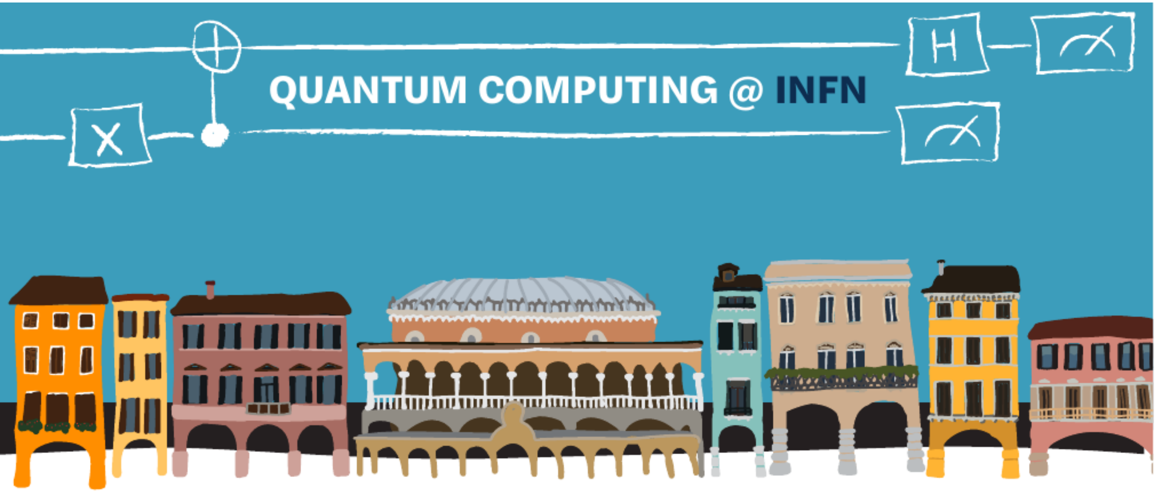
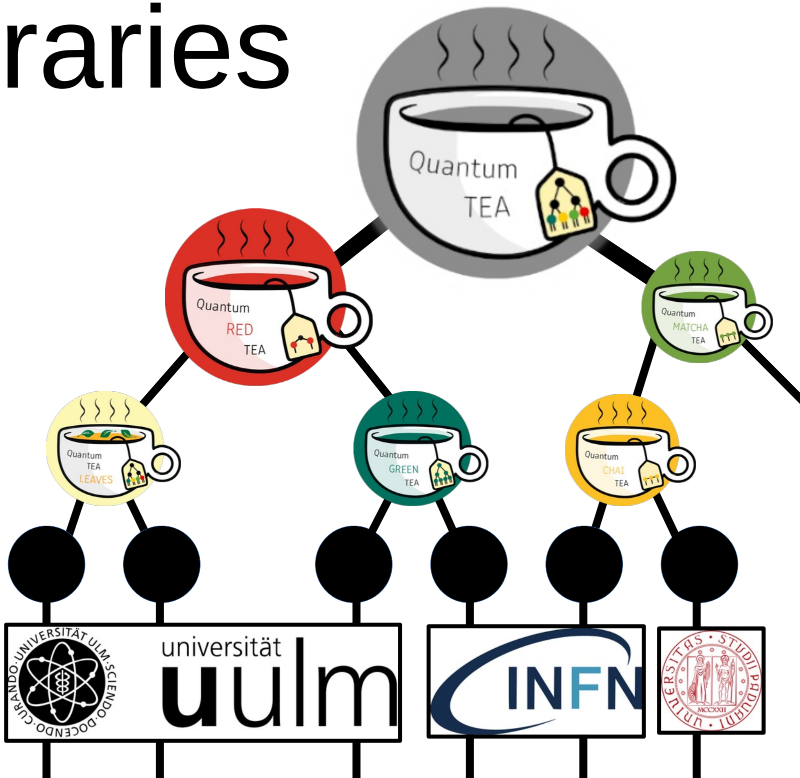
SLIDES

Boost Quantum TEA performance via flexible choices for numerical libraries

Daniel Jaschke, Marco Ballarin, Nora Reinić, Luka Pavešić, and Simone Montangero





Overview Quantum TEA

Quantum **T**ensor network **E**mulator **A**pplications

Can I just have tensor networks for quantum information?

How to emulate a digital quantum circuits?

+

Overview Quantum TEA

Quantum **T**ensor network **E**mulator **A**pplications

Can I just have tensor networks for quantum information?

How to emulate a digital quantum circuits? +

How to solve the Schrödinger equation? +

Can we do tensor network machine learning? … soon: +

QUANTUM COMPUTING @ INFN

Speaker: Daniel Jaschke
Boost Quantum TEA ...

universität ulm

Overview Quantum TEA

Quantum **T**ensor network **E**mulator **A**pplications

Can I just have tensor networks for quantum information?

Quantum TEA LEAVES

How to emulate a digital quantum circuits?

Quantum TEA LEAVES + Quantum MATCHA TEA

How to solve the Schrödinger equation?

Quantum TEA LEAVES + Quantum GREEN TEA

Can we do tensor network machine learning? … soon:

Quantum TEA LEAVES + Quantum CHAI TEA

Are they tasty?

Why do you need another tea flavor?

Quantum RED TEA

Applications Schrödinger Equation

Are they tasty?

14. Quantum circuit compilation with quantum computers
Davide Rattacaso (Istituto Nazionale di Fisica Nucleare)
10/30/24, 10:15 AM
Technological aspects
Compilation optimizes quantum algorithms performances on real-world quantum computers. To date, it is performed via

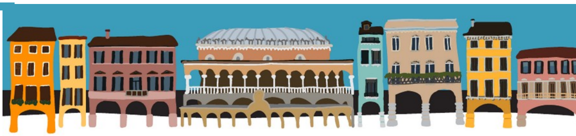
arXiv:2408.00077

21. Optimisation of ultrafast singlet fission in 1D rings towards unit efficiency
Francesco Campaioli (Istituto Nazionale di Fisica Nucleare)
10/31/24, 12:20 PM
Quantum Simulation
Singlet fission (SF) is an electronic transition that in the last decade has been under the spotlight for its applications in

Accepted in PRX Energy

25. Entanglement in finite-temperature Rydberg atom arrays
Nora Reinić (Istituto Nazionale di Fisica Nucleare)
10/31/24, 12:40 PM
Quantum Simulation
Tensor network methods are a family of numerical techniques that efficiently compress the information of quantum

Phys. Rev. Research 6, 033322

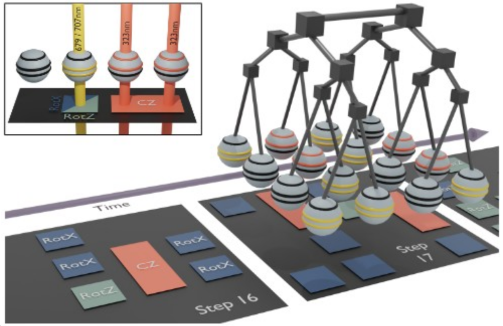


Applications Schrödinger Equation

Are they tasty?

Digital twin on pulse level

D. Jaschke et al., QST 9, 035055



14. Quantum circuit compilation with quantum computers
Davide Rattacaso (Istituto Nazionale di Fisica Nucleare)
10/30/24, 10:15 AM
Technological aspects
Compilation optimizes quantum algorithms performances on real-world quantum computers. To date, it is performed via

arXiv:2408.00077

21. Optimisation of ultrafast singlet fission in 1D rings towards unit efficiency
Francesco Campaioli (Istituto Nazionale di Fisica Nucleare)
10/31/24, 12:20 PM
Quantum Simulation
Singlet fission (SF) is an electronic transition that in the last decade has been under the spotlight for its applications in

Accepted in PRX Energy

25. Entanglement in finite-temperature Rydberg atom arrays
Nora Reinić (Istituto Nazionale di Fisica Nucleare)
10/31/24, 12:40 PM
Quantum Simulation
Tensor network methods are a family of numerical techniques that efficiently compress the information of quantum

Phys. Rev. Research 6, 033322

Applications quantum circuits

Are they tasty?

Quantum
the open journal for quantum science

PAPERS PERSPECTIVES

Entanglement entropy production in Quantum Neural Networks

Marco Ballarin[1,2,3], Stefano Mangini[1,4,5], Simone Montangero[2,3,6], Chiara Macchiavello[4,5,7], and Riccardo Mengoni[8]

Citation: Quantum 7, 1023 (2023).

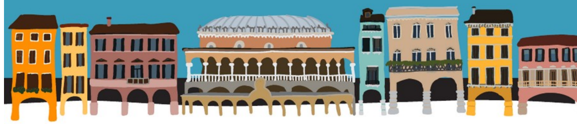
Quantum
the open journal for quantum science

PAPERS PERSPECTIVES

Digital quantum simulation of lattice fermion theories with local encoding

Marco Ballarin[1,2,3], Giovanni Cataldi[1,2,3], Giuseppe Magnifico[2,3,4], Daniel Jaschke[2,3,5], Marco Di Liberto[2,3,6], Ilaria Siloi[2,3,6], Simone Montangero[2,3,6], and Pietro Silvi[2,3,6]

Citation: Quantum 8, 1460 (2024).



Quantum red TEA

Why do you need another tea flavor?

How to integrate new technology in a library?

Can we use GPUs? … no, because ...

Can we use jax? … no, ...

Quantum red TEA

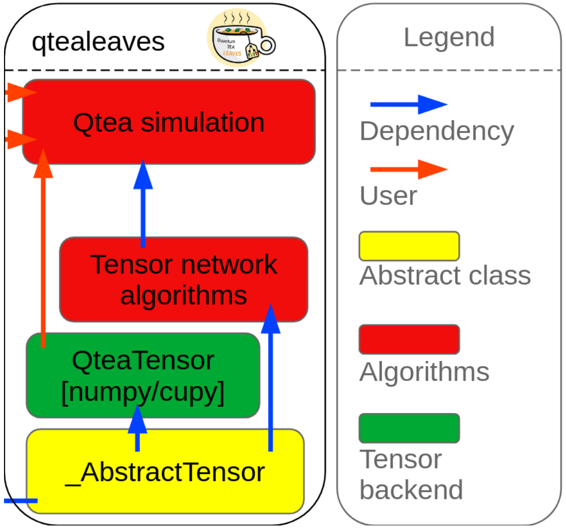
Why do you need another tea flavor?

How to integrate new technology in a library?

Can we use GPUs? … no, because ...

Can we use jax? … no, ...

Major update qtealeaves v1.0.0+



Quantum red TEA

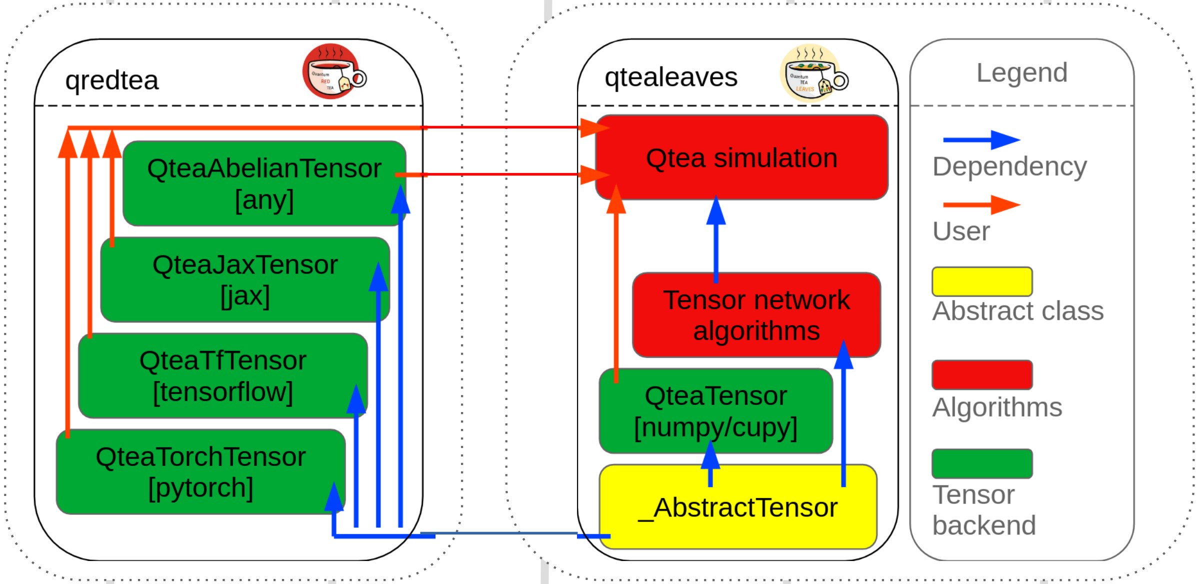
Why do you need another tea flavor?

How to integrate new technology in a library?

Can we use GPUs? … no, because ...

Can we use jax? … no, ...

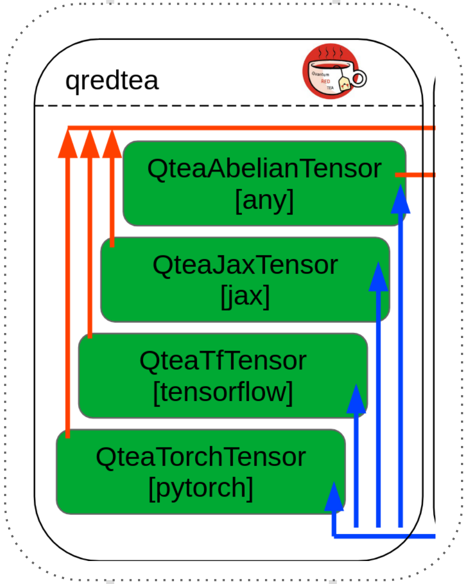
Major update qtealeaves v1.0.0+



Benefits of Quantum red TEA

Switch linear algebra backend in “one” line

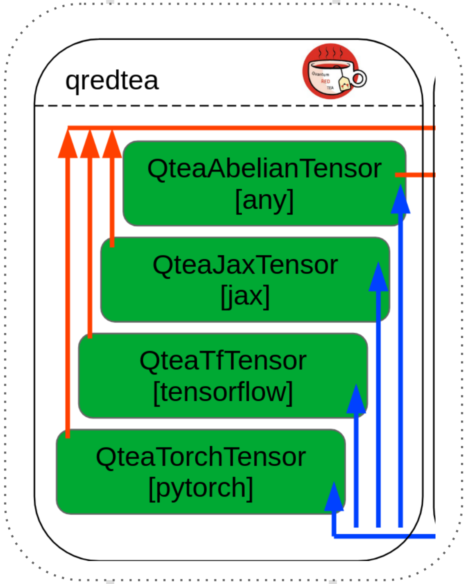
Switch between dense and symmetric tensors in a few lines



Benefits of Quantum red TEA

Switch linear algebra backend in “one” line

Switch between dense and symmetric tensors in a few lines



Automatic differentiation

Jitted function via jit (just-in-time compilation)

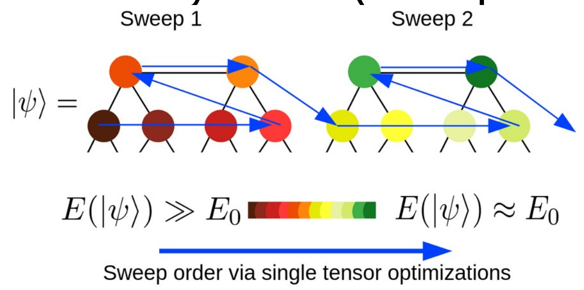
New hardware support
Example: TPUs

The benchmark: quantum Ising 2d

Quantum Ising model in 2d
Ground state search

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^x \sigma_j^x - g \sum_i \sigma_i^z$$

Key facts:
- 16x16 systems (256 qubits)

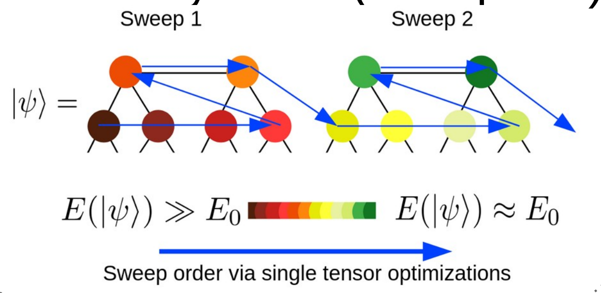


The benchmark: quantum Ising 2d

Quantum Ising model in 2d
Ground state search

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^x \sigma_j^x - g \sum_i \sigma_i^z$$

Key facts:
- 16x16 systems (256 qubits)



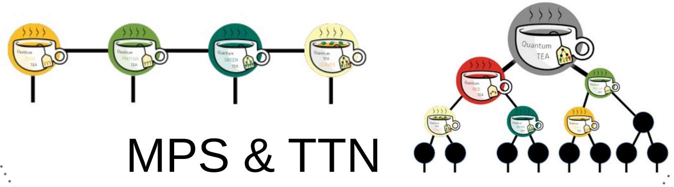
Leonardo (CINECA)

DCGP: dual-socket, 112 cores
Booster: nvidia A100 GPU

Computation challenges:
tensor contractions and linear algebra decompositions

Biggest tensor:
$\chi \times \chi \times \chi$

Tensor networks in a nutshell
- Choose tree tensor network (TTN) over matrix product states (MPS)
- Higher bond dimensions χ capture more entanglement (better approximation)



The ideas: what features to benchmark

Numpy-cupy versus torch versus tensorflow versus jax

Parallelization via CPU threads
(BLAS / LAPACK / EIGEN)

Parallelization via single GPU (CUDA)

Parallelization via single TPU
(Tensor processing unit)

The ideas: what features to benchmark

Numpy-cupy versus torch versus tensorflow versus jax

Parallelization via CPU threads
(BLAS / LAPACK / EIGEN)

“Mixed precision” approach

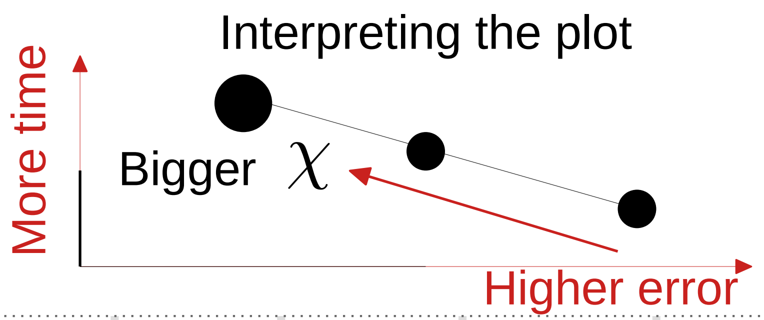
Skipping tensors

Parallelization via single GPU (CUDA)

Parallelization via single TPU
(Tensor processing unit)

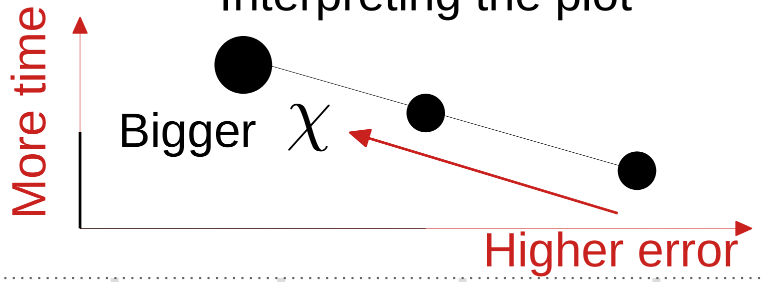
Enforce bond dimension to
fit hardware suggestions
(memory blocksize)

CPU benchmark in details (baseline)



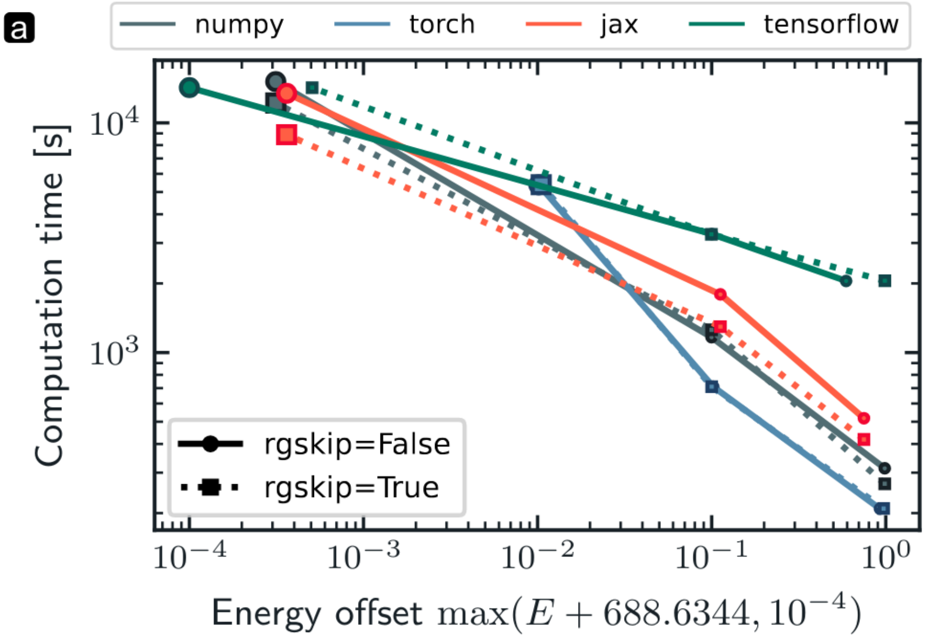
CPU benchmark in details (baseline)

Interpreting the plot

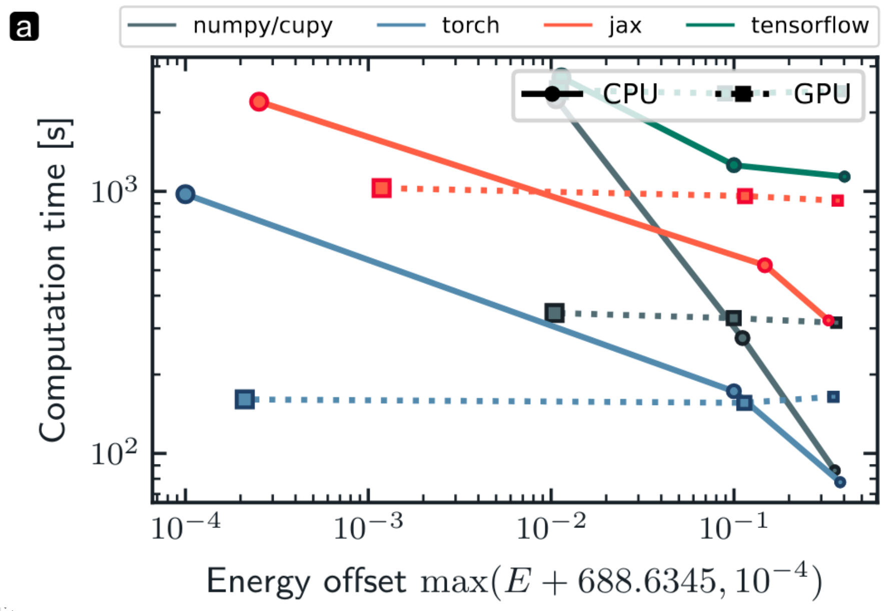


Backends for CPU mostly comparable

Precision fluctuates a bit within one order of magnitude



CPU versus GPU benchmark in details



Backends for GPU show differences now!

Check out the slope of the curve … the advantage of the GPU grows with the bond dimension.

Jax runtime includes compiling jit.

The winner in a nutshell: torch backend

Machine learning library supporting linear algebra similar to numpy

PyTorch

Torch **speedup** with all optimization over initial starting point without optimizations:

34x

Torch **speedup** GPU over best CPU:

2.76x

Conclusions and outlook

Speed-up is application dependent, but has already paid off

Torch has the best performance; most of our new projects use it now from the beginning

Qmatcha TEA benchmarks for all backends are ahead

Also: Quantum Circuits as a Service platform running via qiskit backend in CloudVeneto: www.quantumtea.it

Ref: DJ et al., arXiv 2409.03818

Quantum TEA

SPONSORED BY THE Federal Ministry of Education and Research

Backup slides

What is “Quantum TEA” not?

Quantum CHAI TEA

Quantum MATCHA TEA

Quantum GREEN TEA

Quantum TEA LEAVES

Quantum RED TEA

Quantum TEA

To [redacted]

Subject Re: [quantumtea] Quick solution for [instant tea powder]

Paragraph | Variable Width

Dear Diana,

I think you got the wrong idea about Quantum TEA ...

On 02.08.24 12:13, Diana wrote:

> Hi quantumtea,
>
> This might sound impossible but this is [Instant Tea Powder] made easy. Our high-quality instant tea powder is designed to ensure convenience and taste, all in one. You can enjoy the rich flavor and health benefits of tea in an instant.
>
> In addition to this, our product offers:
>
> - **High Solubility:** Dissolves quickly and completely in both hot and cold water.
> - **Rich Flavor:** Maintains the authentic taste of freshly brewed tea.
> - **Health Benefits:** Preserves the natural antioxidants and nutrients.
> - **Customizable:** We can adjust the content of ingredients based on your needs and produce in various forms such as capsules or pills.

What is "Quantum TEA"?

To [redacted]

Subject Re: [quantumtea] Quick solution for [instant tea powder]

Paragraph | Variable Width

Dear Diana,

I think you got the wrong idea about Quantum TEA ...

On 02.08.24 12:13, Diana wrote:

> Hi quantumtea,
>
> This might sound impossible but this is [Instant Tea Powder] made easy. Our high-quality instant tea powder is designed to ensure convenience and taste, all in one. You can enjoy the rich flavor and health benefits of tea in an instant.
>
> In addition to this, our product offers:
>
> - **High Solubility:** Dissolves quickly and completely in both hot and cold water.
> - **Rich Flavor:** Maintains the authentic taste of freshly brewed tea.
> - **Health Benefits:** Preserves the natural antioxidants and nutrients.
> - **Customizable:** We can adjust the content of ingredients based on your needs and produce in various forms such as capsules or pills.

Quantum
Tensor network **E**mulator **A**pplications

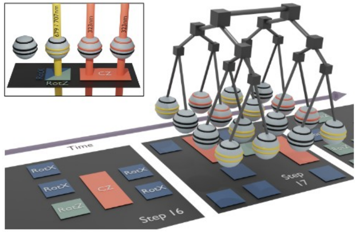
This might sound impossible, but we have ...
High "solutionability" for many models
Rich flavors possible, e.g., flavorful bosons
Health benefits as it is pure python
Customizable, e.g., with numpy or torch

Applications Schrödinger Equation

Are they tasty?

Digital twin on pulse level, D. Jaschke et al., QST 9, 035055

- Digital circuit translated to pulses
- Single-site addressing & scheduling
- Cross-talk of long-range interactions



Quantum-inspired integer factorization up to 100-bit RSA number in polynomial time,
M. Tesoro, …, DJ, et al., arxiv 2410.16355

- Uses tensor network for ground state problem
  and sampling


arXiv > cs > arXiv:2410.16355

Computer Science > Cryptography and Security

[Submitted on 21 Oct 2024]

Quantum inspired factorization up to 100-bit RSA number in polynomial time

Marco Tesoro, Ilaria Siloi, Daniel Jaschke, Giuseppe Magnifico, Simone Montangero

Mixed precision & optimization flags

Ground states converge in sweeps.

First sweep does not need high precision as we optimize a random guess.

Try sweep patterns with ...
- S: single real
- D: double real
- Z : double complex

b

Sweeps	torch
SSSSSS	627s -688.62502
SSSSDD	976s -688.63446
SSSDDD	1515s -688.62908
DDDDDD	2031s -688.63394
ZZZZZZ	5379s -688.62395

Mixed precision & optimization flags

Ground states converge in sweeps.

First sweep does not need high precision as we optimize a random guess.

Try sweep patterns with ...
  S: single real
  D: double real
  Z : double complex

b

Sweeps	torch
SSSSSS	627s -688.62502
SSSSDD	976s -688.63446
SSSDDD	1515s -688.62908
DDDDDD	2031s -688.63394
ZZZZZZ	5379s -688.62395

Approaches without success

Skip exact RG tensors … these tensors are small anyway, so skipping them does not save use enough for a significant speedup.

Enforcing bond dimensions did not have any effect unlike in the nvidia GEMM docs. Not even for jax with its jit-compilation.

Open science: source code and zenodo

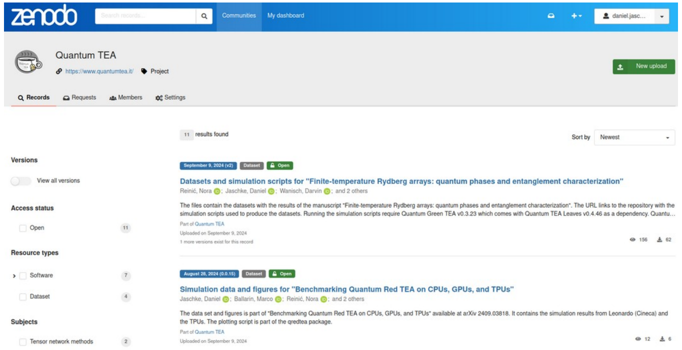
Source code lives on INFN platform baltig

https://baltig.infn.it/groups/quantum_tea/-/shared

Module on Leonardo (CINECA)

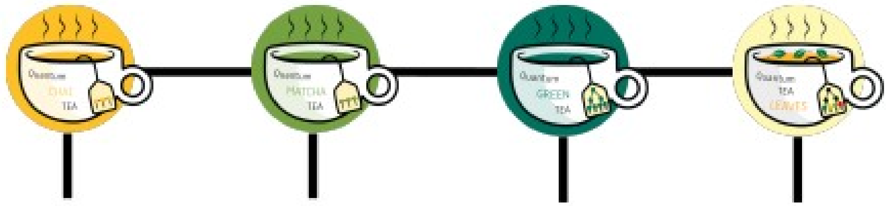
Available via pip install

Zenodo group for software and research examples



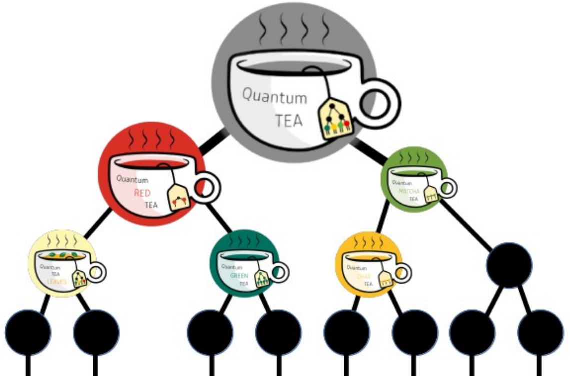
Tensor network ansätze

Matrix Product States (MPS)



Tree Tensor Networks (TTN)

2 $\log_2(N)$ links connect any two tensors

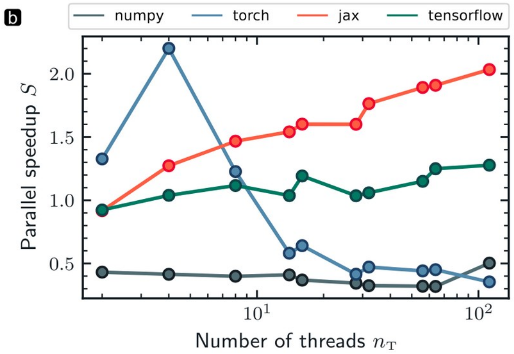


Difference (solver etc.)

CPU linear algebra

Numpy & torch: BLAS & LAPACK
Jax and tensorflow: EIGEN

CPU threading (not scalable)



Lanczos

Numpy: Arpack
Torch, jax, tensorflow: qtea eigensolver

Decompositions

Tuned based on hardware / device

Plotting logic: error plot without exact result

b

Sweeps	numpy	torch	jax	tensorflow
SSSSSS	t = 977s E = -688.62346	627s -688.62502	1365s -688.60703	1914s -688.59615
SSSSDD	t = 2249s E = -688.62380	976s -688.63446	2198s -688.63420	2721s -688.62307
SSSDDD	t = 3249s E = -688.62382	1515s -688.62908	2352s -688.63421	3101s -688.63315
DDDDDD	t = 5046s E = -688.63382	2031s -688.63394	3394s -688.62370	4050s -688.63370
ZZZZZZ	t = 12207s E = -688.63413	5379s -688.62395	8881s -688.63408	14233s -688.63394

Set error of the best simulation to a value ϵ .

Other simulations calculate their error towards the best simulation.

Generalization to Abelian symmetries

Challenge: symmetric tensors are basically sparse tensors formed of smaller dense tensors.

Our symmetric tensors support all methods for a ground state search.

The dense tensors inside the symmetric Tensors are numpy, torch, tensorflow or jax.

Physical systems conserve symmetries: Z2

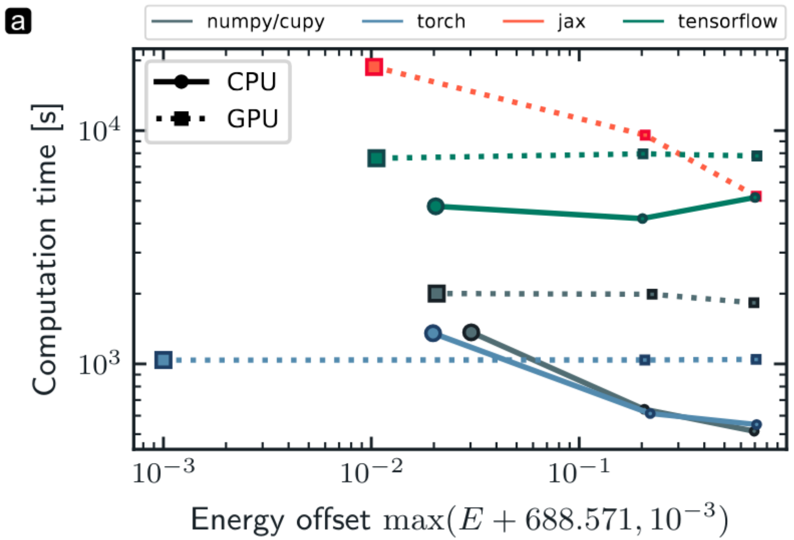
Generalization to Abelian symmetries

Challenge: symmetric tensors are basically sparse tensors formed of smaller dense tensors.

Our symmetric tensors support all methods for a ground state search.

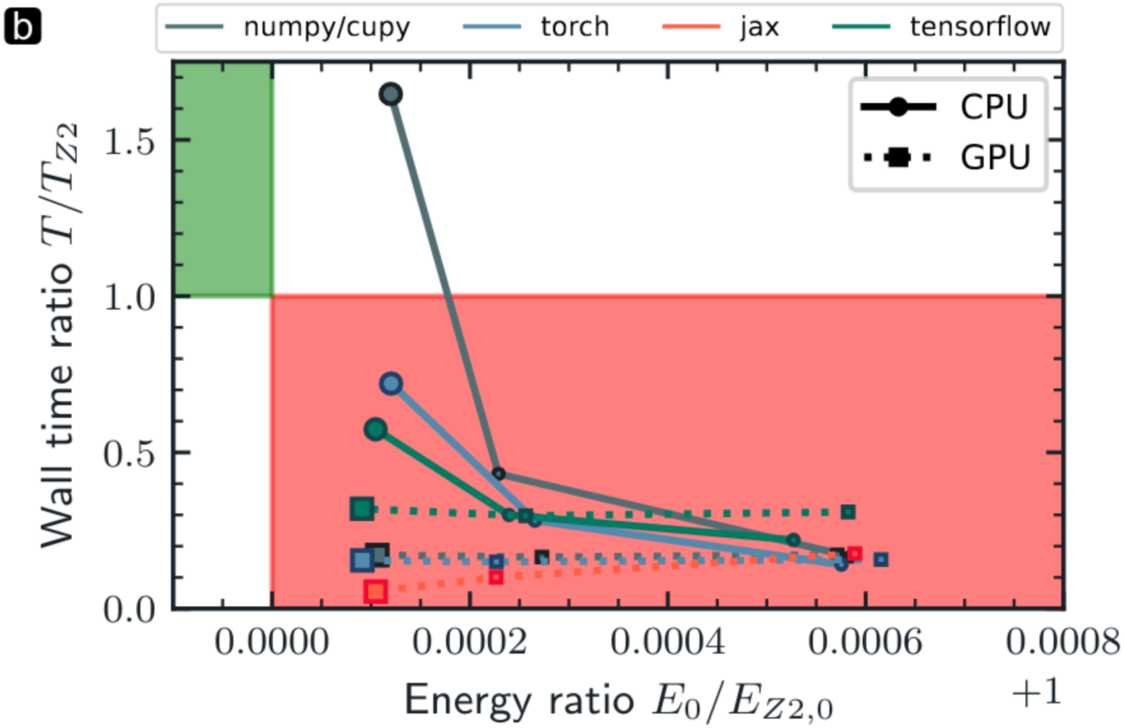
The dense tensors inside the symmetric Tensors are numpy, torch, tensorflow or jax.

Physical systems conserve symmetries: Z2



Break-even for GPU at larger bond dimension

Symmetries versus no symmetries



TPU data (jax only)

a

Bond dimension	CPU	XLA	XLA + tile=128
$\chi = 32$	t=1065s E=-688.51693	1131s -687.98895	n.a. n.a.
$\chi = 64$	t=1823s E=-688.61092	1180s -687.06878	1625s -684.32439
$\chi = 128$	t=4692s E=-688.57309	1244s -686.84771	1701s -668.38448