How to Square Tensor Networks and Circuits Without Squaring Them

Lorenzo Loconte
`loreloc.github.io`

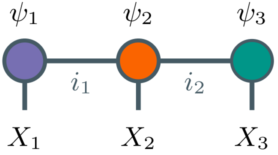
Adrián Javaloy **Antonio Vergari**

University of Edinburgh, UK

ICLR 2026

Tensor networks as probabilistic models

$$\psi(X_1, X_2, X_3) = \sum_{r_1=1}^{R} \sum_{r_2=1}^{R} \psi_1[X_1, r_1]\psi_2[r_1, X_2, r_2]\psi_3[r_2, X_3]$$

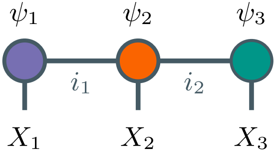


Orús, “A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States”, 2013

Bonnevie and Schmidt, “Matrix Product States for Inference in Discrete Probabilistic Models”, 2021

Tensor networks as probabilistic models

$$\psi(X_1, X_2, X_3) = \sum_{r_1=1}^{R} \sum_{r_2=1}^{R} \psi_1[X_1, r_1]\psi_2[r_1, X_2, r_2]\psi_3[r_2, X_3]$$

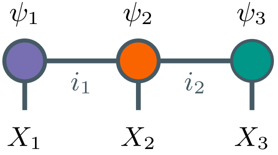


$$p(X_1, X_2, X_3) \propto |\psi(X_1, X_2, X_3)|^2 \qquad \text{(Born rule)}$$

Orús, "A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States", 2013

Bonnevie and Schmidt, "Matrix Product States for Inference in Discrete Probabilistic Models", 2021

Tensor networks as probabilistic models



$$\psi(X_1, X_2, X_3) = \sum_{r_1=1}^{R} \sum_{r_2=1}^{R} \psi_1[X_1, r_1]\psi_2[r_1, X_2, r_2]\psi_3[r_2, X_3]$$

$p(X_1, X_2, X_3) \propto |\psi(X_1, X_2, X_3)|^2$ (Born rule)

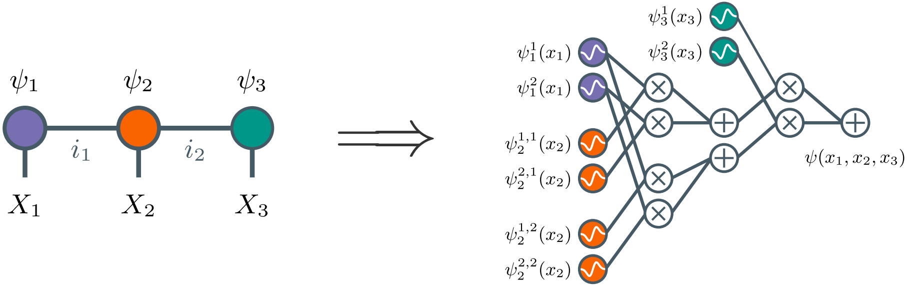
Canonical forms: e.g., $\psi_1 \psi_1^\dagger = \mathbf{I}$, $\psi_2[:,:,r_2]\psi_2[:,:,r_2]^\dagger = \mathbf{I}$

Computing $p(X_3)$ requires time $\mathcal{O}(R)$ instead of $\mathcal{O}(R^2)$

Orús, “A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States”, 2013

Bonnevie and Schmidt, “Matrix Product States for Inference in Discrete Probabilistic Models”, 2021

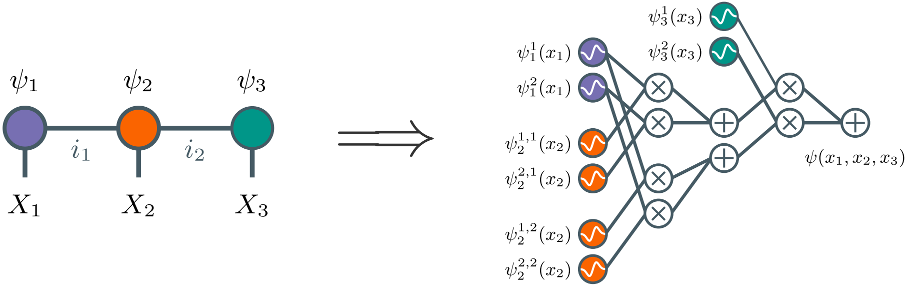
Tensor networks are circuits



Loconte et al., "Subtractive Mixture Models via Squaring: Representation and Learning", 2024

Loconte et al., "What is the Relationship between Tensor Factorizations and Circuits (and How Can We Exploit it)?", 2025

Tensor networks are circuits

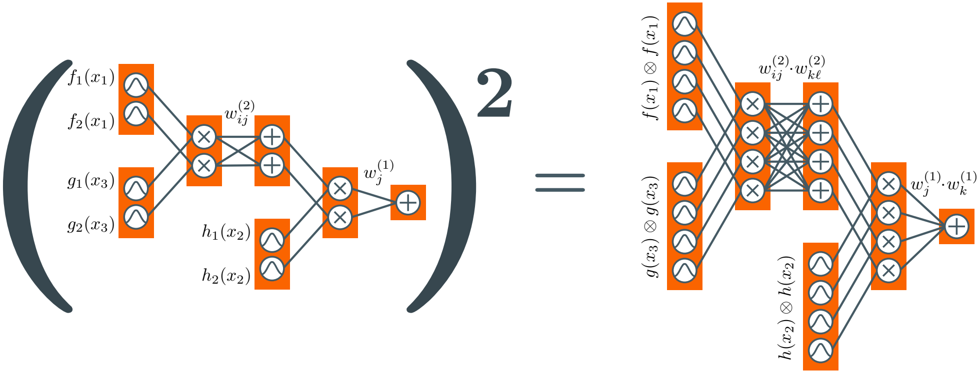


Circuits can be squared as well to recover $p(\mathbf{X})$ *...*

Loconte et al., "Subtractive Mixture Models via Squaring: Representation and Learning", 2024

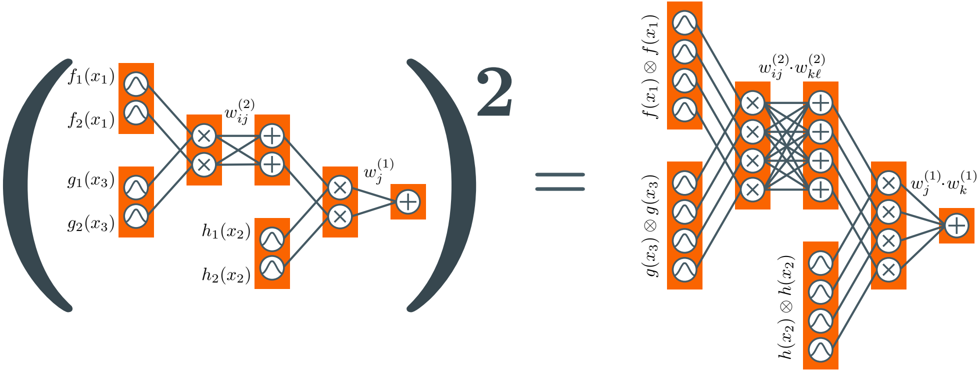
Loconte et al., "What is the Relationship between Tensor Factorizations and Circuits (and How Can We Exploit it)?", 2025

Squared circuits grow in size



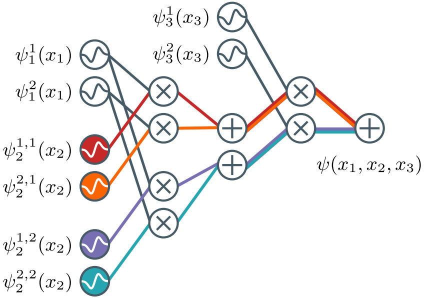
Computing marginals requires time $\mathcal{O}(|c|^2)$

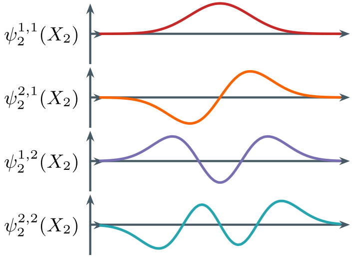
Squared circuits grow in size



How to generalize canonical forms to circuits?

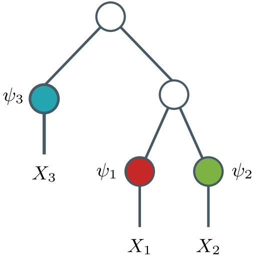
Canonical forms as unit-wise orthonormality





From $\mathcal{O}(|\mathsf{c}|^2)$ marginalization complexity to $\mathcal{O}(|\mathsf{c}|)$

Unlocking a strictly larger set of factorizations

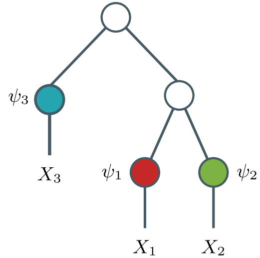


From a single hierarchical variable partitioning…

$$\mathbf{X} = (\boldsymbol{X_3},\ (\boldsymbol{X_1}, \boldsymbol{X_2}))$$

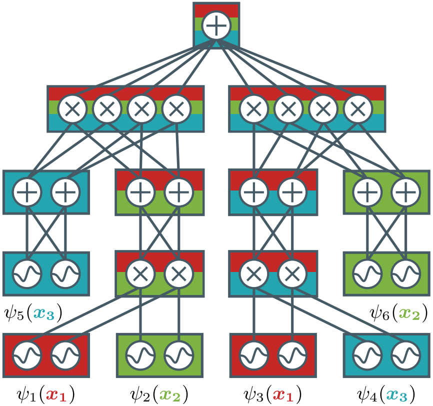
Pipatsrisawat and Darwiche, "New Compilation Languages Based on Structured Decomposability.", 2008

Unlocking a strictly larger set of factorizations



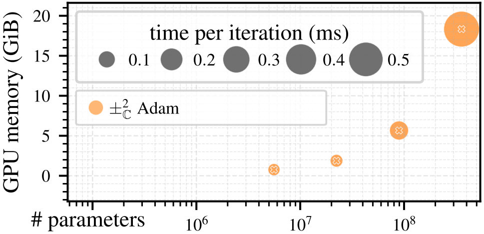
...to multiple hierarchical variable partitionings

$$\mathbf{X} = (\boldsymbol{X}_3,\ (\boldsymbol{X}_1, \boldsymbol{X}_2))$$
$$\mathbf{X} = ((\boldsymbol{X}_1, \boldsymbol{X}_3),\ \boldsymbol{X}_2)$$



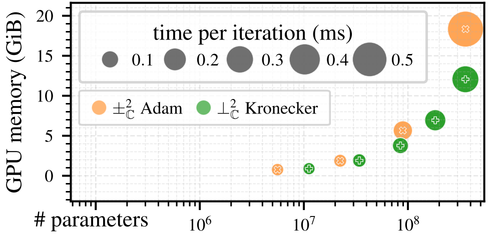
Pipatsrisawat and Darwiche, "New Compilation Languages Based on Structured Decomposability.", 2008

Efficient learning over the Stiefel manifold



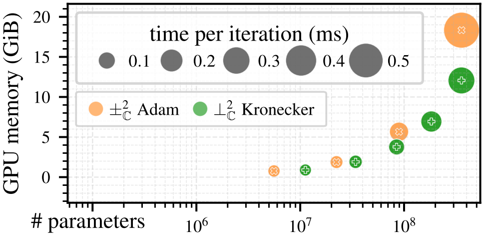
Computing $Z = \sum_{\mathbf{x}} |c(\mathbf{x})|^2$ explicitly at training time is expensive ...

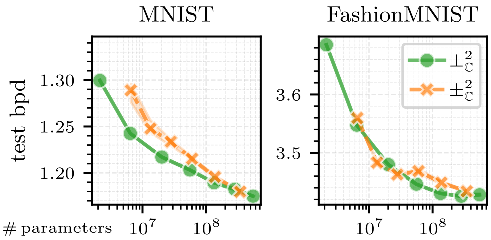
Efficient learning over the Stiefel manifold



...but we ensure $Z = 1$ during optimization!

Efficient learning over the Stiefel manifold





…which comes with no expressiveness loss

Takeaways

1. Structural properties inspired by canonical forms to speed-up marginalization in circuits

Poster ID: 10008162

Sat, Apr 25, 2026 • 3:15 PM – 5:45 PM

Paper ***Code***

Takeaways

1. Structural properties inspired by canonical forms to speed-up marginalization in circuits

2. Unlock efficient marginalization in squared factorizations that would not otherwise support it

Poster ID: 10008162

Sat, Apr 25, 2026 • 3:15 PM – 5:45 PM

Paper ***Code***

Takeaways

1. Structural properties inspired by canonical forms to speed-up marginalization in circuits

2. Unlock efficient marginalization in squared factorizations that would not otherwise support it

3. Our experiments show more efficient learning on the Stiefel manifold, with no expressiveness loss

Poster ID: 10008162

Sat, Apr 25, 2026 • 3:15 PM – 5:45 PM

Paper ***Code***