TaQos

Encoding of Probability Distributions for Quantum Monte Carlo Using **Tensor Networks**

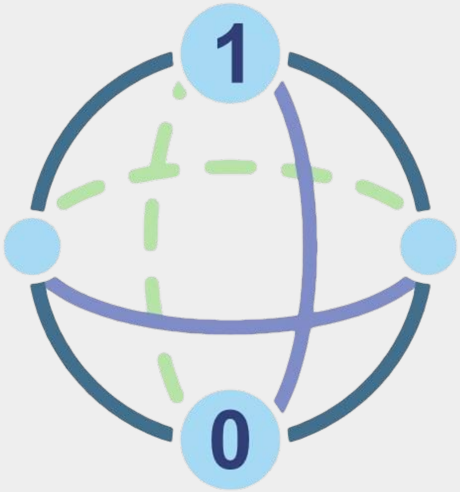
Problem statement

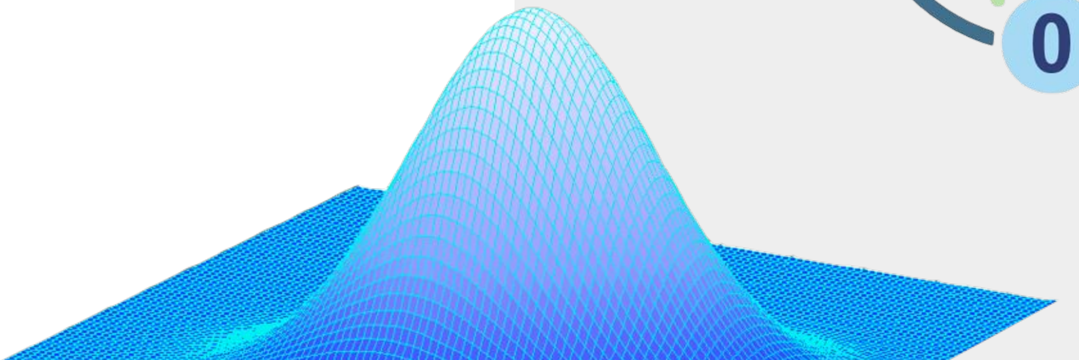
Challenge on Quantum Monte Carlo (QMC)

Probability distribution

$$f(\mathbf{x}, \mu, \mathbf{\Sigma})$$

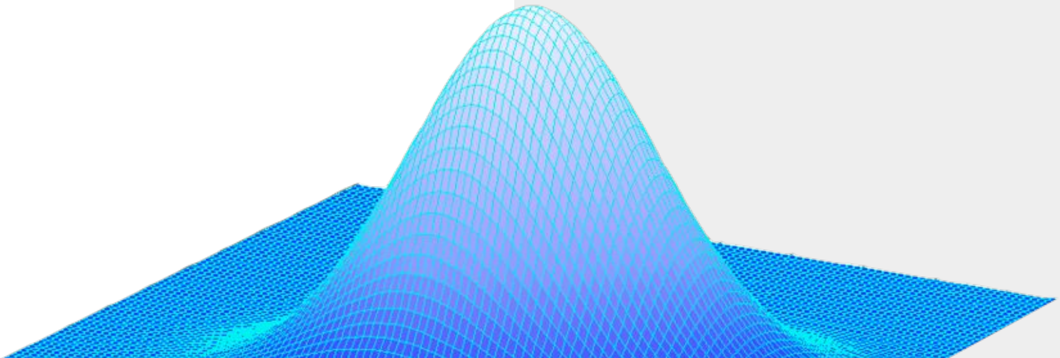
ENCODE





Motivation

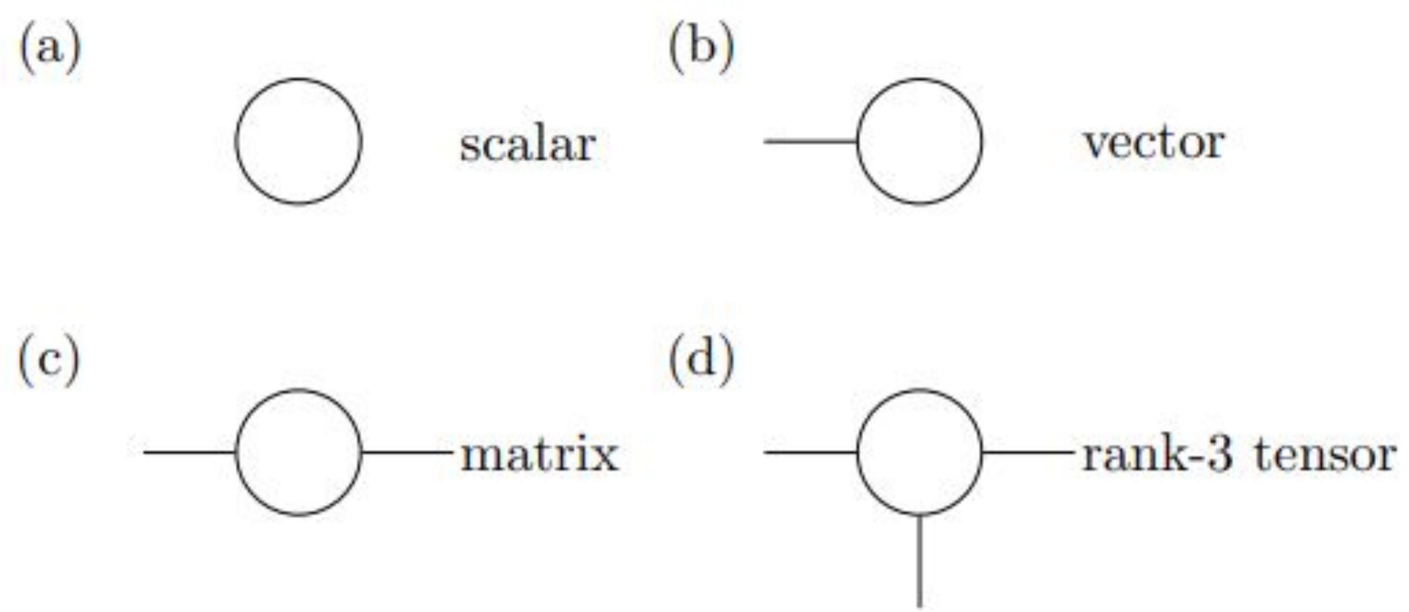
- Useful for solving high-dimensional integrals that arise in **Many Body Physics** and other **domains of science**.
- Real-world applications like **Quantum Finance**, **optimization**, and **risk assessment**.



Tensor Networks

Represents and manipulates **high-dimensional data**.

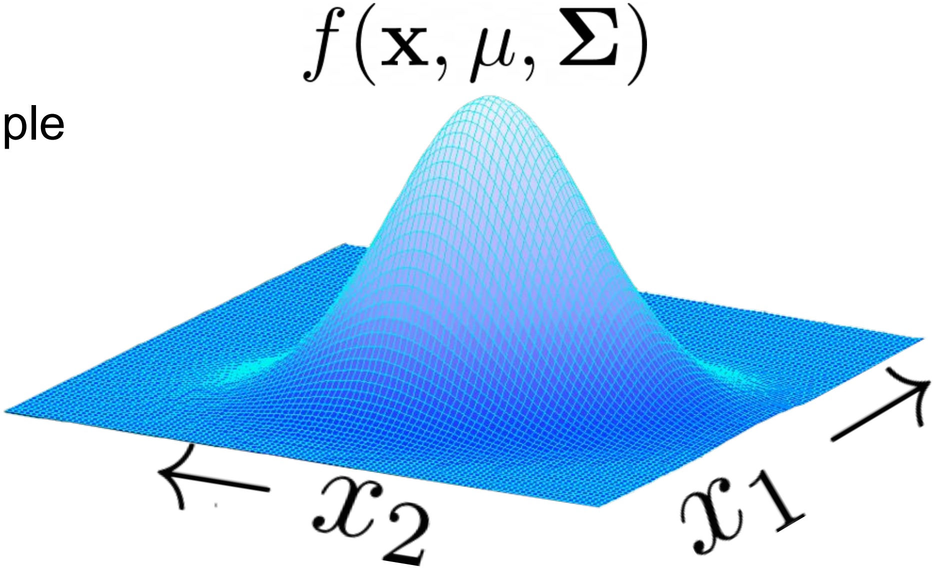
Let us **decompose complex tensors** into simpler ones.



Data Encoding ("quantization")

Approach:

- Define number of qubits $n \text{ qubits} \Rightarrow 2^n \text{ values}$
- Define dimensions
- Create a volume to sample

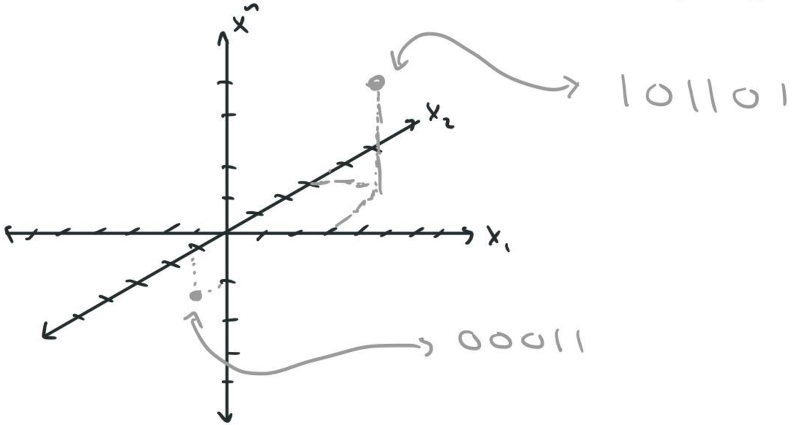


Data Encoding ("quantization")

Approach:

- Map one bitstring to one volume coordinate
  - Different maps are different quantizations

$$m : b \to i \text{ in } \mathbf{x}_i$$



Data Encoding ("quantization")

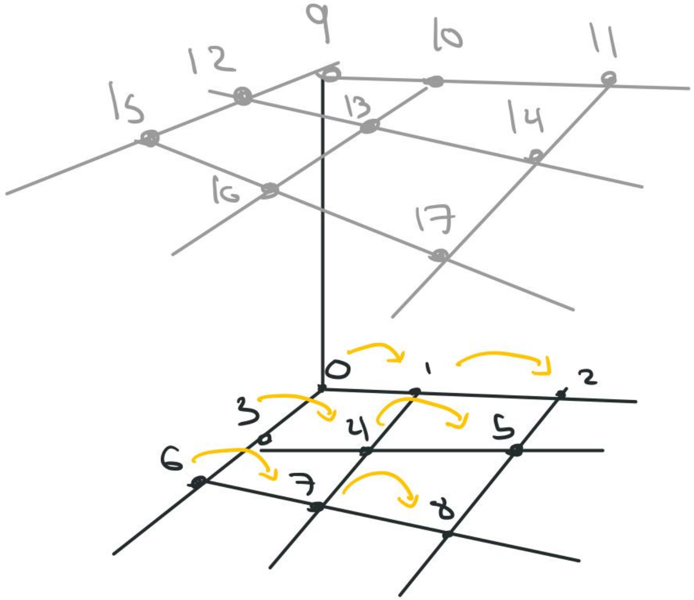
Approach:

- **Sequential Quantization**

$$f : \mathbb{R}^d \rightarrow \mathbb{R}$$

$$p(x_i) = A(i_1, i_2, ...i_d)$$

- **Interleaving Quantization (braiding)**



Quantum tensor train (TT) approximation

$$A(i_1, \ldots, i_d) = \sum_{r_0, r_1, \ldots, r_d} G_1(r_0, i_1, r_1) G_2(r_1, i_2, r_2) \cdots G_d(r_{d-1}, i_d, r_d),$$

High Dimensional Array → Sequence of lower-dimensional tensors

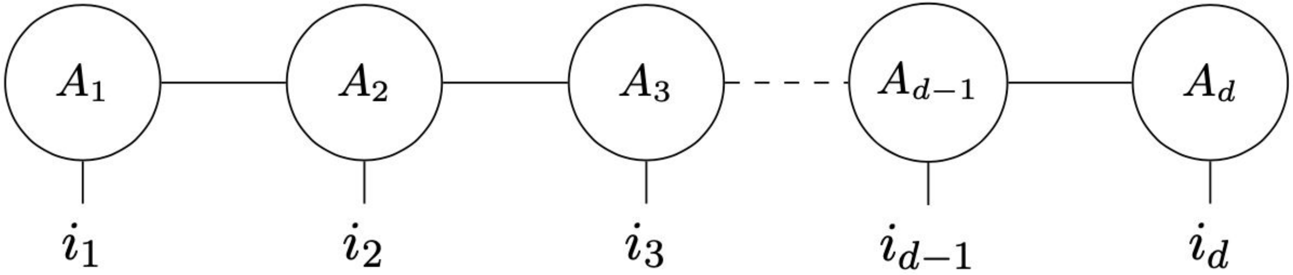
```
10D TT tensor:

  2   2   2   2   2   2   2   2   2   2
  |   |   |   |   |   |   |   |   |   |
 (0) (1) (2) (3) (4) (5) (6) (7) (8) (9)
 / \ / \ / \ / \ / \ / \ / \ / \ / \ / \
1   2   4   8   8   8   8   8   4   2   1
```

- Define “physical dimensions” = single qubit dimension
- Create a domain space from each qubit

Tensor train (TT) approximation

Approach:

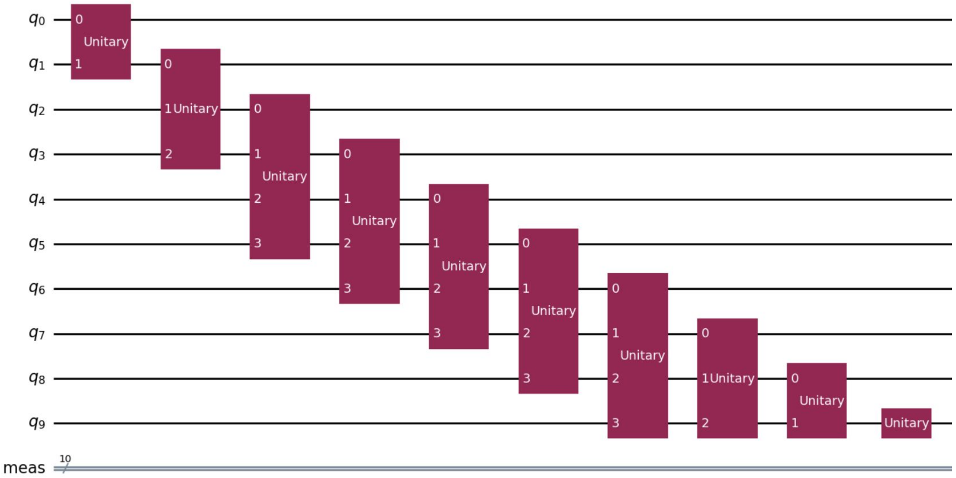
- Cross TT approximation usage
  - Define bond-dimension (max amount of interacting qubits)
  - Samples probability distribution over random subset of domain
  - Optimizes contractions to form tensor which represents data
  - Output tensor has indices corresponding to the physical dimensions

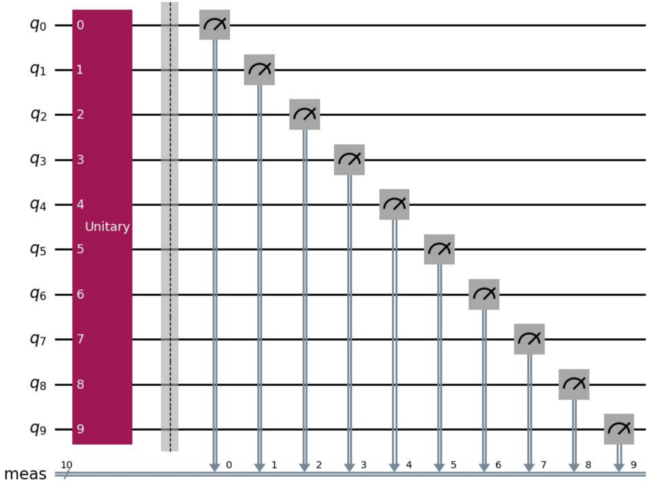


Mapping TT to circuit

Approach:

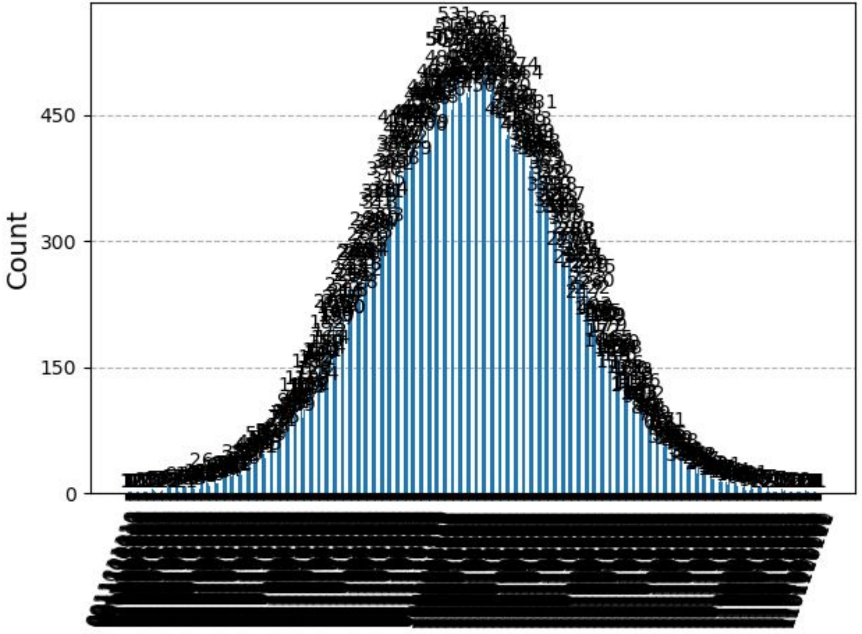
- Very smart reshaping per tensor core and SVD decompositions
- Get’s unitary matrices which recreate tensor
- Can then map to one single unitary :)

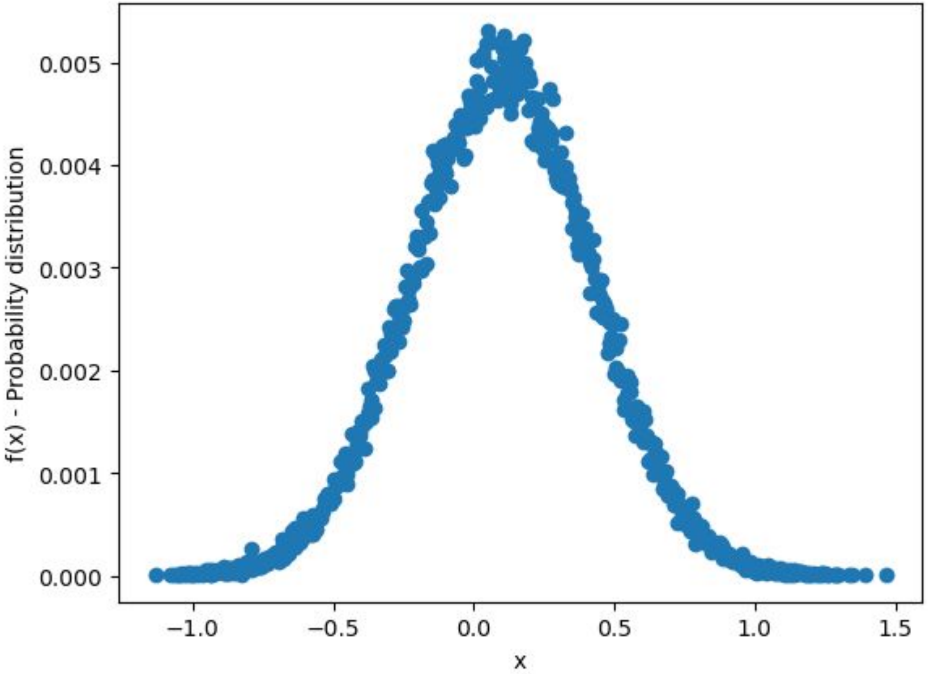




1D results

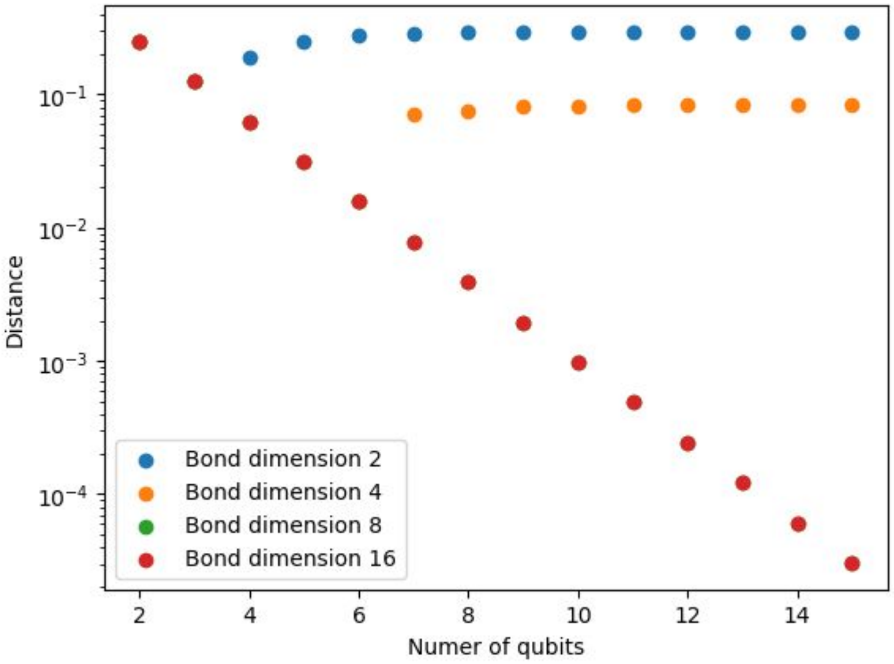
Results 1D

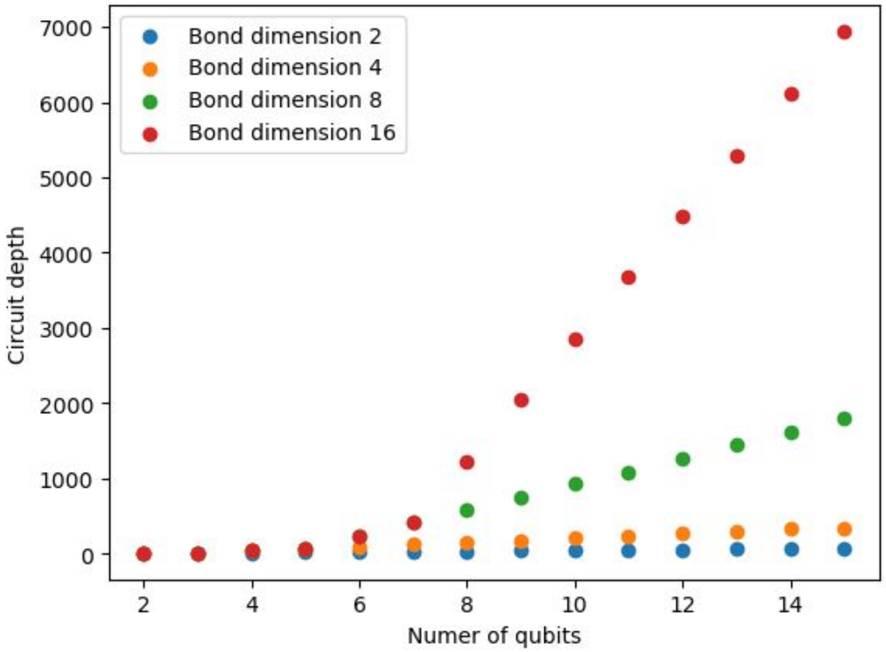




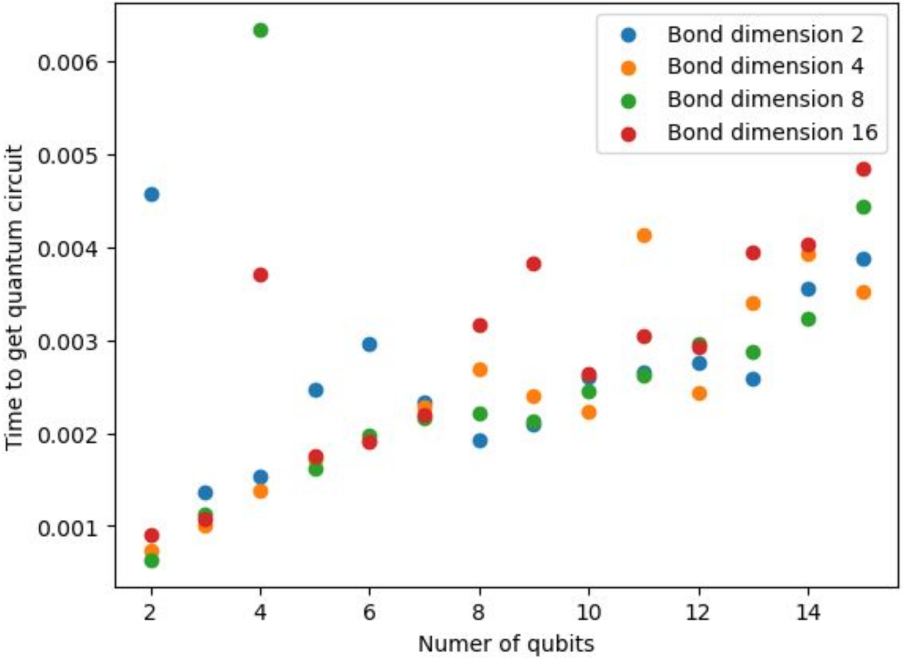
Bitstrings → x
Counts → f(x)

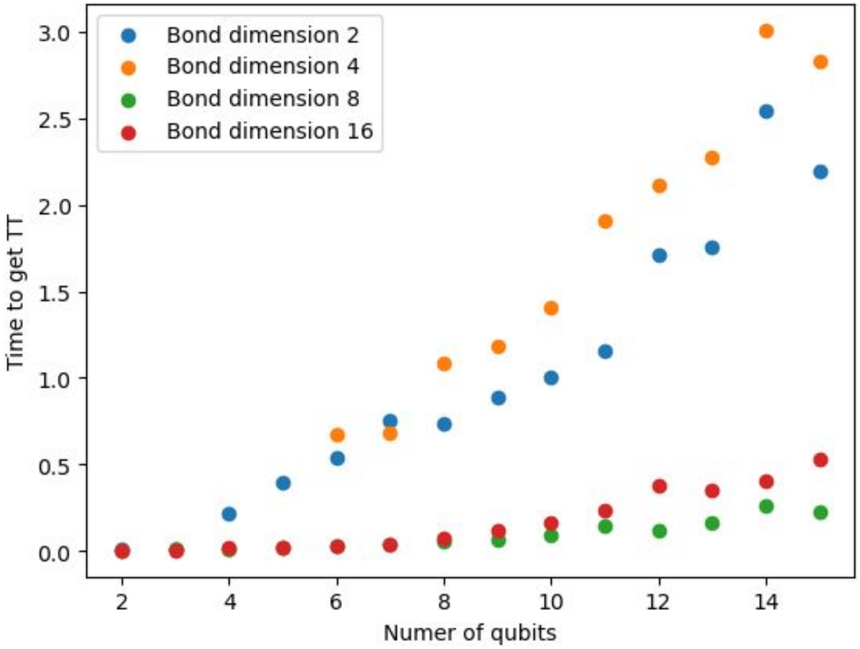
Distance and scaling



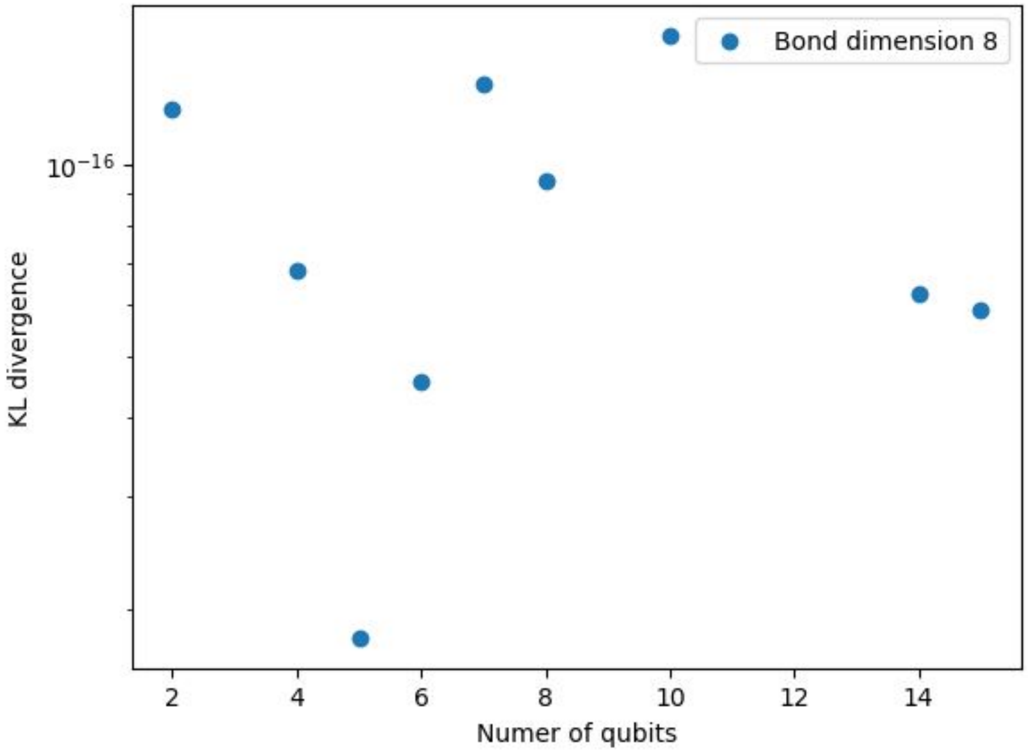


Time scaling



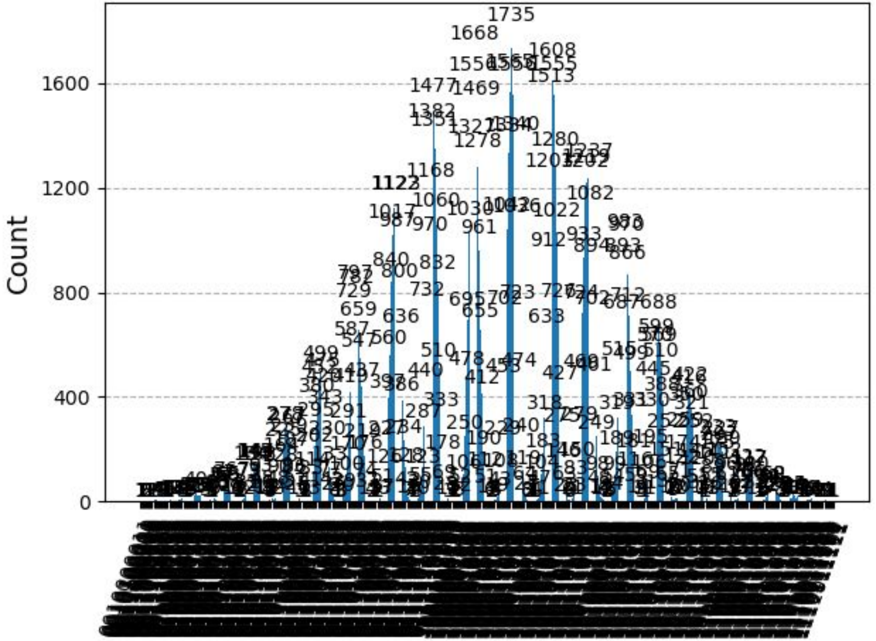


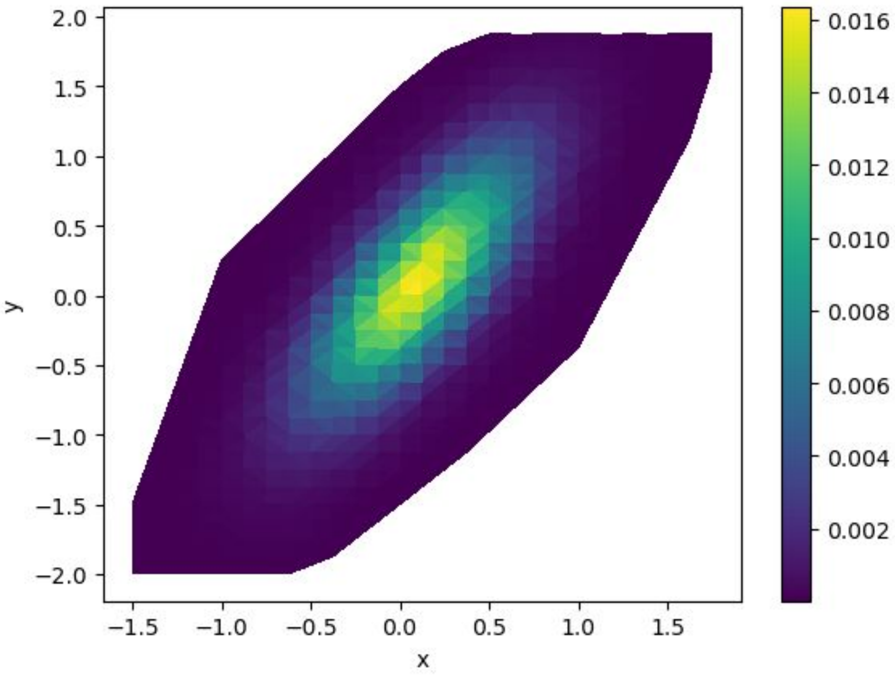
KL Divergence



2D results

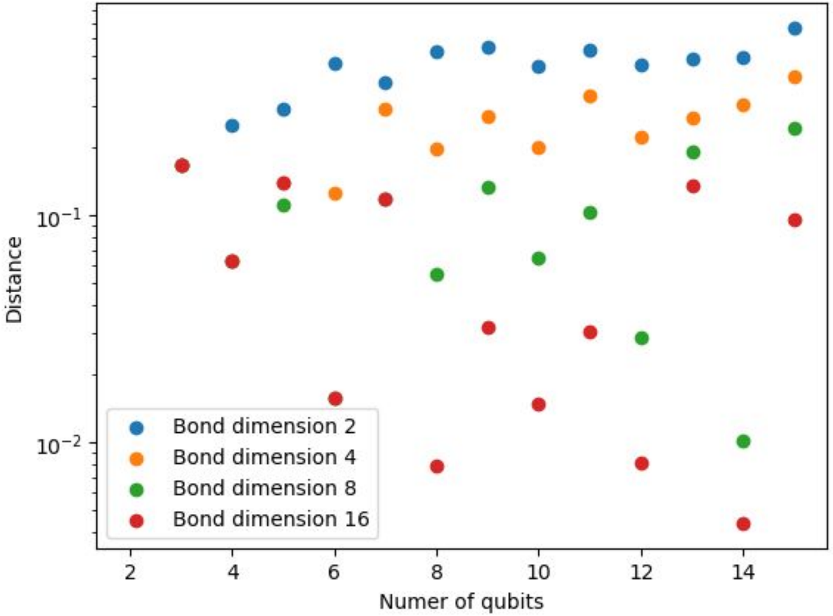
Results 2D

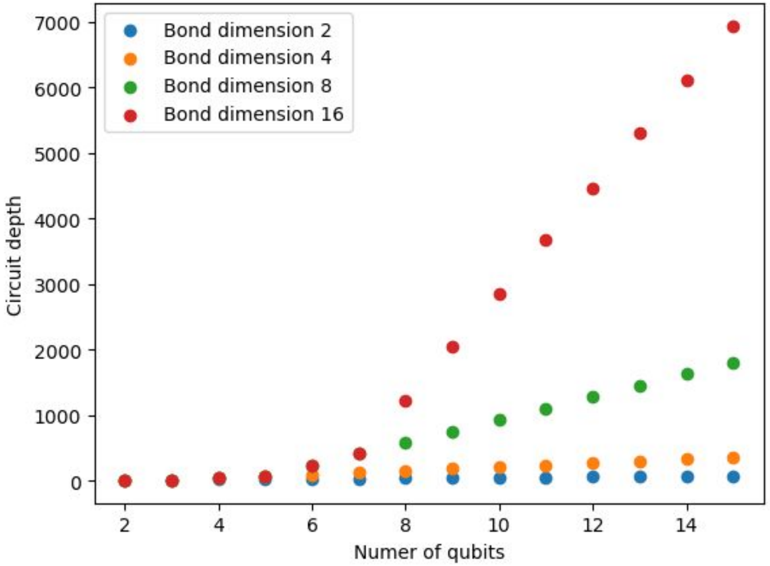




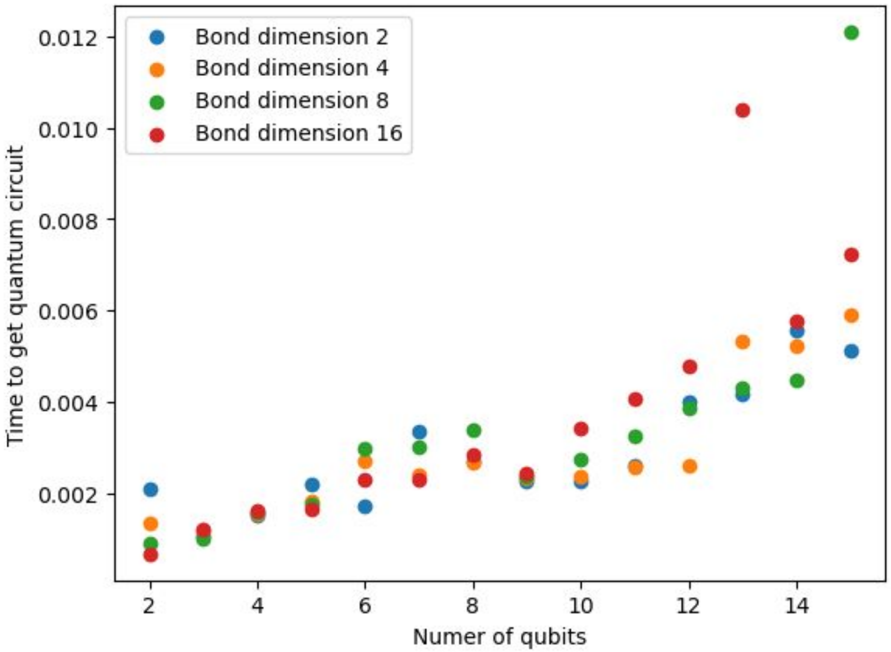
Bitstrings → (x, y)
Counts → f(x, y)

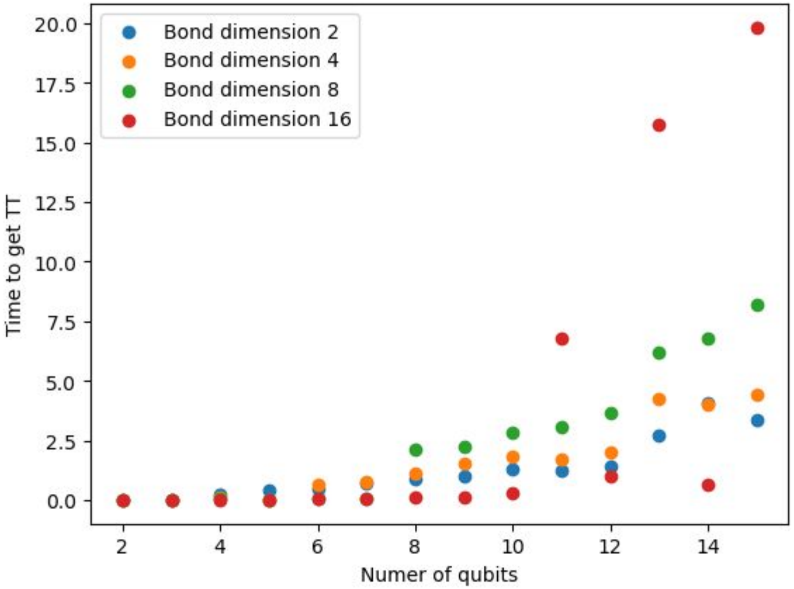
Distance and scaling





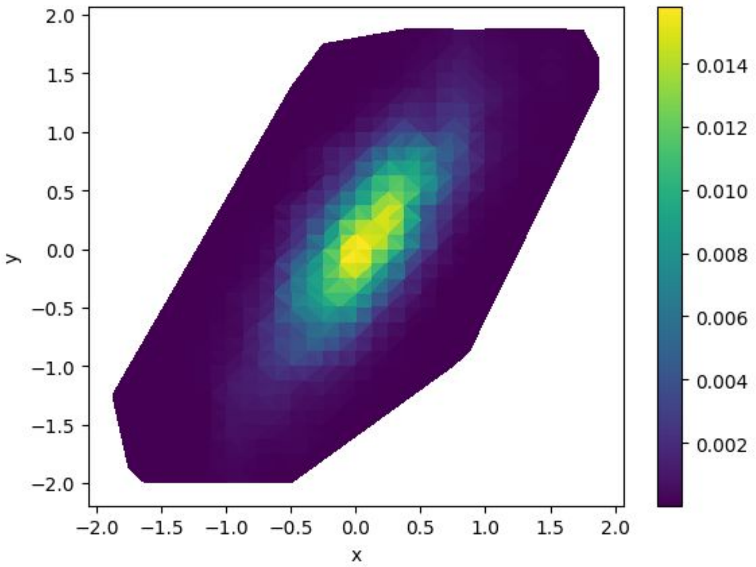
Time scaling

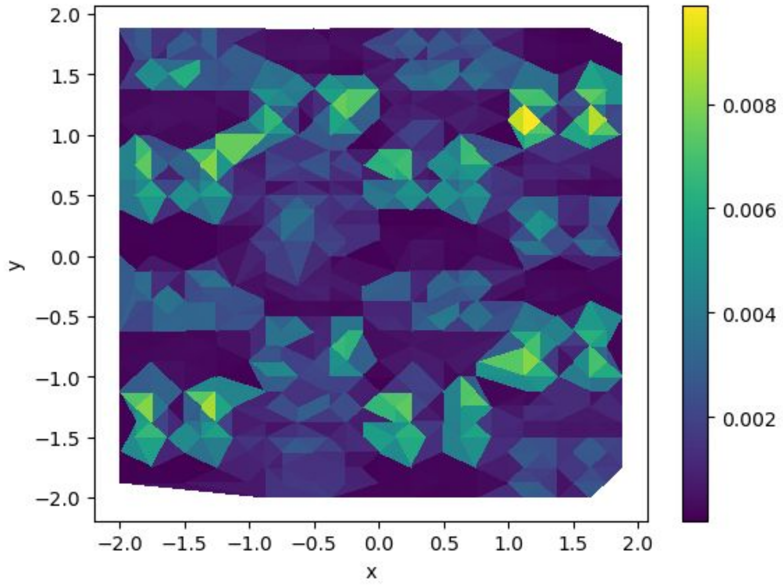




Comparison: different quantizations

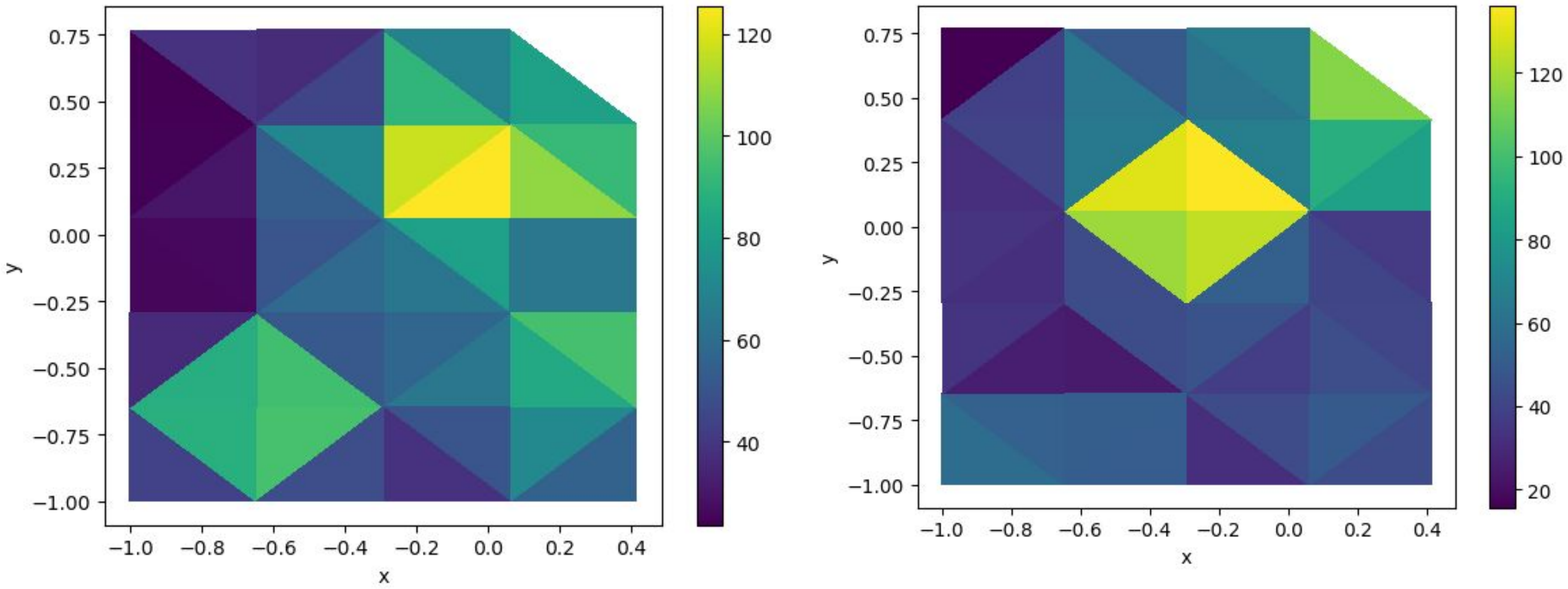
- Interleaving pattern & some weird pattern



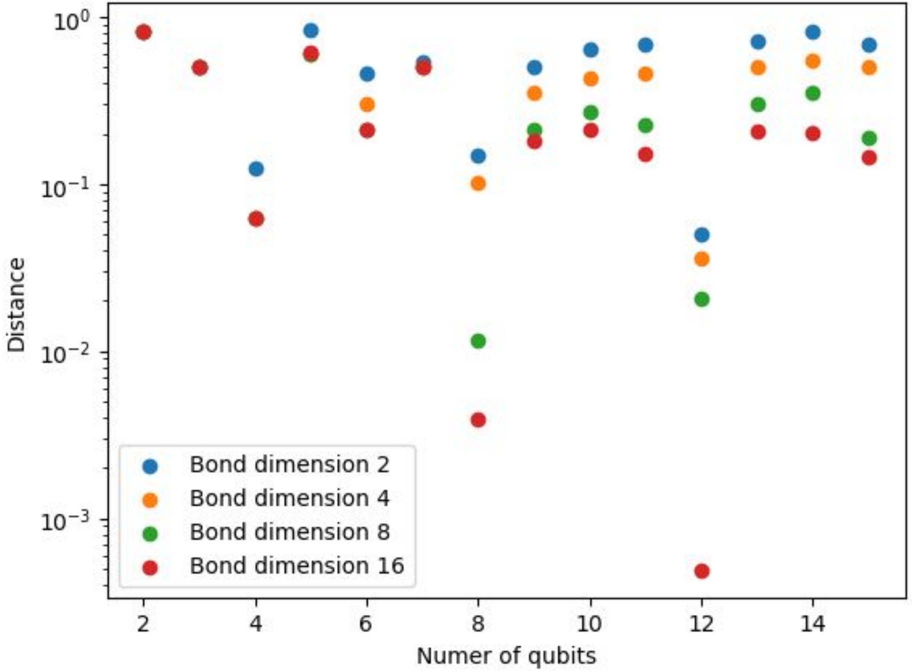


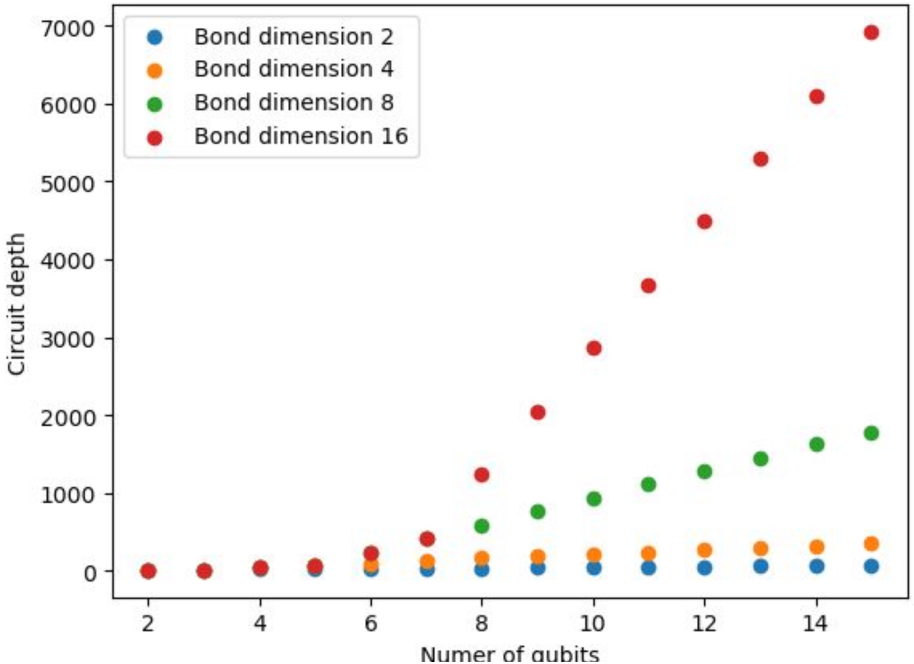
4D results

Results 4D sequenced vs interleaved

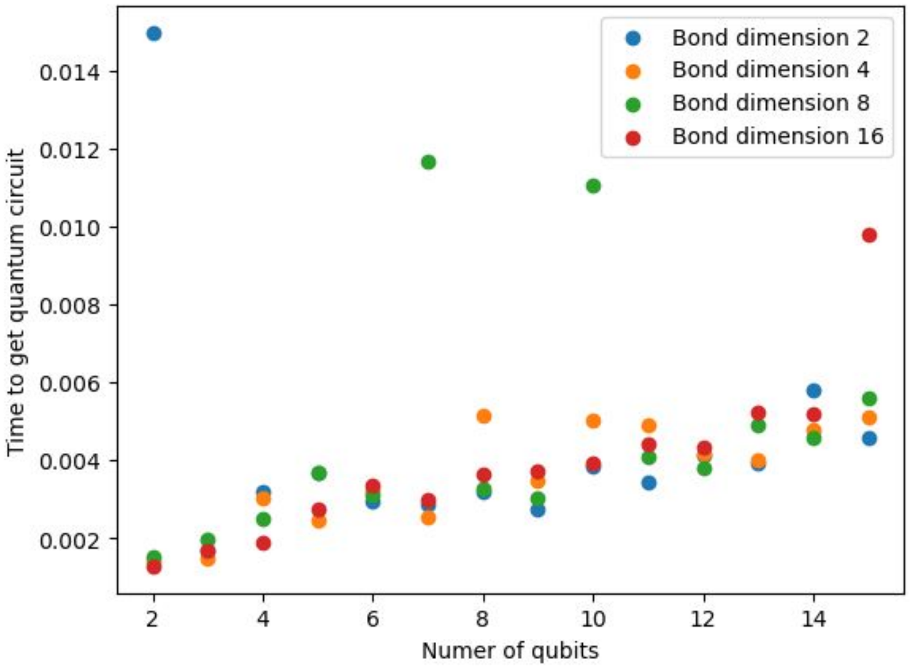


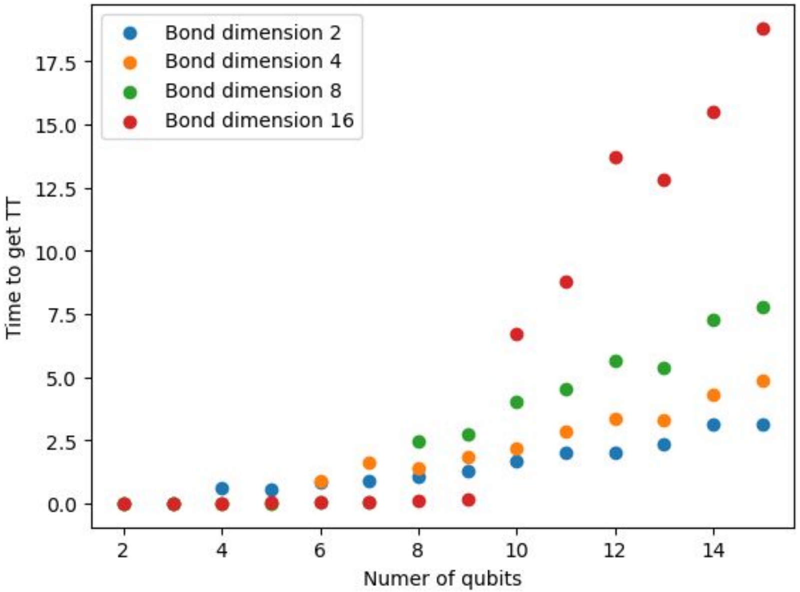
Distance and scaling (sequenced)





Time scaling (sequenced)





Analysis

TT train & QC prep time

- Linear scaling with number of qubits
- Quasi-linear with bond dimension

Quantum Circuit depth

- Linear scaling with number of qubits
- Quasi-linear with bond dimension

Distance

- Sequenced data worsens for more dimensions as expected
- Interleaved performs better
- Often improves with high bond dimension

Conclusion

- Can prepare d-dimensional distribution function, scaling linearly (time and depth) with number of qubits
- Only hard task → Good quantization
- Trade-off between resources vs good approximation (time,depth vs bond dimension)

Discussion

Algorithms assume state-preparation

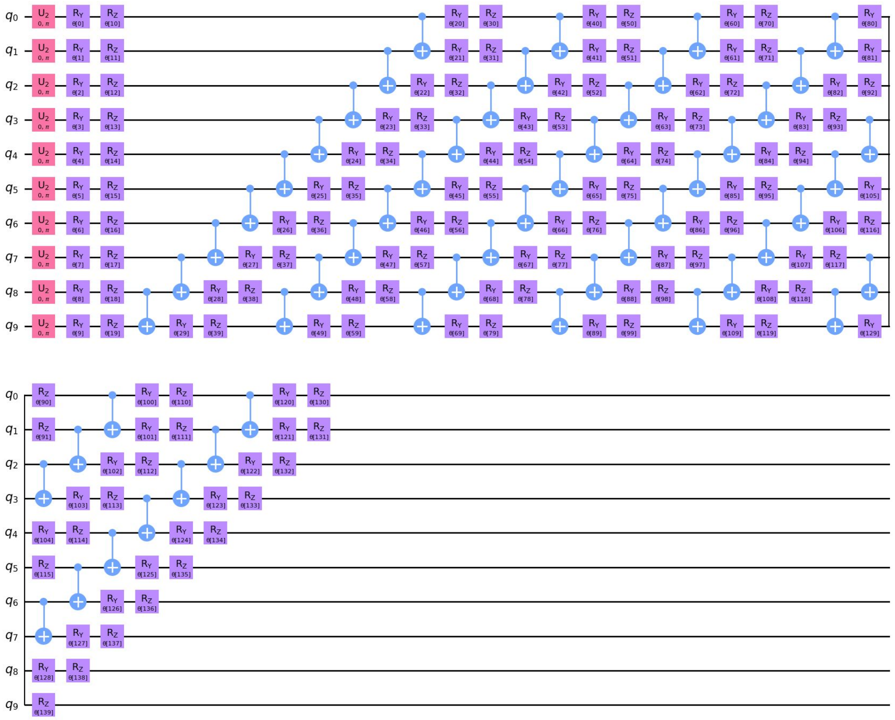
- Examples:
  - Topological invariant calculations in condensed matter physics.
  - Adiabatic simulations
- Don’t work without effective density matrix upload.

Probability distribution preparation

- In many-dimensional systems not one single method works for all.
- Encoding method open to creativity.
- Usually big trade-off between precision and resources.

Bonus

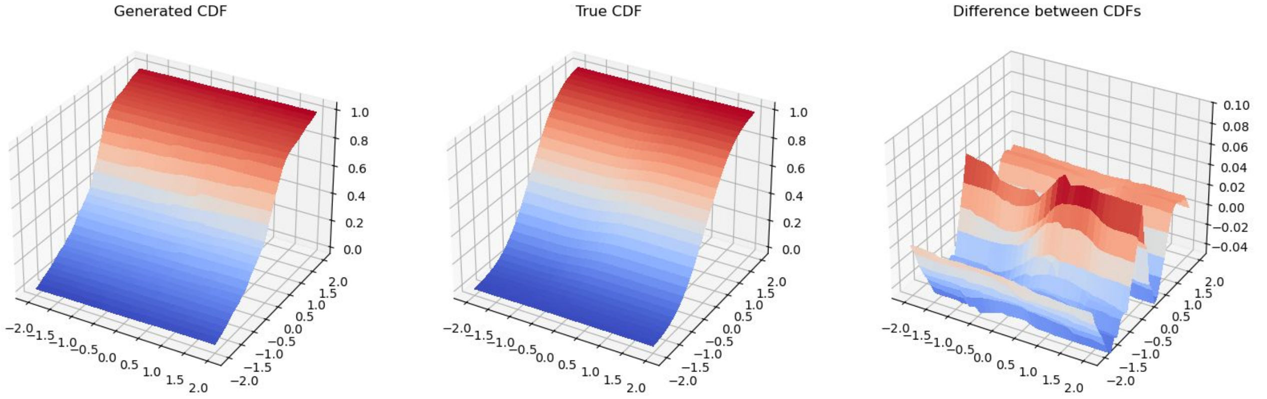
qGAN 2D

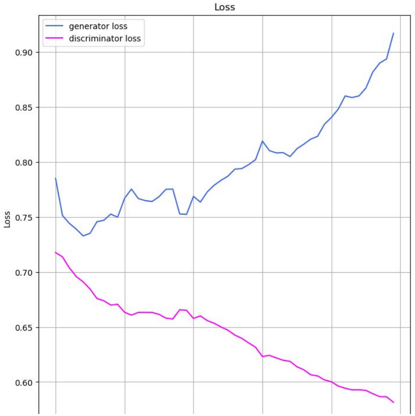


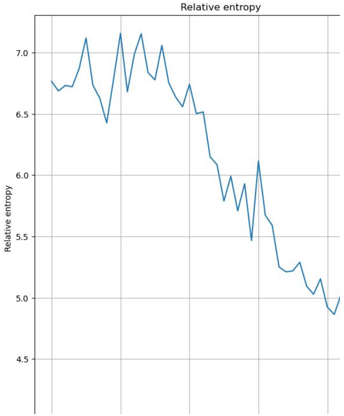
General idea

- Optimize a parametrized quantum circuit to map a probability distribution to a state.
- A neural network tries to discriminate the circuit's output from the real distribution and gives feedback.

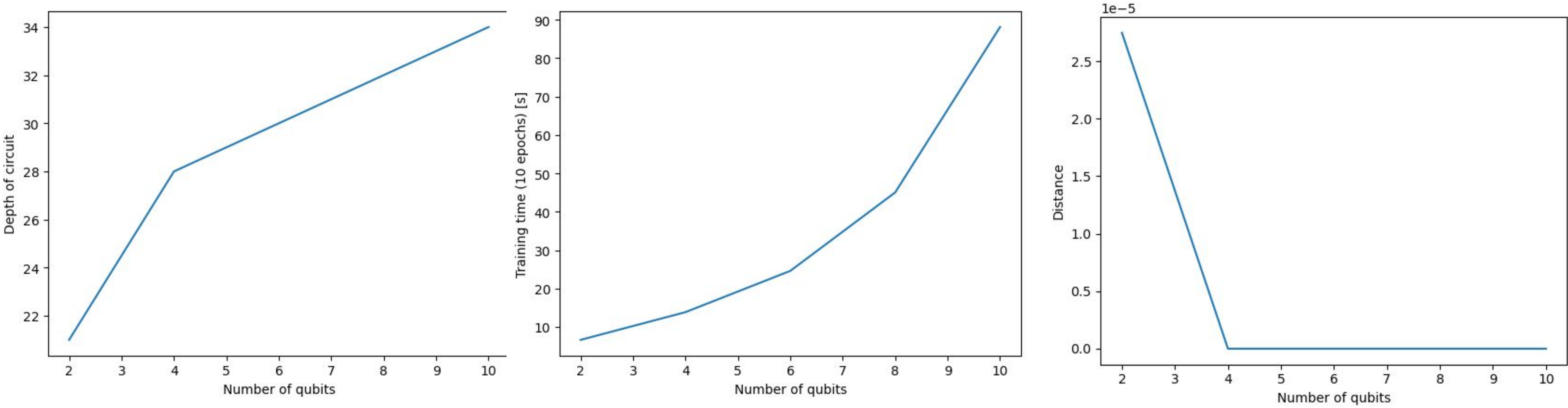
qGAN 2D results







Scaling


Depth of circuit
34
32
30
28
26
24
22
2 3 4 5 6 7 8 9 10
Number of qubits
Training time (10 epochs) [s]
90
80
70
60
50
40
30
20
10
2 3 4 5 6 7 8 9 10
Number of qubits
1e−5
Distance
2.5
2.0
1.5
1.0
0.5
0.0
2 3 4 5 6 7 8 9 10
Number of qubits

Discussion

- Training time longer than TT-train circuit
- Time seems to have quadratic scaling with qubits → Very bad
- Circuit depth good (but iterations are not accounted for)
- Very good distance

Thank you for your attention!

Questions?