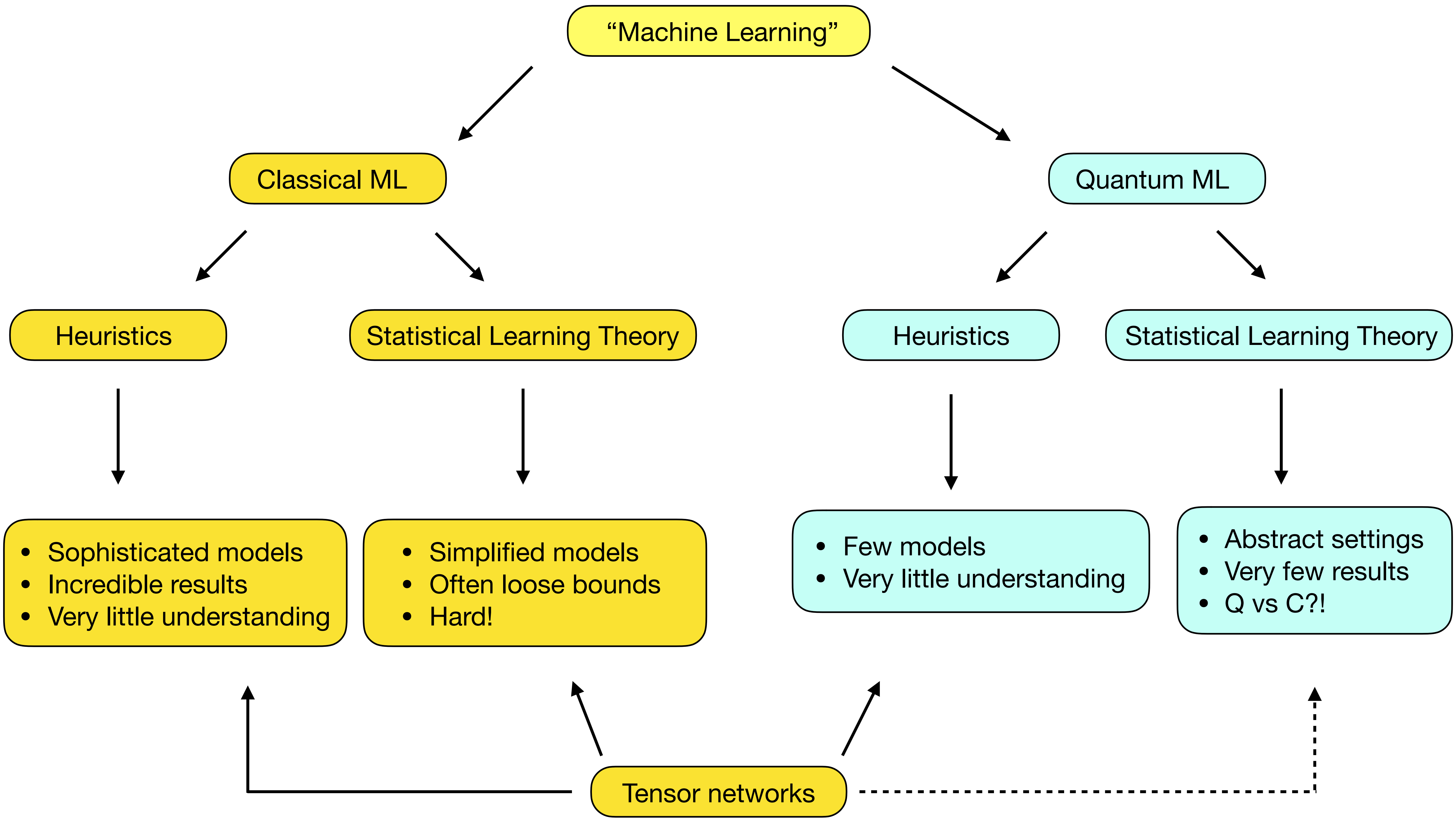
# Probabilistic Modelling with Tensor Networks:

## *From Hidden Markov Models to Quantum Circuits*

**Ryan Sweke**

**Freie Universität Berlin**

# The Big Picture




TN's provide a nice language to bridge heuristics with theory, and quantum with classical!

# What is this talk about?

This talk is about *Probabilistic Modelling…*

**Given:** Samples $\{\vec{d}_1, \ldots, \vec{d}_M\}$ from an unknown discrete multivariate probability distribution $P(X_1, \ldots, X_N)$.

$\vec{d}_j = (X_1^j, \ldots, X_N^j)$

$X_i \in \{1, \ldots, d\}$

**Task:** "Learn" a parameterized model $P(X_1, \ldots, X_N | \vec{\theta})$.

This may mean many different things, depending on the task you are interested in…

→ Performing inference (i.e. calculating marginals).

→ Calculating expectation values.

→ Generating samples.

Depending on your goal, your model/approach may differ significantly!

# Probabilistic Modelling

I like to think of there being three distinct elements:

(1) The model $P(X_1, \ldots, X_N | \vec{\theta})$. → Key Question: Expressivity?

(2) The learning algorithm: $\{\vec{d}_1, \ldots, \vec{d}_M\} \rightarrow \vec{\theta}$ → Model Dependent!

→ Typically by maximising the (log) likelihood: $\mathscr{L} = \sum_i \log[P(\vec{d}_i | \vec{\theta})]$

(3) The "task" algorithm. → Model Dependent!

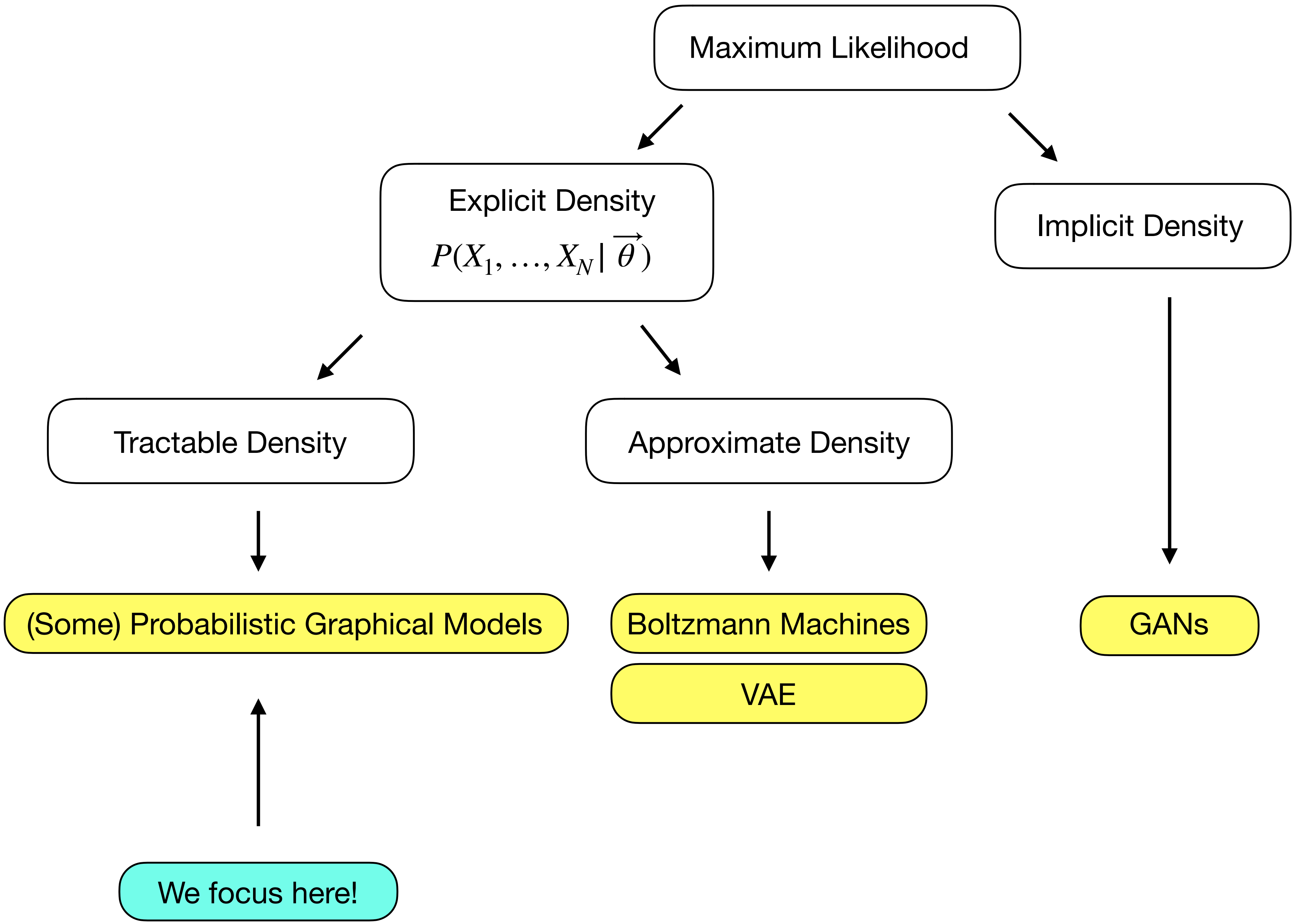
→ Performing inference via belief propagation for Probabilistic Graphical Models.

→ Expectation values via sampling for Boltzmann Machines.

→ Generating samples directly via a GAN.

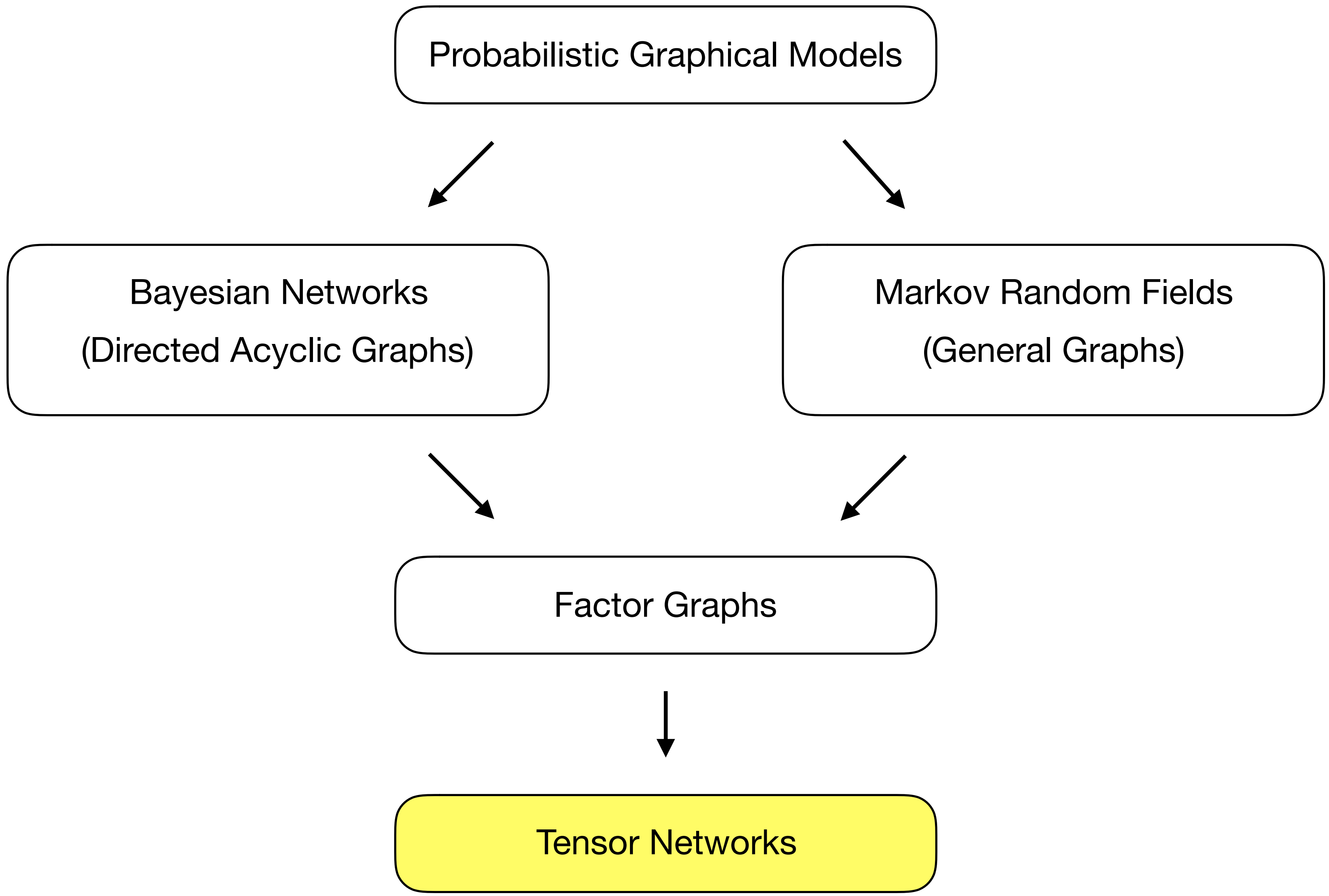
# Probabilistic Modelling

This overall picture is summarised quite nicely by the following "hierarchy of generative models":



# Probabilistic Graphical Models

We will see that tensor networks provide a unifying framework for analyzing probabilistic graphical models:

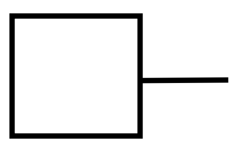


# Tensor Networks

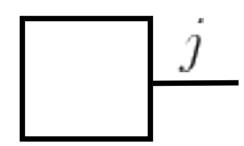
Tensor network notation provides a powerful and convenient diagrammatic language for tensor manipulation...

We represent tensors as boxes, with an "open leg" for each tensor index

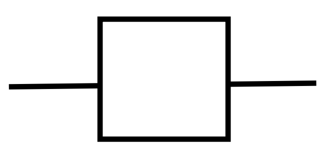
- A vector is a 1-tensor:

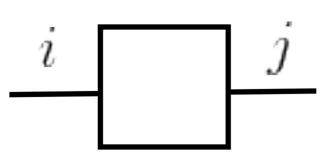
$\vec{v} \in \mathbb{C}^d =$ 

An element of the vector is a scalar ("close" the index)

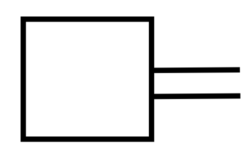
$\vec{v}_j =$ 

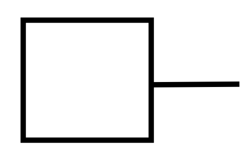

- A matrix is a 2-tensor:

$M \in \mathbb{C}^{d \times d} =$ 

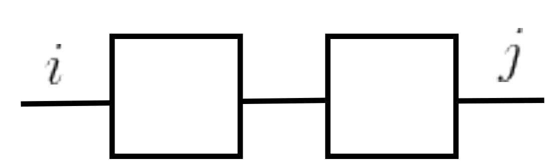
$M_{i,j} =$ 


"vectorization" is very natural in this notation...

$M \in \mathbb{C}^{d \times d} =$ 

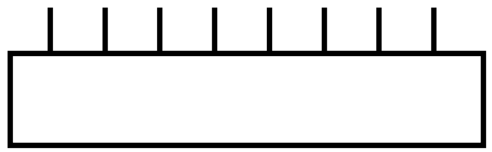
$=$  $\in \mathbb{C}^{d^2}$

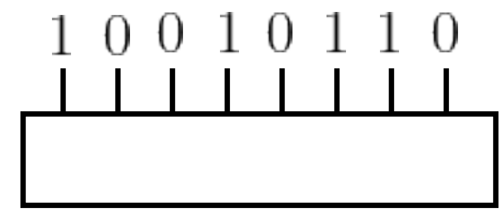
- A shared index denotes a contraction over that index:

$A_{i,j} = (BC)_{i,j} = \sum_k B_{i,k} C_{k,j} =$ 

# Tensor Networks

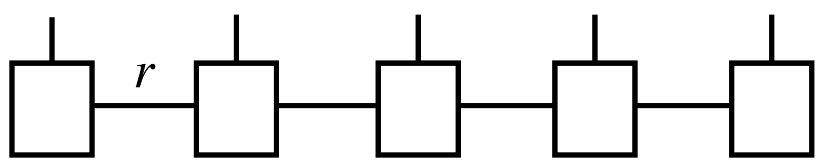
A discrete multivariate probability distribution is naturally represented as an N-tensor...

$P(X_1, \ldots, X_N) =$ 

$P(1,0,0,1,0,1,1,0) =$ 


$d^N$ parameters!

A *tensor network* decomposition of P is a decomposition into a network of contracted tensors...

$P(X_1, \ldots, X_N) =$ 


Matrix Product State

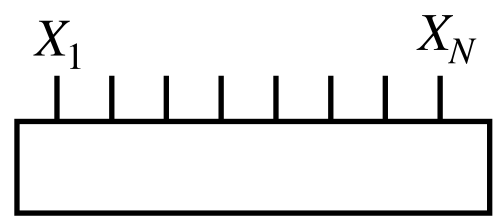
$Ndr^2$ parameters!

We call $r$ the *bond dimension* - directly related to the underlying correlation structure.

eg: $r = 1$ for independent (uncorrelated random variables).

These representations are very well understood in the context of many-body quantum physics.

# Probabilistic Graphical Models: Bayesian Networks

Given a probability distribution $P(X_1, \ldots, X_N) =$ 


A BN models this distribution via a directed acyclic graph expressing the structure of conditional dependencies.

For example: A Hidden Markov Model…

$$\begin{tikzcd} H_1 \arrow[r] & H_2 \arrow[r] \arrow[d] & H_3 \arrow[d] \\ X_1 \arrow[u] & X_2 & X_3 \end{tikzcd}$$

$< d^N$ parameters!

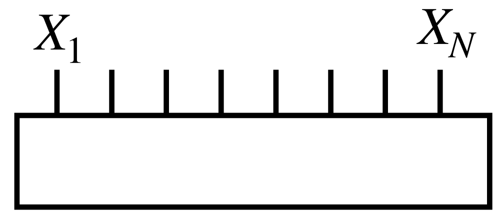
$$P(X_1, X_2, X_3, H_1, H_2, H_3) = P(X_1)P(H_1 | X_1)P(H_2 | H_1)P(X_2 | H_2)P(H_3 | H_2)P(X_3 | H_3)$$

The probability of “visible” variables is via marginalisation:

$$P(X_1, X_2, X_3) = \sum_{H_1, H_2, H_3} P(X_1, X_2, X_3, H_1, H_2, H_3)$$

# Probabilistic Graphical Models: Markov Random Fields

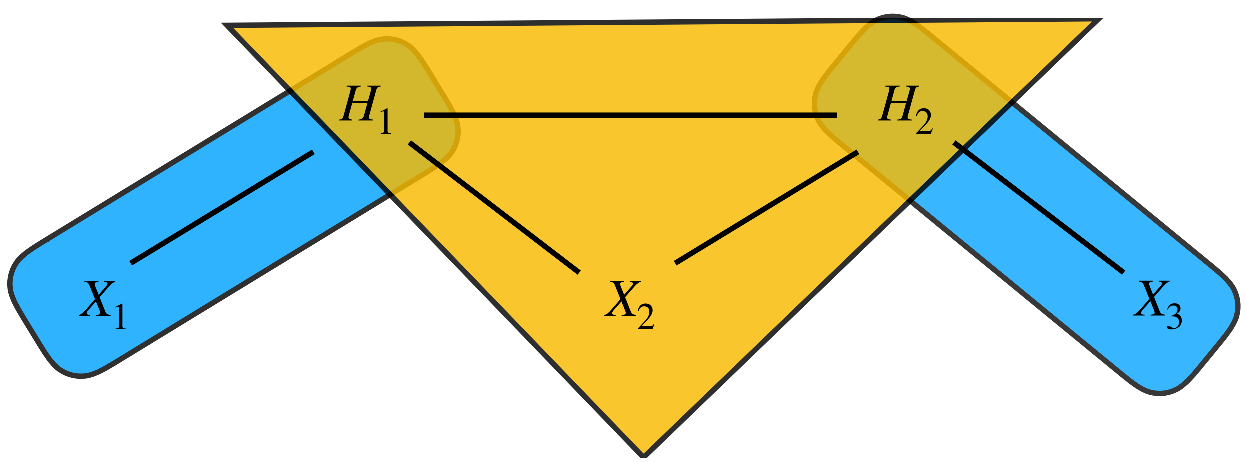
Given a probability distribution $P(X_1, \ldots, X_N) =$




A Markov Random Field models the distribution via the product of *clique potentials* defined by a generic graph.

maximal fully-connected subgraph

For example:




$$P(X_1, X_2, X_3, H_1, H_2, H_3) = g_1(X_1, H_1)\, g_2(H_1, X_2, H_2)\, g_3(H_2, X_3)\, \frac{1}{Z}$$

NB - Clique potentials are *not* normalised - explicit normalisation is necessary!

# Probabilistic Graphical Models: Factor Graphs

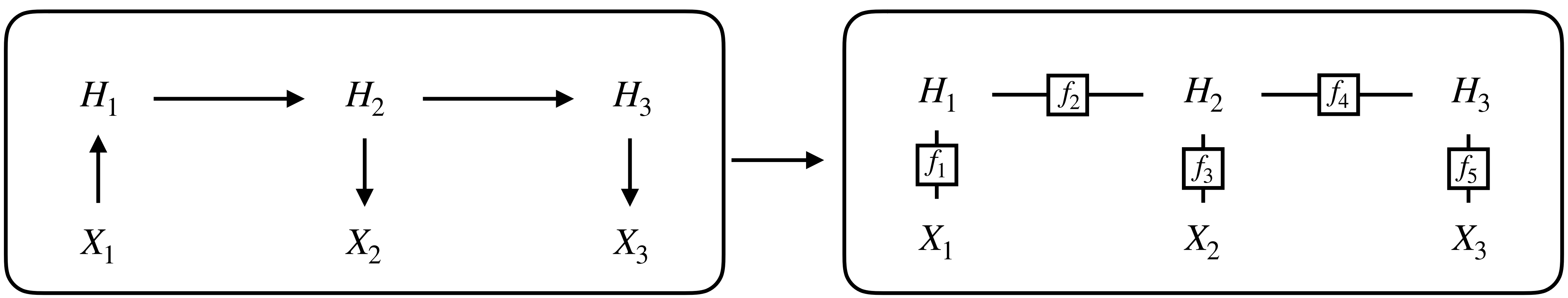
Bayesian Networks and Markov Random Fields are unified via *Factor Graphs…*

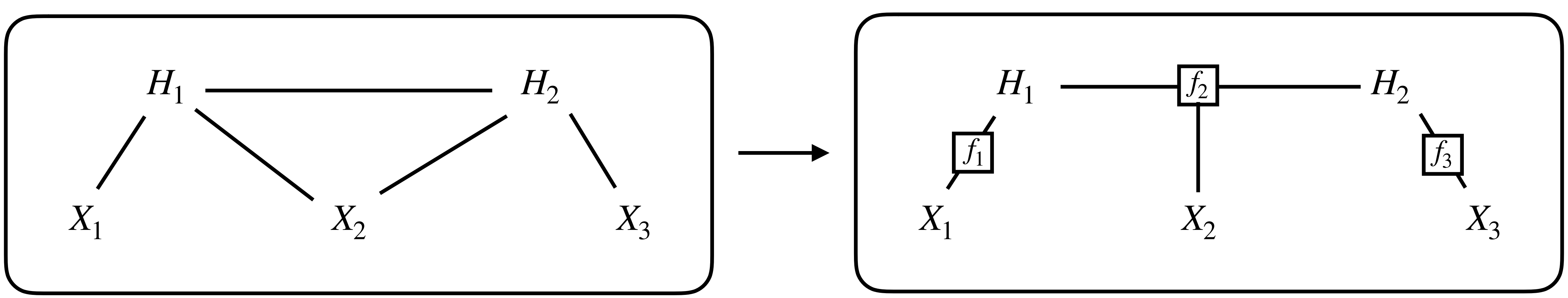
$$P(X_1, \ldots, X_N) = \frac{1}{Z}\prod_j f_j(\overrightarrow{X}_j)$$

→ Bayesian Networks: Factors are conditional probability distributions (inherently normalised)

→ Markov Random Fields: Factors are clique potentials (explicit normalisation necessary)

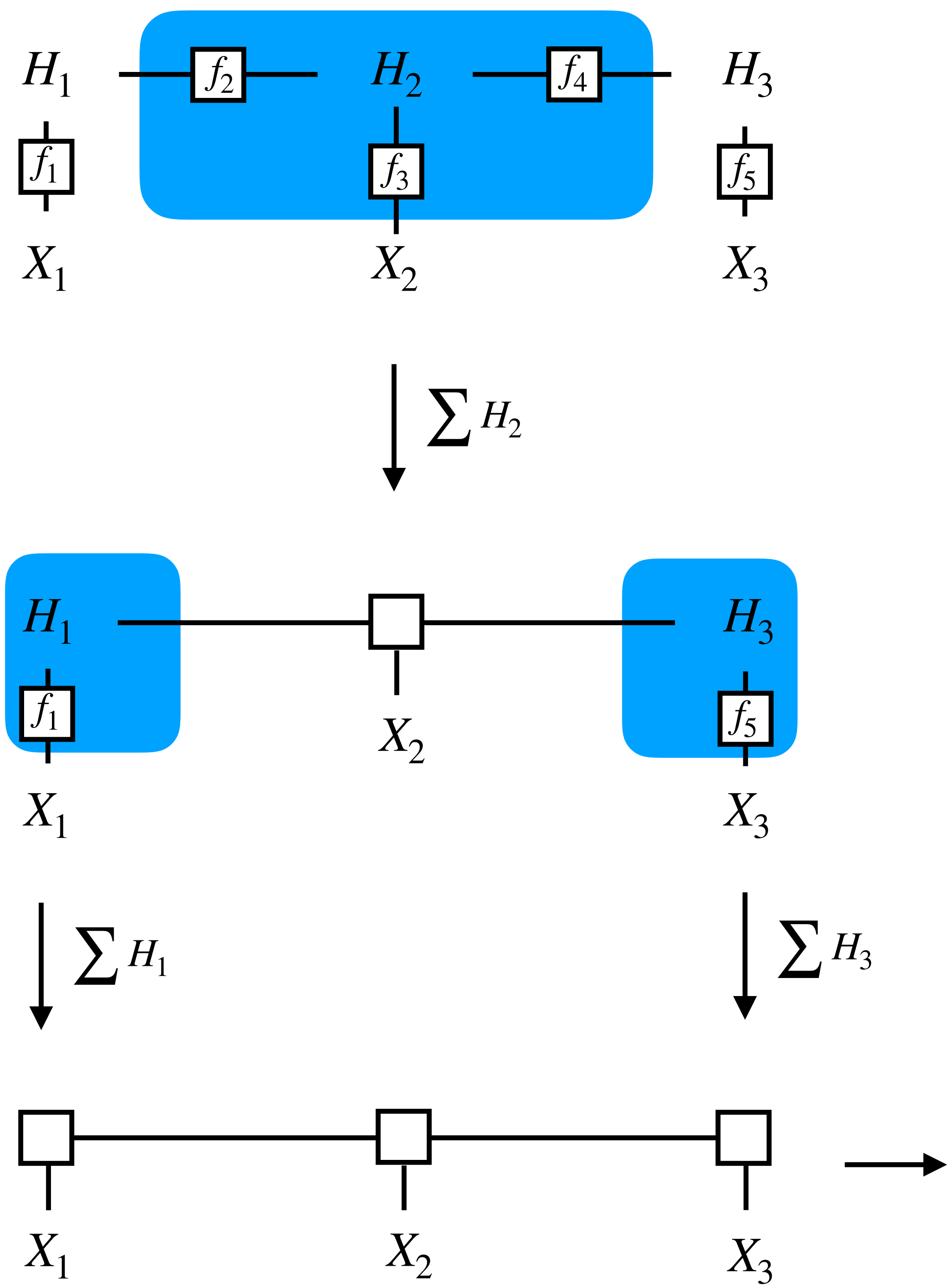
Explicitly:



# Probabilistic Graphical Models: Factor Graphs to Tensor Networks

Let's consider the Hidden Markov Model in more detail…



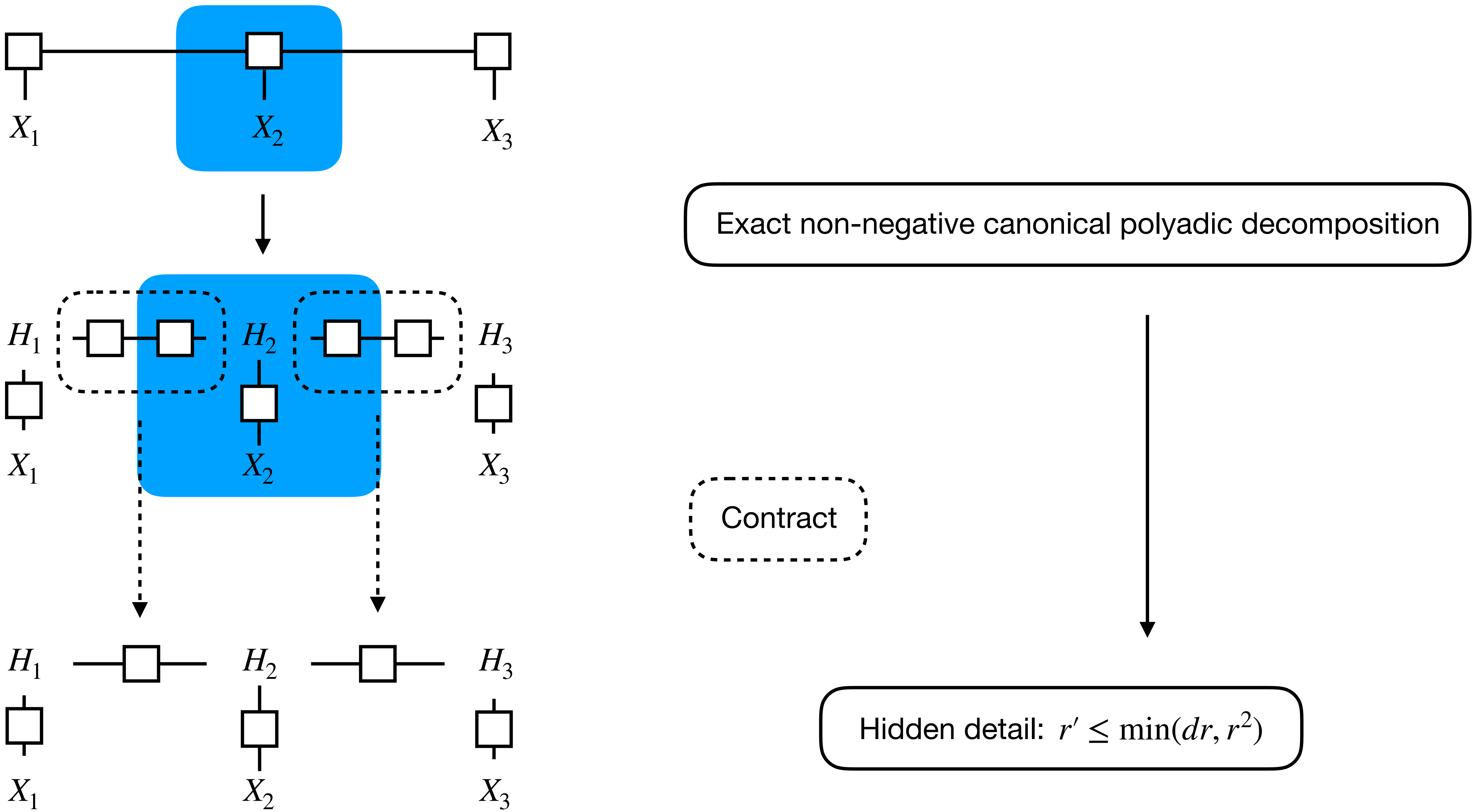

Marginalizing out the hidden variables means contracting the connected factor tensors!

The probability distribution over the visible variables is exactly equivalent to an MPS decomposition of the global probability tensor!

With non-negative tensors!

# Probabilistic Graphical Models: Factor Graphs to Tensor Networks

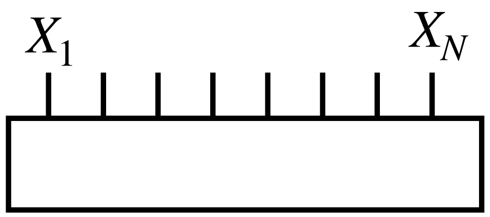
The other direction also holds…




Hidden Markov Models and non-negative MPS are *almost* exactly equivalent

# Probabilistic Graphical Models: Factor Graphs to Tensor Networks

Take home message - we can use Tensor Networks to study and to generalise probabilistic graphical models!




= Any tensor network which yields a non-negative tensor when contracted!

↓

Includes *all* probabilistic graphical models

See I. Glasser et al “Supervised Learning with generalised tensor networks” (Formal connection and heuristic algorithms)

Goal: By studying MPS based decompositions can we…

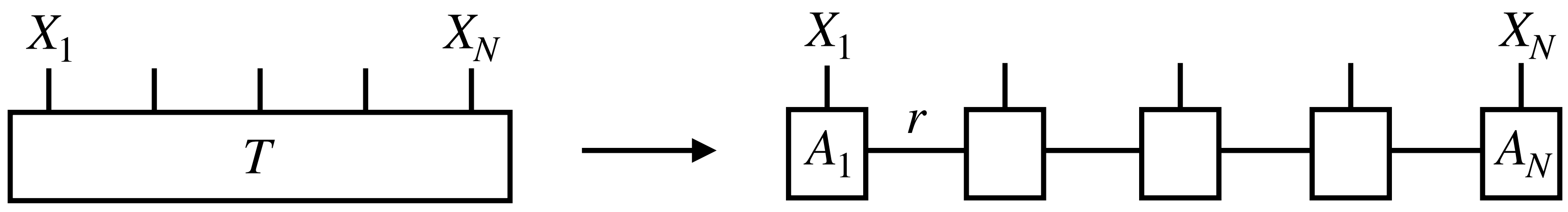
- → Make rigorous claims concerning expressivity?
- → Draw connections to quantum circuits?
- → Make claims concerning expressivity of classical vs quantum models?

↓

Yes.

## Tensor Network Models: HMM are MPS

The first model we consider is *non-negative* MPS - which we already showed are equivalent to HMM…




NB: All tensors have only non-negative (real) entries!

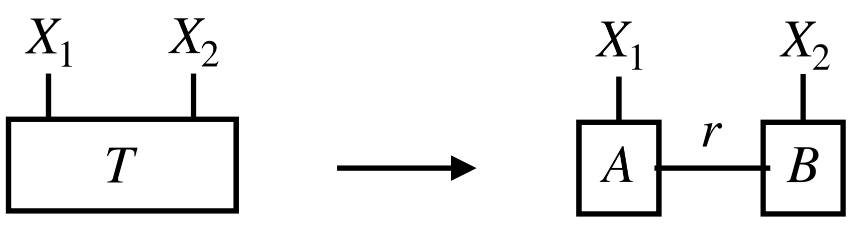
We call the minimal bond dimension $r$ necessary to factorise $T$ exactly the $\mathrm{TT} - \mathrm{Rank}_{\mathbb{R}_{\geq 0}}$.

‶Tensor-Train” rank

The bond-dimension necessary to represent a class of tensors characterises the *expressivity* of the model!

## Tensor Network Models: HMM are MPS

Note that for probability distributions over two variables (matrices) the $\mathrm{TT} - \mathrm{Rank}_{\mathbb{R}_{\geq 0}}$ is the non-negative rank:



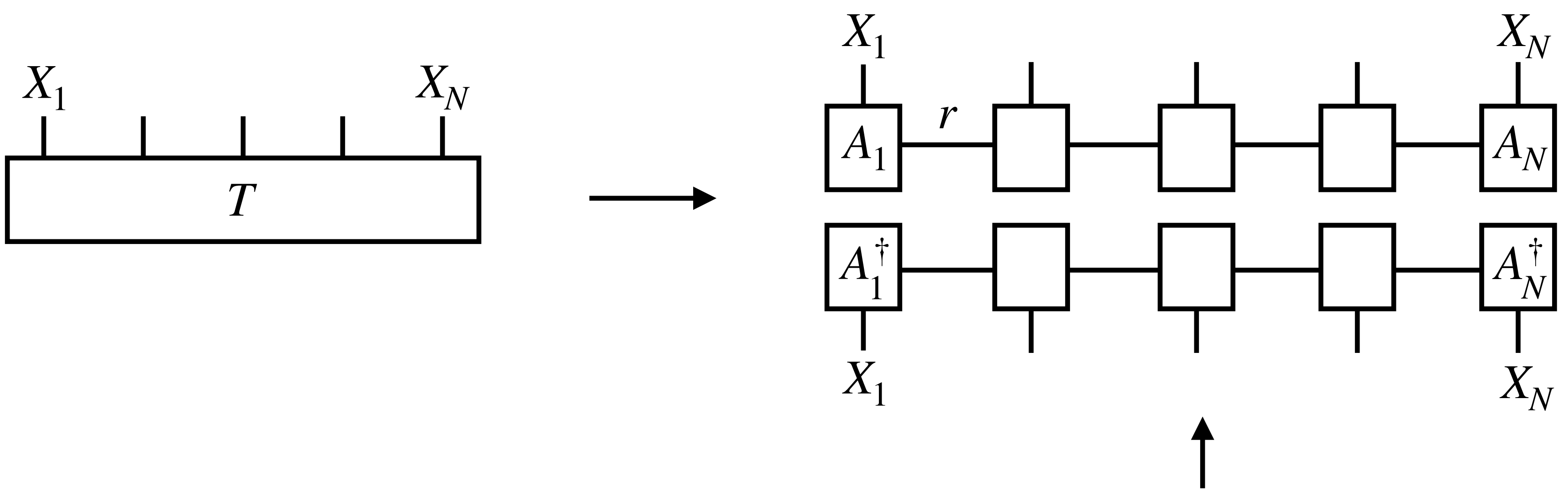

i.e. the smallest $r$ such that $T = AB$ with $A$ and $B$ non-negative.

Not such an easy rank to determine!

(NP-hard to determine whether rank is equal to non-negative rank.)

# Tensor Network Models: Born Machines

The second model we consider is *Born Machines*…

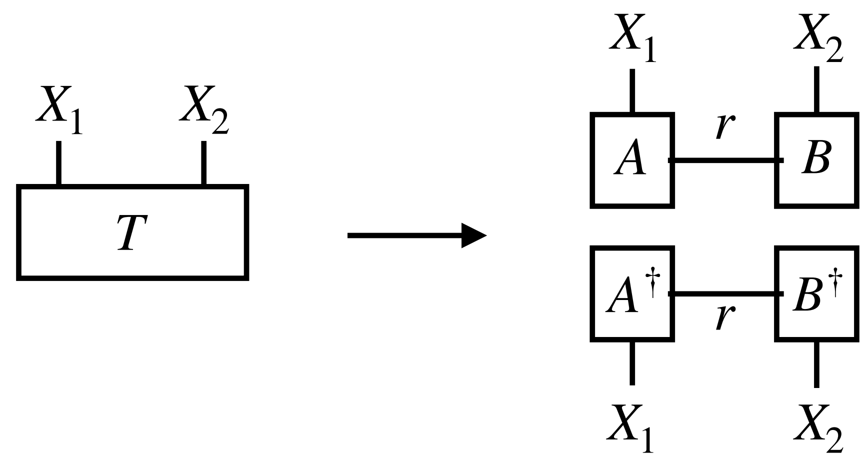



We can use either real or complex tensors!

We call the minimal bond dimension $r$ necessary to factorise $T$ exactly the $\mathrm{Born-Rank}_{\mathbb{R}/\mathbb{C}}$.

# Tensor Network Models: Born Machines

In the case of only two variables this is the real/complex Hadamard (entry-wise) square root rank…




$AB$ is an element wise square root!

i.e. the smallest $r$ such that $T = |AB|^{\circ 2}$

In the real case:

$$r = \min_{\pm}\left[\operatorname{rank}\begin{pmatrix} \pm\sqrt{t_{11}} & \dots & \pm\sqrt{t_{1d}} \\ \vdots & & \vdots \\ \pm\sqrt{t_{d1}} & \dots & \pm\sqrt{t_{dd}} \end{pmatrix}\right]$$

$2^{d^2}$ combinations - bad fast!

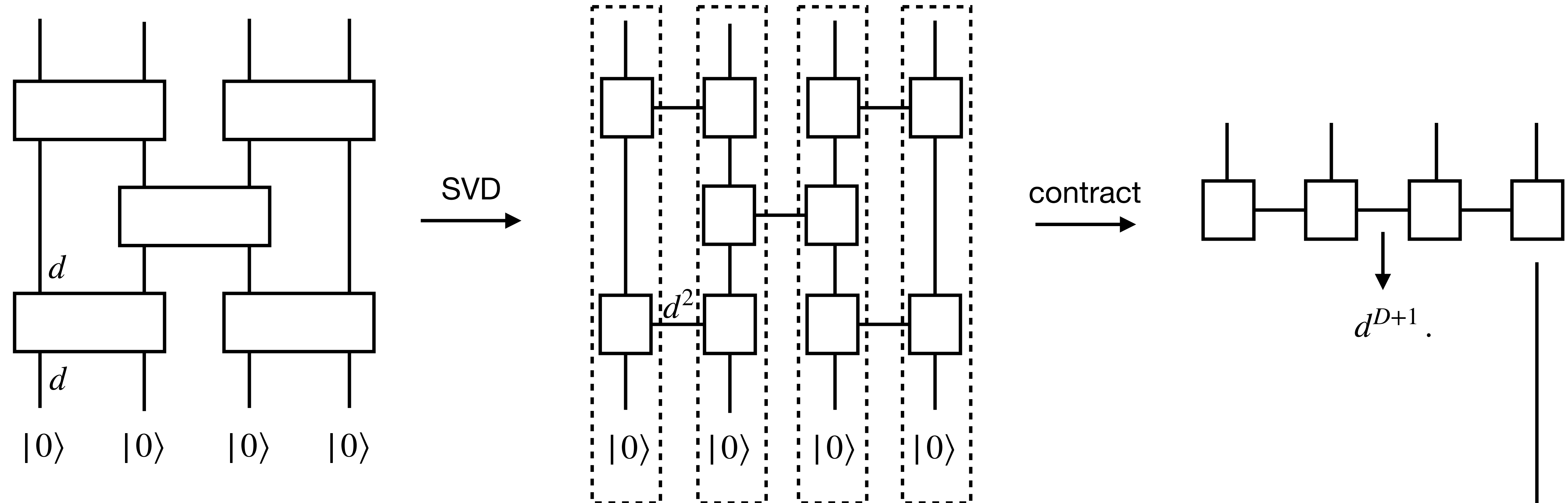
In the complex case:

$$r = \min_{\vec{\theta}}\left[\operatorname{rank}\begin{pmatrix} e^{i\theta_{11}}\sqrt{t_{11}} & \dots & e^{i\theta_{1d}}\sqrt{t_{1d}} \\ \vdots & & \vdots \\ e^{i\theta_{d1}}\sqrt{t_{d1}} & \dots & e^{i\theta_{dd}}\sqrt{t_{dd}} \end{pmatrix}\right]$$

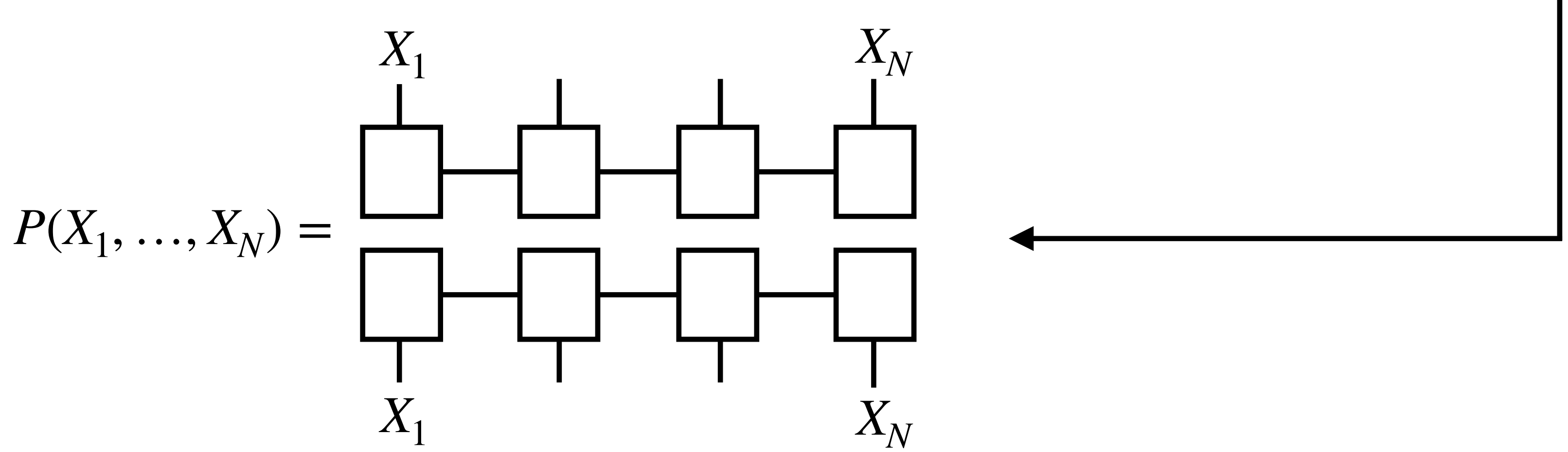
even worse :(

# Tensor Network Models: Born Machines

Outcome probabilities of a 2-local quantum circuit of depth $D$ are described exactly by a BM of bond dimension $d^{D+1}$.

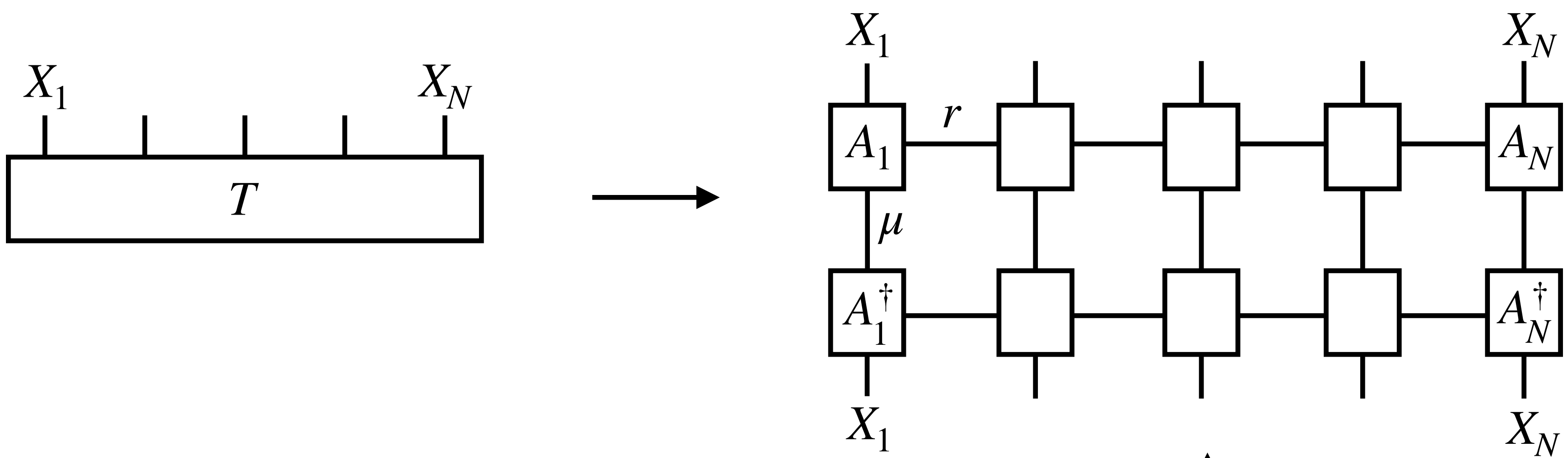



The probability of a measurement outcome is described by the BM defined via the circuit MPS:



# Tensor Network Models: Locally Purified States

The final model we consider is *Locally Purified States*…




We can use either real or complex tensors!

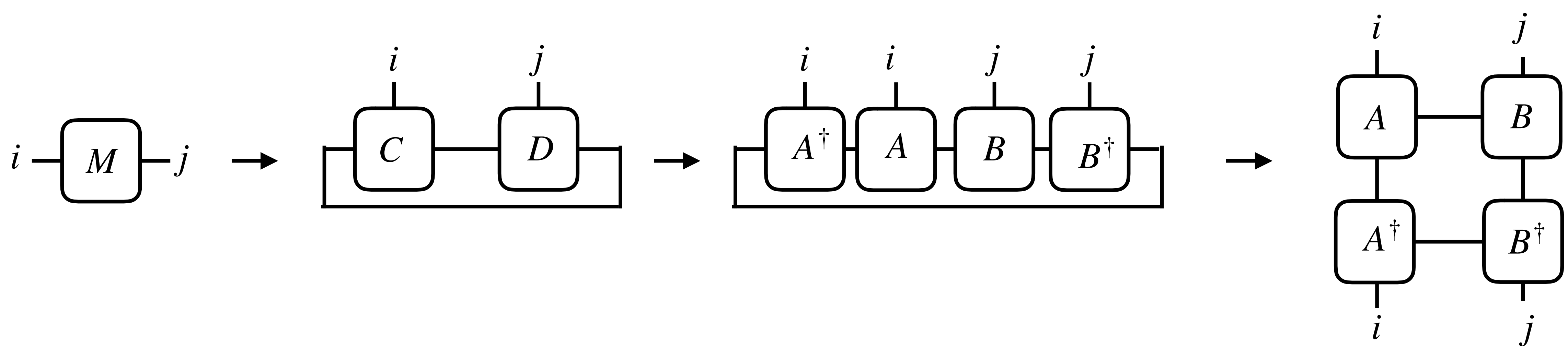
We call the minimal bond dimension $r$ necessary to factorise $T$ exactly the $\mathrm{Puri-Rank}_{\mathbb{R}/\mathbb{C}}$.

In the case of only two variables this is the positive-semidefinite rank

# Tensor Network Models: Locally Purified States

In the case of only two variables this is positive semidefinite rank…

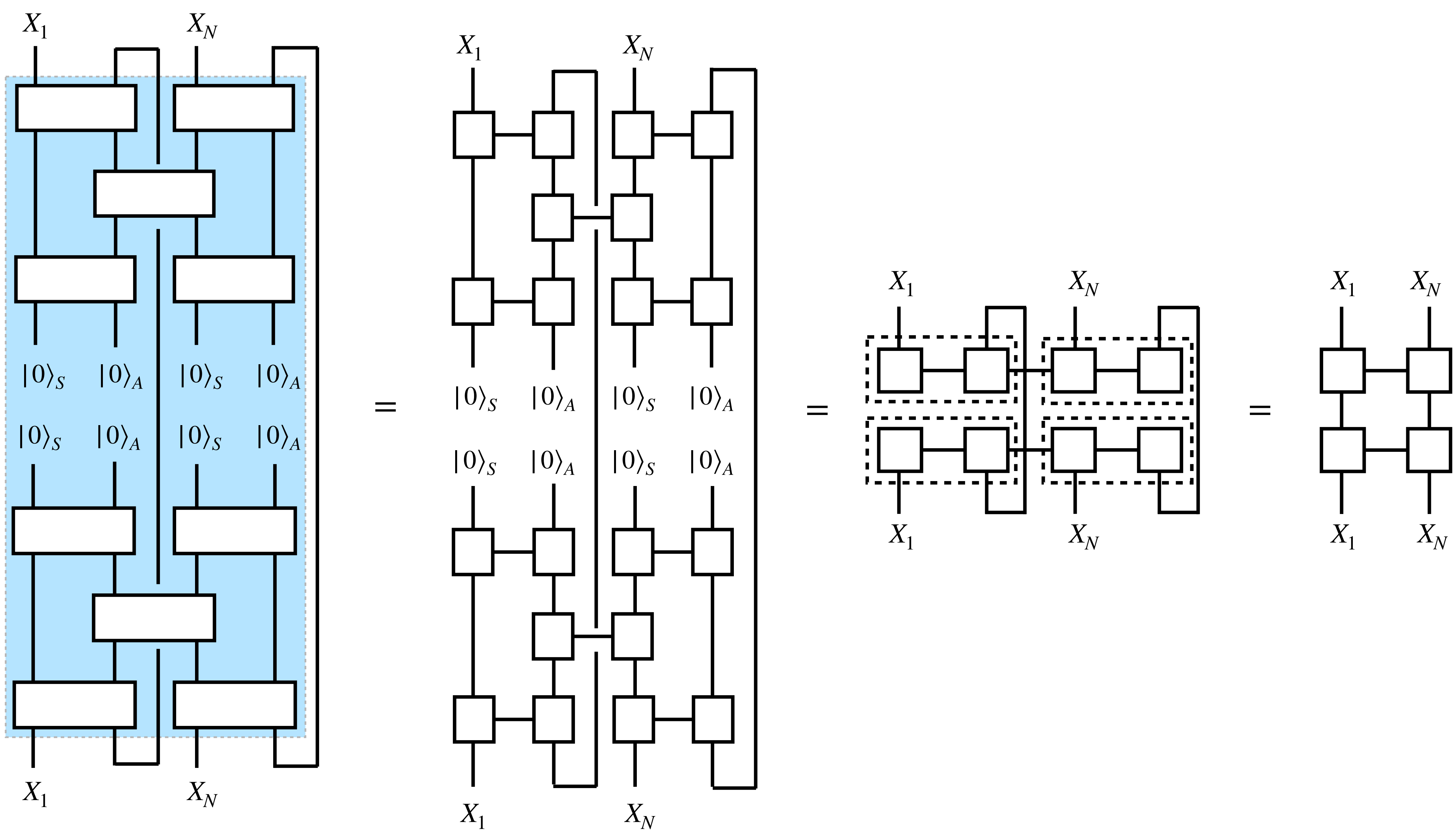
Given a matrix $M$, the PSD rank is the smallest $r$ for which there exist positive semidefinite matrices $A_i, B_j$ of size $r \times r$ such that $M = \mathrm{Tr}(A_i B_j)$.



# Tensor Network Models: Locally Purified States

LPS are equivalent to 2-local circuits with local ancillas…

$P(X_1, \ldots, X_N) =$




Crux: We can sample LPS by partial measurements of quantum circuits!

# Tensor Network Models Summary

Note that as *classical models*:

- → Learning is efficient - i.e. tractable likelihood and gradients.
- → Inference is efficient - marginalization is a simple efficient contraction.
- → Efficient sampling algorithms also exist (eg. ancestral sampling)

However, as quantum models (i.e. in an HQC setting):

- → Sampling is easy!
- → Learning is not straightforward - likelihood and gradients need to be estimated or bounded.

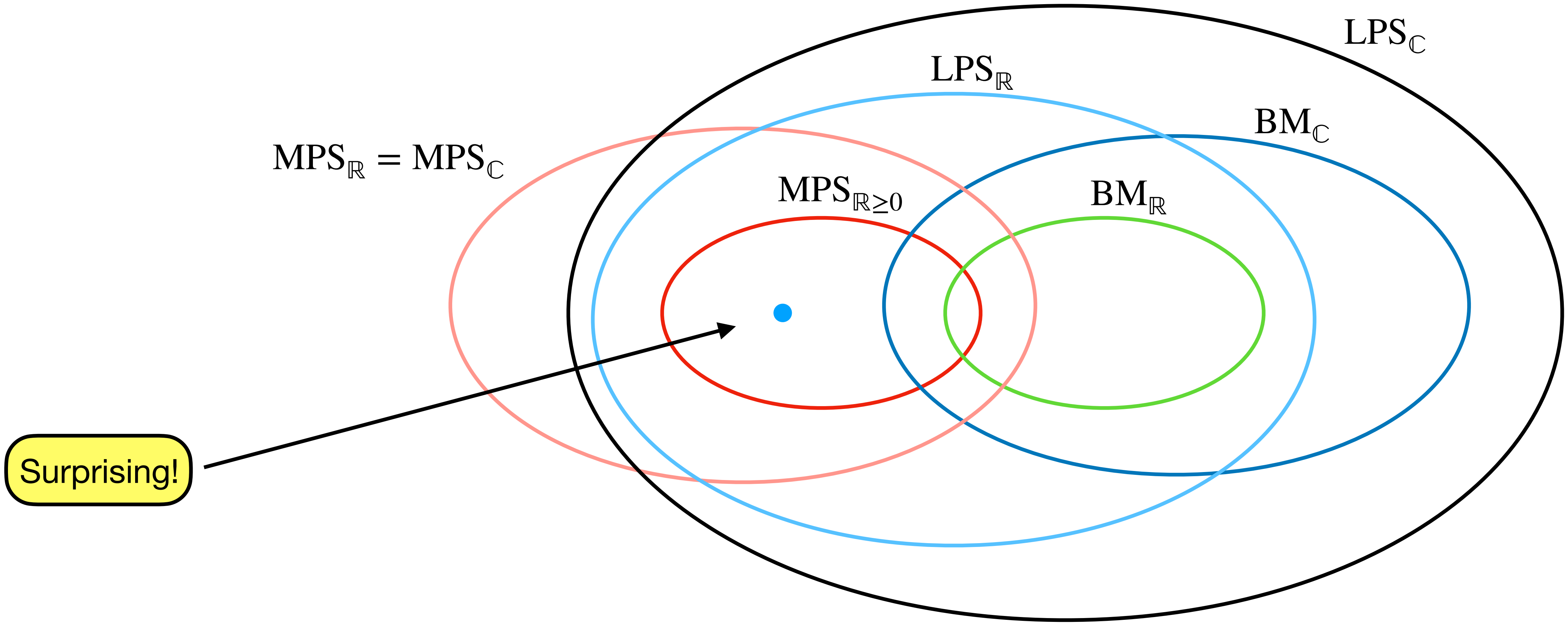
↓

But, exponential bond dimension of classical models requires only linear depth of quantum models!

Independent of the *learning* and *task* algorithms, we are interested in the relative expressivity!

## Expressivity Results

We first ask: For a fixed bond-dimension how are all the representations related?




Much more interesting though is the following question:

- → Given one representation of bond dimension $r$, (eg: $\mathrm{BM}_{\mathbb{C}}$)
- → what bond dimension $r'$ is necessary to write this tensor using another representation? (eg: $\mathrm{BM}_{\mathbb{R}}$)

We know that in the worst case $r' > r$, but by how much?

# Expressivity Results

We answer the question of relative overheads as follows:

| | TT-rank$_{\mathbb{R}}$ | TT-rank$_{\mathbb{R}_{\geq 0}}$ | Born-rank$_{\mathbb{R}}$ | Born-rank$_{\mathbb{C}}$ | puri-rank$_{\mathbb{R}}$ | puri-rank$_{\mathbb{C}}$ |
|---|---|---|---|---|---|---|
| TT-rank$_{\mathbb{R}}$ | = | $\leq x$ | $\leq x^2$ | $\leq x^2$ | $\leq x^2$ | $\leq x^2$ |
| TT-rank$_{\mathbb{R}_{\geq 0}}$ | No | = | No | No | No | No |
| Born-rank$_{\mathbb{R}}$ | No | No | = | No | No | No |
| Born-rank$_{\mathbb{C}}$ | No | No$^*$ | $\leq x$ | = | No$^*$ | No$^*$ |
| puri-rank$_{\mathbb{R}}$ | No | $\leq x$ | $\leq x$ | $\leq 2x$ | = | $\leq 2x$ |
| puri-rank$_{\mathbb{C}}$ | No | $\leq x$ | $\leq x$ | $\leq x$ | $\leq x$ | = |

We find two very distinct types of result:

1) Controlled overheads: eg $\text{puri} - \text{rank}_{\mathbb{R}} \leq 2(\text{Born} - \text{rank}_{\mathbb{C}})$

2) *Unbounded overheads*: eg

There exists a family of probability distributions over an increasing number of random variables N, with:

→ constant $\text{Born} - \text{rank}_{\mathbb{C}}$ .

→ $\text{Born} - \text{rank}_{\mathbb{R}}$ scales with N.

For Born machines complex numbers provide an *unbounded* amount of expressive power!

# Expressivity Results

Some other results to highlight:

| | TT-rank$_{\mathbb{R}}$ | TT-rank$_{\mathbb{R}_{\geq 0}}$ | Born-rank$_{\mathbb{R}}$ | Born-rank$_{\mathbb{C}}$ | puri-rank$_{\mathbb{R}}$ | puri-rank$_{\mathbb{C}}$ |
|---|---|---|---|---|---|---|
| TT-rank$_{\mathbb{R}}$ | = | $\leq x$ | $\leq x^2$ | $\leq x^2$ | $\leq x^2$ | $\leq x^2$ |
| TT-rank$_{\mathbb{R}_{\geq 0}}$ | No | = | No | No | No | No |
| Born-rank$_{\mathbb{R}}$ | No | No | = | No | No | No |
| Born-rank$_{\mathbb{C}}$ | No | No$^*$ | $\leq x$ | = | No$^*$ | No$^*$ |
| puri-rank$_{\mathbb{R}}$ | No | $\leq x$ | $\leq x$ | $\leq 2x$ | = | $\leq 2x$ |
| puri-rank$_{\mathbb{C}}$ | No | $\leq x$ | $\leq x$ | $\leq x$ | $\leq x$ | = |

1) Neither real Born Machines nor HMM should be preferred over the other!

2) Conjecture:

There exists a family of probability distributions which requires:

→ constant circuit depth with local ancillas.

→ unbounded circuit depth without ancillas!

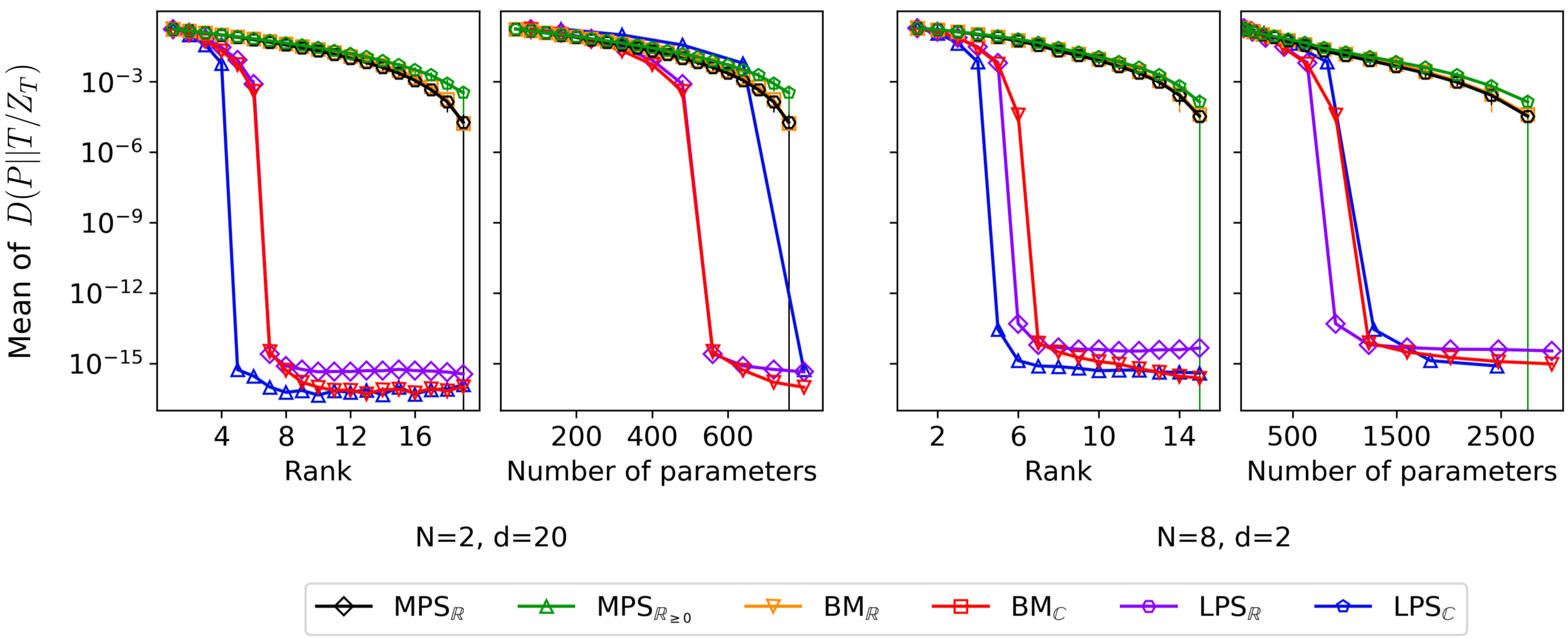
3) Locally purified states should always be preferred over all other models.

(and might exhibit unbounded expressive advantage!)

# Expressivity Results

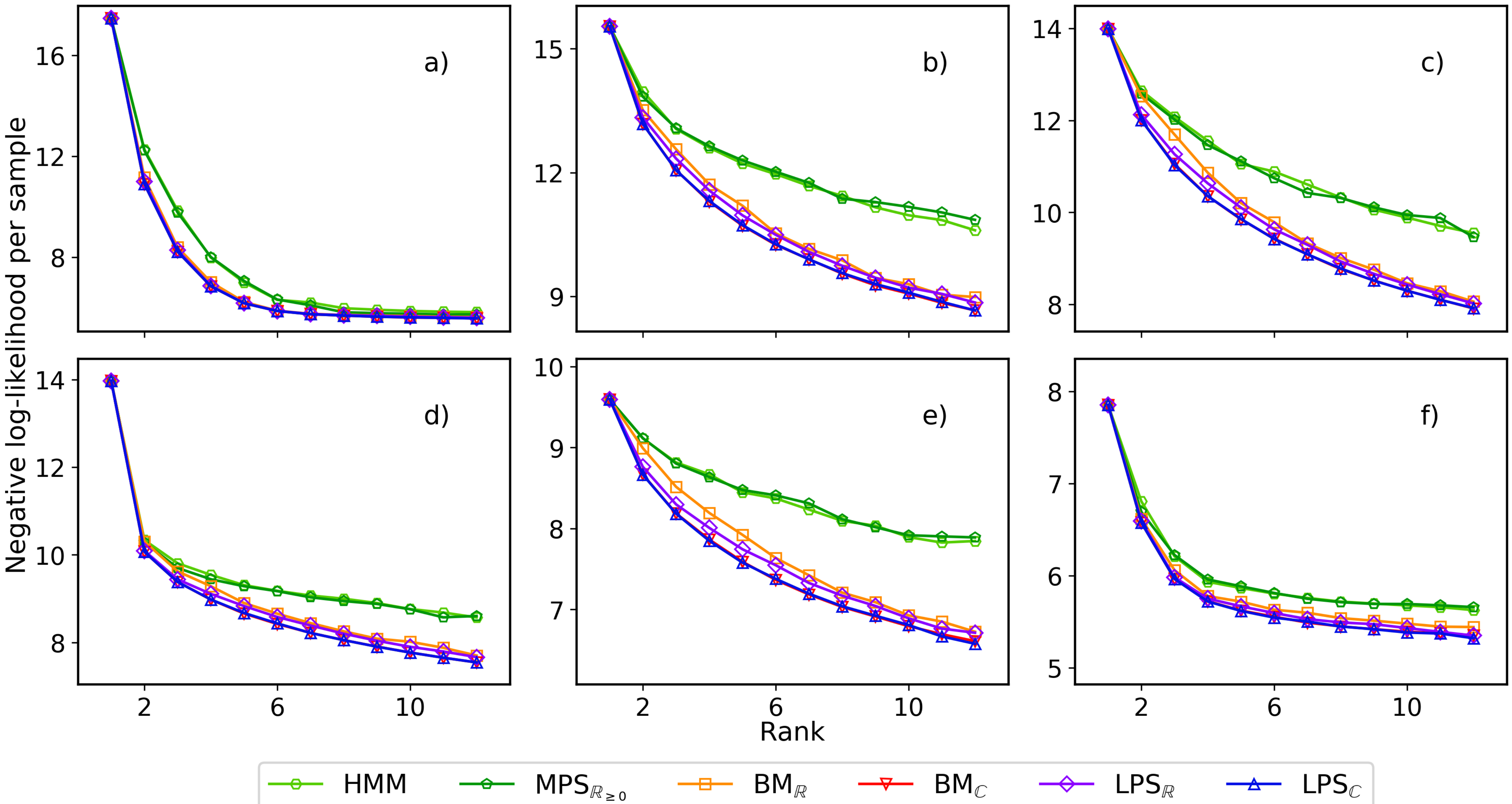
These are *exact* results! In practice we are interested in *approximations*…

We can explore this numerically:



# Expressivity Results

In addition, how well do these models perform as hypothesis classes?



# Future Directions + Vision

1) We need good algorithms for learning in an HQC setting, to turn these results into good heuristics!

Already some ideas here…

2) Of course, we would like to prove the conjectures :) Help is welcome!

3) Overheads in the approximate case? Techniques are needed!

3) Can the general strategy be expanded to other rigorous quantum/classical comparisons?

- Deep Neural Networks (already some ideas)
- More complicated circuit topologies (also some ideas)

4) Even more generally, can we identify well posed mathematical questions in statistical learning theory which

- Lie at the quantum/classical interface?
- Would lead to enhanced heuristics if solved?

(Also, thanks to Ivan Glasser, Nicola Pancotti, Ignacio Cirac and Jens Eisert!)