Variational Quantum Factoring

Open Source Research project

05.05.2020, Zaiku Group webinar

About me

- Quantum Software Engineer at Zapata Computing
    - Building Orquestra
    - Implementing algorithms
    - Helping with research projects
- Quantum Open Source Foundation
    - QC Mentorship program
- Q4Climate
    - Getting it all started!
- Musty Thoughts (blog)

Some context

- Presentation done as a part of recruitment process to Zapata in May 2019.
- I assume you have some idea of how the QAOA works.
- It’s implementation of the algorithm by Eric Anschuetz et al.

Goal of the presentation

- Explain how VQF works
- Give you more intuition on practical use of QAOA
- Show how you can do a QC research project

Presentation plan

1. Introduction
2. Algorithm overview
3. Implementation
4. Research
5. Summary

Why implementing VQF?

Because it:

- is interesting

- is a good example of hybrid approach

- is an unconventional use of QAOA

- is an important problem

- is a good way to learn

- makes a good open source project

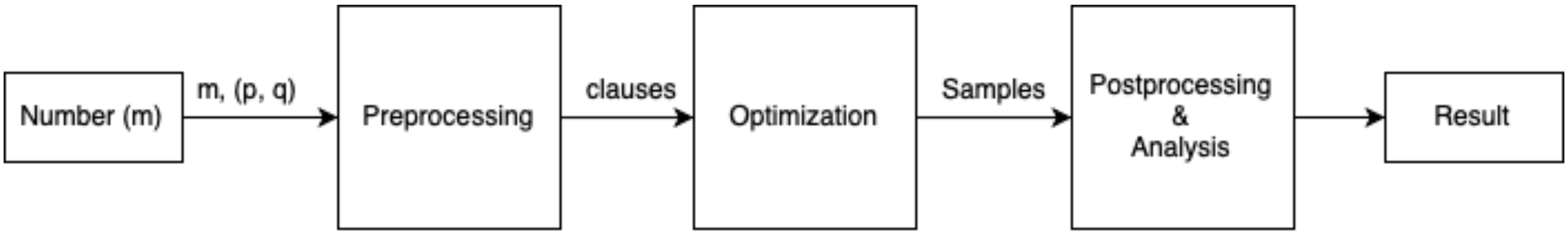
- seems suitable for the recruitment for Zapata

Goal of the algorithm

Given a number m, find p and q such that $m = p * q$.

Both p and q are primes.

Algorithm overview



Mathematical representation

$$m = \sum_{k=0}^{n_m-1} 2^i m_k, \qquad p = \sum_{k=0}^{n_p-1} 2^i p_k, \qquad q = \sum_{k=0}^{n_q-1} 2^i q_k,$$

From m = p*q we get a set of equations:

$$0 = \sum_{j=0}^{i} q_j p_{i-j} + \sum_{j=0}^{i} z_{j,i} - m_i - \sum_{j=1}^{n_c} 2^j z_{i,i+j}$$

$$C_i = \sum_{j=0}^{i} q_j p_{i-j} + \sum_{j=0}^{i} z_{j,i} - m_i - \sum_{j=1}^{n_c} 2^j z_{i,i+j},$$

$$0 = \sum_{i=0}^{n_c} C_i^2.$$

Source: Variational Quantum Factoring

Example

Table 1: *Multiplication table for* $11 \times 13 = 143$ *in binary.*

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
p					1	p_2	p_1	1
q					1	q_2	q_1	1
					1	p_2	p_1	1
				q_1	p_2q_1	p_1q_1	q_1	
			q_2	p_2q_2	p_1q_2	q_2		
		1	p_2	p_1	1			
carries	z_{67}	z_{56}	z_{45}	z_{34}	z_{23}	z_{12}		
	z_{57}	z_{46}	z_{35}	z_{24}				
$p \times q = 143$	1	0	0	0	1	1	1	1

$$
\begin{aligned}
p_1 + q_1 &= 1 + 2z_{12} \\
p_2 + p_1q_1 + q_2 + z_{12} &= 1 + 2z_{23} + 4z_{24} \\
1 + p_2q_1 + p_1q_2 + 1 + z_{23} &= 1 + 2z_{34} + 4z_{35} \\
&\vdots \\
q_2 + p_2 + z_{45} + z_{35} &= 0 + 2z_{56} + 4z_{57} \\
1 + z_{56} + z_{46} &= 0 + 2z_{67} \\
z_{67} + z_{57} &= 1.
\end{aligned}
$$

Source: Quantum factorization of 56153 with only 4 qubits

Preprocessing

$$
\begin{aligned}
p_1 + q_1 &= 1 + 2z_{12} \\
p_2 + p_1q_1 + q_2 + z_{12} &= 1 + 2z_{23} + 4z_{24} \\
1 + p_2q_1 + p_1q_2 + 1 + z_{23} &= 1 + 2z_{34} + 4z_{35} \\
&\vdots \\
q_2 + p_2 + z_{45} + z_{35} &= 0 + 2z_{56} + 4z_{57} \\
1 + z_{56} + z_{46} &= 0 + 2z_{67} \\
z_{67} + z_{57} &= 1.
\end{aligned}
\quad + \quad
\begin{aligned}
xy - 1 = 0 &\implies x = y = 1, \\
x + y - 1 = 0 &\implies xy = 0, \\
a - bx = 0 &\implies x = 1, \\
\sum_i x_i = 0 &\implies x_i = 0, \\
\sum_{i=1}^{a} x_i - a = 0 &\implies x_i = 1.
\end{aligned}
\quad = \quad
\begin{aligned}
p_1 + q_1 - 1 &= 0 \\
p_2 + q_2 - 1 &= 0 \\
p_2q_1 + p_1q_2 - 1 &= 0.
\end{aligned}
$$

Clauses + Apply preprocessing rules = Simplified clauses

Optimization

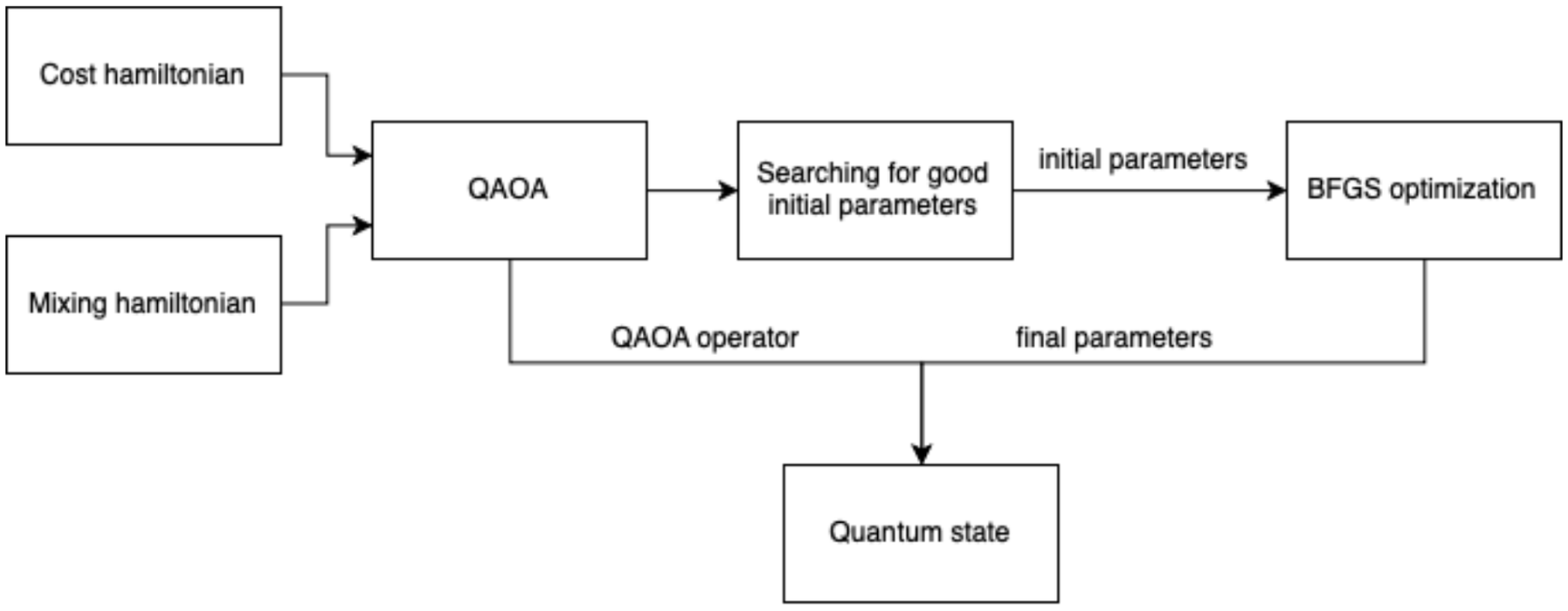
Clauses

$$
\begin{aligned}
p_1 + q_1 - 1 &= 0 \\
p_2 + q_2 - 1 &= 0 \\
p_2 q_1 + p_1 q_2 - 1 &= 0.
\end{aligned}
$$

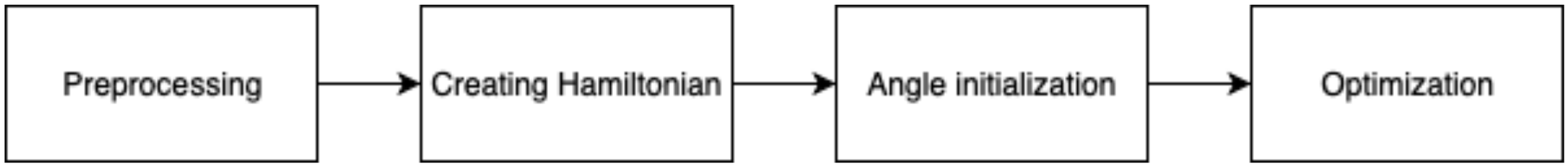
Cost function -> Hamiltonian

$$
\begin{aligned}
H &= (p_1 + q_1 - 1)^2 + (p_2 + q_2 - 1)^2 + (p_2 q_1 + p_1 q_2 - 1)^2, \\
&= 5 - 3p_1 - p_2 - q_1 + 2p_1 q_1 - 3p_2 q_1 + 2p_1 p_2 q_1 - 3q_2 + p_1 q_2 + 2p_2 q_2 + 2p_2 q_1 q_2 \\
a_i &= \frac{1}{2}\left(1 - \sigma_z^{(i)}\right)
\end{aligned}
$$

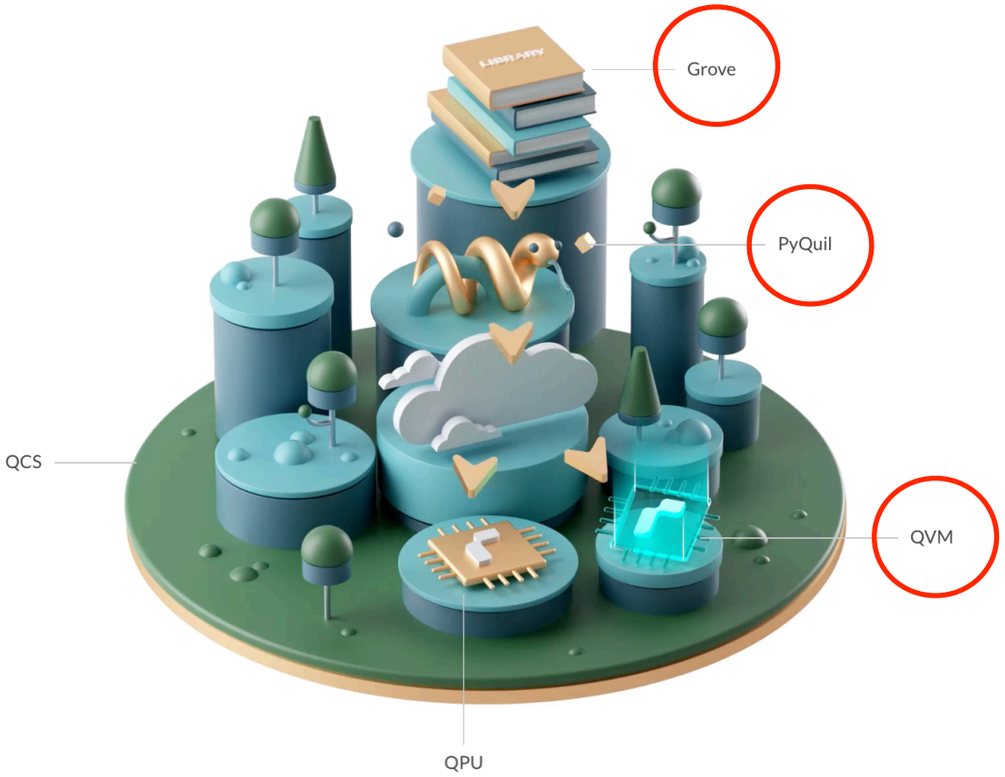
Optimization



VQF - scheme


Preprocessing
Creating Hamiltonian
Angle initialization
Optimization

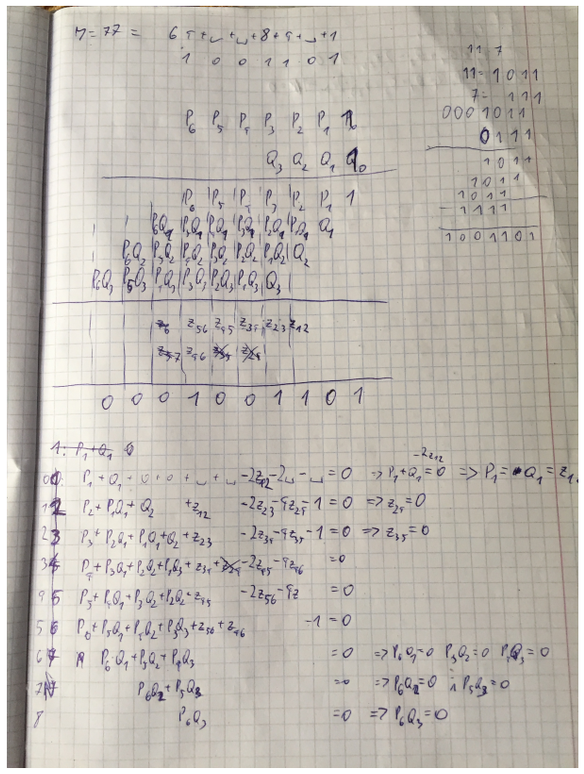
Tools


Grove
PyQuil
QCS
QVM
QPU


SymPy

git

GitHub



\+


HAMILTON

\+


GUARDIANS OF THE GALAXY
AWESOME MIX VOL. 1

Implementation

https://github.com/mstechly/vqf

Research

My research process

Research for this project:
https://github.com/mstechly/vqf/tree/master/research

Research methodology explained:
https://github.com/BOHRTECHNOLOGY/public_research/blob/master/Guidelines/research_guidelines.md

Experiments done

- Preprocessing
- Optimization - performance check
- Optimization - better initialization strategy
- Visualizing optimization space

Preprocessing

Goal: Implement preprocessing procedure from the paper.

My path:

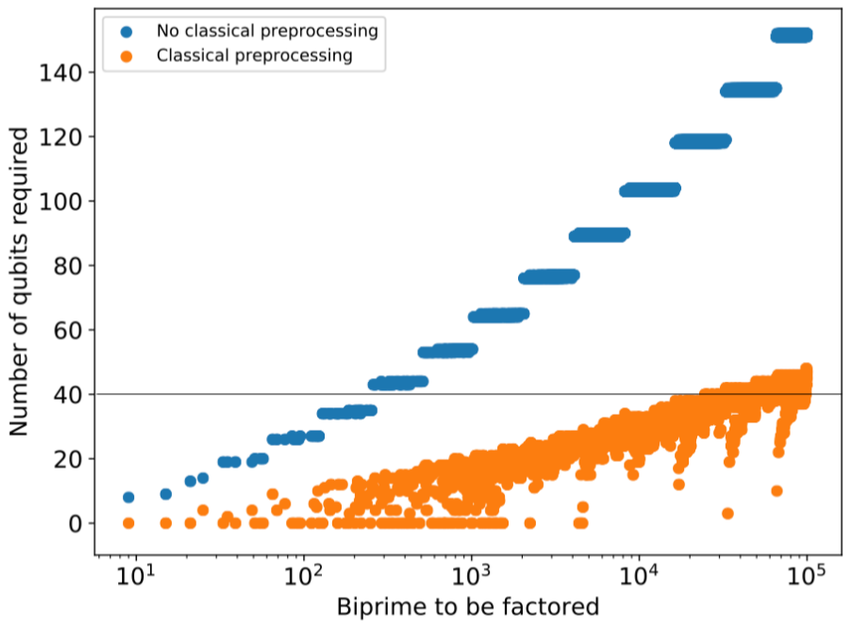
1. Implemented the logic
2. Couldn’t reproduce the results
3. Fixed the problem
4. Performed simulations and described the results

Improvements: Parity rule

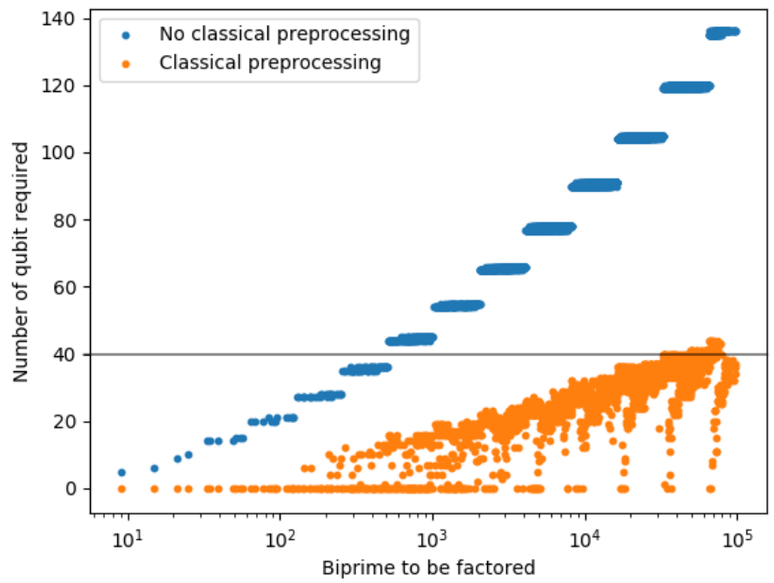
Clause	Relation
x + y - 2*z = 0	x = y = z
x + 2*y - 2*z = 0	x = 0 & y = z
x - 2*z +1 = 0	x = 1 & z = 1
a + b + 2*c + 2*d - 2*x - 4*y =0	a = b
x - y - 2*z + 2 = 0	z = y

Results

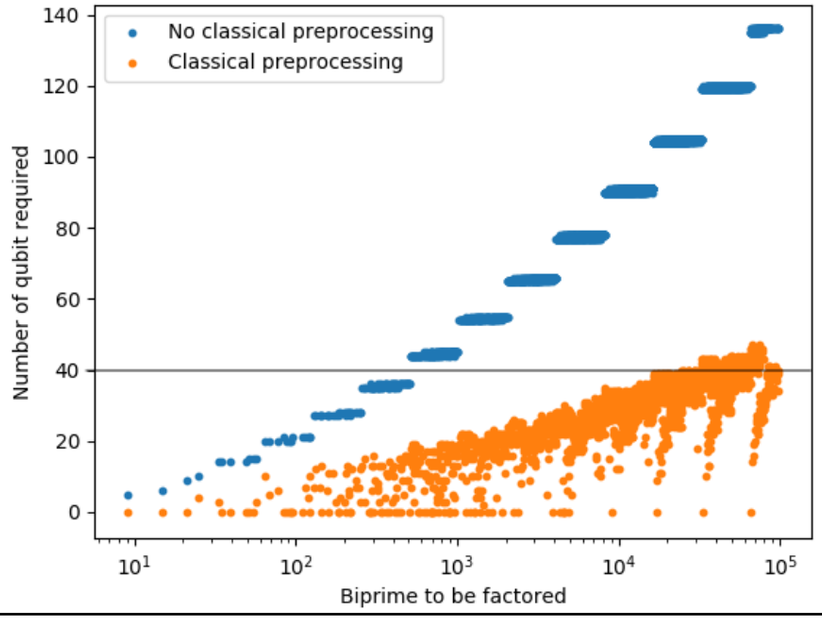
Original plot



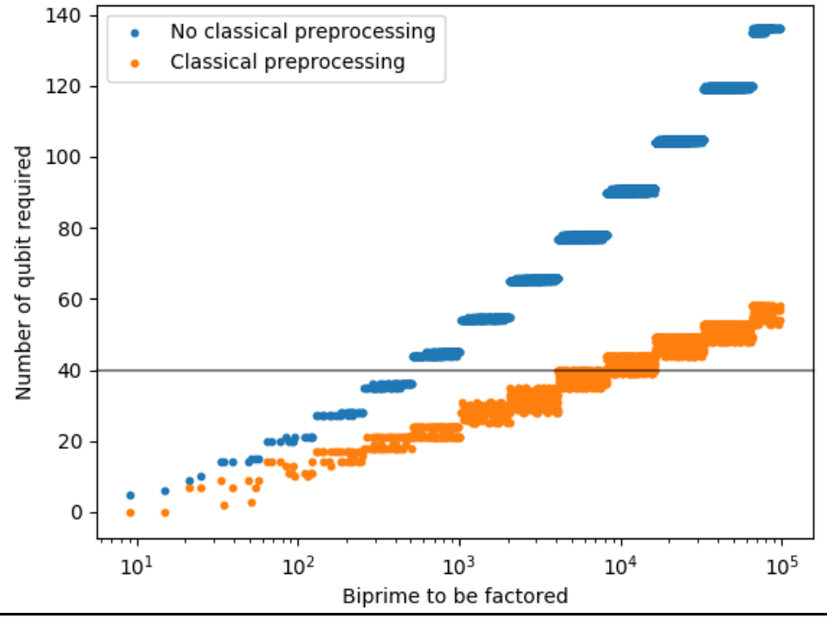
My preprocessing



Preprocessing without the parity rule



Preprocessing without known length of p and q



Preprocessing: software challenges

- Parsing SymPy expressions (a lot of ifs)
- Simplifying SymPy expressions
- SymPy randomness

Preprocessing: potential improvements

- Change loops to recurrence
- Improve rules
- Re-formulate parity rule
- Add more rules

Optimization: performance check

Goal: Reproduce optimization procedure from the paper.

My path:

1. Implemented the core code
2. Selected set of numbers
3. Added custom preprocessing rules for 56153 and 291311
4. Performed the simulations and described the results.
5. Added noise and sampling
6. Added L-BFGS-B

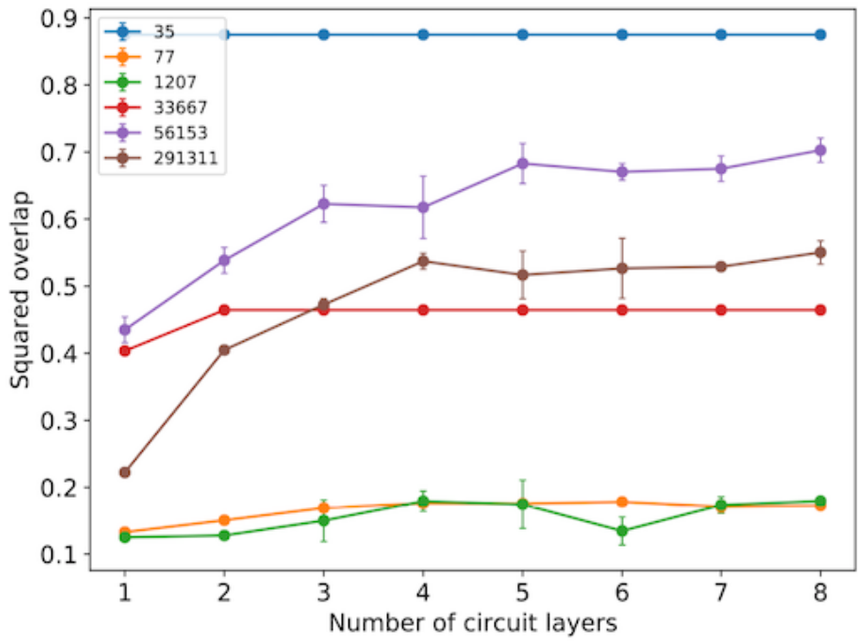
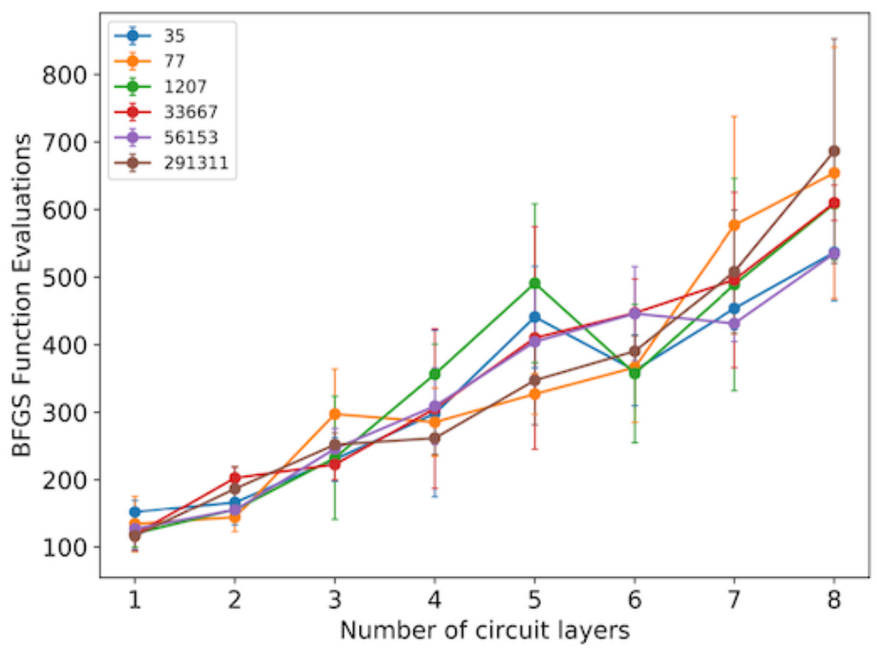
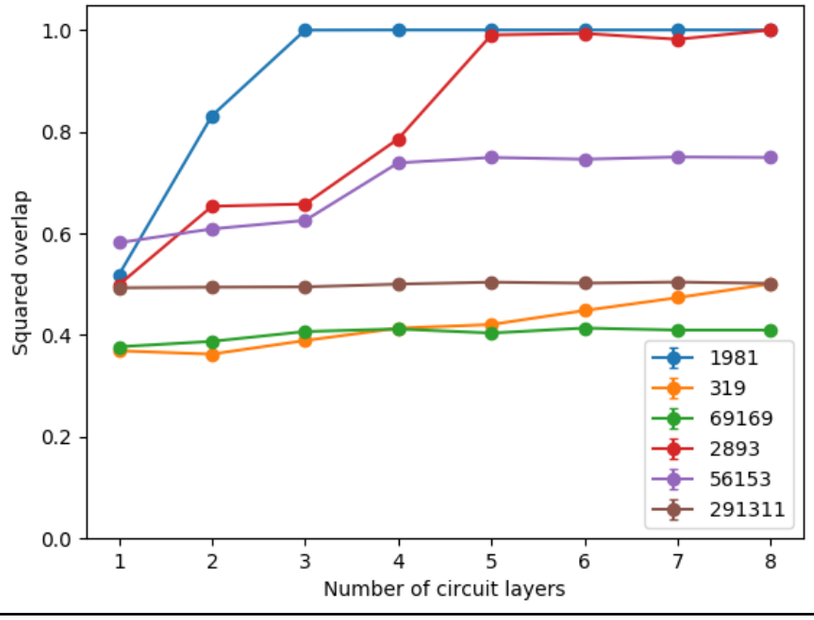
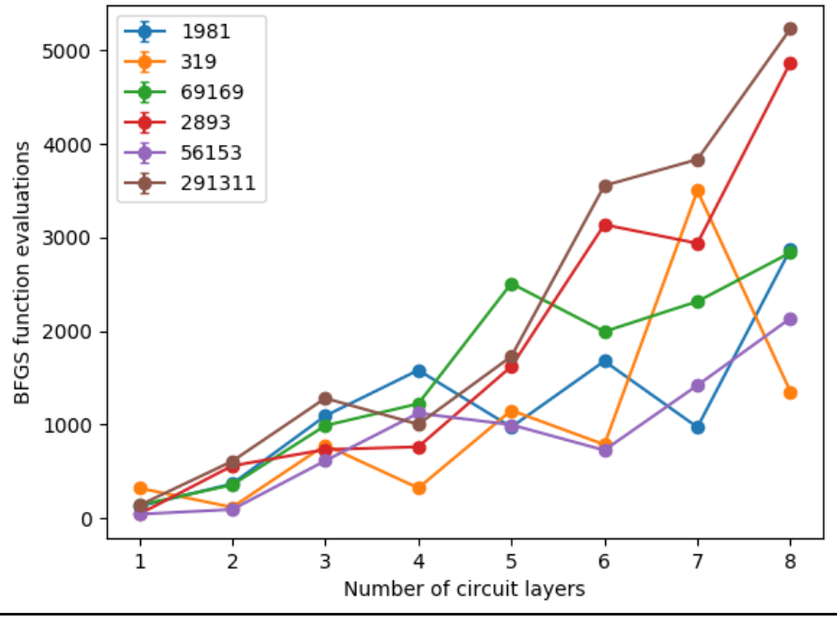
Performance check: Differences from the paper

1. Different numbers
2. Lack of noise
3. Results are deterministic
4. Implementation differences

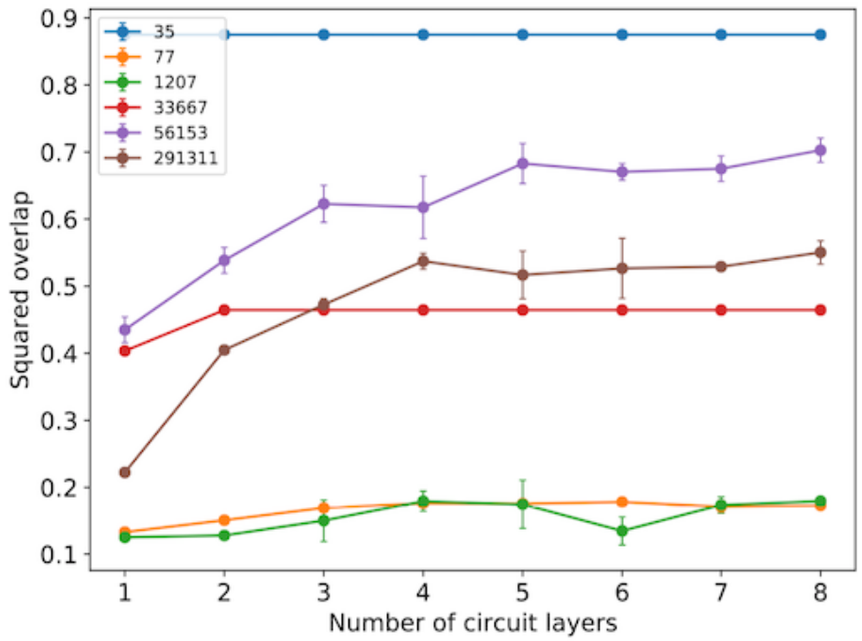
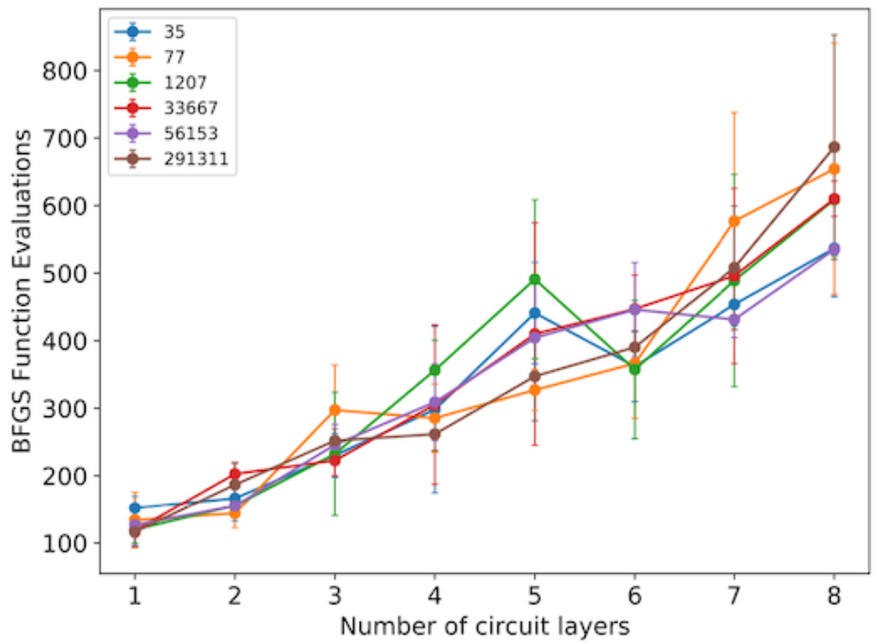
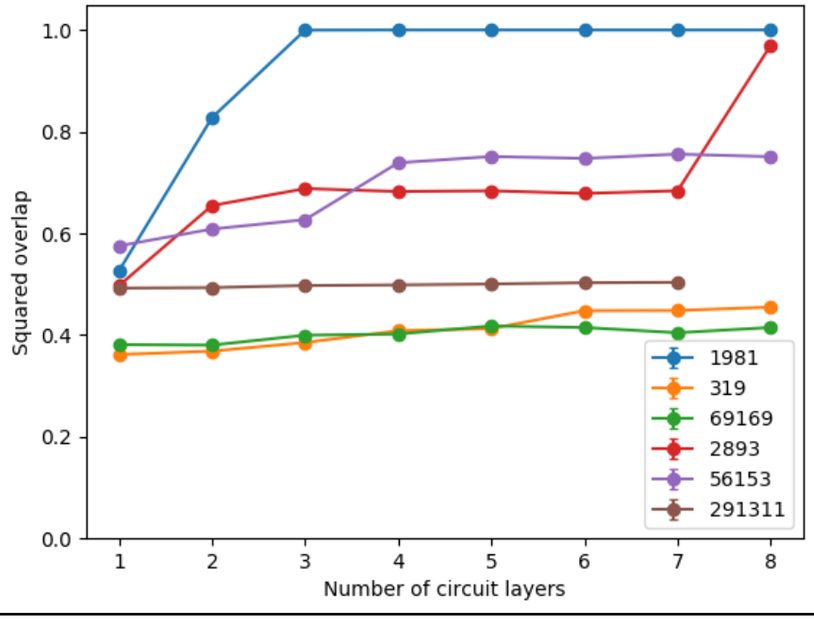
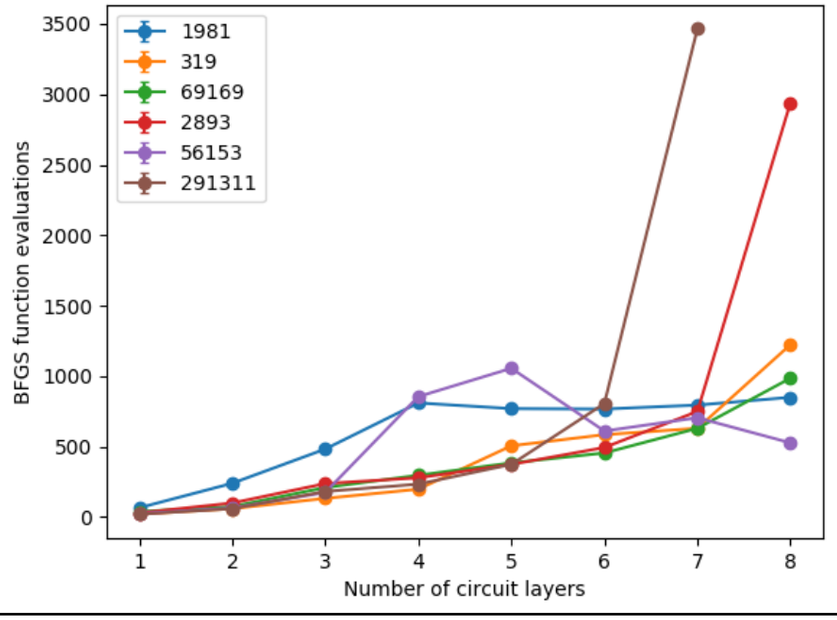
Performance check: Numbers used

Original m	New m	Number of qubits	Number of carry bits	Original symmetry	New symmetry
35 = 5 x 7	1981 = 7 x 283	2	0	Yes	No
77 = 7 x 11	319 = 11 x 29	6	3	Yes	Yes
1207 = 17 x 71	69169 = 263 x 263	8	5	No	Yes
33667 = 131 x 257	2893 = 263 x 11	3	1	No	No
56153 = 233 x 241	56153 = 233 x 241	4	0	Yes	Yes
291311 = 523 x 557	291311 = 523 x 557	6	0	Yes	Yes

Results - BFGS

Original	Original
	
My results	My results
	

Results - L-BFGS-B

Original	Original
	
My results	My results
	

Optimization: potential improvements

- Run tests with noise
- Investigate BFGS discrepancy
- Implement different optimization methods
- Circuit analysis

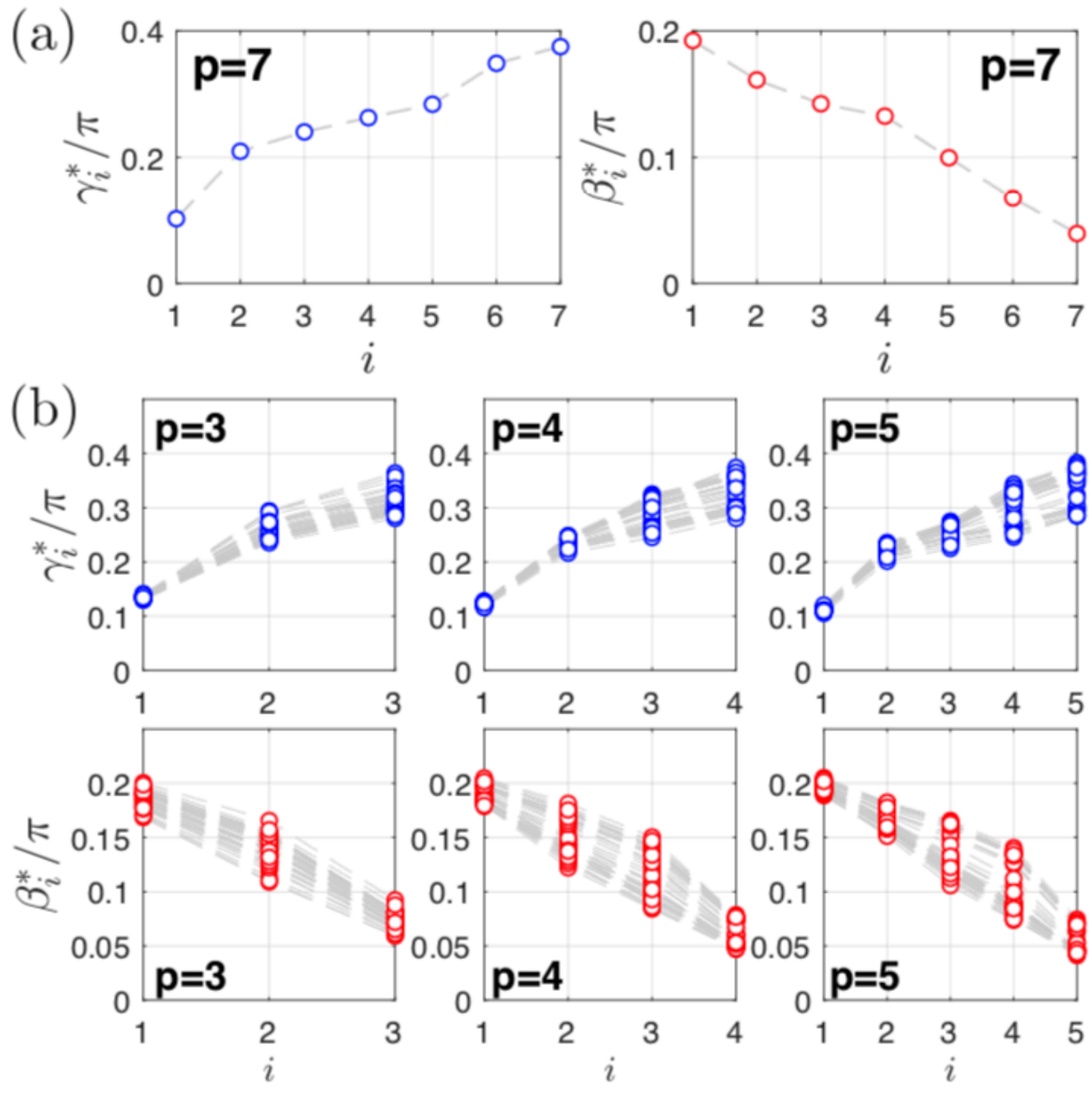
Optimization: initialization strategy

Goal: Evaluate INTERP strategy from Zhou et al (arxiv.org/abs/1812.01041)

My path:

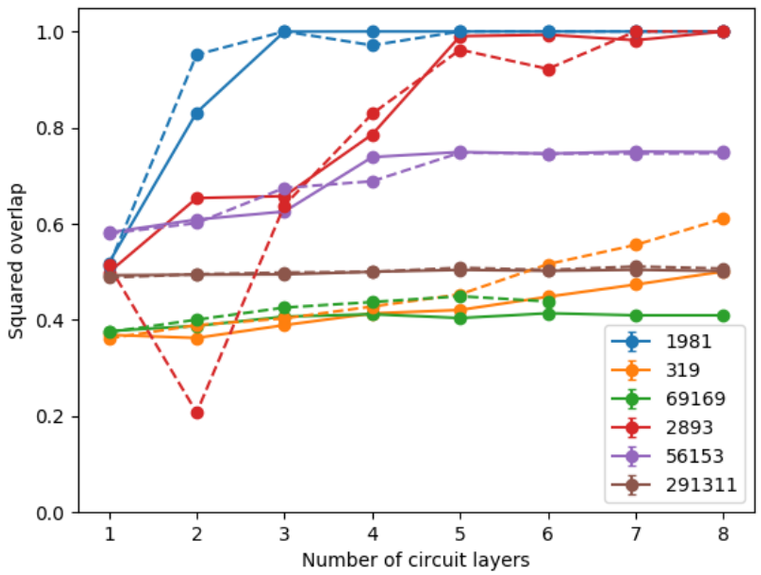
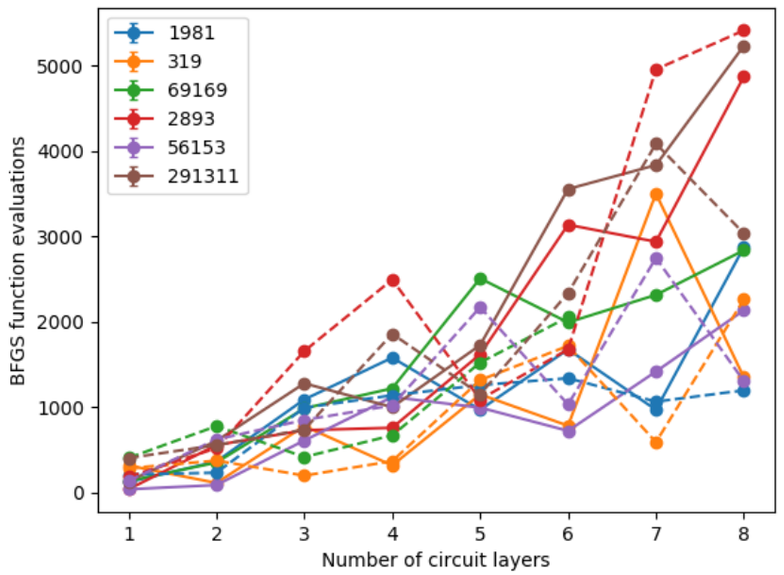
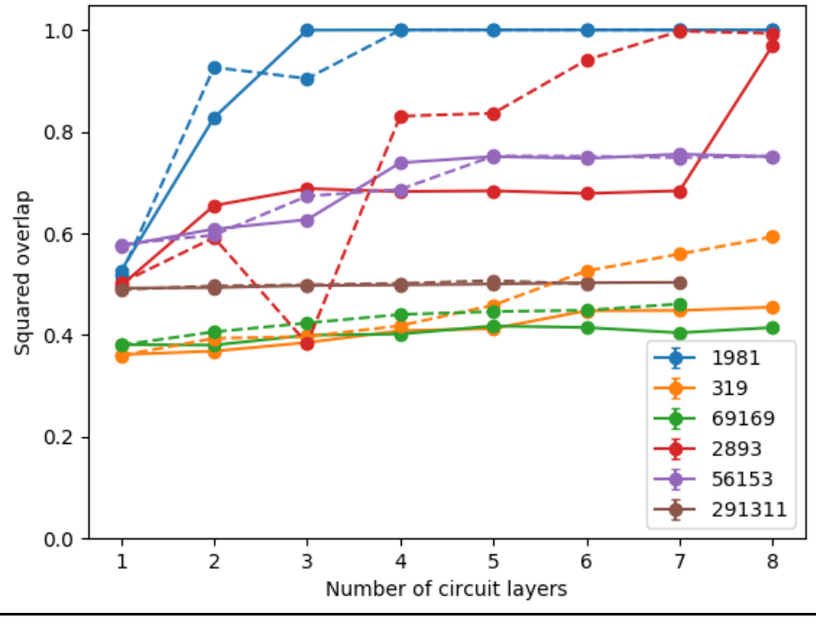
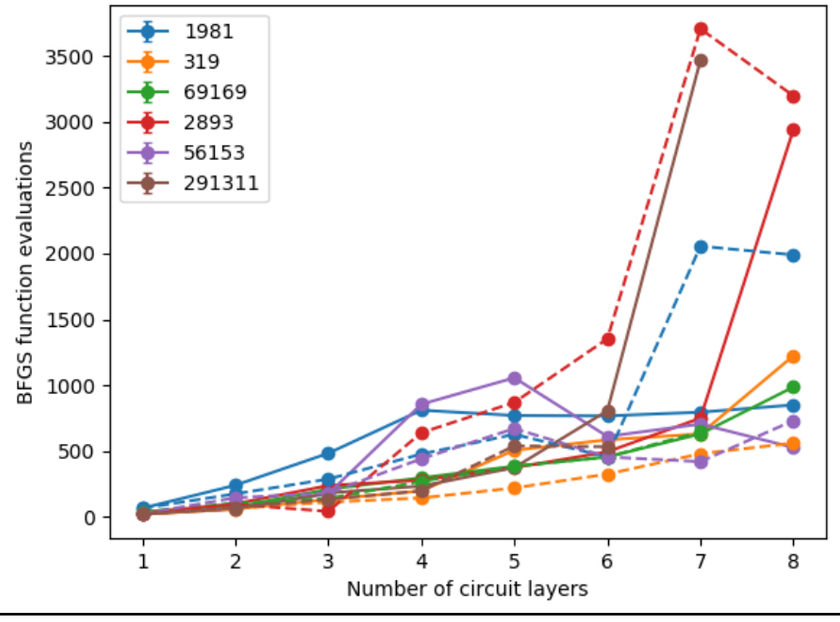
1. Implemented INTERP strategy
2. Produced results
3. Compared it with previous results

Optimization: initialization strategy



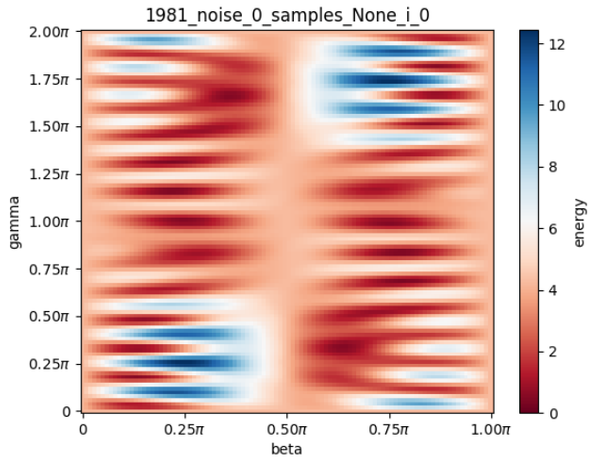
Source: Zhou et al

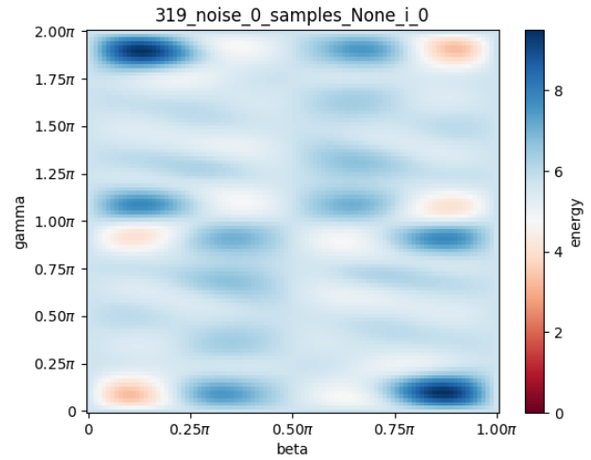
Results

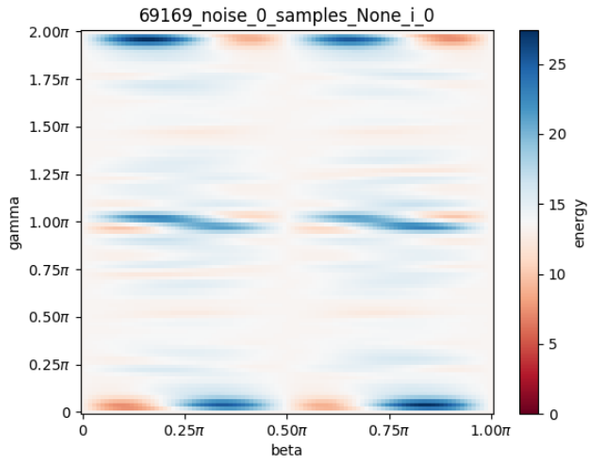
BFGS	BFGS
	
L-BFGS-B	L-BFGS-B
	

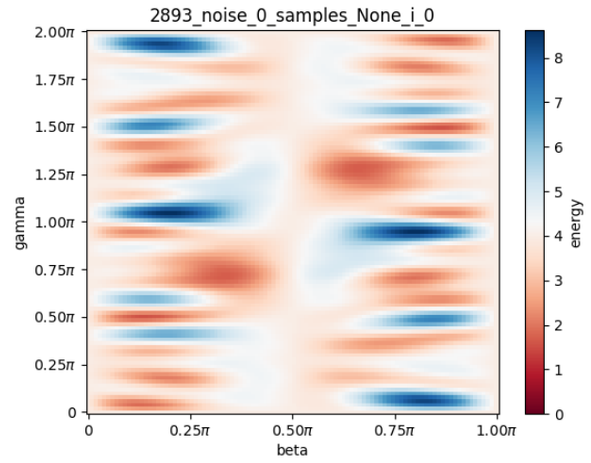
Visualization of optimization space

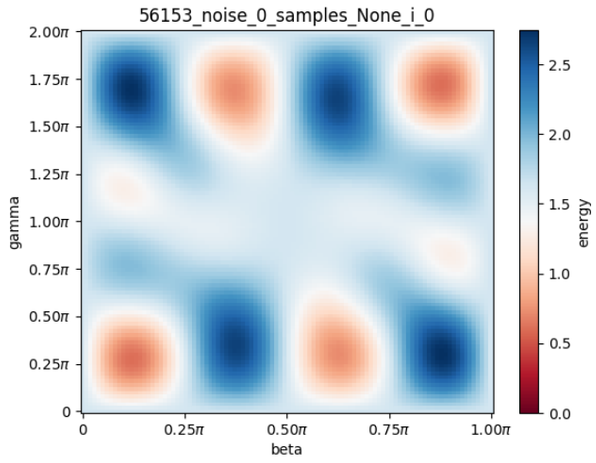
No noise, no sampling

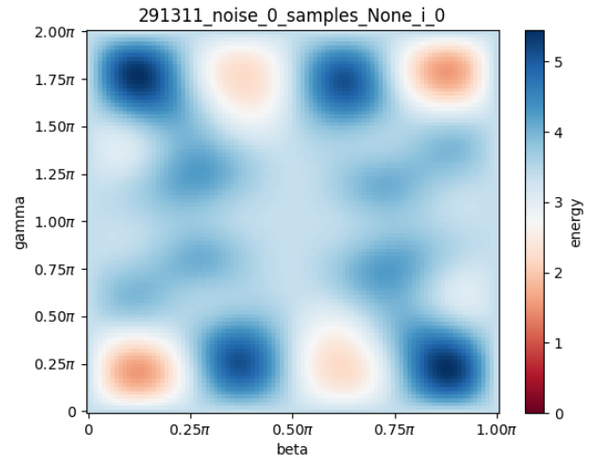




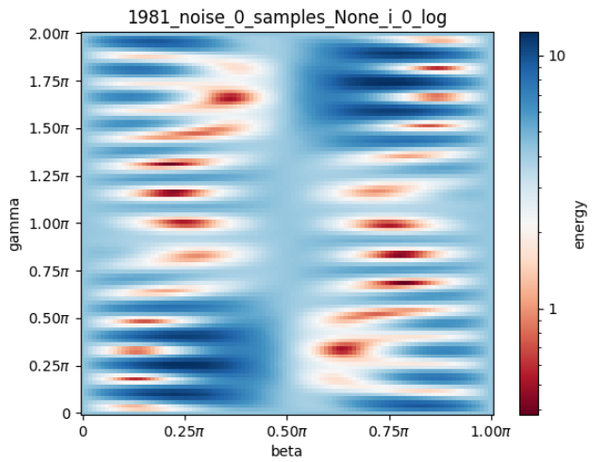


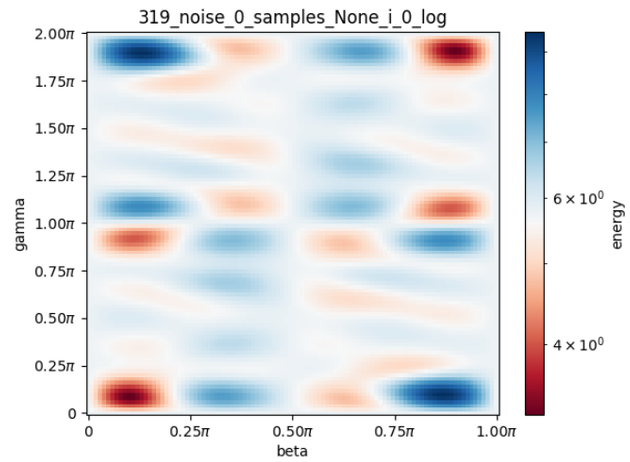


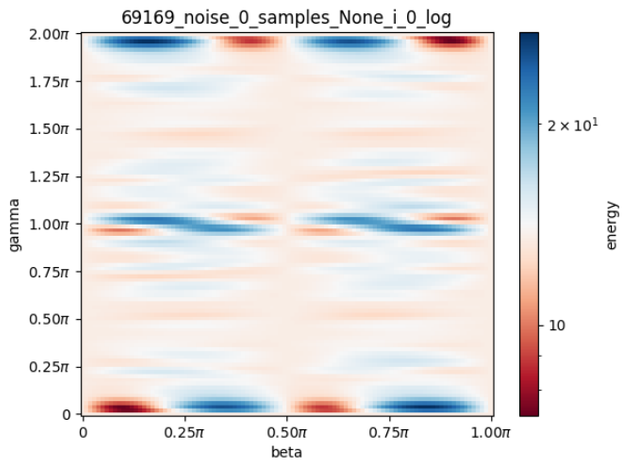


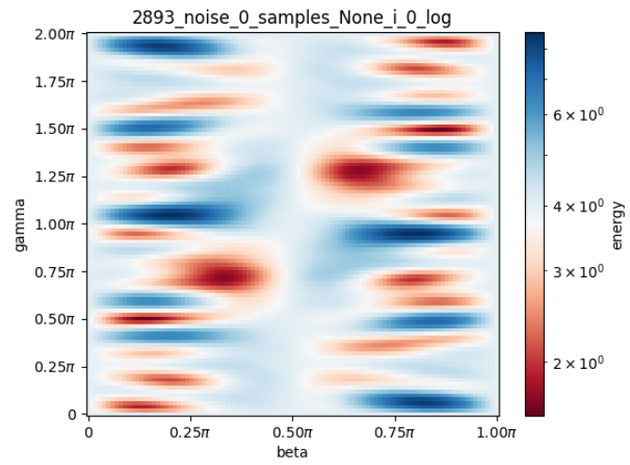


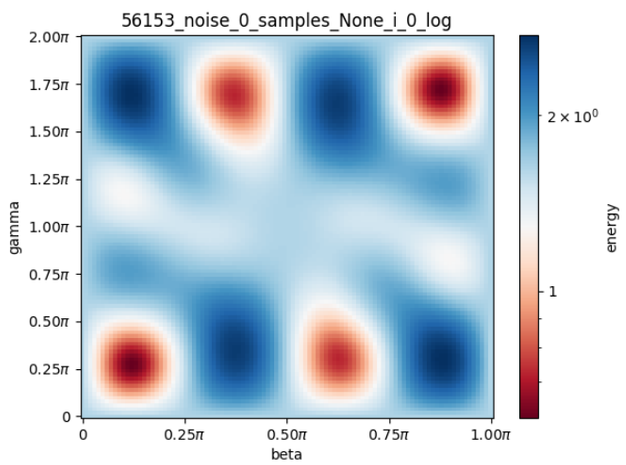
No noise, no sampling (log)

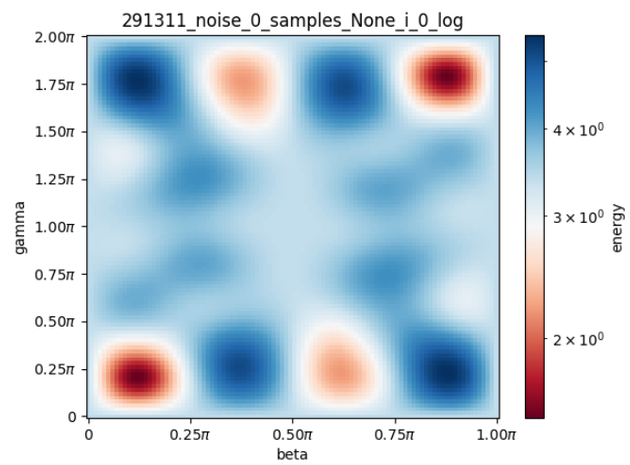






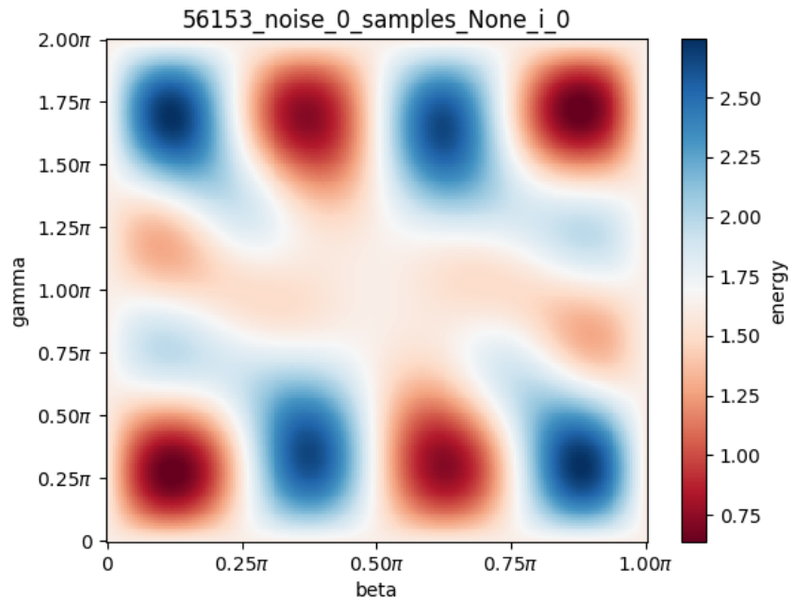




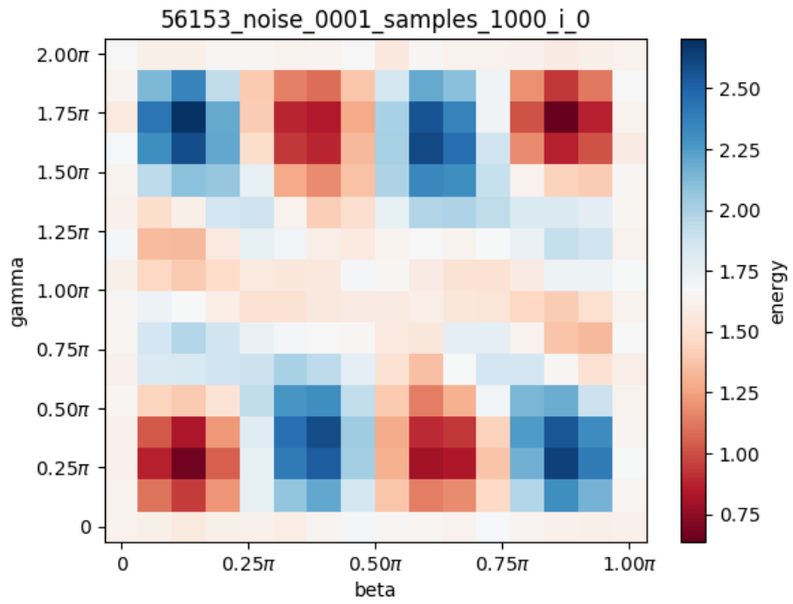


56153: noise

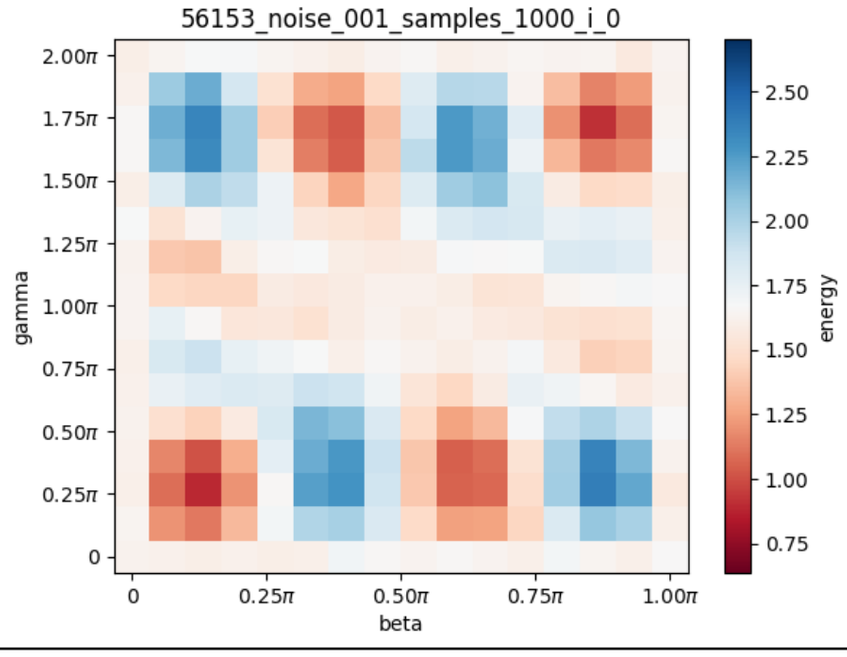
Reference



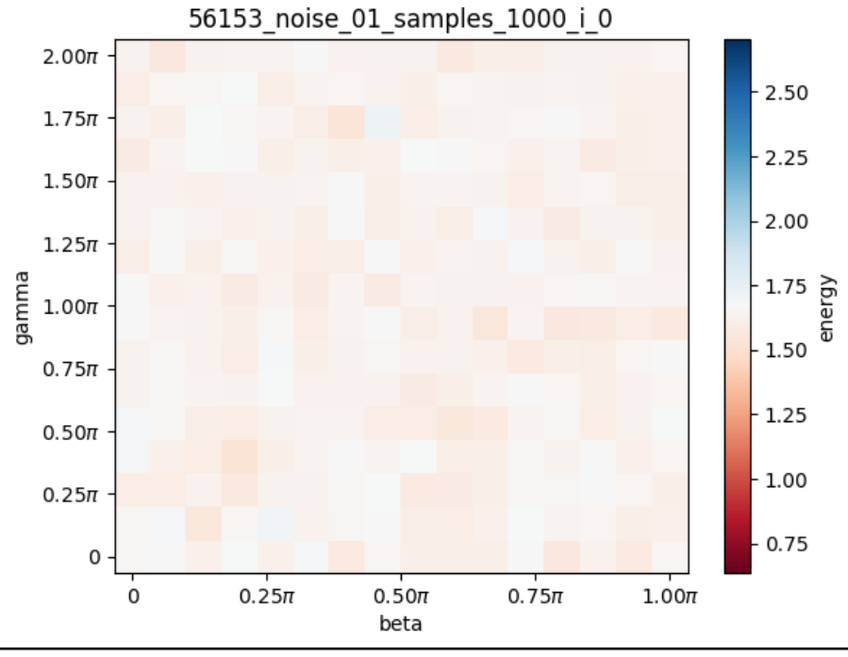
Noise: 1e-4, samples: 1e3



Noise: 1e-3, samples: 1e3

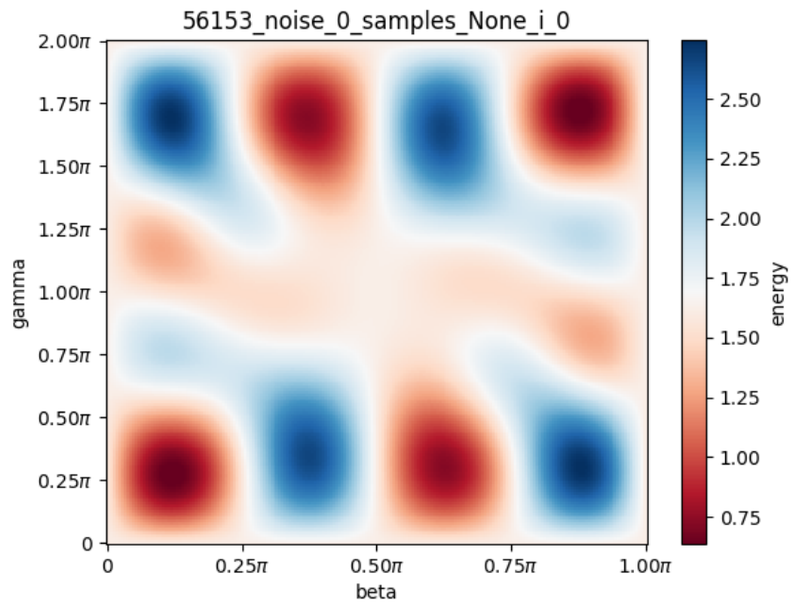


Noise: 1e-2, samples: 1e3

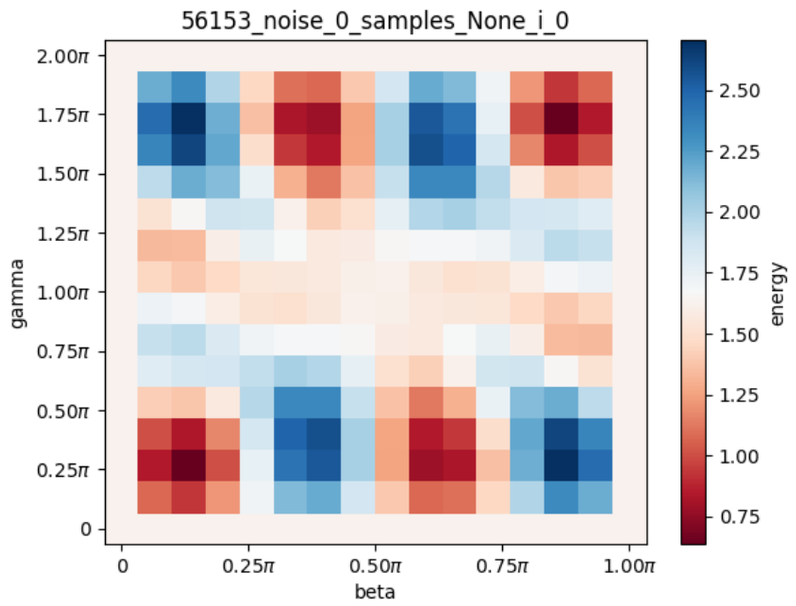


56153: sampling

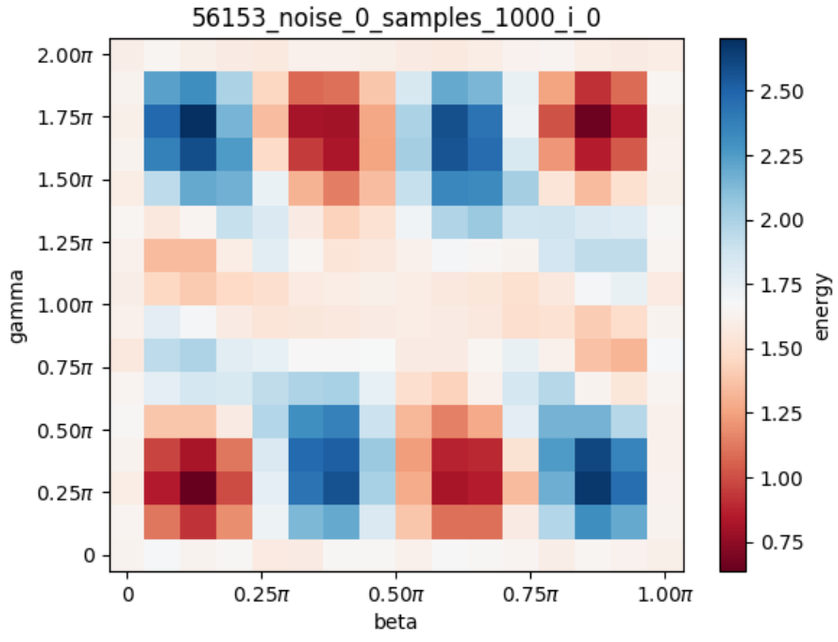
Reference



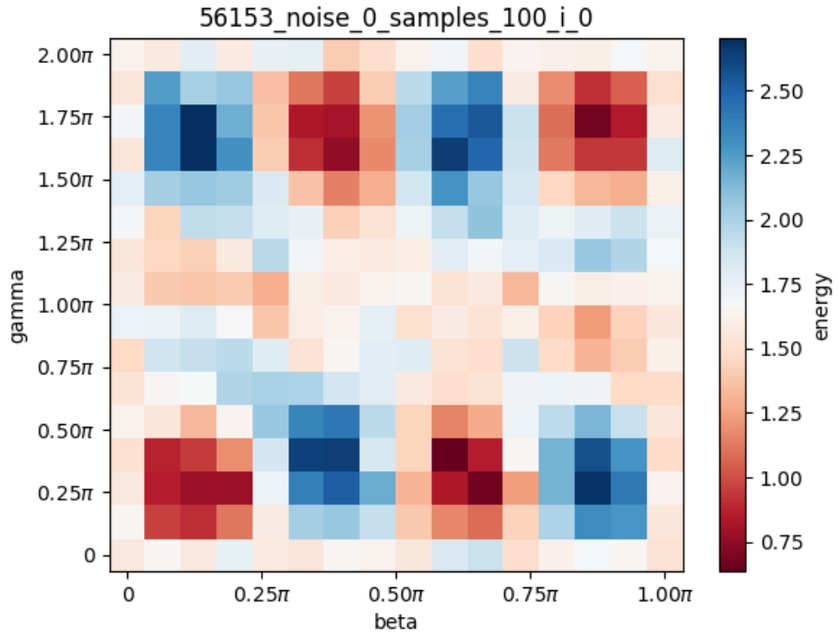
Noise: None, samples: None



Noise: None, samples: 1e3



Noise: None, samples: 1e2



What I've accomplished

- Implemented the core algorithm
- Improved on preprocessing part
- Implemented additional initialization strategy
- Learned a lot

Additional information

- Whole project took me about 80 hours to accomplish
- Out of this I’ve spent 5-8 hours on preparing and practicing this presentation.
- I’ve done it using only the paper as a reference.
- Everything is available on GitHub.
- I got the job :)

If this topic sounds interesting…

- http://arxiv.org/abs/1907.09631
- http://arxiv.org/abs/1910.04881
- http://arxiv.org/abs/1812.01041
- http://arxiv.org/abs/1907.04769
- http://arxiv.org/abs/2004.09002
- https://arxiv.org/abs/2004.04197

Thank you!

www.mustythoughts.com
Twitter: @mstechly