{3}$	$h_{6,2} = -6$	$h_{6,3} = -\frac{32}{7}$	$h_{6,4} = -\frac{64}{9}$
${}^1\mu_3 = 4$				

$$\xrightarrow{\mathcal{N}(\mathbb{L}_2)} \mathbf{c}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{bmatrix}$$

- 1 L<sub>1</sub> along  ${}^1x$ , for  
 ${}^2x = {}^2\lambda_2 = -3$
- 3 L<sub>1</sub> along  ${}^2x$  for  
 ${}^2x = \{{}^1\lambda_1, {}^1\lambda_2, {}^1\lambda_3\}$
- Scaled null space  $\mathbf{c}_2^\top =$   

$$[ \mathbf{c}_1^{{}^1\lambda_1} \cdot [\mathbf{c}_1^{{}^2\lambda_2}]_1 \quad \mathbf{c}_1^{{}^1\lambda_2} \cdot [\mathbf{c}_1^{{}^2\lambda_2}]_2 \quad \mathbf{c}_1^{{}^1\lambda_3} \cdot [\mathbf{c}_1^{{}^2\lambda_2}]_3 ]^\top$$

# Taming the curse of dimensionality

2-D case (example cont'd, to get the idea)

Data generated from  $\mathbf{H}(^1x, ^2x) = \mathbf{H}(s, t) = (s^2t)/(s - t + 1)$  of complexity (2, 1)

${}^2x$	${}^2\lambda_1 = -1$	${}^2\lambda_2 = -3$	${}^2\mu_1 = -2$	${}^2\mu_2 = -4$
${}^1x$				
${}^1\lambda_1 = 1$	$h_{1,1} = -\frac{1}{3}$	$h_{1,2} = -\frac{3}{5}$	$h_{1,3} = -\frac{1}{2}$	$h_{1,4} = -\frac{2}{3}$
${}^1\lambda_2 = 3$	$h_{2,1} = -\frac{9}{5}$	$h_{2,2} = -\frac{27}{7}$	$h_{2,3} = -3$	$h_{2,4} = -\frac{9}{2}$
${}^1\lambda_3 = 5$	$h_{3,1} = -\frac{25}{7}$	$h_{3,2} = -\frac{25}{3}$	$h_{3,3} = -\frac{25}{4}$	$h_{3,4} = -10$
${}^1\mu_1 = 0$	$h_{4,1} = 0$	$h_{4,2} = 0$	$h_{4,3} = 0$	$h_{4,4} = 0$
${}^1\mu_2 = 2$	$h_{5,1} = -1$	$h_{5,2} = -2$	$h_{5,3} = -\frac{8}{5}$	$h_{5,4} = -\frac{16}{7}$
${}^1\mu_3 = 4$	$h_{6,1} = -\frac{8}{3}$	$h_{6,2} = -6$	$h_{6,3} = -\frac{32}{7}$	$h_{6,4} = -\frac{64}{9}$

$$\xrightarrow{\mathcal{N}(\mathbb{L}_2)} \mathbf{c}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{bmatrix}$$

- 1  $\mathbb{L}_1$  along  ${}^1x$ , for

$${}^2x = {}^2\lambda_2 = -3$$

- 3  $\mathbb{L}_1$  along  ${}^2x$  for

$${}^1x = \{{}^1\lambda_1, {}^1\lambda_2, {}^1\lambda_3\}$$

- Scaled null space  $\mathbf{c}_2^\top =$

$$\left[ \frac{{}^1\lambda_1}{{}^2\lambda_2} \cdot [{}^2\lambda_2]_1 \quad \frac{{}^1\lambda_2}{{}^2\lambda_2} \cdot [{}^2\lambda_2]_2 \quad \frac{{}^1\lambda_3}{{}^2\lambda_2} \cdot [{}^2\lambda_2]_3 \right]^\top$$

$$\mathbf{c}_1^{{}^2\lambda_2} = \begin{bmatrix} \frac{5}{9} \\ -\frac{14}{9} \\ 1 \end{bmatrix}$$

# Taming the curse of dimensionality

2-D case (example cont'd, to get the idea)

Data generated from  $\mathbf{H}(^1x, ^2x) = \mathbf{H}(s, t) = (s^2t)/(s - t + 1)$  of complexity (2, 1)

${}^2x$	${}^2\lambda_1 = -1$	${}^2\lambda_2 = -3$	${}^2\mu_1 = -2$	${}^2\mu_2 = -4$
${}^1x$	$h_{1,1} = -\frac{1}{3}$	$h_{1,2} = -\frac{3}{5}$	$h_{1,3} = -\frac{1}{2}$	$h_{1,4} = -\frac{2}{3}$
${}^1\lambda_1 = 1$	$h_{2,1} = -\frac{9}{5}$	$h_{2,2} = -\frac{27}{7}$	$h_{2,3} = -3$	$h_{2,4} = -\frac{9}{2}$
${}^1\lambda_2 = 3$	$h_{3,1} = -\frac{25}{7}$	$h_{3,2} = -\frac{25}{3}$	$h_{3,3} = -\frac{25}{4}$	$h_{3,4} = -10$
${}^1\lambda_3 = 5$	$h_{4,1} = 0$	$h_{4,2} = 0$	$h_{4,3} = 0$	$h_{4,4} = 0$
${}^1\mu_1 = 0$	$h_{5,1} = -1$	$h_{5,2} = -2$	$h_{5,3} = -\frac{8}{5}$	$h_{5,4} = -\frac{16}{7}$
${}^1\mu_2 = 2$	$h_{6,1} = -\frac{8}{3}$	$h_{6,2} = -6$	$h_{6,3} = -\frac{32}{7}$	$h_{6,4} = -\frac{64}{9}$
${}^1\mu_3 = 4$				

$$\xrightarrow{\mathcal{N}(\mathbb{L}_2)} \mathbf{c}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{bmatrix}$$

- 1  $\mathbb{L}_1$  along  ${}^1x$ , for

$${}^2x = {}^2\lambda_2 = -3$$

- 3  $\mathbb{L}_1$  along  ${}^2x$  for

$${}^1x = \{{}^1\lambda_1, {}^1\lambda_2, {}^1\lambda_3\}$$

- Scaled null space  $\mathbf{c}_2^\top =$

$$\left[ {}^1\lambda_1 \cdot [{}^2\lambda_2]_1 \quad {}^1\lambda_2 \cdot [{}^2\lambda_2]_2 \quad {}^1\lambda_3 \cdot [{}^2\lambda_2]_3 \right]^\top$$

$$\mathbf{c}_1^{{}^2\lambda_2} = \begin{bmatrix} \frac{5}{9} \\ -\frac{14}{9} \\ 1 \end{bmatrix} \text{ and } \mathbf{c}_1^{{}^1\lambda_1} = \begin{bmatrix} -\frac{3}{5} \\ 1 \end{bmatrix}, \mathbf{c}_1^{{}^1\lambda_2} = \begin{bmatrix} -\frac{5}{7} \\ 1 \end{bmatrix}, \mathbf{c}_1^{{}^1\lambda_3} = \begin{bmatrix} -\frac{7}{9} \\ 1 \end{bmatrix}$$

# Taming the curse of dimensionality

2-D case (example cont'd, to get the idea)

Data generated from  $\mathbf{H}(^1x, ^2x) = \mathbf{H}(s, t) = (s^2t)/(s - t + 1)$  of complexity (2, 1)

${}^2x$	${}^2\lambda_1 = -1$	${}^2\lambda_2 = -3$	${}^2\mu_1 = -2$	${}^2\mu_2 = -4$
${}^1x$	$h_{1,1} = -\frac{1}{3}$	$h_{1,2} = -\frac{3}{5}$	$h_{1,3} = -\frac{1}{2}$	$h_{1,4} = -\frac{2}{3}$
${}^1\lambda_1 = 1$	$h_{2,1} = -\frac{9}{5}$	$h_{2,2} = -\frac{27}{7}$	$h_{2,3} = -3$	$h_{2,4} = -\frac{9}{2}$
${}^1\lambda_2 = 3$	$h_{3,1} = -\frac{25}{7}$	$h_{3,2} = -\frac{25}{3}$	$h_{3,3} = -\frac{25}{4}$	$h_{3,4} = -10$
${}^1\lambda_3 = 5$	$h_{4,1} = 0$	$h_{4,2} = 0$	$h_{4,3} = 0$	$h_{4,4} = 0$
${}^1\mu_1 = 0$	$h_{5,1} = -1$	$h_{5,2} = -2$	$h_{5,3} = -\frac{8}{5}$	$h_{5,4} = -\frac{16}{7}$
${}^1\mu_2 = 2$	$h_{6,1} = -\frac{8}{3}$	$h_{6,2} = -6$	$h_{6,3} = -\frac{32}{7}$	$h_{6,4} = -\frac{64}{9}$
${}^1\mu_3 = 4$				

$$\xrightarrow{\mathcal{N}(\mathbb{L}_2)} \mathbf{c}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{bmatrix}$$

- 1  $\mathbb{L}_1$  along  ${}^1x$ , for

$${}^2x = {}^2\lambda_2 = -3$$

- 3  $\mathbb{L}_1$  along  ${}^2x$  for

$${}^1x = \{{}^1\lambda_1, {}^1\lambda_2, {}^1\lambda_3\}$$

- Scaled null space  $\mathbf{c}_2^\top =$

$$[ \mathbf{c}_1^{{}^1\lambda_1} \cdot [\mathbf{c}_1^{{}^2\lambda_2}]_1 \quad \mathbf{c}_1^{{}^1\lambda_2} \cdot [\mathbf{c}_1^{{}^2\lambda_2}]_2 \quad \mathbf{c}_1^{{}^1\lambda_3} \cdot [\mathbf{c}_1^{{}^2\lambda_2}]_3 ]^\top$$

$$\mathbf{c}_1^{{}^2\lambda_2} = \begin{bmatrix} \frac{5}{9} \\ -\frac{14}{9} \\ 1 \end{bmatrix} \text{ and } \mathbf{c}_1^{{}^1\lambda_1} = \begin{bmatrix} -\frac{3}{5} \\ 1 \end{bmatrix}, \mathbf{c}_1^{{}^1\lambda_2} = \begin{bmatrix} -\frac{5}{7} \\ 1 \end{bmatrix}, \mathbf{c}_1^{{}^1\lambda_3} = \begin{bmatrix} -\frac{7}{9} \\ 1 \end{bmatrix}$$

# Taming the curse of dimensionality

2-D case

## Theorem: 2-D to 1-D

Being given the tableau  $\text{tab}_2$  tensor in response of the 2-variables  $\mathbf{H}({}^1x, {}^2x)$  function, the null space of the corresponding 2-D Loewner matrix  $\mathbb{L}_2$ , is spanned by

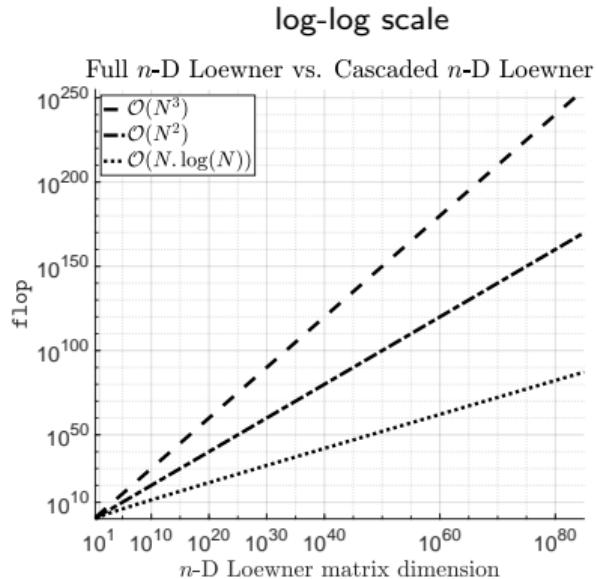
$$\mathcal{N}(\mathbb{L}_2) = \text{vec} \left[ \mathbf{c}_1^{{}^2\lambda_1} \cdot \left[ \mathbf{c}_1^{{}^1\lambda_{k_1}} \right]_1, \dots, \mathbf{c}_1^{{}^2\lambda_{k_2}} \cdot \left[ \mathbf{c}_1^{{}^1\lambda_{k_1}} \right]_{k_2} \right],$$

where

- ▶  $\mathbf{c}_1^{{}^1\lambda_{k_1}} = \mathcal{N}(\mathbb{L}_1^{{}^1\lambda_{k_1}})$ ,  
i.e. the null space of the **1-D Loewner matrix** for frozen  ${}^1x = {}^1\lambda_{k_1}$ , and
- ▶  $\mathbf{c}_1^{{}^2\lambda_{j_2}} = \mathcal{N}(\mathbb{L}_1^{{}^2\lambda_{j_2}})$ ,  
i.e. the  $j_1$ -th null space of the **1-D Loewner matrices** for frozen  ${}^2x = \{{}^2\lambda_1, \dots, {}^2\lambda_{k_2}\}$ .

# Taming the curse of dimensionality

Null space - flop complexity



## Null space underlying problem

Let (rows)  $Q = q_1 q_2 \dots q_n$  and (columns)  $K = k_1 k_2 \dots k_n$   
 $\mathbb{L}_n \in \mathbb{C}^{Q \times K}$

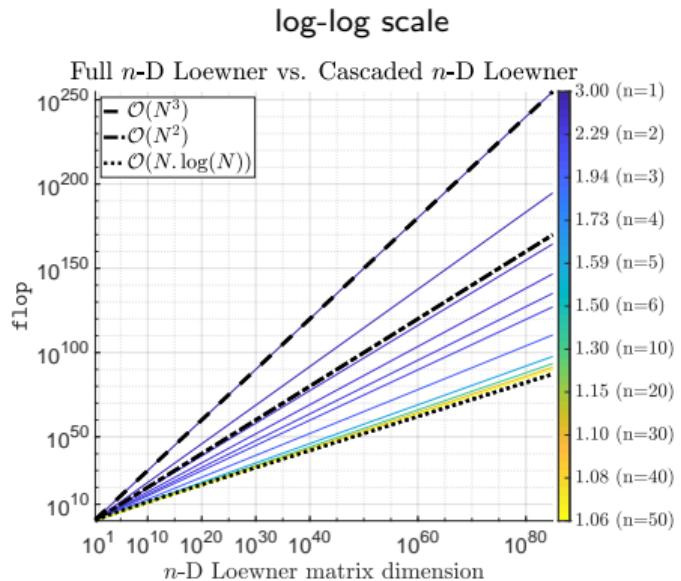
Note that  $Q \times K$  matrix SVD flop estimation is

- ▶  $QK^2$  (if  $Q > K$ )
- ▶  $N^3$  (if  $Q = K = N$ )

⇒ The CURSE of dimensionality

# Taming the curse of dimensionality

Null space - flop complexity



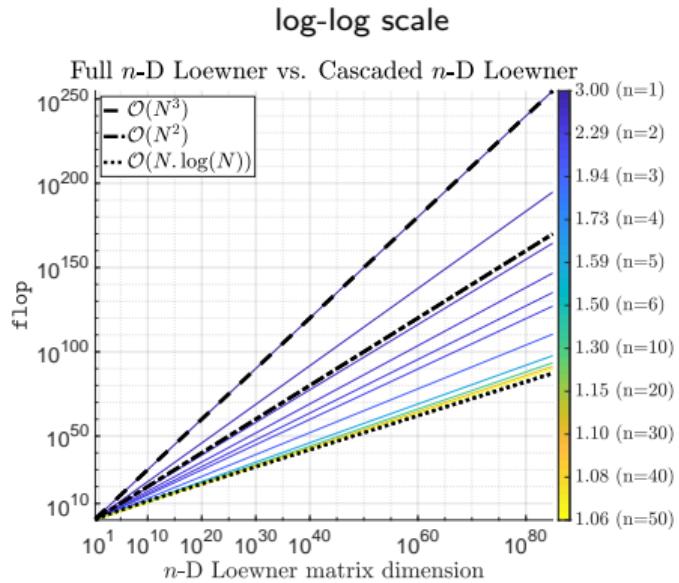
**Theorem: Recursive complexity**

$$\text{flop}_1(n) = \sum_{j=1}^n \left( k_j^3 \prod_{l=1}^j k_{l-1} \right) \text{ where } k_0 = 1.$$

⇒ The CURSE of dimensionality is TAMED

# Taming the curse of dimensionality

Null space - flop complexity



**Corollary: Worst case complexity**

$k$  interpolation points per variables.

$$\overline{\text{flop}_1} = k^3 \frac{1 - k^n}{1 - k} = k^3 \frac{1 - N}{1 - k},$$

which is a ( $n$  finite) geometric series of ratio  $k$ .

⇒ The CURSE of dimensionality is TAMED

$$\begin{aligned} \mathcal{O}(N^3) &\rightarrow \mathcal{O}(N^{2.29}) && \text{for } n = 2 \\ &\rightarrow \mathcal{O}(N^{1.94}) && \text{for } n = 3 \\ &\vdots && \\ &\rightarrow \mathcal{O}(N^{1.5}) && \text{for } n = 6 \\ &\vdots && \\ &\rightarrow \mathcal{O}(N^{1.06}) && \text{for } n = 50 \end{aligned}$$

# Taming the curse of dimensionality

Null space - memory

With similar importance, the **data storage is a key element** in the curse of dimensionality. The data (tableau) storage is (complex and double precision)

$$\frac{8}{2^{20}} \prod_l^n q_l + k_l \text{ MB} \text{ (example tableau } 2 \cdot [20, 6, 4, 6, 8, 2] = 2 \cdot [k_1, k_2, k_3, k_4, k_5, k_6] \text{ needs } 45 \text{ MB})$$

## Full $n$ -D Loewner

Construction of

$$\mathbb{L}_n \in \mathbb{C}^{N \times N}$$

where  $N = k_1 k_2 \cdots k_n$ , needs

$$\frac{8}{2^{20}} N^2 \text{ MB}$$

Example:  $N = 46,080$

Memory: **31.64 GB**

flop:  $9.78 \cdot 10^{13}$

# Taming the curse of dimensionality

Null space - memory

With similar importance, the **data storage** is a key element in the curse of dimensionality. The data (tableau) storage is (complex and double precision)

$$\frac{8}{2^{20}} \prod_l^n q_l + k_l \text{ MB} \text{ (example tableau } 2 \cdot [20, 6, 4, 6, 8, 2] = 2 \cdot [k_1, k_2, k_3, k_4, k_5, k_6] \text{ needs 45 MB)}$$

## Full $n$ -D Loewner

Construction of

$$\mathbb{L}_n \in \mathbb{C}^{N \times N}$$

where  $N = k_1 k_2 \cdots k_n$ , needs

$$\frac{8}{2^{20}} N^2 \text{ MB}$$

Example:  $N = 46,080$

Memory: **31.64 GB**

flop:  $9.78 \cdot 10^{13}$

## Cascaded $n$ -D Loewner

Construction of

$$\mathbb{L}_1 \in \mathbb{C}^{\bar{k} \times \bar{k}}$$

where  $\bar{k} = \max_j k_j$ , needs

$$\frac{8}{2^{20}} \bar{k}^2 \text{ MB}$$

Example:  $\bar{k} = 20$

Memory: **6.25 KB**

flop:  $8.13 \cdot 10^5$

# Variables decoupling, KST and KANs

Loewner, Kolmogorov Superposition Theorem and Hilbert's 13th problem

## Variable decoupling

Given data  $\text{tab}_n$ , the latter achieves variables decoupling, and the null space can be equivalently written as:

$$\mathbf{c}_n = \underbrace{\mathbf{c}^{n_x}}_{\text{Bary}^{(n_x)}} \odot \underbrace{(\mathbf{c}^{n-1_x} \otimes \mathbf{1}_{k_n})}_{\text{Bary}^{(n-1_x)}} \odot \underbrace{(\mathbf{c}^{n-2_x} \otimes \mathbf{1}_{k_n k_{n-1}})}_{\text{Bary}^{(n-2_x)}} \odot \cdots \odot \underbrace{(\mathbf{c}^1_x \otimes \mathbf{1}_{k_n \dots k_2})}_{\text{Bary}^{(1_x)}}.$$

where  $\mathbf{c}^l_x$  denotes the vectorized barycentric coefficients related to the  $l$ -th variable.

"Kolmogorov proved that every continuous function of several variables can be represented as a superposition of continuous functions of one variable and the operation of addition (1957). Thus, it is as if there are no functions of several variables at all. Seriously speaking, Kolmogorov's theorem is a brilliant example of his mastery. In particular, the theorem shows that Hilbert's conjecture (to its 13th problem) is wrong."



A.G. Vitushkin, "On Hilbert's thirteenth problem and related questions", Russian Math. Surveys 59:1, pp. 11-25.

# Variables decoupling, KST and KANs

KANs via Loewner with rational activation functions ( $\mathbf{H} = {}^1x \cdot {}^2x$ )

$$\begin{aligned} {}^1\lambda_{j_1} &= \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \\ {}^2\lambda_{j_2} &= \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \end{aligned}$$

$$\left( \begin{array}{ccc|c} \mathbf{c} & \mathbf{w} & \mathbf{c} \cdot \mathbf{w} & \text{Lag} \\ 1.0 & 1.0 & 1.0 & \frac{1}{({}^1x+1.0)({}^2x+1.0)} \\ -1.0 & -1.0 & 1.0 & \frac{1}{({}^1x+1.0)({}^2x-1.0)} \\ -1.0 & -1.0 & 1.0 & \frac{1}{({}^1x-1.0)({}^2x+1.0)} \\ 1.0 & 1.0 & 1.0 & \frac{1}{({}^1x-1.0)({}^2x-1.0)} \end{array} \right)$$

$$\begin{aligned} \mathbf{c}^{{}^2x} &= \text{vec} \begin{pmatrix} -1.0 & -1.0 \\ 1.0 & 1.0 \end{pmatrix} \\ \mathbf{c}^{{}^1x} &= \begin{pmatrix} -1.0 \\ 1.0 \end{pmatrix} \end{aligned}$$

$$\mathbf{c}_2 = \mathbf{c}^{{}^2x} \odot (\mathbf{c}^{{}^1x} \otimes \mathbf{1}_{k_2})$$

$$\mathbf{D} = \begin{pmatrix} \mathbf{c}^{{}^1x} \cdot \text{Lag}({}^1x) & \mathbf{c}^{{}^2x} \cdot \text{Lag}({}^2x) \\ \frac{1.0}{{}^1x+1.0} & \frac{1.0}{{}^2x+1.0} \\ \frac{1.0}{{}^1x+1.0} & \frac{1}{{}^2x-1.0} \\ \frac{1}{{}^1x-1.0} & \frac{1.0}{{}^2x+1.0} \\ \frac{1}{{}^1x-1.0} & \frac{1}{{}^2x-1.0} \end{pmatrix}$$

Equivalent denominator and numerator read:

$$\sum_{i-\text{th row}} \prod_{j-\text{th col}} [\mathbf{D}]_{i,j} \quad \text{and} \quad \sum_{i-\text{th row}} \mathbf{w} \cdot \prod_{j-\text{th col}} [\mathbf{D}]_{i,j}$$

# Variables decoupling, KST and KANs

KANs via Loewner with rational activation functions ( $\mathbf{H} = {}^1x \cdot {}^2x$ )

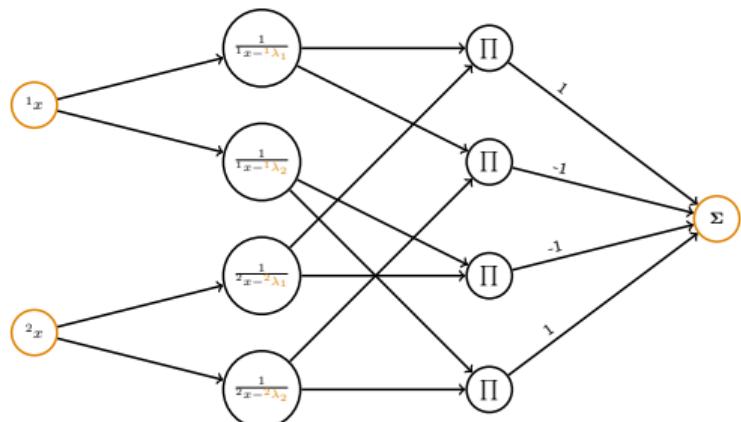
$$\begin{aligned} {}^1\lambda_{j_1} &= \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \\ {}^2\lambda_{j_2} &= \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \end{aligned}$$

$$\left( \begin{array}{cccc} \mathbf{c} & \mathbf{w} & \mathbf{c} \cdot \mathbf{w} & \text{Lag} \\ 1.0 & 1.0 & 1.0 & \frac{1}{({}^1x+1.0)({}^2x+1.0)} \\ -1.0 & -1.0 & 1.0 & \frac{1}{({}^1x+1.0)({}^2x-1.0)} \\ -1.0 & -1.0 & 1.0 & \frac{1}{({}^1x-1.0)({}^2x+1.0)} \\ 1.0 & 1.0 & 1.0 & \frac{1}{({}^1x-1.0)({}^2x-1.0)} \end{array} \right)$$

$$\begin{aligned} \mathbf{c}^{{}^2x} &= \text{vec} \begin{pmatrix} -1.0 & -1.0 \\ 1.0 & 1.0 \end{pmatrix} \\ \mathbf{c}^{{}^1x} &= \begin{pmatrix} -1.0 \\ 1.0 \end{pmatrix} \end{aligned}$$

$$\mathbf{c}_2 = \mathbf{c}^{{}^2x} \odot (\mathbf{c}^{{}^1x} \otimes \mathbf{1}_{k_2})$$

Denominator Network view



# Comparisons

## Some competitors

### Other methods

KAN (P/P 2025)

- ▶ Kolmogorov Arnold Network

Rat. app (B/G 2025)

- ▶ p-AAA

MLP (TensorFlow 2025)

- ▶ Multi Layer Perceptron
- ▶ Keras is the high-level API of TensorFlow (by Google)  
<https://www.tensorflow.org/guide/keras?hl=en>
- ▶ Dense connected network / ReLU activation / ADAM optim. / 1000 iterations / random init.

### TensorFlow interface (Python code)

```
1 import numpy as np
2 import math
3 import matplotlib.pyplot as plt
4 import tensorflow as tf
5 from keras.models import Sequential
6 from matplotlib import cm
7
8 # Data
9 def H(x):
10    y = pow(x[:,0],2)*x[:,1]
11    zy = 1/2*x[:,0] + np.abs(x[:,0])) + 1/10*x[:,1] #1
12    ey = x[:,0]*x[:,1] #2
13    ay = np.exp(x[:,0]*x[:,1])/( (pow(x[:,0],2)-1.44)*(pow(x[:,1],2)-1.44) ) #6
14    dy = np.tanh(4*(x[:,0]-x[:,1])) #8
15    gy = pow(np.abs(x[:,0]-x[:,1]),3) #10
16    y = (pow(x[:,0],2) + pow(x[:,1],2) + x[:,0] - x[:,1] + 1) / (pow(x[:,0],3) + pow(x[:,1],2) + 4) #15
17    return np.transpose(np.array([y]))
18
19 n   = 2
20 N1  = 40
21 N2  = 40
22 x1  = np.linspace(-1, 1, N1)
23 x2  = np.linspace(-1, 1, N2)
24 |
25 # IP
26 N   = N1*N2
27 tab = np.zeros([N,1])
```

<https://arxiv.org/abs/2506.04791>  
<https://github.com/cpoussot/mLF>

---

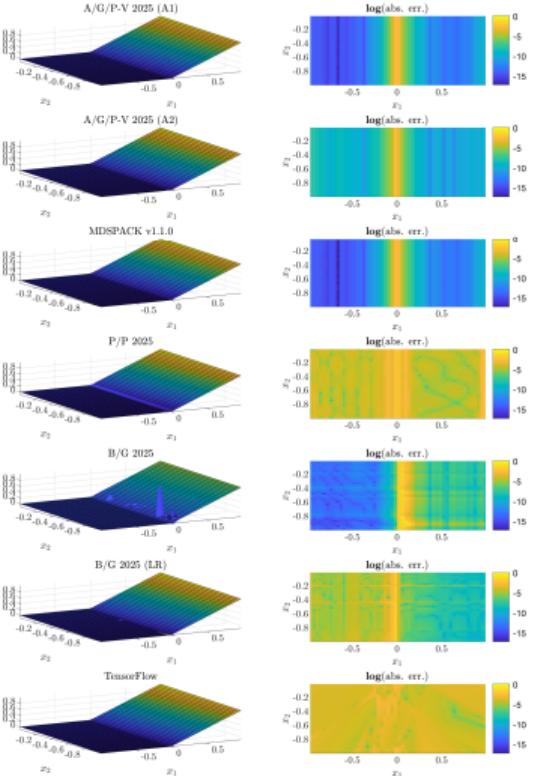
 M. Poluektov and A. Polar, "*Construction of the Kolmogorov-Arnold representation using the Newton-Kaczmarz method*",  
<https://arxiv.org/abs/2305.08194>.

 L. Balicki and S. Gugercin, "*Multivariate Rational Approximation via Low-Rank Tensors and the p-AAA Algorithm*", SISC, 2025.

 M. Abadi et al., "*TensorFlow: Large-scale machine learning on heterogeneous systems, 2015*", Software available from [tensorflow.org](http://tensorflow.org).

# Comparisons

## Irrational functions (example #1)



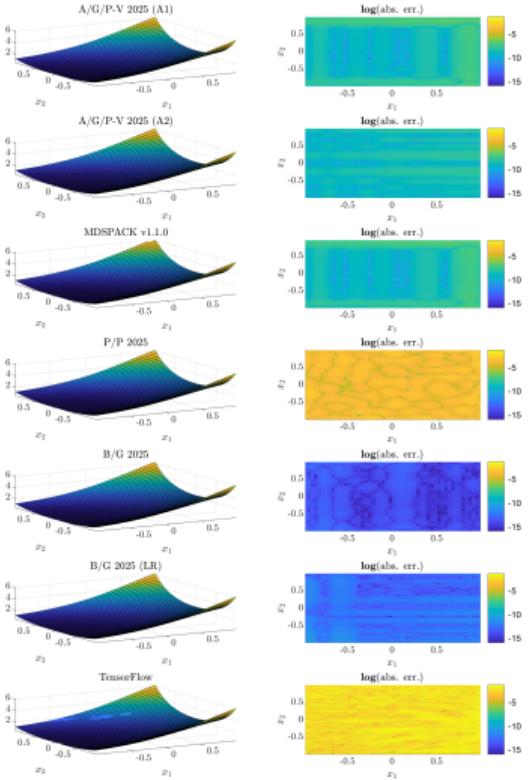
#	Alg.	Parameters	Dim.	CPU [s]	RMSE	min err.
1	A/G/P-V 2025 (A1)	1e-11,3	144	<b>0.0171</b>	0.000699	<b>1.39e - 17</b>
	A/G/P-V 2025 (A2)	1e-15,3	160	0.0823	<b>0.000389</b>	4.49e-13
	MDSPACK v1.1.0	1e-11,1e-06	144	0.0623	0.000699	2.43e-17
	P/P 2025	1,0.95,50,0.01,4,6,9	<b>130</b>	0.282	0.0017	3.08e-07
	B/G 2025	0.001,2,0	612	0.353	0.0288	3.75e-16
	B/G 2025 (LR)	0.001,2,0,4	480	0.678	0.00147	7.82e-12
	TensorFlow		257	14.6	0.00074	6.31e-08

$$\text{ReLU}({}^1x) + \frac{1}{100} {}^2x$$

- ▶ Reference: Personal communication, [none]
- ▶ Domain:  $\mathbb{R}$
- ▶ Tensor size: 12.5 KB (40<sup>2</sup> points)
- ▶ Bounds:  $(-1 \quad 1) \times (-1 \quad -10^{-10})$

# Comparisons

## Irrational functions (example #2)



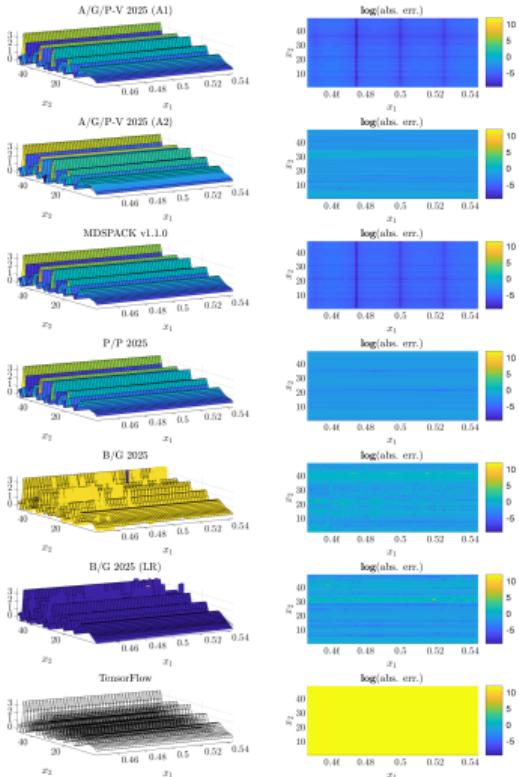
#	Alg.	Parameters	Dim.	CPU [s]	RMSE	min err.	max err.
2	A/G/P-V 2025 (A1)	1e-06,1	<b>168</b>	0.0235	3.36e-08	4.2e-13	1.7e-01
	A/G/P-V 2025 (A2)	1e-15,1	196	0.0888	4.48e-09	2.58e-12	1.62e-01
	MDSPACK v1.1.0	1e-06,0.0001	168	<b>0.0161</b>	3.36e-08	2.09e-12	1.7e-01
	P/P 2025	1,1,50,0.01,6,6,13	238	0.591	0.000287	1.84e-07	0.001
	B/G 2025	1e-09,20	396	0.0665	<b>4.88e - 14</b>	<b>0</b>	<b>4.32e-01</b>
	B/G 2025 (LR)	1e-09,20,3	480	0.827	1.29e-12	2.22e-16	1.38e-01
	TensorFlow		257	14.6	0.00937	9.75e-05	0.036

$$\exp(\sin(x_1) + x_2^2)$$

- ▶ Reference: Personal communication, [none]
- ▶ Domain:  $\mathbb{R}$
- ▶ Tensor size: 12.5 KB ( $40^2$  points)
- ▶ Bounds:  $(-1 \quad 1) \times (-1 \quad 1)$

# Comparisons

## Irrational functions (example #34)



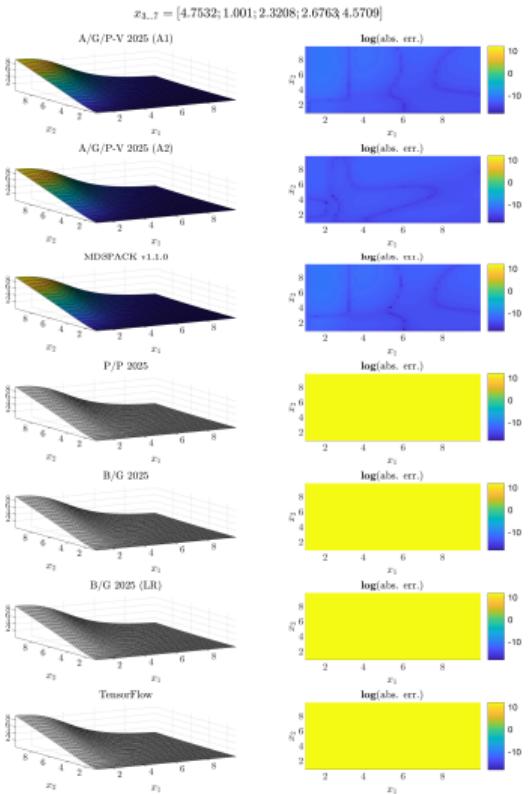
#	Alg.	Parameters	Dim.	CPU [s]	RMSE	min err.
34	A/G/P-V 2025 (A1)	1e-10,2	2.32e+03	1.5	4.86e-05	<b>2.74e-05</b>
	A/G/P-V 2025 (A2)	1e-15,2	864	<b>0.654</b>	0.137	1.51e-06
	MDSPACK v1.1.0	1e-10,1e-09	2.3e+03	1.58	<b>3.21e-05</b>	1.87e-08
	P/P 2025	1,0.95,50,0.01,10,12,21	<b>676</b>	79	0.0254	1.36e-05
	B/G 2025	0.001,20	1.22e+03	85.6	2.53	0.000305
	B/G 2025 (LR)	1e-06,20,3	1.36e+03	16.9	4.74	1.23e-05
	TensorFlow	NaN	NaN	NaN	NaN	NaN

$$\operatorname{Re}(\zeta(1x + i^2x))$$

- ▶ Riemann  $\zeta$  function (real part), [none]
- ▶ Domain:  $\mathbb{R}$
- ▶ Tensor size: 1.22 MB (400<sup>2</sup> points)
- ▶ Bounds:  $\left( \begin{array}{cc} \frac{9}{20} & \frac{11}{20} \end{array} \right) \times \left( \begin{array}{cc} 1 & 50 \end{array} \right)$

# Comparisons

## Irrational functions (example #43)



#	Alg.	Parameters	Dim.	CPU [s]	RMSE	min err.
43	A/G/P-V 2025 (A1)	0.0001,1	<b>1.73e + 04</b>	<b>5.58</b>	1.41e-12	1.76e-16
	A/G/P-V 2025 (A2)	1e-15,1	1.73e+04	12.5	<b>2.39e - 13</b>	<b>1.21e - 17</b>
	MDSPACK v1.1.0	0.0001,1e-06	1.73e+04	5.72	1.4e-12	1.49e-16
	P/P 2025	NaN	NaN	NaN	NaN	NaN
	B/G 2025	NaN	NaN	NaN	NaN	NaN
	B/G 2025 (LR)	NaN	NaN	NaN	NaN	NaN
	TensorFlow	NaN	NaN	NaN	NaN	NaN

$$\frac{3x^2x^3 + 1}{1x^4 + 2x^2 \cdot 3x + 4x^2 + 5x + 6x^3 + 7x}$$

- ▶ Reference: Personal communication, [none]
- ▶ Domain:  $\mathbb{R}$
- ▶ Tensor size:  $\mathbb{R}$
- ▶ Tensor size: **76.3 MB** ( $10^7$  points)
- ▶ Bounds:  $(1 \quad 10)^7$

# Conclusion

## Take home message

### Main contributions

From any  $n$ -th order multi-variate transfer function / data tensor

- ▶ Construct a transfer function in barycentric form
- ▶ Construct a realization with controlled complexity
- ▶ Tame the computational complexity
- ▶ Two algorithms (direct & iterative)
- ▶ Connection with Kolmogorov theorem
- ▶ Connection with Kolmogorov networks

### Side effects

[Sci. con.] Tensor rank approximation

[Sci. con.] Achieve multi-linearization of NEVP

[Sci. con.] Exact (Loewner) matrix null space computation

[Dyn. sys.] Multi-variate / parametric realization

Collaboration with

A.C. Antoulas [Rice Univ.]

I.V. Goșea [MPI]

P. Vuillemin [ONERA]

<https://arxiv.org/abs/2405.00495>

<https://arxiv.org/abs/2506.04791>

<https://github.com/cpoussot/mLF>

<https://cpoussot.github.io>



# In parting... if enough time

Numerical examples, 20-D example

$\mathbf{H}(^1x, ^2x, \dots, ^{20}x) =$

$$\frac{3 \cdot {}^1x^3 + 4 \cdot {}^8x + {}^{12}x + {}^{13}x \cdot {}^{14}x + {}^{15}x}{{}^1x + {}^2x^2 \cdot {}^3x + {}^4x + {}^5x + {}^6x + {}^7x \cdot {}^8x + {}^9x \cdot {}^{10}x \cdot {}^{11}x + {}^{13}x + {}^{13}x^3 \cdot \pi + {}^{17}x + {}^{18}x \cdot {}^{19}x - {}^{20}x}$$

## Statistics

- ▶ 20-D tensor of dimension ( $\geq 48$  TB in real double precision)
- ▶ Complexity: (3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1)
- ▶  $n$ -D Loewner matrix  $6,291,456^2 \rightarrow 288$  TB of storage in real double precision
- ▶ Full SVD:  $2.49 \cdot 10^{20}$  flop  
Recursive SVD:  $5.43 \cdot 10^7$  flop
- ▶ error  $\approx 10^{-11}$

# In parting... if enough time

Numerical examples, 20-D example

$\mathbf{H}(^1x, ^2x, \dots, ^{20}x) =$

$$\frac{3 \cdot {}^1x^3 + 4 \cdot {}^8x + {}^{12}x + {}^{13}x \cdot {}^{14}x + {}^{15}x}{{}^1x + {}^2x^2 \cdot {}^3x + {}^4x + {}^5x + {}^6x + {}^7x \cdot {}^8x + {}^9x \cdot {}^{10}x \cdot {}^{11}x + {}^{13}x + {}^{13}x^3 \cdot \pi + {}^{17}x + {}^{18}x \cdot {}^{19}x - {}^{20}x}$$

## Statistics

- ▶ 20-D tensor of dimension ( $\geq 48$  TB in real double precision)
- ▶ Complexity: (3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
- ▶  $n$ -D Loewner matrix  $6,291,456^2 \rightarrow 288$  TB of storage in real double precision
- ▶ Full SVD:  $2.49 \cdot 10^{20}$  flop  
Recursive SVD:  $5.43 \cdot 10^7$  flop  $\rightarrow 5.03 \cdot 10^7$  flop
- ▶ error  $\approx 10^{-11}$

# In parting... if enough time

## Numerical examples (from 2 to 20 variables)

#4 Rational function

$$s_4^3 + \frac{s_1 s_3}{s_3^2 + s_1 + s_2 + 1}$$

#5 Rational function

$$\frac{s_3^2 + s_1 s_3 s_5^3}{s_1^3 + s_4 + s_2 s_3}$$

#6 Rational function

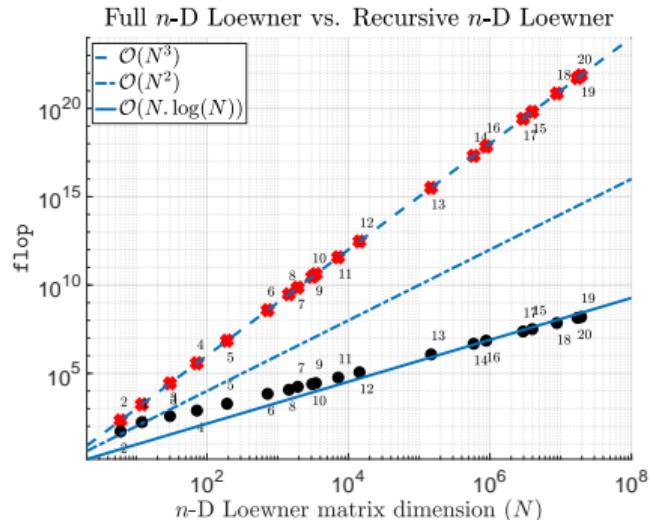
$$\frac{-\sqrt{2} s_6^2 + s_1 + s_3}{s_1^2 + s_4^3 + s_5^2 + s_6 + s_2 s_3}$$

#7 Rational function

$$\frac{s_3 s_2^3 + 1}{s_3 s_2^2 + s_4^2 + s_6^3 + s_1 + s_5 + s_7}$$

#19 Rational function

$$\frac{3 s_1^3 + s_{18}^2 + 4 s_8 + s_{12} + s_{15} + s_{13} s_{14}}{s_3 s_2^2 + \pi s_{16}^3 + s_{17}^2 + s_1 + s_4 + s_5 + s_6 + s_{13} + s_{19} + s_7 s_8 + s_9 s_{10} s_{11}}$$



# In parting... if enough time

## Numerical examples (rational and irrational)

#16 Arc-tangent function

$$\frac{\operatorname{atan}(x_1) + \operatorname{atan}(x_2) + \operatorname{atan}(x_3) + \operatorname{atan}(x_4)}{x_1^2 x_2^2 - x_1^2 - x_2^2 + 1}$$

#17 Exponential function

$$\frac{e^{x_1 x_2 x_3 x_4}}{x_1^2 + x_2^2 - x_3 x_4 + 3}$$

#18 Sinc function

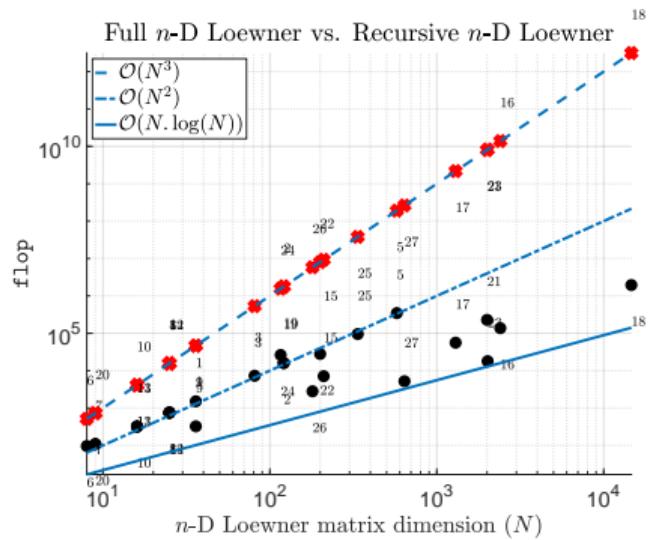
$$\frac{10 \sin(x_1) \sin(x_2) \sin(x_3) \sin(x_4)}{x_1 x_2 x_3 x_4}$$

#19 Sinc function

$$\frac{10 \sin(x_1) \sin(x_2)}{x_1 x_2}$$

#20 Polynomial function

$$x_1^2 + x_1 x_2 + x_2^2 - x_2 + 1$$

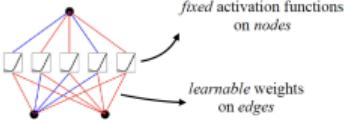
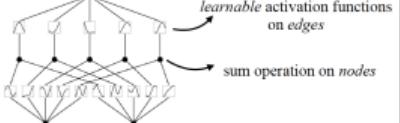
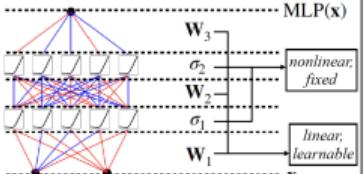
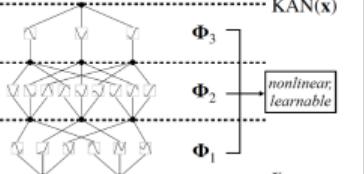


# In parting... if enough time

## About KANs

### KANs features

- ▶ Inspired by the Kolmogorov-Arnold representation theorem
- ▶ The model output is a composition of **sums** and **learnable activation functions** (e.g. splines)
- ▶ Alternate to Multi-Layer Perceptrons (MLP), having fixed activation functions (e.g. ReLU), **inspired by the universal approximation theorem**

Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(c)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^n \phi_{q,p}(x_p) \right)$
Model (Shallow)	(a) 	(b) 
Formula (Deep)	$\text{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\text{KAN}(\mathbf{x}) = (\Phi_3 \circ \Phi_2 \circ \Phi_1)(\mathbf{x})$
Model (Deep)	(c) 	(d) 

Comparison between MLP and KAN (figure from Z. Liu et al.)



# In parting... if enough time

## KANs with splines

Building the Kolmogorov-Arnold model as follows

$$F(^1x, ^2x, \dots, ^nx) = \sum_{k=1}^{2m+1} \Phi_k \left( \sum_{j=1}^m f_{kj}(^jx) \right)$$

$f_{kj} : [0, 1] \mapsto \mathbb{R}$  and  $\Phi_k : \mathbb{R} \mapsto \mathbb{R}$  are continuous functions.

The relation is approximated by  $k = 1, \dots, d = 2m + 1$  as

$$\hat{F}(^1x, ^2x, \dots, ^nx) = \sum_{k=1}^d \Phi_k \underbrace{\left( \sum_{j=1}^m f_{kj}(^jx_i) \right)}_{\theta_{ik}}$$

where  $\theta_{ik}$  denotes the k-th component of  $\theta_i$  vector (interpreted as a hidden variable between two layers), which describes splines

# In parting... if enough time

## KANs with splines

Building the Kolmogorov-Arnold model as follows

$$F(^1x, ^2x, \dots, ^nx) = \sum_{k=1}^{2m+1} \Phi_k \left( \sum_{j=1}^m f_{kj}(^jx) \right)$$

$f_{kj} : [0, 1] \mapsto \mathbb{R}$  and  $\Phi_k : \mathbb{R} \mapsto \mathbb{R}$  are continuous functions.

The relation is approximated by  $k = 1, \dots, d = 2m + 1$  as

$$\hat{F}(^1x, ^2x, \dots, ^nx) = \sum_{k=1}^d \Phi_k \underbrace{\left( \sum_{j=1}^m f_{kj}(^jx_i) \right)}_{\theta_{ik}}$$

where  $\theta_{ik}$  denotes the k-th component of  $\theta_i$  vector (interpreted as a hidden variable between two layers), which describes **splines**