Exponential separations between classical and quantum learners

arXiv:2306.16028

Casper Gyurik

joint work w/ Vedran Dunjko

applied Quantum algorithms (aQa), Leiden University

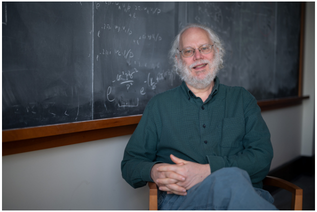
QTML 2023

Universiteit Leiden
The Netherlands

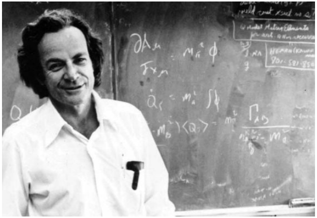
Quantum Software Consortium

Quantum vs Classical computing

where do we *know* quantum exponentially beats classical computation?



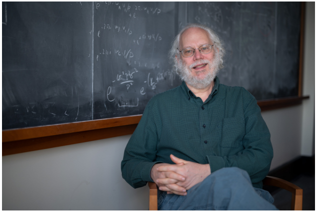
(a) Shor: cryptanalysis



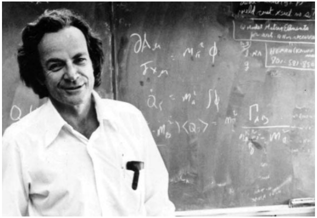
(b) Feynman: studying physics

Quantum vs Classical ~~computing~~ ML

where do we *know* quantum exponentially beats classical ~~computation~~ ML?



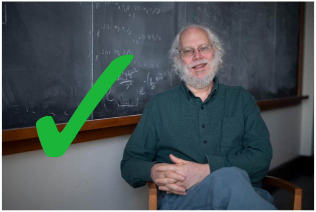
(a) Shor: cryptanalysis



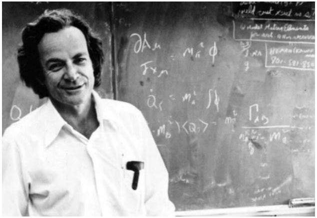
(b) Feynman: studying physics

Quantum vs Classical ~~computing~~ ML

where do we *know* quantum exponentially beats classical ~~computation~~ ML?



(a) Shor: cryptanalysis

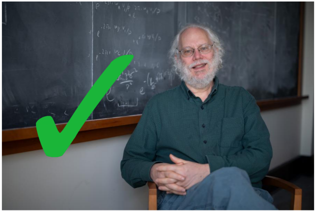


(b) Feynman: studying physics

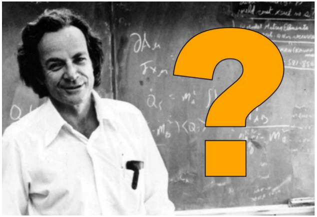
- Many/all known separations are based on cryptanalysis.
  - "Quantum versus classical learnability" (Servedio & Gortler – SICOMP 2004), "On the quantum versus classical learnability of discrete distributions" (Sweke, Seifert et al. – Quantum 2021), "A rigorous and robust quantum speed-up in supervised machine learning" (Lui, Arunachalam & Temme – Nat. Phys. 2021) "Parametrized Quantum Policies for Reinforcement Learning" (Jerbi, Gyurik et al. – NeurIPS 2021).

Quantum vs Classical ~~computing~~ ML

where do we *know* quantum exponentially beats classical ~~computation~~ ML?



(a) Shor: cryptanalysis

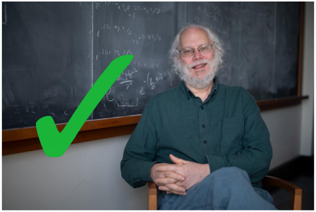


(b) Feynman: studying physics

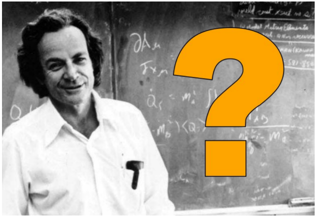
- Many/all known separations are based on cryptanalysis!
  - "Quantum versus classical learnability" (Servedio & Gortler – SICOMP 2004), "On the quantum versus classical learnability of discrete distributions" (Sweke, Seifert et al. – Quantum 2021), "A rigorous and robust quantum speed-up in supervised machine learning" (Lui, Arunachalam & Temme – Nat. Phys. 2021) "Parametrized Quantum Policies for Reinforcement Learning" (Jerbi, Gyurik et al. – NeurIPS 2021).
- What about QML advantages for data from "quantum processes"?
  - Common folklore that QML should have advantages here.

Quantum vs Classical ~~computing~~ ML

where do we *know* quantum exponentially beats classical ~~computation~~ ML?



(a) Shor: cryptanalysis



(b) Feynman: studying physics

- Many/all known separations are based on cryptanalysis!
  - "Quantum versus classical learnability" (Servedio & Gortler – SICOMP 2004), "On the quantum versus classical learnability of discrete distributions" (Sweke, Seifert et al. – Quantum 2021), "A rigorous and robust quantum speed-up in supervised machine learning" (Lui, Arunachalam & Temme – Nat. Phys. 2021) "Parametrized Quantum Policies for Reinforcement Learning" (Jerbi, Gyurik et al. – NeurIPS 2021).
- What about QML advantages for data from "quantum processes"?
  - Common folklore that QML should have advantages here.

Our work: classical data (i.e., measurements of quantum system), **no quantum states**.

Separations classical and quantum learners

understanding when/why/how quantum outperforms classical in ML

Main research question

When do complexity theoretic separations imply (various kinds of) learning separations?

Separations classical and quantum learners

understanding when/why/how quantum outperforms classical in ML

Main research question

When do complexity theoretic separations imply (various kinds of) learning separations?

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Separations classical and quantum learners

understanding when/why/how quantum outperforms classical in ML

Main research question

When do complexity theoretic separations imply (various kinds of) learning separations?

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of the above construction *fails*? and how do we get around this?

Defining a learning separation

the PAC learning framework

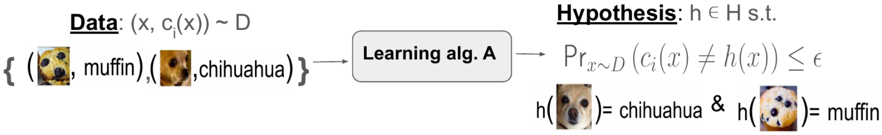
PAC learning problem: concept class $\mathcal{C} = \{c_i\}_{i \in I}$ & data distribution $\mathcal{D}$.

Defining a learning separation

the PAC learning framework

PAC learning problem: concept class $\mathcal{C} = \{c_i\}_{i \in I}$ & data distribution $\mathcal{D}$.

- Solved by learning consisting of learning algorithm A and hypothesis class H:



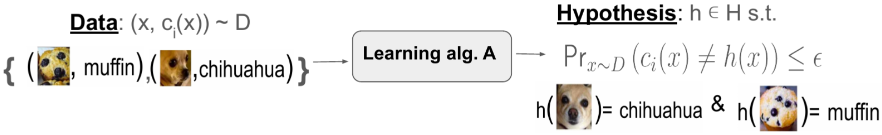
  - "efficient" w.r.t **both** time and number of data samples.

Defining a learning separation

the PAC learning framework

PAC learning problem: concept class $\mathcal{C} = \{c_i\}_{i \in I}$ & data distribution $\mathcal{D}$.

- Solved by learning consisting of learning algorithm A and hypothesis class H:



  - "efficient" w.r.t **both** time and number of data samples.

Definition (learning separations)

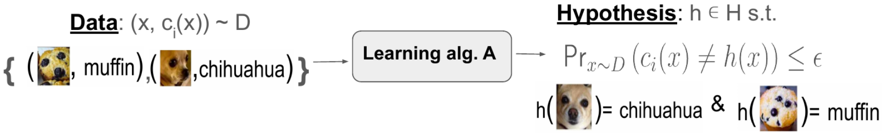
- efficient solvable with either A or H **quantum**,
- not efficiently solvable with **classical** A and **classical** H.

Defining a learning separation

the PAC learning framework

PAC learning problem: concept class $\mathcal{C} = \{c_i\}_{i \in I}$ & data distribution $\mathcal{D}$.

- Solved by learning consisting of learning algorithm A and hypothesis class H:



  - "efficient" w.r.t **both** time and number of data samples.

Definition (learning separations)

- efficient solvable with either A or H **quantum**,
- not efficiently solvable with **classical** A and **classical** H.

Different tasks that might require quantum: *evaluation* or *identification*.

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?
- and precisely *what learning task* do they show is impossible classically?

Three major differences computation vs learning:

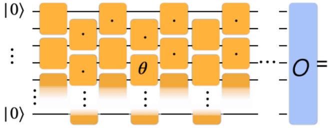
- **Data gap**: machine learning comes with data, which radically enhances power.

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?
- and precisely *what learning task* do they show is impossible classically?

Three major differences computation vs learning:

- **Data gap**: machine learning comes with data, which radically enhances power.



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

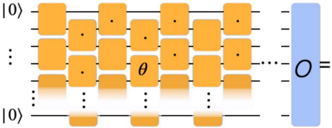
Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Three major differences computation vs learning:

- **Data gap**: machine learning comes with data, which radically enhances power.



$\Rightarrow$ $f(\theta) = \mathrm{Tr}[\rho(\theta)O(\cdot)]$
$= \alpha \sin(\theta + \beta) + \gamma$
-> 3 evals for fit!

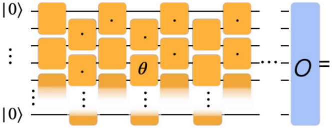
Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Three major differences computation vs learning:

- **Data gap**: machine learning comes with data, which radically enhances power.



$\Rightarrow$ $f(\theta) = \mathrm{Tr}[\rho(\theta) O(\cdot)]$
$= \alpha \sin(\theta + \beta) + \gamma$
-> 3 evals for fit!

- **Quantum learnability**: e.g., even shallow q. circuits not q. learnable[1].

[1] see Nietner et al. (arXiv:2305.05765)

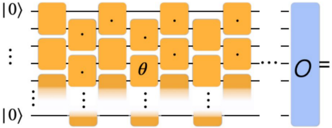
Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Three major differences computation vs learning:

- **Data gap**: machine learning comes with data, which radically enhances power.



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

$= \alpha \sin(\theta + \beta) + \gamma$

-> 3 evals for fit!

- **Quantum learnability**: e.g., even shallow q. circuits not q. learnable[1].
- **Worst-case/Heuristic**: ML cares about correctness on fraction of inputs.

[1] see Nietner et al. (arXiv:2305.05765)

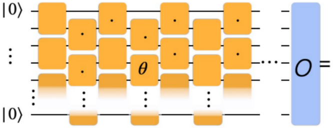
Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Three major differences computation vs learning:

- **Data gap**: machine learning comes with data, which radically enhances power.



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

$= \alpha \sin(\theta + \beta) + \gamma$

-> 3 evals for fit!

- **Quantum learnability**: e.g., even shallow q. circuits not q. learnable[1].
- **Worst-case/Heuristic**: ML cares about correctness on fraction of inputs.

But, how do existing learning separations overcome this?

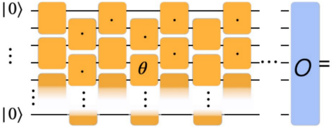
[1] see Nietner et al. (arXiv:2305.05765)

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?
- and precisely *what learning task* do they show is impossible classically?

Three major differences computation vs learning:

- **Data gap**: machine learning comes with data, which radically enhances power.



$\Rightarrow \quad f(\theta) = \mathrm{Tr}[\rho(\theta) O(\cdot)]$
$= \alpha \sin(\theta + \beta) + \gamma$
-> 3 evals for fit!

- **Quantum learnability**: e.g., even shallow q. circuits not q. learnable[1].
- **Worst-case/Heuristic**: ML cares about correctness on fraction of inputs.

But, how do existing learning separations overcome this?

Formula to proving learning separations from complexity theory

Efficient learner + **Efficient data** $\Longrightarrow$ Efficient (non-learning) algorithm.

- classical hardness of learning directly from heuristic hardness of concepts!

[1] see Nietner et al. (arXiv:2305.05765)

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Existing learning separations: use concepts that are heuristicly hard classically!

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Existing learning separations: use concepts that are heuristicly hard classically!

- classical non-learnability strongly relies on demanding *evaluation* from classical learner.

- more like computational separation instead of learning (quantum learner needs no data).

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Existing learning separations: use concepts that are heuristically hard classically!

- classical non-learnability strongly relies on demanding *evaluation* from classical learner.

- more like computational separation instead of learning (quantum learner needs no data).

Our result (part 1)

Modular exponentiation concept class: $c_d(x) = x^d \mod N$.

- Efficiently evaluatable $\implies$ hardness must lie in identification!

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Existing learning separations: use concepts that are heuristically hard classically!

- classical non-learnability strongly relies on demanding *evaluation* from classical learner.

- more like computational separation instead of learning (quantum learner needs no data).

Our result (part 1)

Modular exponentiation concept class: $c_d(x) = x^d \mod N$.

- Efficiently evaluatable $\implies$ hardness must lie in identification!

Classical non-learnability: identification of d breaks $\texttt{RSA}$-encryption.

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Existing learning separations: use concepts that are heuristically hard classically!

- classical non-learnability strongly relies on demanding *evaluation* from classical learner.

- more like computational separation instead of learning (quantum learner needs no data).

Our result (part 1)

Modular exponentiation concept class: $c_d(x) = x^d \mod N$.

- Efficiently evaluatable $\implies$ hardness must lie in identification!

Classical non-learnability: identification of d breaks RSA-encryption.

Quantum learnability: example $(x, x^d \mod N)$ induces congruence $d \equiv \ell \mod r$

- $r = \mathrm{ord}(x)$ and $\ell = \log_x x^d$.

Understanding existing learning separations

RQ1: How do *existing learning separations* build on computational separations?

- and precisely *what learning task* do they show is impossible classically?

Existing learning separations: use concepts that are heuristically hard classically!

- classical non-learnability strongly relies on demanding *evaluation* from classical learner.

- more like computational separation instead of learning (quantum learner needs no data).

Our result (part 1)

Modular exponentiation concept class: $c_d(x) = x^d \mod N$.

- Efficiently evaluatable $\implies$ hardness must lie in identification!

Classical non-learnability: identification of d breaks $\texttt{RSA}$-encryption.

Quantum learnability: example $(x, x^d \mod N)$ induces congruence $d \equiv \ell \mod r$

- $r = \mathrm{ord}(x)$ and $\ell = \log_x x^d$.

Proof: poly many examples suffice to determine d using congruences.

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction ***fails***? and how to get around this?

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction *fails*? and how to get around this?

For "*quantum processes*" (i.e., BQP function) cannot assume efficient data generation.

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction *fails*? and how to get around this?

For "*quantum processes*" (i.e., BQP function) cannot assume efficient data generation.

Our new approach to learning separations

Classical nonlearnability: $c \notin$ HeurP/poly.

- /poly $\leftrightarrow$ hard even when given data.

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction *fails*? and how to get around this?

For "*quantum processes*" (i.e., BQP function) cannot assume efficient data generation.

Our new approach to learning separations

Classical nonlearnability: $c \notin$ HeurP/poly.

- /poly $\leftrightarrow$ hard even when given data.

Quantum learnability: $c \in$ BQP, $|C| = \mathrm{poly}(n)$.

- Emprical risk minimization works!

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction *fails*? and how to get around this?

For "*quantum processes*" (i.e., BQP function) cannot assume efficient data generation.

Our new approach to learning separations

Classical nonlearnability: $c \notin$ HeurP/poly.

- /poly $\leftrightarrow$ hard even when given data.

Quantum learnability: $c \in$ BQP, $|C| = \mathrm{poly}(n)$.

- Emprical risk minimization works!

Ingredients for learning sep: $L \in$ BQP s.t. $(L, \mathcal{D}) \notin$ HeurP/poly.

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction *fails*? and how to get around this?

For "*quantum processes*" (i.e., BQP function) cannot assume efficient data generation.

Our new approach to learning separations

Classical nonlearnability: $c \notin$ HeurP/poly.
- /poly $\leftrightarrow$ hard even when given data.

Quantum learnability: $c \in$ BQP, $|C| = \mathrm{poly}(n)$.
- Emprical risk minimization works!

Ingredients for learning sep: $L \in$ BQP s.t. $(L, \mathcal{D}) \notin$ HeurP/poly.

Theorem

Any $\exists L \in$ BQP such that $L \notin$ P/poly and it is random self-reducible, can be used to construct a learning separation from every BQP-complete problem.

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction *fails*? and how to get around this?

For "*quantum processes*" (i.e., BQP function) cannot assume efficient data generation.

Our new approach to learning separations

Classical nonlearnability: $c \notin$ HeurP/poly.

- /poly $\leftrightarrow$ hard even when given data.

Quantum learnability: $c \in$ BQP, $|C| = \mathrm{poly}(n)$.

- Emprical risk minimization works!

Ingredients for learning sep: $L \in$ BQP s.t. $(L, \mathcal{D}) \notin$ HeurP/poly.

Theorem

Any $\exists L \in$ BQP such that $L \notin$ P/poly and it is random self-reducible, can be used to construct a learning separation from every BQP-complete problem.

- e.g., can use DLP to build learning seps. from BQP-complete problems!

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction *fails*? and how to get around this?

For "*quantum processes*" (i.e., BQP function) cannot assume efficient data generation.

Our new approach to learning separations

Classical nonlearnability: $c \notin$ HeurP/poly.

- /poly $\leftrightarrow$ hard even when given data.

Quantum learnability: $c \in$ BQP, $|C| = \mathrm{poly}(n)$.

- Emprical risk minimization works!

Ingredients for learning sep: $L \in$ BQP s.t. $(L, \mathcal{D}) \notin$ HeurP/poly.

Theorem

Any $\exists L \in$ BQP such that $L \notin$ P/poly and it is random self-reducible, can be used to construct a learning separation from every BQP-complete problem.

- e.g., can use DLP to build learning seps. from BQP-complete problems!

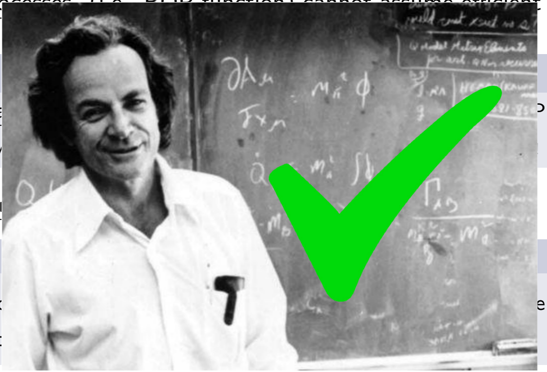
$$c_H(\beta) = \mathrm{sign}\left(\left|\langle 0^n| e^{iH(\beta)} Z_1 e^{-iH(\beta)} |0^n\rangle\right|^2 - \frac{1}{2}\right) \implies \text{“learning separation!”}$$

Learning separations from quantum processes

RQ2: Why don't we have similar learning separarions for "quantum process"?

- what part of existing construction *fails*? and how to get around this?

For "*quantum pro*… data generation.



Our new approach

Classical nonlearna… P, $|C| = \mathrm{poly}(n)$.

- /poly ↔ hard ev…

Ingredients for lea…

Theorem

Any $\exists L \in$ BQP suc… e used to construct a learning separatio…

- e.g., can use DLP to build learning seps. from BQP-complete problems!

$$c_H(\beta) = \mathrm{sign}\left(\left| \langle 0^n | e^{iH(\beta)} Z_1 e^{-iH(\beta)} | 0^n \rangle \right|^2 - \frac{1}{2} \right) \implies \text{"learning separation!"}$$

Thank you!