

Tensorization of neural networks into matrix product states

for interpretable physical representations

José R. Pareja Monturiol
Universidad Politécnica de Madrid

Interplay between neural and tensor networks
Grenoble, July 7th 2026



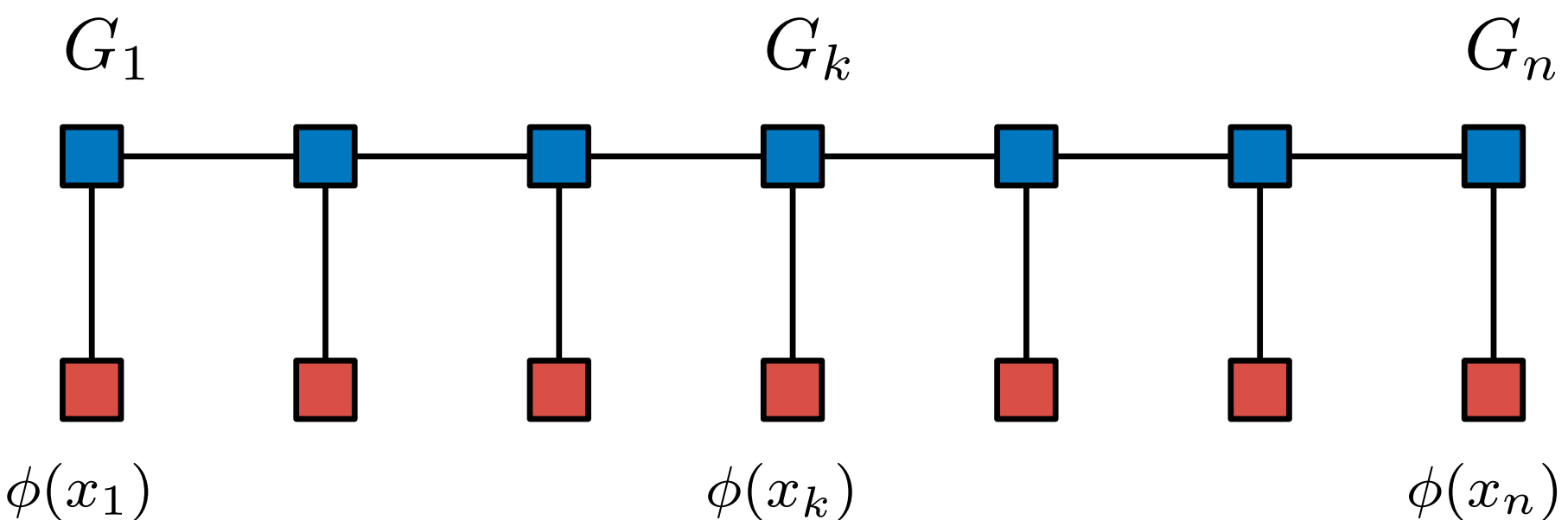
UNIVERSIDAD
POLITÉCNICA
DE MADRID



ESCUELA TÉCNICA
SUPERIOR DE INGENIEROS
INFORMÁTICOS

Motivation

Learning functions in Tensor Train (MPS) form

$$f(x) \approx \hat{f}_{\text{TT}}(x) = \langle \Phi(x), w_{\text{TT}} \rangle =$$


Domain, embedding:

$$x_k \in \{0, \dots, b-1\}, \quad \phi(x_k) = \langle x_k |$$

$$x_k \in [a, b], \quad \phi(x_k) = \left[\cos\left(\frac{\pi}{2}x_k\right), \sin\left(\frac{\pi}{2}x_k\right) \right]$$

$$\phi(x_k) = [1, x_k, x_k^2]$$

$$\dots$$

Optimization problem:

$$|\hat{f}_{\text{TT}}\rangle = |\psi\rangle, \quad \min_{G_k} \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$$

$$\min_{G_k} d(f, \hat{f}_{\text{TT}})$$

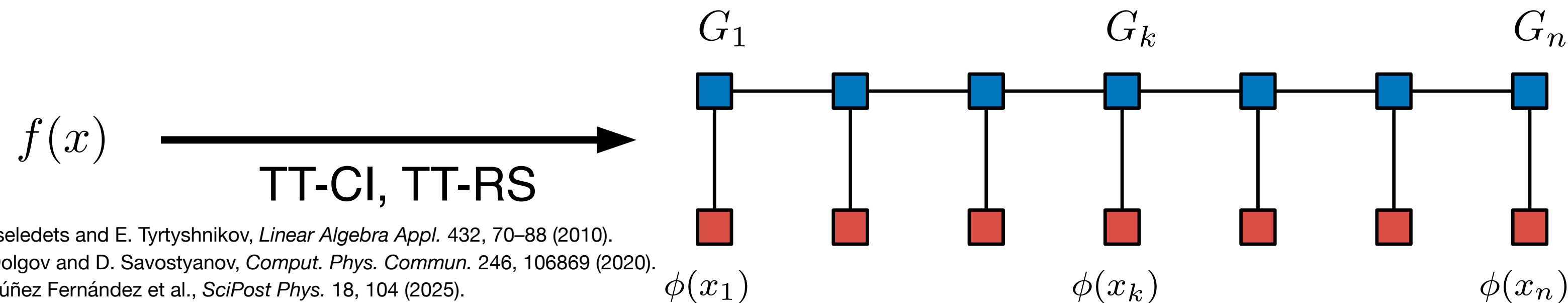
$$\min_{G_k} \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}(\hat{f}_{\text{TT}}(x^i), y^i)$$

Issues:

- Initialization
- Stability
- Expressivity (D, ϕ)
- Network topology, hyperparam. opt., optimizers, ...

Motivation

Learning functions in Tensor Train (MPS) form



I. Oseledets and E. Tyrtyshnikov, *Linear Algebra Appl.* 432, 70–88 (2010).
 S. Dolgov and D. Savostyanov, *Comput. Phys. Commun.* 246, 106869 (2020).
 Y. Núñez Fernández et al., *SciPost Phys.* 18, 104 (2025).
 Y. Hur et al., *Appl. Comput. Harmon. Anal.* 67, 101575 (2023).

Domain, embedding:

$$x_k \in \{0, \dots, b-1\}, \quad \phi(x_k) = \langle x_k |$$

$$x_k \in [a, b], \quad \phi(x_k) = \left[\cos\left(\frac{\pi}{2}x_k\right), \sin\left(\frac{\pi}{2}x_k\right) \right]$$

$$\phi(x_k) = [1, x_k, x_k^2]$$

...

Optimization problem:

$$|\hat{f}_{\text{TT}}\rangle = |\psi\rangle, \quad \min_{G_k} \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$$

$$\min_{G_k} d(f, \hat{f}_{\text{TT}})$$

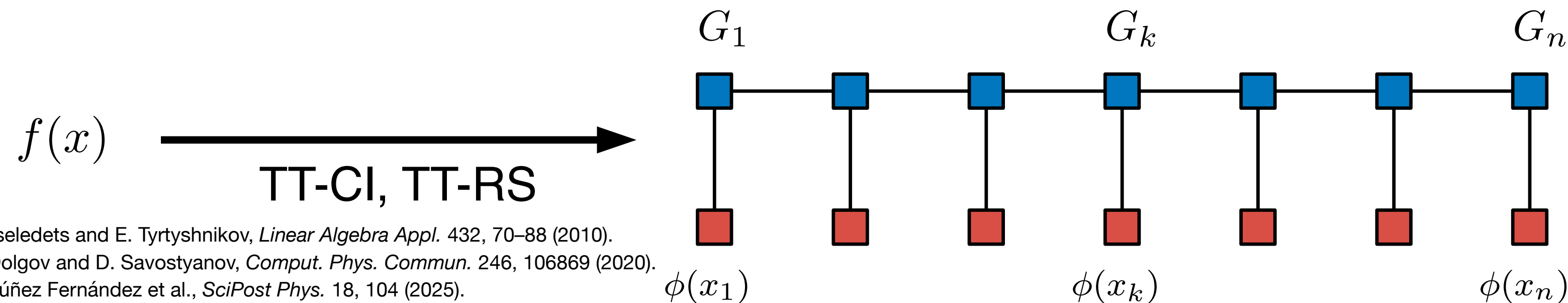
$$\min_{G_k} \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}(\hat{f}_{\text{TT}}(x^i), y^i)$$

Issues:

- Initialization
- Stability
- Expressivity (D, ϕ)
- Network topology, hyperparam. opt., optimizers, ...

Motivation

Learning functions in Tensor Train (MPS) form



I. Oseledets and E. Tyrtyshnikov, *Linear Algebra Appl.* 432, 70–88 (2010).
 S. Dolgov and D. Savostyanov, *Comput. Phys. Commun.* 246, 106869 (2020).
 Y. Núñez Fernández et al., *SciPost Phys.* 18, 104 (2025).
 Y. Hur et al., *Appl. Comput. Harmon. Anal.* 67, 101575 (2023).

Domain, embedding:

$$x_k \in \{0, \dots, b-1\}, \quad \phi(x_k) = \langle x_k |$$

$$x_k \in [a, b], \quad \phi(x_k) = \left[\cos\left(\frac{\pi}{2}x_k\right), \sin\left(\frac{\pi}{2}x_k\right) \right]$$

$$\phi(x_k) = [1, x_k, x_k^2]$$

...

Optimization problem:

$$|\hat{f}_{\text{TT}}\rangle = |\psi\rangle, \quad \min_{G_k} \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$$

$$\min_{G_k} d(f, \hat{f}_{\text{TT}})$$

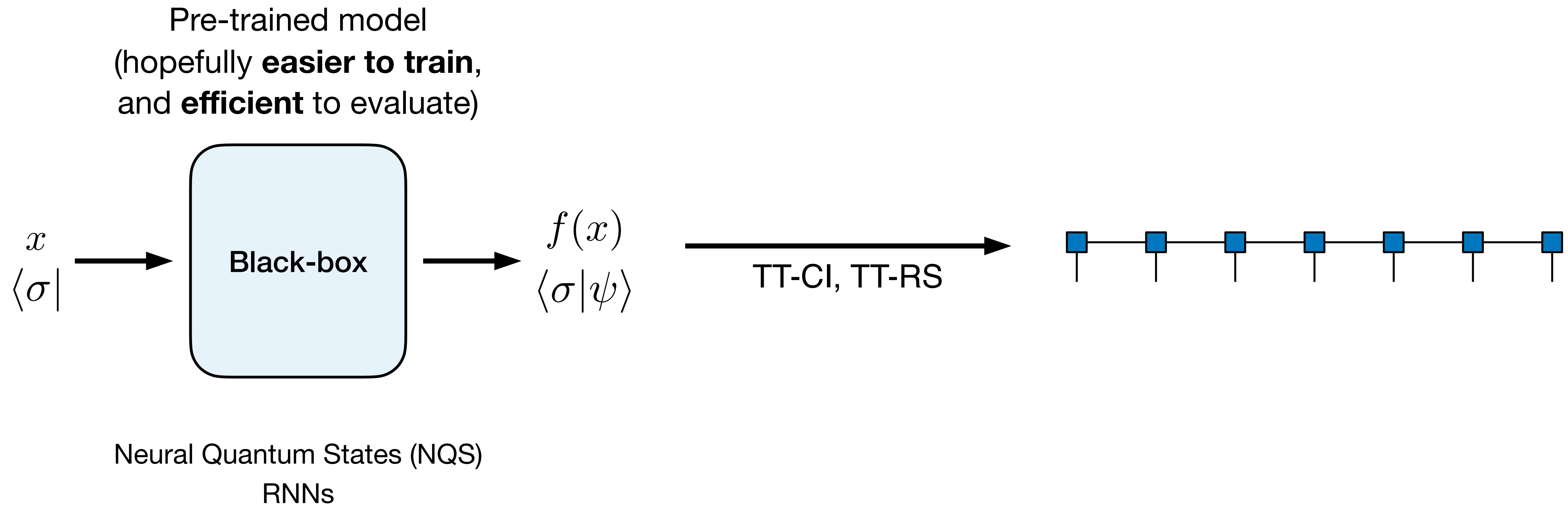
$$\min_{G_k} \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}(\hat{f}_{\text{TT}}(x^i), y^i)$$

Issues:

- Initialization
- Stability
- Expressivity (D, ϕ)
- Network topology, hyperparam. opt., optimizers, ...

Motivation

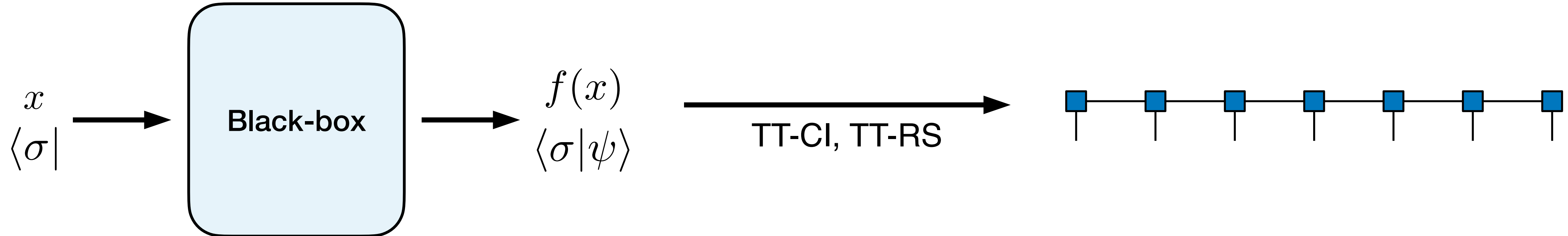
TTs are more **interpretable**:



G. Carleo and M. Troyer, *Science* 355, 602–606 (2017).
M. Hibat-Allah et al., *Phys. Rev. Res.* 2, 023358 (2020).

Motivation

Pre-trained model
(hopefully **easier to train**,
and **efficient** to evaluate)



Neural Quantum States (NQS)
RNNs

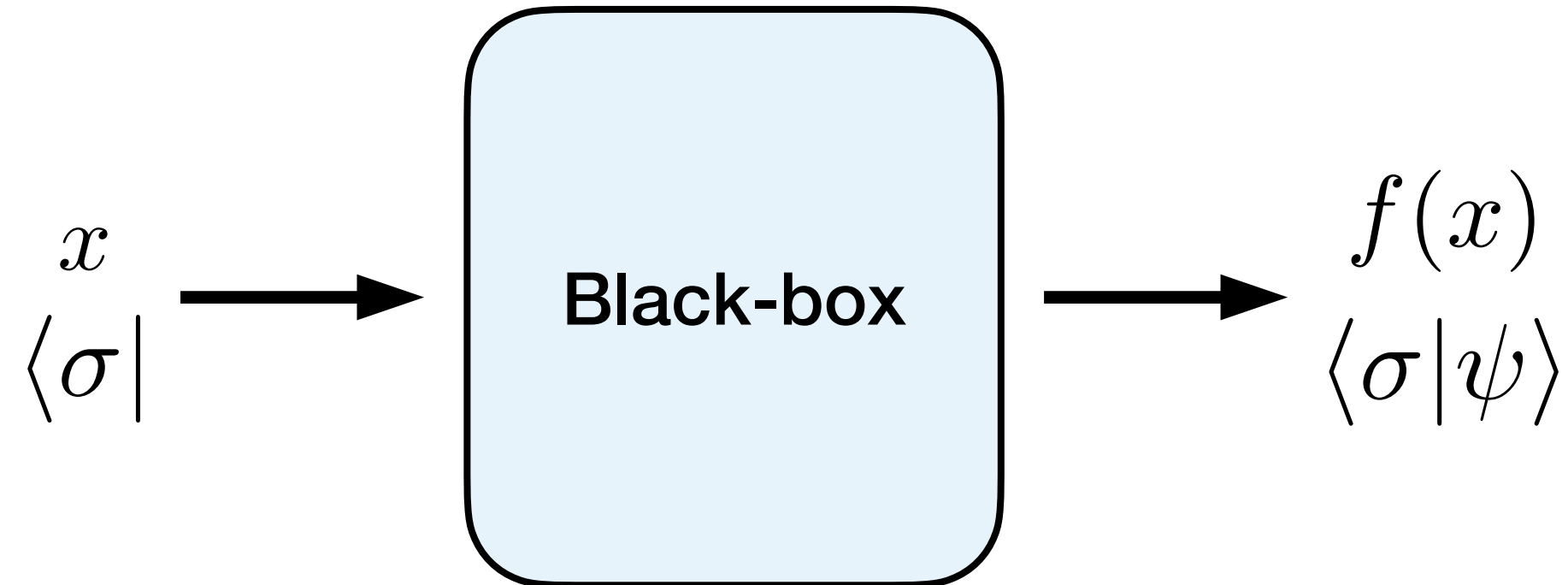
TTs are more **interpretable**:

- Efficient access to marginal and conditional distributions, entanglement, etc.

G. Carleo and M. Troyer, *Science* 355, 602–606 (2017).
M. Hibat-Allah et al., *Phys. Rev. Res.* 2, 023358 (2020).

Motivation

Pre-trained model
(hopefully **easier to train**,
and **efficient** to evaluate)

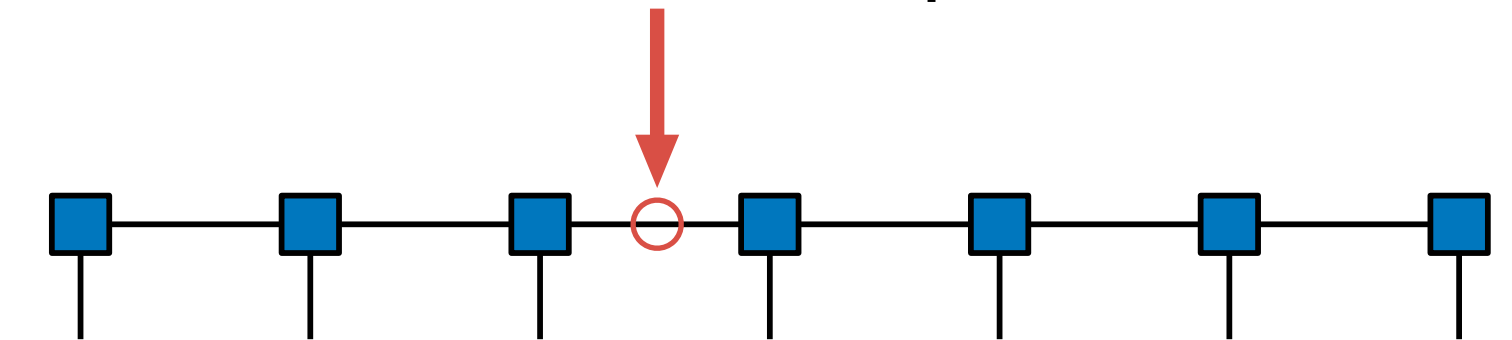


Neural Quantum States (NQS)
RNNs

TT-CI, TT-RS

TTs are more **interpretable**:

- Efficient access to marginal and conditional distributions, entanglement, etc.
- Access to virtual representations



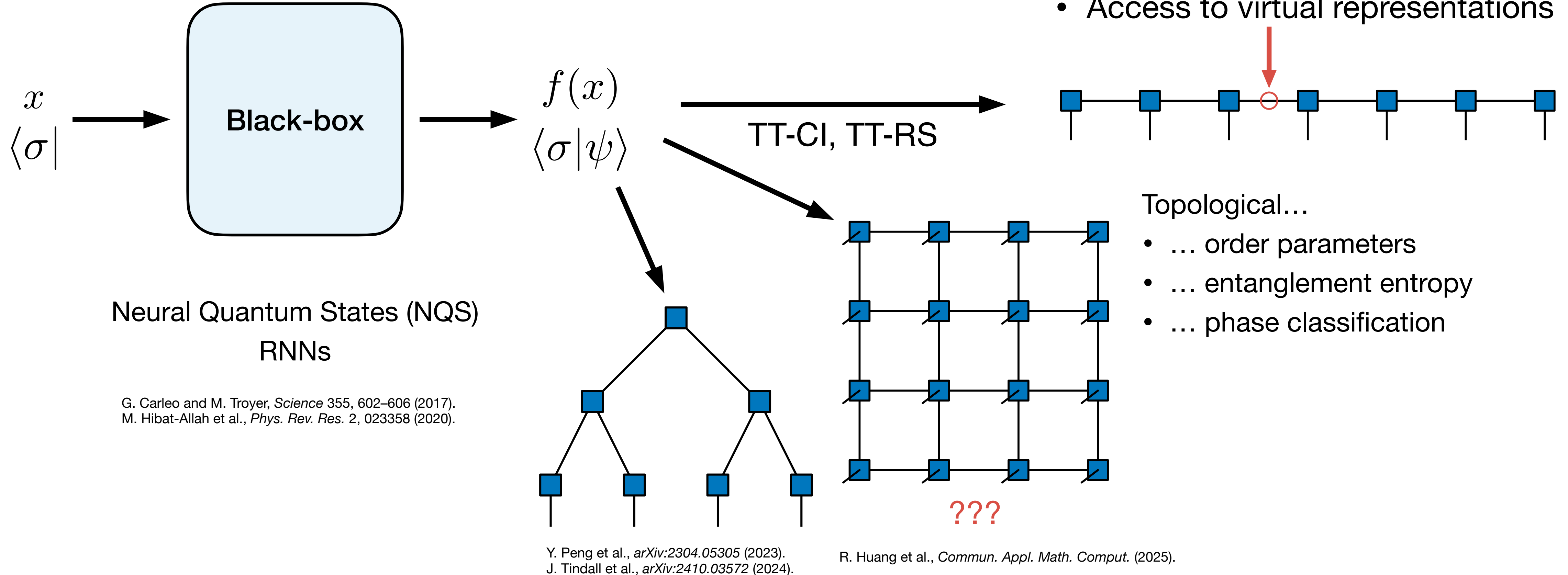
Topological...

- ... order parameters
- ... entanglement entropy
- ... phase classification

G. Carleo and M. Troyer, *Science* 355, 602–606 (2017).
M. Hibat-Allah et al., *Phys. Rev. Res.* 2, 023358 (2020).

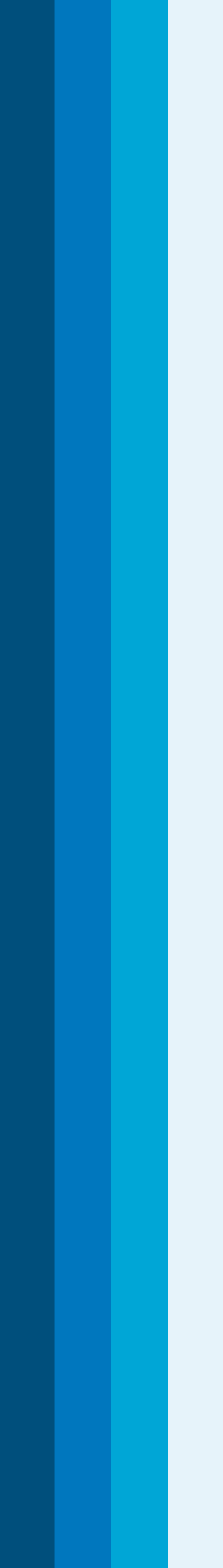
Motivation

Pre-trained model
(hopefully **easier to train**,
and **efficient** to evaluate)



Outline

- 1 How to tensorize ML models into TT form?
- 2 Applications
- 3 Next steps: Tensor Rings and PEPS



Tensorization

Tensorization

General form of **matrix** decomposition

$$A \in \mathbb{C}^{m \times n}, \quad \text{rank}(A) = r$$

$$B \in \mathbb{C}^{m \times m'}, \quad \text{rank}(AB) = r, \quad C \in \mathbb{C}^{n' \times n}, \quad \text{rank}(CA) = r$$

$$m', n' \geq r$$

$$A = (AB)(CAB)^+(CA)$$

Tensorization

General form of **matrix** decomposition

$$A \in \mathbb{C}^{m \times n}, \quad \text{rank}(A) = r$$

$$B \in \mathbb{C}^{m \times m'}, \quad \text{rank}(AB) = r, \quad C \in \mathbb{C}^{n' \times n}, \quad \text{rank}(CA) = r$$
$$m', n' \geq r$$

$$A = (AB)(CAB)^+(CA)$$

$$A = U_r V_r$$

$$\begin{aligned} A &= (U_r V_r B)(C U_r V_r B)^+(C U_r V_r) \\ &= U_r (V_r B)(V_r B)^+(C U_r)^+(C U_r) V_r \\ &= U_r V_r = A \end{aligned}$$

Tensorization

General form of **matrix** decomposition

$$\begin{aligned} A &\in \mathbb{C}^{m \times n}, \quad \text{rank}(A) = r \\ B &\in \mathbb{C}^{m \times m'}, \quad \text{rank}(AB) = r, \quad C \in \mathbb{C}^{n' \times n}, \quad \text{rank}(CA) = r \\ m', n' &\geq r \end{aligned}$$

$$A = (AB)(CAB)^+(CA)$$

Singular Value Decomposition

$$\begin{aligned} A &= USV^\dagger \\ B &= V, \quad C = U^\dagger \\ A &= (US)S^+(SV^\dagger) \end{aligned}$$

Randomized SVD

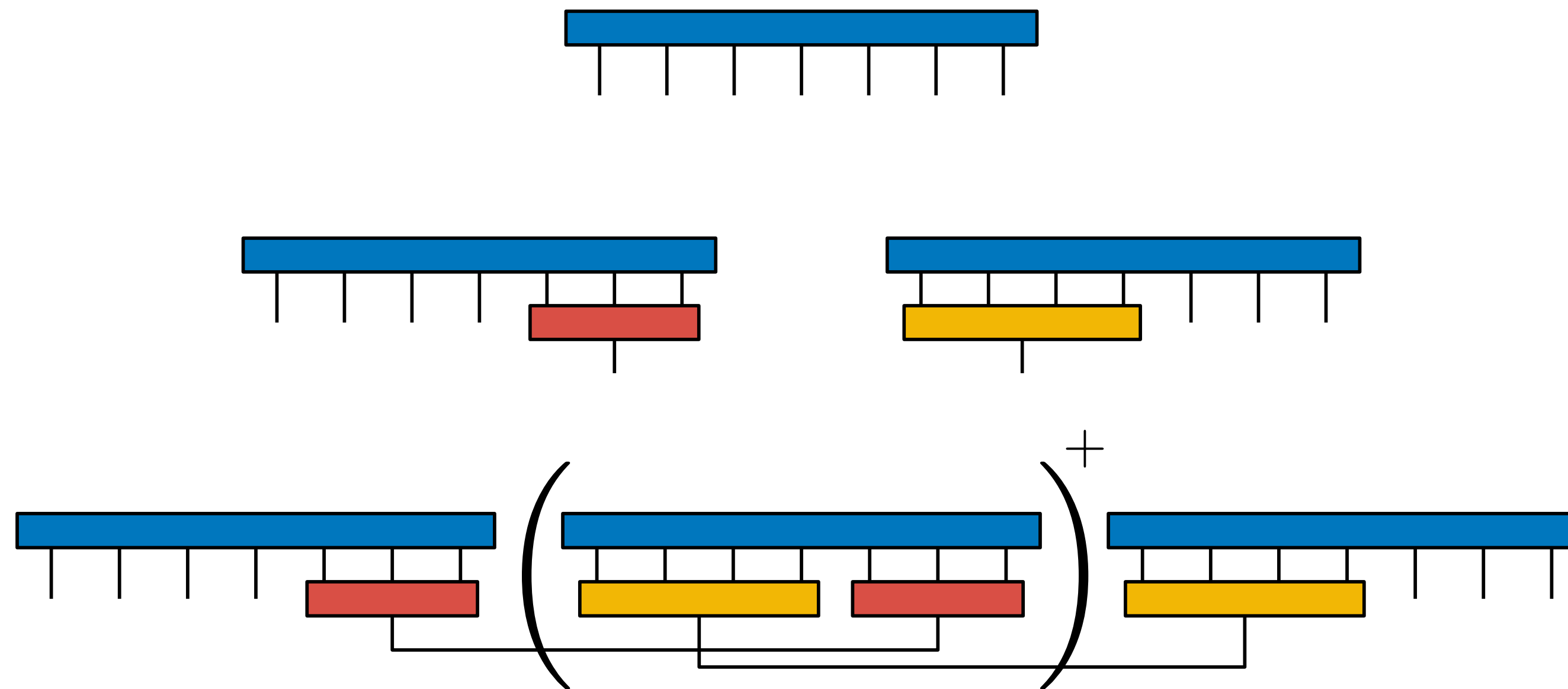
$$\begin{aligned} Y &= A\Omega = U_Y S_Y V_Y^\dagger \\ B &= \Omega V_Y, \quad C = \Phi \\ A &= (U_Y S_Y)(\Phi A V_Y)^+(\Phi A) \end{aligned}$$

Cross Interpolation

$$\begin{aligned} AB &= A(\mathbb{I}, \mathcal{J}), \quad CA = A(\mathcal{I}, \mathbb{J}) \\ A &= A(\mathbb{I}, \mathcal{J})A(\mathcal{I}, \mathcal{J})^{-1}A(\mathcal{I}, \mathbb{J}) \end{aligned}$$

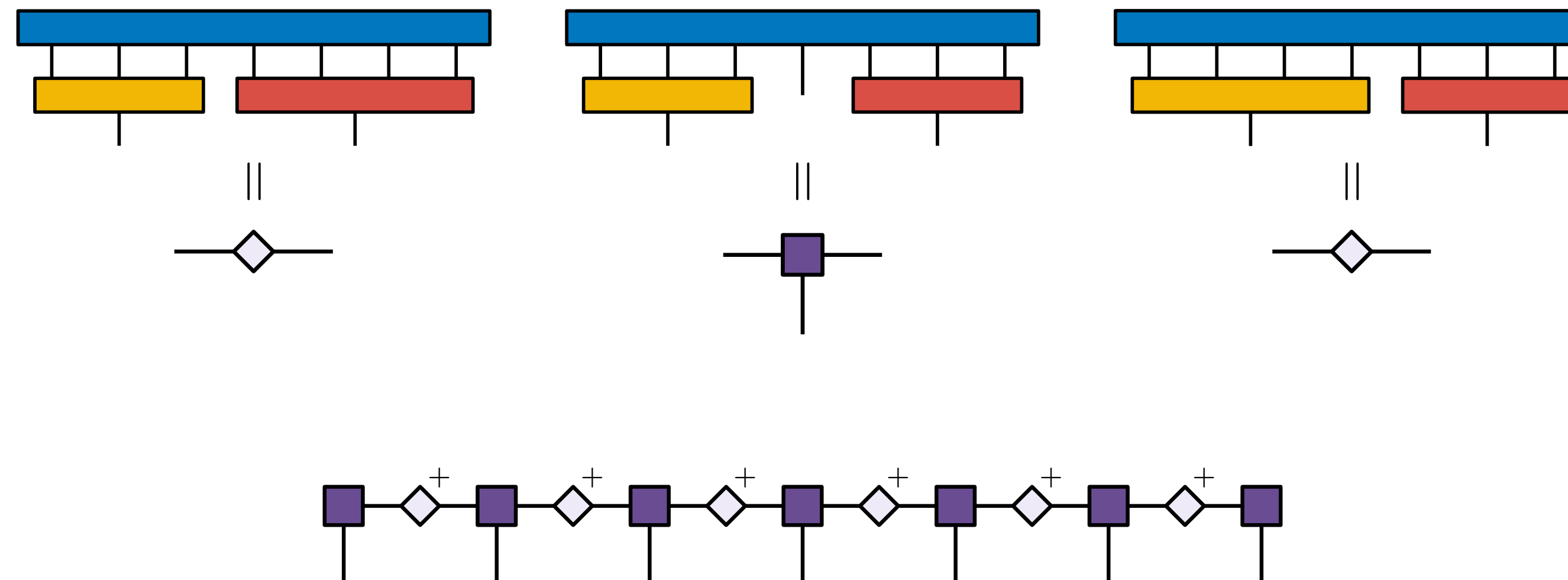
Tensorization

General form of **tensor** decomposition



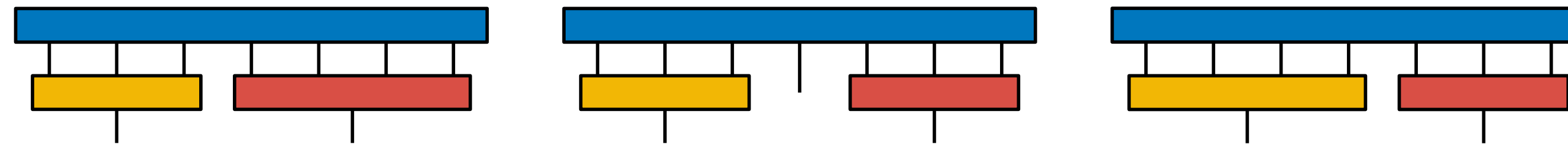
Tensorization

General form of **Tensor Train** decomposition

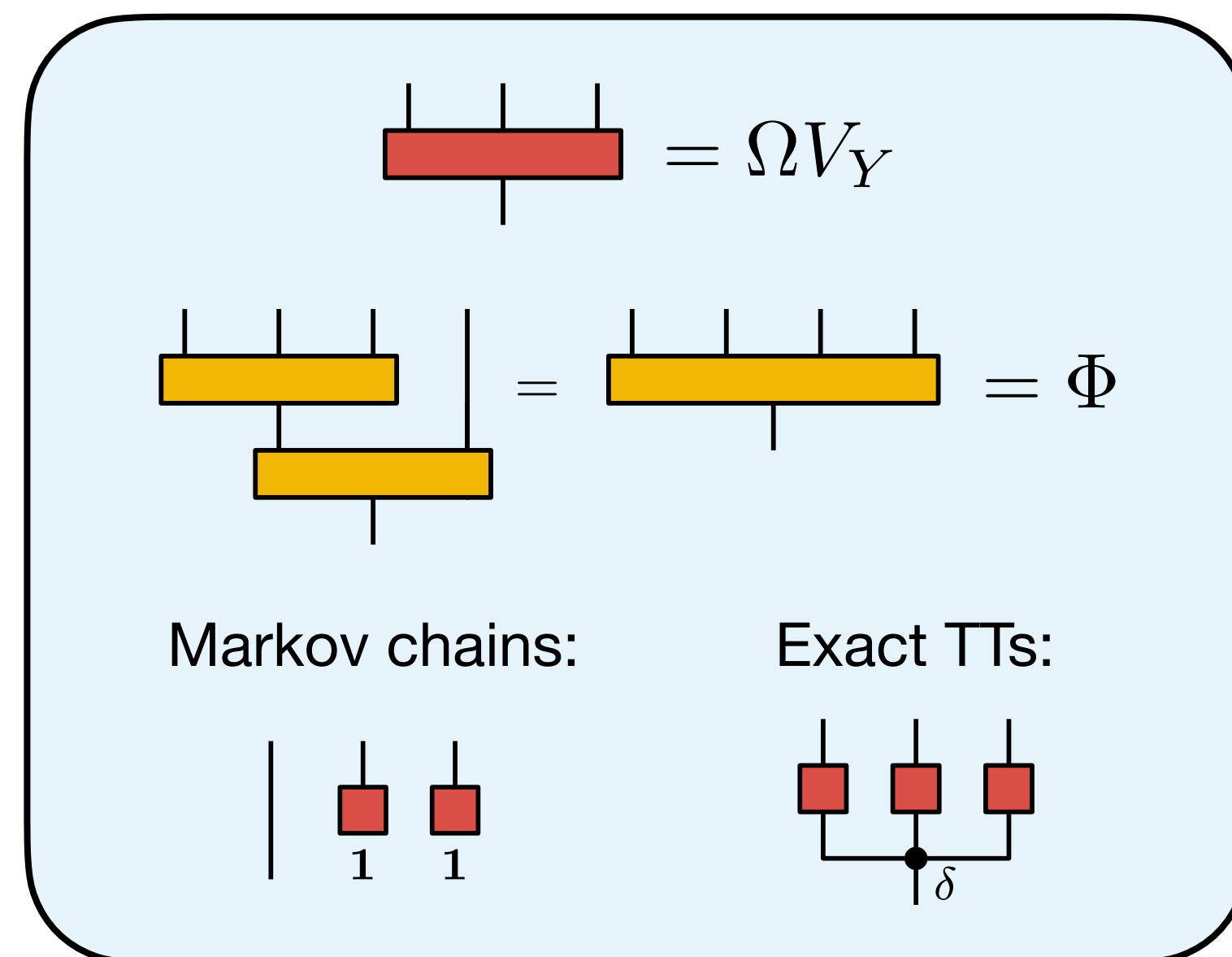


Tensorization

General form of **Tensor Train** decomposition



Tensor Train via Recursive Sketching



Y. Hur et al., *Appl. Comput. Harmon. Anal.* 67, 101575 (2023).

Tensor Train via Cross Interpolation

$$\mathcal{I}_k = \{(i_1^\alpha, \dots, i_k^\alpha)\}_{\alpha=1, \dots, r} \quad \mathcal{I}_{k+1} \subseteq \mathcal{I}_k \times [d]$$

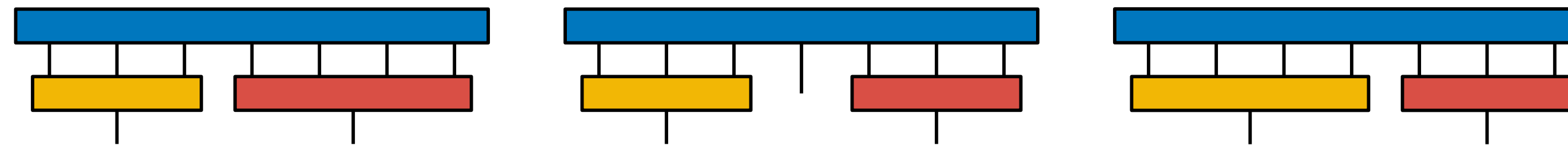
$$\mathcal{J}_k = \{(i_k^\beta, \dots, i_n^\beta)\}_{\beta=1, \dots, r} \quad \mathcal{J}_{k-1} \subseteq [d] \times \mathcal{J}_k$$

$$\left. \begin{aligned} \text{Red bar} &= \langle i_k^\beta | \text{Red bar} | i_n^\beta \rangle \\ \text{Yellow bar} &= \langle i_1^\alpha | \text{Yellow bar} | i_k^\alpha \rangle \end{aligned} \right\} \text{Red bar} = f(\mathcal{I}_{k-1}, i_k, \mathcal{J}_k)$$

I. Oseledets and E. Tyrtyshnikov, *Linear Algebra Appl.* 432, 70–88 (2010).
 S. Dolgov and D. Savostyanov, *Comput. Phys. Commun.* 246, 106869 (2020).
 Y. Núñez Fernández et al., *SciPost Phys.* 18, 104 (2025).

Tensorization

General form of **Tensor Train** decomposition



TT-RSS:

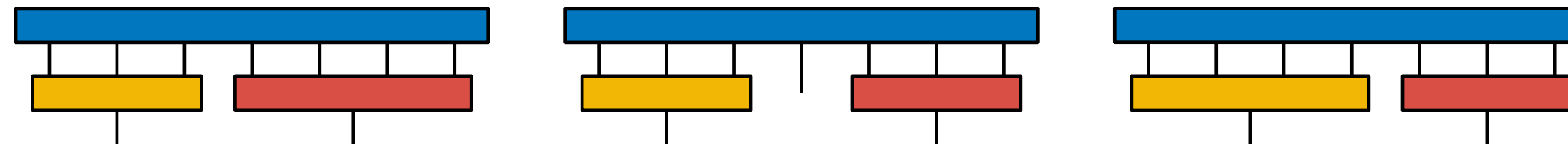
- Efficient for empirical distributions, Markov chains, TTs
- **Generally infeasible**
- A single left-to-right sweep
- Can be adapted to continuous functions

TT-CI:

- Efficient for a variety of scenarios
- Heuristic optimization (maxvol principle)
- Iterative sweeps
- **Ergodicity issues**

Tensorization

General form of **Tensor Train** decomposition



TT-RSS:

- Efficient for empirical distributions, Markov chains, TTs
- **Generally infeasible**
- A single left-to-right sweep
- Can be adapted to continuous functions

TT-CI:

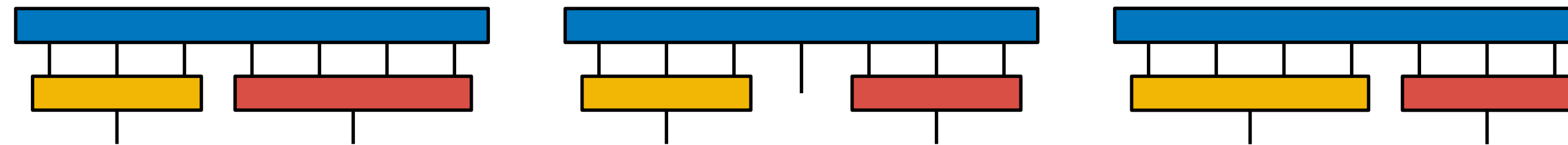
- Efficient for a variety of scenarios
- Heuristic optimization (maxvol principle)
- Iterative sweeps
- **Ergodicity issues**

Common Deep Learning setting:

- Very **high-dimensional, highly sparse** (and non Markovian) functions

Tensorization

General form of **Tensor Train** decomposition



TT-RSS:

- Efficient for empirical distributions, Markov chains, TTs
- **Generally infeasible**
- A single left-to-right sweep
- Can be adapted to continuous functions

TT-CI:

- Efficient for a variety of scenarios
- Heuristic optimization (maxvol principle)
- Iterative sweeps
- **Ergodicity issues**

What projectors to choose?

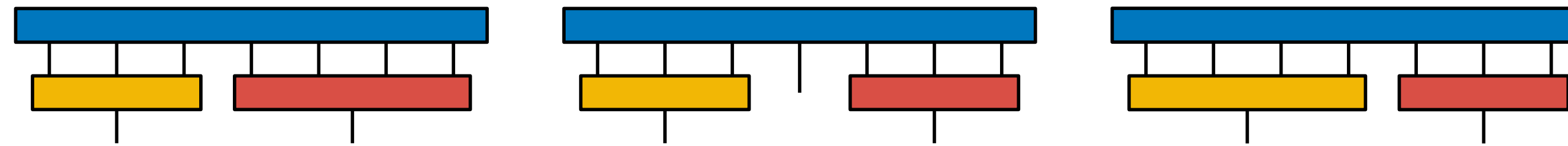
- Generally efficient
- No strong assumptions on the function/model structure
- No strong assumptions on the domain

Common Deep Learning setting:

- Very **high-dimensional, highly sparse** (and non Markovian) functions

Tensorization

General form of **Tensor Train** decomposition



What we do have is *samples*

In Deep Learning:

- We have access to a **training dataset**

$$\mathcal{D} = \{(x_1^j, \dots, x_n^j)\}_{j=1}^N$$

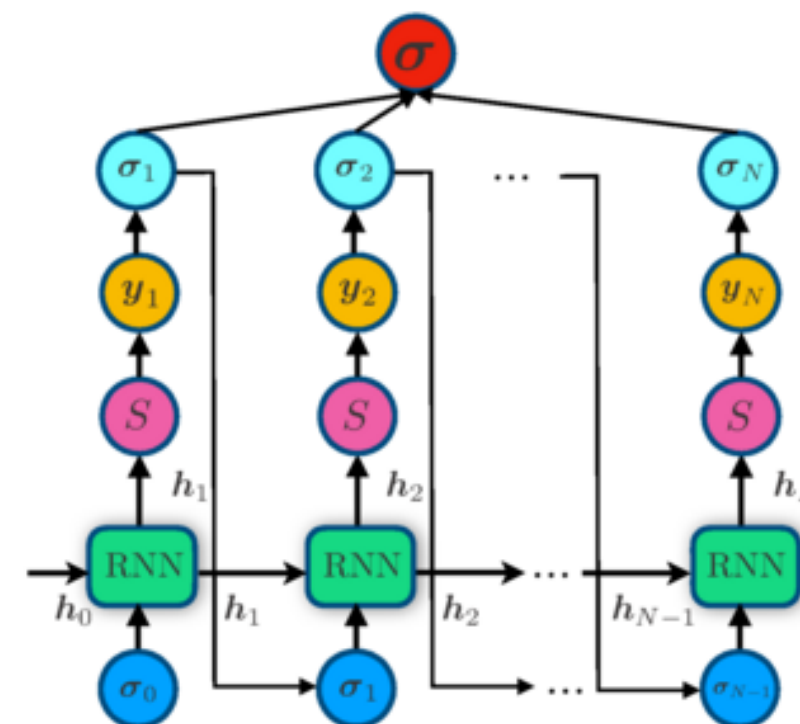
$$f_{\text{NN}}(x) = p_{\text{NN}}(x)$$

$$f_{\text{NN}}(x, y) = p_{\text{NN}}(y \mid x)$$

In Neural Quantum States (RNNs):

- We can efficiently **generate samples**

$$\mathcal{D} = \{(\sigma_1^j, \dots, \sigma_n^j)\}_{j=1}^N$$

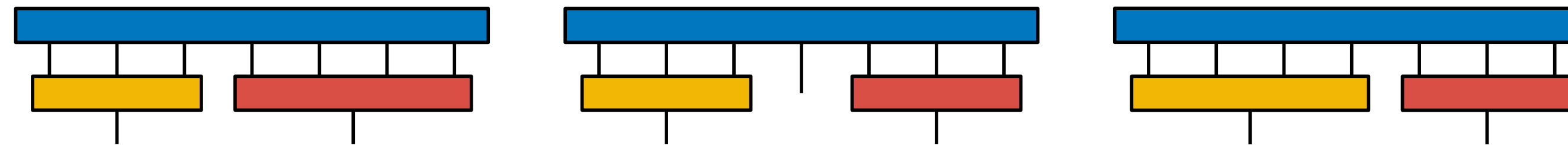


Generally:

- No guarantees
- For sufficiently delocalized / broadly supported functions, draw **uniform random samples**

Tensorization

General form of **Tensor Train** decomposition



Tensor Train via Recursive Sketching *from Samples*

$$\mathcal{I}_k = \{(x_1^\alpha, \dots, x_k^\alpha)\}_{\alpha=1, \dots, N}$$

$$\mathcal{J}_k = \{(x_k^\beta, \dots, x_n^\beta)\}_{\beta=1, \dots, N}$$

$$\text{Red bar} = \langle x_k^\beta | \text{Red bar} | x_n^\beta \rangle$$

Diagram showing a red bar with three vertical lines. The left line is labeled β , the middle line is labeled β , and the right line is labeled β . A red trapezoid labeled V_r is connected to the middle line, with a label β' below it and a label δ to its right.

$$\text{Yellow bar} = \langle x_1^\alpha | \text{Yellow bar} | x_k^\alpha \rangle$$

Diagram showing a yellow bar with three vertical lines. The left line is labeled α , the middle line is labeled α , and the right line is labeled α . A yellow trapezoid is connected to the middle line, with a label α below it and a label δ to its right.

By definition:

$$\mathcal{I}_{k+1} \subseteq \mathcal{I}_k \times [d]$$

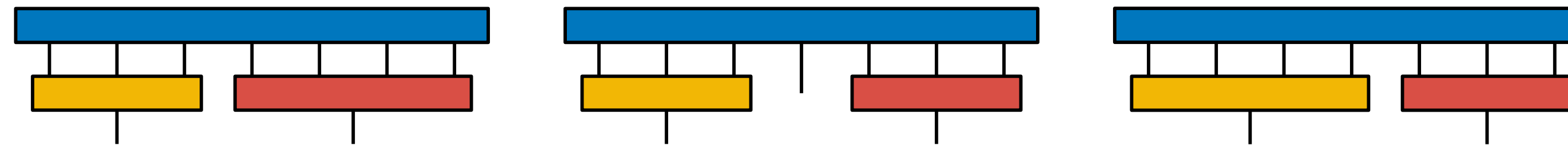
$$\mathcal{J}_{k-1} \subseteq [d] \times \mathcal{J}_k$$

$$\text{Yellow bar} = \text{Yellow bar}$$

Diagram showing a yellow bar with three vertical lines. The left line is labeled δ , the middle line is labeled δ , and the right line is labeled δ . A yellow trapezoid is connected to the middle line, with a label δ below it and a label δ to its right.

Tensorization

General form of **Tensor Train** decomposition



Tensor Train via Recursive Sketching *from Samples*

$$\mathcal{I}_k = \{(x_1^\alpha, \dots, x_k^\alpha)\}_{\alpha=1, \dots, N}$$

$$\mathcal{J}_k = \{(x_k^\beta, \dots, x_n^\beta)\}_{\beta=1, \dots, N}$$

$$\text{Red bar} = \langle x_k^\beta | \text{Red bar} | x_n^\beta \rangle$$

Diagram showing a red bar with three vertical lines, connected to a red trapezoid labeled V_r with a δ label.

$$\text{Yellow bar} = \langle x_1^\alpha | \text{Yellow bar} | x_k^\alpha \rangle$$

Diagram showing a yellow bar with three vertical lines, connected to a yellow trapezoid labeled V_r with a δ label.

By definition:

$$\mathcal{I}_{k+1} \subseteq \mathcal{I}_k \times [d]$$

$$\mathcal{J}_{k-1} \subseteq [d] \times \mathcal{J}_k$$

$$\text{Yellow bar} = \text{Yellow bar}$$

Diagram showing a yellow bar with three vertical lines, connected to a yellow trapezoid labeled V_r with a δ label.

Sample complexity (N)

Gaussian projection (rSVD):

N. Halko, P.-G. Martinsson, and J. A. Tropp, *SIAM Rev.* 53, 217–288 (2011).

$$O(r + \log r)$$

Leverage score sampling:

P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *SIAM J. Matrix Anal. Appl.* 30, 844–881 (2008).

$$O(r \log r)$$

$$O(\mu r \log r)$$

Norm-squared sampling:

M. Rudelson and R. Vershynin, *J. ACM* 54, 21 (2007).

$$O(r' \log r')$$

r' : numerical rank

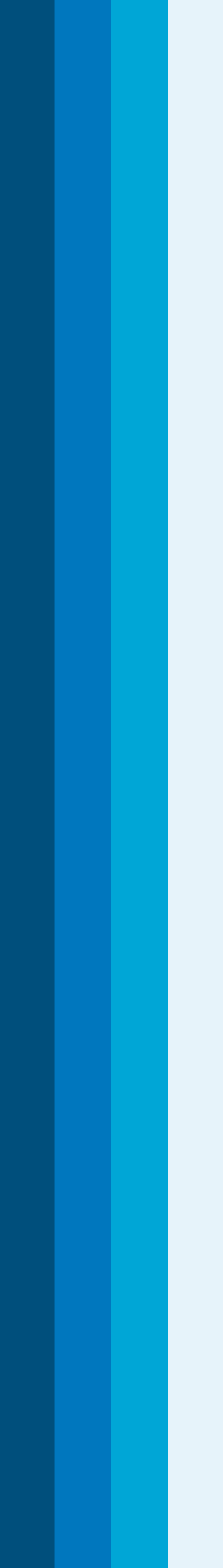
Tensorization

Performance

$r = 10$

f	n	Tensorization algorithm	Error on test	Time in secs.
Random TT	100	TT-RSS ($N = 20$)	3.60×10^{-1}	1.97×10^0
		TT-RSS ($N = 35$)	4.26×10^{-15}	4.25×10^0
		TT-CI	7.56×10^{-15}	3.83×10^1 (4.73×10^2)
	200	TT-RSS ($N = 20$)	9.23×10^{-15}	7.08×10^0
		TT-RSS ($N = 35$)	6.56×10^{-15}	1.89×10^1
		TT-CI	1.08×10^{-14}	1.37×10^2 (2.53×10^3)
Slater functions $f(z) = \frac{e^{-\ z\ _2}}{\ z\ _2}$	100	TT-RSS ($N = 50$)	8.06×10^{-1}	7.70×10^0
		TT-RSS ($N = 100$)	1.58×10^{-1}	2.76×10^1
		TT-CI	6.27×10^{-1}	4.09×10^1 (4.71×10^2)
	200	TT-RSS ($N = 50$)	3.69×10^{-1}	1.76×10^1
		TT-RSS ($N = 100$)	2.12×10^{-1}	5.57×10^1
		TT-CI	5.63×10^{-1}	1.34×10^2 (2.53×10^3)
	500	TT-RSS ($N = 50$)	3.61×10^{-1}	5.22×10^1
		TT-RSS ($N = 100$)	2.29×10^{-1}	1.94×10^2
		TT-CI	6.79×10^{-1}	3.30×10^2 (1.32×10^4)

f	n	Tensorization algorithm	Error on test	Accuracy	Time in secs.
Bars and Stripes	144	TT-RSS ($N = 50$)	5.97×10^{-1}	99.1%	7.03×10^0
		TT-RSS ($N = 100$)	3.95×10^{-1}	99.5%	2.29×10^1
		TT-CI	4.07×10^{-1}	97.9%	4.23×10^1 (5.65×10^2)
	256	TT-RSS ($N = 50$)	4.12×10^{-1}	98.8%	1.49×10^1
		TT-RSS ($N = 100$)	4.52×10^{-1}	95%	4.96×10^1
		TT-CI	4.39×10^{-1}	98.5%	8.62×10^1 (1.70×10^3)
MNIST	144	TT-RSS ($N = 50$)	1.17×10^1	68.8%	8.26×10^0
		TT-RSS ($N = 100$)	5.51×10^0	79.3%	2.44×10^1
		TT-CI	1.20×10^0	15.9%	2.12×10^2 (7.37×10^2)

A decorative vertical bar on the left side of the slide, composed of three parallel stripes in dark blue, medium blue, and light blue.

Applications

Applications

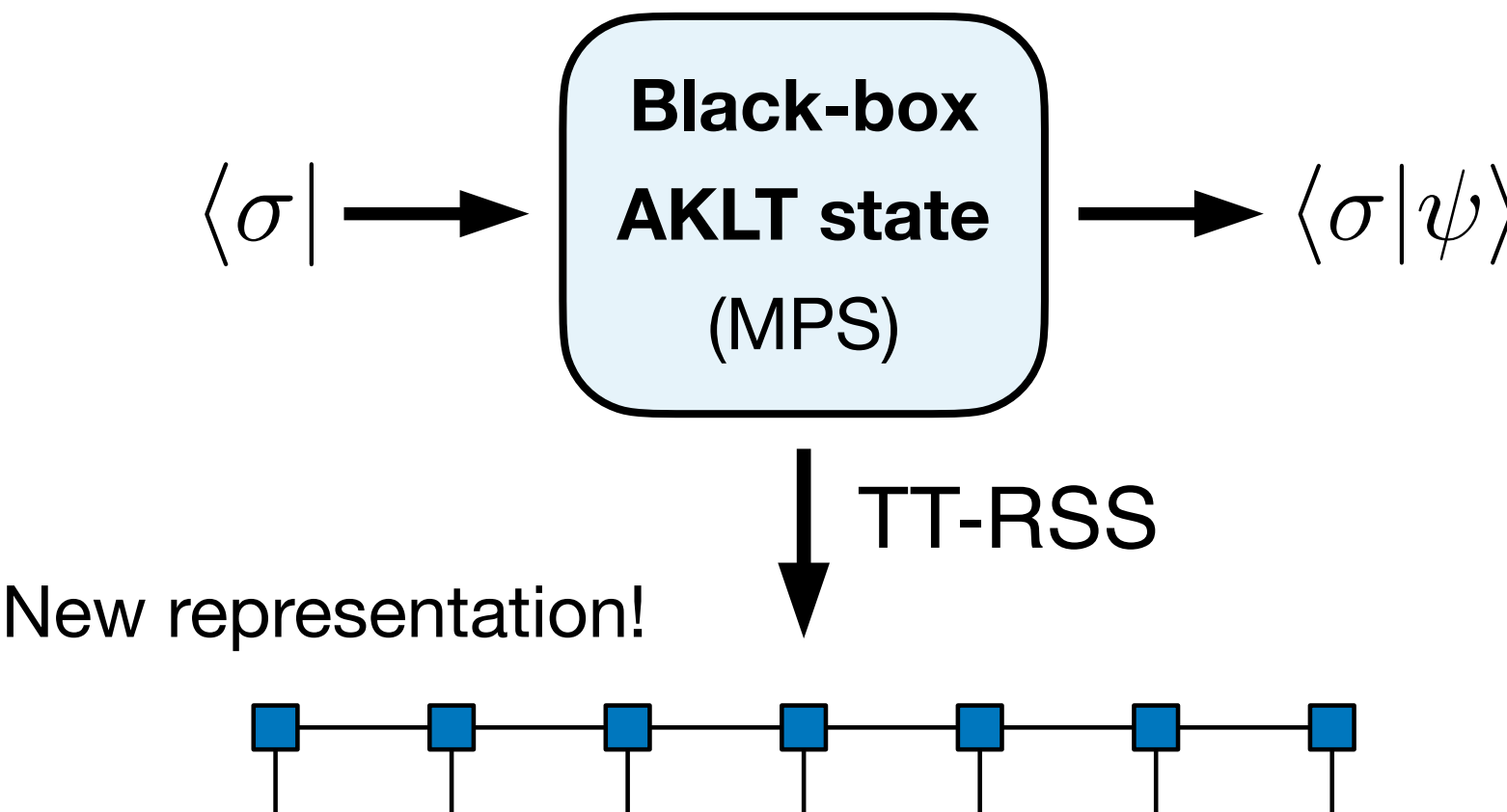
1 Topological order (proof of concept):

JRPM, A. Pozas-Kerstjens, and D. Pérez-García, *SciPost Phys. Core* 8, 095 (2025).

AKLT model:

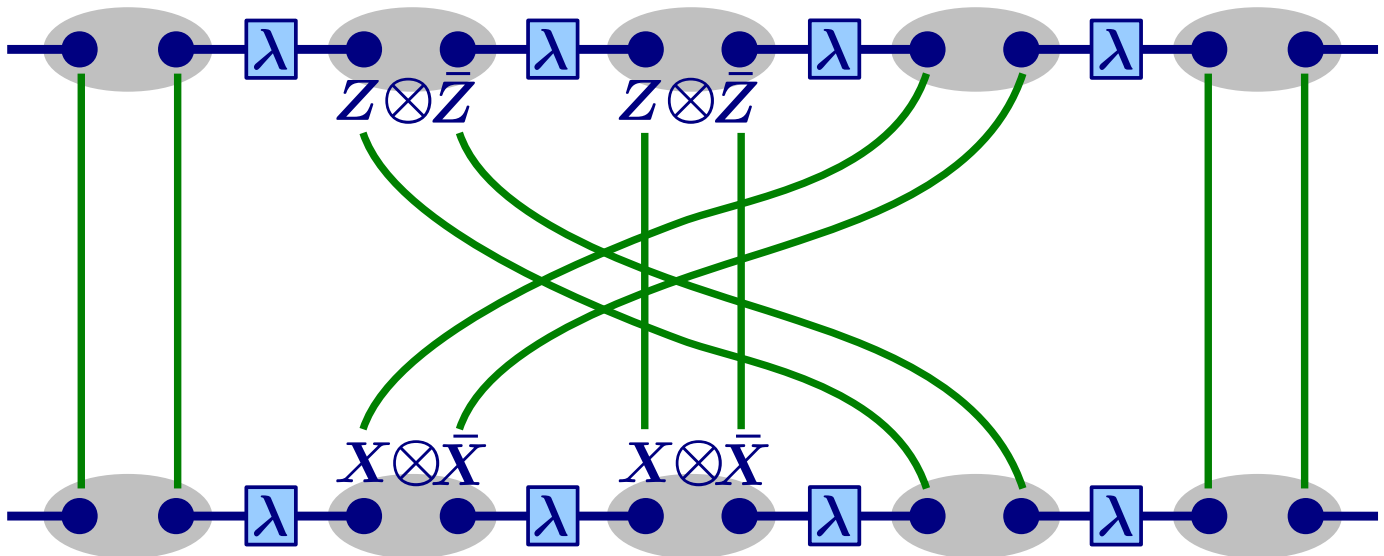
$$H = \sum_{j=1}^{N-1} P_{j,j+1}^{(2)} \equiv \sum_{j=1}^{N-1} \left[\mathbf{S}_j \cdot \mathbf{S}_{j+1} + \frac{1}{3} (\mathbf{S}_j \cdot \mathbf{S}_{j+1})^2 \right]$$

$$G^s = \sqrt{\frac{1}{3}} \sigma_s, \quad s \in \{x, y, z\}$$

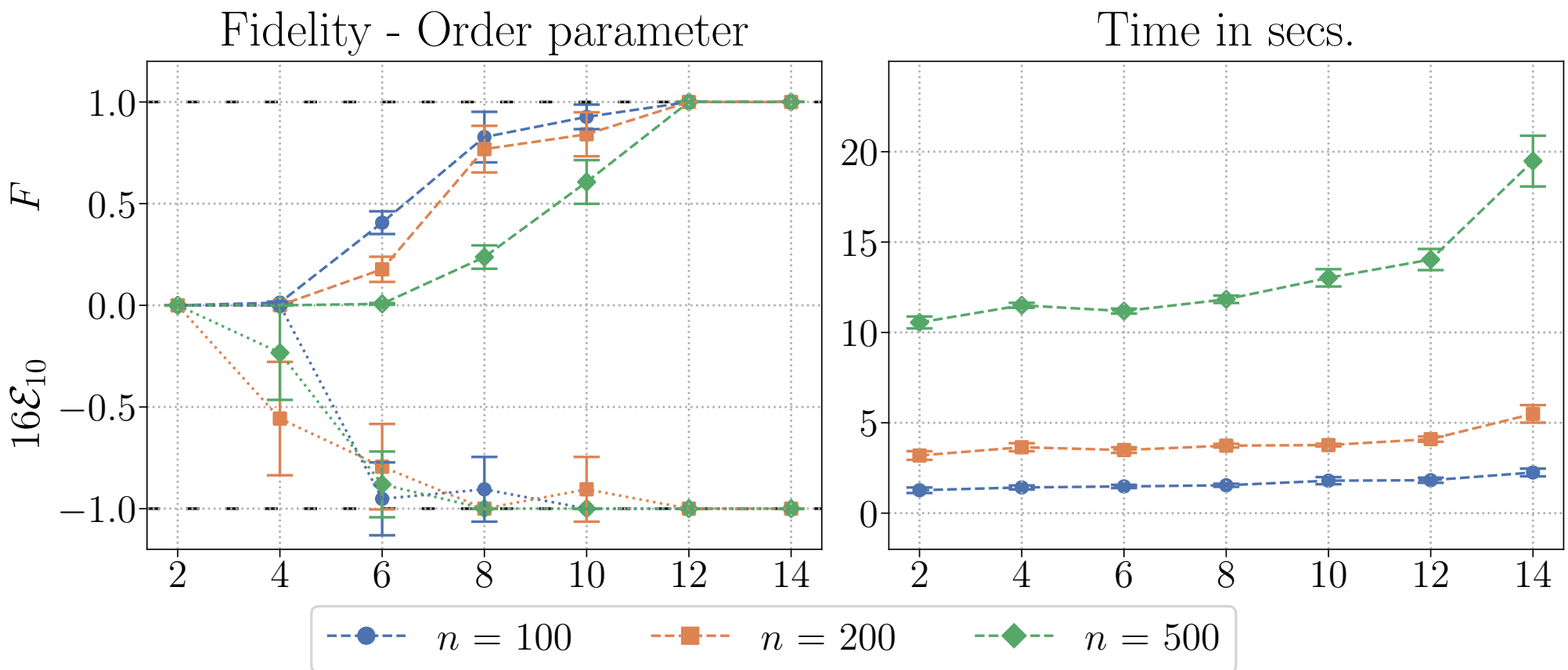


Physical symmetry induces transformation on virtual degrees of freedom:

$$u_g^{\otimes n} |\Psi\rangle = |\Psi\rangle \quad H^s = V_g^\dagger G^s V_g$$



J. Haegeman et al., *Phys. Rev. Lett.* 109, 050402 (2012).



Applications

2

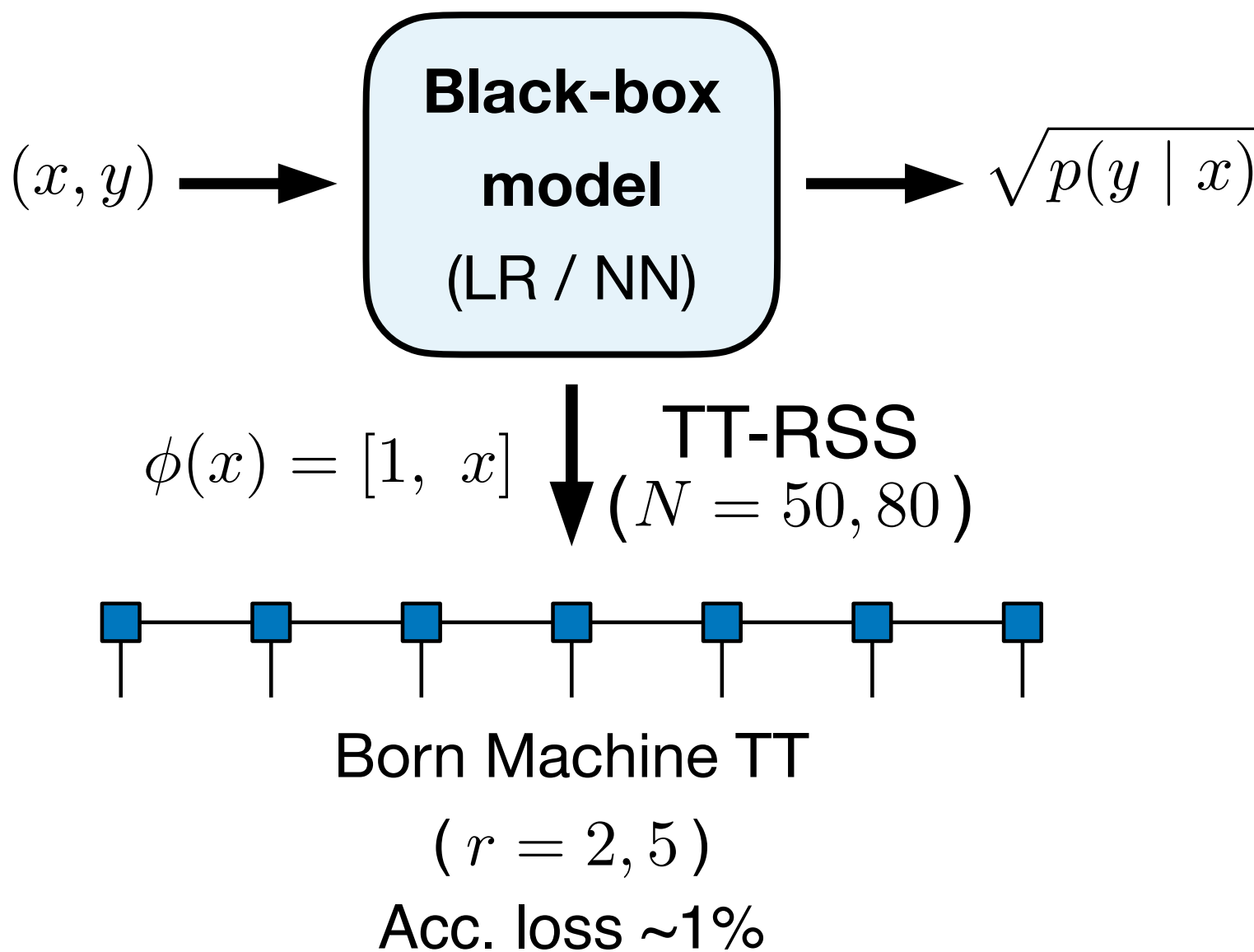
Interpretability in ML:

JRPM, J. Sinnott, R. G. Melko, and M. Kohandel, *arXiv:2602.06110* (2026).

Clinical prediction model:

- **LORIS:** LR model for immunotherapy response prediction
- 6 features + cancer type (16). ~67% bal. acc.

T.-G. Chang et al., *Nat. Cancer* 5, 1158–1175 (2024).



Interpretability of LR:

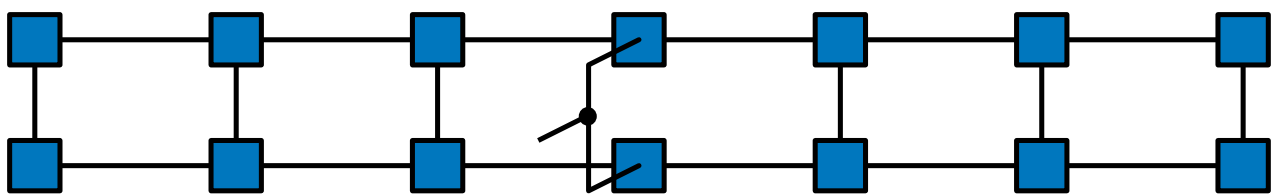
$$p(y = 1 | x) = \sigma \left(\beta_0 + \sum_{j=1}^d \beta_j x_j \right)$$

$$\Delta p_k = p(y = 1 | x_k = 1, x_{-k} = \bar{x}_{-k}) - p(y = 1 | x_k = 0, x_{-k} = \bar{x}_{-k})$$

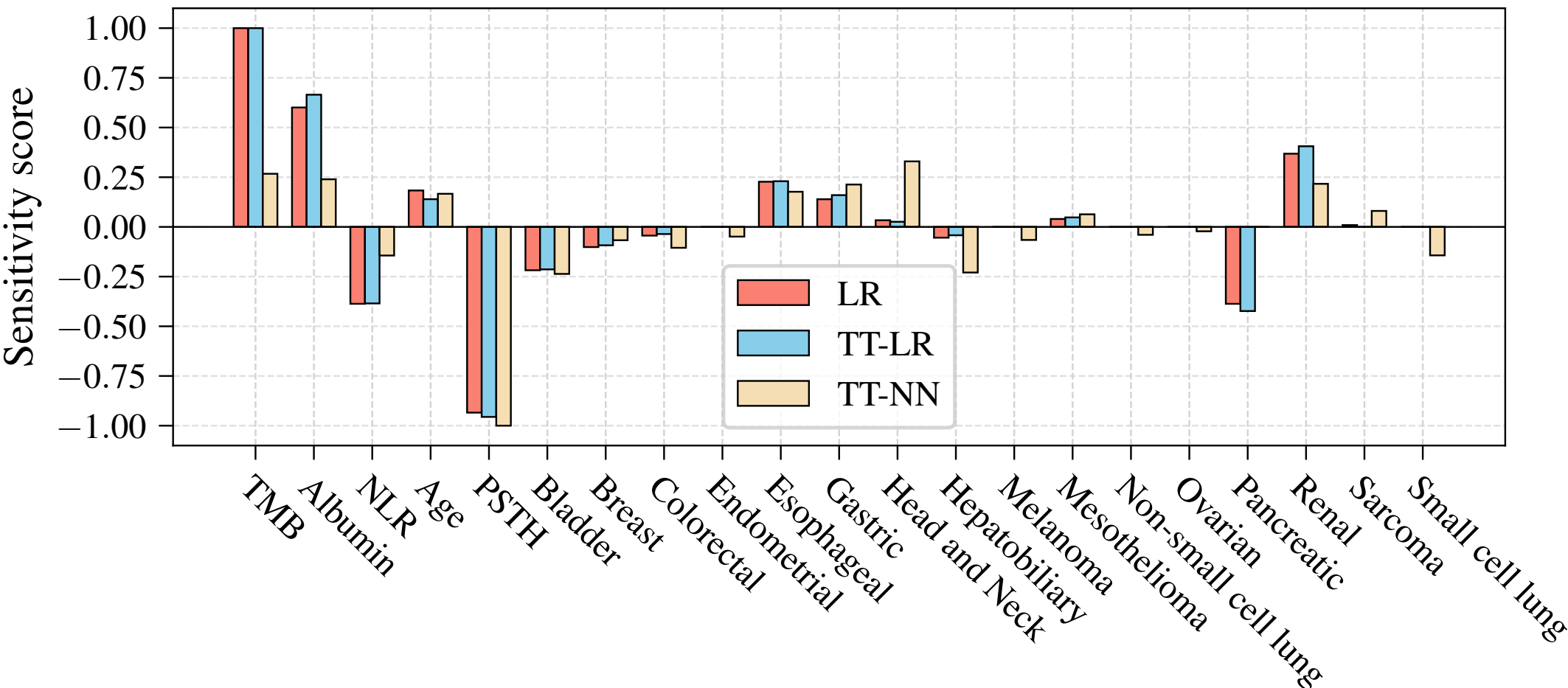
Interpretability of TT:

$$p(y | x) = |f_{\text{TT}}(x, y)|^2$$

$$p(y | x_k) = \sum_{x_{-k}} |f_{\text{TT}}(x_1, \dots, x_k, \dots, x_n, y)|^2$$



Feature sensitivity



Applications

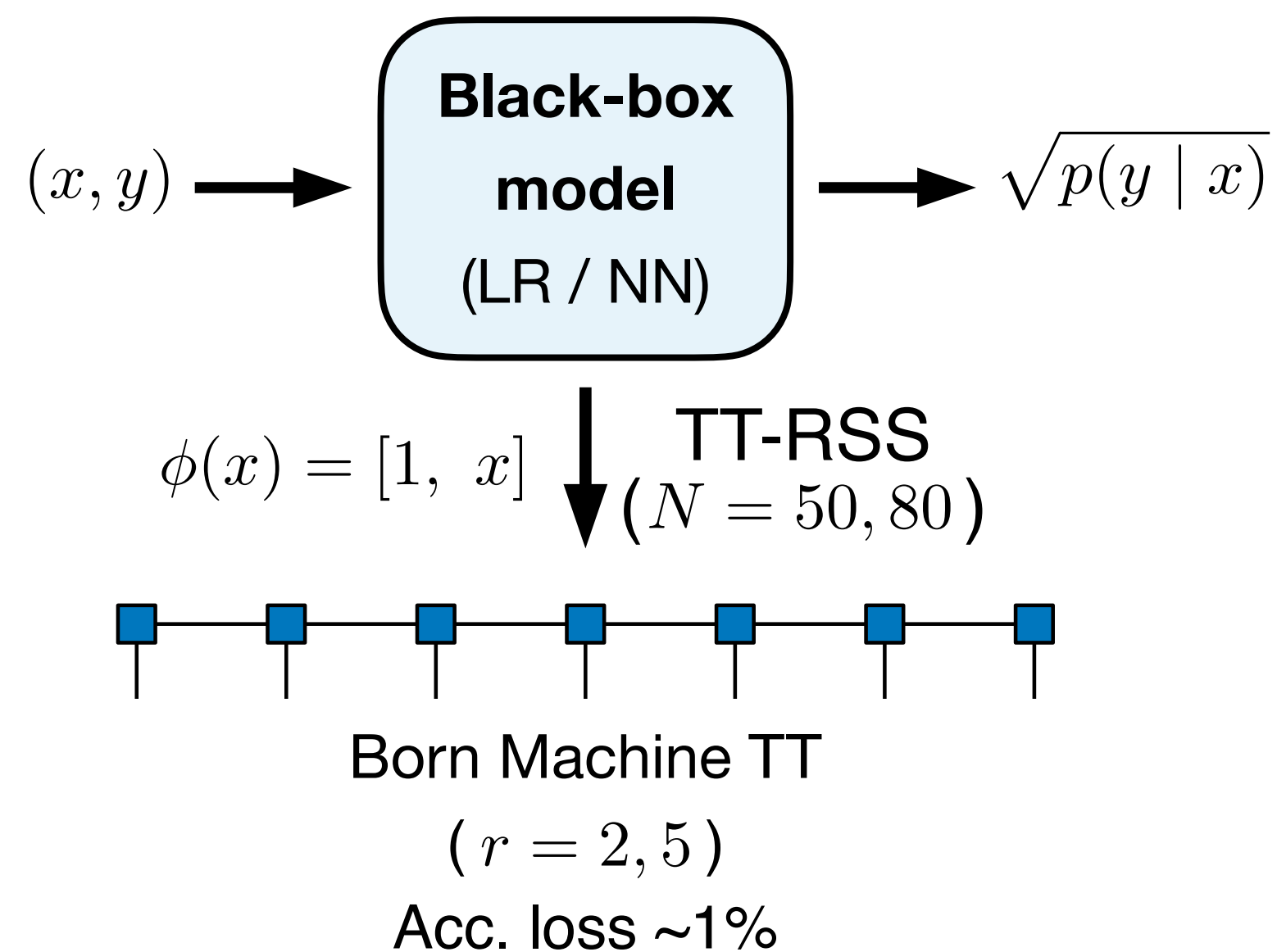
2 Interpretability in ML:

JRPM, J. Sinnott, R. G. Melko, and M. Kohandel, *arXiv:2602.06110* (2026).

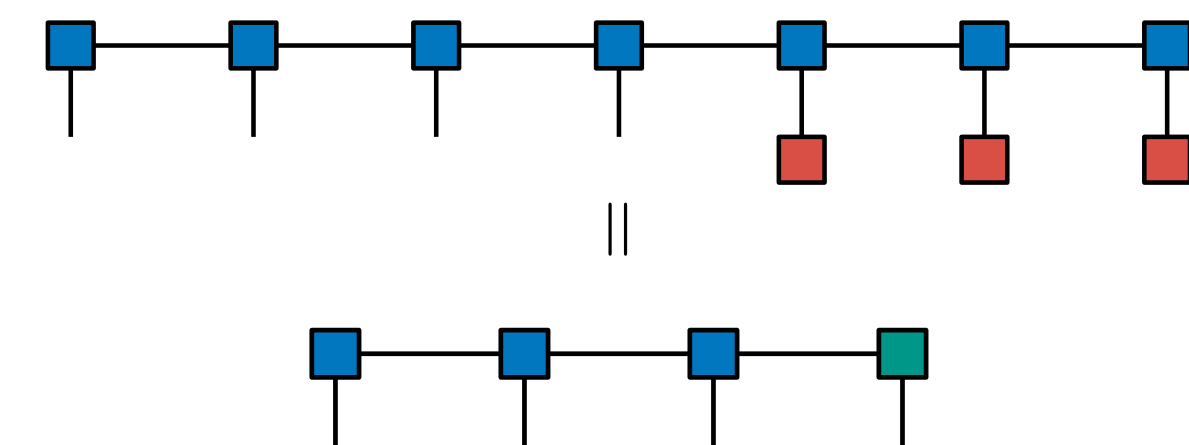
Clinical prediction model:

- **LORIS**: LR model for immunotherapy response prediction
- 6 features + cancer type (16). ~67% bal. acc.

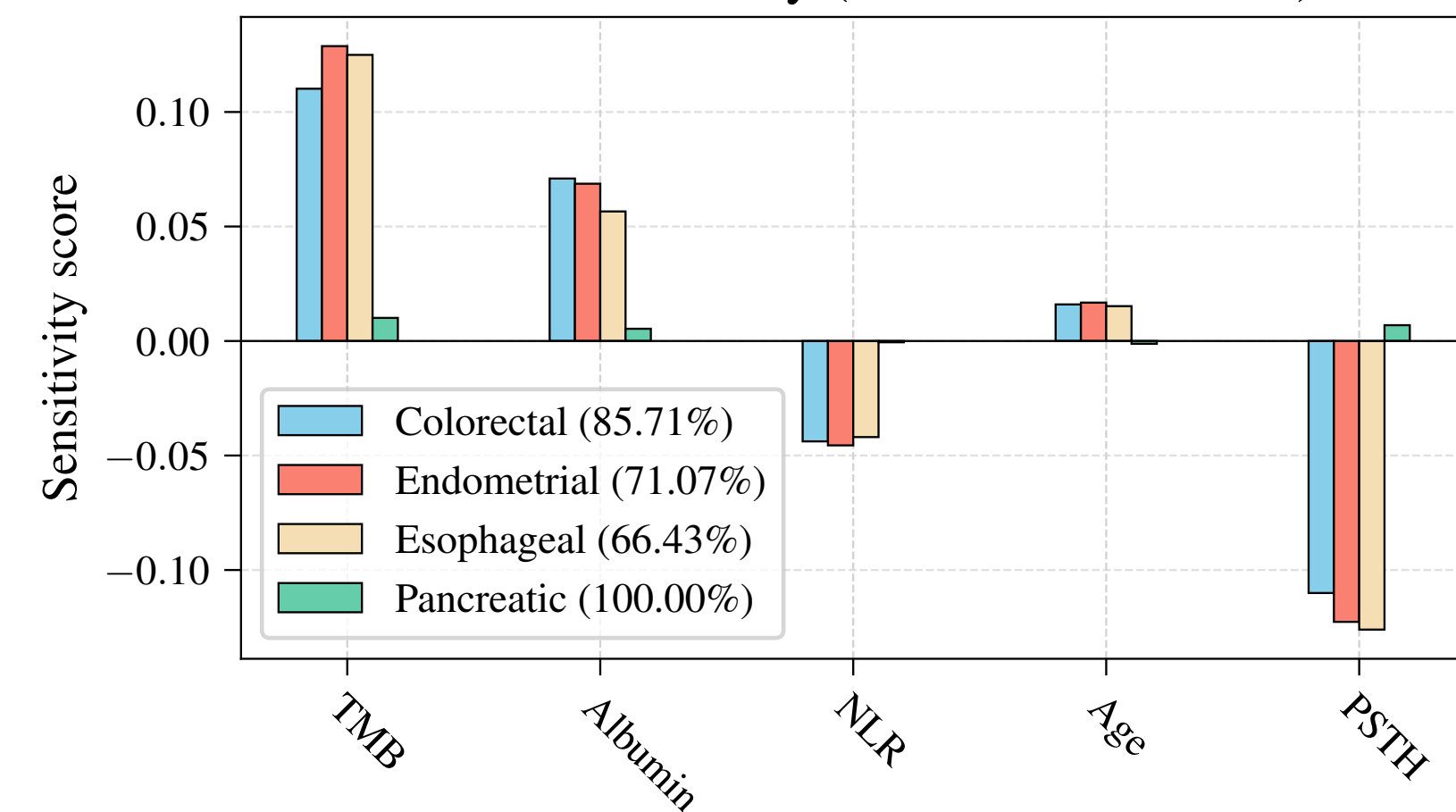
T.-G. Chang et al., *Nat. Cancer* 5, 1158–1175 (2024).



Cancer-type-specific models



Feature sensitivity (conditioned TT-LR)



Applications

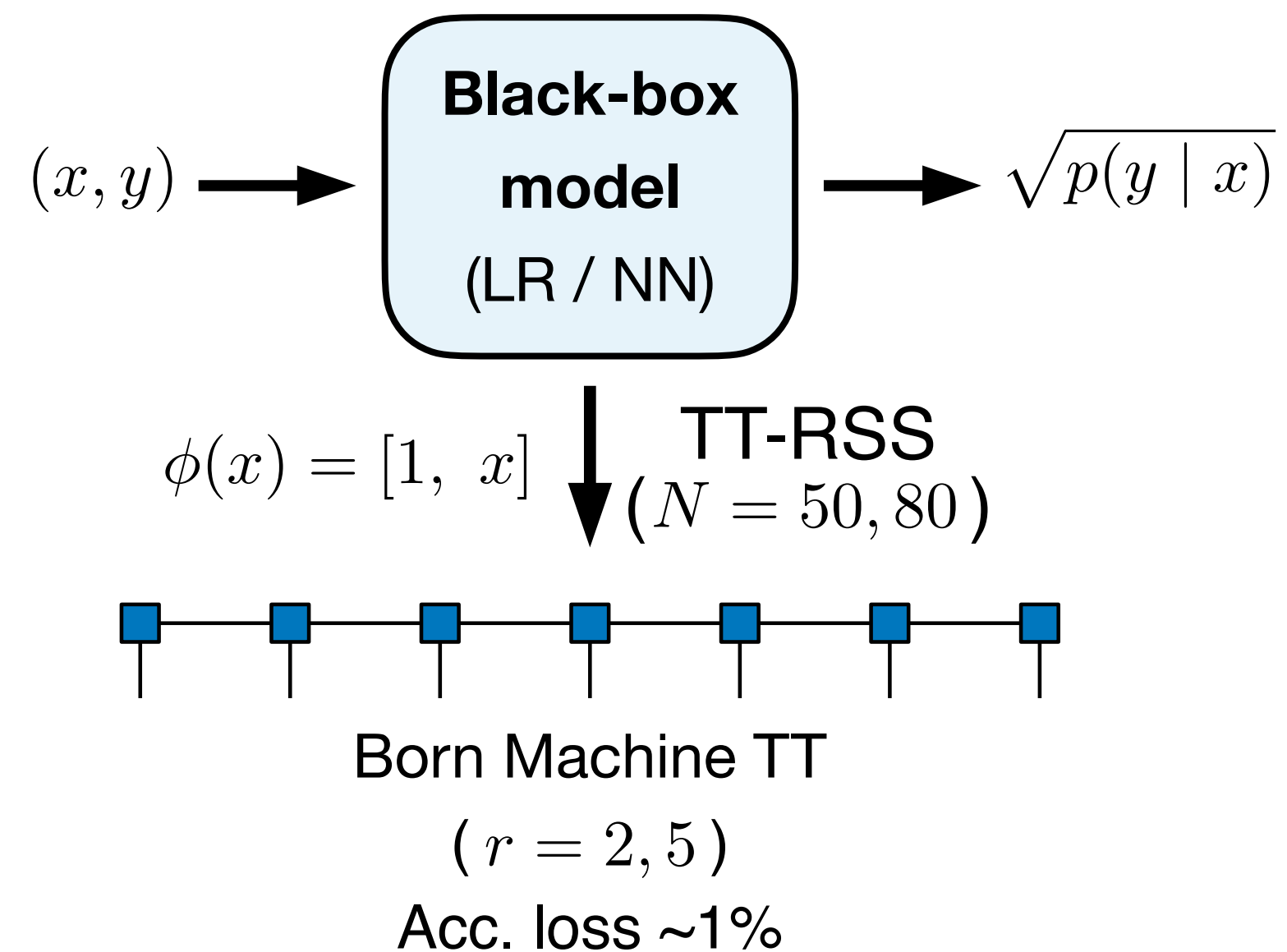
2 Interpretability in ML:

JRPM, J. Sinnott, R. G. Melko, and M. Kohandel, *arXiv:2602.06110* (2026).

Clinical prediction model:

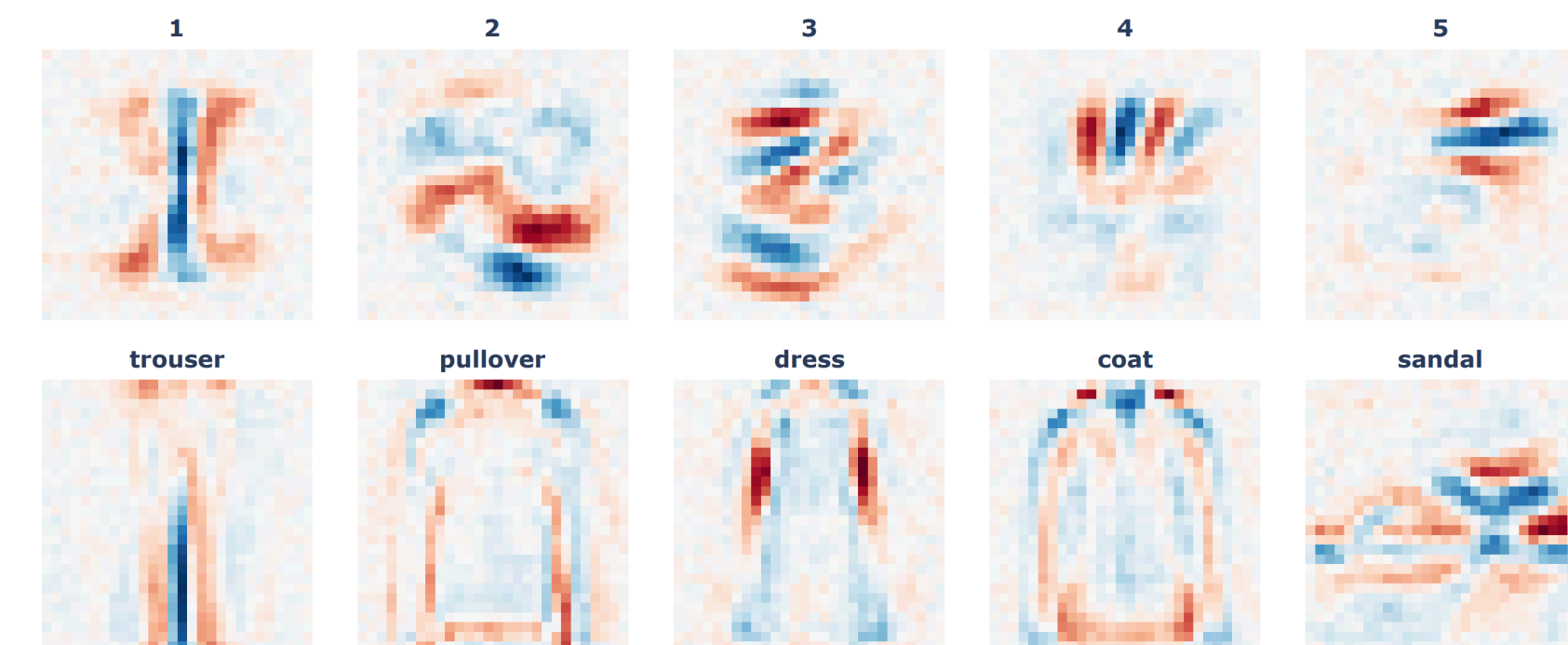
- **LORIS:** LR model for immunotherapy response prediction
- 6 features + cancer type (16). ~67% bal. acc.

T.-G. Chang et al., *Nat. Cancer* 5, 1158–1175 (2024).



Exploit virtual representations?

Weight-based interpretability



M. T. Pearce et al., *Int. Conf. Learn. Represent.* (2025).

Applications

3 Interpretability of NQS (work in progress):

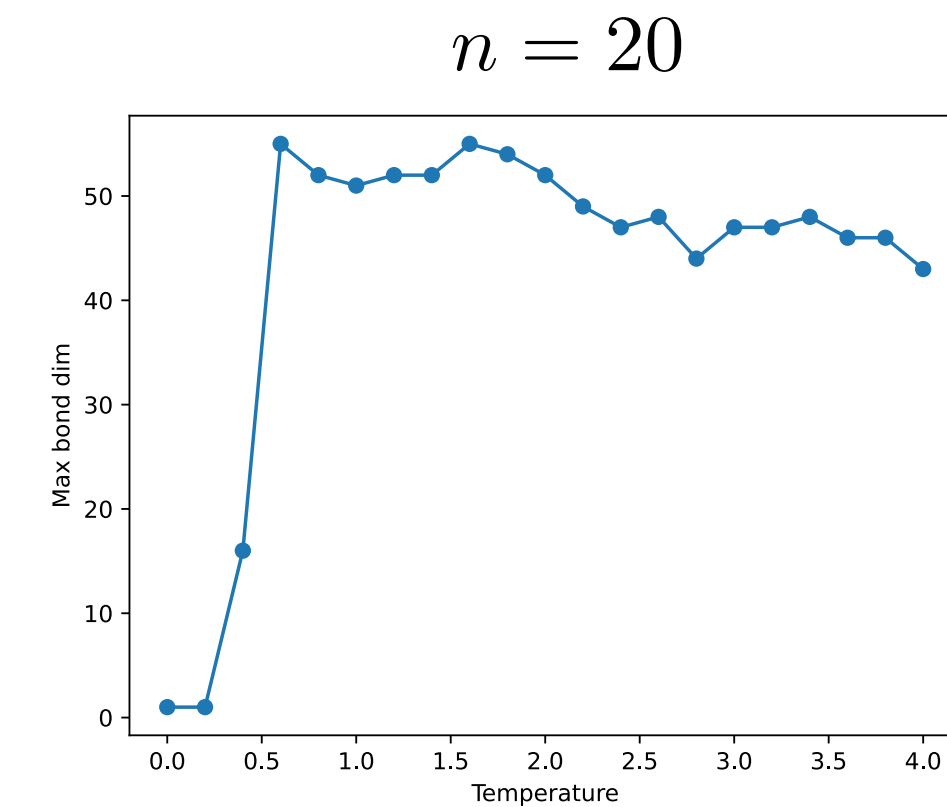
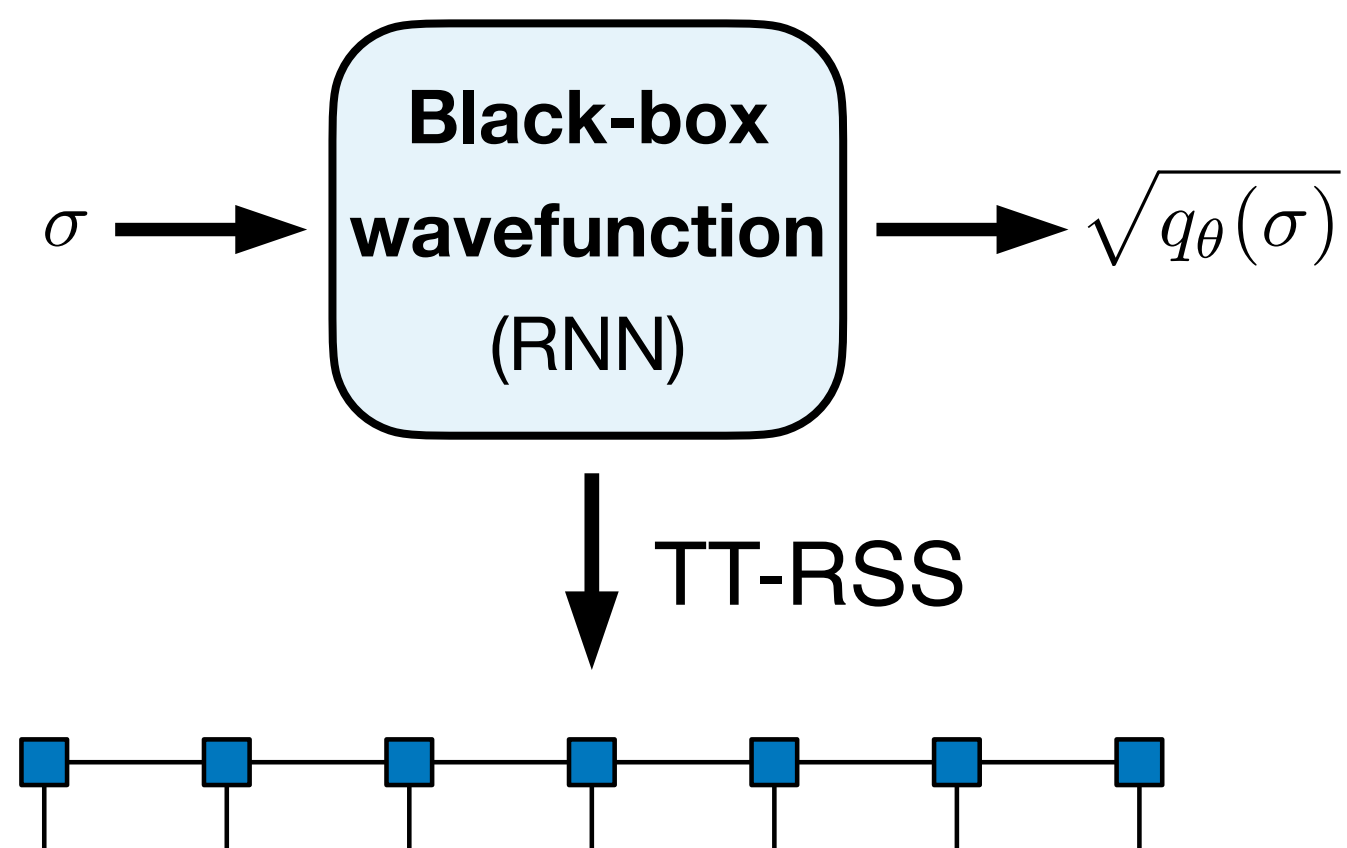
Joint work with Estelle Inack, Nishan Ranabhat and Vichayuth Imchitr

Variational optimization:

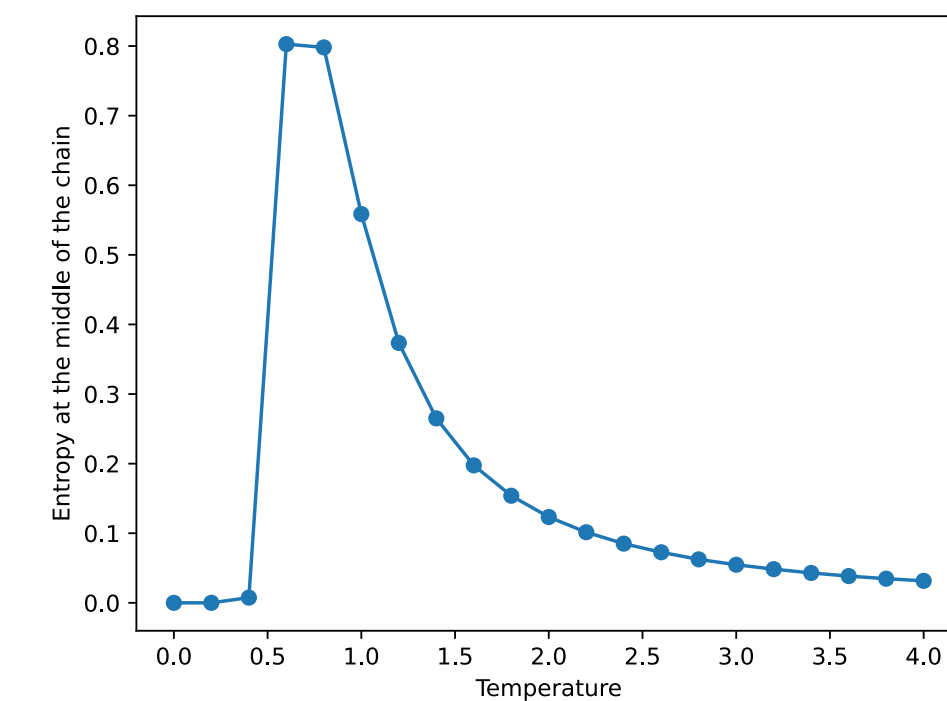
Fully connected Ising model

$$H(\sigma) = - \sum_{i < j} J_{ij} \sigma_i \sigma_j = - \frac{1}{N} \sum_{i < j} \sigma_i \sigma_j$$

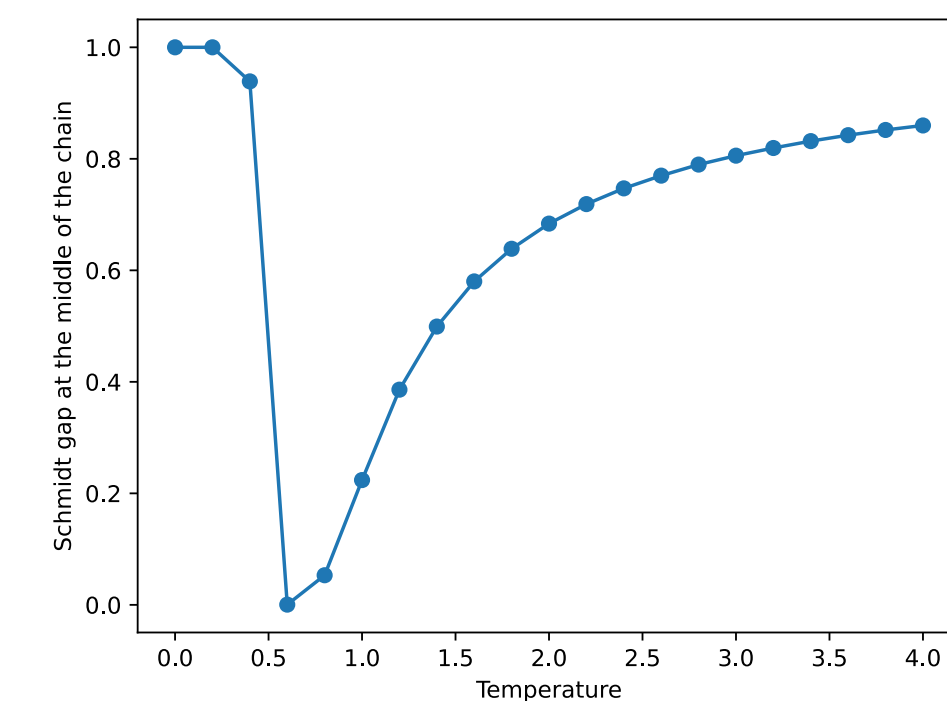
$$q_{\theta}(\sigma) \approx p_T(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z}$$



$$\max_k D_k$$



$$S_A = -\text{Tr}(\rho_A \log \rho_A)$$



$$\Delta_S = s_1^2 - s_2^2$$

Applications

3

Interpretability of NQS (work in progress):

Joint work with Estelle Inack, Nishan Ranabhat and Vichayuth Imchitr

Variational neural annealing:

Portfolio optimization

N. Ranabhat et al., *arXiv:2507.07159* (2025).

For month m :

$$\min_{x_i \in \{0,1\}} C_m(x) = -\mu_m^\top x + \lambda x^\top \Sigma_m x$$

Classical

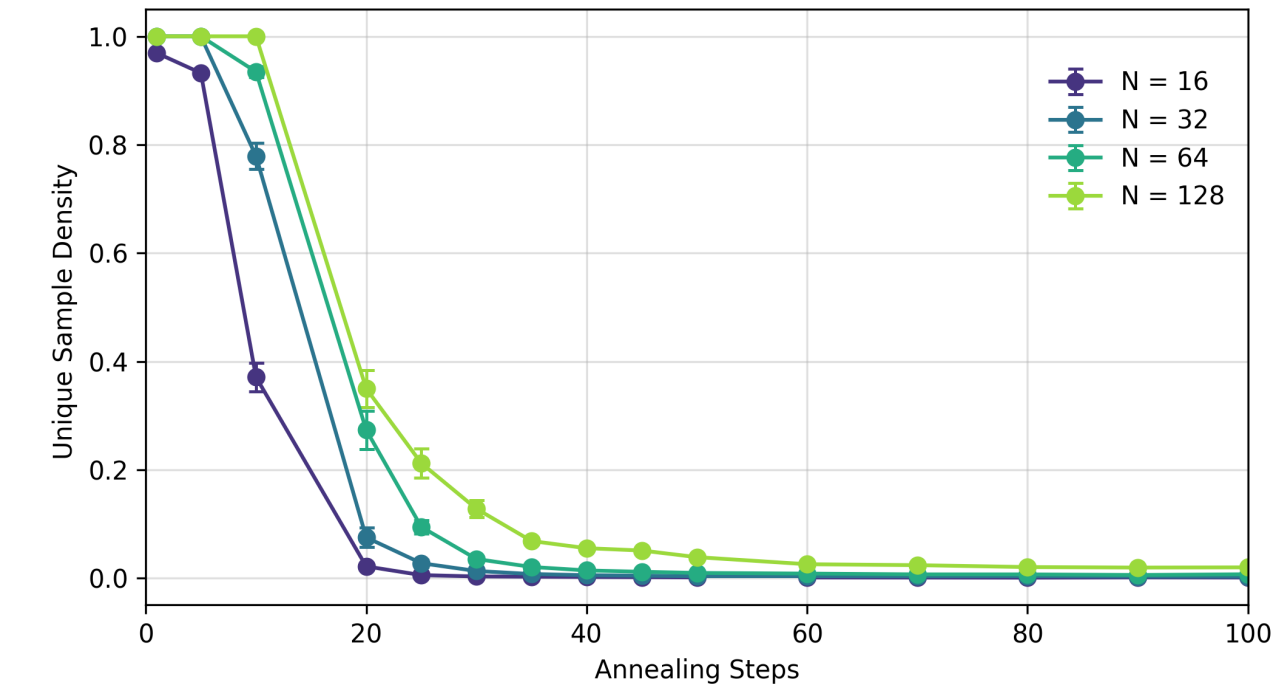
$$H_C^{(m)} = \sum_{x \in \{0,1\}^N} C_m(x) |x\rangle \langle x|$$

$$F_T(\theta) = \mathbb{E}_{q_\theta} [C(x) + T \log q_\theta(x)]$$

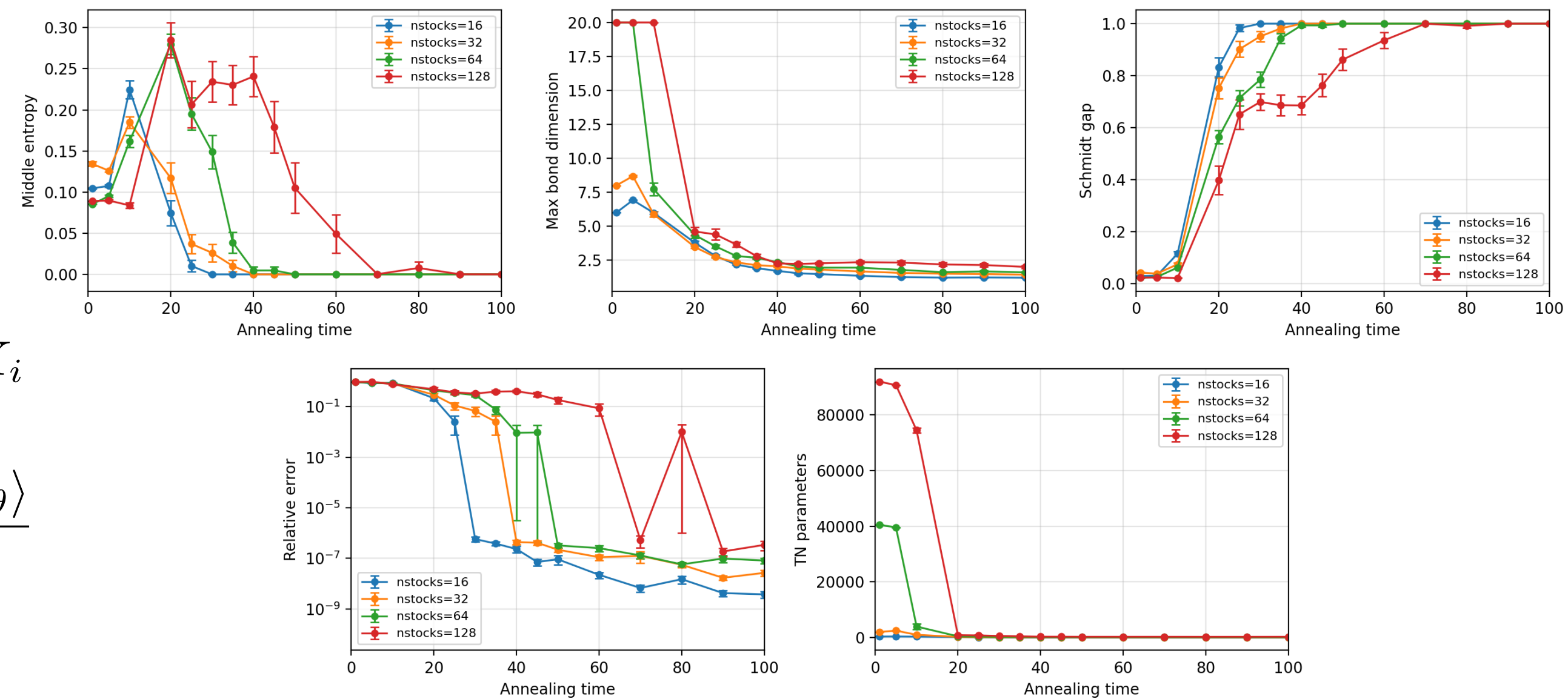
Quantum

$$H(\Gamma) = H_C - \Gamma \sum_i X_i$$

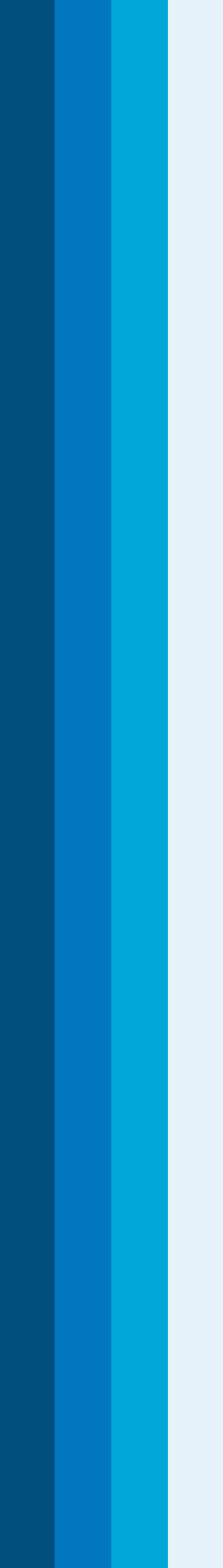
$$E_\theta(\Gamma) = \frac{\langle \psi_\theta | H(\Gamma) | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}$$



Month-averaged TT metrics vs annealing time (all nstocks) (rnn_type=discrete)



Compression ratio ~900%

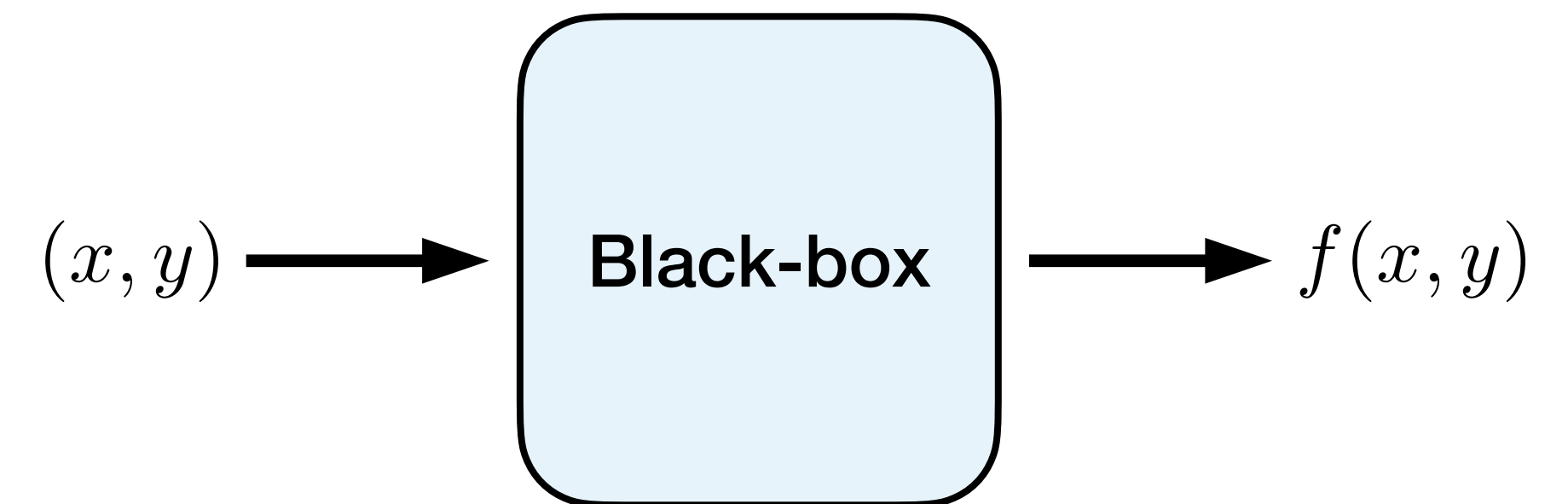
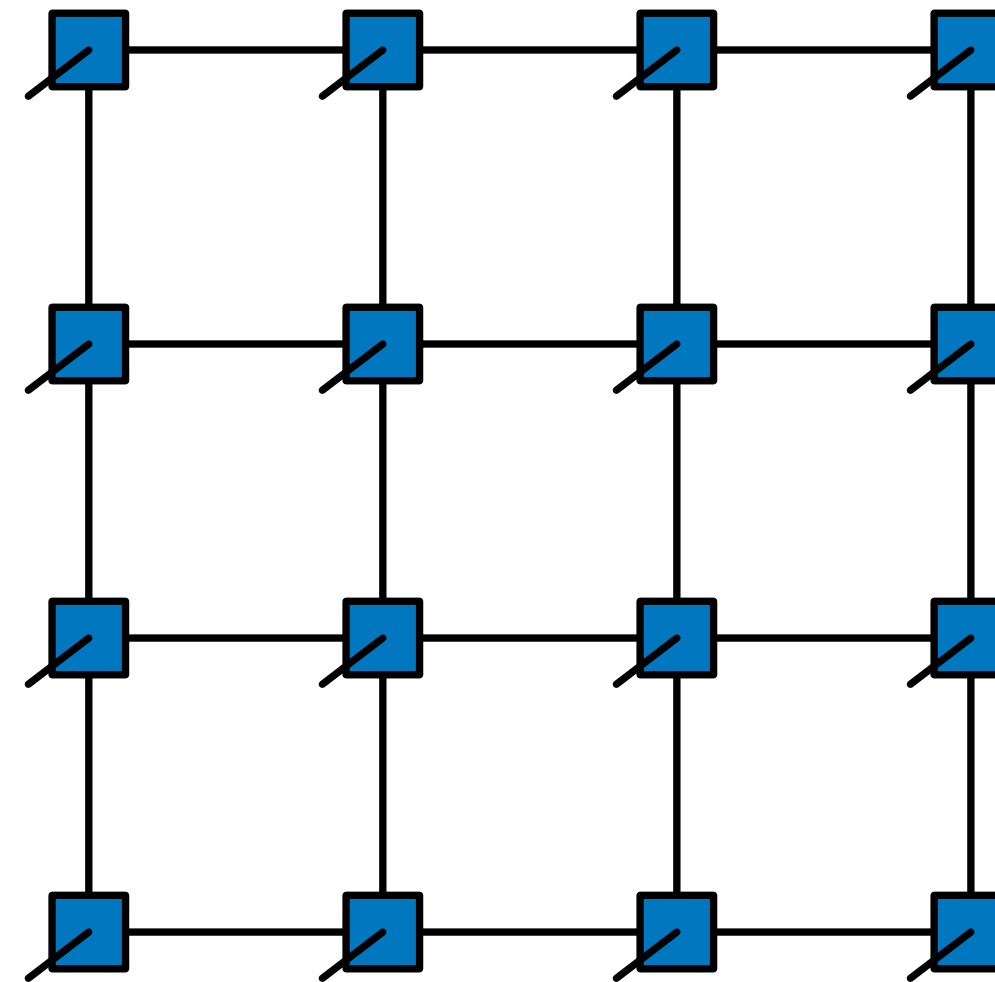


Next steps

Next steps

Is it possible to tensorize into **PEPS** form?

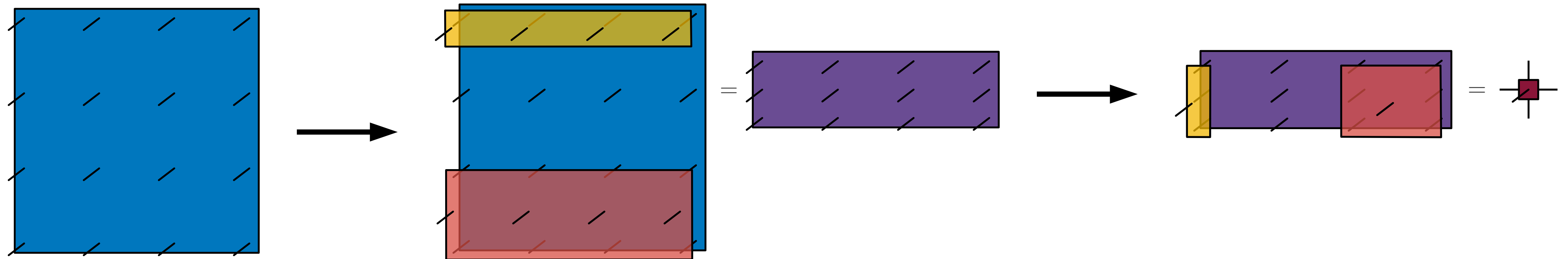
PEPS methods (optimization)
are super **expensive**!



Next steps

Is it possible to tensorize into **PEPS** form?

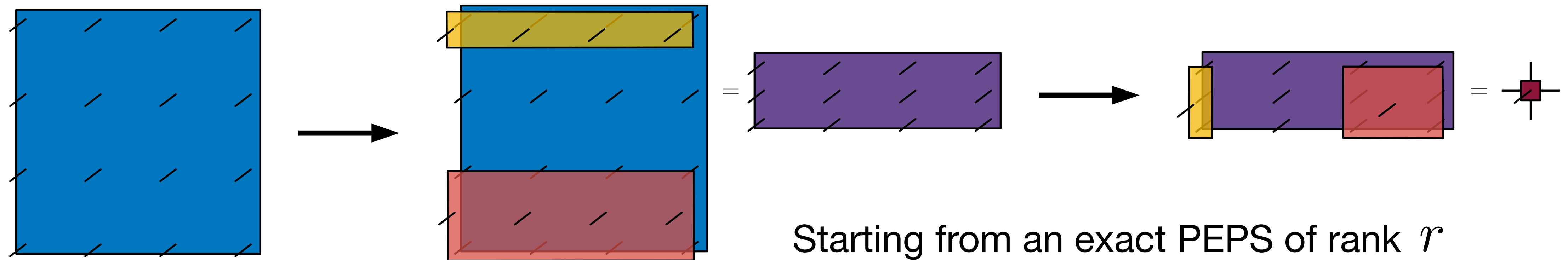
Kind of...



Next steps

Is it possible to tensorize into **PEPS** form?

Kind of...



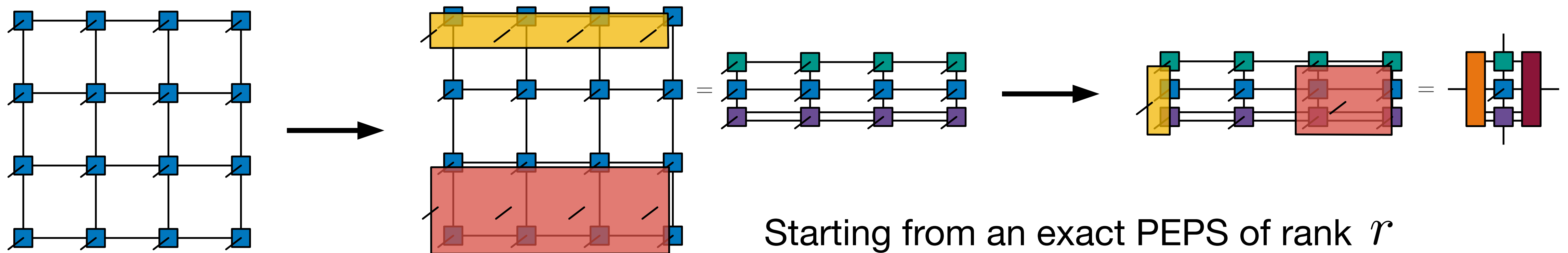
Starting from an exact PEPS of rank r

The reconstructed PEPS has rank $O(r^{n/2})!$

Next steps

Is it possible to tensorize into **PEPS** form?

Kind of...

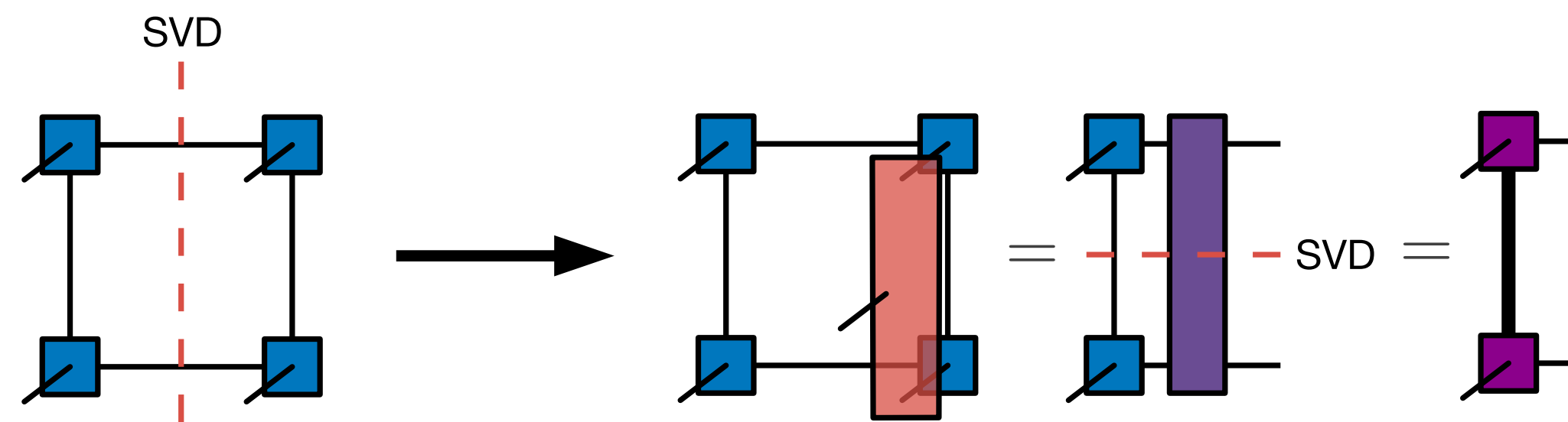


Starting from an exact PEPS of rank r

The reconstructed PEPS has rank $O(r^{n/2})$!

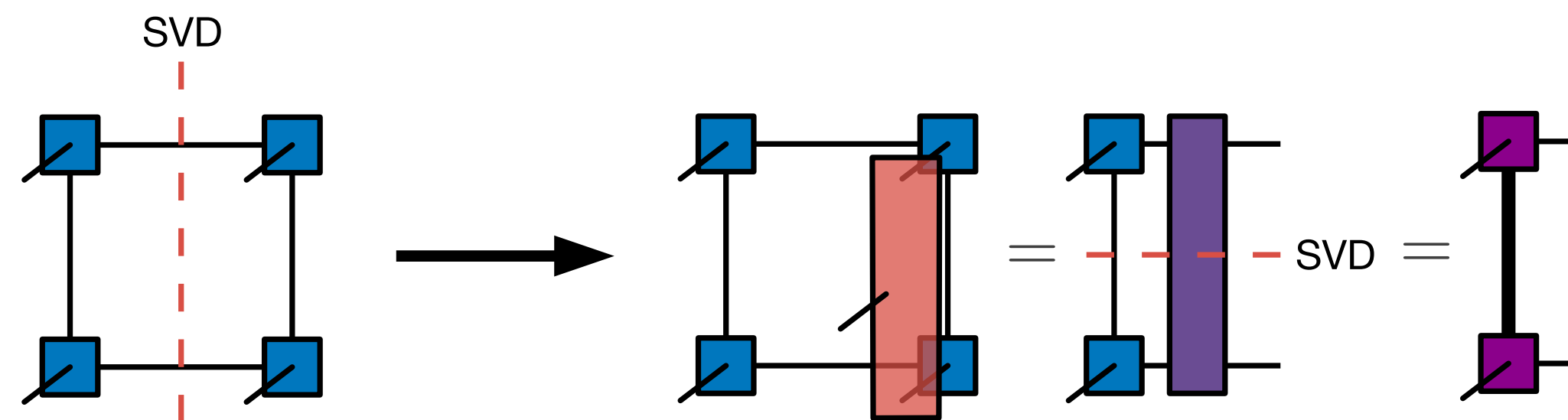
Next steps

The same happens for **Tensor Rings**



Next steps

The same happens for **Tensor Rings**

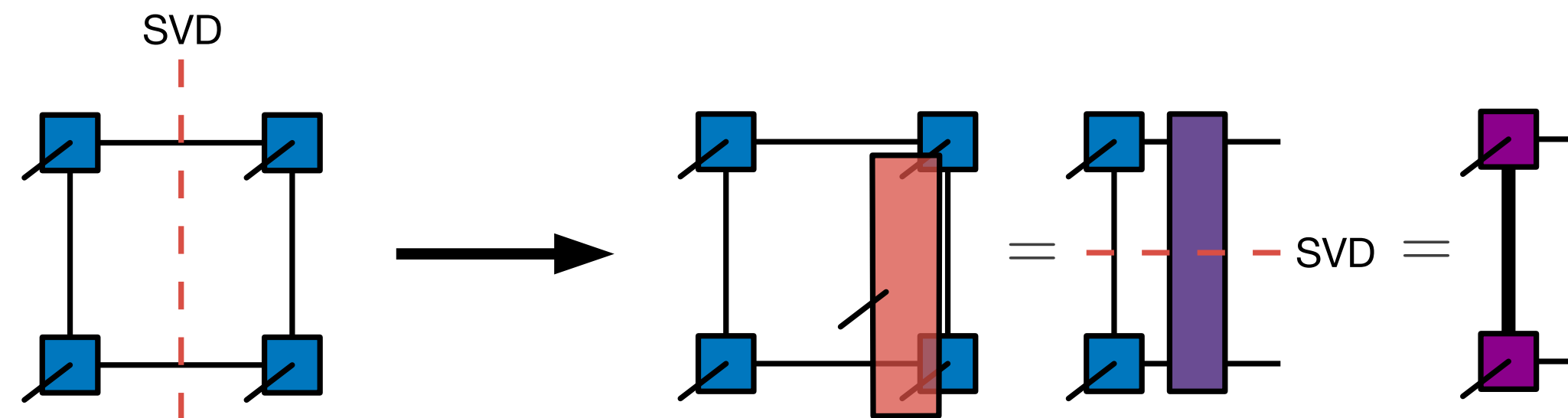


Loops are problematic...

Is it possible to *open loops* exactly?

Next steps

The same happens for **Tensor Rings**



Theory

BLOSTR:

- Solves exact case
- Very **fragile** to perturbations
- **Infeasible** for large tensors

H. Chen, S. Chen, and A. R. Zhang, *arXiv:2512.01016* (2025).

Loops are problematic...

Is it possible to *open loops* exactly?

Practice

ALS:

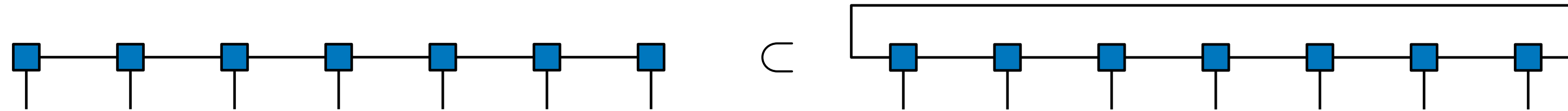
$$G_k^* = \arg \min_{G_k} \|\mathcal{X} - \text{TR}(G_1, \dots, G_k, \dots, G_N)\|_F^2$$

- Solves exact and perturbed cases
- Relies on **optimization** (instabilities, initialization, etc.)
- **Infeasible** for large tensors

Q. Zhao et al., *arXiv:1606.05535* (2016).

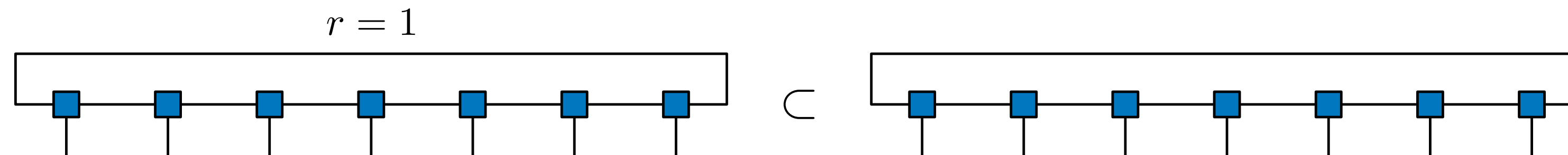
Next steps

Our approach:
from Tensor Trains to Tensor Rings



Next steps

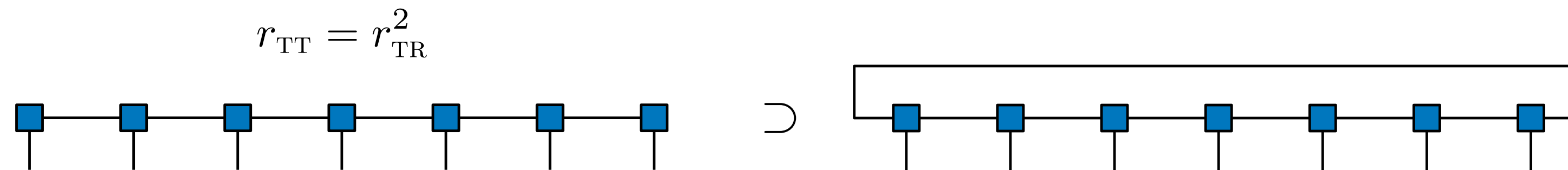
Our approach:
from Tensor Trains to Tensor Rings



Next steps

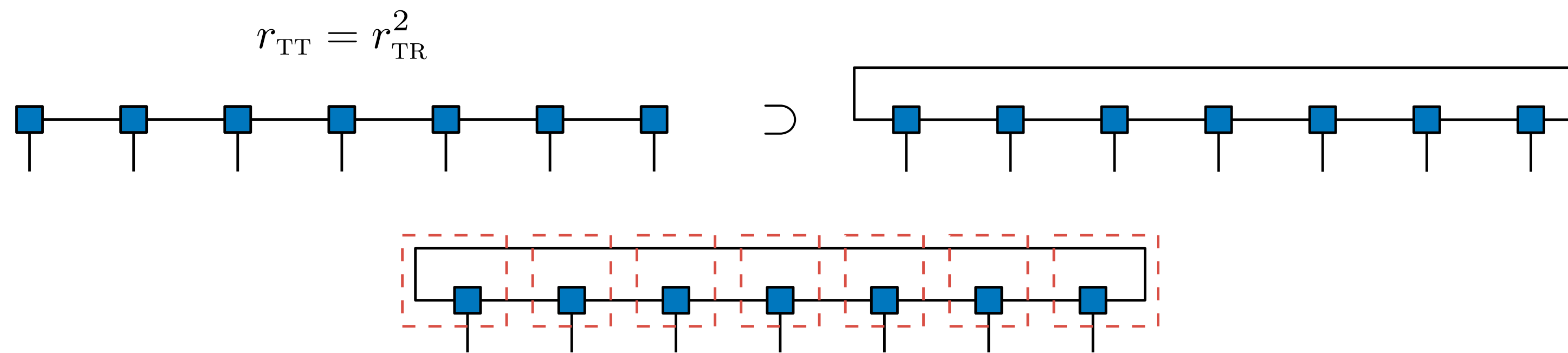
Our approach:

from Tensor Trains to Tensor Rings



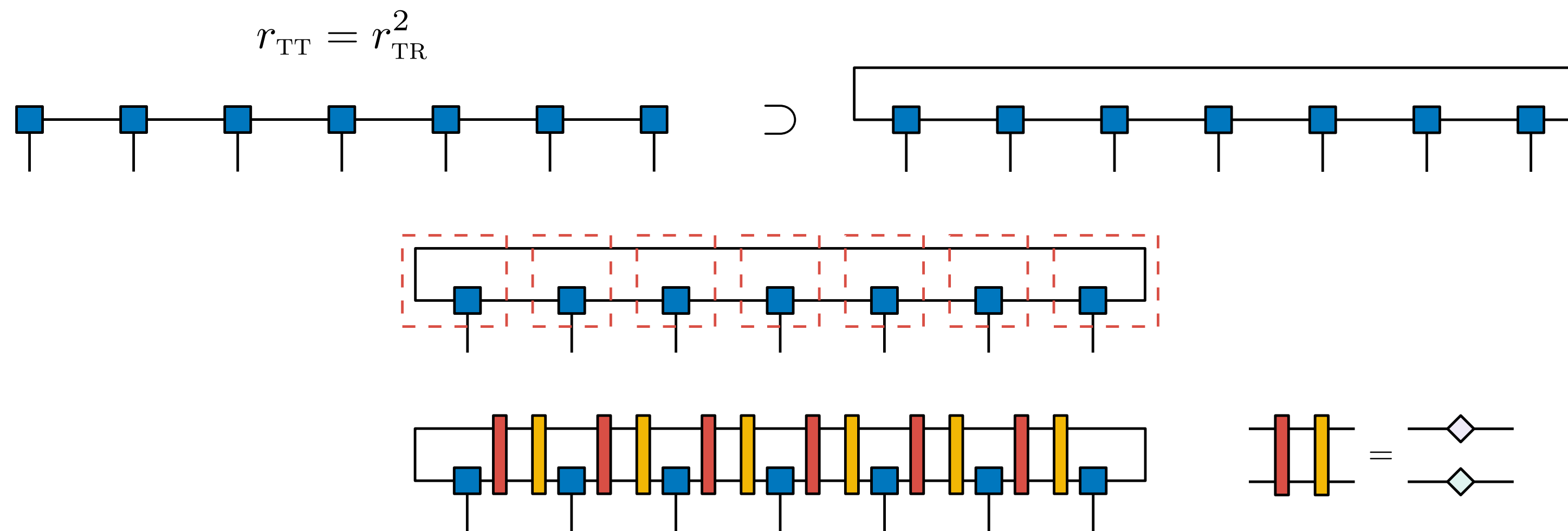
Next steps

Our approach:
from Tensor Trains to Tensor Rings



Next steps

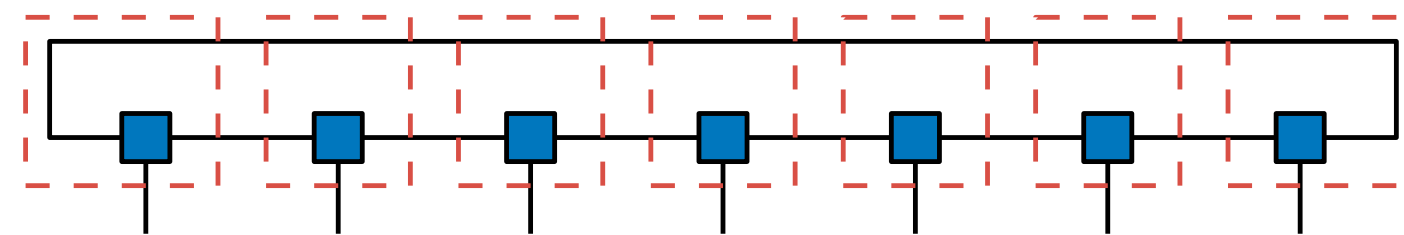
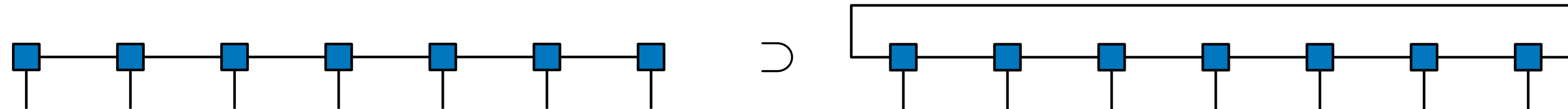
Our approach:
from Tensor Trains to Tensor Rings



Next steps

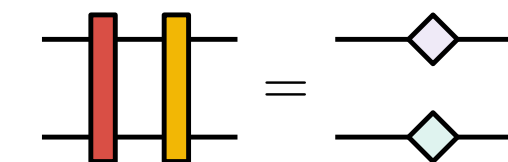
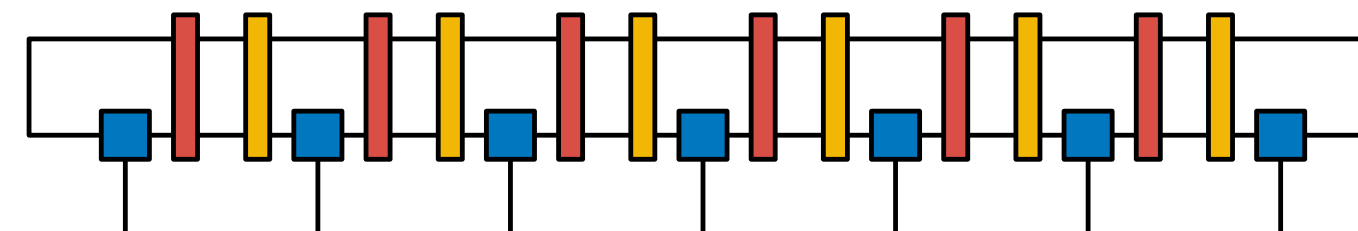
Our approach: from Tensor Trains to Tensor Rings

$$r_{\text{TT}} = r_{\text{TR}}^2$$



Strategy:

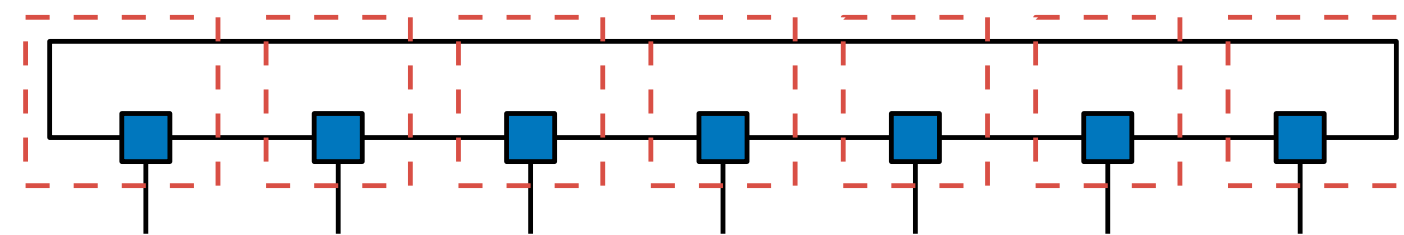
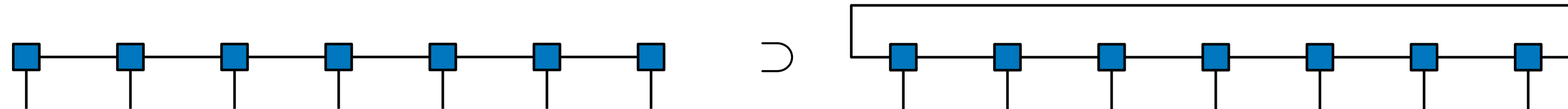
- 1 Tensorize into TT form



Next steps

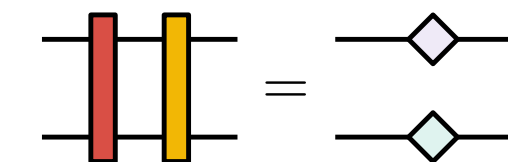
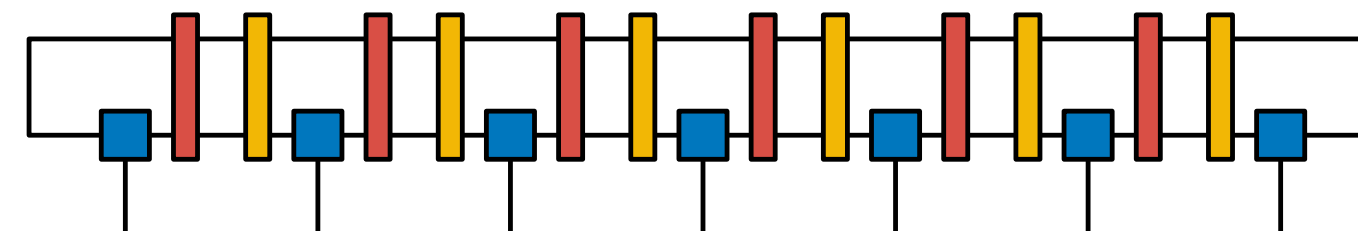
Our approach: from Tensor Trains to Tensor Rings

$$r_{\text{TT}} = r_{\text{TR}}^2$$

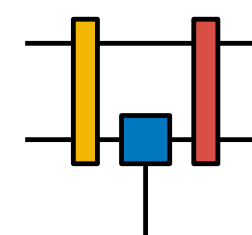


Strategy:

- 1 Tensorize into TT form



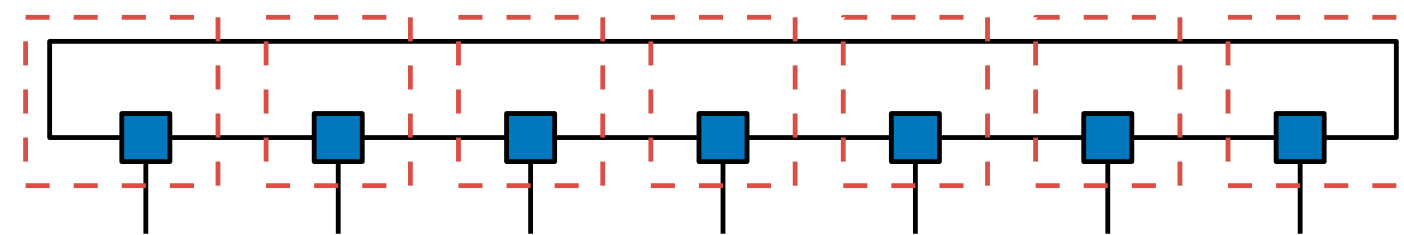
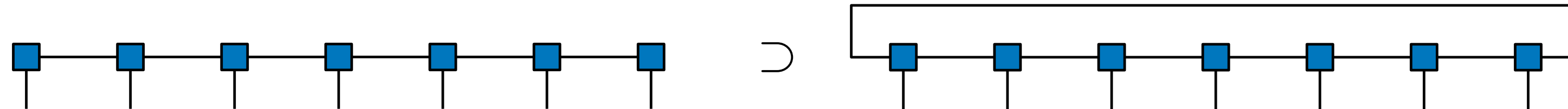
- 2 Open central loop (BLOSTR, ALS)



Next steps

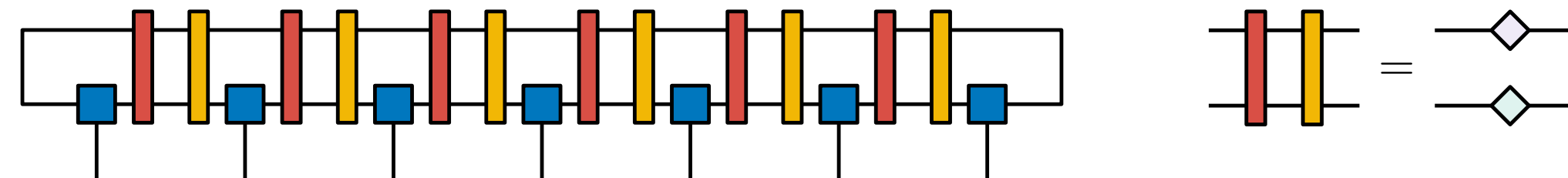
Our approach: from Tensor Trains to Tensor Rings

$$r_{\text{TT}} = r_{\text{TR}}^2$$

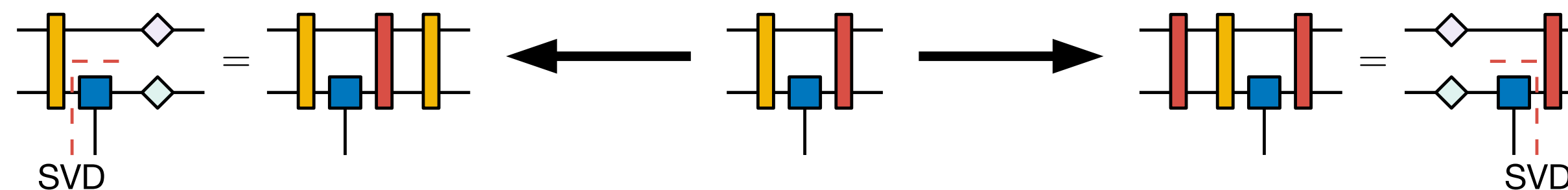


Strategy:

1 Tensorize into TT form



2 Open central loop (BLOSTR, ALS)

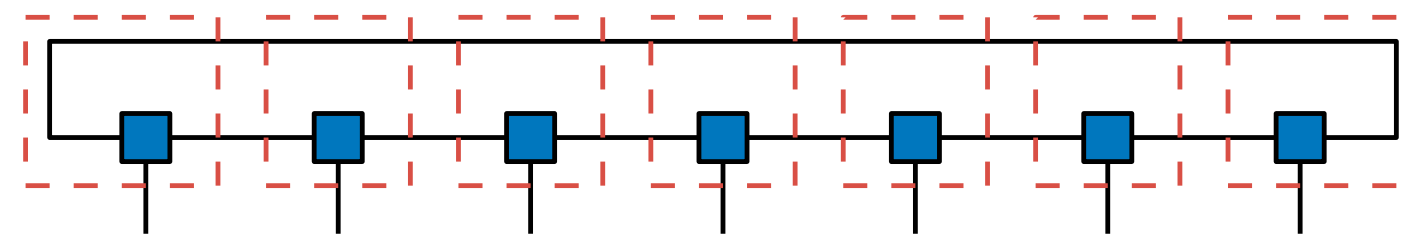
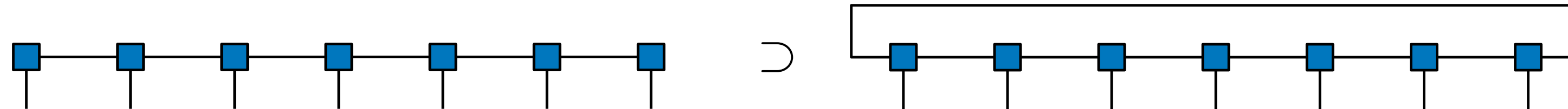


3 Propagate gauges

Next steps

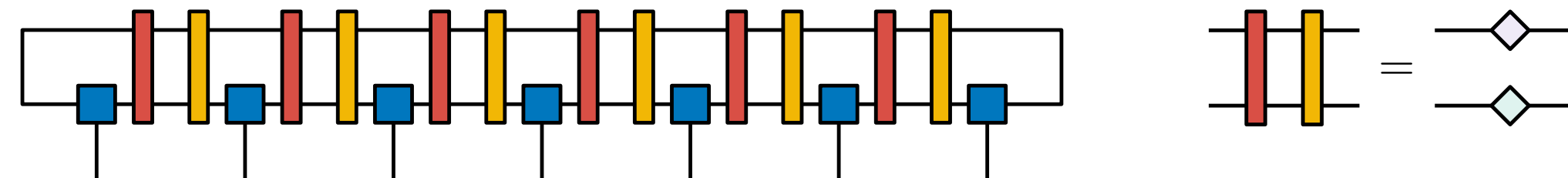
Our approach: from Tensor Trains to Tensor Rings

$$r_{\text{TT}} = r_{\text{TR}}^2$$

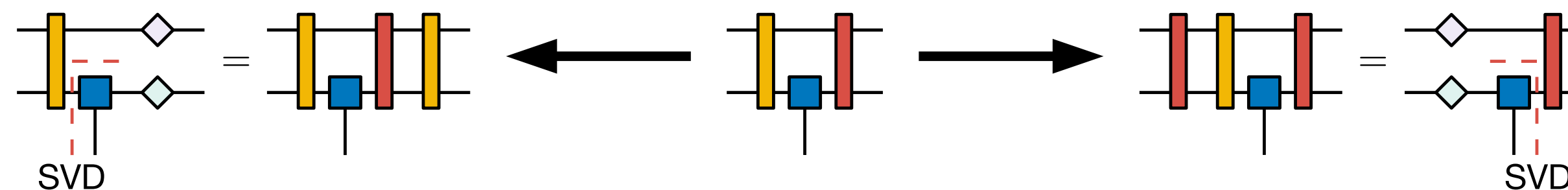


Strategy:

1 Tensorize into TT form



2 Open central loop (BLOSTR, ALS)



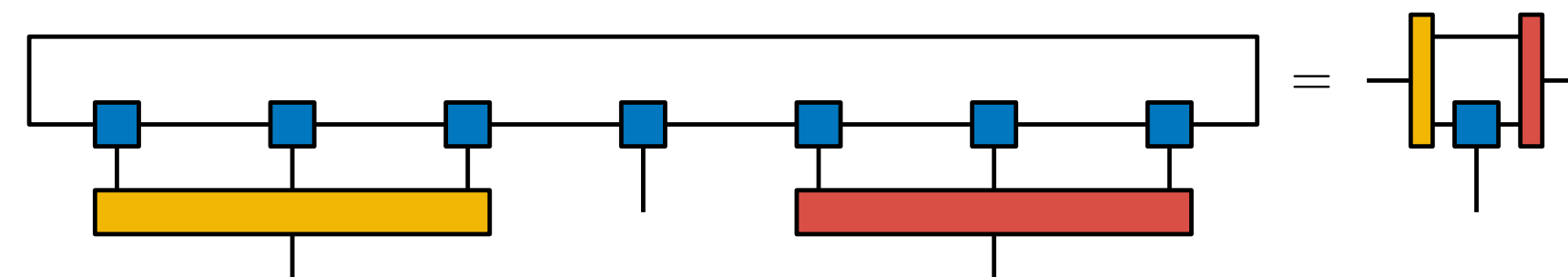
3 Propagate gauges

Similar ideas allow
to define a TR
decomposition in a
single step:

TR-RSS

Next steps

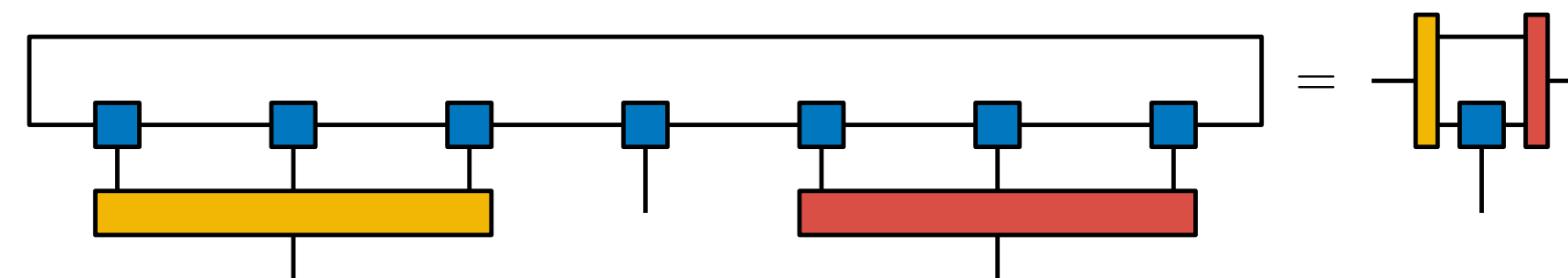
TR-RSS



- 1 Open central loop to find **explicit gauges / environments**

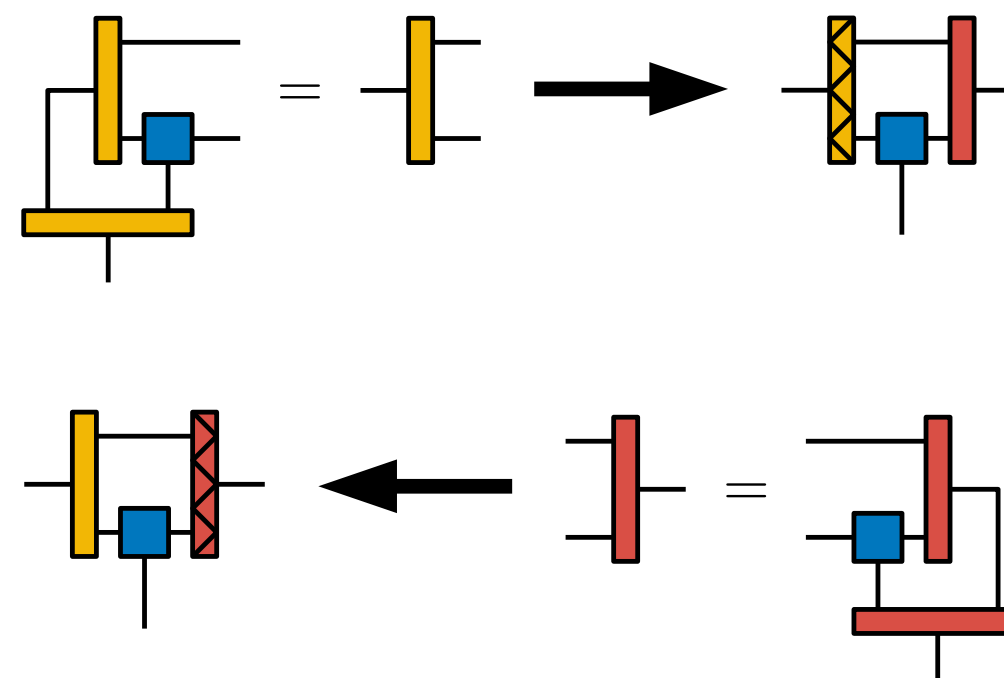
Next steps

TR-RSS



1 Open central loop to find explicit gauges / environments

2 Build next environments recursively and fix in next step



Tensorization

Performance

Exact TR				
n	Algorithm	Error	Time	Rank
5	BLOSTR	2.22e−11	5.73e−3	2.00
	ALS	3.64e−12	1.34e−1	2.00
	ALS-sampled	5.17e−12	1.00e−1	2.00
	TR-SVD	2.10e−15	1.23e−2	5.60
	TT-SVD+TT2TR	3.02e−15	4.12e−2	2.00
	TT-RSS+TT2TR	2.22e−15	1.16e−1	2.00
	TR-RSS	4.55e−15	9.19e−2	2.00
7	BLOSTR	1.65e−7	1.26e−2	2.00
	ALS-sampled	4.21e−12	1.64e1	2.00
	TR-SVD	4.86e−15	1.03	6.29
	TT-SVD+TT2TR	9.36e−15	3.14e−1	2.00
	TT-RSS+TT2TR	2.56e−15	2.41e−1	2.00
	TR-RSS	5.37e−15	2.58e−1	2.00
10	TT-RSS+TT2TR	1.78e−15	4.48e−1	2.00
	TR-RSS	3.02e−15	4.96e−1	2.00
50	TT-RSS+TT2TR	2.17e−15	4.41e−1	2.00
	TR-RSS	1.97e−15	5.50e−1	2.00
100	TT-RSS+TT2TR	7.99e−15	1.46	2.00
	TR-RSS	7.82e−15	1.73	2.00

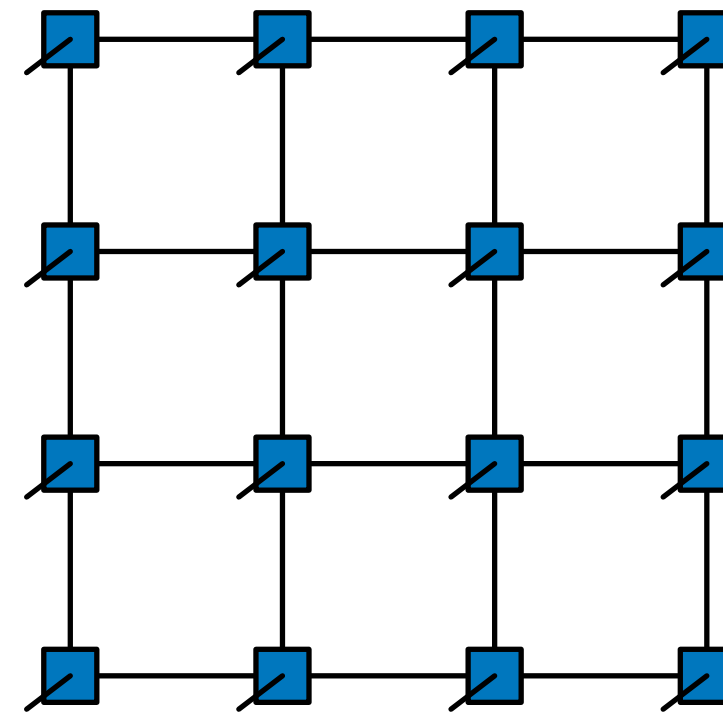
Exact TR + noise				
n	Algorithm	Error	Time	Rank
5	BLOSTR	6.53e3	3.14e−2	2.00
	ALS	2.16e−2	1.66	2.00
	TR-SVD	7.62e−1	1.27e−2	2.00
	TT-RSS+TT2TR	2.18e−2	2.10e−1	2.00
	TR-RSS	2.54e−2	9.55e−2	2.00
6	BLOSTR	1.53e3	6.74e−3	2.00
	ALS	2.33e−2	1.51e1	2.00
	TR-SVD	9.72e−1	1.08e−1	2.00
	TT-RSS+TT2TR	2.37e−2	3.53e−1	2.00
	TR-RSS	3.00e−2	2.57e−1	2.00

$$f(x) = (x + 1) \sin(100(x + 1)^2)$$

Algorithm	Error	Time	Rank
TR-SVD	1.55e−3	3.22e−2	6.11
ALS	1.85e−3	1.49e2	6.11
TT-SVD+TT2TR	5.29e−2	2.39e−1	5.76
TR-RSS	7.94e−3	5.01e−1	5.76

Next steps

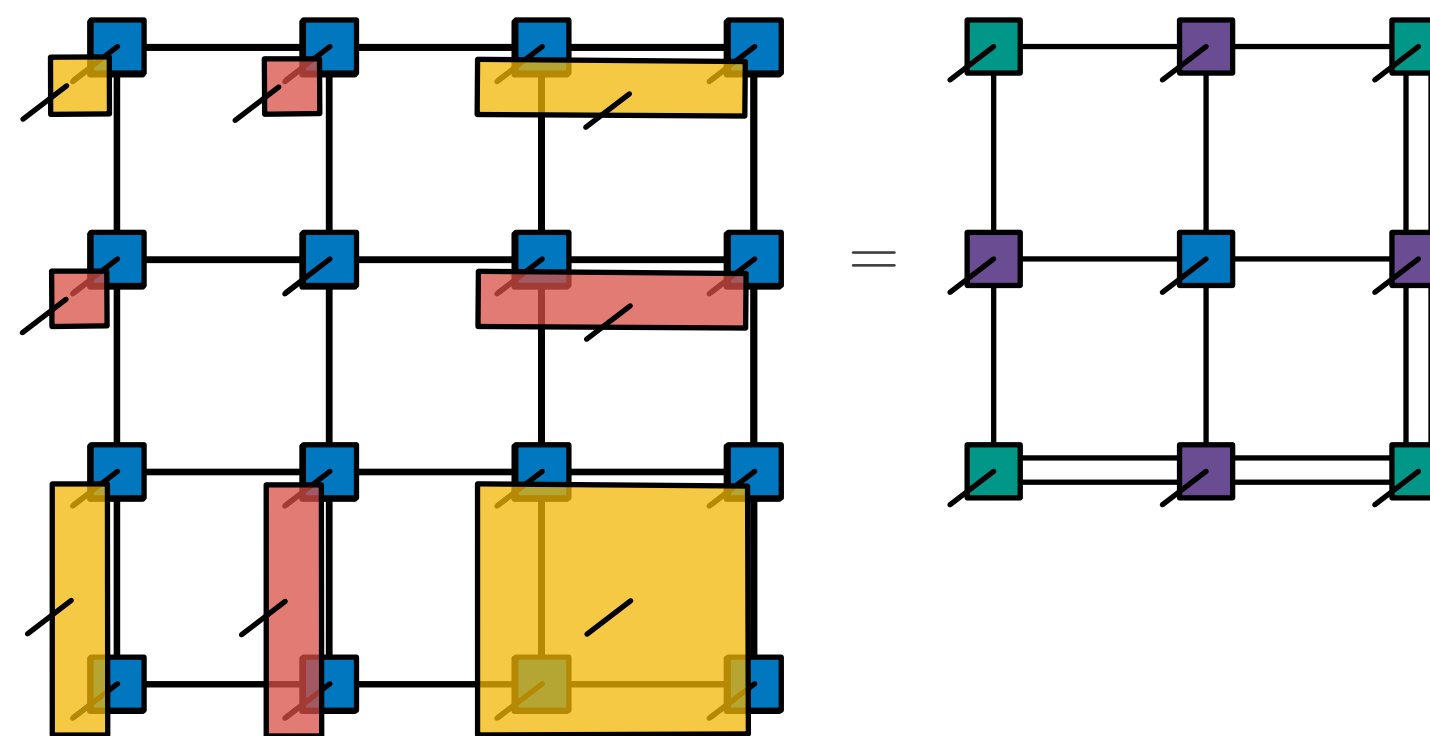
So... Is it possible to tensorize into **PEPS** form?



Next steps

So... Is it possible to tensorize into **PEPS** form?

PEPS-RSS



1 Open central *PEPS* to find **explicit environments**

2 **Build** next environments (*up, down, left, right*) **recursively** and **fix** in next step

Tensorization

Performance

Grid size	d	algorithm	Error	Time	N_{params}
2x2	2	PEPS-RSS	$9.19\text{e-}15$	$7.62\text{e-}2$	32
		PEPS-SVD-Hier	$5.66\text{e-}16$	$2.06\text{e-}4$	64
		PEPS-SVD-Seq	$2.74\text{e-}16$	$2.19\text{e-}4$	40
	3	PEPS-RSS	$1.01\text{e-}14$	$1.57\text{e-}1$	48
		PEPS-SVD-Hier	$8.84\text{e-}16$	$2.17\text{e-}4$	144
		PEPS-SVD-Seq	$1.06\text{e-}15$	$2.50\text{e-}4$	192
	4	PEPS-RSS	$6.73\text{e-}15$	$7.84\text{e-}2$	64
		PEPS-SVD-Hier	$1.35\text{e-}15$	$2.45\text{e-}4$	416
		PEPS-SVD-Seq	$8.12\text{e-}16$	$2.72\text{e-}4$	400
3x3	2	PEPS-RSS	$1.39\text{e-}14$	2.43	128
		PEPS-SVD-Hier	$1.96\text{e-}15$	$6.02\text{e-}4$	832
		PEPS-SVD-Seq	$2.71\text{e-}15$	$7.68\text{e-}4$	1412
	3	PEPS-RSS	$1.13\text{e-}14$	2.65	192
		PEPS-SVD-Hier	$3.74\text{e-}15$	$5.25\text{e-}3$	22923
		PEPS-SVD-Seq	$5.19\text{e-}15$	$1.63\text{e-}2$	53130
	4	PEPS-RSS	$7.38\text{e-}15$	5.35	256
		PEPS-SVD-Hier	$6.25\text{e-}15$	$9.78\text{e-}2$	279552
		PEPS-SVD-Seq	$1.67\text{e-}14$	$2.23\text{e-}1$	499408
4x4	2	PEPS-RSS	$2.01\text{e-}14$	$4.95\text{e}1$	288
		PEPS-SVD-Hier	$6.89\text{e-}15$	$6.07\text{e-}2$	187085
		PEPS-SVD-Seq	$9.77\text{e-}15$	$7.83\text{e-}2$	150992
	3	PEPS-RSS	$2.98\text{e-}14$	$1.72\text{e}2$	432
		PEPS-SVD-Hier	$9.57\text{e-}14$	$4.29\text{e}2$	167759683
		PEPS-SVD-Seq	$3.12\text{e-}13$	$4.34\text{e}2$	131813076
5x5	2	PEPS-RSS	$2.03\text{e-}13$	$3.49\text{e}3$	512
		PEPS-SVD-Hier	$3.97\text{e-}14$	$6.61\text{e}1$	37601744
		PEPS-SVD-Seq	$1.62\text{e-}12$	$2.59\text{e}2$	80275932

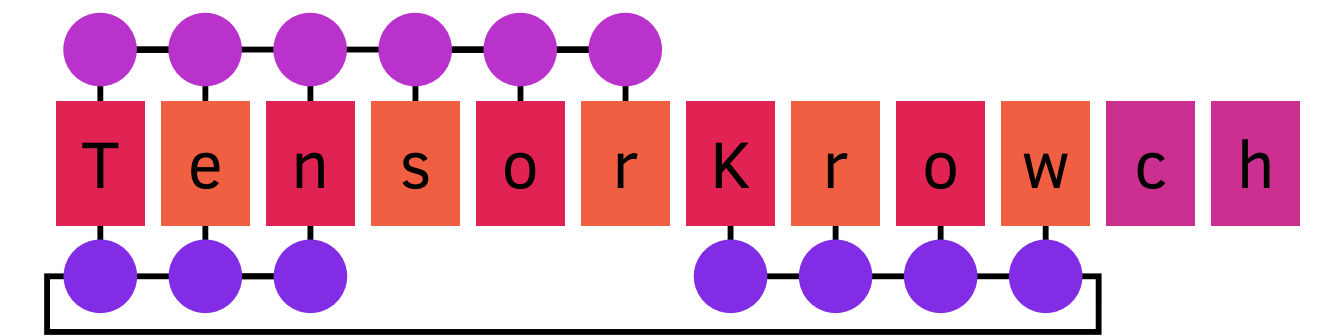
Next steps

JRPM, D. Pérez-García, *arXiv:2607.????*

Summary and outlook

- **Access to samples enables efficient tensorization**, which applies to many interesting scenarios, including ML models, NQS, and more general functions.
- **TN representations provide access to interesting information**, giving better insight into what complex models such as NNs learn.
- **Understanding what is happening under the hood is key** to designing future tensorization algorithms.

Any ideas to stabilize/speed up PEPS decomposition?



<https://github.com/joserapa98/tensorkrowch>

Thank you!

José R. Pareja Monturiol

<https://github.com/joserapa98>

@joserapa98



GOBIERNO
DE ESPAÑA

MINISTERIO
DE CIENCIA, INNOVACIÓN
Y UNIVERSIDADES



UNIVERSIDAD
COMPLUTENSE
MADRID



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID



ESCUELA TÉCNICA
SUPERIOR DE INGENIEROS
INFORMÁTICOS