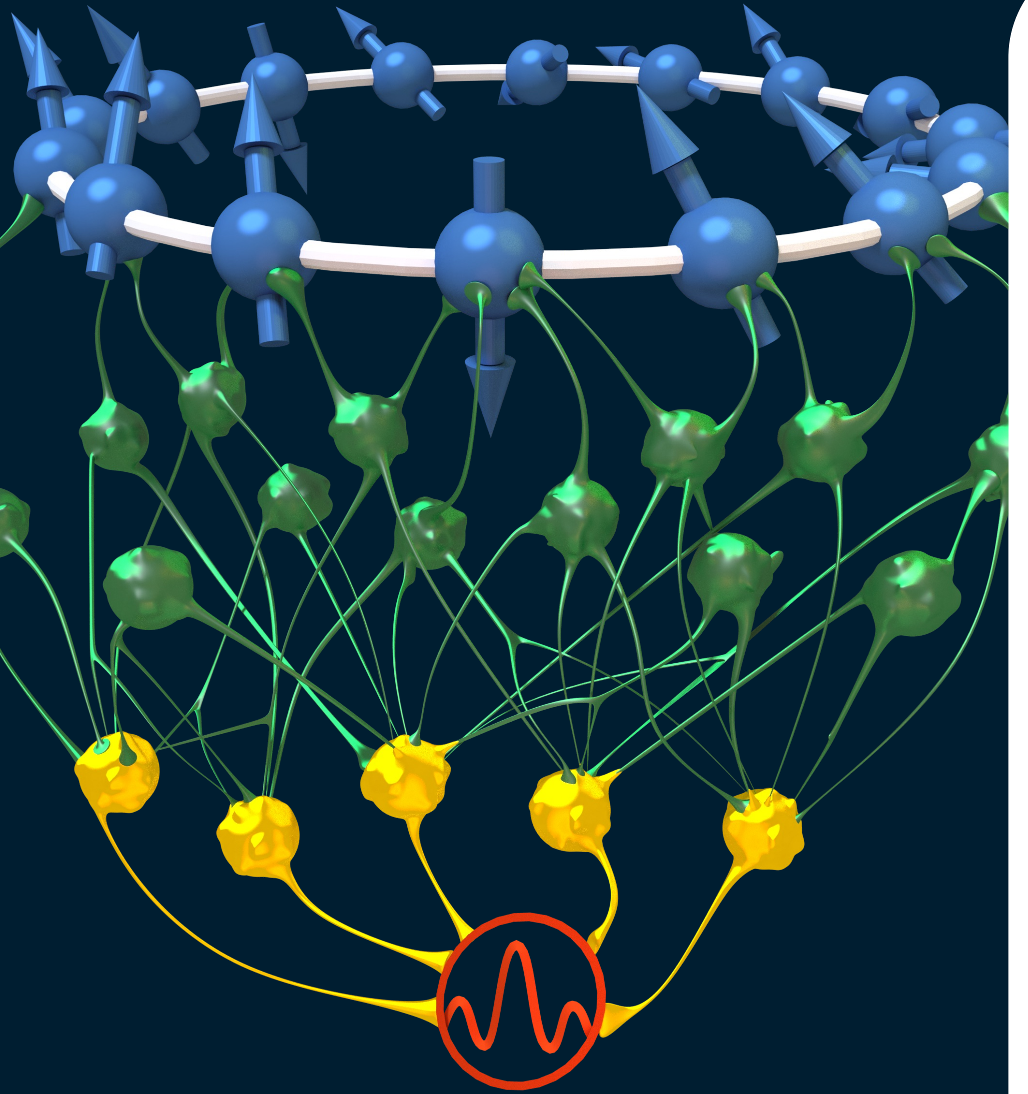
Filippo Vicentini

Neural Quantum States for Finite Temperature and Open Systems

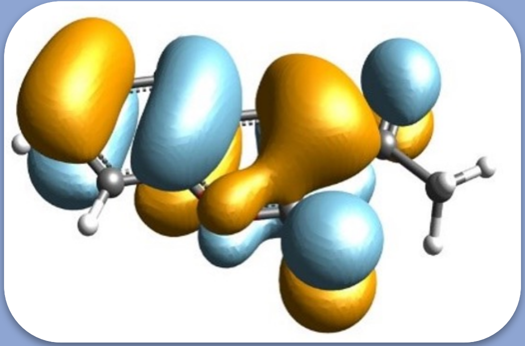


College de France

Paris, 9 May 2023

The Quantum Many-Body Problem

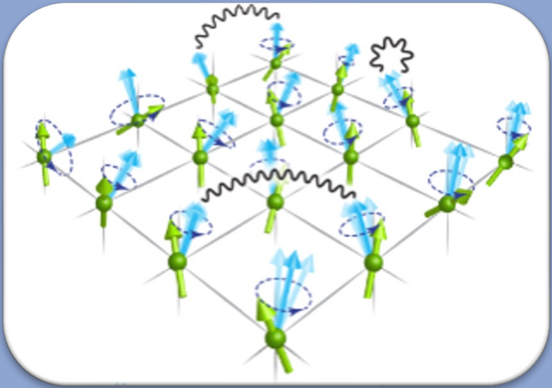
Quantum Chemistry



Schroedinger's Equation

$$\frac{d\,|\psi(t)\rangle}{dt} = -i\hat{H}\,|\psi(t)\rangle$$

Lattice Models



Memory Complexity

Memory required to store the wave-function of a quantum system

Spins	Hilbert Size	Memory
10	10^3	~16 kB
20	10^6	~ 16 MB
30	10^9	~ 16GB
40	10^{12}	~16 TB
50	10^{15}	~ 16 EB

Computational Complexity

of operations to compute exp. values

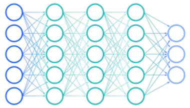
$$\langle\hat{H}\rangle = \sum_{\sigma,\eta} \psi^\star_\sigma H_{\sigma,\eta}\psi_\eta$$

The Quantum Many-Body Problem

Variational Ansatz

Data structure / Compression

$$\psi : \{\pm 1\}^{\otimes N} \rightarrow \mathcal{C}$$

$$\psi \approx$$ 

Memory Complexity

Memory required to store the wave-function of a quantum system

Spins	Hilbert Size	Memory
10	10^3	~16 kB
20	10^6	~ 16 MB
30	10^9	~ 16GB
40	10^{12}	~16 TB
50	10^{15}	~ 16 EB

Monte-Carlo Sampling

of operations to compute exp. values

$$\langle \hat{H} \rangle = \mathbb{E}\left[\frac{\langle \sigma | \hat{H} | \psi \rangle}{\langle \sigma | \psi \rangle} \right]$$

Computational Complexity

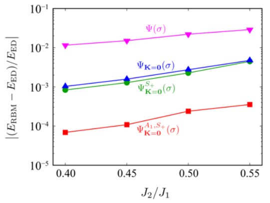
of operations to compute exp. values

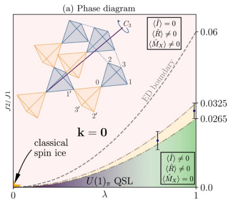
$$\langle \hat{H} \rangle = \sum_{\sigma,\eta} \psi^\star_\sigma H_{\sigma,\eta} \psi_\eta$$

Neural Quantum States: Applications

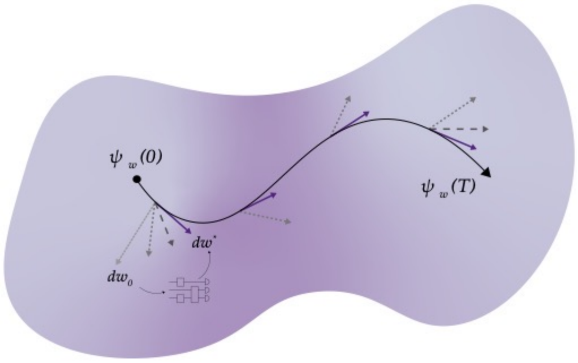
Ground States

Ising [Carleo and Troyer, Science (2017)]
J1-J2 + Symm [Nomura, JPCM **33** (2021)]
Heisenberg Pyrocl. [Astrakhantsev et Al, PRX **11** (2021)]
Continuous [Pescia et Al, PRR **4** (2022)]
[Lovato et Al, PRR **4** (2022)
Chemistry [Zhao et Al, 2208.05637]
[Choo et Al, Nat Comms 11 (2020)]
Machine Precision [Chen and Heyl, ArXiv:2302.01941 (23)]





Unitary Dynamics



$$\frac{dW}{dt} = -iS\nabla_W E(W)$$

[Carleo et Al, PRX (2017)]
[Yuan et Al, Quantum (2019)]
[Barison, F.V. et Al., Quantum (2021)]

Excited States

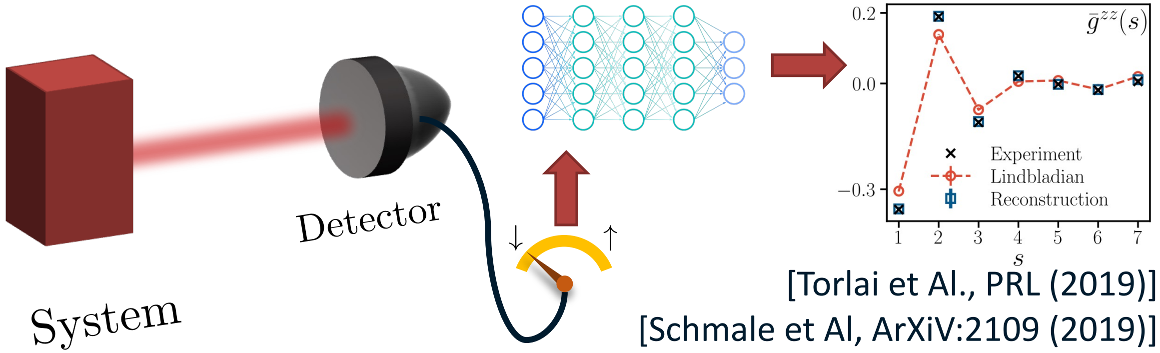
Group-Equivariant Neural Networks

[Cohen and Welling, ICML (2016)]
[Roth et Al, ArXiV:2109 (2021)]
[Szabo et Al, ArXiV:2203 (2022)]

Representation Sampling

[Choo et Al, PRL 121 (2018)]
[Nomura et Al, ArXiV:2009 (2020)]
[Szabo et Al, ArXiV:2203 (2022)]

Quantum State Reconstruction



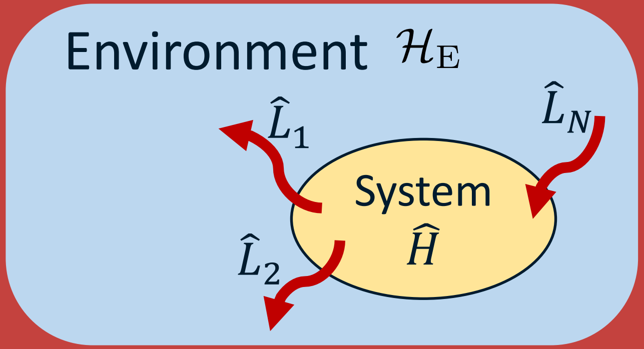
[Torlai et Al., PRL (2019)]
[Schmale et Al, ArXiV:2109 (2019)]
[Melkani et Al, PRA 102 (2020)]

Beyond Pure states & Equilibrium

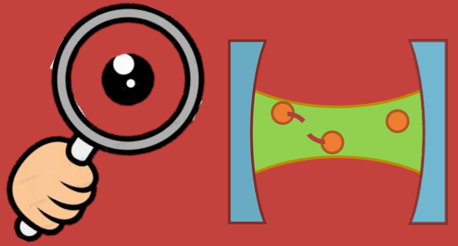
Finite Temperature

$$\hat{\rho}_\beta = e^{-\beta \hat{H}}$$

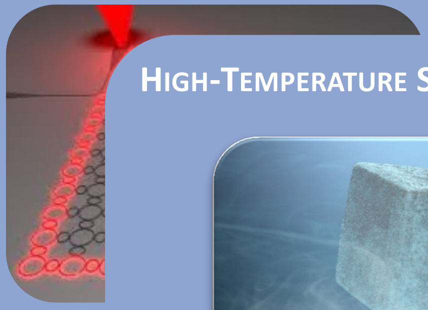
Open Systems



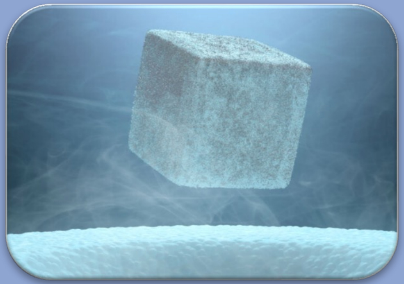
Measurement Backaction



Topological Physics



High-Temperature Superconductors



Array of resonators

[Koch et Al, PRA 82 (2010)]

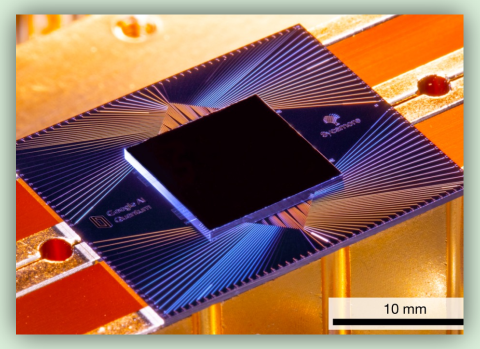
Dissipative Phase Transitions

[Minganti et Al, PRA 98 (2018)]
[Vicentini et Al, PRA 97 (2018)]

Cavity Quantum Chemistry

[Flick et Al, Nanophot. (2018)]
[Rubio et Al, PNAS (2019)]
[Vidal et Al, Science (2021)]

Quantum Technologies



F. Arute et Al.
Nature 574, 505 (2019)

Stochastic Unravelings

[Molmer et Al, JOSA B (1993)]
[Daley AIP **63** (2014)]
[Bartolo et Al, EPJ 226 (2017)]
[Biella et Al, Quantum 5 (2021)]

Measurement Induced phase transitions

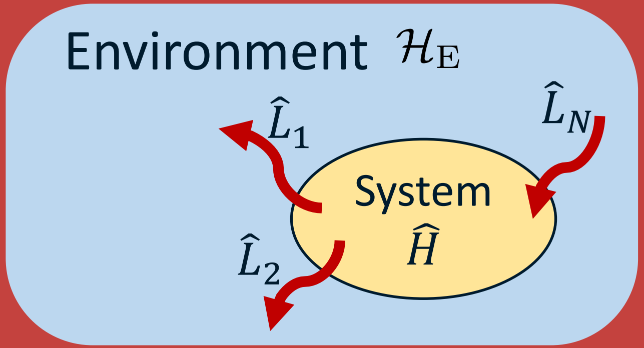
[Skinner et Al, PRX **9** (2019)]
[Turkeshi et Al, PRB **103** (2021)]

Beyond Pure states & Equilibrium

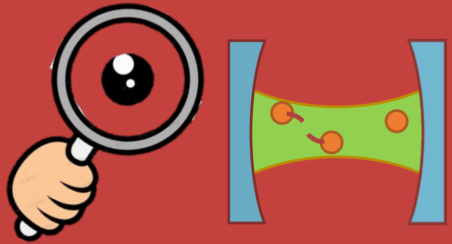
Finite Temperature

$$\hat{\rho}_\beta = e^{-\beta \hat{H}}$$

Open Systems



Measurement Backaction

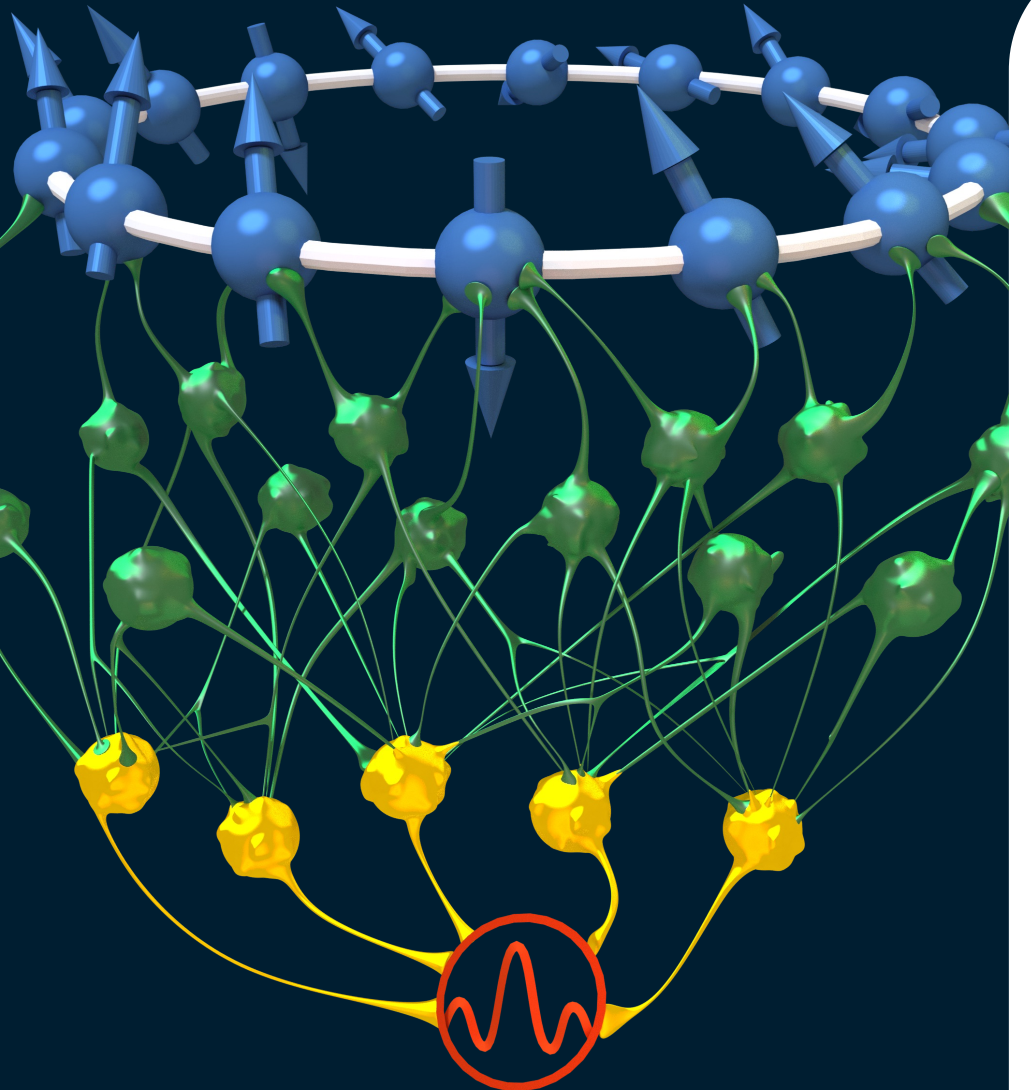


www.netket.org

NetKet: The Machine Learning toolbox for Quantum Many-Body Physics

Filippo Vicentini[1,2*], Damian Hofmann[3], Attila Szabó[4,5], Dian Wu[1,2], Christopher Roth[6], Clemens Giuliani[1,2], Gabriel Pescia[1,2], Jannes Nys[1,2], Vladimir Vargas-Calderón[7], Nikita Astrakhantsev[8] and Giuseppe Carleo[1,2]

Describing Mixed States



Density Matrices

$$\hat{\rho} = \sum_i p_i \left|\psi_i\right\rangle\!\left\langle\psi_i\right|$$

Requirements

- Hermitian
- Positive Definite
- ~~Trace 1~~

Mixed Neural Quantum States

Pure-state Expansion

$$\hat{\rho} = \sum_i p_i |\psi_i\rangle \langle\psi_i|$$

Used in few works in Quantum State Reconstruction by Nori et Al, Finite-temperature unravelings

[Melkani, PRA **102** (2020)]
[Hendry et Al, PRB **106** (2022)]

✅ Physical Parametrisation

❌ Expensive for high entropy

Pauli-Z basis

$$\hat{\rho} = \sum_{\sigma,\eta} \exp[\log\rho(\sigma,\eta)] |\sigma\rangle \langle\eta|$$

Good results in Steady-State, Tomography

[Torlai et Al, PRL **120** (2018)]
[Vicentini et Al, PRL **122** (2019)]
[Vicentini et Al, 2206.13488 (2022)]

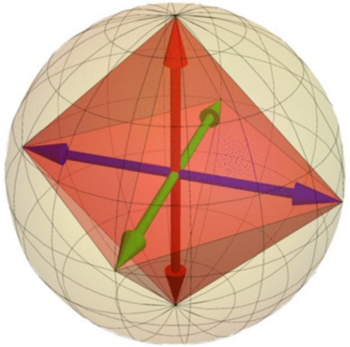
🤔 Properties depend on NN architecture

POVM basis

$$\hat{\rho} = \sum_a p(a) \hat{K}_a$$

Good results for tomography, very good results for dynamics

[Carrasquilla et Al, Nat Phys (2020)]
[Schmitt et Al, PRL **127** (2021)]
[Schmale et Al, NPJ QI **8** (2022)]

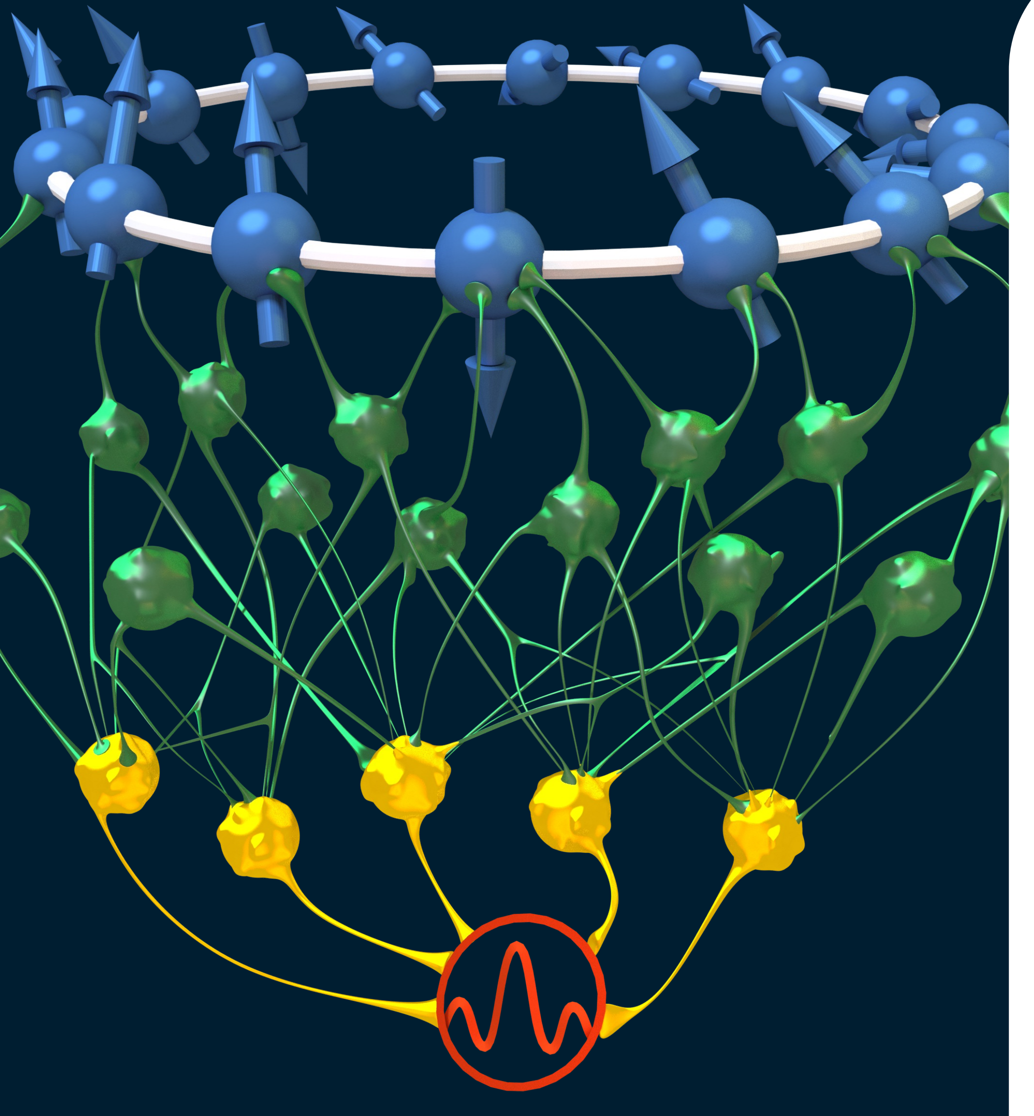


✅ Efficient

❌ Overcomplete basis

❌ Unphysical

Finite Temperature Caculations



Using "Pure" NQS

$$\frac{e^{-\beta \hat{H}}}{\mathcal{Z}_\beta}$$

Typical Thermal States

$$\hat{\rho}_\beta = e^{-\beta \hat{H}}$$

$$\hat{\rho}_\beta = \sum_\sigma e^{-\frac{\beta}{2}\hat{H}} |\sigma\rangle\langle\sigma| e^{-\frac{\beta}{2}\hat{H}}$$

$$= \sum_\sigma |\psi_{\theta^\sigma_\beta}\rangle\langle\psi_{\theta^\sigma_\beta}|$$

Step 1: «Prepare» the NQS for the initial State

Minimise the distance $\theta^\sigma_0 \mid \min_\theta \left[\frac{\| |\psi_\theta\rangle - |\sigma\rangle \|^2}{\langle\psi_\theta|\psi_\theta\rangle} \right]$

Can be done efficiently with the Stochastic estimator $\mathbb{E}_{\sigma\sim|\psi^2_\theta|} \left[|\psi(\eta) - \delta_{\sigma,\eta}|^2 \right]$

Step 2: «Simulate» the imaginary time Dynamics

At every step, find the change of parameters $\delta\theta$ approximating the Imaginary time-step $\delta\beta$, minimizing the distance

$$\min_{\delta\theta} \mathcal{D}(|\Psi_{\theta_\beta+\delta\theta}\rangle, (\mathbb{I} - \delta\beta\hat{H})|\Psi_{\theta(t)}\rangle)$$

This yields a differential Equation for the parameters

$$\frac{d\theta_\beta}{d\beta} = \bar{\bar{S}}^{-1}\vec{F}$$

[Yuan et Al, Quantum **3** 191 (2019)]

Typical Thermal States

$$\hat{\rho}_\beta = e^{-\beta \hat{H}}$$

$$\hat{\rho}_\beta = \sum_\sigma e^{-\frac{\beta}{2}\hat{H}} |\sigma\rangle\langle\sigma| e^{-\frac{\beta}{2}\hat{H}}$$

$$= \sum_\sigma |\psi_{\theta_\beta^\sigma}\rangle\langle\psi_{\theta_\beta^\sigma}|$$

[Hendry et Al, PRB **106**, 165111 (2022)]

Step 3: Computing Expectation Values

$$\langle\hat{A}\rangle = \frac{\mathrm{Tr}\left[\hat{A}\rho_\beta\right]}{\mathrm{Tr}[\rho_\beta]}$$

Typical Thermal States

$$\hat{\rho}_\beta = e^{-\beta \hat{H}}$$

$$\hat{\rho}_\beta = \sum_\sigma e^{-\frac{\beta}{2}\hat{H}} |\sigma\rangle\langle\sigma| e^{-\frac{\beta}{2}\hat{H}}$$

$$= \sum_\sigma |\psi_{\theta^\sigma_\beta}\rangle\langle\psi_{\theta^\sigma_\beta}|$$

[Hendry et Al, PRB **106**, 165111 (2022)]

Step 3: Computing Expectation Values

$$\langle\hat{A}\rangle = \frac{\mathrm{Tr}\left[\hat{A}\rho_\beta\right]}{\mathrm{Tr}[\rho_\beta]} = \frac{\sum_\sigma \langle\psi_{\theta^\sigma_\beta}|\hat{A}|\psi_{\theta^\sigma_\beta}\rangle}{\mathcal{Z}_\beta} = \sum_\sigma \frac{\langle\psi_{\theta^\sigma_\beta}|\psi_{\theta^\sigma_\beta}\rangle}{\mathcal{Z}_\beta} \frac{\langle\psi_{\theta^\sigma_\beta}|\hat{A}|\psi_{\theta^\sigma_\beta}\rangle}{\langle\psi_{\theta^\sigma_\beta}|\psi_{\theta^\sigma_\beta}\rangle}$$

Problem: how can we know the norm of the pure-state?

We keep track of its evolution $$\partial_\beta \langle\psi_{\theta^\sigma_\beta}|\psi_{\theta^\sigma_\beta}\rangle = -\langle\psi_{\theta^\sigma_\beta}|\hat{H}|\psi_{\theta^\sigma_\beta}\rangle$$

Overall Algorithm:

- Sample configurations
- Prepare a pure-nqs, evolve using Imaginary time evolution
- Keep track of the Energy along the evolution
- Average along enough trajectories

(ME)TTS: Heisenberg 2D

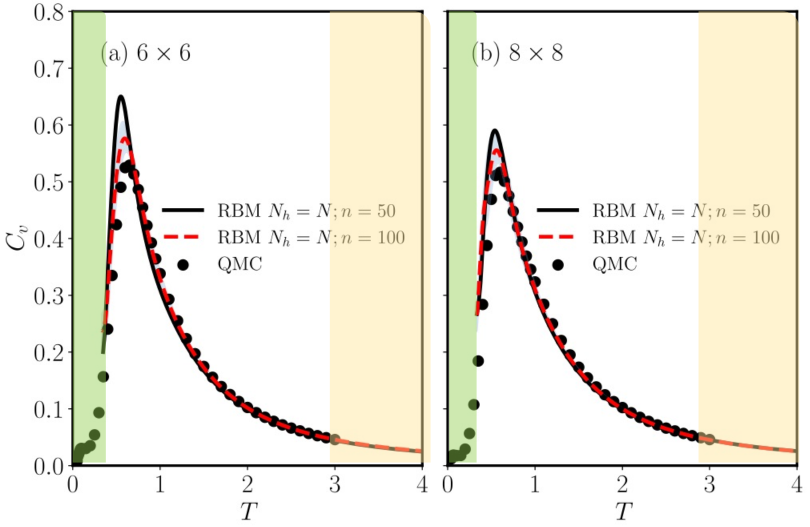


FIG. 3. Specific heat of the two-dimensional Heisenberg model comparing results obtained with restricted Boltzmann machines and quantum Monte Carlo for two system sizes (a) 6 × 6 and (b) 8 × 8 with varying number of initial random states n and $N_h = N$ hidden variables. The shaded area represents the error for the dashed curve.

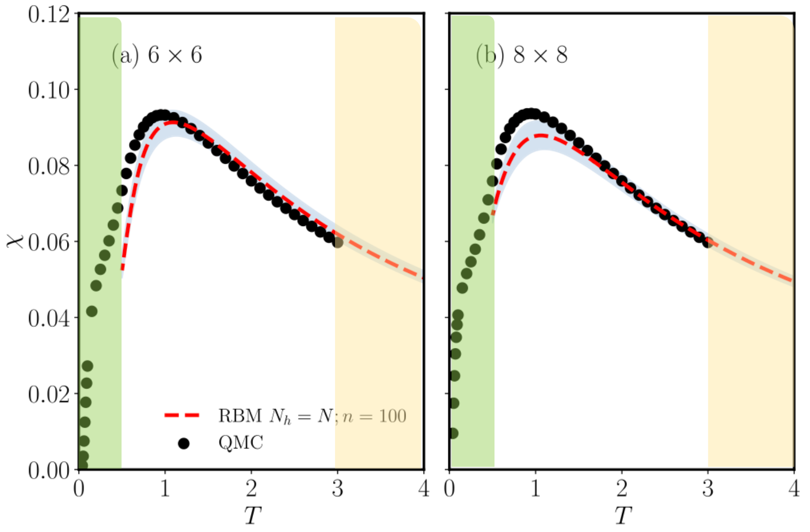


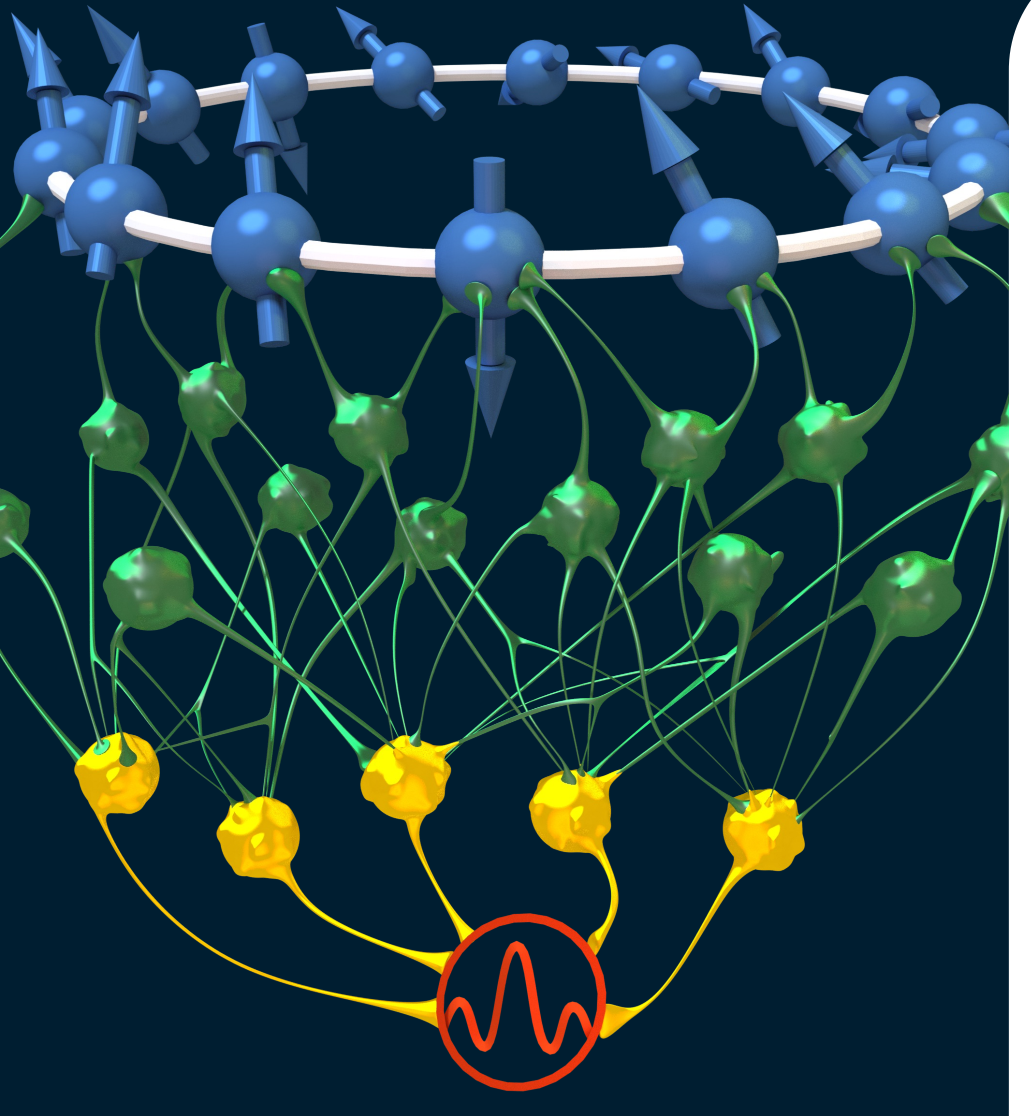
FIG. 5. Magnetic susceptibility of the two-dimensional Heisenberg model comparing results obtained with restricted Boltzmann machines and quantum Monte Carlo for two system sizes (a) 6 × 6 and (b) 8 × 8 with $n = 100$ initial random states n and $N_h = N$ hidden variables. The shaded area represents the error for the dashed curve.

Issues:

- Hard to initialize the state: other basis states are used
- Discrepancy at low temperatures (Error in the dynamics: not enough parameters)
- High Computational cost (full dynamics simulation per trajectory)

[Hendry et Al, PRB **106**, 165111 (2022)]

ThermoField Neural Dynamics



$$\hat{\rho} = \sum_{\sigma,\eta} \exp[\log \rho(\sigma,\eta)] \, |\sigma\rangle \langle\eta|$$

Purification Ansatz

Consider a wavefunction in a larger system

$$\psi \in \mathcal{H} \otimes \mathcal{H}_a$$

Trace the ancilla to get a physical Density matrix

$$\rho(\sigma,\eta) = \sum_a \psi(\sigma,a)\psi^\star(\eta,a)$$

ThermoField

Prepare an Infinite Temperature State

$$\rho_0 = \mathrm{Tr}_a[|\Psi(T=\infty)\rangle\langle\Psi(T=\infty)|]$$

$$\rho_\beta = e^{-\frac{\beta}{2}\hat{H}}\,\mathrm{Tr}_a[|\Psi(T=\infty)\rangle\langle\Psi(T=\infty)|]e^{-\frac{\beta}{2}\hat{H}}$$

where

$$|\Psi(T=\infty)\rangle = \otimes_{i=1}^{N_{\mathrm{sites}}}(|\uparrow\downarrow'\rangle + |\downarrow\uparrow'\rangle)_i$$

Therefore we Can simply Evolve the «Pure state» in the doubled-space

Integrate with any method for the dynamics

$$|\psi_\beta\rangle = \left(e^{-\frac{\beta}{2}\hat{H}} \otimes \mathbb{I}_a\right)|\Psi(T=\infty)\rangle$$

Then "thermal" expectation values are standard ones
In the doubled space

$$\langle\hat{A}\rangle_\beta = \frac{\mathrm{Tr}\left[\hat{A}\,|\psi_\beta\rangle\langle\psi_\beta|\right]}{\mathrm{Tr}[|\psi_\beta\rangle\langle\psi_\beta|]} = \frac{\langle\psi_\beta|\hat{A}|\psi_\beta\rangle}{\langle\psi_\beta|\psi_\beta\rangle}$$

[Nomura et Al, **PRL 127** 060601 (2021)]

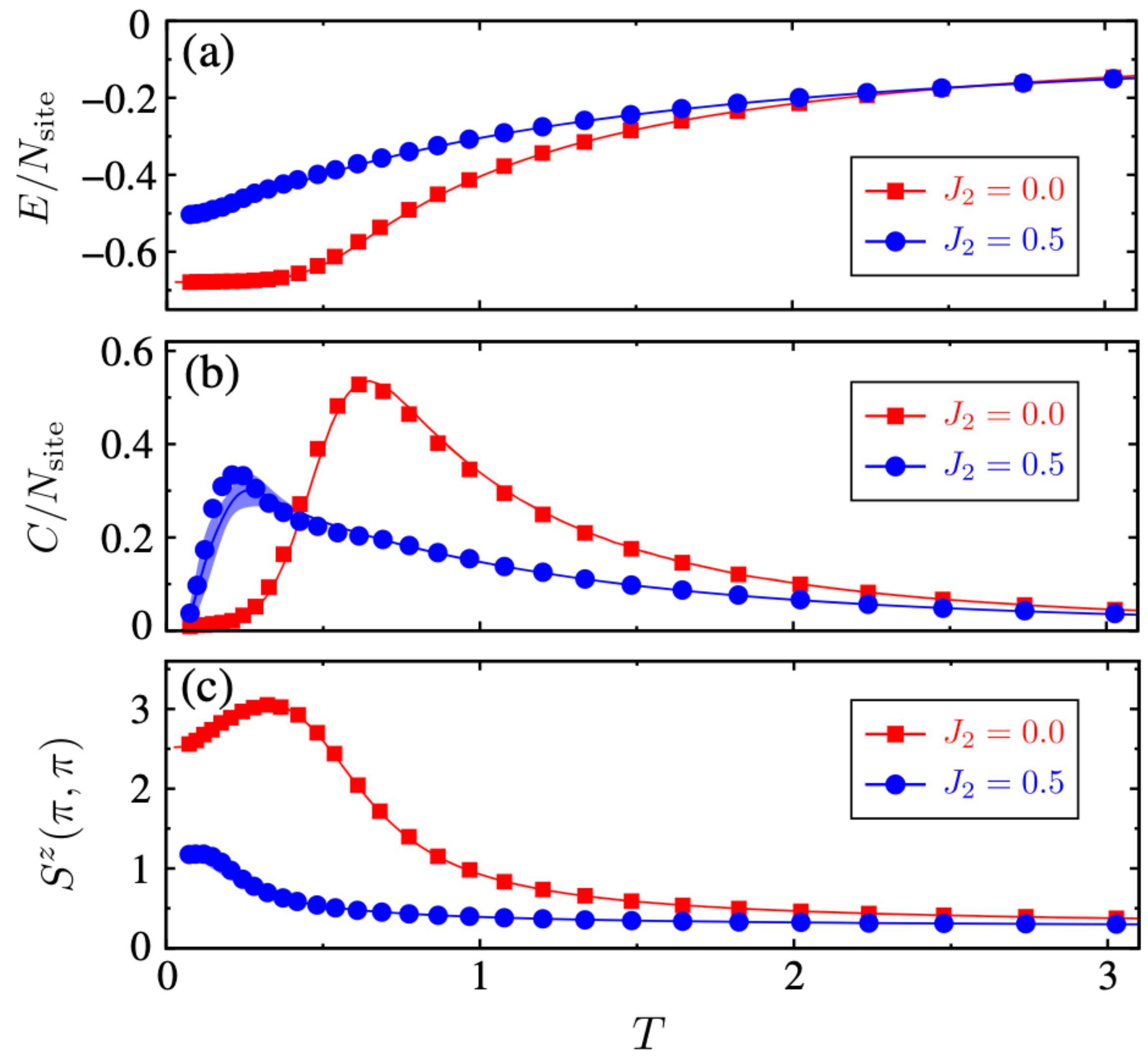
ThermoField : J1-J2 in 2D

$$\hat{H} = J_1 \sum_{\langle ij \rangle} \vec{S}_i \cdot \vec{S}_j + J_2 \sum_{\langle\langle ij \rangle\rangle} \vec{S}_i \cdot \vec{S}_j$$

2D, 6x6 Lattice

Issues:

This approach only works for Imaginary time dynamics. Lindblad dynamics “breaks” this approach and cannot be Applied.



[Nomura et Al, **PRL 127** 060601 (2021)]

Finite Temperature

Typical Thermal States :

- Originally introduced in the ctxt. Tensor Networks
- «unravel» the density matrix into a set of trajectories
- Expensive: many trajectories

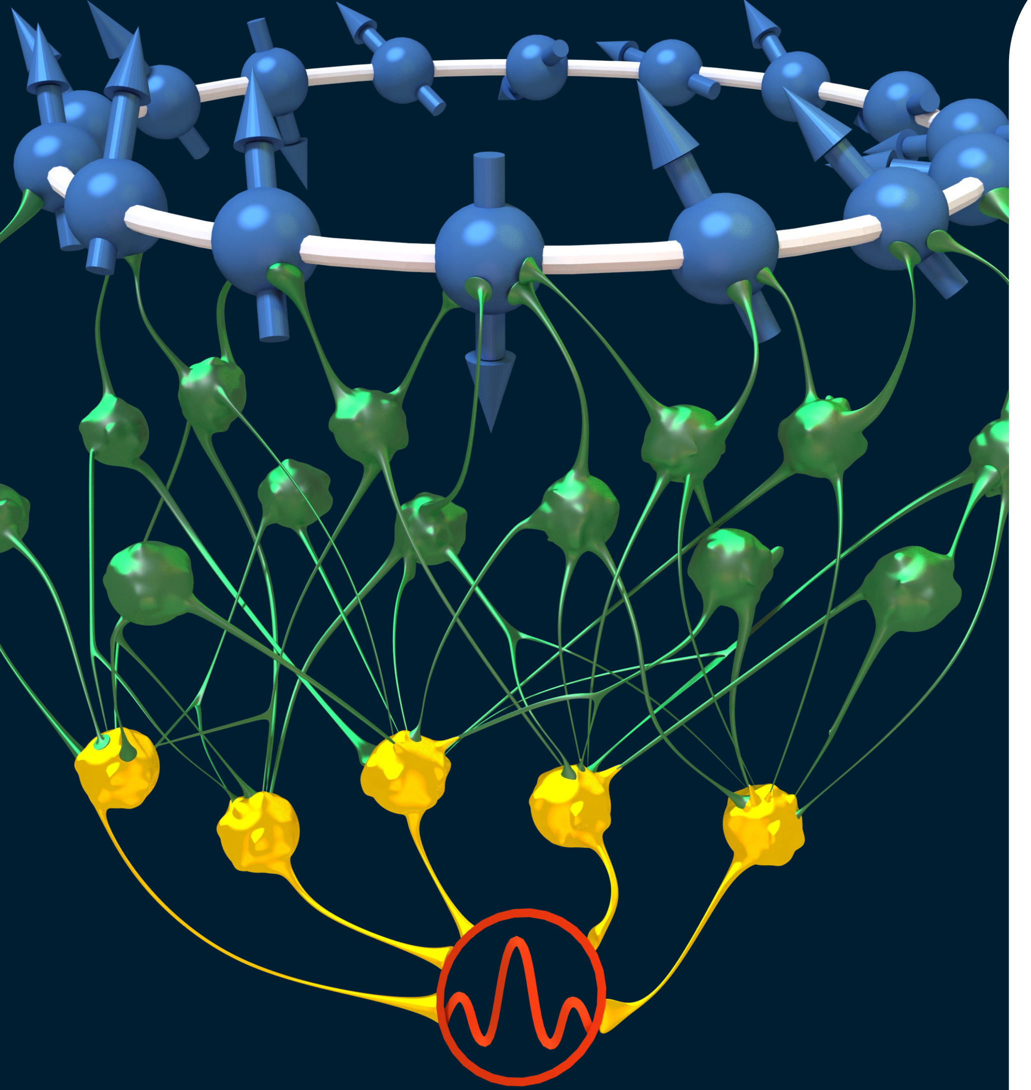
Thermofield:

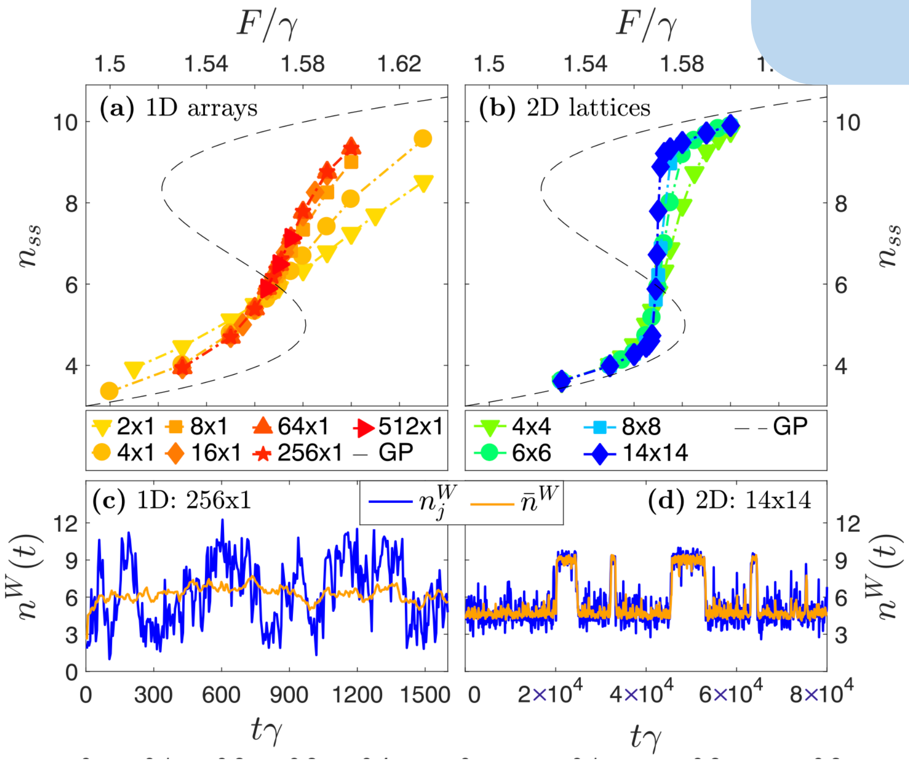
- Also coming from the community of Tensor Networks
- Only 1 trajectory
- Does not work For Open Systems

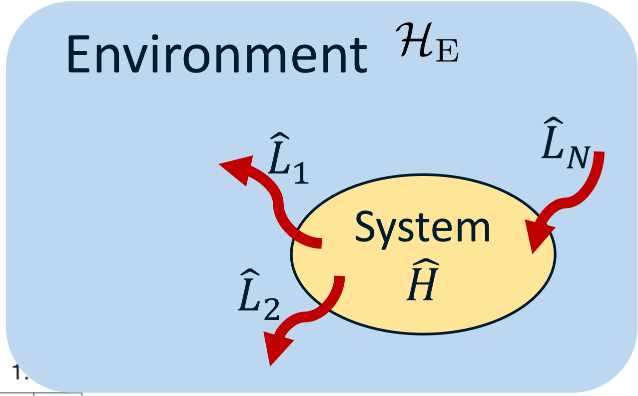
Problem:

- Initialization on a «peaked» state is hard
- Dynamics from the initial «peaked state» is hard

Neural Network Quantum States out of Equilibrium



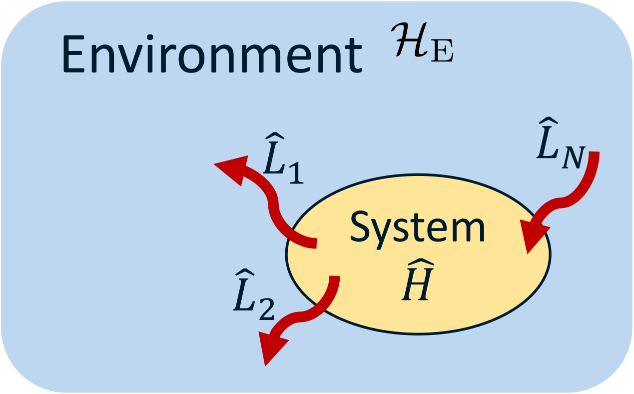




The Linblad Master Equation

The state

$$|\psi\rangle \in \mathcal{H}_{\text{sys}} \otimes \mathcal{H}_{\text{E}} \quad \Longrightarrow \quad \hat{\rho} = \text{Tr}_{\text{E}}[|\psi\rangle\langle\psi|]$$



The master equation

$$\frac{\partial \hat{\rho}}{\partial t} = -i\left[\hat{H}, \hat{\rho}\right] + \sum_{j=1}^{N_{\text{channels}}} \left(\hat{L}_j \hat{\rho} \hat{L}_j^\dagger - \frac{1}{2}\left\{ \hat{L}_j^\dagger \hat{L}_j, \hat{\rho} \right\} \right)$$

Coherent evolution — Incoherent Evolution

Assumption:
- Secular Approx.
- Born-Markov Approx.

Rewritten as a linear super-operatorial equation

$$\partial_t \hat{\rho} = \mathcal{L}\hat{\rho}$$

We are usually interested in the steady-state

$$\mathcal{L}\hat{\rho}_{\text{ss}} = 0$$

$$\hat{\rho}_{\text{ss}} = \lim_{t\to\infty} e^{\mathcal{L}t}\hat{\rho}$$

[Bruer & Petruccione, Open Quantum Systems (2012)]

Numerical Methods for the Master Eq.

Density Matrix methods

Steady-state only: solve the system

$$\begin{cases} \mathcal{L}\hat{\rho} = 0 \\ \mathrm{Tr}[\hat{\rho}] = 1 \end{cases}$$

Solve the time-evolution ODE

$$\hat{\rho}(t) = e^{\mathcal{L}t}\hat{\rho}(0)$$

Trajectory methods

Density matrix as an ensemble of trajectories

$$\hat{\rho}(t) \approx \frac{1}{N}\sum_{i=1}^{N} |\psi_i(t)\rangle \langle\psi_i(t)|$$

Evolve individual trajectories

$$\frac{d\,|\psi(t)\rangle}{dt} = -iH_{\mathrm{eff}}\,|\psi(t)\rangle + dP(t)\hat{L}_{\mathrm{eff}}\,|\psi(t)\rangle$$

Numerical Methods for the Master Eq.

Density Matrix methods

Steady-state only: solve the system

$$\begin{cases} \mathcal{L}\hat{\rho} = 0 \\ \mathrm{Tr}[\hat{\rho}] = 1 \end{cases}$$

Solve the time-evolution ODE

$$\hat{\rho}(t) = e^{\mathcal{L}t}\hat{\rho}(0)$$

Steady-State optimisation (VMC)

Steady State

$$\frac{d\rho_{\mathrm{ss}}}{dt} = \mathcal{L}\rho_{\mathrm{ss}} = 0$$

Variational Principle

$$\mathcal{C}(\boldsymbol{v}) = \frac{\| d\hat{\rho}_{\boldsymbol{v}}/dt \|_2^2}{\|\hat{\rho}_{\boldsymbol{v}}\|_2^2} = \frac{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^{\dagger}\mathcal{L}^{\dagger}\mathcal{L}\hat{\rho}_{\boldsymbol{v}}\right]}{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^{\dagger}\hat{\rho}_{\boldsymbol{v}}\right]}$$

$$\mathcal{C}(\boldsymbol{v}_{ss}) = 0 \iff \hat{\rho}_{\boldsymbol{v}_{ss}} = \hat{\rho}_{ss}$$

$$\mathcal{C}(\boldsymbol{v}) \geq 0$$

[Vicentini et Al, PRL **122** (2019)]
[Weimer, PRL **114** (2015)]

Numerical Methods for the Master Eq.

Density Matrix methods

Steady-state only: solve the system

$$\begin{cases} \mathcal{L}\hat{\rho} = 0 \\ \mathrm{Tr}[\hat{\rho}] = 1 \end{cases}$$

Solve the time-evolution ODE

$$\hat{\rho}(t) = e^{\mathcal{L}t}\hat{\rho}(0)$$

Dynamical Simulation (TDVP or Implicit)

$$\frac{d\rho(t)}{dt} = \mathcal{L}\rho(t) \quad \Leftrightarrow \quad \frac{d\mathbf{v}}{dt} = S^{-1}F$$

[Nagy et Al, PRL **122** (2019)]
[Hartmann and Carleo, PRL **122** (2019)]

Steady-State optimisation (VMC)

Steady State

$$\frac{d\rho_{\mathrm{ss}}}{dt} = \mathcal{L}\rho_{\mathrm{ss}} = 0$$

Variational Principle

$$\mathcal{C}(\boldsymbol{v}) = \frac{\|d\hat{\rho}_{\boldsymbol{v}}/dt\|_2^2}{\|\hat{\rho}_{\boldsymbol{v}}\|_2^2} = \frac{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger \mathcal{L}^\dagger \mathcal{L}\hat{\rho}_{\boldsymbol{v}}\right]}{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger \hat{\rho}_{\boldsymbol{v}}\right]}$$

$$\mathcal{C}(\boldsymbol{v}_{ss}) = 0 \iff \hat{\rho}_{\boldsymbol{v}_{ss}} = \hat{\rho}_{ss}$$

$$\mathcal{C}(\boldsymbol{v}) \geq 0$$

[Vicentini et Al, PRL **122** (2019)]
[Weimer, PRL **114** (2015)]

Neural Quantum States

Variational Ansatz

$$\hat{\rho} \approx \hat{\rho}_w$$

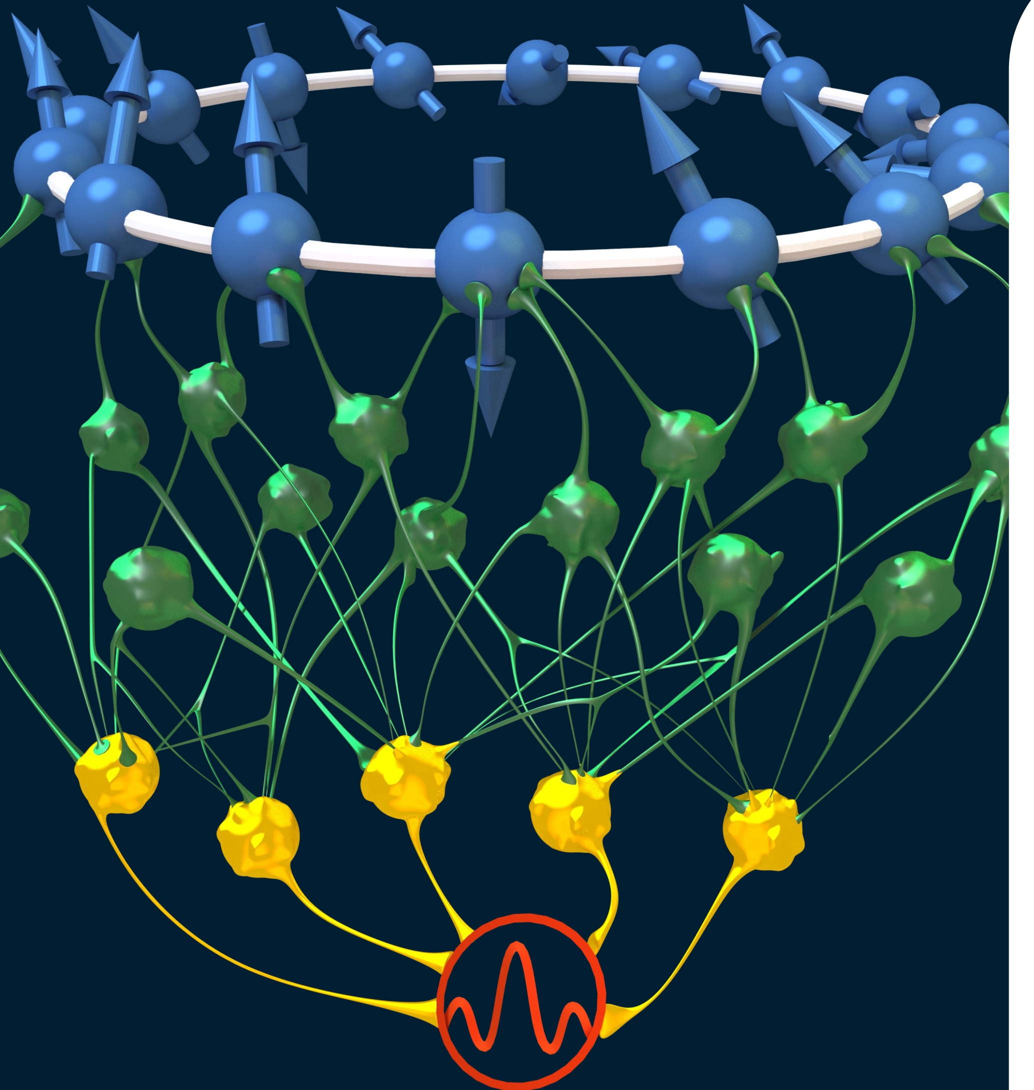
Lowers Memory complexity

Cost Function

$$\mathcal{C}(w)$$

To determine the variational parameters

Neural Density Operators



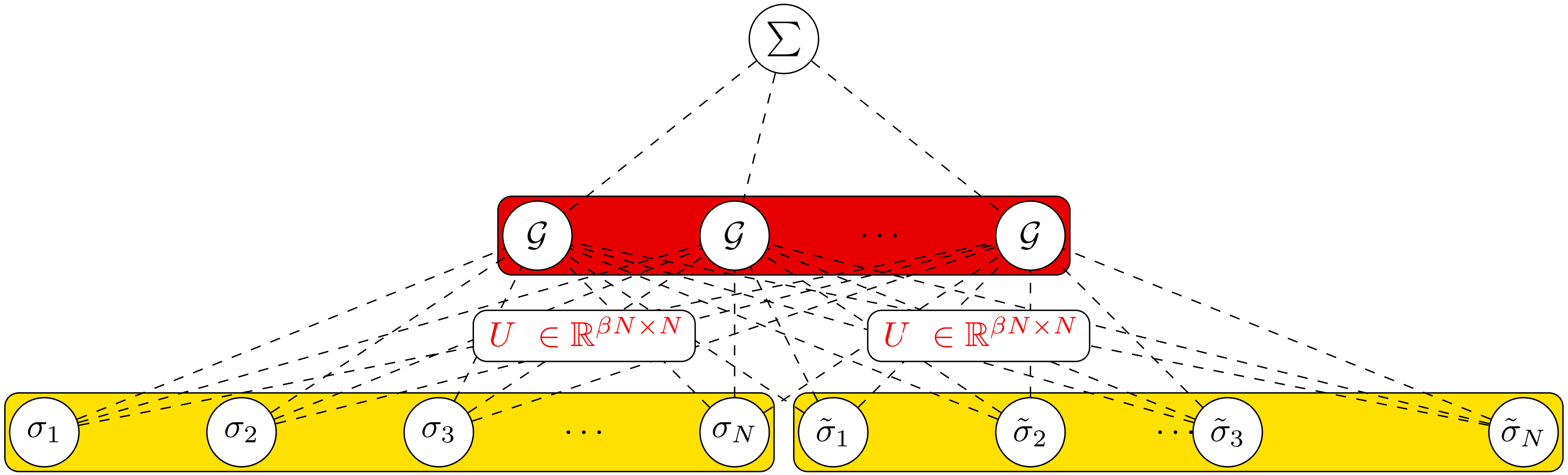
$$\hat{\rho} = \sum_{\sigma,\eta} \exp[\log \rho(\sigma, \eta)] \, |\sigma\rangle \langle \eta|$$

$$\log \rho : \{\pm 1\}^{\otimes 2N} \to \mathbb{C}$$

NEURAL DENSITY OPERATORS

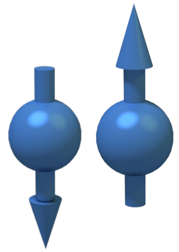
$$\hat{\rho} = \sum_{\sigma,\eta} \exp\left[\log \rho(\sigma, \eta)\right] |\sigma\rangle \langle\eta|$$

$$\log \rho(\sigma, \eta) = \sum_{j} \mathcal{G}(U^{[j]}_{,i}\sigma_i + U^{[j]*}_{,i}\eta_i + d^{[j]})$$



Neural Density Operators

$$\hat{\rho} = \sum_{\sigma,\eta} \exp\left[\log \rho(\sigma,\eta)\right] |\sigma\rangle \langle\eta|$$



$$\log \rho(\sigma,\eta) = \sum_{j} \mathcal{G}\left(U^{[j]}_{,i}\sigma_i + U^{[j]*}_{,i}\eta_i + d^{[j]}\right)$$

Purification Ansatz

Consider a wavefunction in a larger system

$$\psi \in \mathcal{H} \otimes \mathcal{H}_a$$

Trace the ancilla to get a physical Density matrix

$$\rho(\sigma,\eta) = \sum_{a} \psi(\sigma,a)\psi^{\star}(\eta,a)$$

Issues:

- Analytical summation -> Exponentially costly in ancilla size
- Stochastic summation -> Does it work?

Exception: «shallow» ancilla /RBM

- Leads to a condition on the nonlinearity G

$$\psi(\boldsymbol{\sigma},\boldsymbol{a}) = \Gamma_{\psi}(\boldsymbol{\sigma}) \exp\left[-\mathbf{a}^T \left(U\boldsymbol{\sigma} + \mathbf{d}\right)\right]$$

[Torlai, PRL 120, 240503 (2018)]

Purified NDO Ansatz

Shallow Ancilla Ansatz

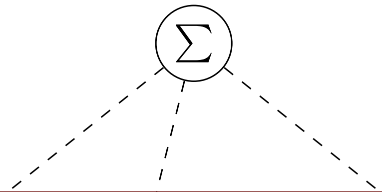
$$\psi(\boldsymbol{\sigma}, \boldsymbol{a}) = \Gamma_\psi(\boldsymbol{\sigma}) \exp\left[-\mathbf{a}^T \left(U\boldsymbol{\sigma} + \mathbf{d}\right)\right]$$

«Purification»

$$\rho(\boldsymbol{\sigma}, \boldsymbol{\eta}) = \sum_{\boldsymbol{a}} \psi(\boldsymbol{\sigma}, \mathbf{a})\psi^\star(\boldsymbol{\eta}, \mathbf{a})$$

$$\log \rho(\boldsymbol{\sigma}, \boldsymbol{\eta}) = \Gamma_\psi(\boldsymbol{\sigma})\Gamma^\star_\psi(\boldsymbol{\eta}) \sum_j^H \mathcal{G}(U_i^{[j]}\sigma_i + U_i^{[j],\star}\eta_i + d^{[j]})$$

$$\mathcal{G} = \log\cosh$$



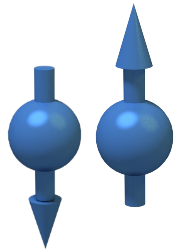
It's not deep.

Can we make it better?

[Torlai, PRL 120, 240503 (2018)]

Gram-Hadamard Density operator

[Gram]: Purification of a small Ancilla



Consider a bosonic ancilla of size d, which can be purified by brute-force summation

$$\pi_i(\boldsymbol{\sigma}, \boldsymbol{\eta}) = \sum_{a=1}^{d} \psi_i(\boldsymbol{\sigma}, a)\psi_i^\star(\boldsymbol{\eta}, a)$$

$$\mathrm{Rank}[\pi_i] = d$$

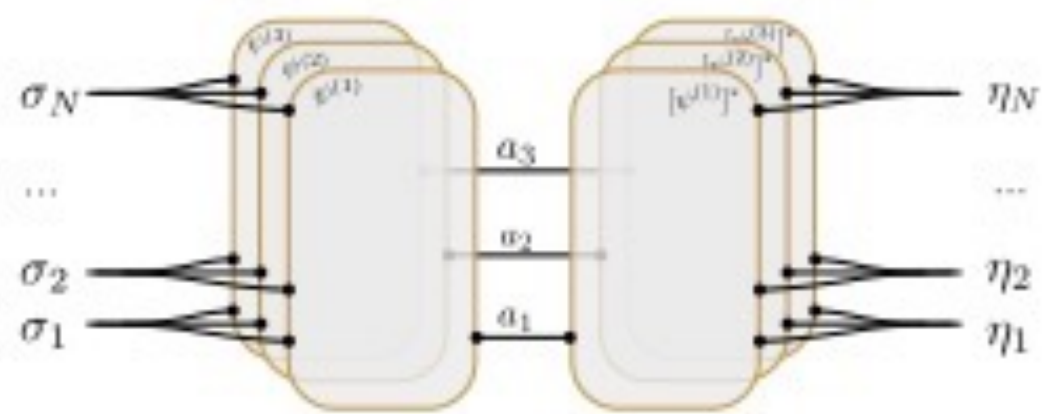
[Hadamard]: Product of two Positive definite matrices is Positive Definite

$$\rho(\boldsymbol{\sigma}, \boldsymbol{\eta}) = \prod_{i}^{M} \pi_i(\boldsymbol{\sigma}, \boldsymbol{\eta})$$

$$\mathrm{Rank}[\hat{\rho}] \leq \prod_{i}^{M} \mathrm{Rank}[\pi_i] \leq d^M$$

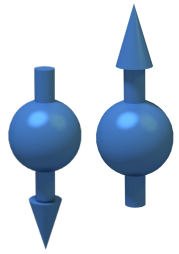
The $\hat{\pi}_i$ defined here is PD because it's a Gram Matrix

$\psi_i \ (\sigma, a)$ Can be an Arbitrary NN



Gram-Hadamard Density Matrix

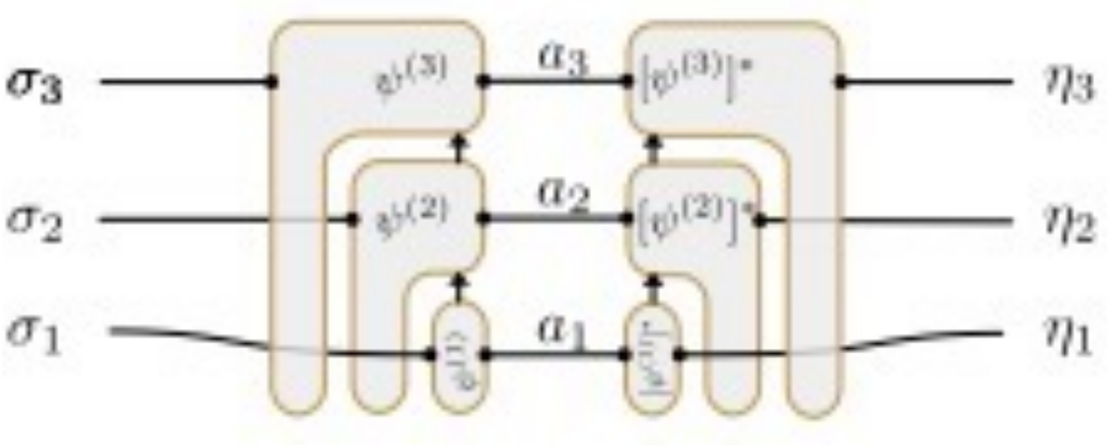
I want to perform Autoregressive Sampling of the diagonal



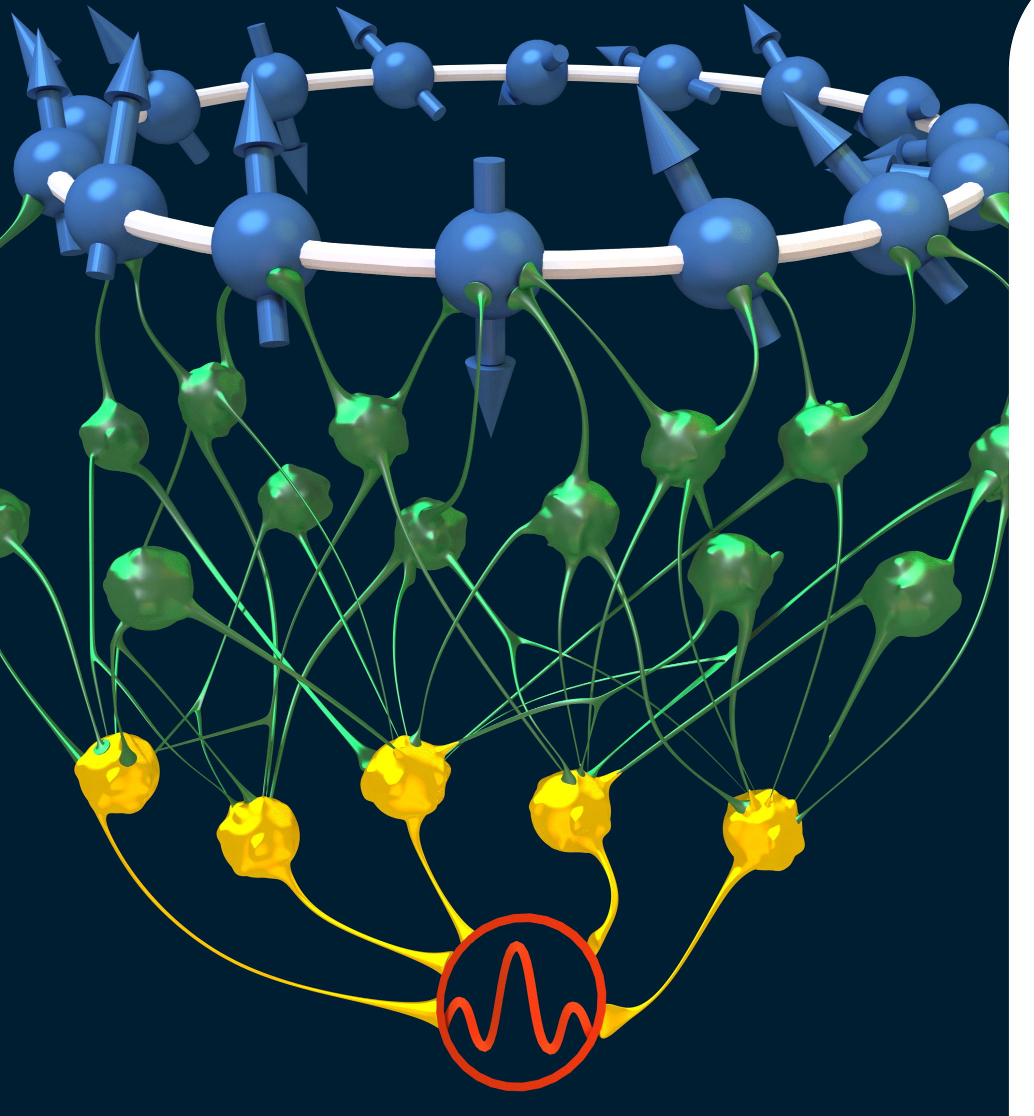
$$\rho(\boldsymbol{\sigma},\boldsymbol{\sigma}) = p_1(\sigma_1)p_2(\sigma_2|\sigma_1)p_3(\sigma_3|\sigma_2,\sigma_1)\ldots p_N(\sigma_N|\sigma_{<N})$$

Equivalent to considering ϕ_i in autoregressive order:

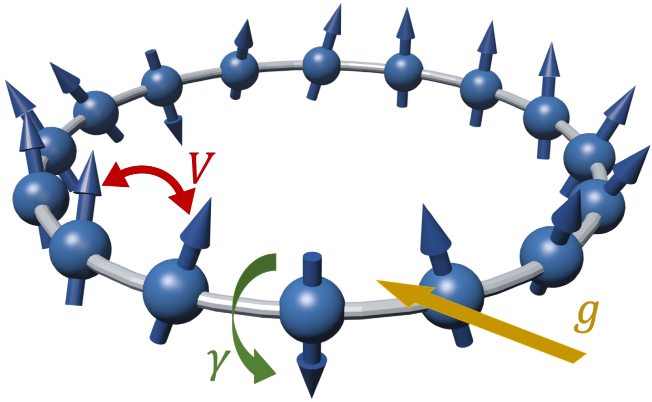
$$\psi^{(i)}_{a_i}(\sigma_i|\boldsymbol{\sigma}_{<i}) = \frac{\tilde{\psi}^{(i)}_{a_i}(\sigma_i|\boldsymbol{\sigma}_{<i})}{\sum_{\sigma_i=\{\pm 1\}}\tilde{\psi}^{(i)}_{a_i}(\sigma_i|\boldsymbol{\sigma}_{<i})}$$



[**F.V.**, Rossi, Carleo, arXiv:2206.13488]



The Driven-Dissipative Transverse-Field Ising Model



The hamiltonian is

$$H = \frac{V}{4} \sum_{\langle j,l \rangle} \hat{\sigma}_j^z \hat{\sigma}_l^z + \frac{g}{2} \sum_j \hat{\sigma}_j^x$$

With de-exitation jump operators $\hat{L}_j = \sigma_j^-$

We study the magnetizations $m^\alpha = \frac{1}{N} \sum_{j=1}^{N} \sigma_j^\alpha$

Benchmark: D-D Transverse Field Ising

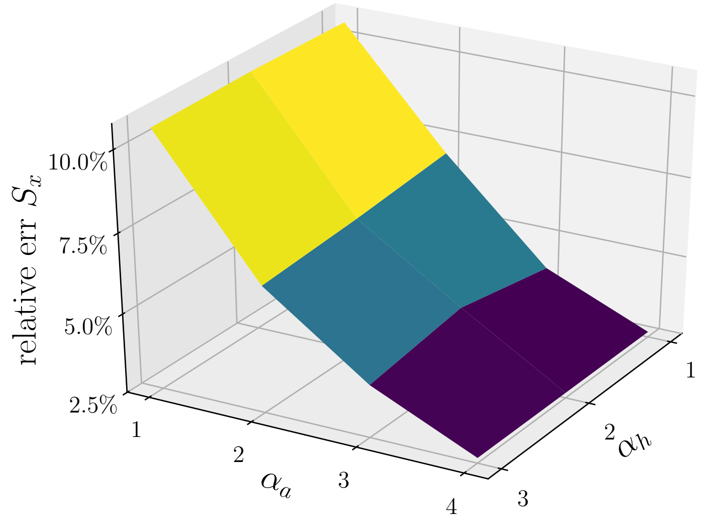
The hamiltonian is

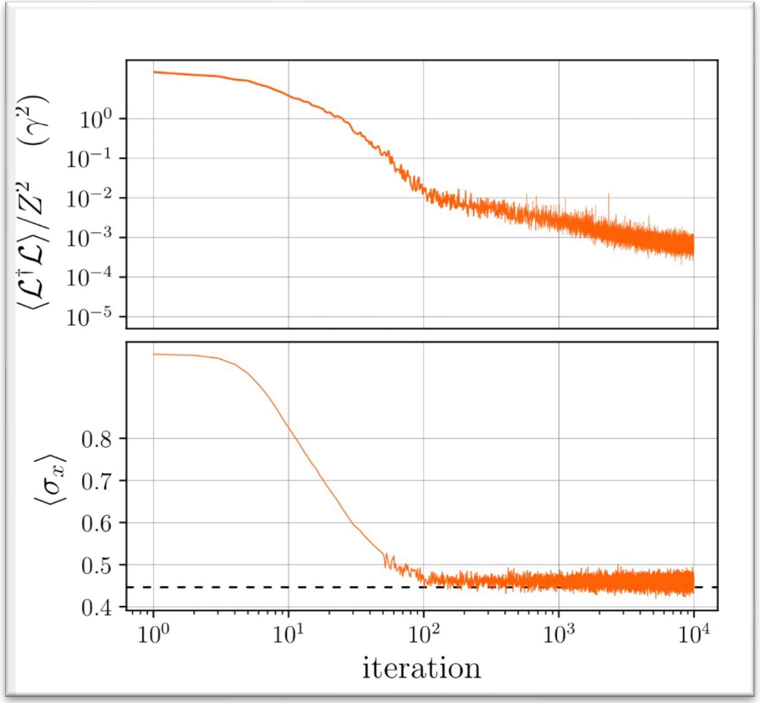
$$H = \frac{V}{4} \sum_{\langle j,l \rangle} \hat{\sigma}_j^z \hat{\sigma}_l^z + \frac{g}{2} \sum_j \hat{\sigma}_j^x$$

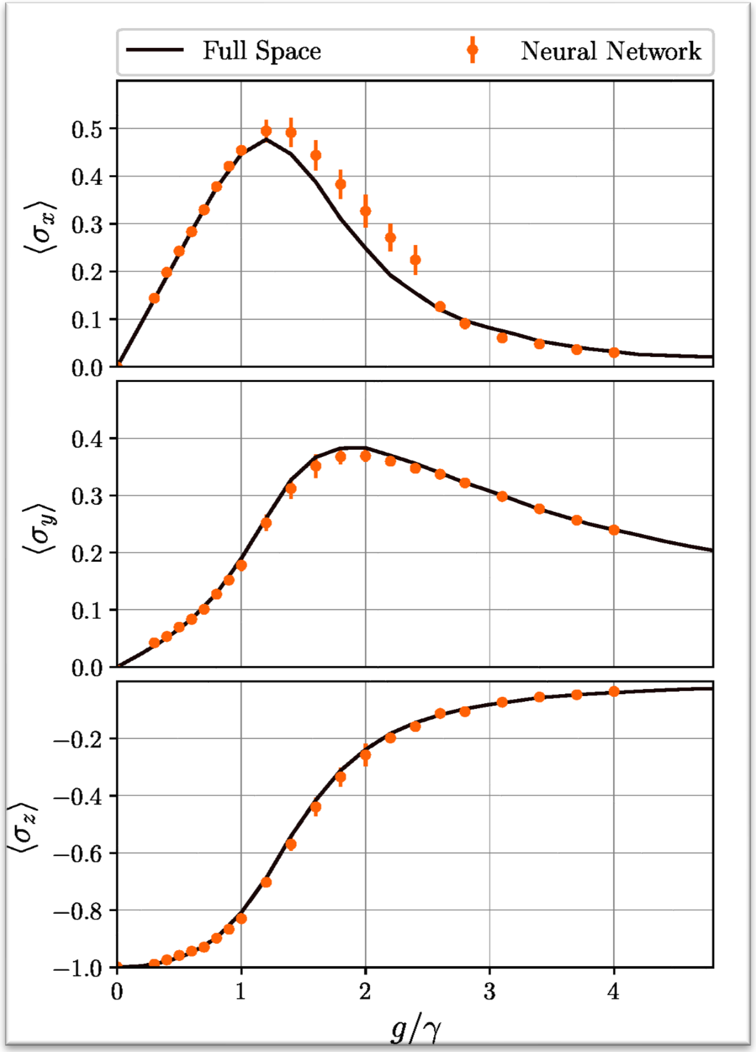
With de-exitation jump operators $\hat{L}_j = \sigma_j^-$

$$\mathcal{C}(\boldsymbol{v}) = \frac{\| d\hat{\rho}_{\boldsymbol{v}}/dt \|_2^2}{\| \hat{\rho}_{\boldsymbol{v}} \|_2^2} = \frac{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger \mathcal{L}^\dagger \mathcal{L} \hat{\rho}_{\boldsymbol{v}} \right]}{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger \hat{\rho}_{\boldsymbol{v}} \right]}$$

With «Shallow» Purified Ansatz







[**F.V.**, Biella, Regault, Ciuti, PRL **122** (2019)]

Benchmark: D-D Transverse Field Ising

The hamiltonian is

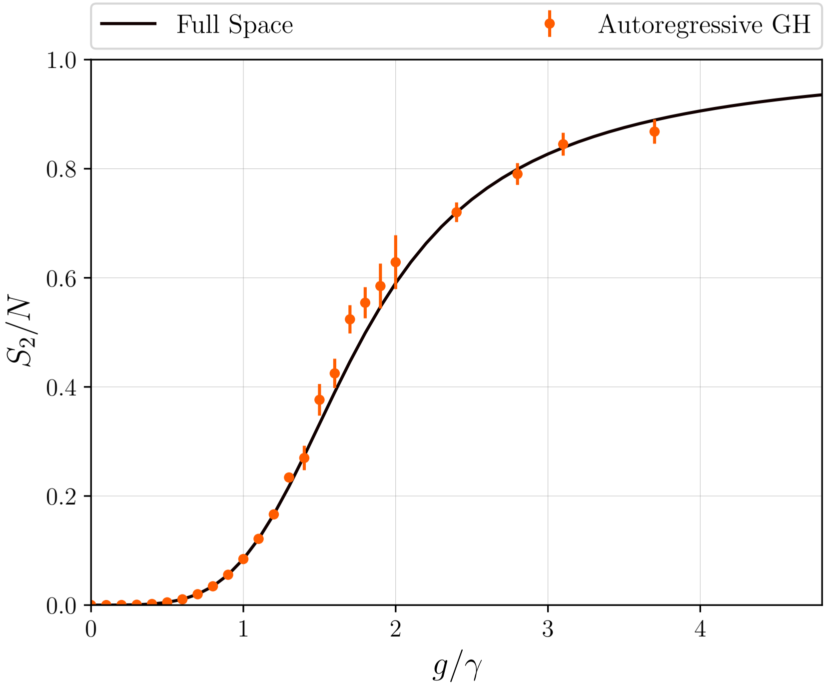
$$H = \frac{V}{4}\sum_{\langle j,l\rangle}\hat{\sigma}_j^z\hat{\sigma}_l^z + \frac{g}{2}\sum_j \hat{\sigma}_j^x$$

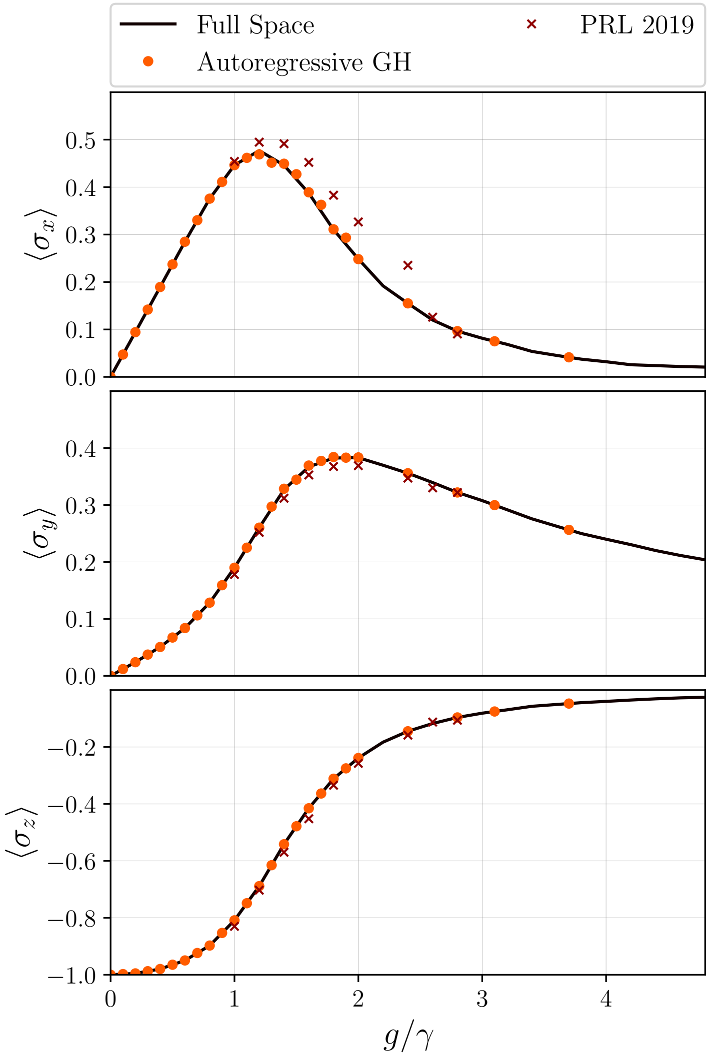
With de-exitation jump operators $\hat{L}_j = \sigma_j^-$

$$\mathcal{C}(\boldsymbol{v}) = \frac{\|d\hat{\rho}_{\boldsymbol{v}}/dt\|_2^2}{\|\hat{\rho}_{\boldsymbol{v}}\|_2^2} = \frac{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger\mathcal{L}^\dagger\mathcal{L}\hat{\rho}_{\boldsymbol{v}}\right]}{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger\hat{\rho}_{\boldsymbol{v}}\right]}$$

~~With «Shallow» Purified Ansatz~~

With Deep Gram-Hadamard Ansatz





[**F.V.**, Rossi, Carleo, arXiv:2206.13488]

Benchmark: D-D Transverse Field Ising

The hamiltonian is

$$H = \frac{V}{4}\sum_{\langle j,l\rangle} \hat{\sigma}_j^z \hat{\sigma}_l^z + \frac{g}{2}\sum_j \hat{\sigma}_j^x$$

With de-exitation jump operators $\hat{L}_j = \sigma_j^-$

~~With «Shallow» Purified Ansatz~~

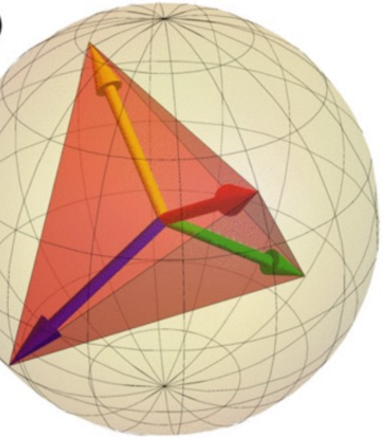
With Deep Gram-Hadamard Ansatz

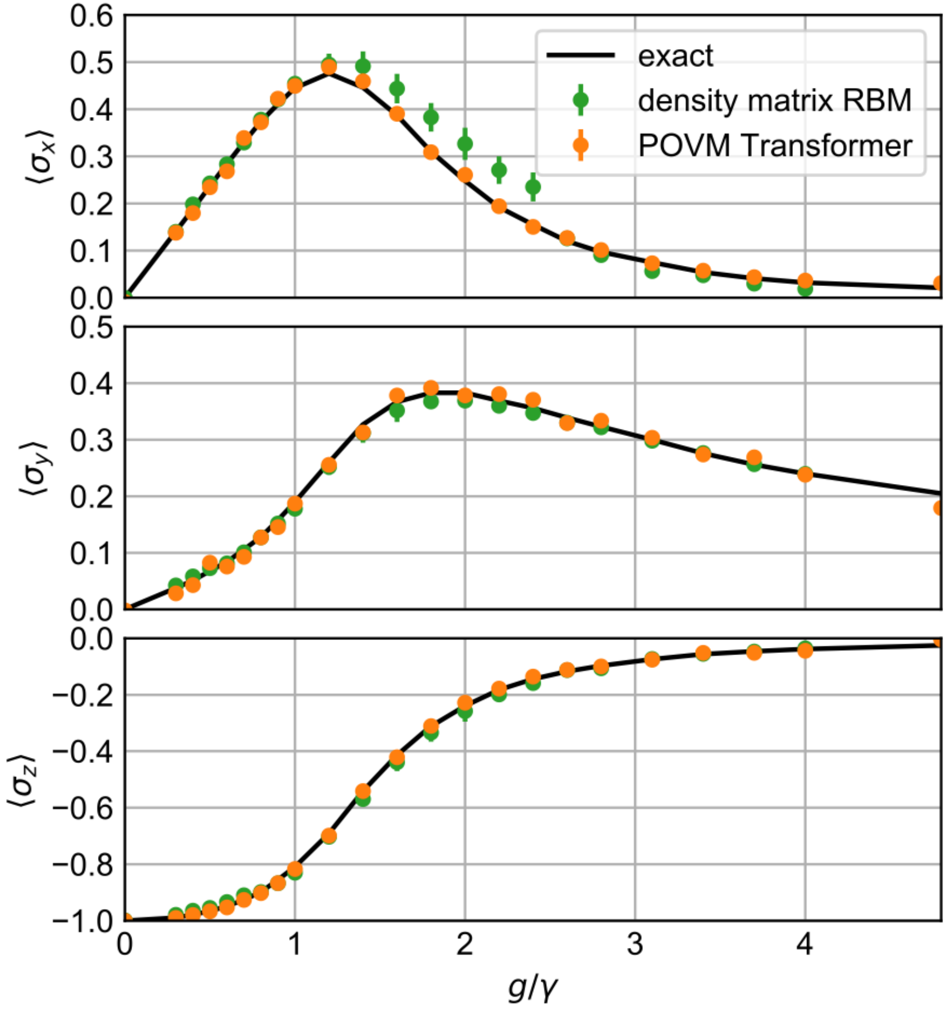
With Positive Operator Valued Measure Basis

$$\hat{\rho} = \sum_a P(a)\hat{K}_a$$

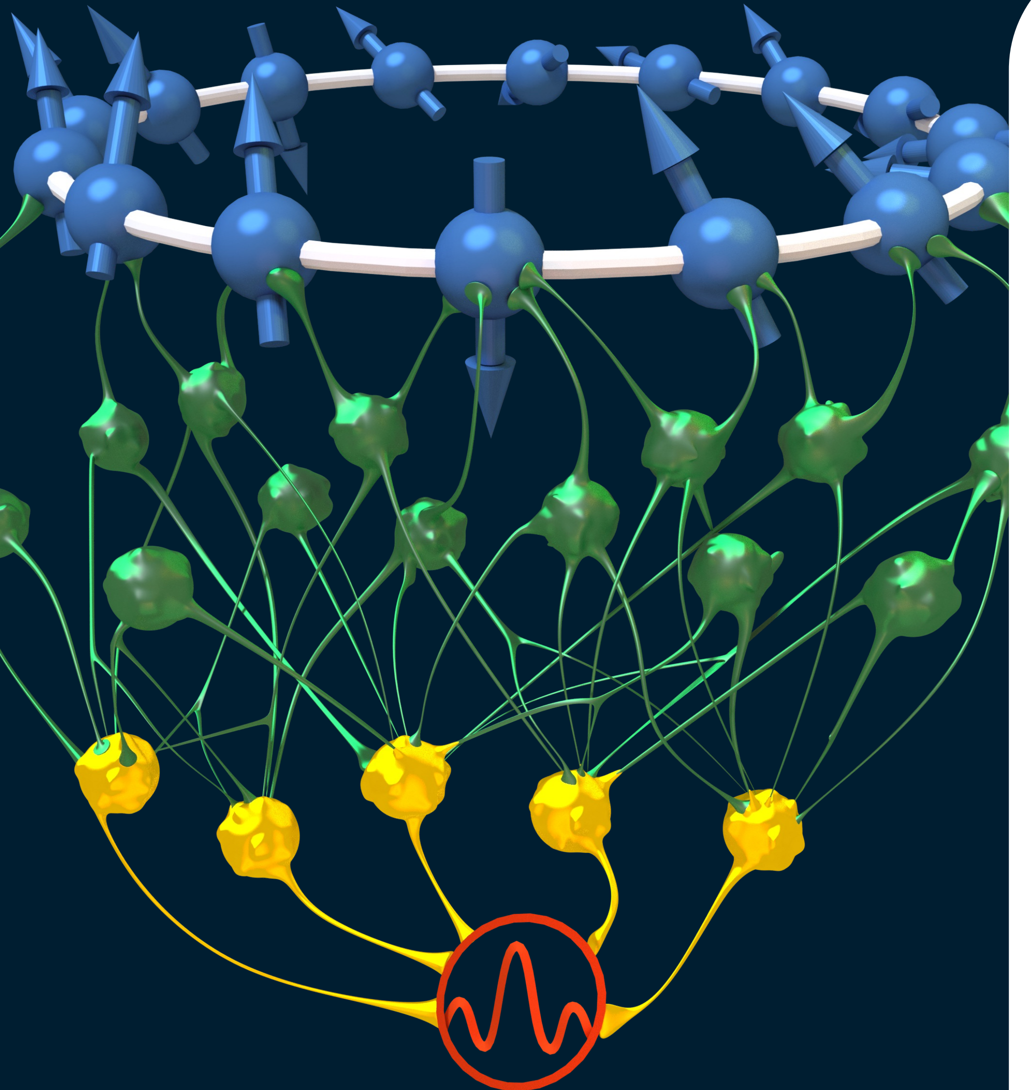
$$\mathcal{C}(v) = \mathbb{E}_{a\sim P_v(a)}\left[\sum_b \mathcal{L}_{a,b}\frac{P_v(b)}{P_v(a)}\right]$$

(b)





[Di Luo et Al, PRL **128**, 090501 (2022)]



OPEN SYSTEM DYNAMICS

DYNAMICAL SIMULATION (TDVP OR IMPLICIT)

$$\frac{d\rho(t)}{dt} = \mathcal{L}\rho(t) \quad \Leftrightarrow \quad \frac{d\mathbf{v}}{dt} = S^{-1}F$$

[Nagy et Al, PRL **122** (2019)]
[Hartmann and Carleo, PRL **122** (2019)]

Benchmark: Dissipative XYZ model

The hamiltonian is $\hat{H} = \sum_{\langle i,j \rangle} (J_x \, \hat{X}_i \vec{X}_j + J_y \, \hat{Y}_i \vec{Y}_j + J_z \, \hat{Z}_i \vec{Z}_j)$

With de-exitation jump operators $\hat{L}_j = \sigma_j^-$

With «Shallow» Purified Ansatz

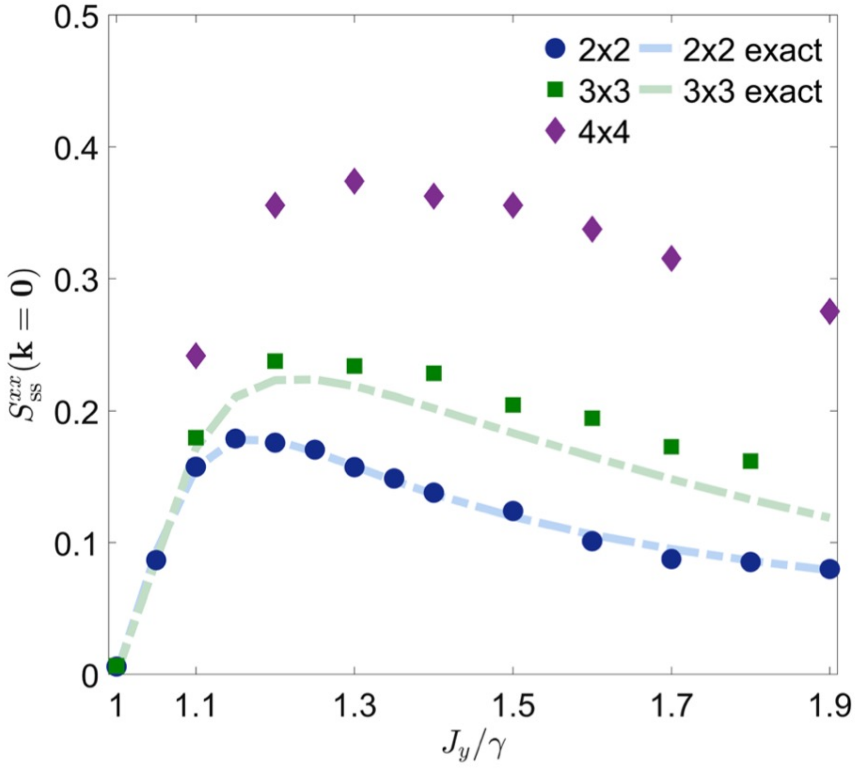


FIG. 4. The steady-state spin structure factor $S_{\rm SS}^{xx}(\mathbf{k}=\mathbf{0})$ computed as a function of the coupling J_y/γ. VMC and exact values are compared. Other parameters: $J_x/\gamma = 0.9$, $J_z/\gamma = 1.0$, $\alpha = \beta = 3$.

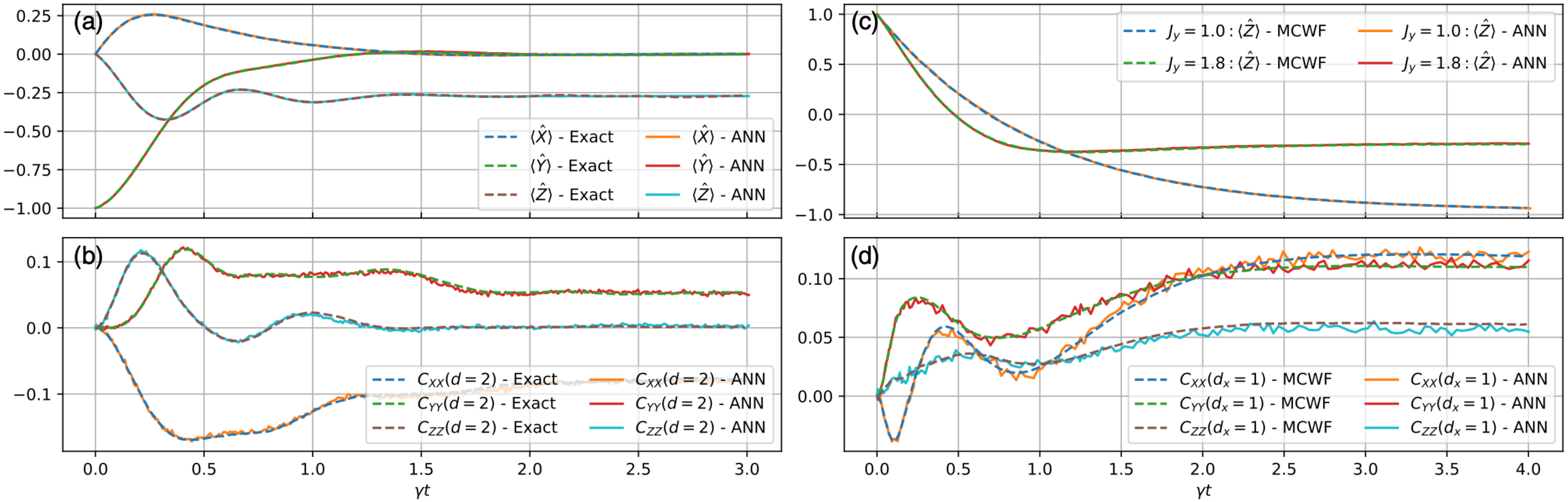
[Nagy and Savona, PRL **122** (2019)]

Benchmark: D-D XYZ model

The hamiltonian is $$\hat{H} = \sum_{\langle ij \rangle} (J_x \hat{X}_i \vec{X}_j + J_y \hat{Y}_i \vec{Y}_j + J_z \hat{Z}_i \vec{Z}_j) + \sum_i h_z \hat{Z}_i$$

With de-exitation jump operators $\hat{L}_j = \sigma_j^-$

2D 4x4



[Reh et Al, PRL **127** (2021)]

CONCLUSIONS

Finite Temperature: structured problem

- Not necessary to really represent the density matrix
- Imaginary time evolution ⇔ initialization problem
- Few works on the subject

Mixed States: encoding positivity of matrix is HARD

- Direct sampling + Positivity for deep networks
- Mapping to EPS **[F.V. et Al, arXiv: 2206.13488]**
- Unclear the advantage/disadvantage of POVM

Open Problems: working with zeros in our parametrisation

- Measurements enforce zeros in the wavefunction. Working on it (see seminar by G. Carleo)
- Density Matrices: full of zeros

Aknowledgements

Alberto Biella

Nicolas Regnault

Cristiano Ciuti

Alessandro Sinibaldi (EPFL)

Riccardo Rossi (Sorbonne, Paris)

Giuseppe Carleo (EPFL)

www.netket.org

NetKet: The Machine Learning toolbox for Quantum Many-Body Physics

Filippo Vicentini[1,2*], Damian Hofmann[3], Attila Szabó[4,5], Dian Wu[1,2], Christopher Roth[6], Clemens Giuliani[1,2], Gabriel Pescia[1,2], Jannes Nys[1,2], Vladimir Vargas-Calderón[7], Nikita Astrakhantsev[8] and Giuseppe Carleo[1,2]

Easy-Monte Carlo

Standard algorithms are easy to use

```
import netket as nk
comp_basis = nk.hilbert.Spin(1/2)**20
sampler = nk.sampler.MetropolisLocal(comp_basis,
                                     n_chains=32,
                                     n_sweeps=100)
ψ = nk.vqs.MCState(sampler, nk.models.RBM(),
                   n_samples=10**4)
ψ.expect(nk.operator.spin.sigmax(comp_basis, site=0))
# 0.99996 ± 0.00018 [σ²=0.00024, R̂=1.0015, τ=0.9<1.3]

ψ.grad(nk.operator.spin.sigmax(ψ.hilbert, site=0))
# dict({
    Dense: {
        bias: Array([ 2.85870451e-04,  1.34708020e-04,
                2.62814971e-04,  1.83681735e-04,
          }})
```

Jax - Powered

Write your Neural Networks in any jax-compatible framework or use one from the built-in library

Geometric Tensor

Efficient, easy to use implementation of the Quantum Geometric Tensor

Fast

Runs on GPUs, scales with MPI across 10s to 100s of nodes and GPUs

Full-Featured

VMC, Lindblad Master Equation, TDVP dynamics, Quantum state Reconstruction, continuous Systems (1^{st} quantisation) …

SciPost Physics Codebases 7 (2022)

Variational Monte Carlo in Practice

You want to perform a Variational Monte Carlo. You need to:

1. Write the Neural Network representing log-wavefunction
2. Write the formula for the gradient
3. Write a Monte-Carlo sampler
4. Represent the Hamiltonian
5. Compute local energies, estimate errors
6. Write an advanced optimiser

And…

- Maybe make it run on GPUs
- Change the network -> change the gradient
- Make it FAAAAST

Variational Monte Carlo in Practice

You want to perform a Variational Monte Carlo. You need to:

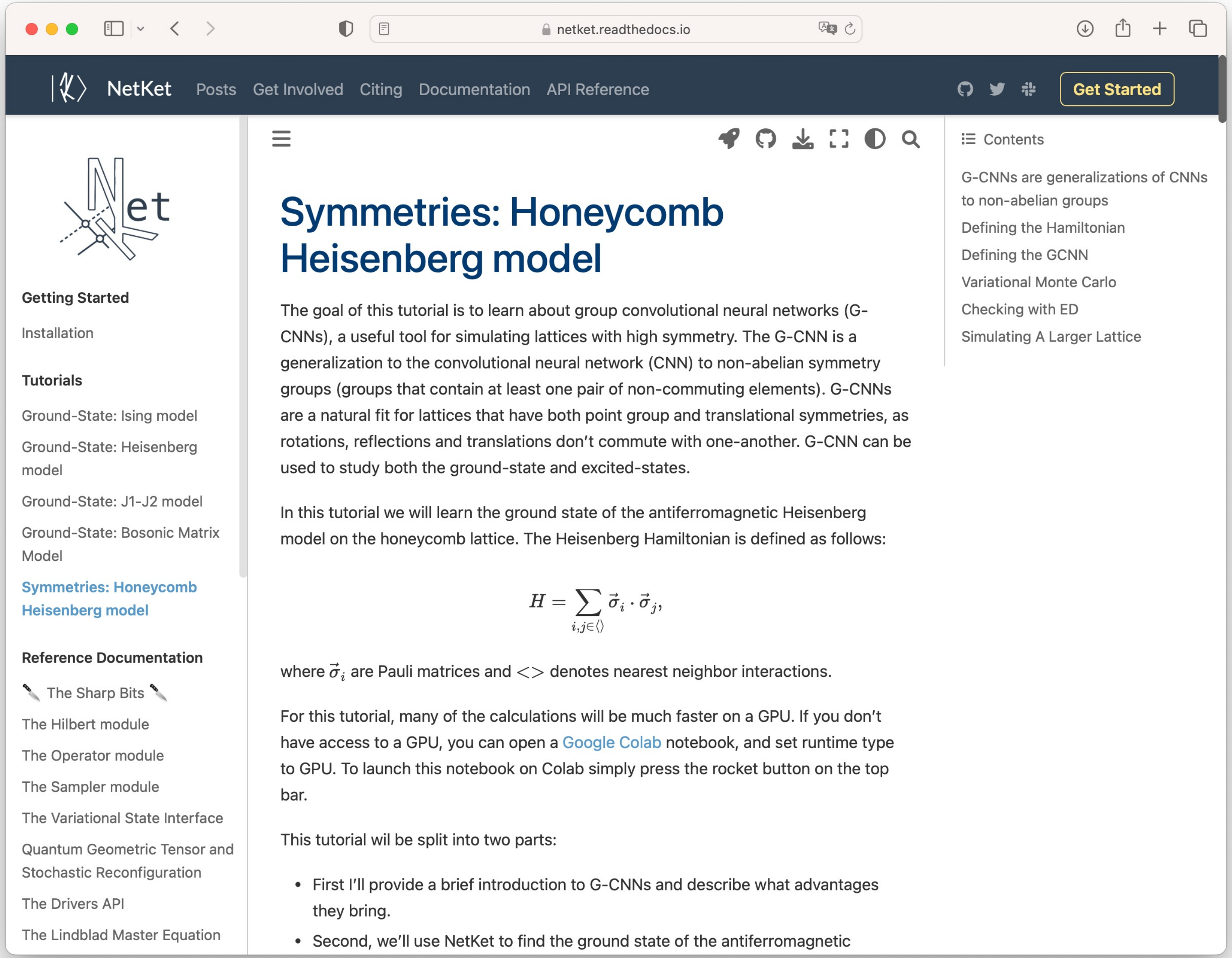
1. Write the Neural Network representing log-wavefunction
2. Write the formula for the gradient
3. Write a Monte-Carlo sampler
4. Represent the Hamiltonian
5. Compute local energies, estimate errors
6. Write an advanced optimiser

Optax

And…

- Maybe make it run on GPUs
- Change the network -> change the gradient
- Make it FAAAAST

Variational Monte Carlo in Practice



netket.readthedocs.io

NetKet Posts Get Involved Citing Documentation API Reference Get Started

Getting Started

Installation

Tutorials

Ground-State: Ising model

Ground-State: Heisenberg model

Ground-State: J1-J2 model

Ground-State: Bosonic Matrix Model

Symmetries: Honeycomb Heisenberg model

Reference Documentation

🔪 The Sharp Bits 🔪

The Hilbert module

The Operator module

The Sampler module

The Variational State Interface

Quantum Geometric Tensor and Stochastic Reconfiguration

The Drivers API

The Lindblad Master Equation

Symmetries: Honeycomb Heisenberg model

The goal of this tutorial is to learn about group convolutional neural networks (G-CNNs), a useful tool for simulating lattices with high symmetry. The G-CNN is a generalization to the convolutional neural network (CNN) to non-abelian symmetry groups (groups that contain at least one pair of non-commuting elements). G-CNNs are a natural fit for lattices that have both point group and translational symmetries, as rotations, reflections and translations don't commute with one-another. G-CNN can be used to study both the ground-state and excited-states.

In this tutorial we will learn the ground state of the antiferromagnetic Heisenberg model on the honeycomb lattice. The Heisenberg Hamiltonian is defined as follows:

$$H = \sum_{i,j \in \langle\rangle} \vec{\sigma}_i \cdot \vec{\sigma}_j,$$

where $\vec{\sigma}_i$ are Pauli matrices and $<>$ denotes nearest neighbor interactions.

For this tutorial, many of the calculations will be much faster on a GPU. If you don't have access to a GPU, you can open a Google Colab notebook, and set runtime type to GPU. To launch this notebook on Colab simply press the rocket button on the top bar.

This tutorial wil be split into two parts:

- First I'll provide a brief introduction to G-CNNs and describe what advantages they bring.
- Second, we'll use NetKet to find the ground state of the antiferromagnetic

Contents