

Title: Language models for simulating the dynamics of quantum systems

Speakers: Juan Felipe Carrasquilla  $\tilde{\text{A}}$ lvarez

Collection: Quantum Simulators of Fundamental Physics

Date: June 06, 2023 - 10:30 AM

URL: <https://pirsa.org/23060008>

# Language models for simulating the dynamics of quantum systems

Juan Felipe Carrasquilla Alvarez  
Vector Institute

Quantum Simulators of Fundamental Physics, June 6th, 2023, Perimeter Institute.





# Simulating physics with computers

## Simulating Physics with Computers

Richard P. Feynman

*Department of Physics, California Institute of Technology, Pasadena, California 91107*

*Received May 7, 1981*

### 1. INTRODUCTION

On the program it says this is a keynote speech—and I don't know what a keynote speech is. I do not intend in any way to suggest what should be in this meeting as a keynote of the subjects or anything like that. I have my own things to say and to talk about and there's no implication that anybody needs to talk about the same thing or anything like it. So what I want to talk about is what Mike Dertouzos suggested that nobody would talk about. I want to talk about the problem of simulating physics with computers and I mean that in a specific way which I am going to explain. The reason for doing this is something that I learned about from Ed Fredkin, and my entire interest in the subject has been inspired by him. It has to do with learning something about the possibilities of computers, and also something about possibilities in physics. If we suppose that we know all the physical laws perfectly, of course we don't have to pay any attention to computers. It's interesting anyway to entertain oneself with the idea that we've got something to learn about physical laws; and if I take a relaxed view here (after all I'm here and not at home) I'll admit that we don't understand everything.

The first question is, What kind of computer are we going to use to simulate physics? Computer theory has been developed to a point where it realizes that it doesn't make any difference; when you get to a *universal computer*, it doesn't matter how it's manufactured, how it's actually made. Therefore my question is, Can physics be simulated by a universal computer? I would like to have the elements of this computer *locally interconnected*, and therefore sort of think about cellular automata as an example (but I don't want to force it). But I do want something involved with the



Motivated and shaped the field of quantum computing which hopes to revolutionize computation through exploitation of quantum mechanical effects



## Simulating physics with computers

### Simulating Physics with Computers

Richard P. Feynman

#### 5. CAN QUANTUM SYSTEMS BE PROBABILISTICALLY SIMULATED BY A CLASSICAL COMPUTER?

Now the next question that I would like to bring up is, of course, the interesting one, i.e., Can a quantum system be probabilistically simulated by a classical (probabilistic, I'd assume) universal computer? In other words, a computer which will give the same probabilities as the quantum system does. If you take the computer to be the classical kind I've described so far, (not the quantum kind described in the last section) and there're no changes in any laws, and there's no hocus-pocus, the answer is certainly, **No!** This is called the hidden-variable problem: it is impossible to represent the results of quantum mechanics with a classical universal device. To learn a little bit about it, I say let us try to put the quantum equations in a form as close as

#### Feynman concludes:

the great difficulty. The only difference between a probabilistic classical world and the equations of the quantum world is that somehow or other it appears as if the probabilities would have to go negative, and that we do not know, as far as I know, how to simulate. Okay, that's the fundamental problem. I don't know the answer to it, but I wanted to explain that if I try my best to make the equations look as near as possible to what would be imitable by a classical probabilistic computer, I get into trouble.

Motivated the field  
of quantum computing

This is all still true  
today and is  
**fundamentally**  
linked to the notion  
of quantum speed-  
up in quantum  
computing.



**We can't simulate quantum systems exactly using probability. Can we do it approximately with language models?**

# Natural language processing

- Branch of computer science, linguistics, and machine learning concerned with giving computers the ability to process text and spoken words in a similar way humans do it.
- Machine translation
- Speech recognition
- Sentiment analysis
- Automatic summarization of text
- Text to image/video generation



# ChatGPT

- ChatGPT is a machine learning model which interacts in a conversational way
- Dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

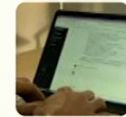
<https://openai.com/blog/chatgpt/>



## Can the new AI tool ChatGPT replace human work? Judge for yourself

A new artificial intelligence tool using natural language processing has captured the public's imagination, amassing more than a million...

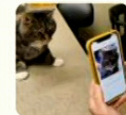
1 day ago



## ChatGPT: Everything to know about the viral, 'groundbreaking' AI bot - National | Globalnews.ca

Users can ask the AI to write essays, poems or scripts, or even translate or summarize text. ChatGPT can also answer questions on a wide...

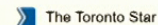
21 hours ago



## The 5 Best Uses (So Far) for ChatGPT's AI Chatbot

The new AI tool ChatGPT has inspired excitement and worry with its ability to instantly answer complex questions. In the days after its...

1 day ago



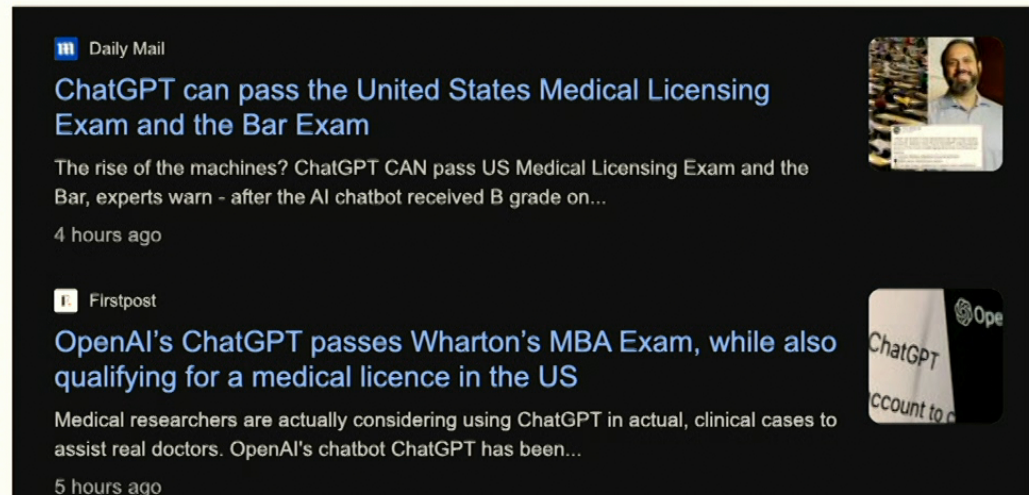
## I wrote a story about ChatGPT's AI. Then I dared it to write a better one

Like a million or so other artificial intelligence dumb-dumbs who discovered ChatGPT this week, I couldn't wait to show off the capabilities...



# ChatGPT

- People now believe these models will likely have deep technological, educational, and societal implications.



<https://openai.com/blog/chatgpt/>



# Is Mona Lisa smiling? ChatGPT's take



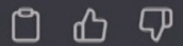
is mona lisa smiling?



The smile of the Mona Lisa has been a subject of much discussion and interpretation over the years. Leonardo da Vinci's famous portrait depicts a slight smile, which has often been described as enigmatic or mysterious. It's not an overt, obvious smile, but rather a subtle expression that has captivated viewers and sparked numerous theories and debates.

The perception of the Mona Lisa's smile can vary from person to person, and different individuals may interpret it differently. Some see it as a gentle, content smile, while others perceive it as somewhat ambiguous, with a touch of sadness or even a hint of irony.

Ultimately, the interpretation of the smile is subjective and open to personal perception.



# ChatGPT: is Mona Lisa smiling?



I'll do my best to create an ASCII art representation of the Mona Lisa, however, please keep in mind that it may not be a very detailed or realistic representation. Here is my attempt:



```
Copy code
```

Please note that this is a very simple and minimal representation of Monalisa, as ASCII art is not my primary function. It may not be the best representation of Monalisa.



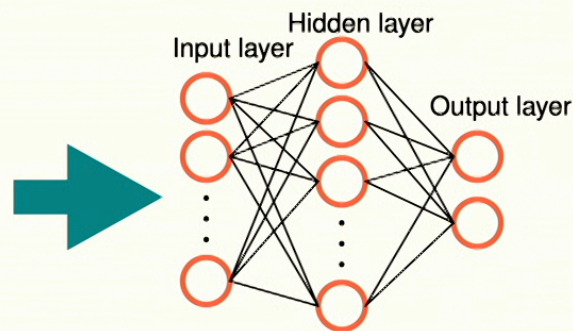
## Language models

- ▶ Large language models are based on neural networks
- ▶ Neural networks are powerful universal function approximators and can in principle represent any function if given enough parameters
- ▶ In particular, in NLP these model objects like this:

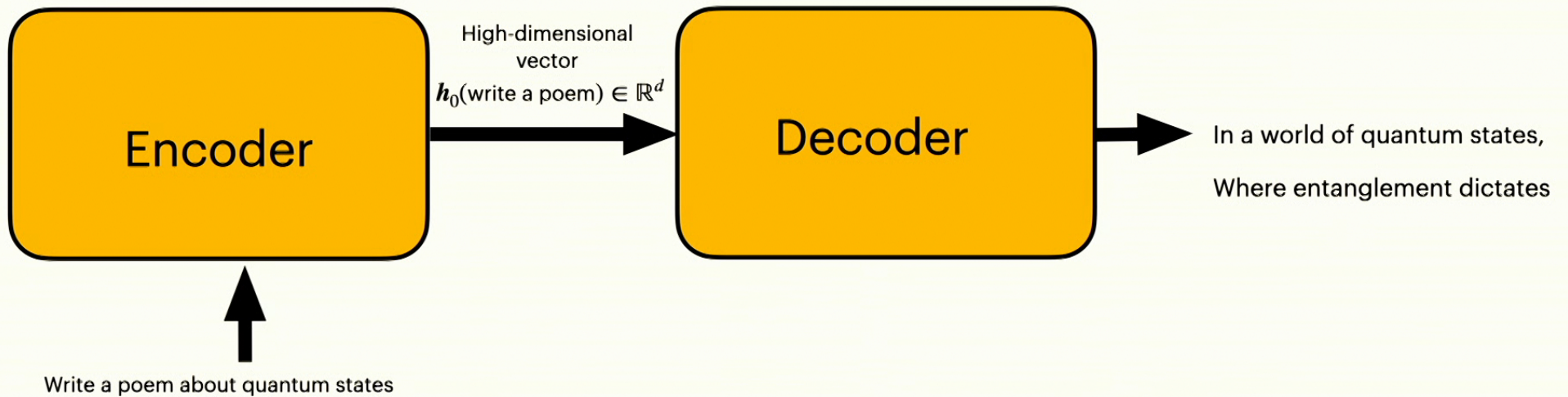
$P(\text{output text}|\text{input text})$

$P(\text{English}|\text{Spanish})$

$P(\text{Spanish})$



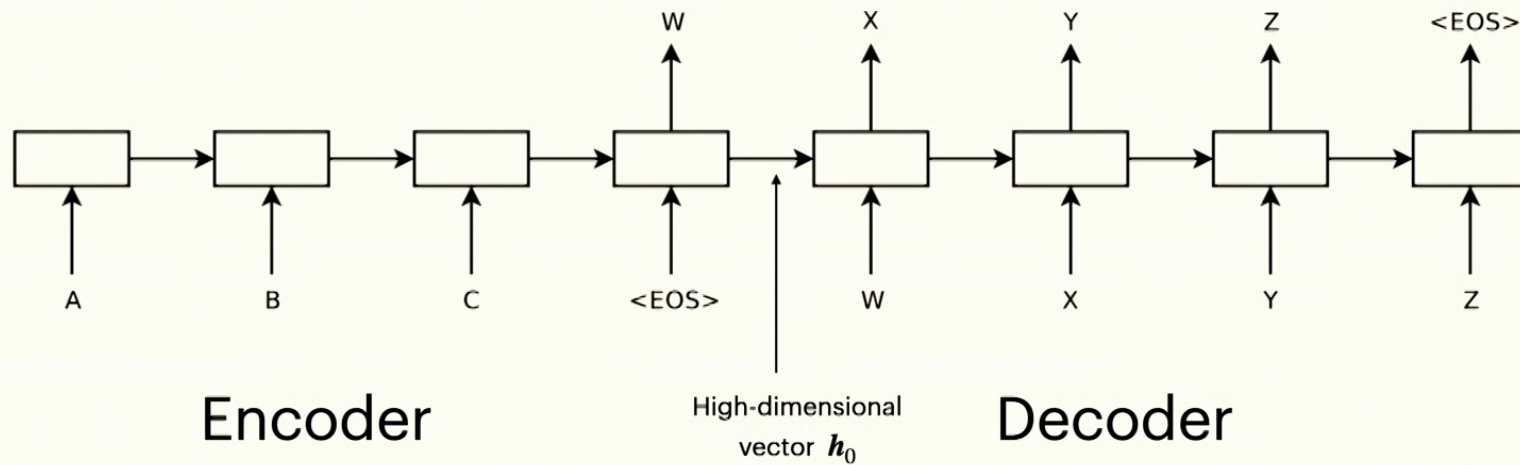
# Language models





# Large Language models

## Recurrent neural networks



Sequence to Sequence Learning with Neural Networks. [Ilya Sutskever, Oriol Vinyals, Quoc V. Le.](https://arxiv.org/abs/1409.3215)  
<https://arxiv.org/abs/1409.3215>

# Large Language models

## Transformer

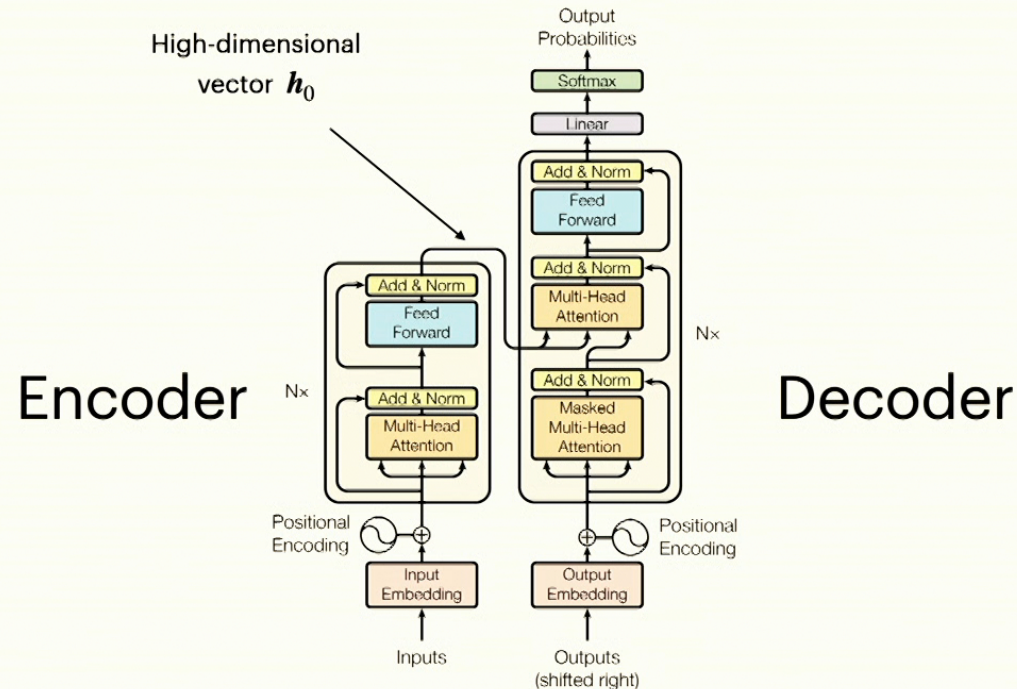


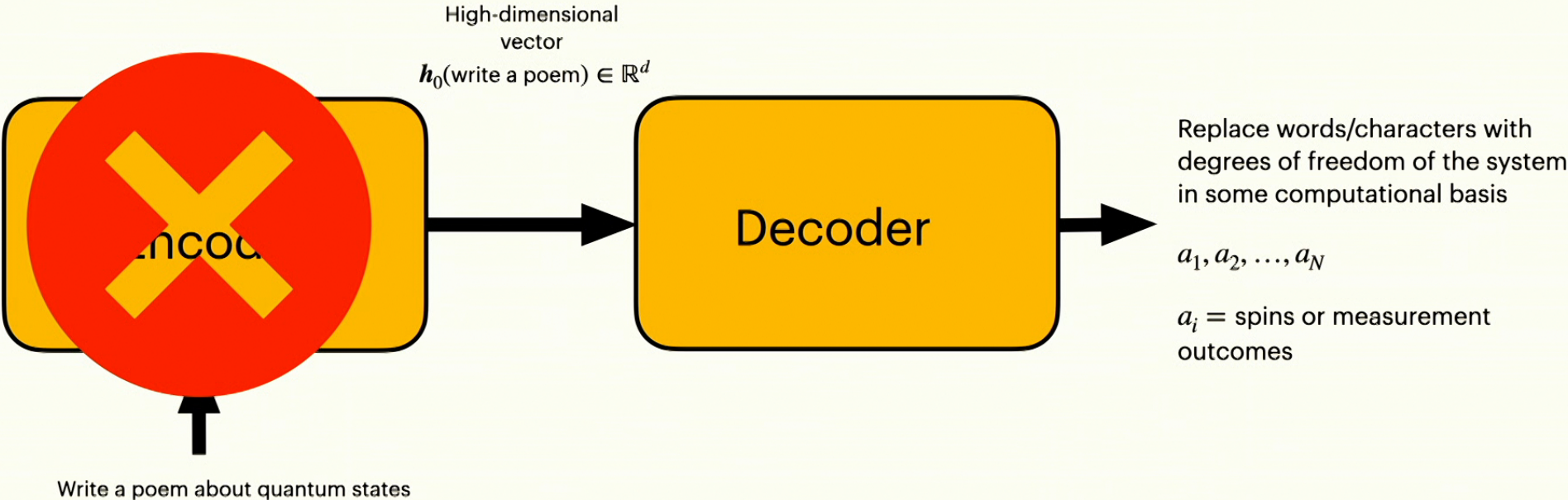
Figure 1: The Transformer - model architecture.

Attention Is All You Need. [Ashish Vaswani](#), [Noam Shazeer](#), [Niki Parmar](#), [Jakob Uszkoreit](#), [Llion Jones](#), [Aidan N. Gomez](#), [Lukasz Kaiser](#), [Illia Polosukhin](#) [arXiv:1706.03762](https://arxiv.org/abs/1706.03762)



**How to use these models to  
represent physical states?**

# Representing states with language models



Input/output is given using some word dictionary = [ word 1, word 2,...,word N] → Input/output is given using some word dictionary = [ spin up, spin down]



# Representing states with language models: Dictionaries

Input/output of a conditional is given using some word dictionary  
[ word 1, word 2,...,word M],  $M \approx 8000$



[  $|0\rangle$ ,  $|1\rangle$ ,  $|+\rangle$ ,  $|-\rangle$ ,  $|r\rangle$ ,  $|l\rangle$ ] – the  
outcomes of a quantum measurement

# LARGE LANGUAGE MODELS ARE AUTOREGRESSIVE

- The term *autoregressive* originates from time-series models: observations from the previous time-steps are used to predict the value at the current time step.
- Consider a probability distribution  $P(\boldsymbol{\sigma}) = P(\sigma_1, \sigma_2, \dots, \sigma_N)$ ,

$$P(\sigma_1, \sigma_2, \dots, \sigma_N) = P(\sigma_1)P(\sigma_2|\sigma_1)P(\sigma_3|\sigma_1, \sigma_2) \dots P(\sigma_N|\sigma_1, \sigma_2, \dots, \sigma_{N-1})$$

- To specify P in a tabular form requires **exponential** resources
- To alleviate this exponential issue: parametrize the conditionals

$$P(\sigma_i|\sigma_{<i}) = P_\theta(\sigma_i|\sigma_{<i})$$

- These conditionals are parameterized through large neural networks
- We can represent complex distributions using these models



## Autoregressive models and their tractability advantages

- ✓ Computing  $P(\sigma)$  or the amplitude  $\Psi(\sigma)$  is efficient given  $\sigma$
- ✓ Can be exactly sampled efficiently in linear time in  $N$
- ✓ Can be easily defined in **any** spatial dimension (e.g. 2d and 3d ) — no tractability issues (compared to, eg, PEPS)
- ✓ Their expressive power can be systematically improved
- ✓ Easy to encode product states and mean-field theories (e.g. Gutzwiller)
- ✓ We can impose some symmetries and conservation laws
- ✓ Advantages remain true for models of the quantum states  $\Psi(\sigma)$  in any dimension.

## How?

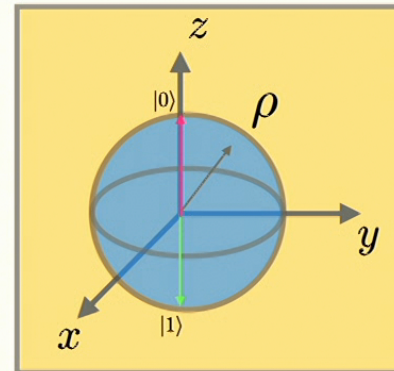
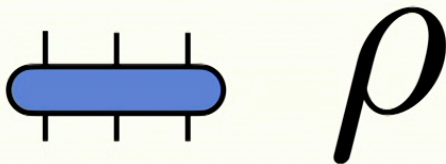
- We introduce a formulation of quantum theory that looks like probability—make quantum look like ML
- We use generative models, in particular transformers to parametrize quantum states
- Examples:
  - Open quantum system dynamics (today)
  - Quantum state tomography
  - Quantum circuits



# Quantum states, measurements, and probability distributions

## How is a quantum state traditionally described?

- A density matrix describes the statistical state of a system in quantum mechanics. Everything we can possibly know about a quantum system is encoded in the density matrix.
- A quantum state is a positive semidefinite, Hermitian operator of trace 1 acting on the state space.
- For one qubit, all the states live in the Bloch sphere.





How to represent a quantum state with only probability?

## Measurements: Positive operator-valued measure (POVM)

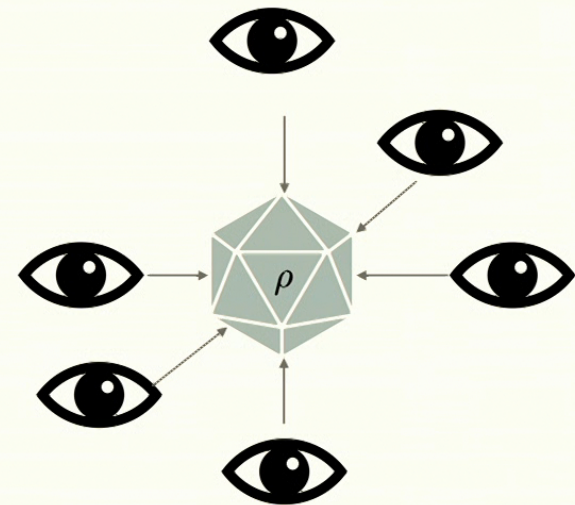
- ▶ Born Rule  $P(\mathbf{a}) = \text{Tr } \rho M^{\mathbf{a}}$  quantum theory  $\leftrightarrow$  experiment
- ▶ POVM elements  $\mathbf{M} = \{M^{(a)} \mid a \in \{1, \dots, m\}\}$
- ▶ Positive semidefinite operators  $\sum_i M^{(a_i)} = \mathbb{1}$

### Informationally complete POVM

- The measurement statistics  $P(\mathbf{a})$  contains all of the information about the state.
- Relation between  $\rho$  and distribution  $P(\mathbf{a})$  can be inverted

### Take a single qubit POVM and make a tensor product

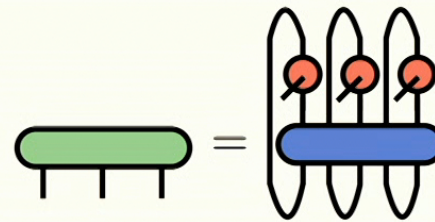
$$\mathbf{M} = \{M^{(a_1)} \otimes M^{(a_2)} \otimes \dots \otimes M^{(a_N)}\}_{a_1, \dots, a_N}$$



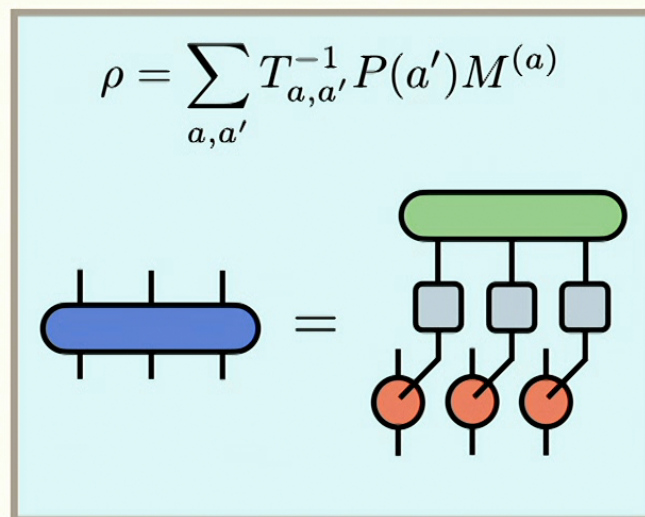


## Graphical notation and inverse

**Born Rule**  $P(\mathbf{a}) = \text{Tr} \rho M^{\mathbf{a}}$



**Informationally completeness  $\longleftrightarrow$  this relation can be inverted**



Positivization/factorization  
of the quantum state

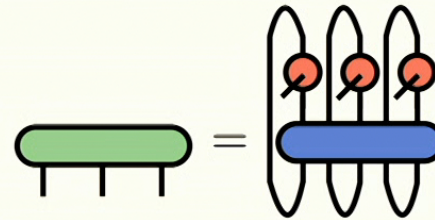
← A probability distribution

×

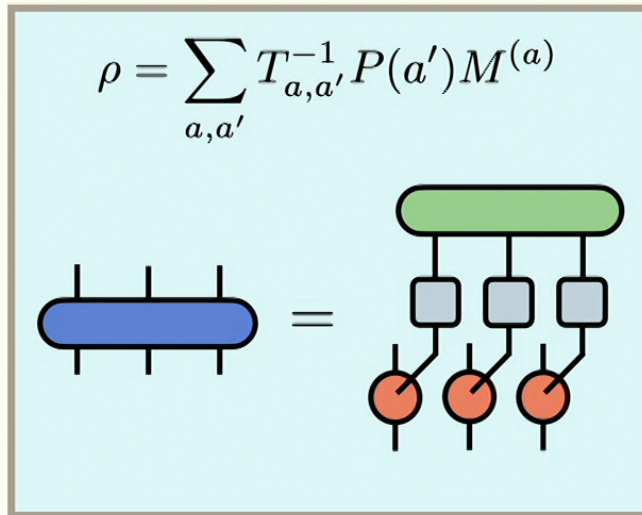
← Tensor product of  
single-qubit tensors

# Graphical notation and inverse

**Born Rule**  $P(\mathbf{a}) = \text{Tr} \rho M^{\mathbf{a}}$



**Informationally completeness**  $\longleftrightarrow$  **this relation Can be inverted**



- Tomography
- Unitary evolution
- Schrodinger equation
- Linblad equation
- Measurements

$$i \frac{\partial \rho}{\partial t} = [\mathcal{H}, \rho] \quad \longleftrightarrow \quad \text{BORN RULE} \quad \longleftrightarrow \quad i \frac{\partial P(\mathbf{a}'', t)}{\partial t} = \sum_{\mathbf{a}, \mathbf{a}'} \text{Tr} \left( [\mathcal{H}, M^{(\mathbf{a})}] M^{(\mathbf{a}'')} T_{\mathbf{a}, \mathbf{a}'}^{-1} P(\mathbf{a}', t) \right)$$



## Insight: parametrize statistics of measurements and invert

Instead of parameterizing  $\rho$ , create an autoregressive model of  $P(\mathbf{a})$

$$P(\mathbf{a}) = \text{Tr } \rho M^{\mathbf{a}}$$

$$\rho = \sum_{a,a'} T_{a,a'}^{-1} P(a') M^a \longrightarrow \rho_{\text{model}} = \sum_{a,a'} T_{a,a'}^{-1} P_{\text{model}}(a') M^a$$

- Carrasquilla, Torlai, Melko, Aolita. [Reconstructing quantum states with generative models](#). Nature Machine Intelligence 1, 200 (2019)
- Peter Cha, Paul Ginsparg, Felix Wu, Juan Carrasquilla, Peter L. McMahon, Eun-Ah Kim. [Attention based quantum tomography](#). Mach. Learn.: Sci. Technol. **3** 01LT01 (2022)
- J. Carrasquilla, D. Luo, F. Pérez, A. Milsted, B. K. Clark, M. Volkovs, L. Aolita. [Probabilistic Simulation of Quantum Circuits using a deep-learning architecture](#). Phys. Rev. A **104**, 032610 (2021)
- D. Luo, Z. Chen, J. Carrasquilla, B.K. Clark. [Autoregressive Neural Network for Simulating Open Quantum Systems via a Probabilistic Formulation](#). Phys. Rev. Lett. **128**, 090501 (2022)

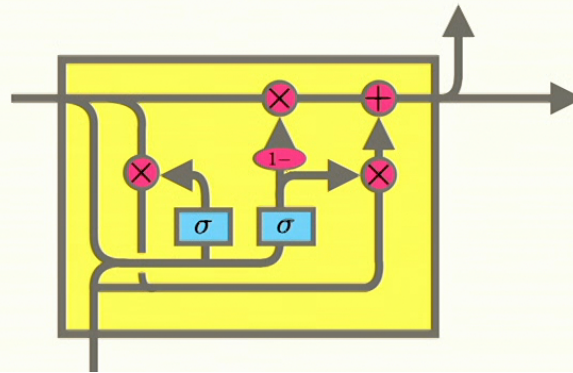
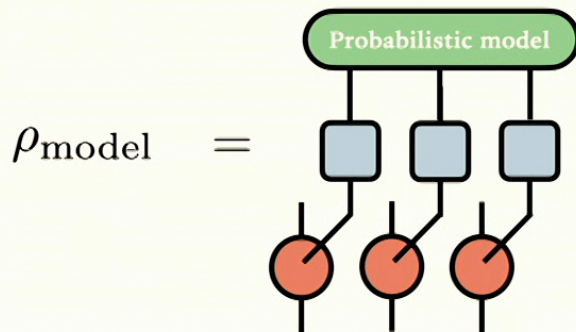
# Insight: parametrize statistics of measurements and invert

$$P(\mathbf{a}) = \text{Tr } \rho M^{\mathbf{a}} \quad \Rightarrow \text{Create a neural model of } P(\mathbf{a})$$

$$P_{\text{model}}(\mathbf{a}) \longrightarrow$$

Autoregressive models (RNNs and transformer)

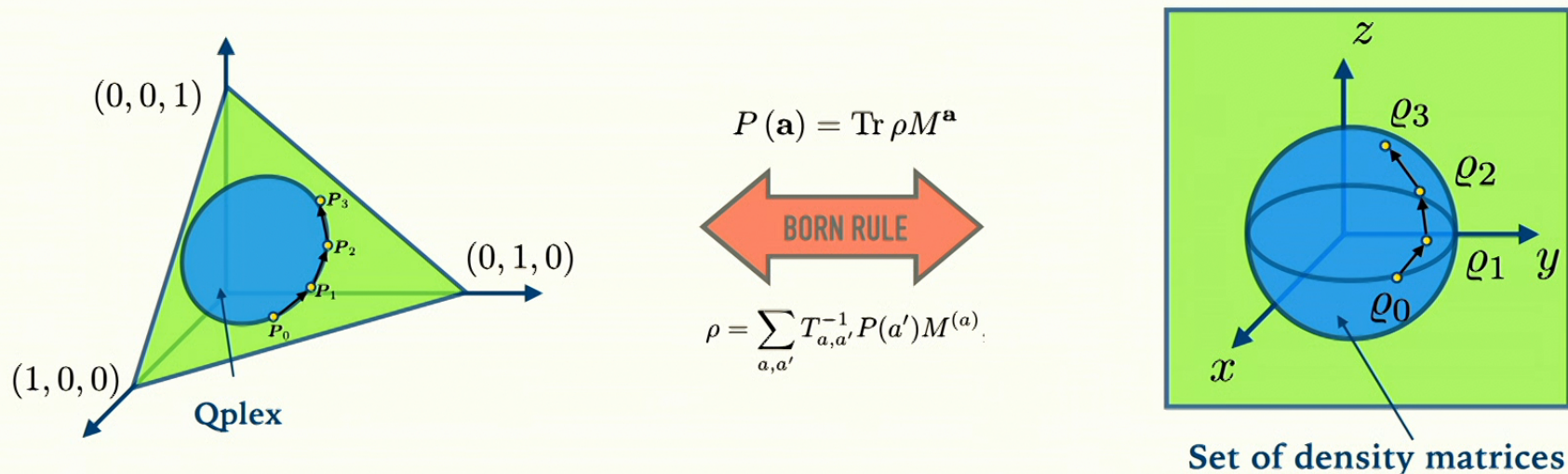
1. Allow for exact sampling
2. Tractable density  $P_{\text{model}}(\mathbf{a})$





# Quantum states represented as probabilities

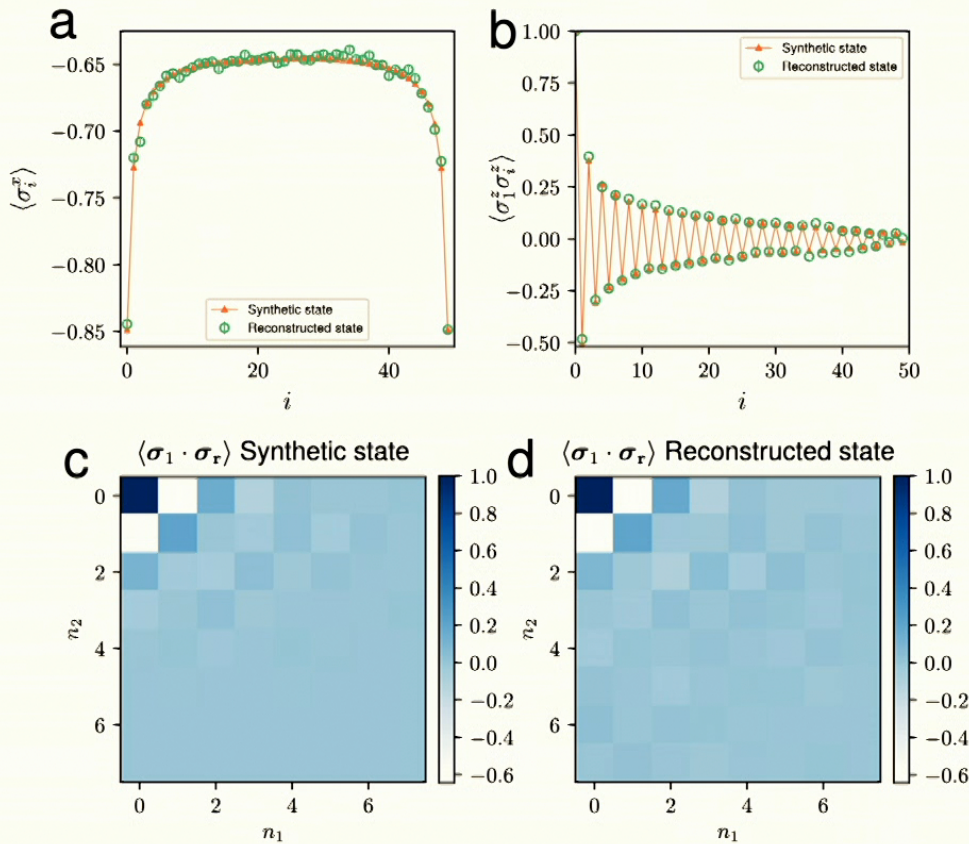
We go from a “Bloch Body” of quantum states to probability distributions — points of the standard simplex in high dimensions



Note that the simplex is too big and general— some distributions do not correspond to quantum states ●



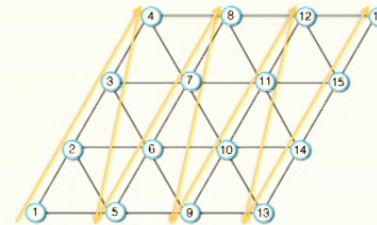
# Learning Ground states from simulated projective measurements (RNN)



$$\mathcal{H} = J \sum_{ij} \sigma_i^z \sigma_j^z + h \sum_i \sigma_i^x$$

N=50 spins. P(a) is a deep (3 layer GRU) recurrent neural network language model.

$$H = J \sum_{i,j} \sigma_i \cdot \sigma_j$$



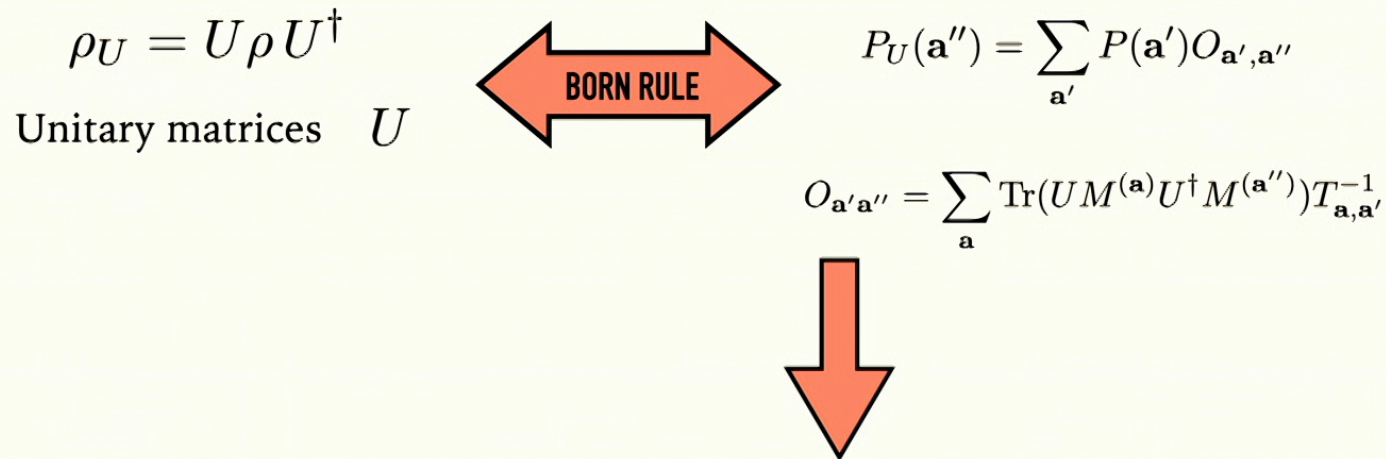
Carrasquilla, Torlai, Melko, Aolita. Nature Machine Intelligence 1, 200 (2019)



# Unitary dynamics

# UNITARY DYNAMICS THROUGH LANGUAGE MODELS

## Unitary dynamics



Probabilistic unitaries: **Somewhat**  
(or quasi-) stochastic matrices

- Probabilistic simulation of quantum circuits using a deep-learning architecture. Carrasquilla, Luo, Perez, Milstead, Clark, Volkovs, Aolita. PHYSICAL REVIEW A **104**, 032610 (2021)



# Dynamics of closed and open quantum systems

Linblad equation

$$\dot{\rho} = \mathcal{L}\rho \equiv -i[H, \rho] + \sum_k \frac{\gamma_k}{2} \left( 2\Gamma_k \rho \Gamma_k^\dagger - \{\rho, \Gamma_k^\dagger \Gamma_k\} \right)$$



Born rule

$$\dot{p}(\mathbf{a}) = \sum_{\mathbf{b}} p(\mathbf{b}) L_{\mathbf{a}}^{\mathbf{b}} = \sum_{\mathbf{b}} p(\mathbf{b}) (A_{\mathbf{a}}^{\mathbf{b}} + B_{\mathbf{a}}^{\mathbf{b}})$$

High-D differential Equation

$$A_{\mathbf{a}}^{\mathbf{b}} = -i \text{Tr} \left( H [N^{(\mathbf{b})}, M_{(\mathbf{a})}] \right); \quad \text{Unitary dynamics}$$

$$B_{\mathbf{a}}^{\mathbf{b}} = \sum_k \frac{\gamma_k}{2} \text{Tr} \left( 2\Gamma_k N^{(\mathbf{b})} \Gamma_k^\dagger M_{(\mathbf{a})} - \Gamma_k^\dagger \Gamma_k \{N^{(\mathbf{b})}, M_{(\mathbf{a})}\} \right). \quad \text{Dissipation}$$

$$N^{(\mathbf{b})} = \sum_{\mathbf{a}} M_{(\mathbf{a})} T_{\mathbf{a}\mathbf{b}}^{-1} \quad T_{\mathbf{a},\mathbf{b}} = \text{Tr} (M_{\mathbf{a}} M_{\mathbf{b}}) \quad \text{Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K. Clark. Phys. Rev. Lett. 128, 090501 (2022)}$$

# Integration of the differential equation

## Strategy:

- We take inspiration from the trapezoidal rule
- Design an objective function that we optimize with gradient descent.
- If the optimization succeeds, the trapezoidal rule is implemented exactly.
- **No data required** for optimizing the model—everything is based on the differential equation



# Quantum dynamics of open quantum systems

Trapezoidal rule:

Suppose that we want to solve the differential equation

$$y' = f(t, y)$$

The trapezoidal rule is given by the formula

$$y_{n+1} = y_n + \frac{1}{2}\tau (f(t_n, y_n) + f(t_{n+1}, y_{n+1}))$$

$$\tau = t_{n+1} - t_n$$

We can in principle apply this but we have exponentially many values of  $y$ .  
Our distributions have way too many outcomes to use the rule directly

# Quantum dynamics of open quantum systems

Trapezoidal rule

$$\sum_b (\delta_a^b - \tau L_a^b) p_{t+2\tau}(b) = \sum_b (\delta_a^b + \tau L_a^b) p_t(b) \quad \text{For each component } \mathbf{a}$$

We designed a cost function (sum over  $\mathbf{a}$ , subtract and take abs, sample)

Samples from model

$$C = \frac{1}{N_s} \sum_{\mathbf{a} \sim p_{\theta(t+2\tau)}} \frac{1}{p_{\theta(t+2\tau)}(\mathbf{a})} \left| \sum_b [p_{\theta(t+2\tau)}(b) (\delta_a^b - \tau L_a^b) - p_{\theta(t)}(b) (\delta_a^b + \tau L_a^b)] \right|$$

Two models: transformers

Requires tractable Likelihood

Transformers, RNNs, MPS offer this flexibility

If C is optimized to zero perfectly, then the trapezoidal rule is implemented exactly

Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K. Clark. Phys. Rev. Lett. 128, 090501 (2022)



# Quantum dynamics of open quantum systems

Fixed point of the dissipative dynamics —  $dp/dt=0$

$$\dot{p}(\mathbf{a}) = \sum_{\mathbf{b}} p(\mathbf{b}) L_{\mathbf{a}}^{\mathbf{b}} = \sum_{\mathbf{b}} p(\mathbf{b}) (A_{\mathbf{a}}^{\mathbf{b}} + B_{\mathbf{a}}^{\mathbf{b}})$$

Dissipation

Unitary dynamics

$$\left\| \frac{\partial p}{\partial t} \right\|_1 \approx \frac{1}{N_s} \sum_{\mathbf{a} \sim p_{\theta}} \frac{|\sum_{\mathbf{b}} p_{\theta}(\mathbf{b}) L_{\mathbf{a}}^{\mathbf{b}}|}{p_{\theta}(\mathbf{a})},$$

Expensive in principle

Easy for transformer/RNN

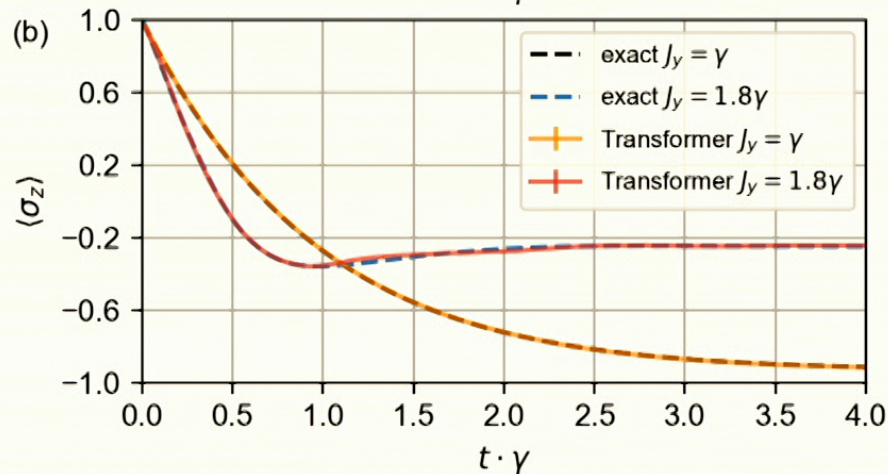
Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K. Clark. Phys. Rev. Lett. **128**, 090501 (2022)



# Quantum dynamics of open quantum systems

## Heisenberg model in 2 dimensions

### 2D Heisenberg model



$$H = \sum_{\langle i,j \rangle} \sum_{w=x,y,z} J_w \sigma_i^{(w)} \sigma_j^{(w)} + B \sum_k \sigma_k^{(z)}.$$

$$\Gamma_k = \sigma_k^{(-)} = \frac{1}{2}(\sigma_k^{(x)} - i\sigma_k^{(y)})$$

$$\dot{\rho} = \mathcal{L}\rho \equiv -i[H, \rho] + \sum_k \frac{\gamma_k}{2} \left( 2\Gamma_k \rho \Gamma_k^\dagger - \{\rho, \Gamma_k^\dagger \Gamma_k\} \right)$$

### 3x3 in 2D

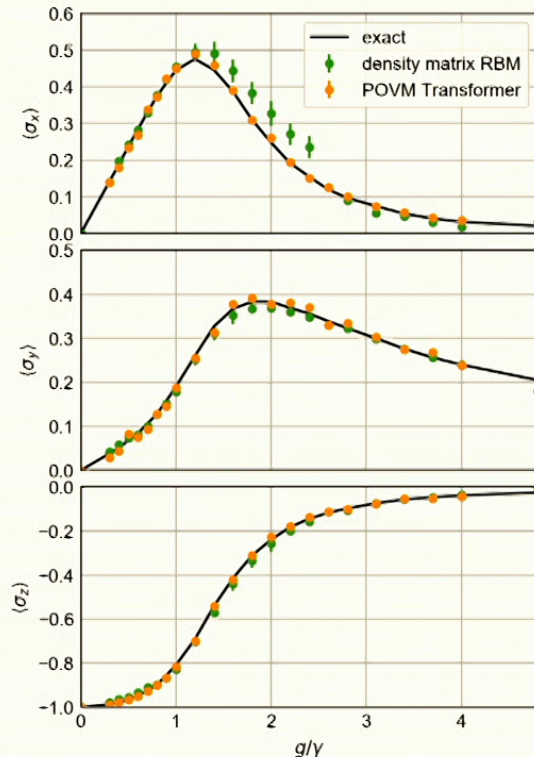
Figure 2. The expectation value  $\langle \sigma_z \rangle$  as a function of time (a) for the 1D Heisenberg model with  $B = \gamma$ ,  $J_x = 2\gamma$ ,  $J_y = 0$ , and  $J_z = \gamma$  using a time step  $\tau = 0.005\gamma^{-1}$ . The initial state is the product state  $\prod_{i=1}^N |\leftarrow\rangle$  ( $\langle \sigma_y \rangle = -1$ ). (b) for the  $3 \times 3$  Heisenberg model with  $B = 0$ ,  $J_x = \gamma$ ,  $J_y = 1.0\gamma, 1.8\gamma$ , and  $J_z = 0.9\gamma$  using a time step time step  $\tau = 0.008\gamma^{-1}$ . The initial state is a product state  $\prod_{i=1}^N |\uparrow\rangle$  ( $\langle \sigma_z \rangle = 1$ ). Both models use periodic boundary conditions. Exact curves are produced using QuTip [82, 83]. The Transformer has one encoder layer and 32 hidden dimensions, and is trained using a forward-backward trapezoid method with a sample size  $N_s = 12000$ .

Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K. Clark. Phys. Rev. Lett. 128, 090501 (2022)



# Quantum dynamics of open quantum systems

## Quenches in the TFIM with dissipation



N=16 qubits 
$$H = \frac{V}{4} \sum_{\langle i,j \rangle} \sigma_i^{(z)} \sigma_j^{(z)} + \frac{g}{2} \sum_k \sigma_k^{(x)}$$

$$\Gamma_k = \sigma_k^{(-)} = \frac{1}{2}(\sigma_k^{(x)} - i\sigma_k^{(y)}) \quad \dot{\rho} = \mathcal{L}\rho \equiv -i[H, \rho] + \sum_k \frac{\gamma_k}{2} (2\Gamma_k \rho \Gamma_k^\dagger - \{\rho, \Gamma_k^\dagger \Gamma_k\})$$

Steady state simulation  $\dot{\rho} = 0$

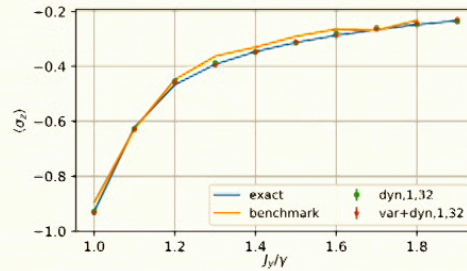
Figure 3. Variational steady state solution for a 16-site TFIM chain with periodic boundary conditions and  $V = 2\gamma$  (orange dots). The initial state is a product state  $\prod_{i=1}^N |\uparrow\rangle$  ( $\langle \sigma_z \rangle = 1$ ). The Transformer has one encoder layer and 32 hidden dimensions, and is trained using Adam [79] in 500 iterations with  $N_s = 12000$ . Green points are the fixed point solution representing the density matrix as an RBM; both the exact curve (black line) and density matrix results are digitized from Ref. 60.

RBM from: F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, *Phys. Rev. Lett.* **122**, 250503 (2019).

Di Luo, Zhuo Chen, Juan Carrasquilla, and Bryan K. Clark. *Phys. Rev. Lett.* **128**, 090501 (2022)

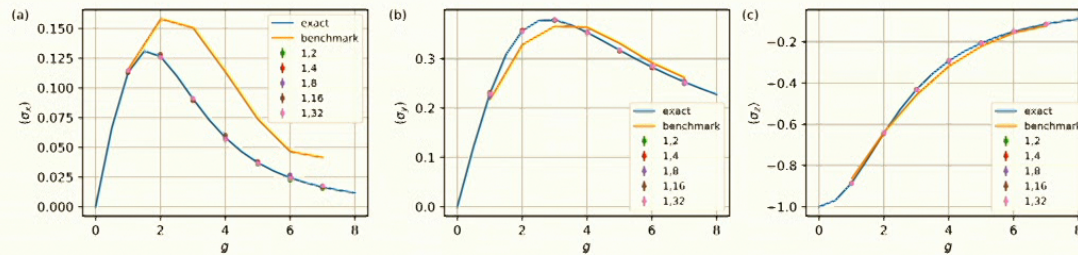


# Benchmark against classical and quantum algorithms



A. Nagy and V. Savona, Phys. Rev. Lett. 122, 250501 (2019).

Figure S6.  $3 \times 3$  Heisenberg model benchmarked with Ref. 8. This system is the same as in Fig. 4 in the main paper. The exact curve (blue) is generated using QuTiP [2, 3]. The benchmark curve (orange, Ref. 8), is based on an RBM. Our results (green and red) are the same as in the main paper. The two numbers in the legend specify the number of layers and hidden dimensions  $n_d$ .



N. Yoshioka, Y. O. Nakagawa, K. Mitarai, and K. Fujii, “Variational quantum algorithm for non-equilibrium steady states,” (2019), arXiv:1908.09836 [quant-ph]

Figure S7. 8 qubit transverse-field Ising model benchmarked with Ref. 9. The system Hamiltonian is the same as in Fig. 3 in the main paper but with open boundary condition, with  $V = 2$  and  $g$  shown in the figure. The jump operators are slightly different from the main paper such that there are two different jump operators with  $\Gamma^{(1)} = \sigma^{(-)}$ , and  $\Gamma^{(2)} = \sigma^{(z)}$ . The corresponding dissipation rates are  $\gamma^{(1)} = 4$  and  $\gamma^{(2)} = 2$ . Ref. 9 uses a slightly different convention resulting in a difference in  $g$  and  $\gamma$  which we have verified by matching their curves in our convention. The exact curve (blue) is generated using QuTiP [2, 3] and is superimposed to the figure in Ref. 9 to check for the correctness of the parameters. The benchmark curve (orange) is from Ref. 9. The two numbers in the legend mean number of layers and hidden dimensions, respectively. We note that while the neural network is not designed to work for a hidden dimension  $n_d$  less than 8, the results presented here are still significantly better than Ref. 9 for  $n_d < 8$ .



## Conclusions

- We have introduced a probabilistic representation of the quantum state
- Combined with powerful probabilistic models, it allows for simulations of circuits and open quantum dynamics.
- Good: Optimization problem is relatively easy
- But: requires a lot of samples/tricks to improve variance in the gradients
- Lots of other things to do: architectures to explore (eg neuromorphic, other ML models), different/more suitable objective functions, implementation of symmetries, different systems, etc.
- Applications beyond quantum mechanics—how to efficiently solve high-dimensional probabilistic differential equations with autoregressive neural networks, eg Chemical Master Equation, other Boltzmann equations.