NOVEMBER 4-5, 2024

NexTGen

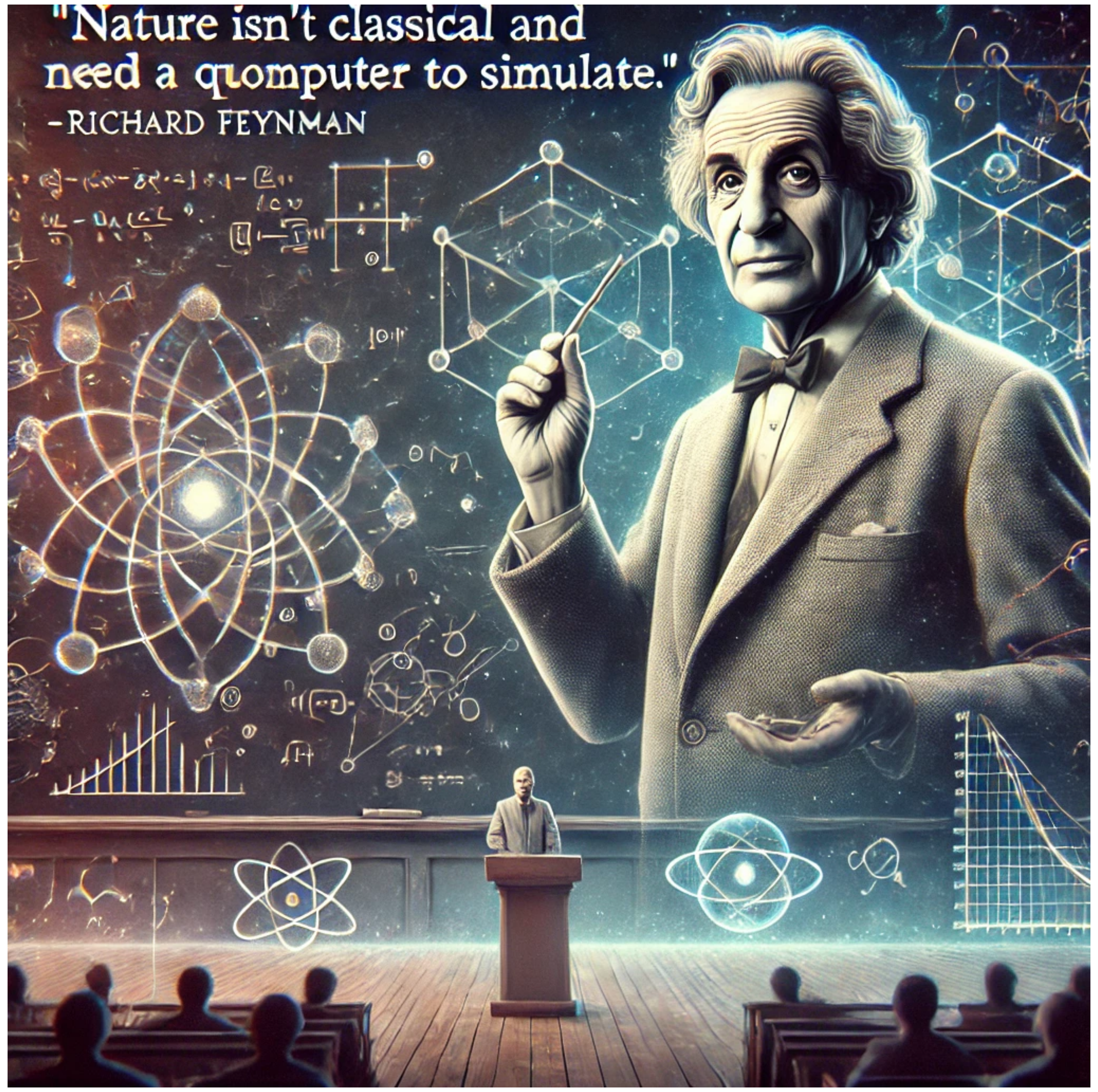
Workshop on Tensor Networks and (Quantum) Machine Learning for High-Energy Physics

# Provable quantum learning advantages for quantum-generated data

Vedran Dunjko
applied Quantum algorithms (aQa) Leiden
v.dunjko@liacs.leidenuniv.nl

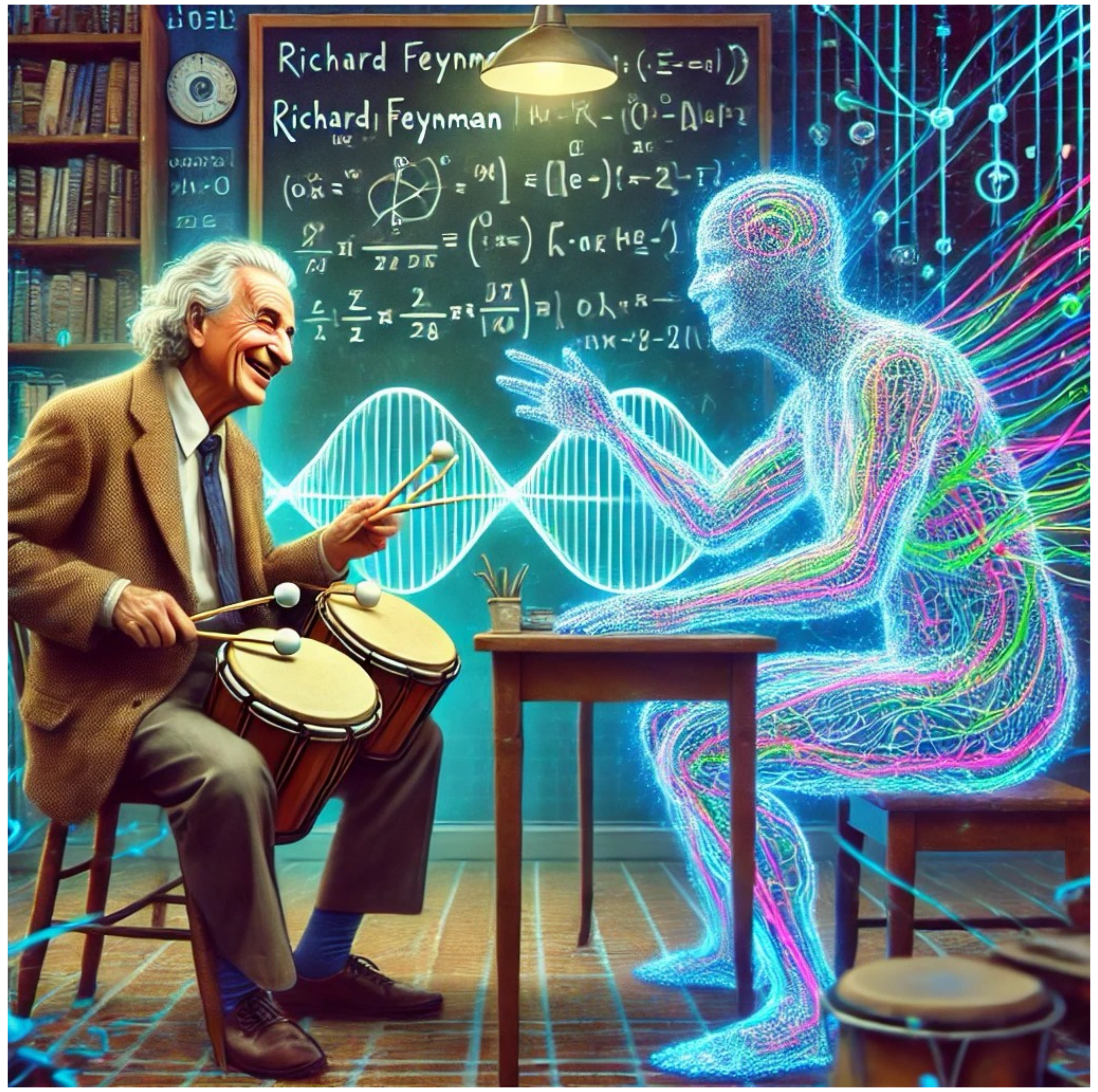
Do you like silent movies?

in anticipation of that
I flooded the slides with way too much text. Soz about that.
But its for the best



Made by my bf chatty

Nature isn't classical dammit, *and if you want to make a simulation of nature*
you'd better make *it quantum mechanical*
and by golly it's a wonderful problem because it doesn't look so easy

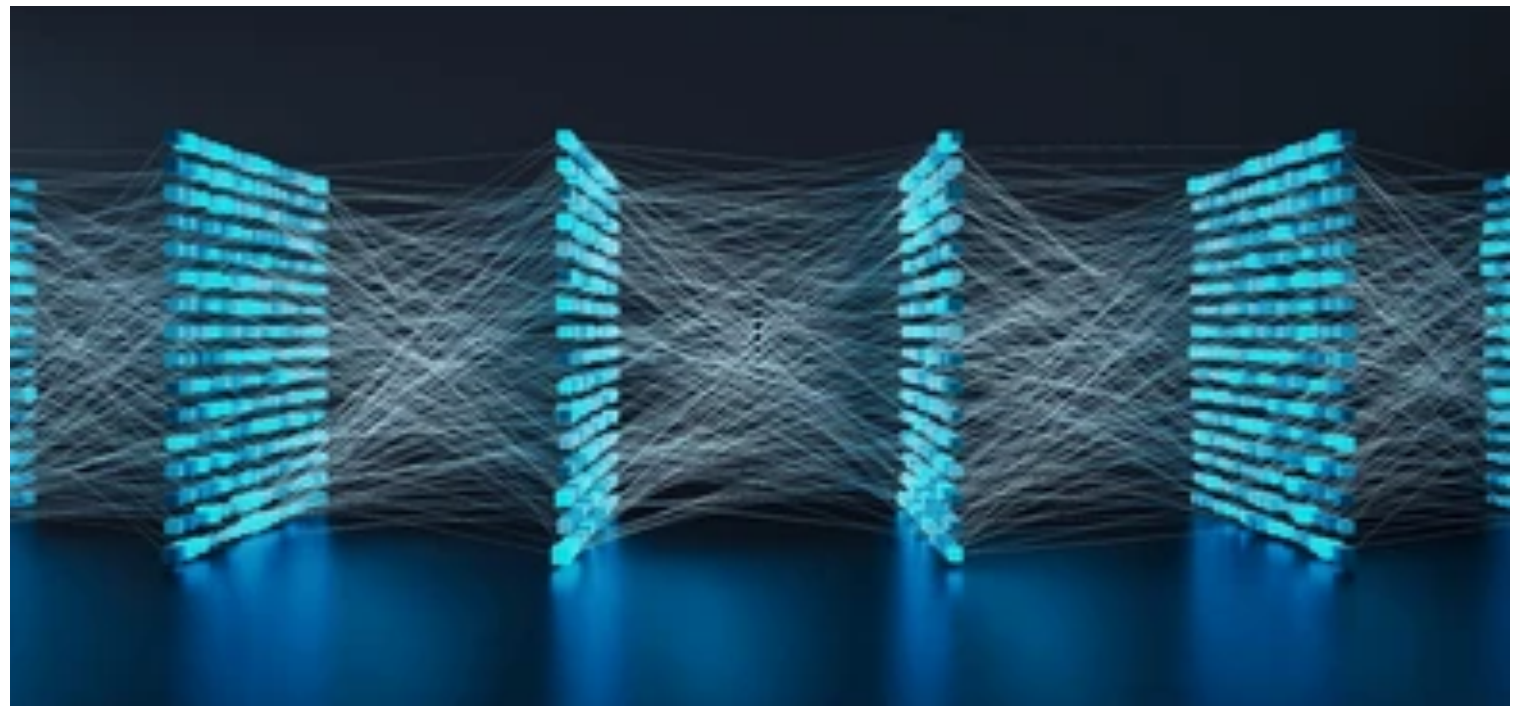



Made by my bf chatty

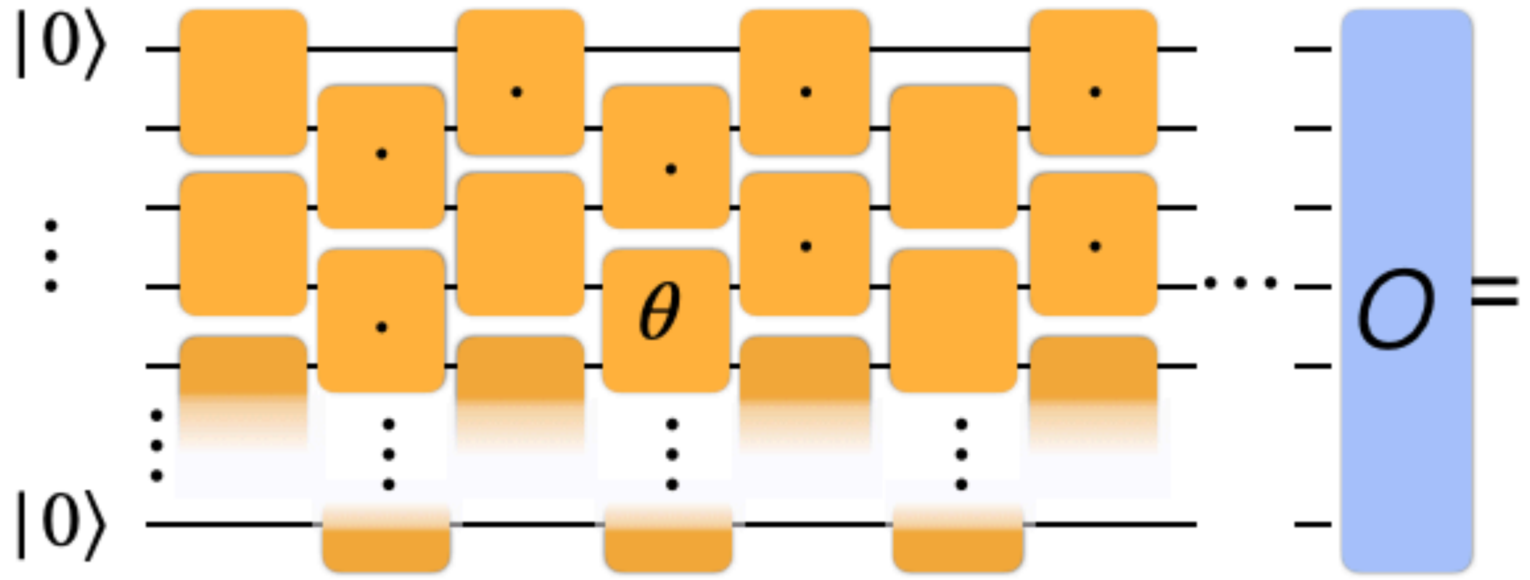
Nature isn’t classical dammit, *and if you want to learn its properties,*
you’d better make a *quantum learner,*
and by golly it’s a wonderful problem because it doesn’t look so easy

Would Feynman agree? Is it *true*?

# Machine learning vs quantum machine learning (for HEP)



vs




## Types of evidence for which is "better"

### Practice

-is simply (generically) better

-large (generic) benchmarks

-small (generic) benchmarks

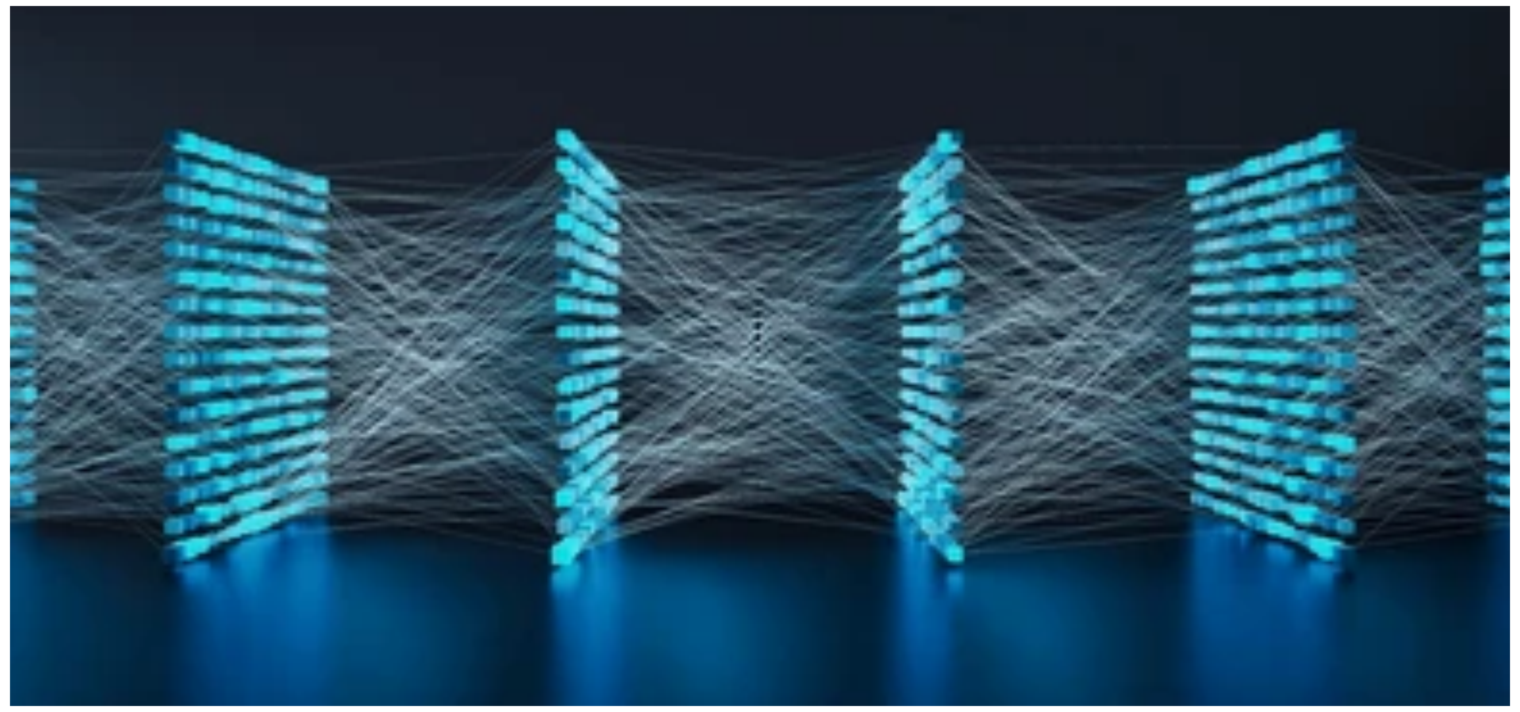
### Theory

-proof: is (generically) *better*
*e.g. identical but faster*

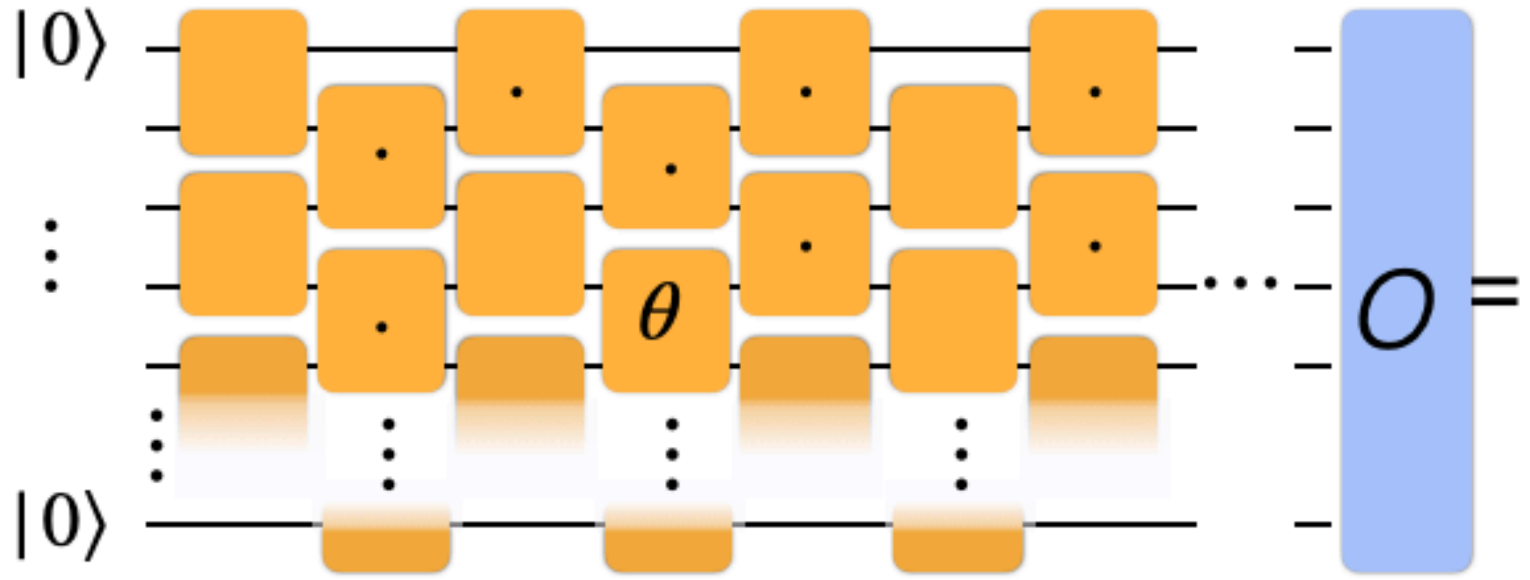
-proof: is (generically) *better*
*yet different*

-proof: better in special cases
or w.r.t. special *metrics*

# Machine learning vs quantum machine learning (for HEP)



vs




## Types of evidence for which is "better"

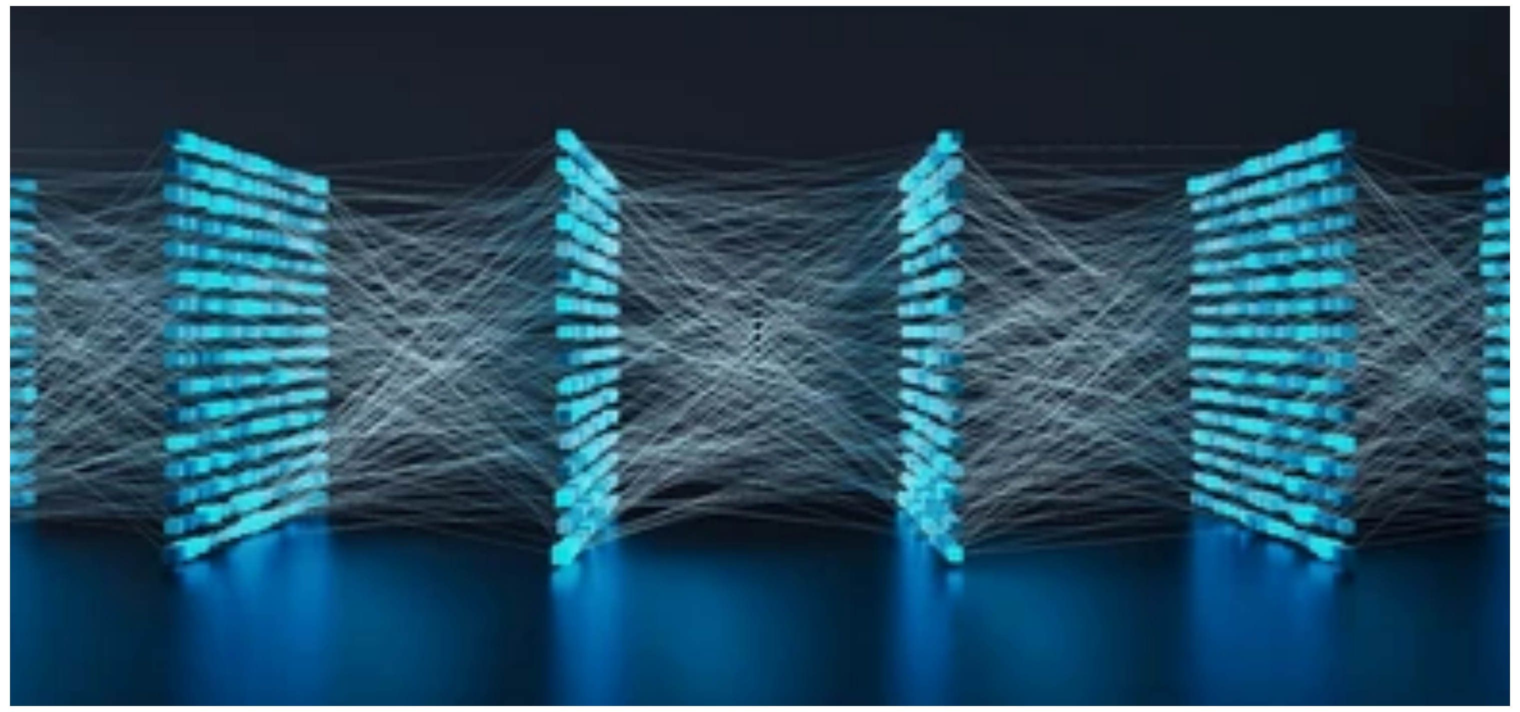
### Practice

| | |
|---|---|
| -is simply (generically) better | *currently practically impossible* |
| -large (generic) benchmarks | *currently practically impossible* |
| -small (generic) benchmarks | *inconclusive - phase transition in size?* |

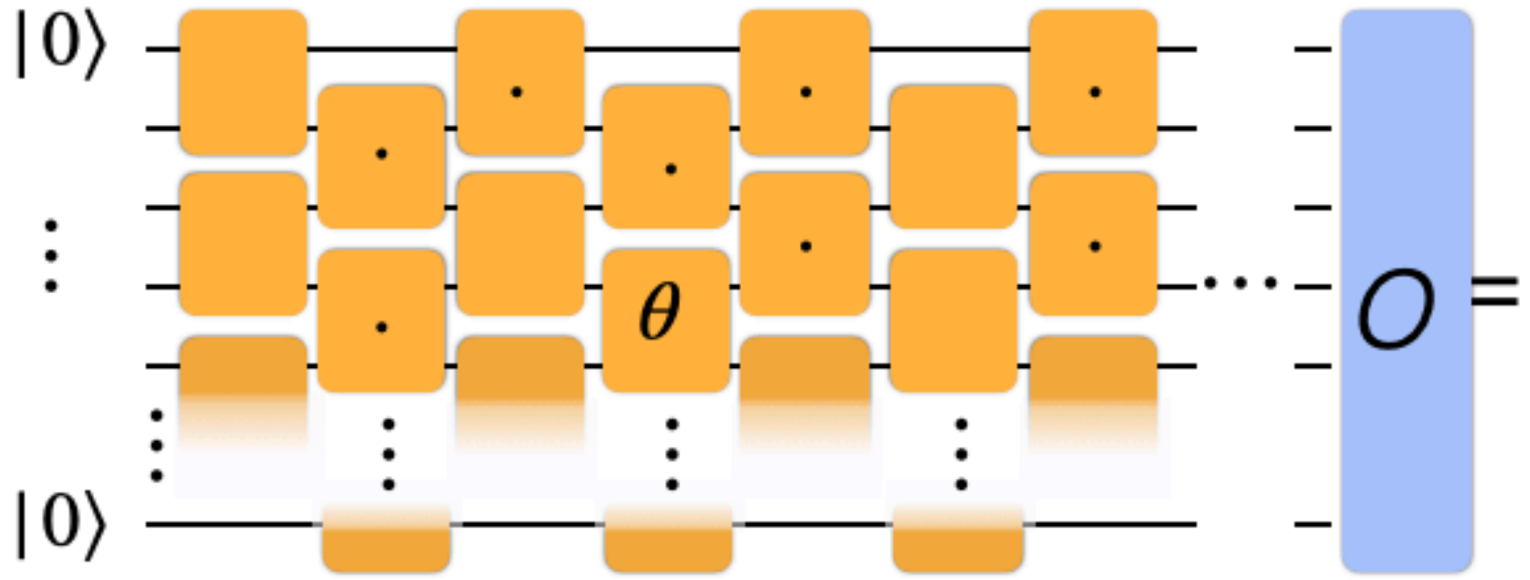
### Theory

| | |
|---|---|
| -proof: is (generically) *better* *e.g. identical but faster* | *rare and *always* with fine print (even if just subquadratic speedups)* |
| -proof: is (generically) *better* *yet different* | *we don't know how to do this!* |
| -proof: better in special cases or w.r.t. special *metrics* | |

# Machine learning vs quantum machine learning (for HEP)



vs




## Types of evidence for which is "better"

### Practice

| | |
|---|---|
| -is simply (generically) better | *currently practically impossible* |
| -large (generic) benchmarks | *currently practically impossible* |
| -small (generic) benchmarks | *inconclusive - phase transition in size?* |

### Theory

| | |
|---|---|
| -proof: is (generically) *better*<br>*e.g. identical but faster* | *rare and *always* with fine print*<br>*(even if just subquadratic speedups)* |
| -proof: is (generically) *better*<br>*yet different* | *we don't know how to do this!* |
| **-proof: better in special cases**<br>**or w.r.t. special *metrics*** | *this we can do today!* |

## Le Menu:

1) Mathematical framework for *learning: Probably Approximately Correct learning*
   *… and types of advantages one could hope to have*

2) *Main result 1:* Generic advantages for "evaluation tasks"
3) Application: learning of observables
4) *Main result 2:* advantages for identification tasks
5) Reflection on potential practical relevance (*spoiler: I have no clue, but at least I have an idea why*)

Main tool: convert learning statements to statements about *complexity theory*

# Learning theory and types of learning advantages

# Machine learning, learning theory, and intuition

-"learning" properties (*of whatever generated the data*) from data

Machine learning, learning theory, and intuition

-“learning” properties (*of whatever generated the data*) from data

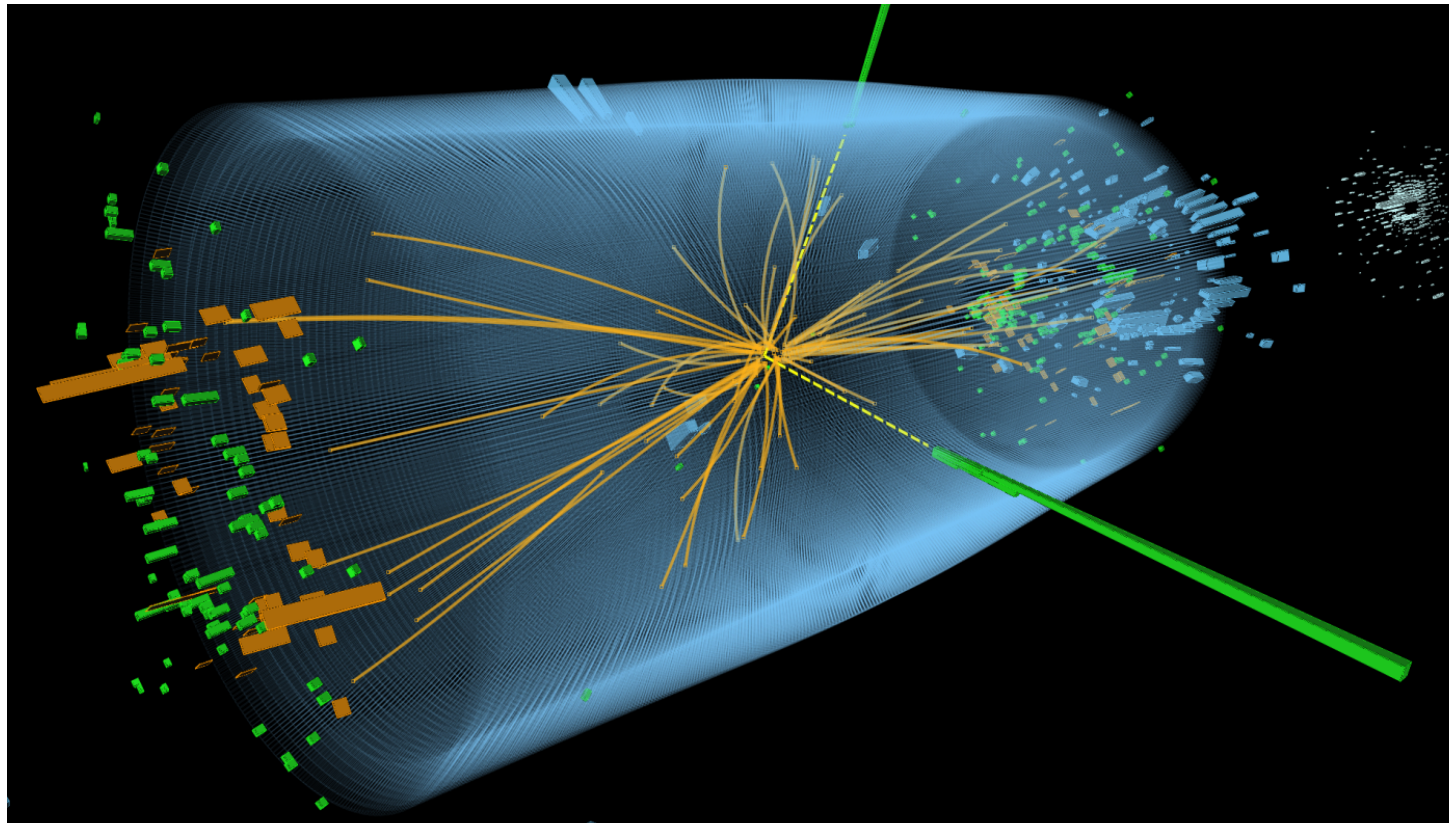
**Example 1:** Supervised Higgs or no Higgs

**Data:**
(x=measurements, y = {Higgs / Background})

**Output:** trained classifier takes on input measurement data (one point)

and outputs *Higgs* or *Background*

Data includes all the realistic noise



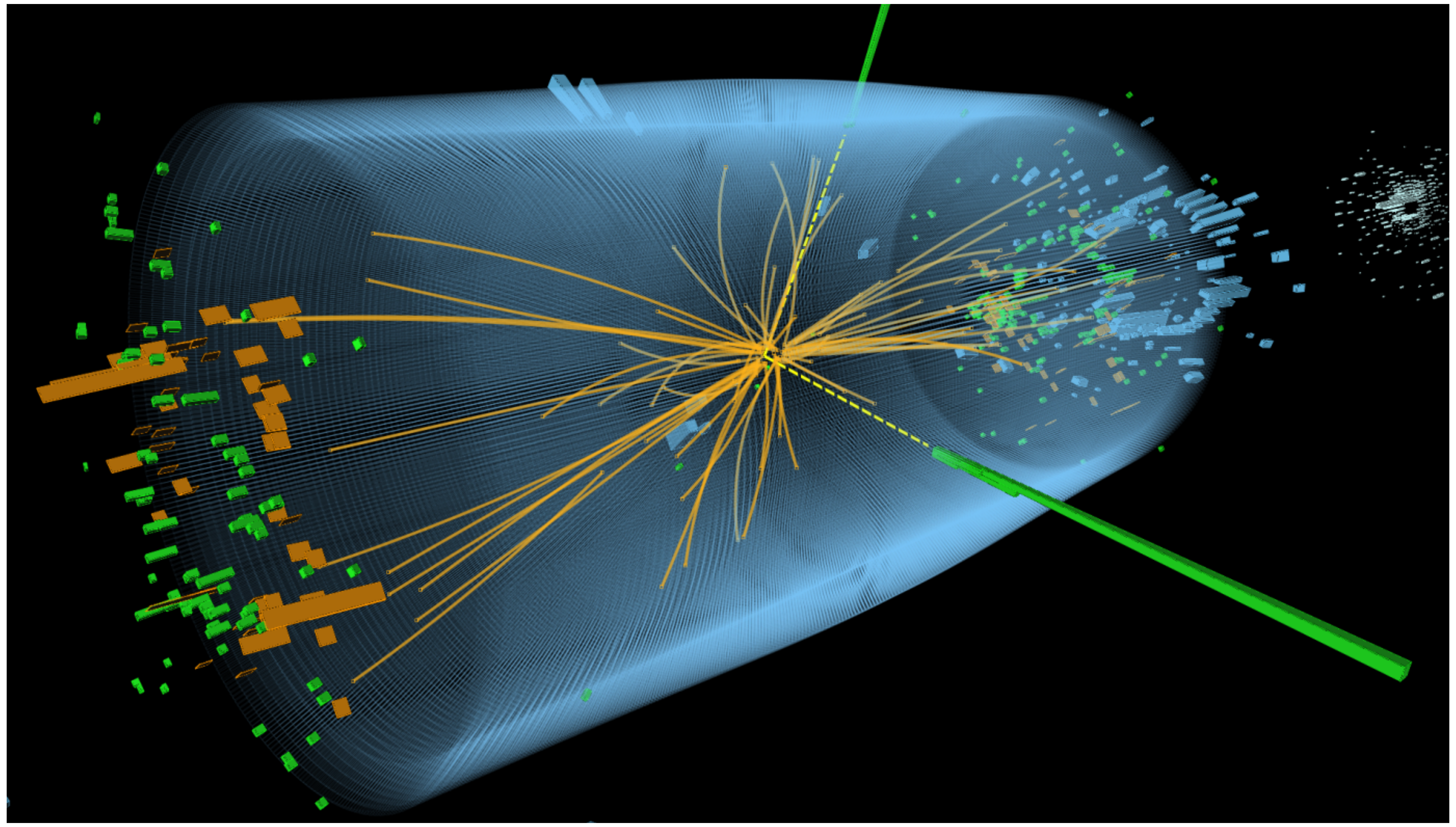
Machine learning, learning theory, and intuition

-"learning" properties (*of whatever generated the data*) from data

**Example 1:** Supervised Higgs or no Higgs

Identifying an **unknown function**
which **we will use**
**and** it will give the right "predictions"

given from examples of what it does



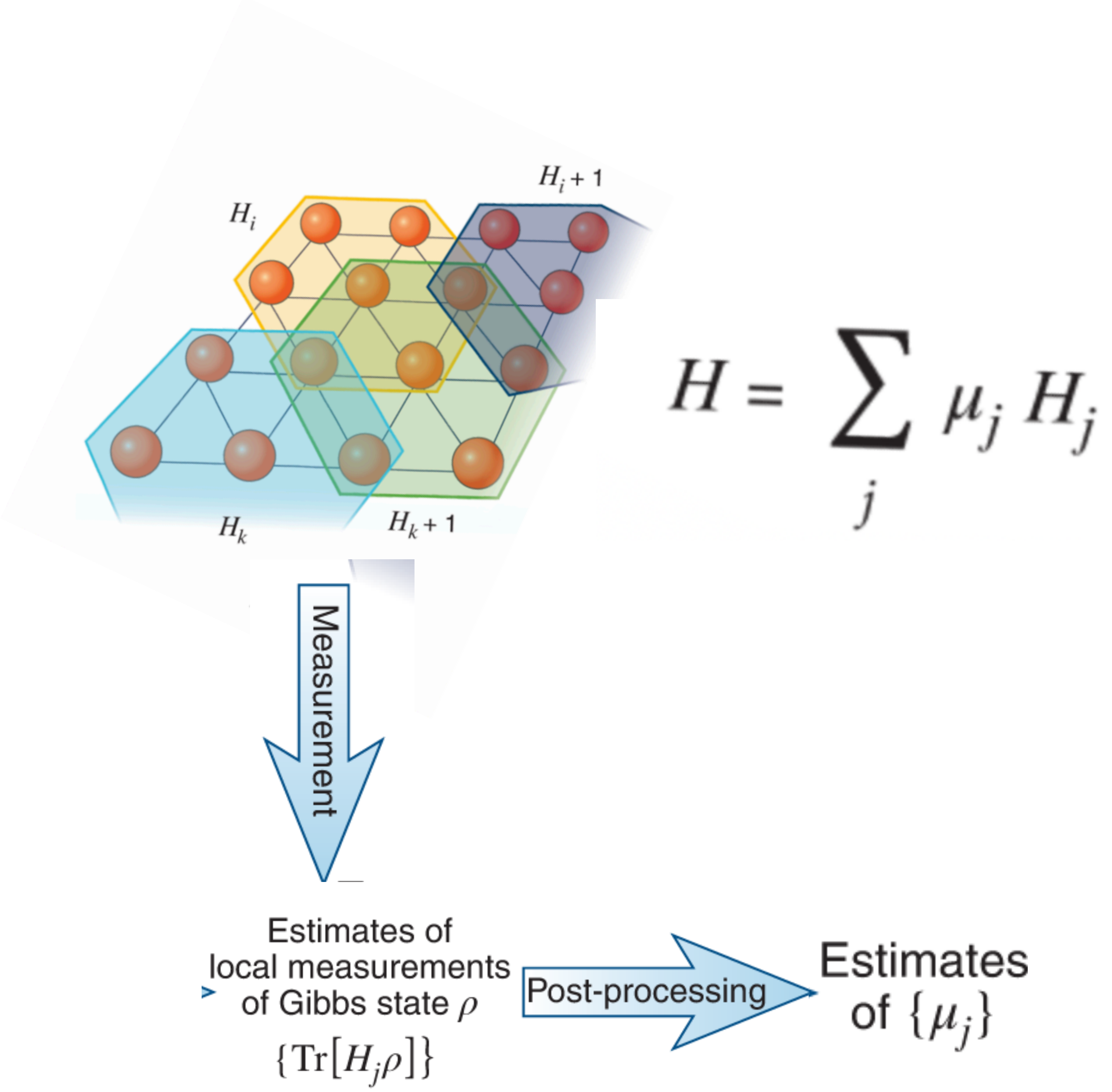
Machine learning, learning theory, and intuition

-"learning" properties (*of whatever generated the data*) from data

**Example 2:** Hamiltonian learning

**Data:**
Measurements of ground or Gibbs states

**Output:** parameters of the Hamiltonian






https://arxiv.org/abs/2004.07266

Machine learning, learning theory, and intuition

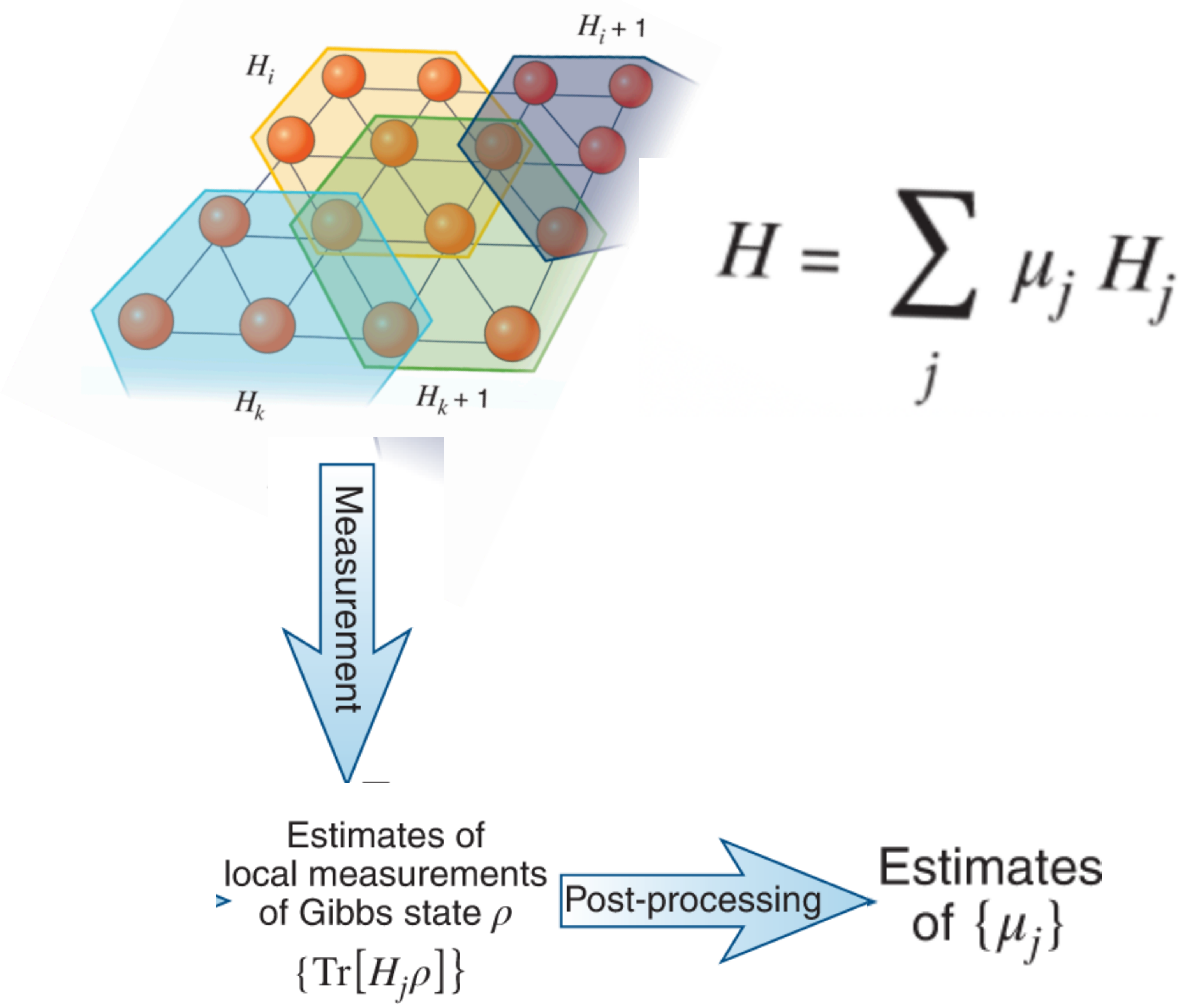
-“learning” properties (*of whatever generated the data*) from data

**Example 2:** Hamiltonian learning

Data was secretly labeled
(term, expectation value)

From the parameters I could
in principle produce new measurement outcomes.

(observable expectation)






https://arxiv.org/abs/2004.07266

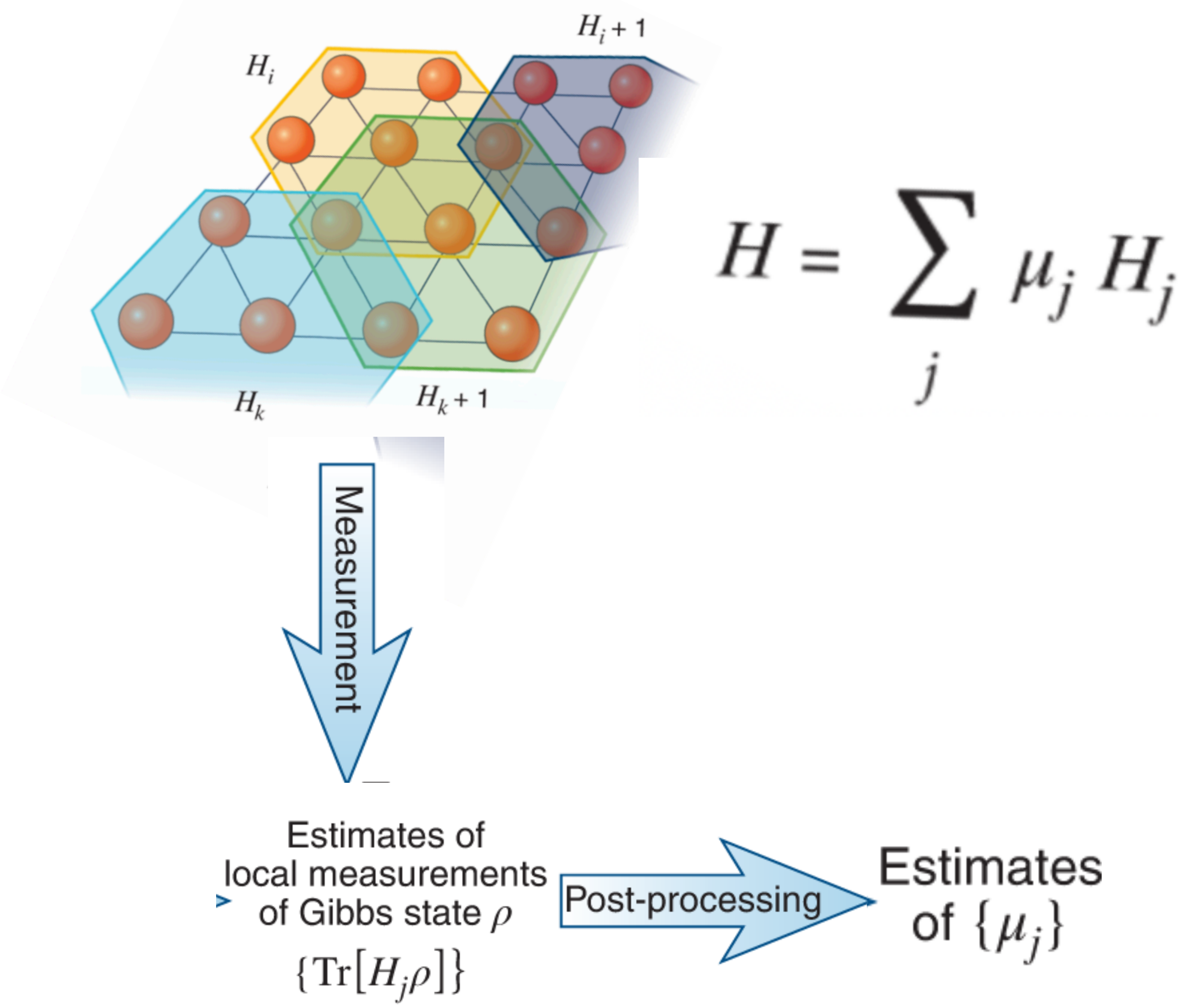
Machine learning, learning theory, and intuition

-"learning" properties (*of whatever generated the data*) from data

**Example 2:** Hamiltonian learning

Learning properties of unknown function...
(specified by the unknown Hamiltonian)

But here just care about its *description*




https://arxiv.org/abs/2004.07266

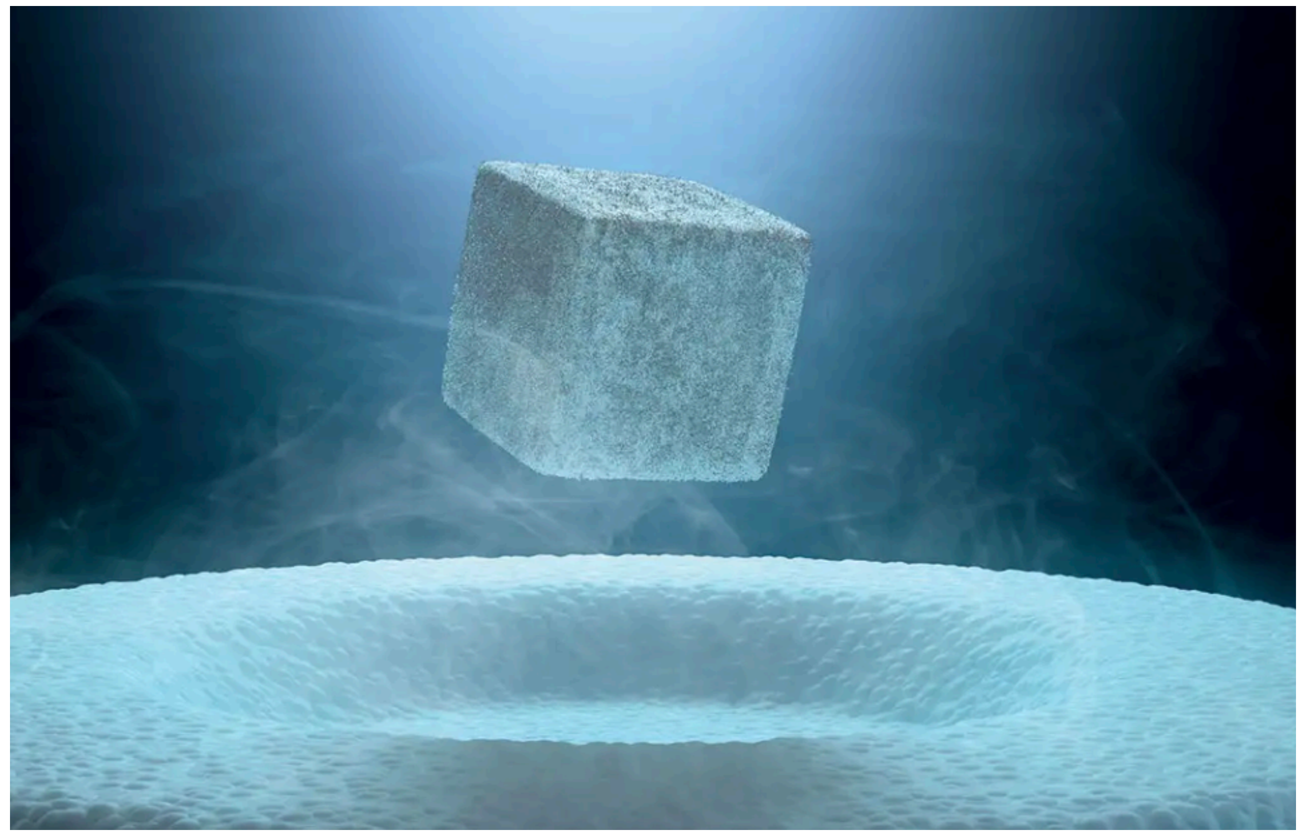
Machine learning, learning theory, and intuition

-"learning" properties (*of whatever generated the data*) from <u>data</u>

**Example 3:** Tc prediction (superconductivity)

**Data:** database of pairs (material, Tc)

**Output:** function which predicts Tc based on (say) chemical composition and structure



Machine learning, learning theory, and intuition

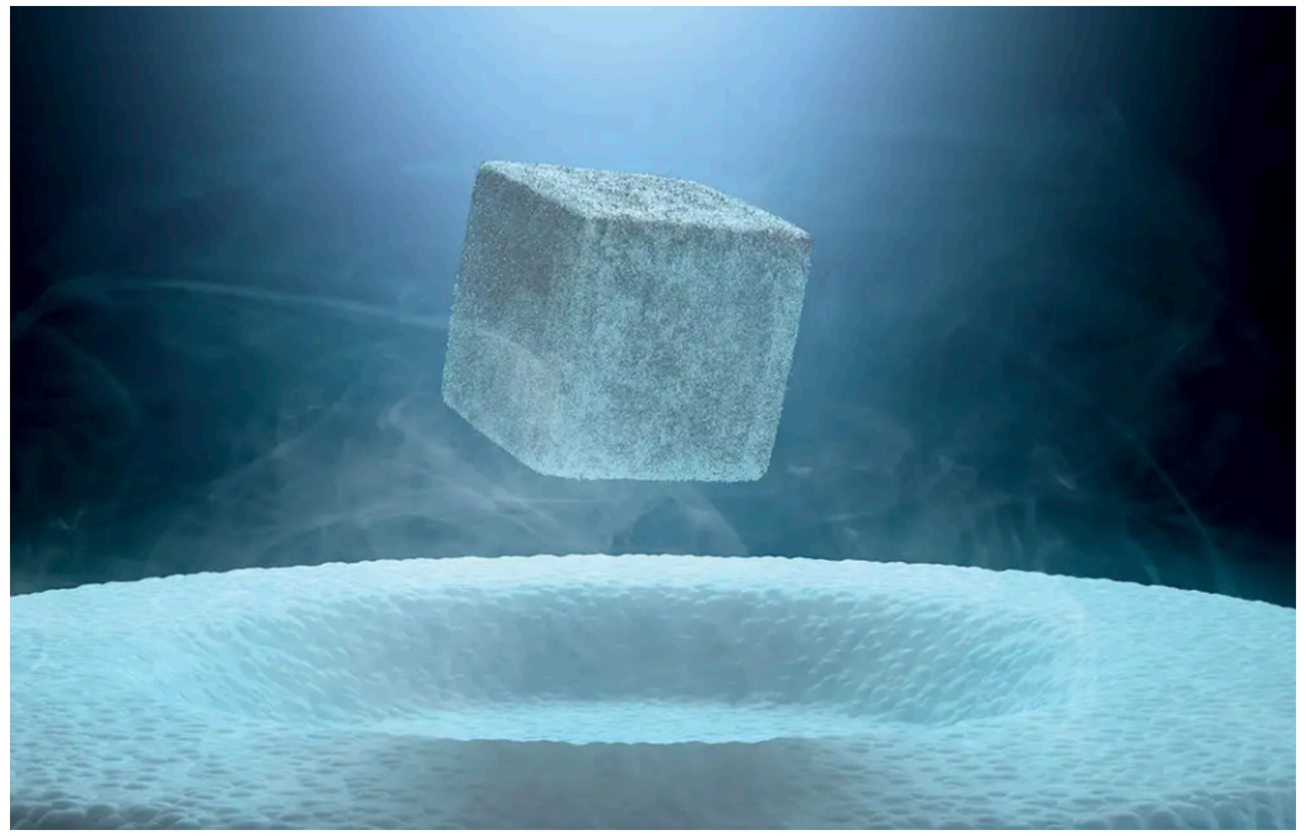
-"learning" properties (*of whatever generated the data*) from data

**Example 3:** Tc prediction (superconductivity)

Here, the function is not *fundamentally unknown.*

*Given* $a\ lot^{a\ lot^{alot}}$ *of compute, you could compute it from* ***first principles***

***We are looking for a much more concise representation of same function***



Machine learning, learning theory, and intuition

-"learning" properties (*of whatever generated the data*) from <u>data</u>

Learn the classifier for the purpose of use ("evaluation")

Learn the classifier properties ("identification")

"Ground truth" can be unknown, partially known, fully known

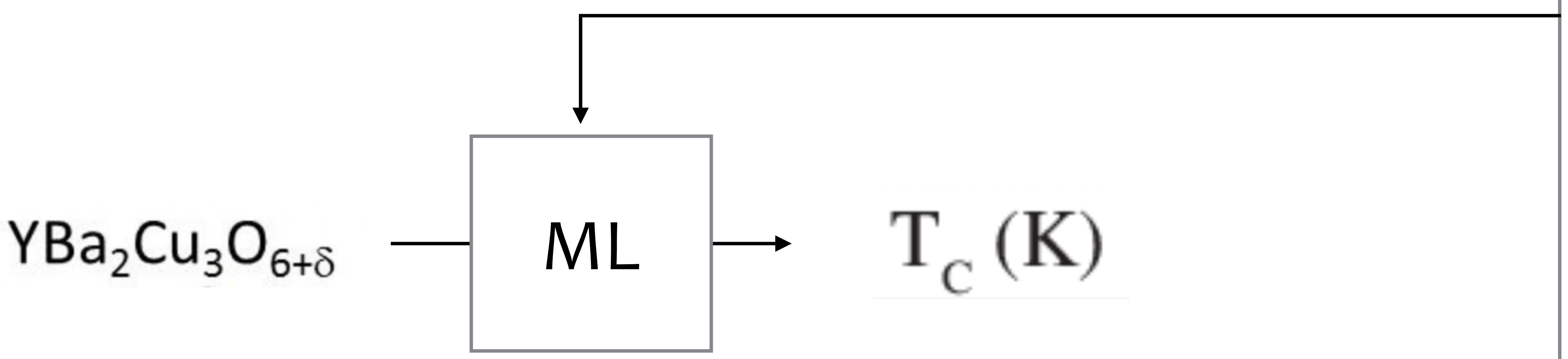
and in many cases we are looking for a **<u>concise</u>** approximation*

*e.g. smaller circuit = smaller time cost

# Proving Learning separations

Is there a learning problem that a QML can learn (efficiently) whereas ML cannot?

Learning problem? E.g., prediction problem:




Database

| Superconductor | $T_C$ (K) |
|---|---|
| $K_yWO_3$ | 6.0 |
| $LiTi_{2+y}O_4$ | 1.2 |
| $BaPb_{1-y}Bi_yO_3$ | 13 |
| $La_{2-y}Ba_yCuO_4$ | 30 |
| $YBa_2Cu_3O_{7-y}$ | 90 |
| $Ba_{1-y}K_yBiO_3$ | 20 |
| $BiSrCaCu_2O_{6+y}$ | 105 |
| $TlBa_2Ca_2Cu_3O_{9+y}$ | 110 |
| $HgBa_2CaCu_2O_{6+y}$ | 120 |
| $GdFeAsO_{1-y}$ | 53.5 |

What does efficiently mean? Why/how could a CC fail?

# A formal framework

## The supervised learning problem - probably approximately correct (PAC) learning — simplified

We learn a "concept" from some concept class $C = \{c_j\}_i, \quad c_j : \vec{x} \mapsto \{0,1\}, \quad \vec{x} \in \{0,1\}^n$

"data"= $\{(\vec{x}_i, c_j(\vec{x}_i))\}_i$ → Machine learning algorithm $A$ → $h := A(data, \cdot\,) : Data \to Labels$

Data-points ($\vec{x}$'s) come from a *fixed distribution* $\mathscr{D}$

⟨aQa^L⟩

# The supervised learning problem - probably approximately correct (PAC) learning — simplified

function function $\mathbb{R}$ or $\{0,1\}^m$ $S \subseteq \mathbb{R}^n$

We learn a "~~concept~~" from some ~~concept~~ class $C = \{c_j\}_i$, $c_j : \vec{x} \mapsto$ ~~$\{0,1\}$~~, $\vec{x} \in$ ~~$\{0,1\}^n$~~

"data"= $\{(\vec{x}_i, c_j(\vec{x}_i))\}_i$ → Machine learning algorithm $A$ → $h := A(data, \cdot) : Data \rightarrow Labels$

Data-points ($\vec{x}$'s) come from a *fixed distribution* $\mathscr{D}$

# The supervised learning problem - probably approximately correct (PAC) learning — simplified

We learn a "concept" from some concept class $C = \{c_j\}_i$, $c_j : \vec{x} \mapsto \{0,1\}$, $\vec{x} \in \{0,1\}^n$

"data"= $\{(\vec{x}_i, c_j(\vec{x}_i))\}_i$ → Machine learning algorithm A → $h := A(data, \cdot) : Data \rightarrow Labels$

Data-points ($\vec{x}$'s) come from a *fixed distribution* $\mathscr{D}$

Learner A learns C **efficiently** if ∀ concepts $c_j$, given data labeled by $c_j$
with probability $\geq 1 - \delta$ it outputs $h$, s.t.
$$P_{x\in\mathscr{D}}(c_j(x) \neq h(x)) \leq \epsilon,$$
with polynomial resources (time, data) in $n, \epsilon^{-1}, \delta^{-1}$

## The supervised learning problem - probably approximately correct (**PAC**) learning — simplified

We learn a "concept" from some concept class $C = \{c_j\}_i$, $\quad c_j : \vec{x} \mapsto \{0,1\}$, $\quad \vec{x} \in \{0,1\}^n$

"data"= $\{(\vec{x}_i, c_j(\vec{x}_i))\}_i$ → **Machine learning algorithm $A$** → $h := A(data, \cdot) : Data \rightarrow Labels$

Data-points ($\vec{x}$'s) come from a *fixed distribution* $\mathscr{D}$

> Learner A learns C **efficiently** if $\forall$ concepts $c_j$, given data labeled by $c_j$
>
> with probability $\geq 1 - \delta$ it outputs $h$, s.t.
>
> $P_{x \in \mathscr{D}}(c_j(x) \neq h(x)) \leq \epsilon$,
>
> with polynomial resources (time, data) in $n$, $\epsilon$ ... $\delta$

"$c_j$ and $h$ are equal up to PAC"
or "*Heur- equal*"

“Learning separation/advantages” $\approx$ $\exists$concept class which a QC can learn efficiently, and a classical computer cannot

Two remarks

“Learning separation/advantages” $\approx$ $\exists$concept class which a QC can learn efficiently, and a classical computer cannot

Note whatever happens, it will *have to be contingent on assumptions in complexity theory*
If (F)BPP = (F)BQP, there *cannot* be a learning separation at all

So the fundamental question is about the relationship between learning and computing

“Learning separation/advantages” ≈ ∃concept class which a QC can learn efficiently, and a classical computer cannot (contingent on assumptions in complexity theory)

**Please note: the definition above inherently carries a notion of “scaling” of the problem**

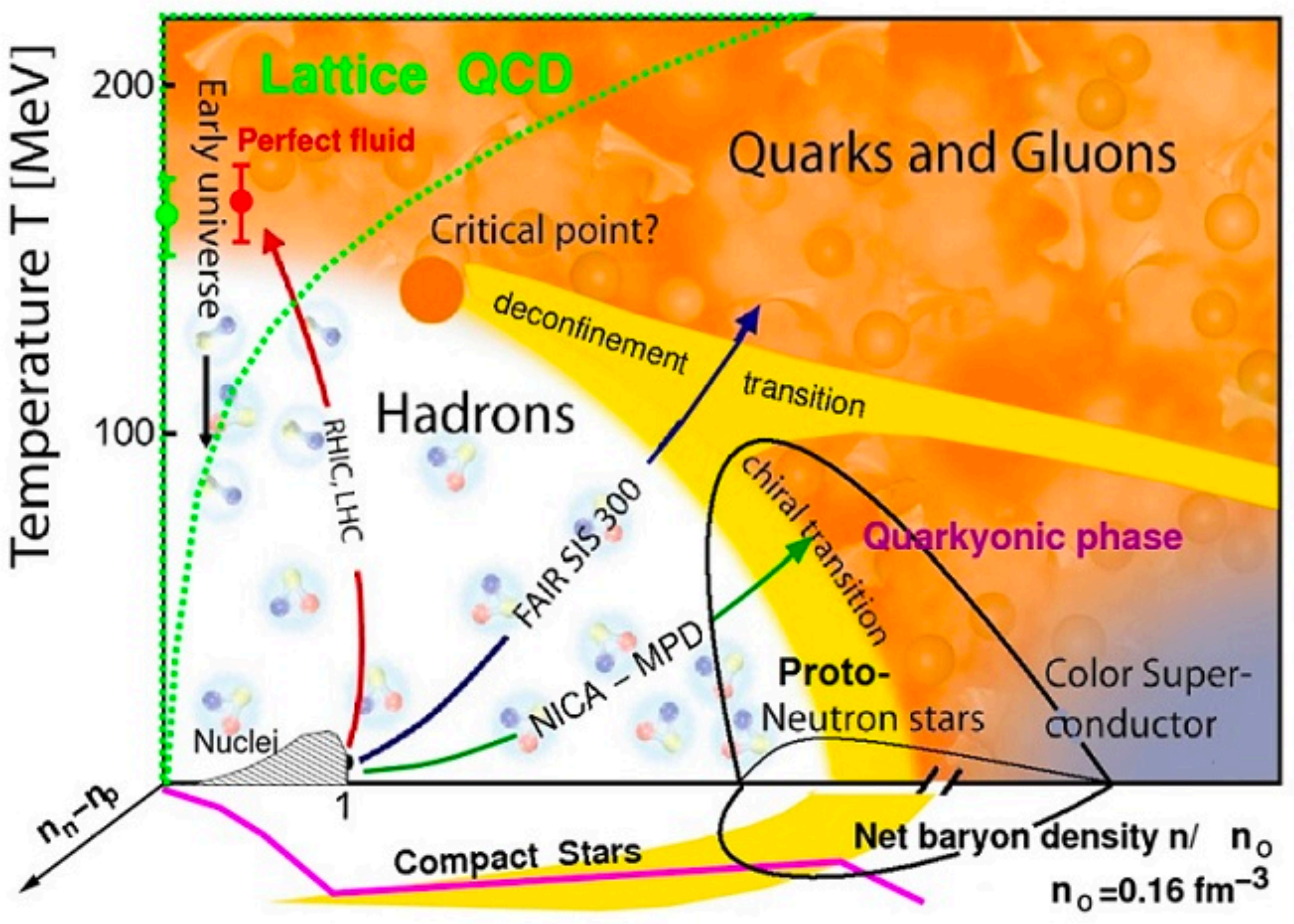
**For a fixed size task… this does not make sense.**

**Nonetheless, we can (I believe) use such arguments as evidence of suitabilty of QML for problems**

# Complexity theoretic and scaling arguments for questions about nature

e.g. computing some property of
quark-gluon plasma may require
a lattice of some huge but fixed, and
(perhaps) knowable size .

Solving a constant-sized problem takes O(1)




*source: slides of Simone*

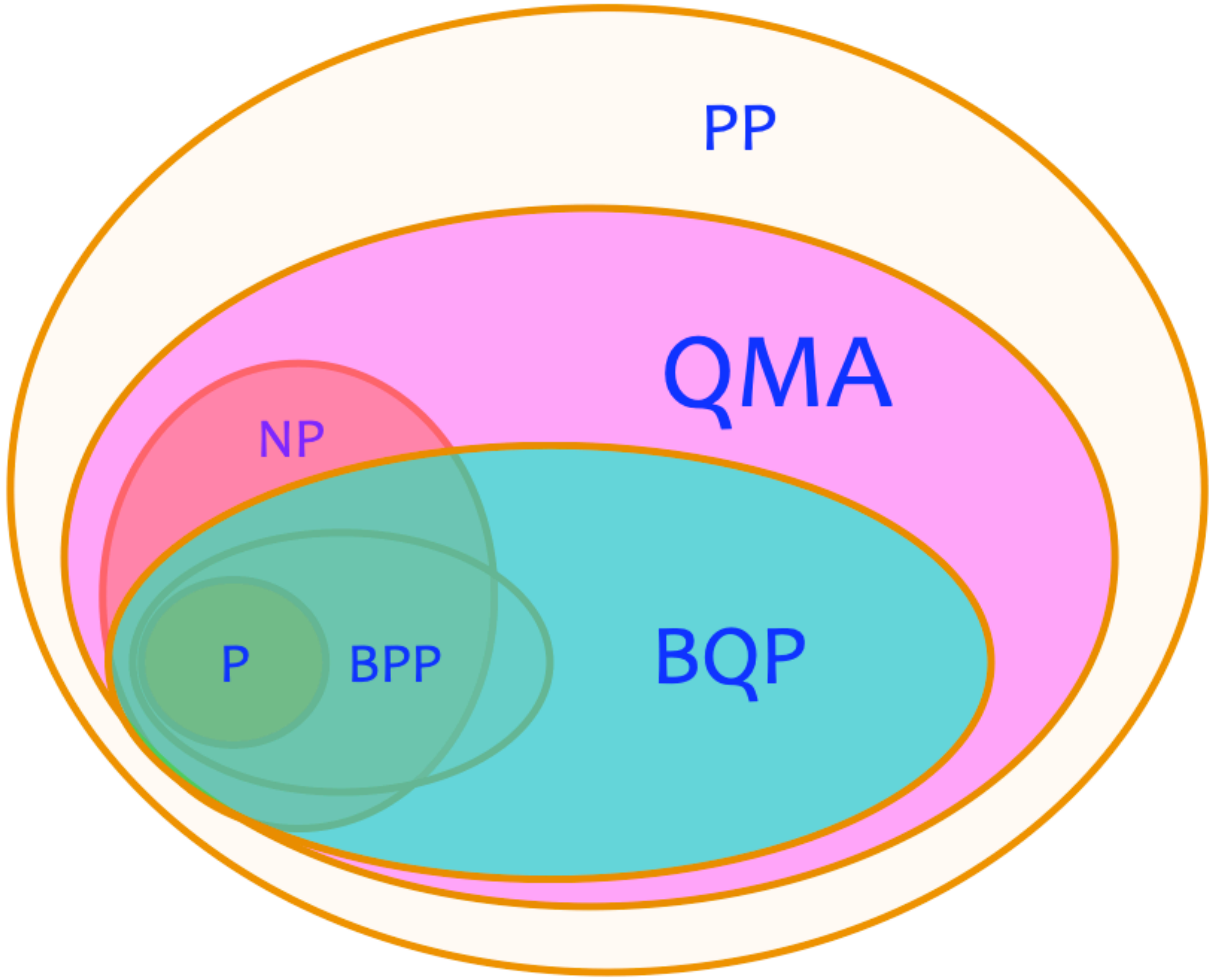
(Provable) exponential scaling differences strong **evidence**
that at relevant size, quantum solution would take
infinitesimal time of (astronomic) time needed for classical solution

## Learning v.s. computing

# Learning versus algorithmic complexity

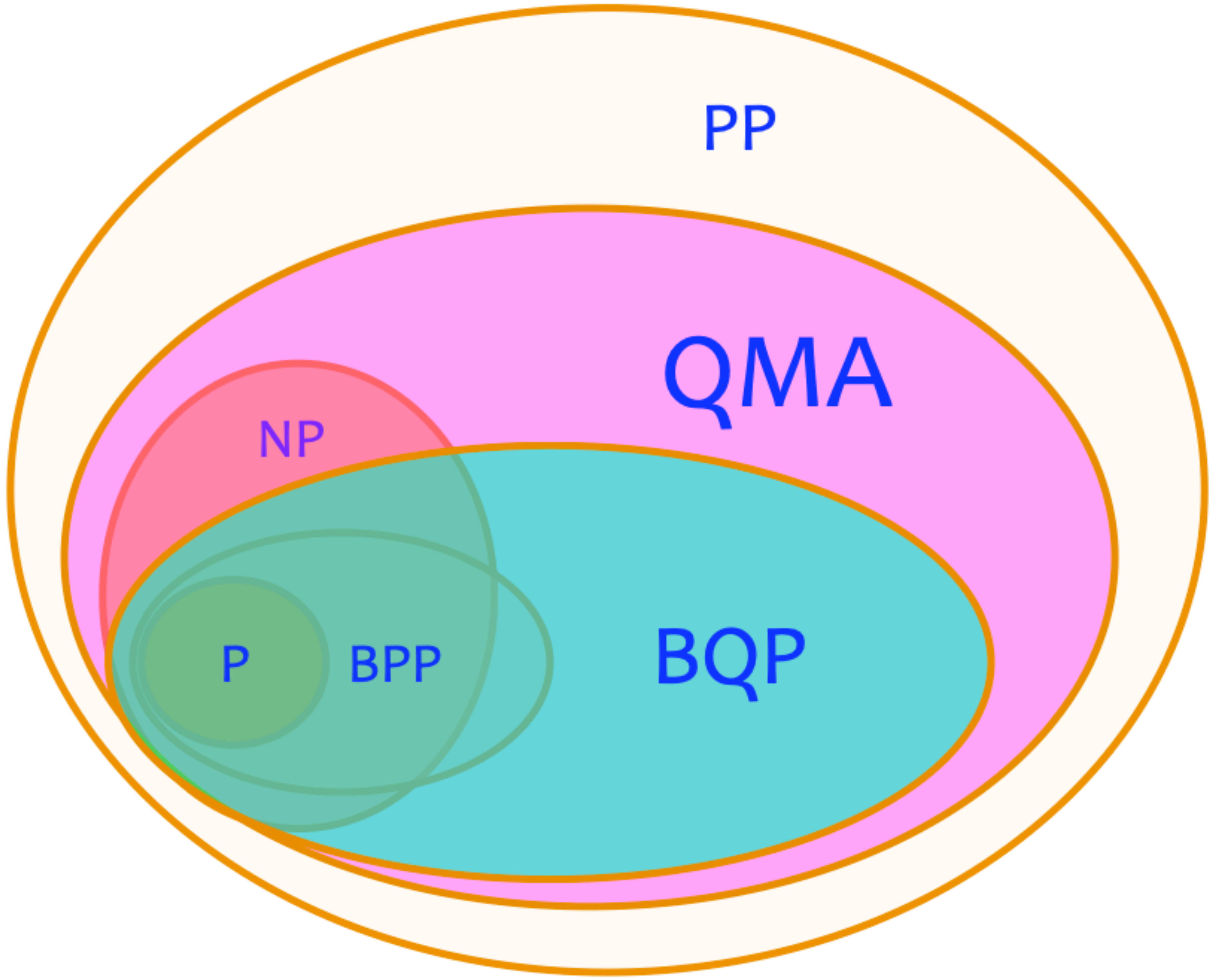
Can we have *learning* separations… if we <u>assume computational separations</u>

And why is this not trivial?



## Learning versus algorithmic complexity

Can we have *learning* separations… if we assume computational separations




And why is this not trivial?

Consider **concept:** $c^{\theta}(x) = Tr[U(\theta)\rho(x)U^{\dagger}(\theta)O]$

*Assume I believe a classical computer cannot compute this function (concept).*

And if it is… is this what we want?

**Why does classical computational hardness of concepts not (immediately) imply non-learnability**

1. **Data gap:** Machine learning comes with **data…** we are given **evaluations of c….**

2. **Quantum learnability:** Must ensure the quantum learner **can** learn it, and already shallow classical classical circuits are *not learnable*

3. **Worst case v.s. heuristics:** what does "cannot compute" mean, exactly?

4. What do we **actually mean by learning:** *evaluation or identification*

⟨aQa^L⟩

## Why does classical computational hardness of concepts not (immediately) imply non-learnability

1. **Data gap:** Machine learning comes with **data…** we are given **evaluations of c.…**

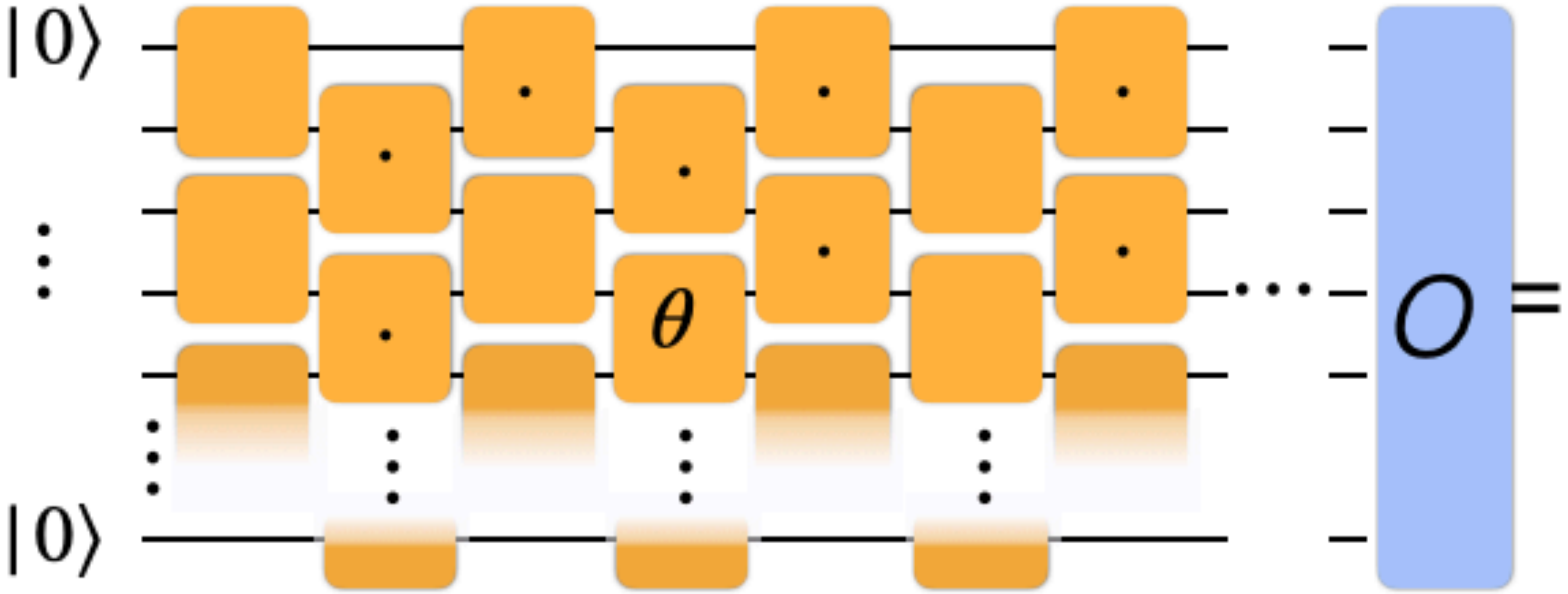
2. **Quantum learnability:** Must ensure the quantum learner **can** learn it, and already shallow classical classical circuits are *not learnable*

3. **Worst case v.s. heuristics:** what does "cannot compute" mean, exactly?

4. What do we **actually mean by learning:** *evaluation or identification*

**Classical computers with data can be *more powerful.***

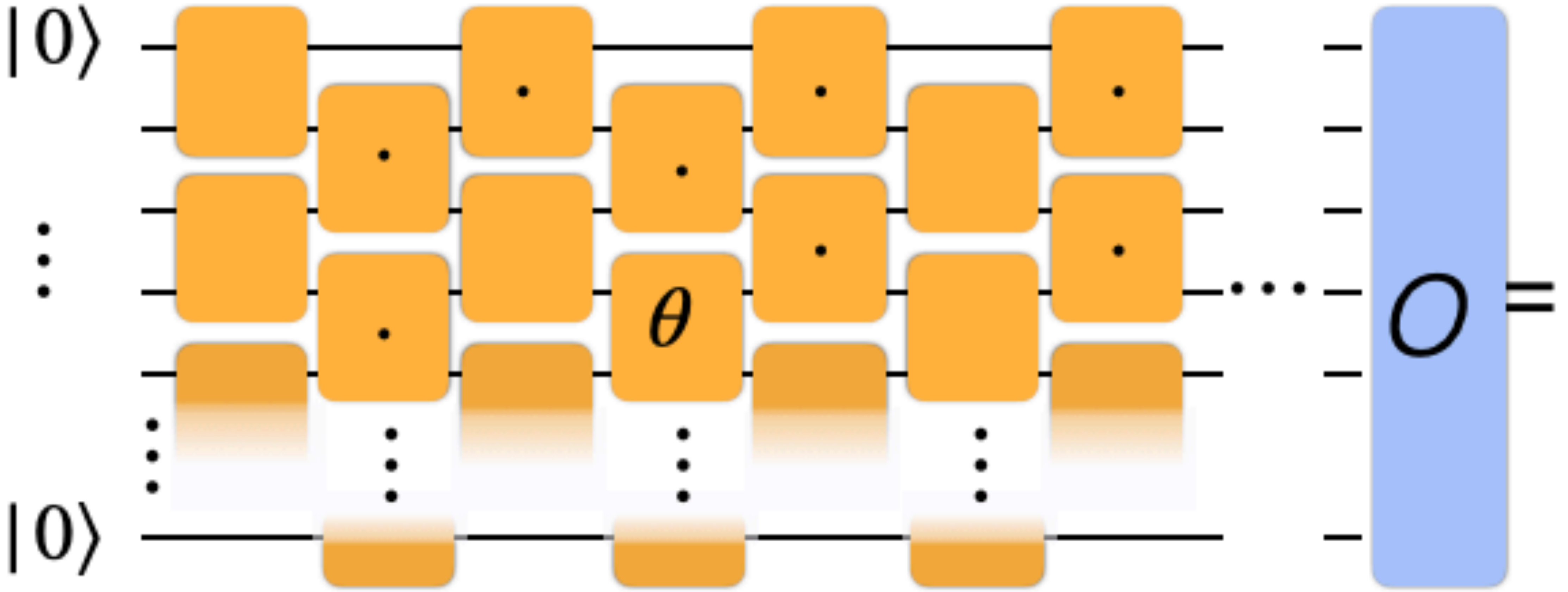
## Data gap:




$$\Rightarrow \quad f(\theta) = \mathrm{Tr}[\rho(\theta) O(\,\cdot\,)]$$


Power of data in quantum machine learning, Hsin-Yuan Huang… Jarrod R. McClean Nat. Com., Vol.12, No. 2631 (2021)
Provably efficient machine learning for quantum many-body problems Hsin-Yuan Huang,…John Preskill , Science Vol 377, Issue 6613 (2022)

## Data gap:



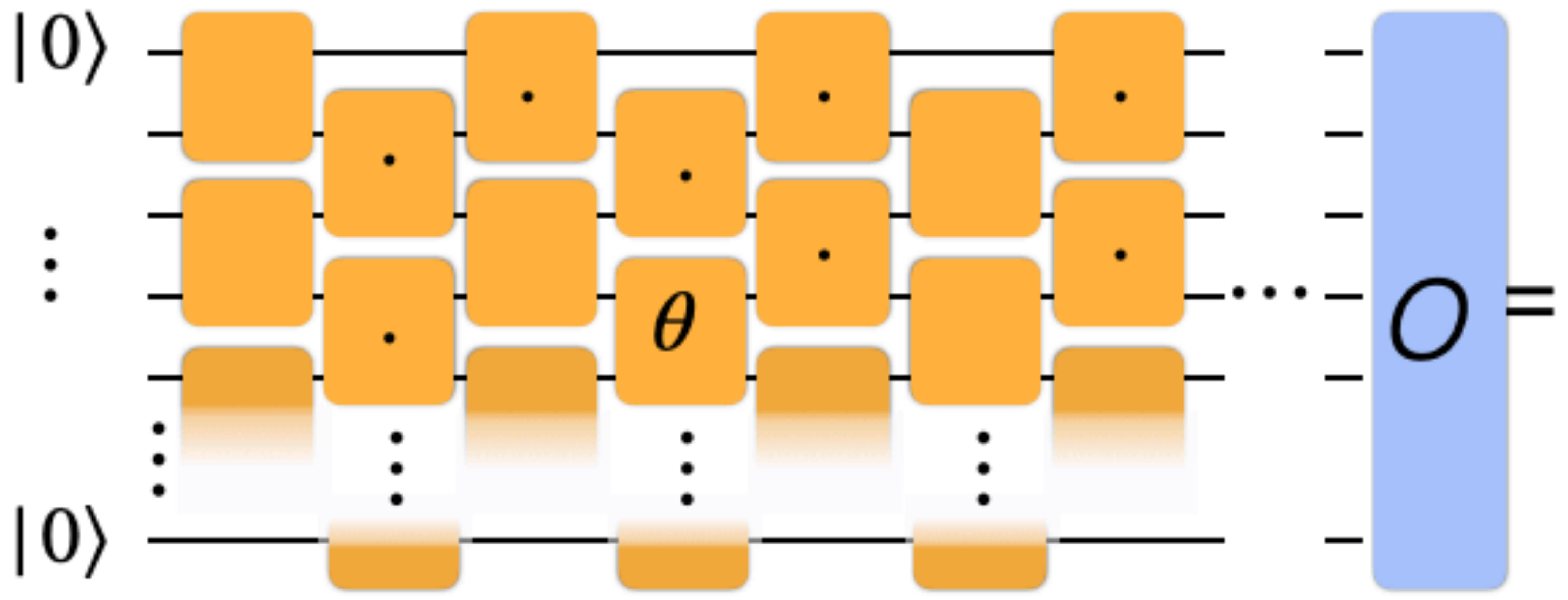

$$\Rightarrow \quad f(\theta) = \mathrm{Tr}[\rho(\theta)O(\,\cdot\,)]$$

However can be shown:

$$f(\theta) = \alpha \sin(\theta + \beta) + \gamma, \quad \alpha, \beta, \gamma \in \mathbb{R}$$


Power of data in quantum machine learning, Hsin-Yuan Huang… Jarrod R. McClean Nat. Com., Vol.12, No. 2631 (2021)
Provably efficient machine learning for quantum many-body problems Hsin-Yuan Huang,…John Preskill , Science Vol 377, Issue 6613 (2022)

## Data gap:




$$\Rightarrow \quad f(\theta) = \mathrm{Tr}[\rho(\theta) O(\,\cdot\,)]$$

However can be shown:

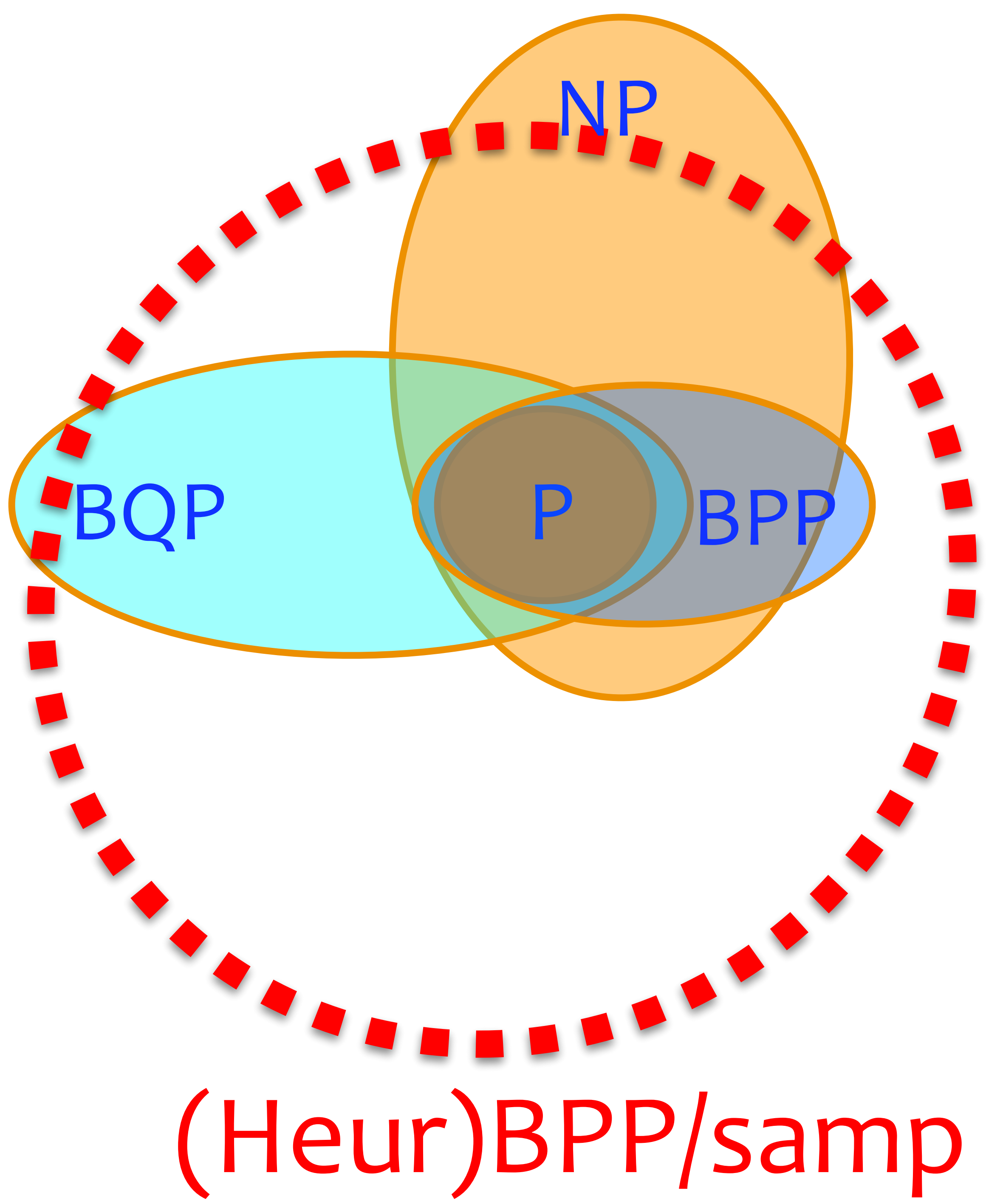
$$f(\theta) = \alpha \sin(\theta + \beta) + \gamma, \quad \alpha, \beta, \gamma \in \mathbb{R}$$

"**obfuscated** function"
related to "trapdoor"

datapoints + fit reveals $\alpha, \beta, \gamma$


Power of data in quantum machine learning, Hsin-Yuan Huang… Jarrod R. McClean Nat. Com., Vol.12, No. 2631 (2021)
Provably efficient machine learning for quantum many-body problems Hsin-Yuan Huang,…John Preskill , Science Vol 377, Issue 6613 (2022)

# Data changes the (computational) game




“Data changes the game”
in complexity lingo

*standard* assumptions
(classical computers cannot simulate
quantum computers)

$\not\Longrightarrow$

classical computers cannot
*learn to solve the same problems that QCs can solve*

*(it’s not even the “in practice” vs. “in theory”dichotomy)*

# … even in *seemingly quinessentially quantum* tasks

Science — Current Issue · First release papers · Archive · About · Submit manuscript

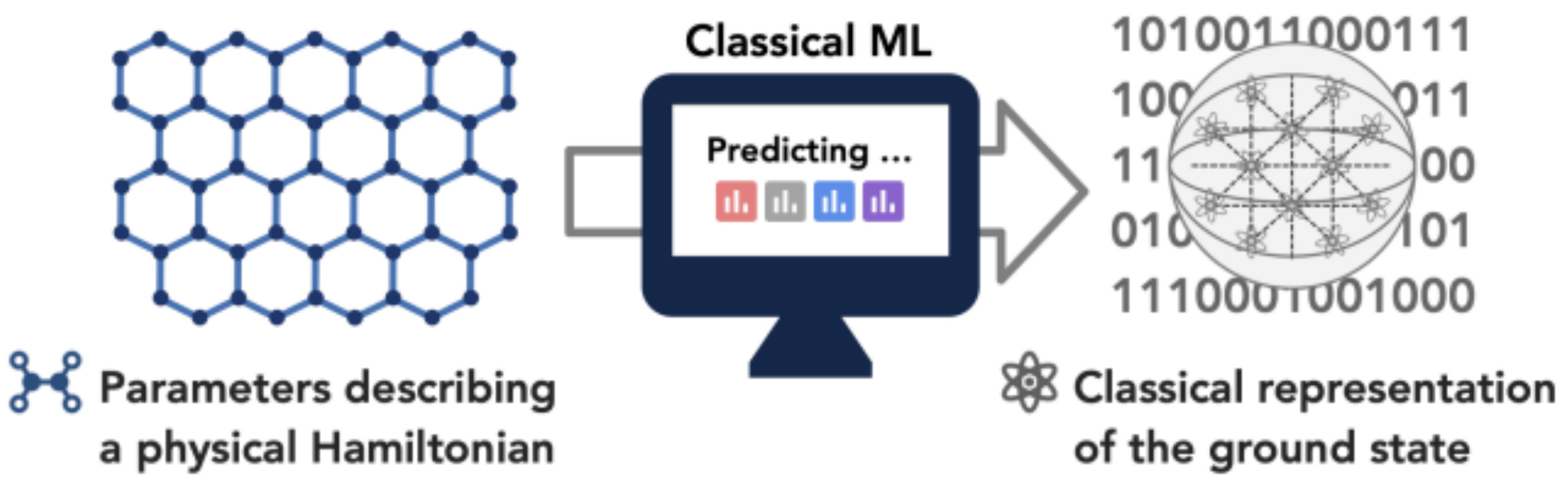
RESEARCH ARTICLE | QUANTUM PHYSICS

**Provably efficient machine learning for quantum many-body problems**

HSIN-YUAN HUANG, RICHARD KUENG, GIACOMO TORLAI, VICTOR V. ALBERT, AND JOHN PRESKILL · Authors Info & Affiliations

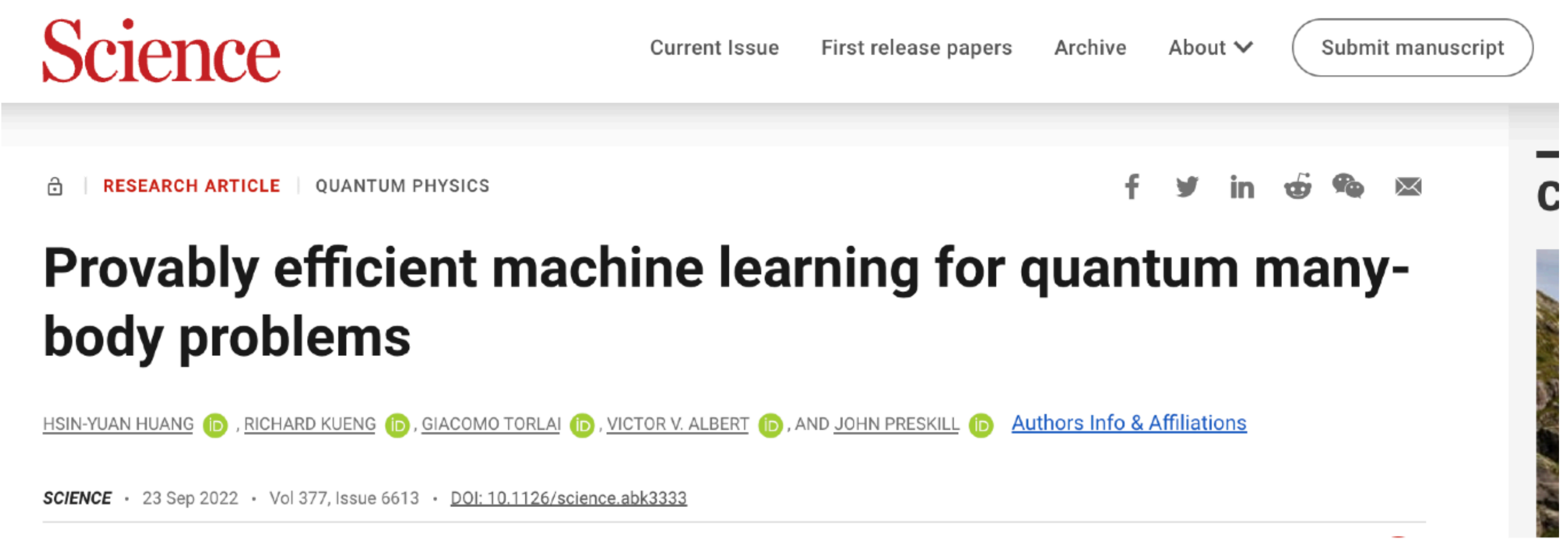
SCIENCE · 23 Sep 2022 · Vol 377, Issue 6613 · DOI: 10.1126/science.abk3333

"Ground state estimation"




under (seemingly) very mild conditions

# … even in *seemingly quinessentially quantum* tasks


Science | Current Issue | First release papers | Archive | About | Submit manuscript

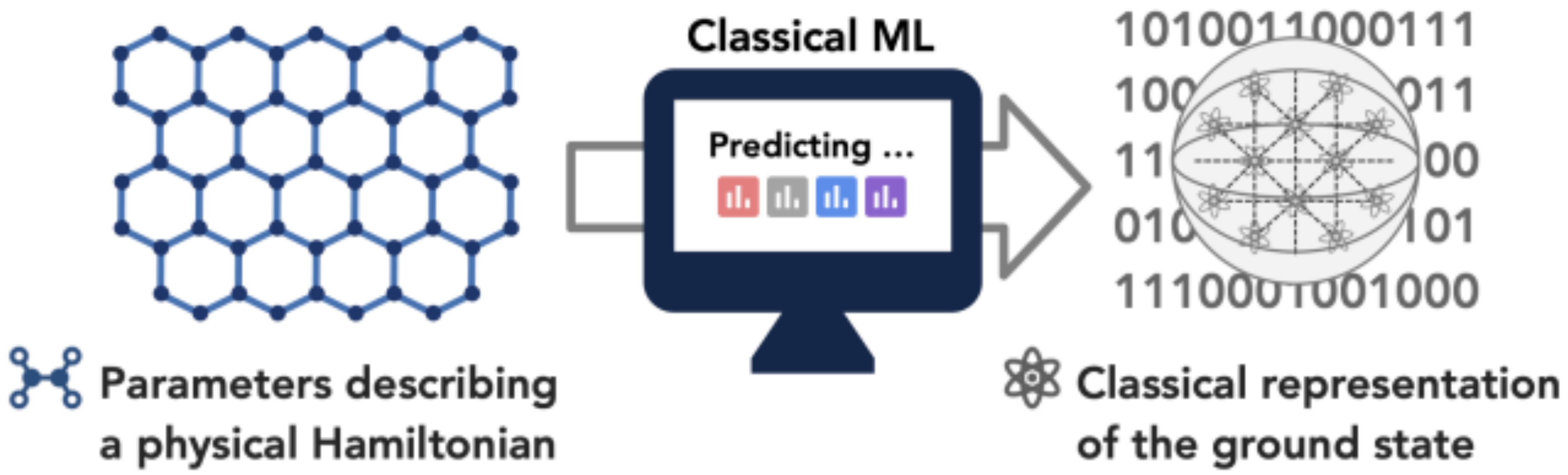
RESEARCH ARTICLE | QUANTUM PHYSICS

**Provably efficient machine learning for quantum many-body problems**

HSIN-YUAN HUANG, RICHARD KUENG, GIACOMO TORLAI, VICTOR V. ALBERT, AND JOHN PRESKILL | Authors Info & Affiliations

SCIENCE • 23 Sep 2022 • Vol 377, Issue 6613 • DOI: 10.1126/science.abk3333

"Ground state estimation"




under (seemingly) very mild conditions

Seems like:

Estimating ground state properties: **hard even on a QC!***

Estimating ground state properties <u>*given a database*</u> **easy on a CC!***

Do we even need QCs??

questions?

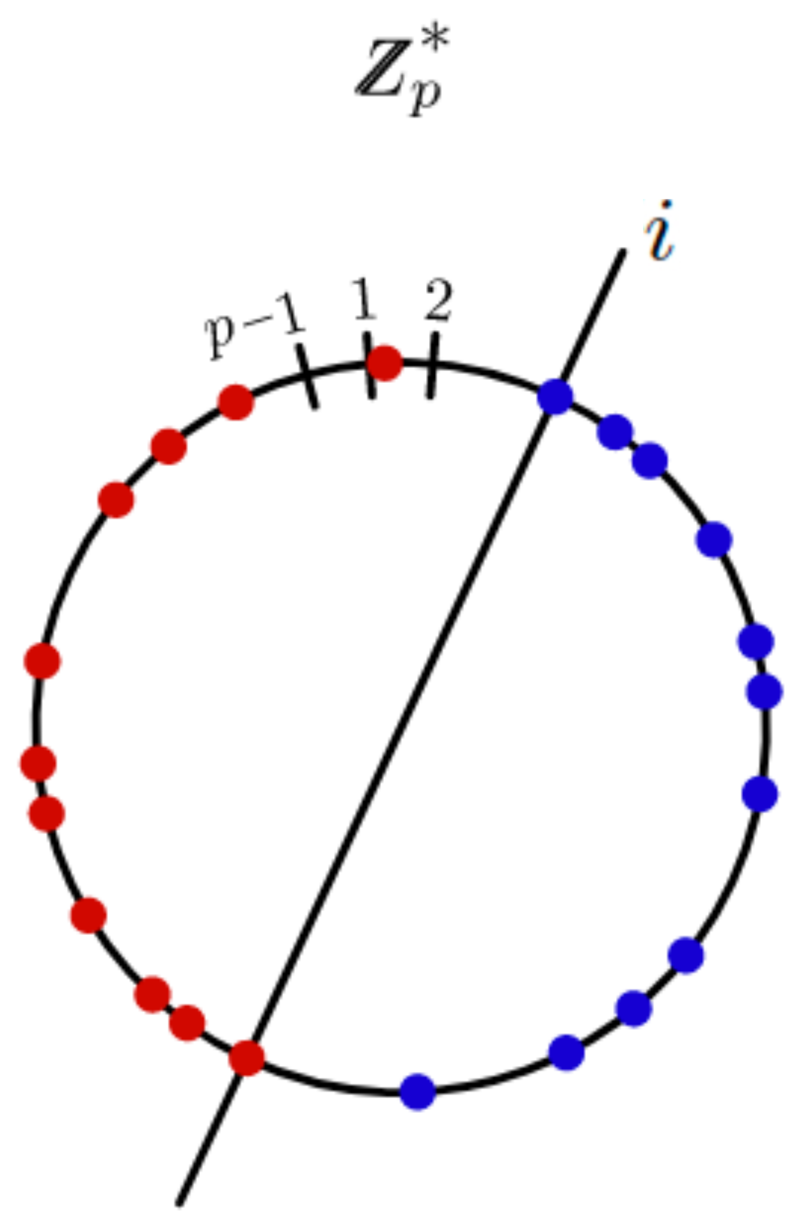
## But we do have a few heroic examples

**Q: How to prove data does not add power to a classical computer?**

**A: Show classical computer could have generated it by itself!**

*(later we will show this is not good enough for us, and we can do better)*

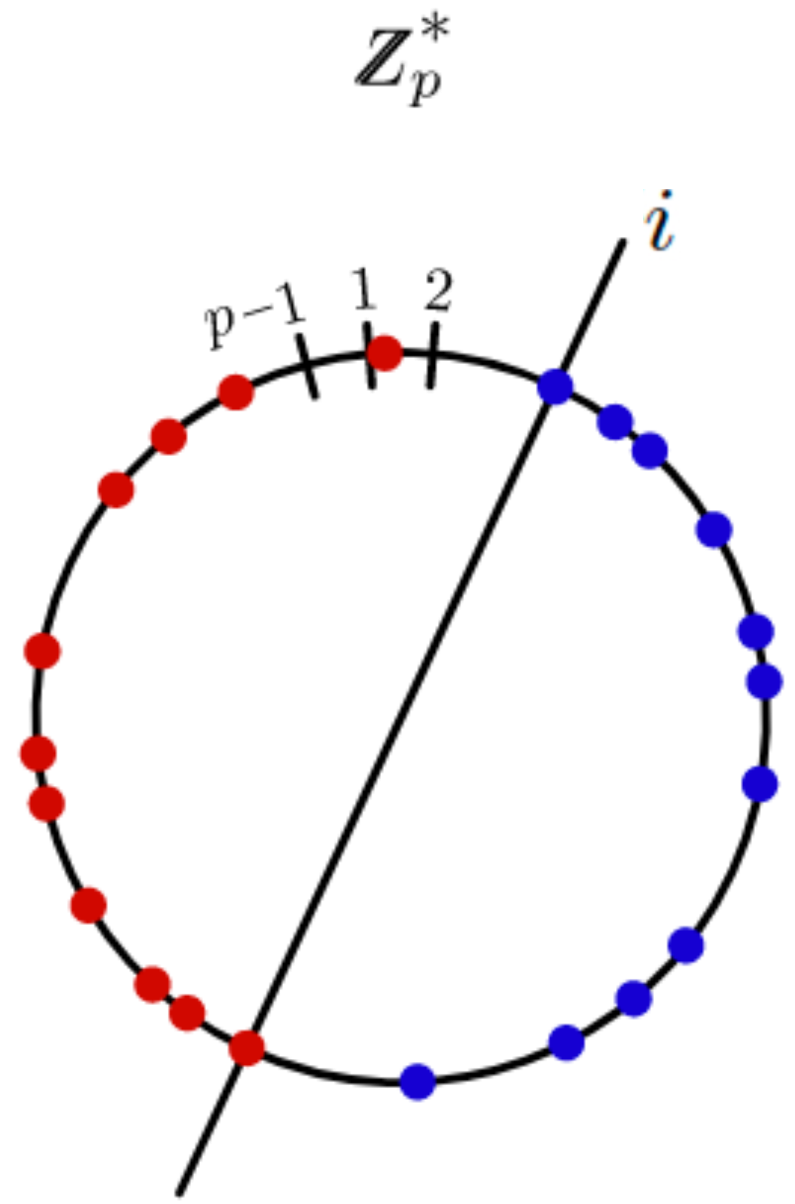
# How does it work?



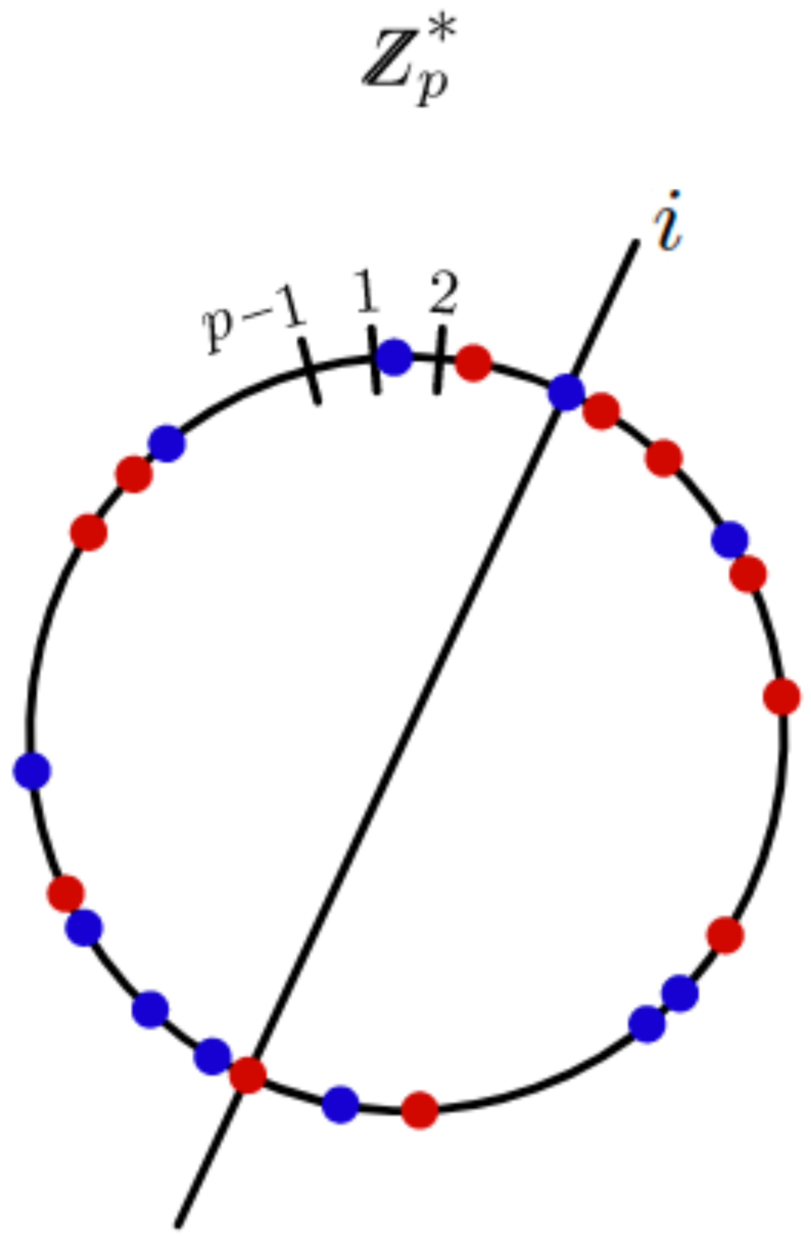

$$\tilde{c}_i(x) = \begin{cases} +1, & x \in [i, i + \frac{p-3}{2}] \\ -1, & \text{else.} \end{cases}$$

**Easy to learn**

# How does it work?



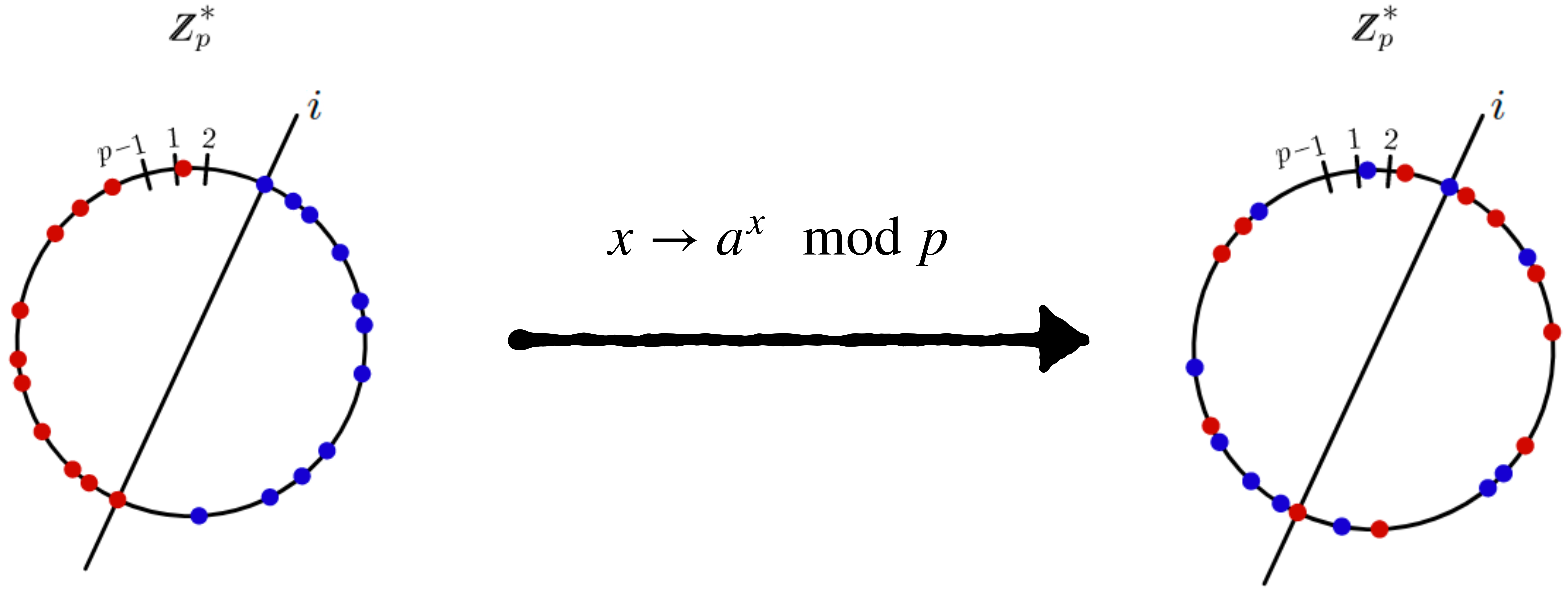

$$x \to a^x \mod p$$




$$\tilde{c}_i(x) = \begin{cases} +1, & x \in [i, i + \frac{p-3}{2}] \\ -1, & \text{else.} \end{cases}$$

$$c_i(x) = \begin{cases} +1, & \text{if } \log_a x \in [i, i + \frac{p-3}{2}] \\ -1, & \text{else.} \end{cases}$$

## How does it work?

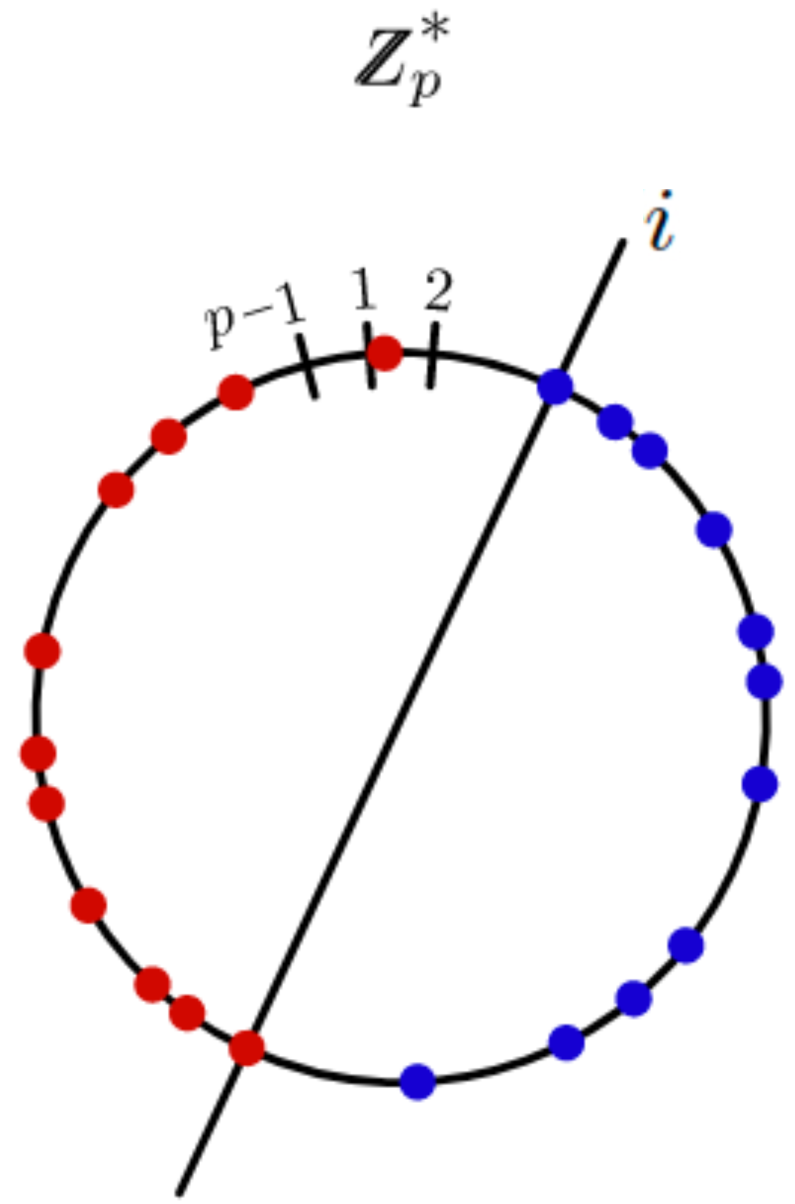



**Easy to learn**

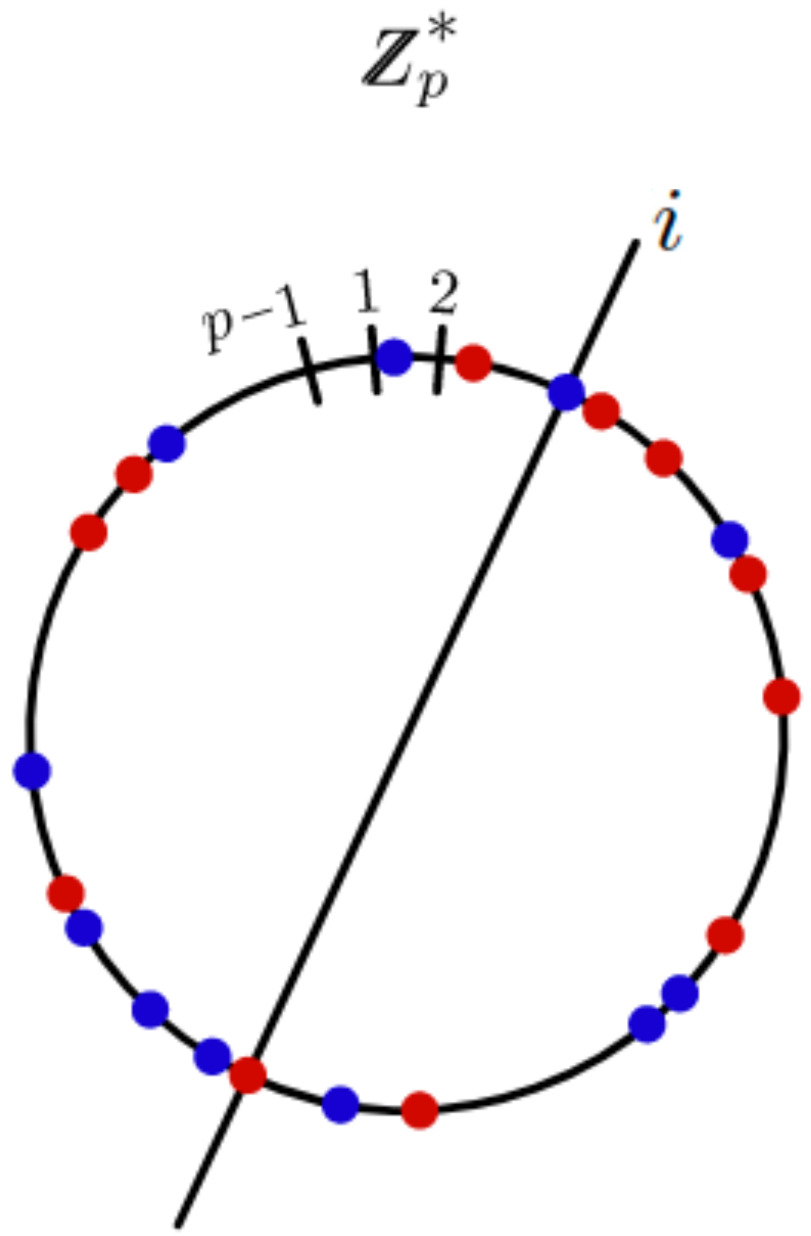
**Could be easy to learn, e.g.**
**if** can compute $x \to \log_a(x) \pmod p$

Shor's algorithm ! (quantum learnable)

## How does it work?




$x \to a^x \mod p$



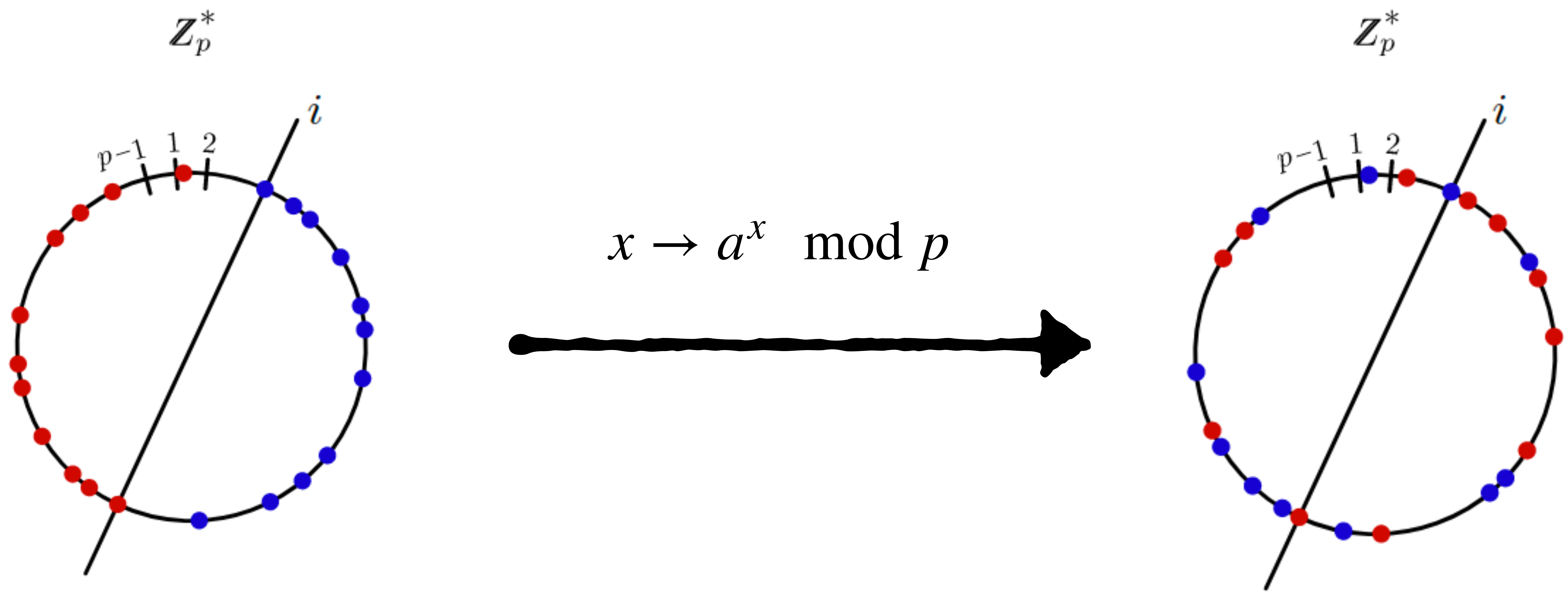

**Easy to learn**

**Could be easy to learn, e.g.**
**if can compute** $x \to \log_a(x) \pmod p$

Shor's algorithm ! (quantum learnable)

But is it **necessary** to apply discrete-log?

## How does it work?




$$c_i(x) = \begin{cases} +1, & \text{if } \log_a x \in [i, i + \frac{p-3}{2}] \\ -1, & \text{else.} \end{cases}$$

**Theorem:**
if A can learn all above efficiently given examples
then A (+classical processing) can solve discrete log.

**So yes, in a way it is necessary! Not classically learnable.**

⟨aQa⟩

## Simplified version of heroic example

Q: How to prove data does not add power to a classical computer?

A: Show classical computer could have generated it by itself!

**Example:** $f_a(k) = a^k \mod N$

$$DLP_a^N(k) = f_a^{-1}(k)$$

$$a^{DLP_a^N(k)} \mod N = k$$

## Simplified version of heroic example

Q: How to prove data does not add power to a classical computer?

A: Show classical computer could have generated it by itself!

easy to compute

**Example:** $f_a(k) = a^k \mod N$

hard to compute

$DLP_a^N(k) = f_a^{-1}(k)$

$a^{DLP_a^N(k)} \mod N = k$

## Simplified version of heroic example

Q: How to prove data does not add power to a classical computer?

A: Show classical computer could have generated it by itself!

easy to compute

**Example:** $f_a(k) = a^k \mod N$

hard to compute

$$DLP_a^N(k) = f_a^{-1}(k)$$

$$a^{DLP_a^N(k)} \mod N = k$$

Given x cannot compute y = DLP(x)

Data is $\{(x, y)\}_{x \sim Unif}$

⟨aQa^L⟩

## Simplified version of heroic example

Q: How to prove data does not add power to a classical computer?

A: Show classical computer could have generated it by itself!

easy to compute

**Example:** $f_a(k) = a^k \mod N$

Do: choose y random;
compute $(a^y \mod N, y)$
re-label: $((x, y = DLP(x)))$
x is also uniform at random!

hard to compute

$$DLP_a^N(k) = f_a^{-1}(k)$$

$$a^{DLP_a^N(k)} \mod N = k$$

Given x cannot compute y = DLP(x)
Data is $\{(x, y)\}_{x \sim Unif}$

## Simplified version of heroic example

**Q: How to prove data does not add power to a classical computer?**

**A: Show classical computer could have generated it by itself!**

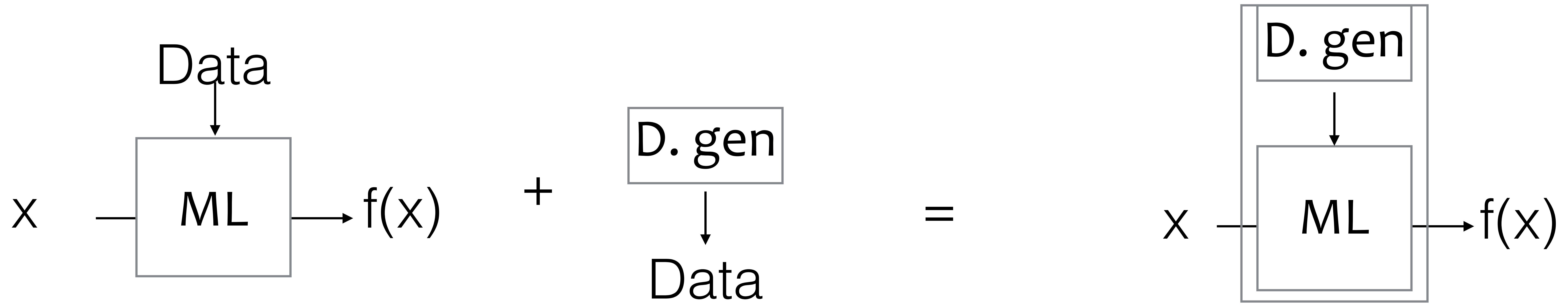
**Learning task:**

given x "predict" DLP(x), having access to valid data: **Data=**$\{(x, DLP_a(x))\}_{x\sim Unif}$

## Simplified version of heroic example

**Q: How to prove data does not add power to a classical computer?**

**Non-learnability by contradiction:**
Let A’ be a learning algorithm that learns DLP.
Then there exists a classical poly time NON-learning algorithm for DLP:




eff. learning algo — eff. data gen. — eff. “non-learning” algo!

# Putting it together

*Differences between computing and learning*

1.**Data gap:** Machine learning comes with **data…** we are given **evaluations of c….**

2.**Quantum learnability:** Must ensure the quantum learner **can** learn it, and already shallow classical classical circuits are *not learnable*

3. **Worst case v.s. heuristics:** what does "cannot compute" mean, exactly?

4.What do we **actually mean by learning:** *evaluation or identification*

1) Can generate data for random!

2) Shor's algo cracks DLP

3) Magic of DLP:* random self-reducibility

4) We need to *evaluate*

→**Learning separation, assuming** $DLP$ **is not in** $P$

# Putting it together

*Differences between computing and learning*

1. **Data gap:** Machine learning comes with **data…** we are given **evaluations of c….** ⟷ 1) Can generate data for random!

2. **Quantum learnability:** Must ensure the quantum learner **can** learn it, and already shallow classical classical circuits are *not learnable* ⟷ **2) Shor's algo cracks DLP**

3. **Worst case v.s. heuristics:** what does "cannot compute" mean, exactly? ⟷ **3) Magic of DLP:* random self-reducibility**

4. What do we **actually mean by learning:** *evaluation or identification* ⟷ **4) We need to *evaluate***

*We also need a bunch of other useful properties of DLP, such as
-average case hardness (if you solve on 1/2, you solve always)
-hard-core bit (computing even a single digit is as hard as everything)
All of these are **key cryptographic properties of DLP**

⟨aQa⟩

# Most known learning separations…

On the Quantum versus Classical Learnability of Discrete Distributions

Ryan Sweke[1], Jean-Pierre Seifert[2,3], Dominik Hangleiter[1], and Jens Eisert[1,4,5]

**A rigorous and robust quantum speed-up in supervised machine learning**

Yunchao Liu, Srinivasan Arunachalam & Kristan Temme

*Nature Physics* **17**, 1013–1017 (2021) | Cite this article

EQUIVALENCES AND SEPARATIONS BETWEEN QUANTUM AND CLASSICAL LEARNABILITY*

ROCCO A. SERVEDIO† AND STEVEN J. GORTLER‡

*v1 in 2000*

**Parametrized Quantum Policies for Reinforcement Learning**

Sofiene Jerbi, Institute for Theoretical Physics, University of Innsbruck
Casper Gyurik, LIACS, Leiden University
Simon C. Marshall, LIACS, Leiden University
Hans J. Briegel, Institute for Theoretical Physics, University of Innsbruck
Vedran Dunjko, LIACS, Leiden University

use this technique to get rid of the data gap (and other).
Data does not help as we can generate it. **Yay**!

**I am not yay, for a major and less major reason.**

What about the case of "quantum functions"?
Must be better since they are harder?

$i$

⟨aQa^L⟩

What about the case of “quantum functions”?
Must be better since they are harder?

$i$

No. Quantum-generated data **does help the classical learner.**

We provably cannot generate the data:

| |
|---|
| Theorem 1 (paper in prep)<br><br>if ∃ randomized poly-time algorithm for random generation of data - even approximately, even with errors - for any $BQP$-hard function then $BQP$ is in the second level of the polynomial hierarchy |

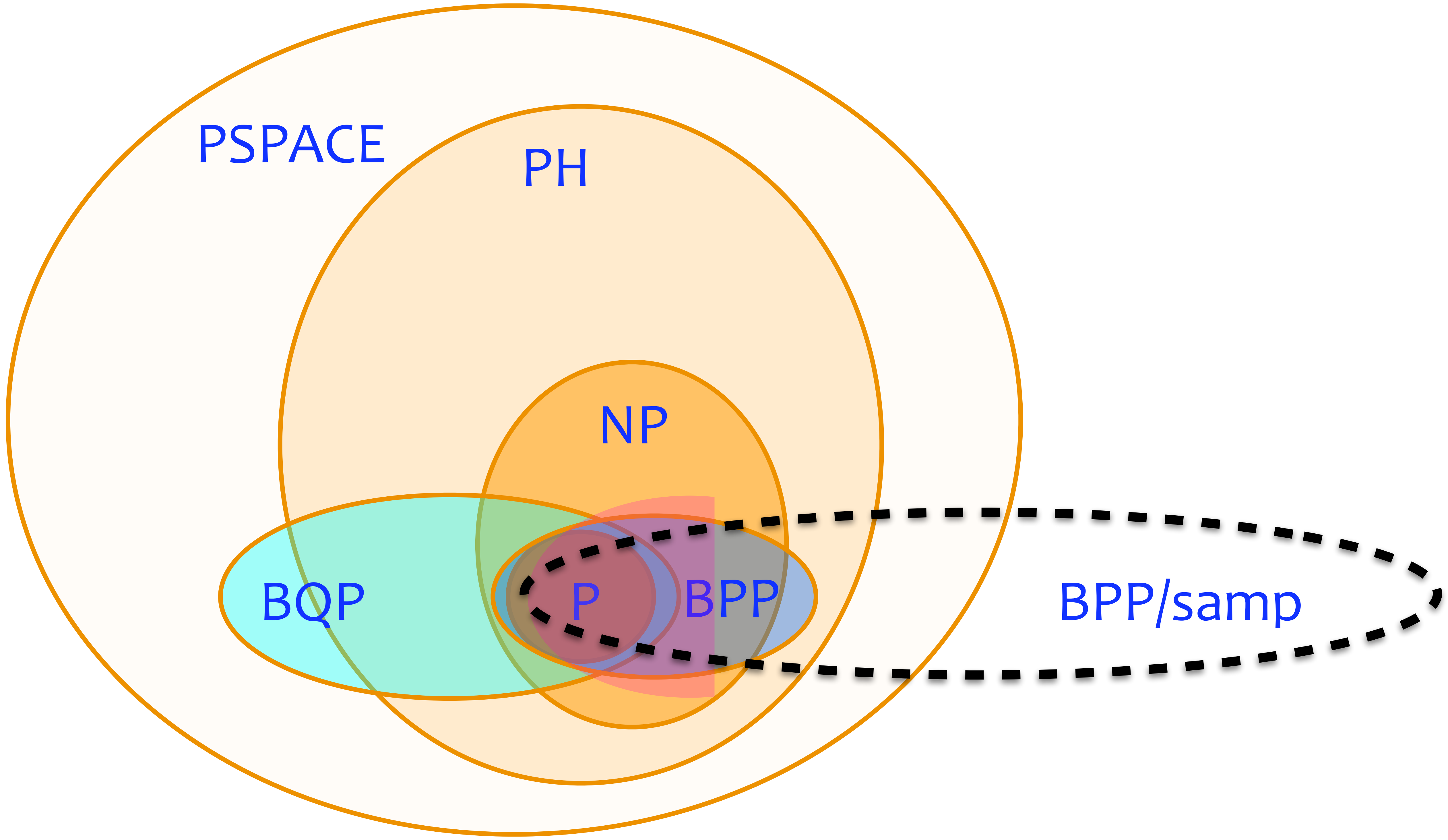
Next: solving the “major issue”:

-how to prove classical impossibility of learning when data *does* help?

*(or relate it to complexity theoretic assumptions)*

-how to prove quantum learnability, for some, and then interesting cases

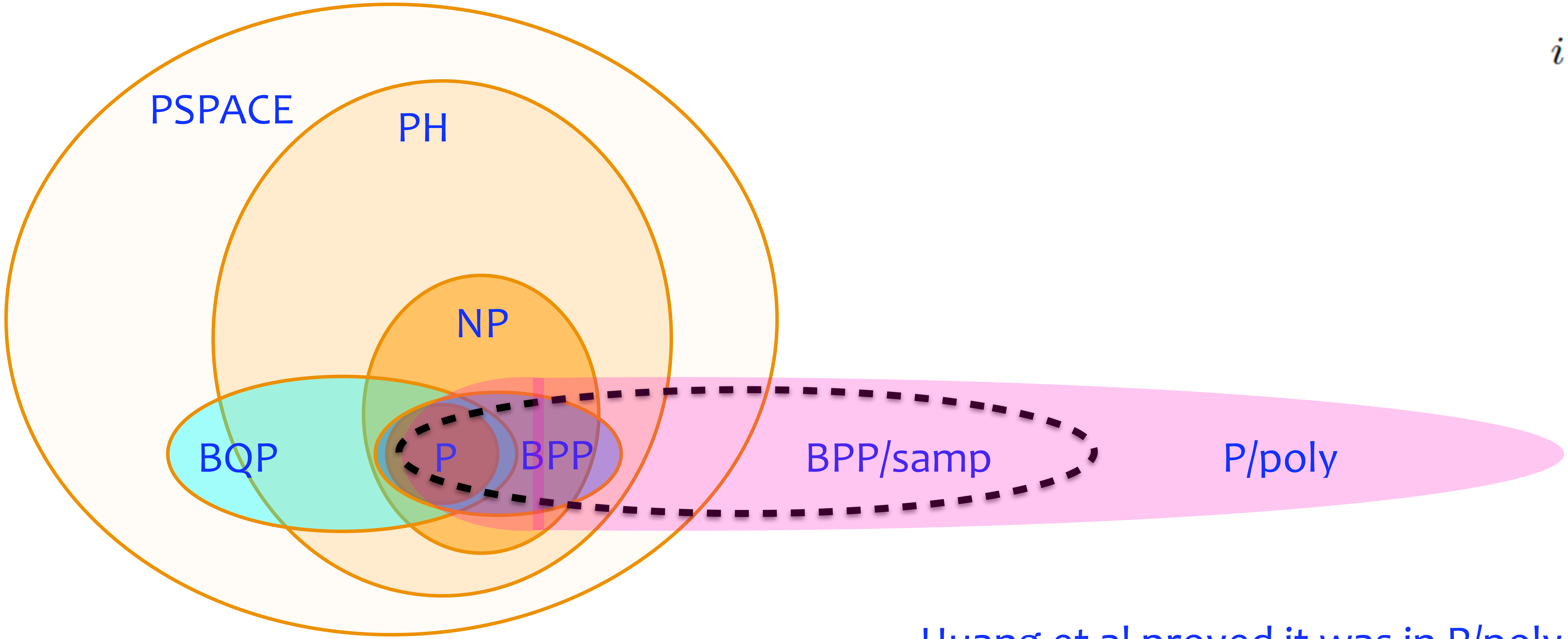
## A simple way out: stronger complexity theoretic assumptions




$i$

This is the class of problems classical computers can solve in the worst case with access to data

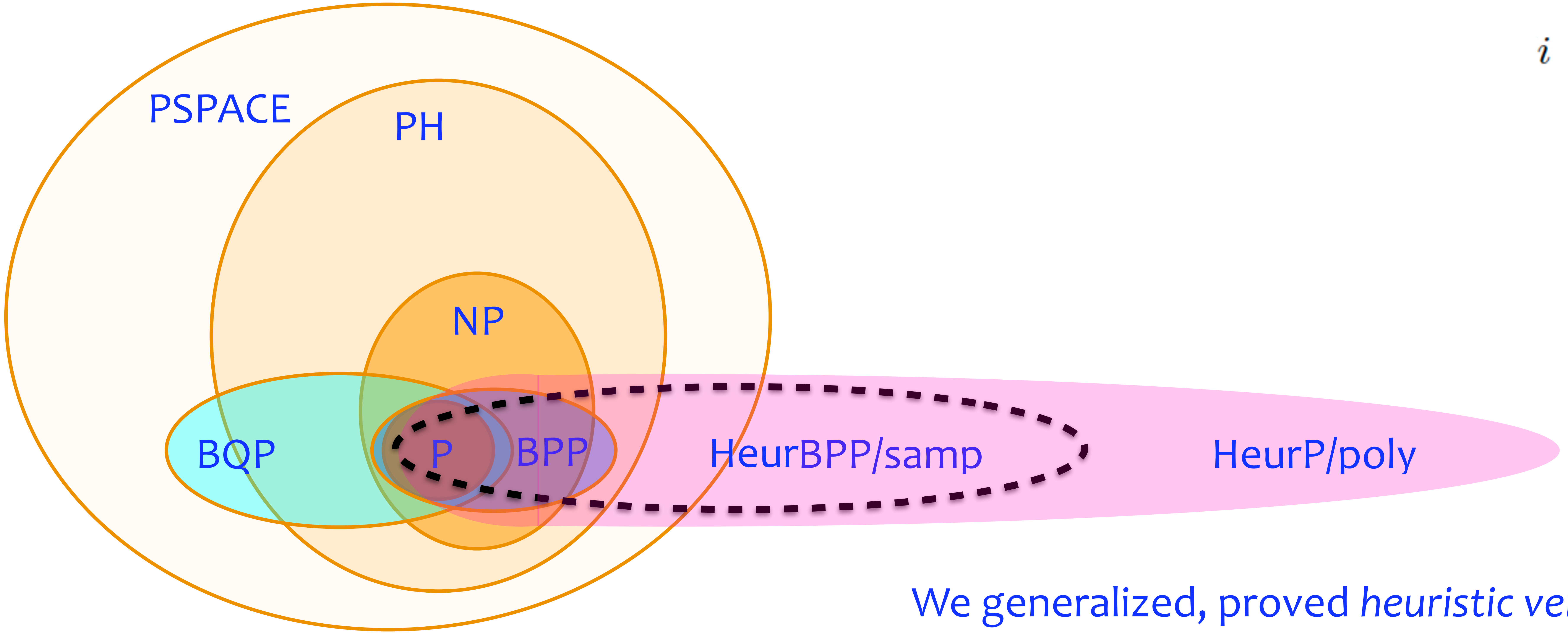
# A simple way out: stronger complexity theoretic assumptions




$i$

Huang et al proved it was in P/poly= problems solvable with polynomial sized classical circuits

# A simple way out: stronger complexity theoretic assumptions



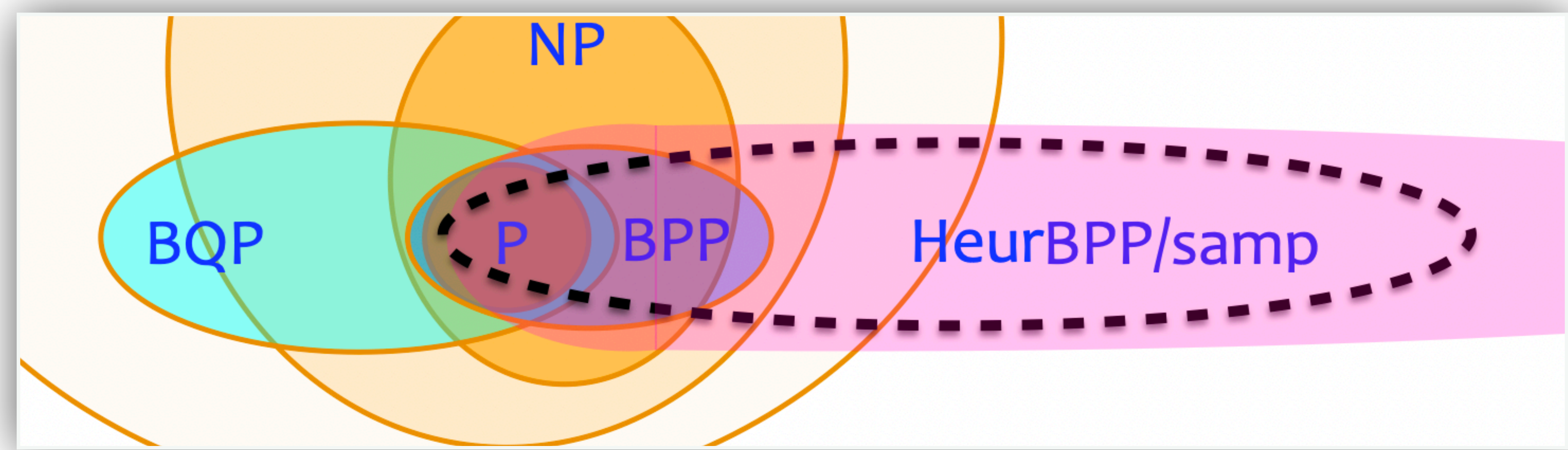

$i$

We generalized, proved *heuristic versions* have same containment, and also that

$$\mathsf{BPP/samp}^{\mathsf{BQP}} \subseteq \mathsf{P/poly}^{\mathsf{Un}\left(\mathsf{ZPP}^{\mathsf{NP}^{\mathsf{BQP}}}\right)}$$

C. Gyurik, VD https://arxiv.org/pdf/2405.18155
S. Marshall, VDhttps://arxiv.org/pdf/2306.16028

# Huh?




$i$

It is believed* BQP is not in (Heur)P/poly

It is even more strongly believed that BQP is not in HeurBPP/samp

*How strongly? At least as strongly as we believe that RSA or Diffie-Hellman are secure against non-uniform adversaries and preprocessing attacks.

If we believe BQP is <u>not</u> in HeurP/poly

$i$

**Meaning:**

-BQP functions cannot be approximated with polynomially sized classical circuits

But then:

-Deep neural networks of size (n,m) can be approximated to arbitrary precision using polynomially sized classical circuits

If we believe BQP is <u>not</u> in HeurP/poly

$i$

**Meaning:**
-BQP functions cannot be approximated with polynomially sized classical circuits

But then:

-Deep neural networks of size (n,m) can be approximated to arbitrary precision using polynomially-sized classical circuits

If we believe BQP is not in HeurP/poly

$i$

**Meaning:**
-BQP functions cannot be approximated with polynomially sized classical circuits

But then:

-Deep neural networks of size $(n, m)$ can be approximated to arbitrary precision using polynomially-sized classical circuits

Deep neural networks cannot even approximate the target functions without exponential growth. So ofc. cannot learn them either.
And neither can **any other future classical ML model.**

**Remark:**

topic: complexity theory and algorithmics… $i$

But, this is a statement about **expressivity**

D**NNs and** other classical methods are *simply not expressive enough.*

**Remark:**

topic: complexity theory and algorithmics… $i$

But, this is a statement about **expressivity**

D**NNs and** other classical methods are *simply not expressive enough.*

C.f.:

**4 Expressive power of tensor-network representations**

In this section we present various relationships between the expressive power of all representations, which constitute the primary results of this work. The proofs of the propositions in this section can be found in the supplementary material.

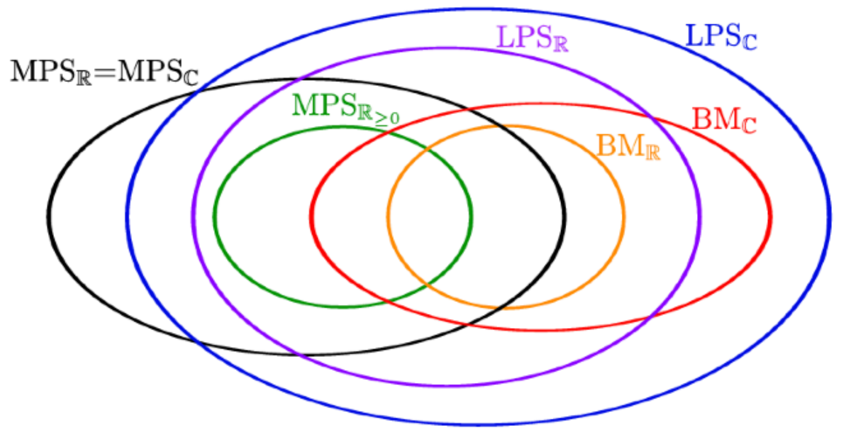



Figure 2: Representation of the sets of non-negative tensors that admit a given tensor-network factorization. In this figure we fix the different ranks of the different tensor networks to be equal.

https://arxiv.org/abs/1907.03741

⟨aQa^L⟩

## Classical **im**possibility

Learning problems with BQP-hard *concepts* which are not in HeurP/poly under some distribution are not learnable under same distribution.

TH: all BQP complete problems have hard-to-learn learning versions under *certain* distributions

Condensed matter, chemistry:

-Bose-Hubbard, XY Hamiltonian (graphs), Fermi-Hubbard on a 2D lattice
-Electronic structure problems

Toward HEP:

-Topological Quantum Field Theory
-Certain supersymmetric theories
-(1+1) massive $\phi^4$
-possibly Kogut-Susskind theories

All of these have BQP-hard variants…. which can be classically not learnable…
(assuming certain claims in complexity theory)

1.**Data gap:** Machine learning comes with **data…** we are given **evaluations of c….** ✓

2.Quantum **learnability:** Must ensure the quantum learner can learn it, and already shallow classical classical circuits are not learnable

3. **Worst case v.s. heuristics:** what does "cannot compute" mean, exactly?

4.What do we **actually mean by learning:** *evaluation or identification*

# The trivial solution

**The supervised learning problem - probably approximately correct (PAC) learning — simplified**

We learn a "concept" from some concept class $C = \{c_j\}_i$, $c_j : \vec{x} \mapsto \{0,1\}$, $\vec{x} \in \{0,1\}^n$

"data"= $\{(\vec{x}_i, c_j(\vec{x}_i))\}_i$ → Machine learning algorithm A → $h := A(data, \cdot) : Data \to Labels$

Data-points ($\vec{x}$'s) come from a *fixed distribution* $\mathscr{D}$

Learner A learns C **efficiently** if ∀ concepts $c_j$, given data labeled by $c_j$
with probability $\geq 1 - \delta$ it outputs $h$, s.t.
$P_{x \in \mathscr{D}}(c_j(x) \neq h(x)) \leq \epsilon$,
with polynomial resources (time, data) in $n, \epsilon^{-1}, \delta^{-1}$

for "evaluate" version this can even be a singleton for classical non-learnability!

**Cor.** Any polynomially sized concept class in BQP with even a single BQP-complete concept is both
(1)classically not-learnable
(2)quantum learnable

Proof:
(1) by representation arguments
(2) try all and pick most likely; this is provably likely a good guess

Non-trivial solutions: i.e provable learning separations with exponentially-sized (or continuous) concepts

**What does not work:**
**parametrized circuits which are trained to fit the concept in a class.**
**In general you hit barren plateaus.**

Non-trivial solutions: i.e provable learning separations with exponentially-sized (or continuous) concepts

**What does work:**
**Exponential learning advantages in learning observables**

(classical hardness will come from previous generic statements
… quantum learnability we will have to work for)

## Exponential learning advantages in learning observables

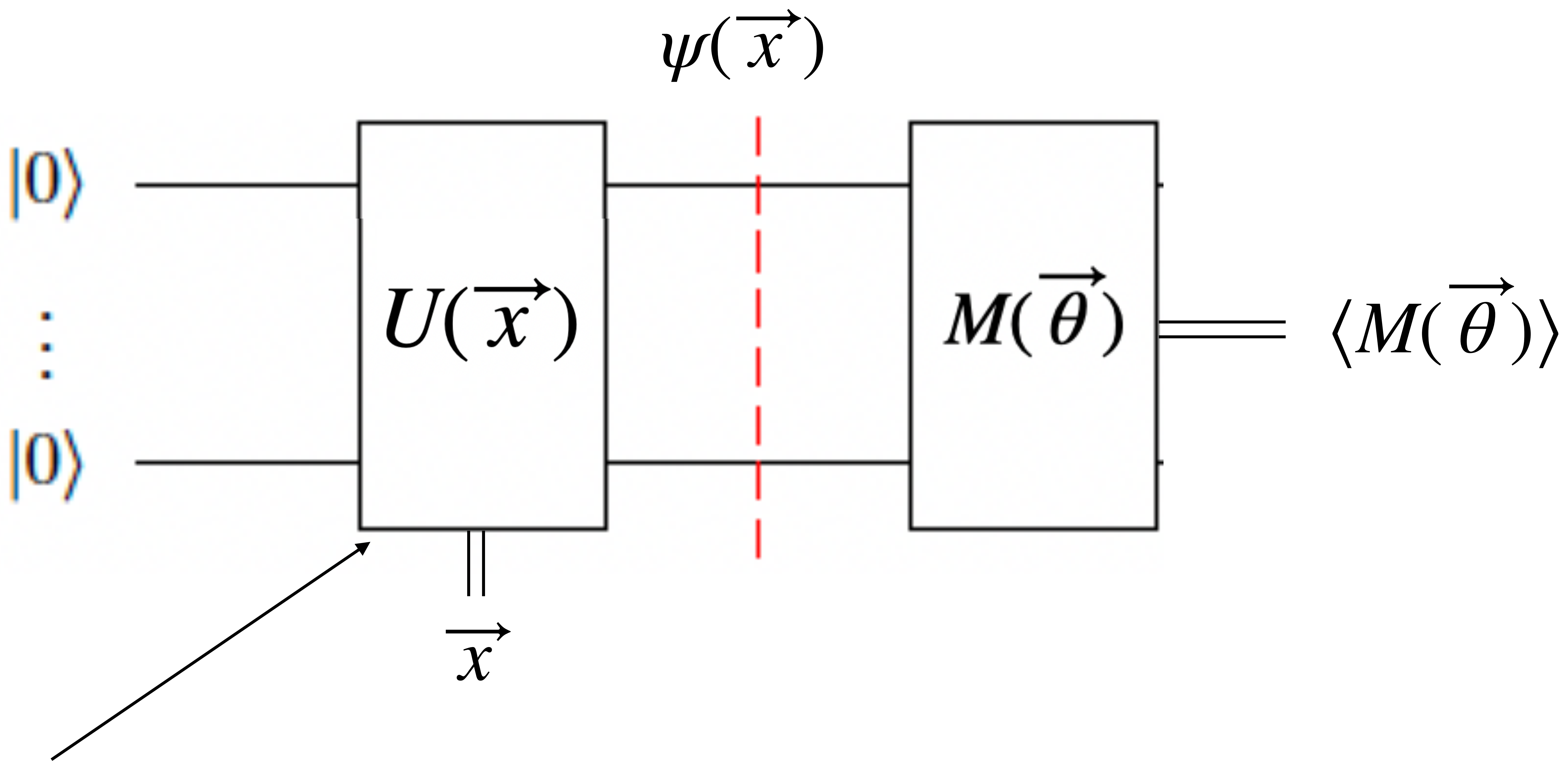
*Setting:*

*Data x describes some complex quantum state* $\psi(x)$ *that is generated. The process* $x \mapsto \psi(x)$ *is known.*
*Label y is an expectation value* $Tr[M\ \psi(x)]$*, for an unknown hermitian M.*

*Task will be: given a new* $\tilde{x}$*, output* $Tr[O\psi(\tilde{x})]$

*The task is *not* to output O (although the QC can do this too).*

## Exponential learning advantages in learning observables

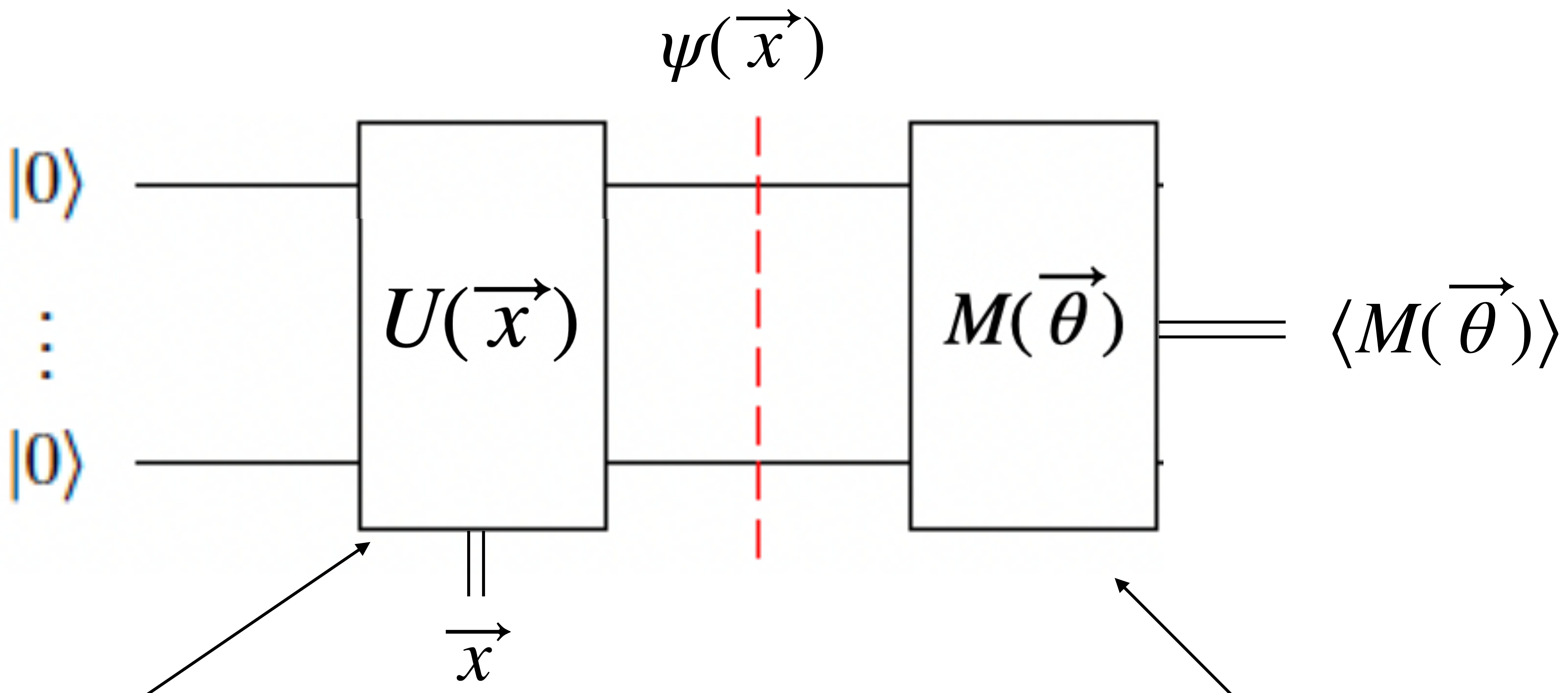



*complex enough* physical process e.g.
-time evolution under a *hard*[1] Hamiltonian
-ground state preparation for a *hard*[2] *Hamiltonian*

*both parametrized by x*

*not controlled (x is random), but heralded (we learn x)*

## Exponential learning advantages in learning observables




*complex enough* physical process e.g.
-time evolution under a *hard*[1] Hamiltonian
-ground state preparation for a *hard*[2] *Hamiltonian*

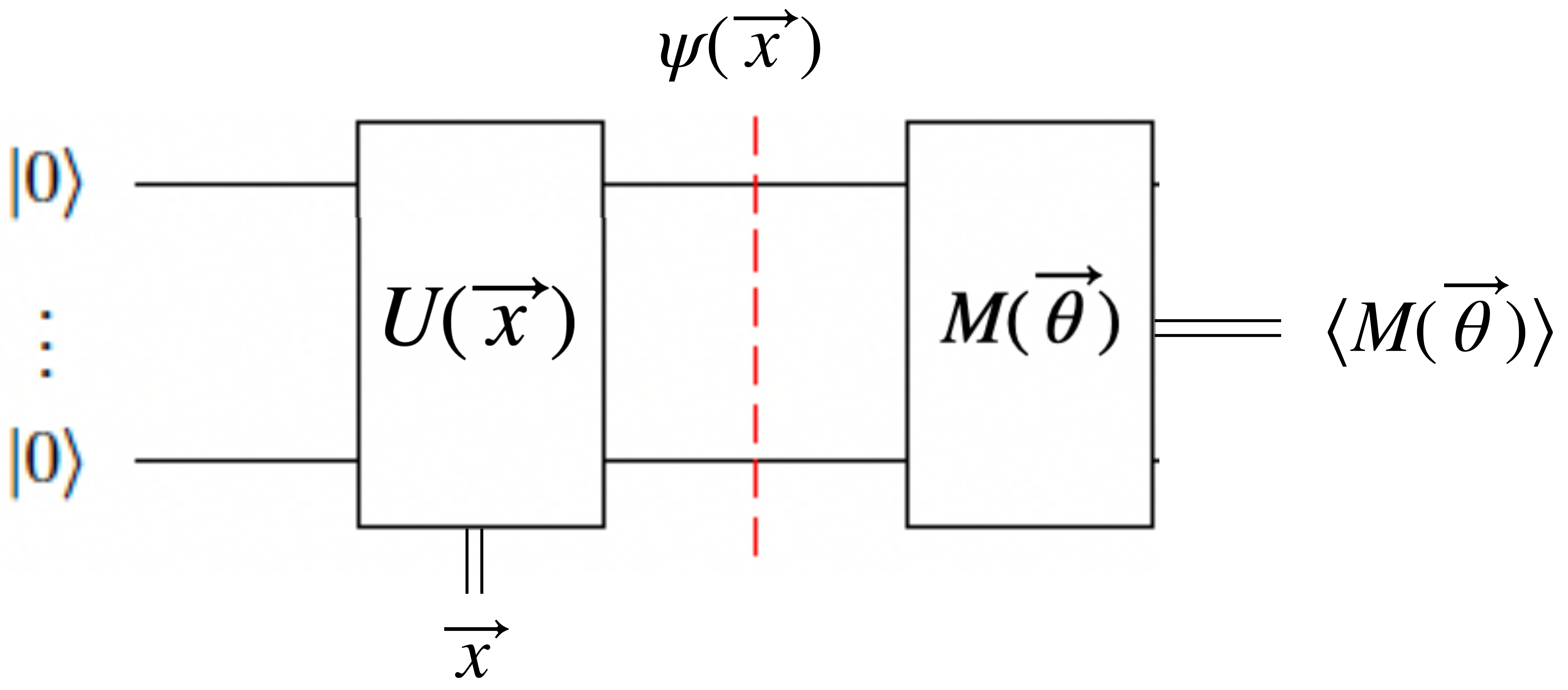
Version 1 (warm-up):

M is a k-local Hamiltonian $H = \sum_j f_j(\vec{\theta})H_j$

*both parametrized by x*

*not controlled (x is random), but heralded (we learn x)*

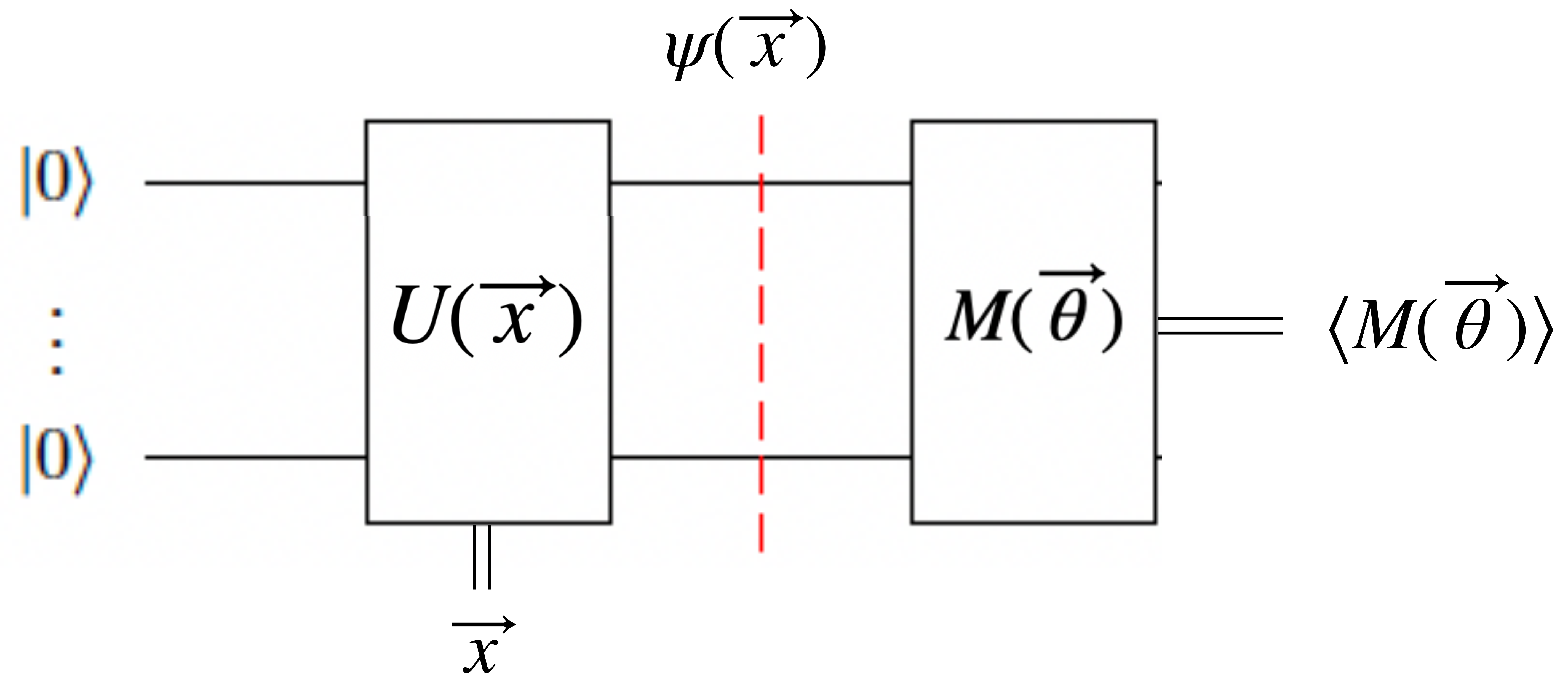
## Exponential learning advantages in learning observables




$$C = \{c_{M(\vec{\theta})}\}_{\vec{\theta}} \qquad Data = \{(\vec{x}, \langle M(\vec{\theta})\rangle_{\psi(\vec{x})})\}$$

**Task: given data above for unknown $\theta$ (=observable), given a new $x'$ output $\langle M(\vec{\theta})\rangle_{\psi(\vec{x})}$**

## Exponential learning advantages in learning observables




**Classical non-learnability:** choose $U$ s.t. $x \mapsto \langle M(\vec{\theta})\rangle_x$ is BQP-complete, so likely not in HeurP/poly

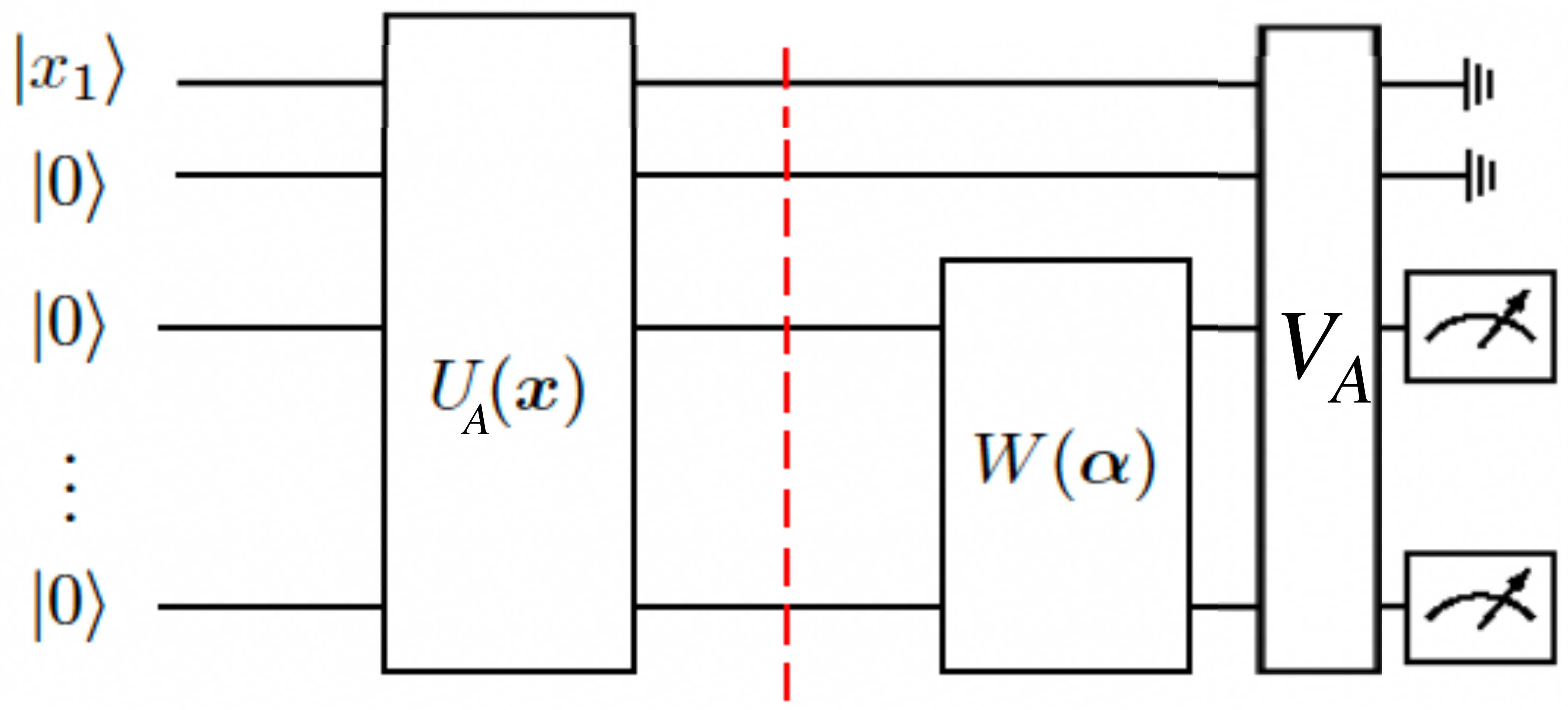
**Quantum learnability:** for every x in dataset D, **a QC can compute** $\langle M(\vec{\theta}')\rangle_x$ **for poly-many different** $\vec{\theta}'$. This yields a noisy system of equations. Nonetheless, LASSO regression can provably find good enough solutions

## Exponential learning advantages in learning observables

More general observables?
Whenever the QC can "learn it" we can use this as a building block.

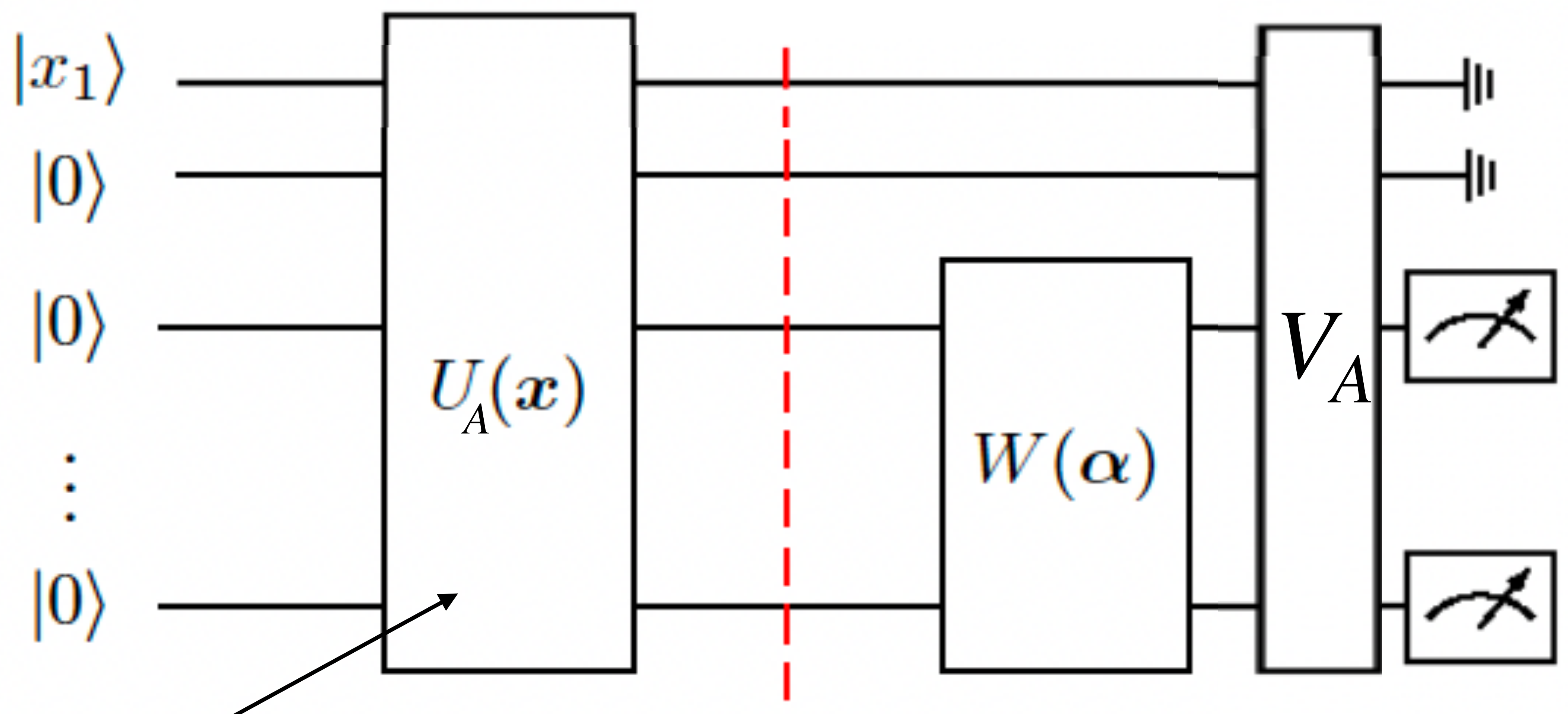
# Exponential learning advantages in learning observables

**TH.** Every (non-adaptive) quantum algorithm A for **learning of a unitary**
(which relies on discterizable or discrete probes and measurements)
induces a classical **data learning separation** (learning an observable parametrized by said unitary)



# Exponential learning advantages in learning observables

**TH.** Every (non-adaptive) quantum algorithm A for learning of a unitary
(which relies on discterizable or discrete probes and measurements)
induces a classical **data learning separation** (learning an observable parametrized by said unitary)




*again, something to do with a "hard" parametrized Hamiltonian*

**Details matter for advantages!**

| | Learning Problem | Existence of a Learning Separation |
|---|---|---|
| **Linear Pauli Observables** | Time Evolution Problem | Yes |
| | Ground State Problem | Yes |
| | Flipped Concepts | No - Classically easy |
| **Unitary-parametrized Observables** | Learning the observable | Yes |
| | Hamiltonian Learning | No - Classically easy |
| | Identifying the concept | Unknown - Classical hardness unknown |

$$\mathcal{F}^{\mathcal{H},O}_{g.s.} = \{ f^{\boldsymbol{\alpha}}(\boldsymbol{x}) \in \mathbb{R} \mid \boldsymbol{\alpha} \in [-1,1]^m \}$$

with $f^{\boldsymbol{\alpha}}(\boldsymbol{x})$ : $\boldsymbol{x} \in \{0,1\}^n \to f^{\boldsymbol{\alpha}}(\boldsymbol{x}) = \mathrm{Tr}[\rho_H(\boldsymbol{x}) O(\boldsymbol{\alpha})]$ & $O(\boldsymbol{\alpha}) = \sum_{i=1} \boldsymbol{\alpha}_i P_i$.

$$C = \{f^{\alpha}(x)\}_{\alpha} \to C' = \{f^{x}(\alpha)\}_{x}$$

## Details matter for separations!

| | Learning Problem | Existence of a Learning Separation |
|---|---|---|
| **Linear Pauli Observables** | Time Evolution Problem | Yes |
| | Ground State Problem | Yes |
| | Flipped Concepts | No - Classically easy |
| **Unitary-parametrized Observables** | Learning the observable | Yes |
| | Hamiltonian Learning | No - Classically easy |
| | Identifying the concept | Unknown - Classical hardness unknown |

$$\mathcal{F}_{g.s.}^{\mathcal{H},O} = \{ f^{\boldsymbol{\alpha}}(\boldsymbol{x}) \in \mathbb{R} \mid \boldsymbol{\alpha} \in [-1,1]^m \}$$

$$\text{with } f^{\boldsymbol{\alpha}}(\boldsymbol{x}) : \ \boldsymbol{x} \in \{0,1\}^n \rightarrow f^{\boldsymbol{\alpha}}(\boldsymbol{x}) = \mathrm{Tr}[\rho_H(\boldsymbol{x}) O(\boldsymbol{\alpha})] \ \ \& \ \ O(\boldsymbol{\alpha}) = \sum_{i=1} \boldsymbol{\alpha}_i P_i .$$

$$C = \{ f^{\alpha}(x) \}_{\alpha} \rightarrow C' = \{ f^{x}(\alpha) \}_{x}$$

# Details matter for separations!

| | **Learning Problem** | **Existence of a Learning Separation** |
|---|---|---|
| **Linear Pauli Observables** | Time Evolution Problem | Yes |
| | Ground State Problem | Yes |
| | Flipped Concepts | No - Classically easy |
| **Unitary-parametrized Observables** | Learning the observable | Yes |
| | Hamiltonian Learning | No - Classically easy |
| | Identifying the concept | Unknown - Classical hardness unknown |

1. **Data gap:** Machine learning comes with **data…** we are given **evaluations of c…** .

2. Quantum **learnability:** Must ensure the quantum learner can learn it, and already shallow classical classical circuits are not learnable

3. **Worst case v.s. heuristics:** what does “cannot compute” mean, exactly?

4. What do we **actually mean by learning:** *evaluation or identification*

# The old… evaluation v.s. identification task

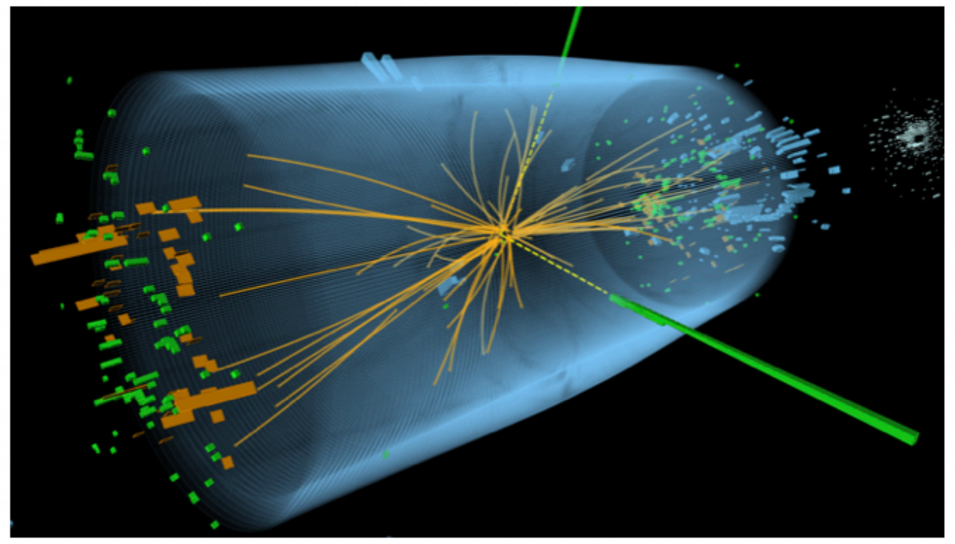
**Example 1:** Supervised Higgs or no Higgs

**Data:**
(measurement outcomes, Higgs background)

**Output:** trained classifier takes on input measurement data (one point)

and outputs *Higgs* or *Background*

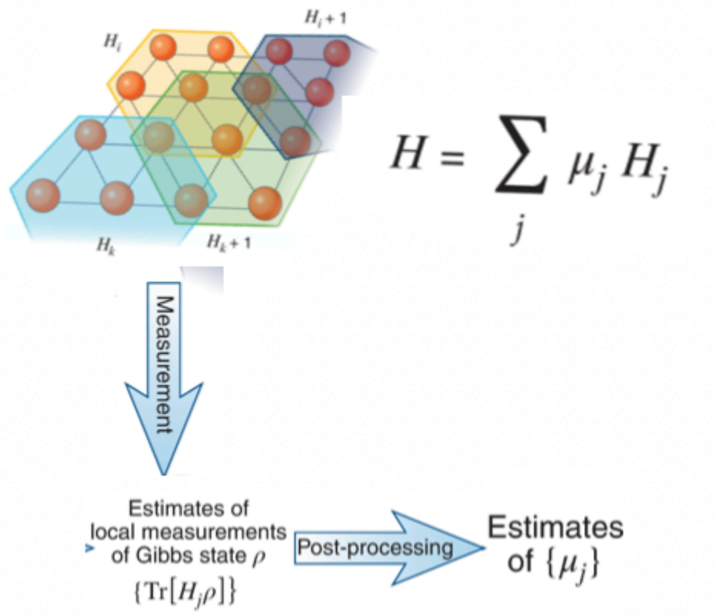
Data includes all the realistic noise



**Example 2:** Hamiltonian learning

**Data:**
Measurements of ground or Gibbs states

**Output:** parameters of the Hamiltonian




## What if the classical learning challenge was: output the *description of the correct classifier*

⟨aQa^L⟩

# The old… evaluation v.s. identification task

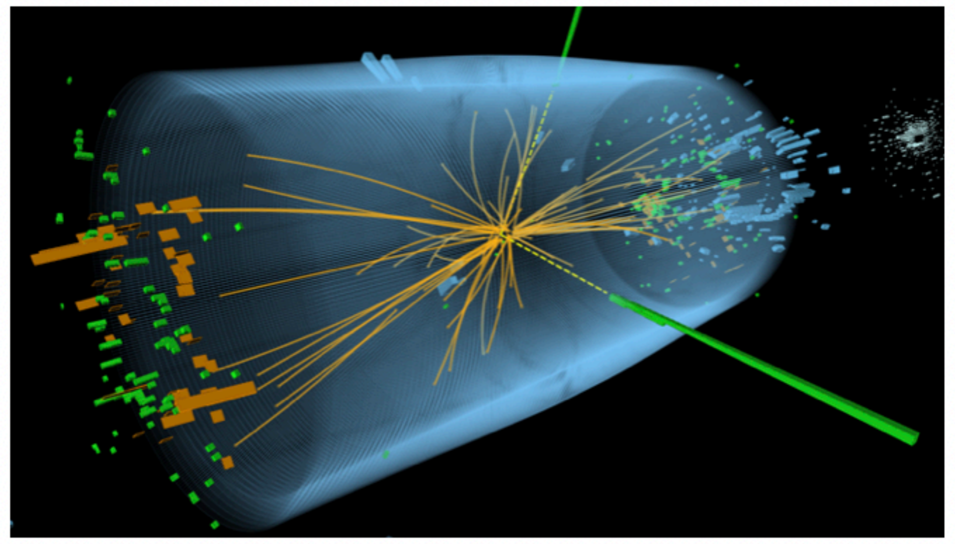
**Example 1:** Supervised Higgs or no Higgs

**Data:**
(measurement outcomes, Higgs background)

**Output:** trained classifier takes on input measurement data (one point)

and outputs *Higgs* or *Background*

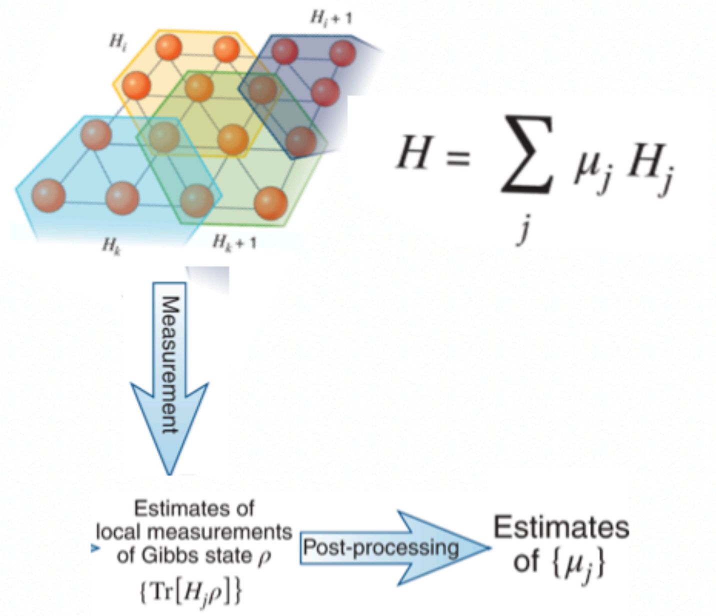
Data includes all the realistic noise



**Example 2:** Hamiltonian learning

**Data:**
Measurements of ground or Gibbs states

**Output:** parameters of the Hamiltonian




**What if the classical learning challenge was: output the *description of the correct classifier***

**Matters because:**

-sometimes we want *just that* like Ham. learning (another example next slide)

-separations in "evaluate" class arguably ***not about learning at all***

-removes our main tool to prove separations

*(very hard challenge - we know sometimes classical works even if concepts intractable )*

**Example: learning of order parameters**

$$c_\alpha(\beta) = \mathrm{sign}\left(\mathrm{Tr}\left[O_\alpha \rho_\beta\right]\right)$$

given labeled data from two different phases (=Hamiltonian settings and phase)
identify which **measurement** would differentiate the phases

⟨aQa^L⟩

**Example: learning of order parameters**

$$c_\alpha(\beta) = \text{sign}\left(\text{Tr}\left[O_\alpha \rho_\beta\right]\right)$$

given labeled data from two different phases (=Hamiltonian settings and phase)
identify which ~~measurement~~ **quantum circuit** would differentiate the phases



https://arxiv.org/pdf/2306.16028

⟨aQa^L⟩

**Example: learning of order parameters**

$$c_\alpha(\beta) = \mathrm{sign}\left(\mathrm{Tr}\left[O_\alpha \rho_\beta\right]\right)$$

given labeled data from two different phases (=Hamiltonian settings and phase)
identify which ~~measurement~~ **quantum circuit** would differentiate the phases

**Oopsie. No-gos.**
If a concept is "evaluate"-learnable for a QC then it is "identify"-learnable for a classical computer.

**Proof**: the "classifier" will include the "quantum training algorithm" and the dataset…
"Learning" will be offloaded to the description..



https://arxiv.org/pdf/2306.16028

⟨aQa^L⟩

**Example: learning of order parameters**

$$c_\alpha(\beta) = \mathrm{sign}\left(\mathrm{Tr}\left[O_\alpha \rho_\beta\right]\right)$$

given labeled data from two different phases (=Hamiltonian settings and phase)
identify which ~~measurement~~ **quantum circuit** would differentiate the phases

**We need to be more precise: limit allowed "descriptions of classifier".**

*If the output has to be "a k-local observable" then it may be leading to a separation*



https://arxiv.org/pdf/2306.16028

**Example: learning of order parameters**

$$c_\alpha(\beta) = \text{sign}\left(\text{Tr}\left[O_\alpha \rho_\beta\right]\right)$$

given labeled data from two different phases (=Hamiltonian settings and phase)
identify which ~~measurement~~ **quantum circuit** would differentiate the phases

**We need to be more precise: limit allowed "descriptions of classifier".**

*If the output has to be "a k-local observable" then it may be leading to a separation*

**More on how to formalize this in extra slides if you want to see…**



https://arxiv.org/pdf/2306.16028

**PAC learning separations for “identify” case**

Proven previously:
-there exist **cryptographic settings** w/ learning separation with classically tractable concepts and with fixed hypotheses classes

-



https://arxiv.org/pdf/2306.16028

# DCR separation

Concept class:

$f(x) = \sqrt[3]{x} \mod N$

Hypothesis class

$\{g_d(x) = x^d \mod N\}_d$

For DCR $\exists\ d_N$ s.t. $f(x) = g_{d_N}(x)$

Computing it given N it is believed to be intractable

Data is generatable.

Learning →computing $d_N$ is easy



https://arxiv.org/pdf/2306.16028

⟨aQa⟩

# DCR separation

Concept class:

$$f(x) = \sqrt[3]{x} \mod N$$

Hypothesis class

$$\{g_d(x) = x^d \mod N\}_d$$

Its about hard representations of easy functions…

Its not about evaluation…

Again… no way to map onto quantum functions



https://arxiv.org/pdf/2306.16028

⟨aQa^L⟩

## PAC learning separations for "identify" case for quantum functions

**New result (paper in prep):**

TH: learning separation for the **proper PAC identify case** (identify = pick for concept class) with quantum functions possible under:

-certain assumptions on overlaps of concepts
*(concepts should be "quite different", but we know this property can be satisfied)*

-**stronger** complexity theoretic assumptions, namely unless **BQP in fourth level of PH**

(probably both can be improved)

## Some general thoughts on why QML and limitations and strengths of approach

1) Fundamental questions answered with a yes (deferred to complexity theory).

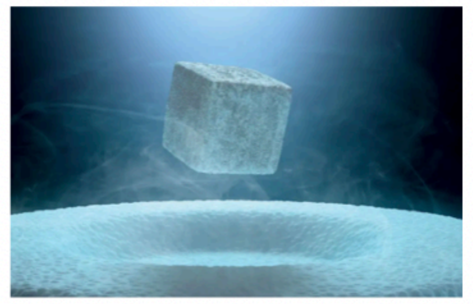
2) Consequences on practice? Unclear for reasons:
   I. about asymptotics, real world usually fixed size
   II. not perfectly aligned with natural problems
   III. there is more to life beyond PAC

3) QML could have applications
   *even before advantage from simulations*
   circuits given **by algorithm** v.s. smallest we can find given data

**Example 3:** Tc prediction (superconductivity)

Here, the function is not *fundamentally* unknown.
*Given* $a\ lot^{a\ lot^{alot}}$ *of compute, you (probably) could compute it from* ***first principles***

***We are looking for a much more concise representation of same function***



4) Theory can point toward "bad idea" learning settings
   (e.g. if learning problem is P(data | input), and input set is small… it *is* classically learnable)

5) Advantage can come from other reasons than expressivity, easier to identify in rigorous framework

## Consequences on near-term quantum computing

1) Learning separations **achievable** when simulation achievable (and probably earlier see 2)

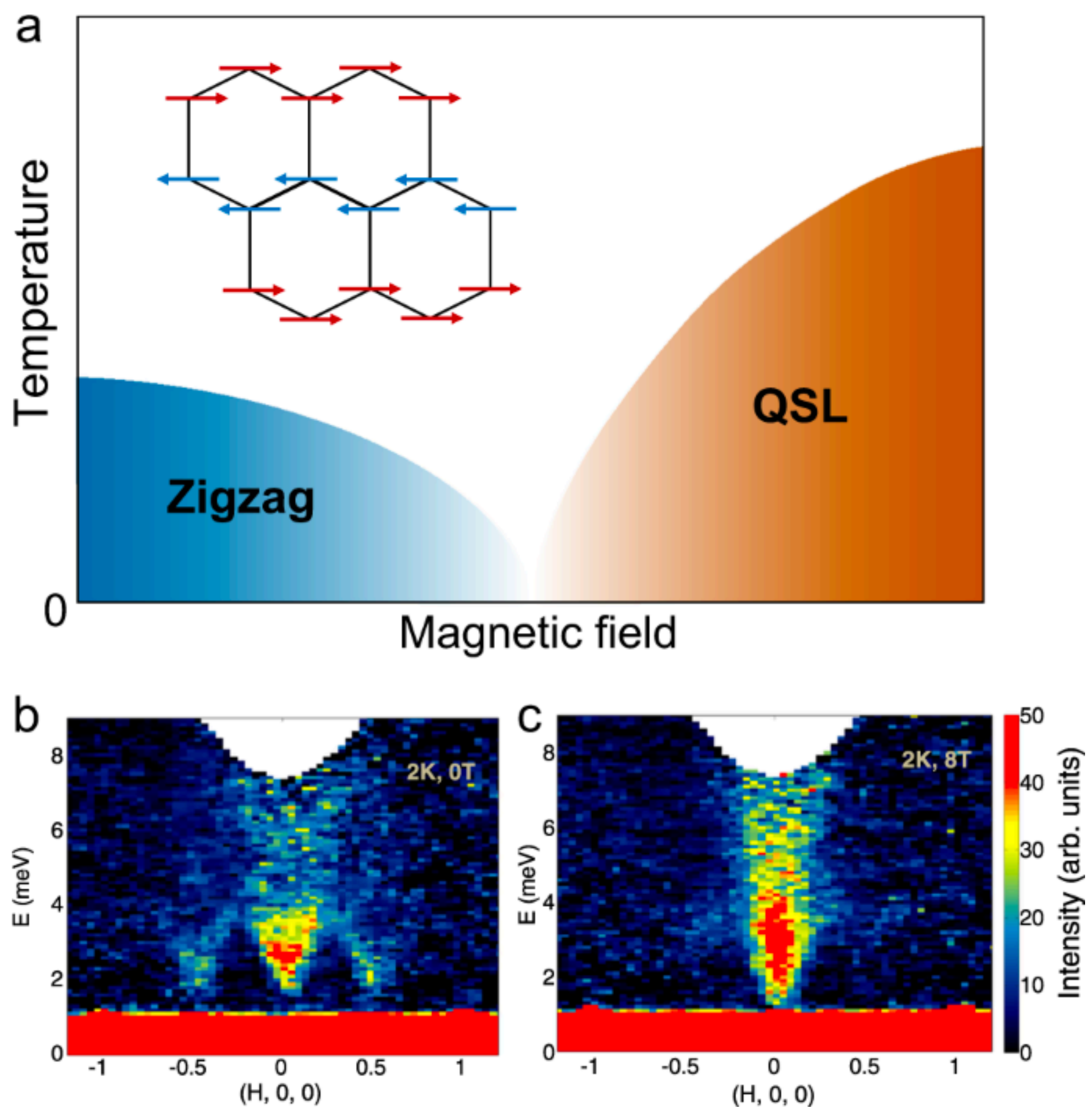
2) Learning separations **conceivable** with fewer resources than for simulation

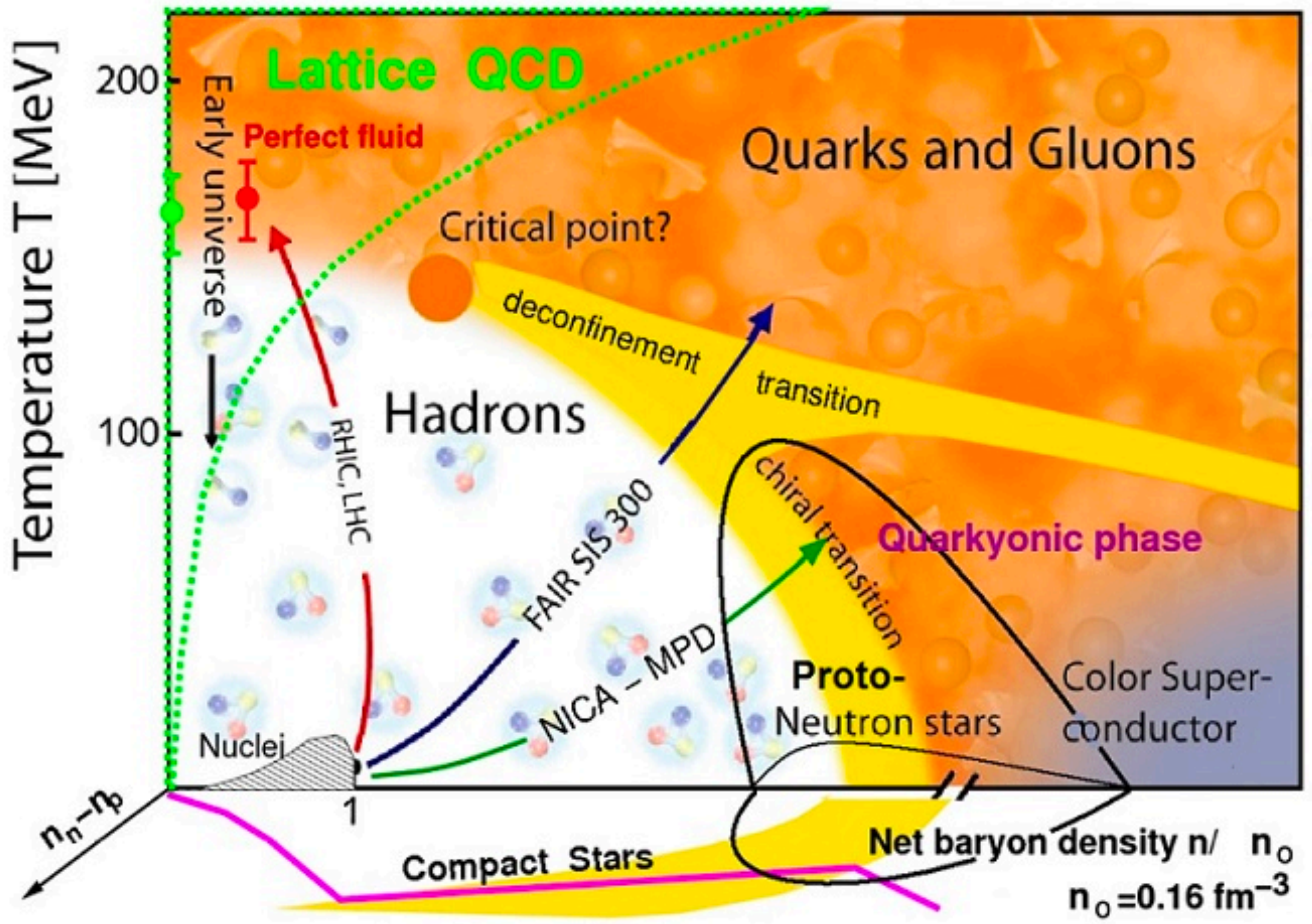
-analogy: trained NNs make a significant dent in NP-hard problems because HeurP/samp likely contains more than BPP

-trained QNNs can do even more with less
open: HeurBQP/samp - how far does it go in e.g. QMA?

(there can be better, smaller quantum circuits for simulation that we can find given data)

QML application, and QML analysis seem to point to



## Summary

**Learning separations pervasive…but a lot of details that matter:**

**THANKS TO:**

*Riccardo Molteni*

*Casper Gyurik*

Simon Marshall

*Mahtab Yaghubi Rad*

**We have PhD openings for this sort of thing!**

## QML and cats v dogs and big data

QML may disappoint if we ask it to do… what it was not *meant* to do

(how precisely do you want to get cats v. dogs!? Is 99% not good enough?)

## QML and complexity theory

Complexity theory may disappoint for similar reasons.

It “works” asymptotically (or “really really big”)
Condensed matter: Avogadro number
(but in exotic matter! Topological etc…)

HEP: Lattice gauge theories, energy scales…

⟨aQa^L⟩

Main parts of learning theory

Concept class
$\{c_j : \mathscr{X} \to \{0,1\}\}$,
distribution over $\mathscr{X}$

Machine learning algorithm A

hypothesis class
$\{h : \mathscr{X} \to \{0,1\}\}$

Class.
Class. Obfs.
Quant.

Class./Quant.

Class./Quant.

*A small zoo of types of separations make sense (CC, CQ) vs (QC, QQ)*

*(some don't)*

| First proposed in | Concepts based on | Separation | Complexity of concepts |
|---|---|---|---|
| [LAT21] | Discrete logarithm | CC/QQ | $\notin$ BPP |
| [SG04] | Discrete cube root | CC/QC | $\notin$ BPP but $\in$ P/poly |
| Section 3.3 | Modular exponentiation | CC/QC | $\in$ P |
| Section 3.4 | Discrete cube root *identification* | $C_{\mathcal{H}}/Q_{\mathcal{H}}$ | $\in$ P |
| Section 4.1 | Genuine quantum process | CC/QQ | $\notin$ HeurP/poly but $\in$ BQP |

| Properties / Concepts | Hardness assumption | Separation | Binary? | Singleton? |
|---|---|---|---|---|
| Fix sequence $\{(p_n, a_n)\}_{n\in\mathbb{N}}$: $c_i(x)$ in Def. 14 | DLP-fixed | CC/QQ | Yes | Yes |
| $c_{(a,p)}(x)$ in Def. 19 | DLP | CC/QQ | Yes | No |
| $c(x, a, p)$ in Def. 20 | DLP | CC/QQ | No | Yes |

DCRI binary a class, with concepts in *P.*
Proven to be hard to learn (so it *is* about learning)
only in the restricted hypothesis case



https://arxiv.org/abs/2208.06339

⟨aQa^L⟩

# Other applications and related results

nature > nature communications > articles > article

Article | Open access | Published: 06 July 2024

**Shadows of quantum machine learning**

Sofiene Jerbi, Casper Gyurik, Simon C. Marshall, Riccardo Molteni & Vedran Dunjko

*Nature Communications* **15**, Article number: 5676 (2024) | Cite this article

→ Provable advantages in QML where *only the training* is quantum, and use classical

**On the relation between trainability and dequantization of variational quantum learning models**

Elies Gil-Fuster,[1,2] Casper Gyurik,[3] Adrián Pérez-Salinas,[3,4] and Vedran Dunjko[3,5]
[1]*Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany*
[2]*Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany*
[3]*⟨aQa^L⟩ Applied Quantum Algorithms, Universiteit Leiden*
[4]*Lorentz Instituut, Universiteit Leiden, Niels Bohrweg 2, 2333 CA Leiden, Netherlands*
[5]*LIACS, Universiteit Leiden, Niels Bohrweg 1, 2333 CA Leiden, Netherlands*
(Dated: June 12, 2024)

→ Existence of vari veryational [sic] QML models which are trainable and non-dequantizable

On Bounded Advice Classes

Simon C. Marshall * Casper Gyurik Vedran Dunjko
Universiteit Leiden

→ Understanding advice when the advice is computationally bounded

Improved separation between quantum and classical computers for sampling and functional tasks

Simon C. Marshall* Scott Aaronson† Vedran Dunjko*

→ Unintended spin-off: best known separation between classical and quantum computers