The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

Brian Coyle, **Daniel Mills**, Vincent Danos, Elham Kashefi

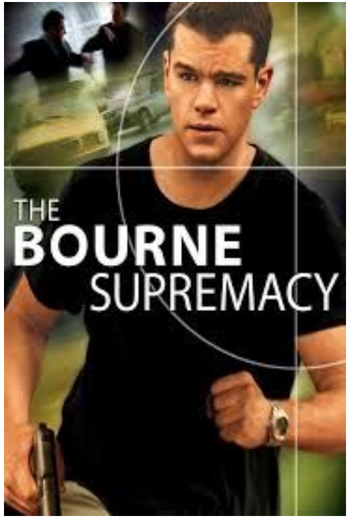
THE UNIVERSITY of EDINBURGH

arXiv:1904.02214

The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

The Born Supremacy:
Quantum Advantage and Training of an Ising Born Machine

The BORN Supremacy



The BORN Supremacy

The BORN Supremacy

The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

The Born Supremacy:

Quantum Advantage and Training of an Ising Born Machine

The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

The Born Supremacy:

Quantum Advantage and Training of an Ising Born Machine

The Born Supremacy:

Quantum Advantage and Training of an Ising Born Machine

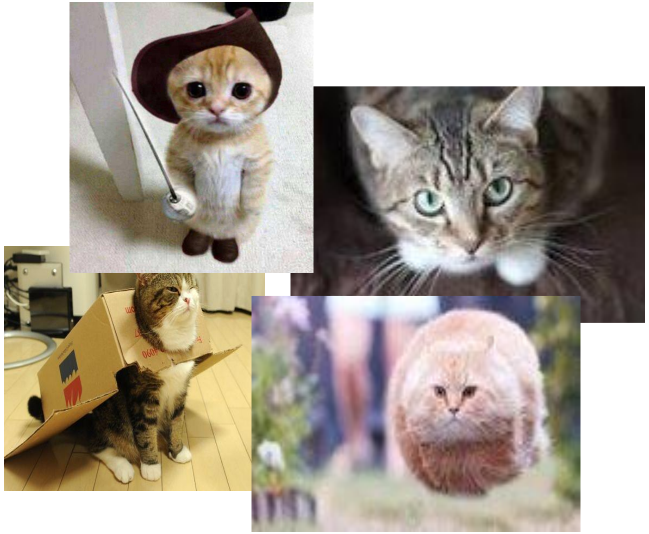
The Born Supremacy:

Quantum Advantage and Training of an Ising Born Machine

Generative Modeling

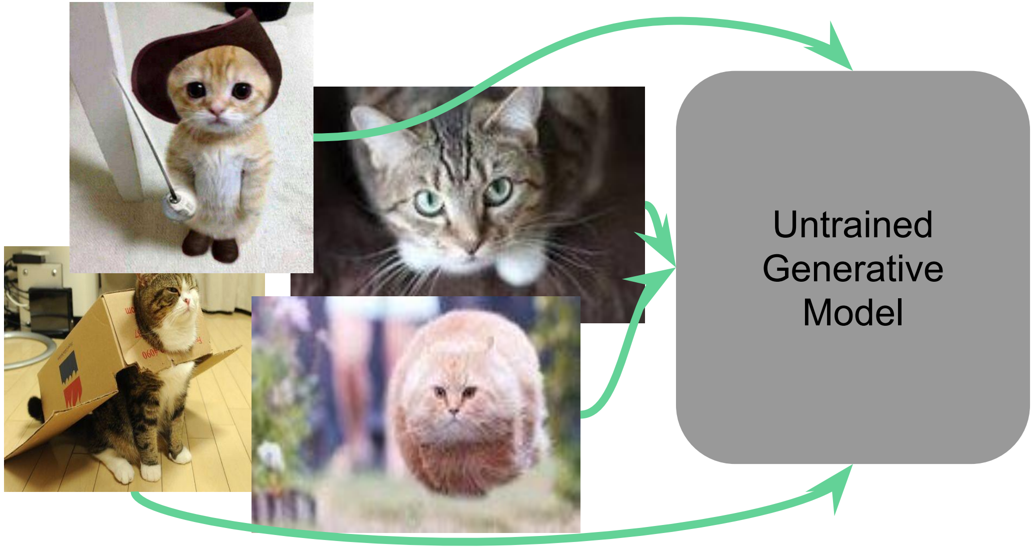
The Task: Generative Modeling

"Generative Modeling is the use of Artificial Intelligence, statistics and probability in applications to produce a representation or abstraction of observed phenomena or target variables that can be calculated from observations."



The Task: Generative Modeling

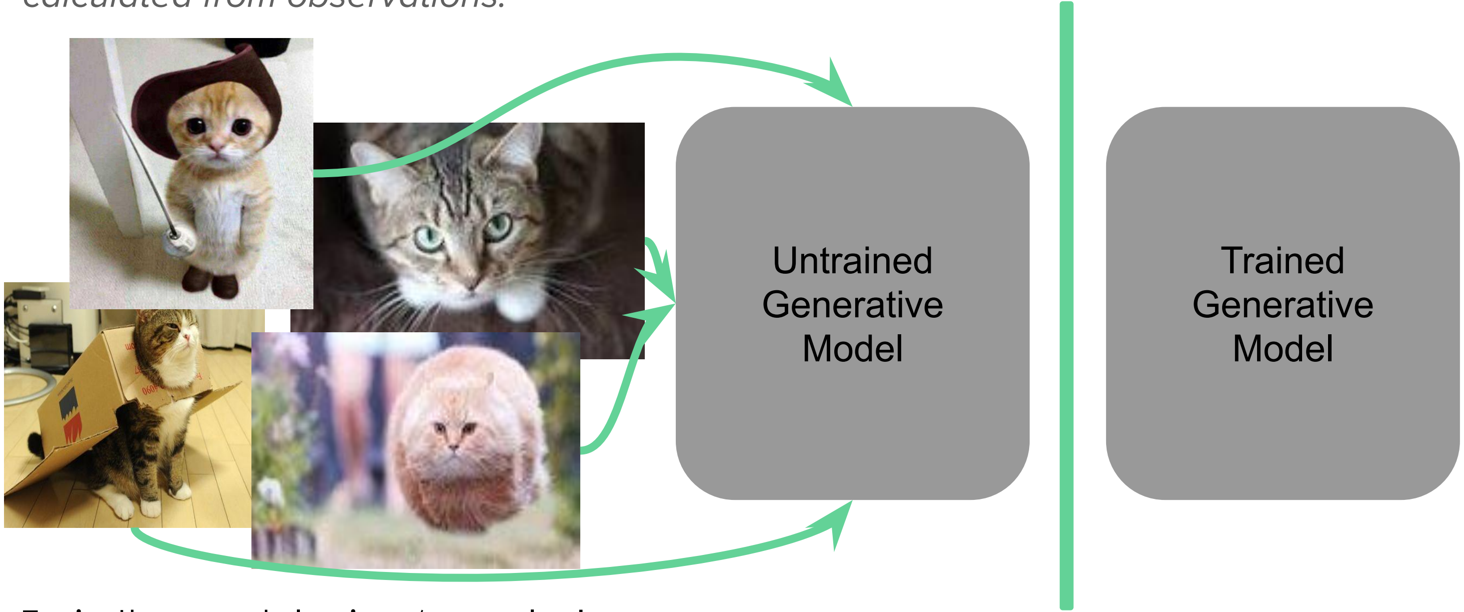
"Generative Modeling is the use of Artificial Intelligence, statistics and probability in applications to produce a representation or abstraction of observed phenomena or target variables that can be calculated from observations."



Train the model using 'samples'

The Task: Generative Modeling

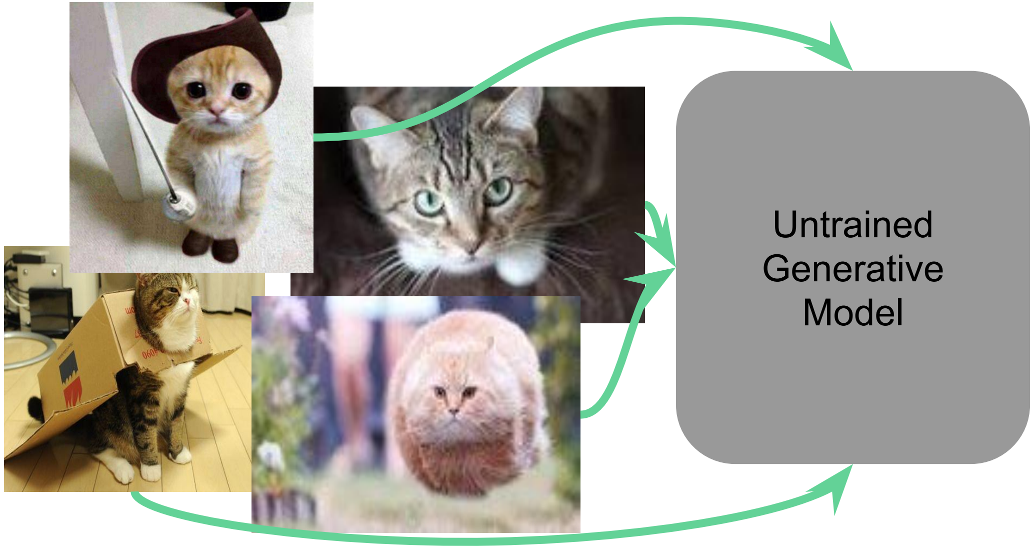
“Generative Modeling is the use of Artificial Intelligence, statistics and probability in applications to produce a representation or abstraction of observed phenomena or target variables that can be calculated from observations.”



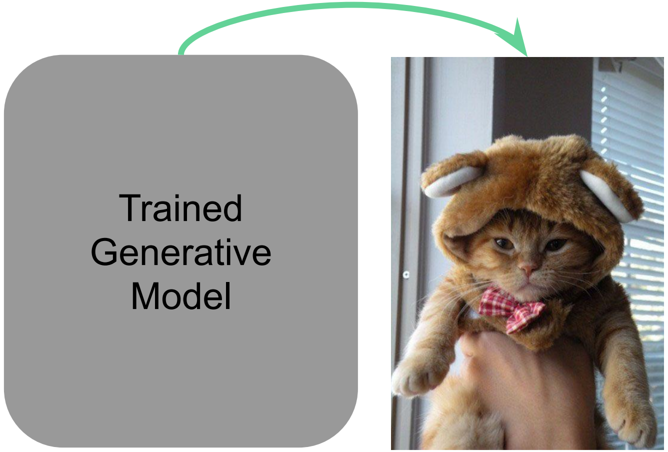
Train the model using ‘samples’

The Task: Generative Modeling

"Generative Modeling is the use of Artificial Intelligence, statistics and probability in applications to produce a representation or abstraction of observed phenomena or target variables that can be calculated from observations."



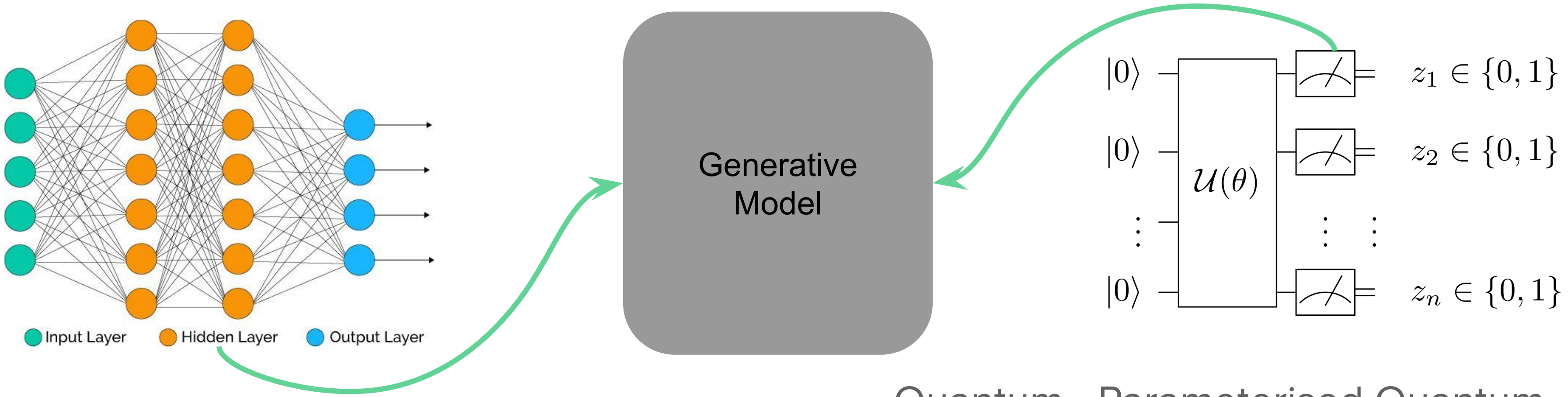
Train the model using 'samples'



Trained Model will (approximately) generate a 'new' cat image

Quantum Computers as Samplers

Classical - Neural Network or other



Quantum - Parameterised Quantum Circuit (PQC) [arXiv:1906.07682](https://arxiv.org/abs/1906.07682)

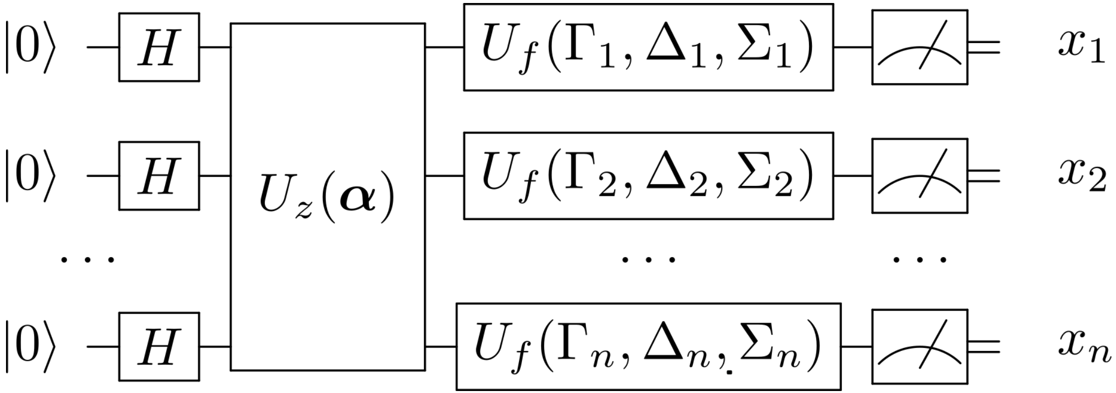
Quantum Computers as Samplers

- Quantum Inspired Training of Boltzmann Machines arXiv:1507.02642
- Quantum Boltzmann Machine Phys. Rev. X 8, 021050
  - Using annealing to prepare thermal state to sample from.
- Gate based Quantum Boltzmann Machine arXiv:1712.05304
  - Use QAOA to prepare approximate thermal state.
- Born Machine npj QI 5:45, Phys. Rev. A 98, 062324, …
  - A ‘new’ model - generates statistics directly from Born rule of Quantum Mechanics

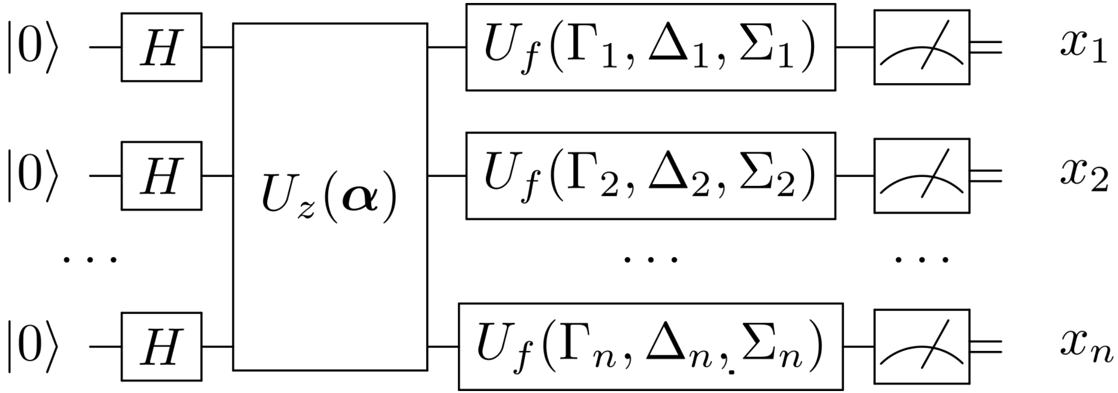
The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

Ising Born Machine



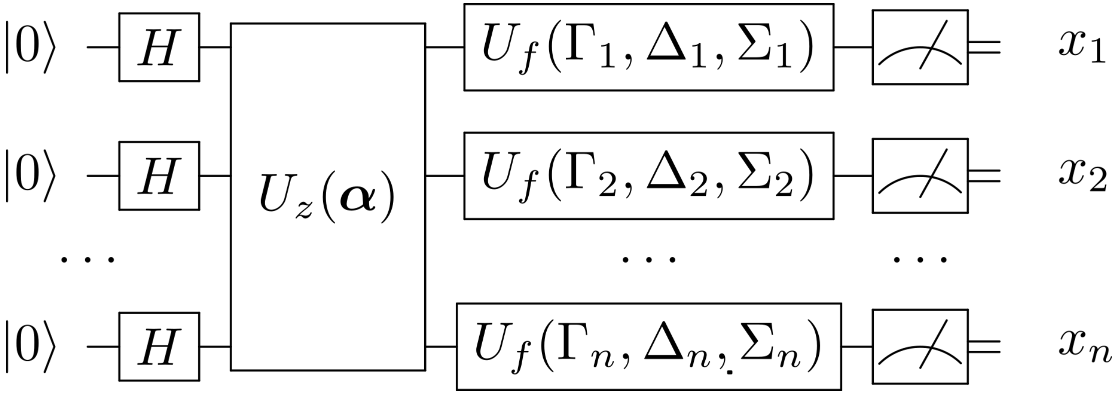
Ising Born Machine



$$\mathbf{x} = x_1 x_2 \ldots x_n \in \{0,1\}^n$$

$$\sim p_{\boldsymbol{\theta}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\boldsymbol{\theta}} \rangle|^2$$

Ising Born Machine



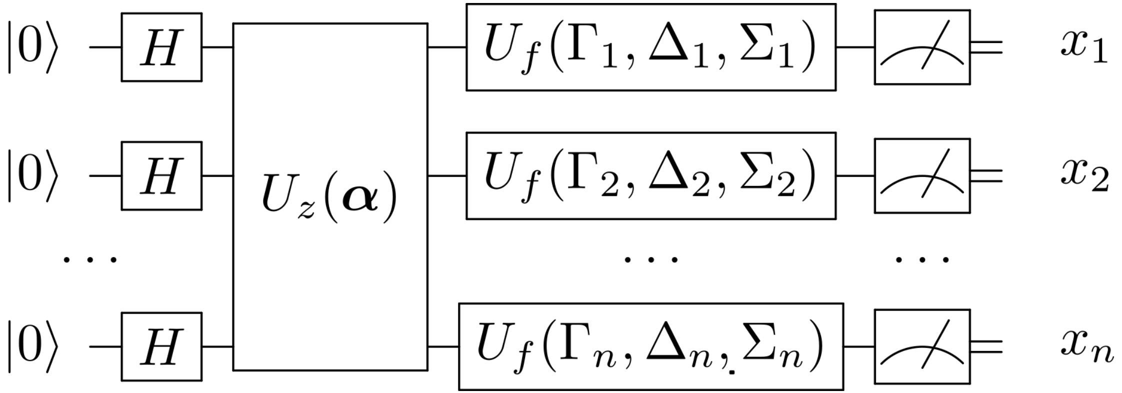
$$\mathbf{x} = x_1 x_2 \dots x_n \in \{0,1\}^n$$

$$\sim p_{\boldsymbol{\theta}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\boldsymbol{\theta}} \rangle|^2$$

$$U_z(\boldsymbol{\alpha}) := \prod_j U_z(\alpha_j, S_j) = \prod_j \exp\left(i\alpha_j \bigotimes_{k \in S_j} Z_k \right)$$

$$U_f(\boldsymbol{\Gamma}, \boldsymbol{\Delta}, \boldsymbol{\Sigma}) := \exp\left(i \sum_{k=1}^{n} \Gamma_k X_k + \Delta_k Y_k + \Sigma_k Z_k \right)$$

Ising Born Machine



$$\mathbf{x} = x_1 x_2 \ldots x_n \in \{0,1\}^n$$

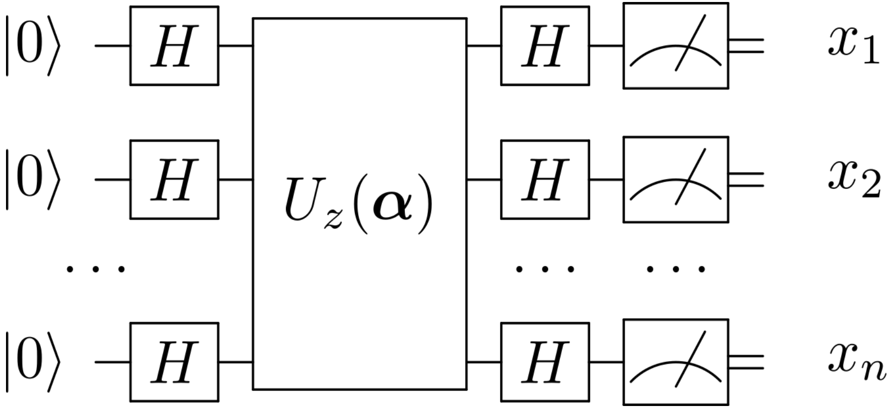
$$\sim p_{\boldsymbol{\theta}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\boldsymbol{\theta}} \rangle|^2$$

Trainable Parameters

$$U_z(\boldsymbol{\alpha}) := \prod_j U_z(\alpha_j, S_j) = \prod_j \exp\left(i\alpha_j \bigotimes_{k \in S_j} Z_k \right)$$

$$U_f(\boldsymbol{\Gamma}, \boldsymbol{\Delta}, \boldsymbol{\Sigma}) := \exp\left(i \sum_{k=1}^{n} \Gamma_k X_k + \Delta_k Y_k + \Sigma_k Z_k \right)$$

Ising Born Machine



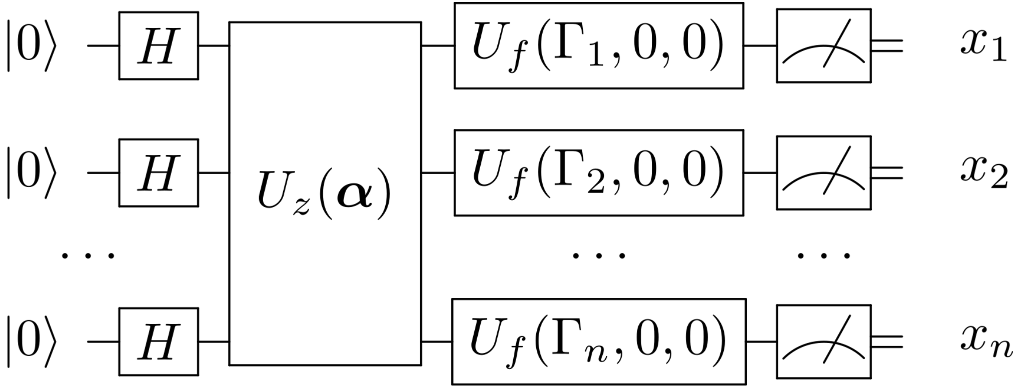
$$\mathbf{x} = x_1 x_2 \dots x_n \in \{0,1\}^n$$
$$\sim p_{\boldsymbol{\theta}}^{\text{IQP}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\boldsymbol{\theta}} \rangle|^2$$

Recover IQP (Instantaneous Quantum Polytime) circuits.

$$U_z(\boldsymbol{\alpha}) := \prod_j U_z(\alpha_j, S_j) = \prod_j \exp\left(i\alpha_j \bigotimes_{k \in S_j} Z_k\right)$$

$$U_f(\boldsymbol{\Gamma}, \boldsymbol{\Delta}, \boldsymbol{\Sigma}) := \exp\left(i \frac{1}{\sqrt{2}} \sum_{k=1}^{n} (X_k + Z_k)\right) = H^{\otimes n}$$

Ising Born Machine



$$\mathbf{x} = x_1 x_2 \dots x_n \in \{0,1\}^n$$

$$\sim p_{\boldsymbol{\theta}}^{p=1 \text{ QAOA}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\boldsymbol{\theta}} \rangle|^2$$

Recover shallowest depth (p=1) version of QAOA circuits.

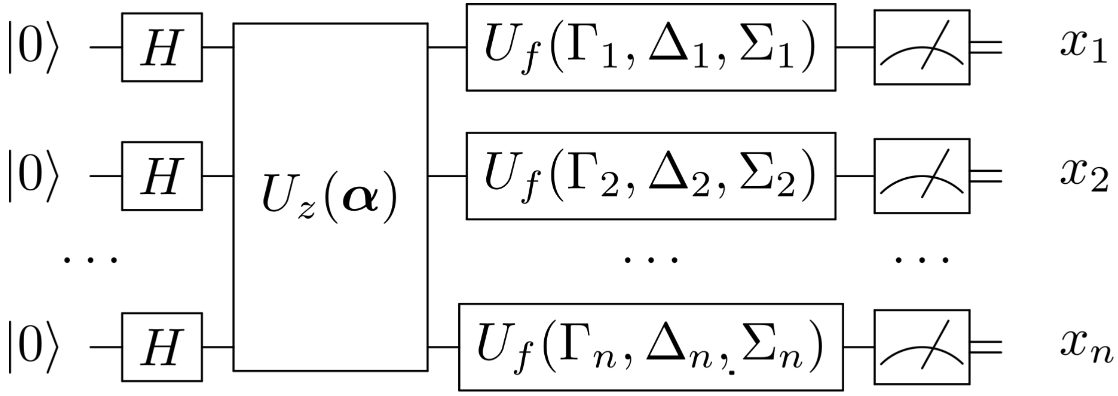
$$U_z(\boldsymbol{\alpha}) := \prod_j U_z(\alpha_j, S_j) = \prod_j \exp\left(i\alpha_j \bigotimes_{k \in S_j} Z_k \right)$$

$$U_f(\boldsymbol{\Gamma}, \mathbf{0}, \mathbf{0}) := \exp\left(i \sum_{k=1}^{n} \Gamma_k X_k \right)$$

The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

Gradient Based Training



$$\mathbf{x} = x_1 x_2 \ldots x_n \in \{0,1\}^n$$
$$\sim p_{\boldsymbol{\theta}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\boldsymbol{\theta}} \rangle|^2$$

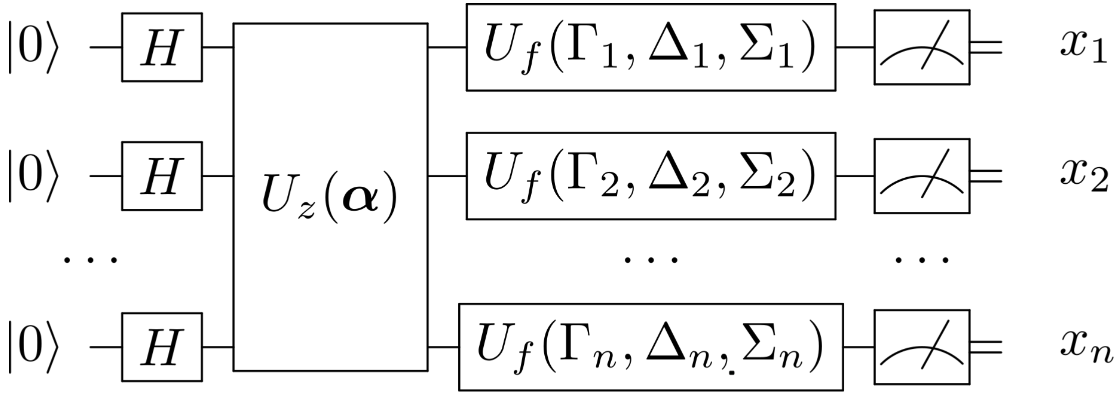
1 - Sample from model.

Gradient Based Training

2 - Evaluate loss & gradient

$$\mathcal{L}_B(p_{\boldsymbol{\theta}}(\mathbf{x}), \pi(\mathbf{y}))$$

$$\partial_{\boldsymbol{\theta}} \mathcal{L}_B$$



$$\mathbf{x} = x_1 x_2 \ldots x_n \in \{0,1\}^n$$

$$\sim p_{\boldsymbol{\theta}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\boldsymbol{\theta}} \rangle|^2$$

1 - Sample from model.

Gradient Based Training

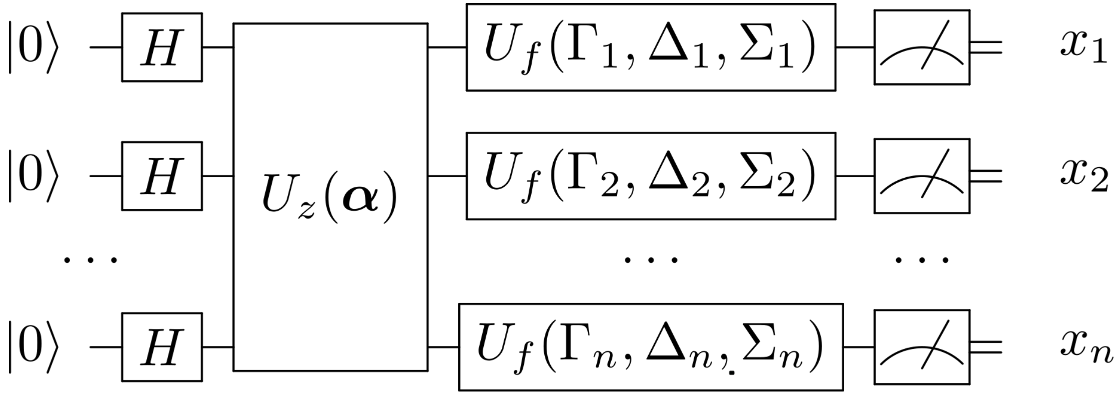
3 - Update parameters by gradient based descent

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \partial_{\boldsymbol{\theta}} \mathcal{L}_B$$

2 - Evaluate loss & gradient

$$\mathcal{L}_B(p_{\boldsymbol{\theta}}(\mathbf{x}), \pi(\mathbf{y}))$$

$$\partial_{\boldsymbol{\theta}} \mathcal{L}_B$$



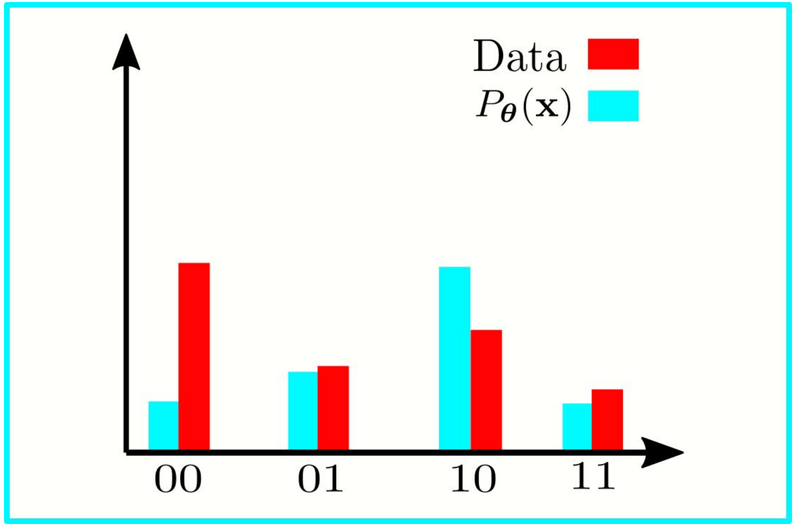
$$\mathbf{x} = x_1 x_2 \dots x_n \in \{0,1\}^n$$

$$\sim p_{\boldsymbol{\theta}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\boldsymbol{\theta}} \rangle|^2$$

1 - Sample from model.

Choosing a Cost Function

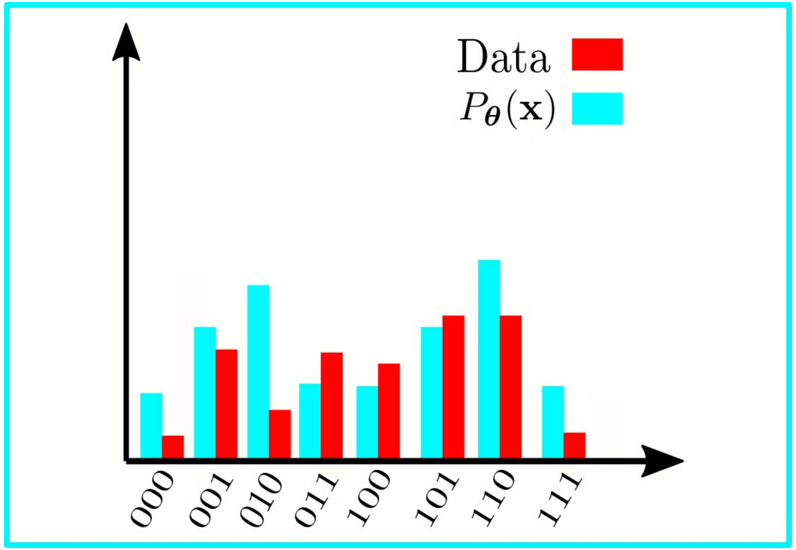
Computing the loss function is a means of checking how well we are doing - comparing the data and the instantaneous model distributions.



How do we compare two probability distributions? - This is hard.

Choosing a Cost Function

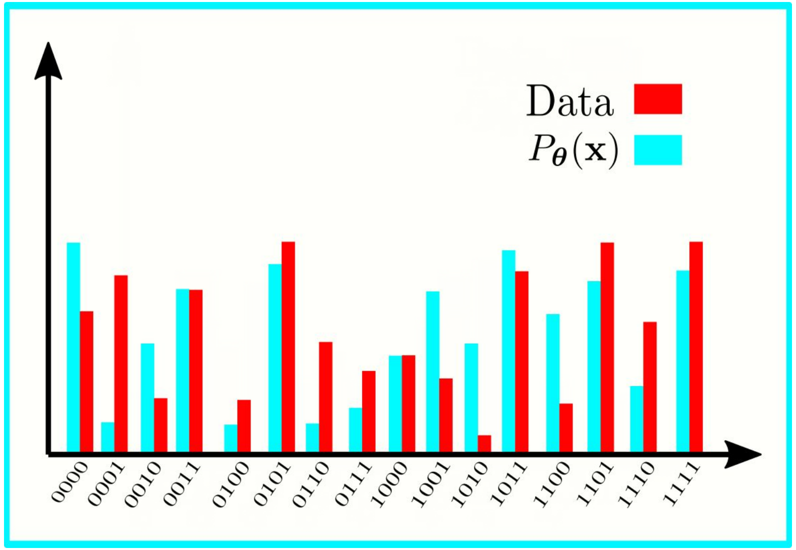
Computing the loss function is a means of checking how well we are doing - comparing the data and the instantaneous model distributions.



How do we compare two probability distributions? - This is hard.

Choosing a Cost Function

Computing the loss function is a means of checking how well we are doing - comparing the data and the instantaneous model distributions.



How do we compare two probability distributions? - This is hard.

Choosing a Cost Function

We need a loss function which is ideally:

- Easily computable (in terms of sample + computational complexity)
- Relatively ‘powerful’ (should be sensitive to differences in the distributions)

The Benchmark – Total Variation Distance

$$\mathrm{TV}(p_{\theta}, \pi) = \frac{1}{2}\sum_{\mathbf{z}} |\overset{\text{Model}}{p_{\theta}}(\mathbf{z}) - \overset{\text{Data}}{\pi}(\mathbf{z})|$$

Why? It's the notion that typically goes with quantum supremacy experiments:

- IQP: Phys. Rev. Lett. 117, 080501- Assume a conjecture about the hardness of computing the Ising partition function. If it is possible to classically sample from the output probability distribution of any IQP circuit C in polynomial time, up to an error of 1/384 in TV, then there is a BPP^NP algorithm to solve any problem in P^#P . Hence the Polynomial Hierarchy would collapse to its third level.

The Benchmark - Total Variation Distance

Model

Data

$$\mathrm{TV}(p_{\theta}, \pi) = \frac{1}{2} \sum_{\mathbf{z}} |p_{\theta}(\mathbf{z}) - \pi(\mathbf{z})|$$

Powerful

Why? It's the notion that typically goes with quantum supremacy experiments:

- IQP: Phys. Rev. Lett. 117, 080501- Assume a conjecture about the hardness of computing the Ising partition function. If it is possible to classically sample from the output probability distribution of any IQP circuit C in polynomial time, up to an error of 1/384 in TV, then there is a BPP^NP algorithm to solve any problem in P^#P . Hence the Polynomial Hierarchy would collapse to its third level.

The Benchmark - Total Variation Distance

Model

Data

$$\mathrm{TV}(p_{\theta}, \pi) = \frac{1}{2} \sum_{\mathbf{z}} |p_{\theta}(\mathbf{z}) - \pi(\mathbf{z})|$$

Powerful

Hard to compute

Why? It's the notion that typically goes with quantum supremacy experiments:

- IQP: Phys. Rev. Lett. 117, 080501- Assume a conjecture about the hardness of computing the Ising partition function. If it is possible to classically sample from the output probability distribution of any IQP circuit C in polynomial time, up to an error of 1/384 in TV, then there is a BPP^NP algorithm to solve any problem in P^#P . Hence the Polynomial Hierarchy would collapse to its third level.

Previous Training – Maximum Mean Discrepancy

Liu & Wang: Phys. Rev. A 98, 062324, Gretton et.al.: JMLR 13 (2012) 723-773

$$\mathcal{L}_{\text{MMD}}(p_\theta, \pi) = \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim p_\theta \\ \mathbf{y}\sim p_\theta}} (\kappa(\mathbf{x}, \mathbf{y})) + \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim \pi \\ \mathbf{y}\sim \pi}} (\kappa(\mathbf{x}, \mathbf{y})) - 2\mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim p_\theta \\ \mathbf{y}\sim \pi}} (\kappa(\mathbf{x}, \mathbf{y}))$$

Previous Training – Maximum Mean Discrepancy

The MMD is very efficient to compute. It has quadratic sample complexity independent of the size of the underlying space arXiv:0901.2698 :

$$\left| \sqrt{\mathcal{L}_{\text{MMD}}} - \sqrt{\hat{\mathcal{L}}_{\text{MMD}}} \right| = \mathcal{O}\left(\frac{1}{\sqrt{M}}\right)$$

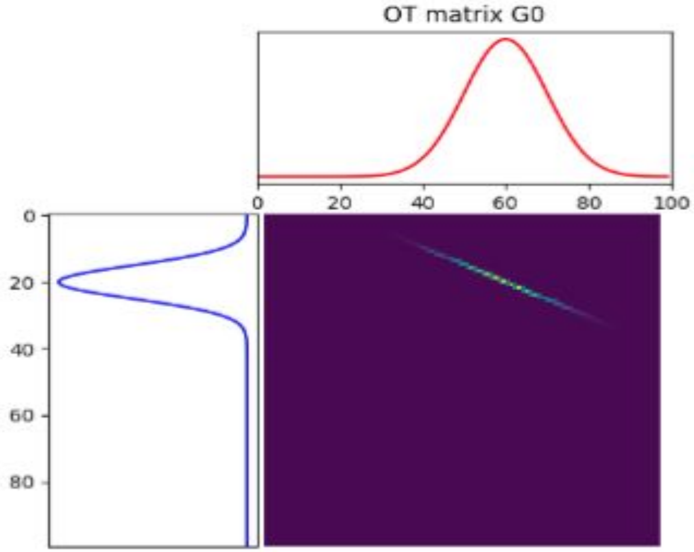
But, it lower bounds Total Variation: On Choosing and Bounding Probability Metrics

$$TV(p,q) \geq \sqrt{\mathcal{L}_{\text{MMD}}(p,q)}$$

So, minimising MMD, does not necessarily do a good job of minimising TV. Can we minimise an efficient upper bound instead?

Wasserstein + Optimal Transport?

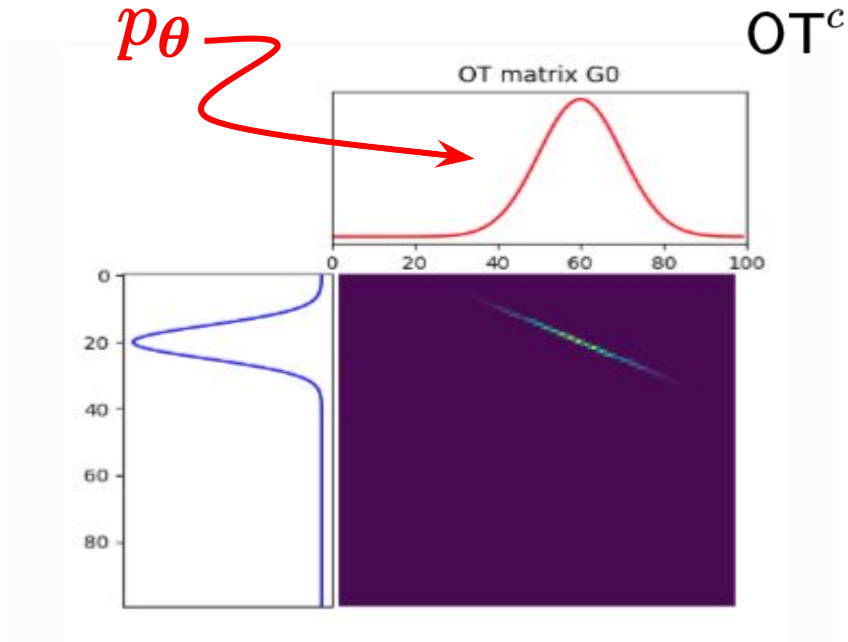
The Wasserstein metric is related to Optimal Transport (Villani, 2009: Optimal transport, old and new.) - a way to compare distributions by determining how to 'transport' one into the other



$$\mathrm{OT}^c(p_{\boldsymbol{\theta}}, \pi) = \min_{U \in \mathcal{U}(p_{\boldsymbol{\theta}}, \pi)} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}} c(\mathbf{x}, \mathbf{y}) U(\mathbf{x}, \mathbf{y})$$

Wasserstein + Optimal Transport?

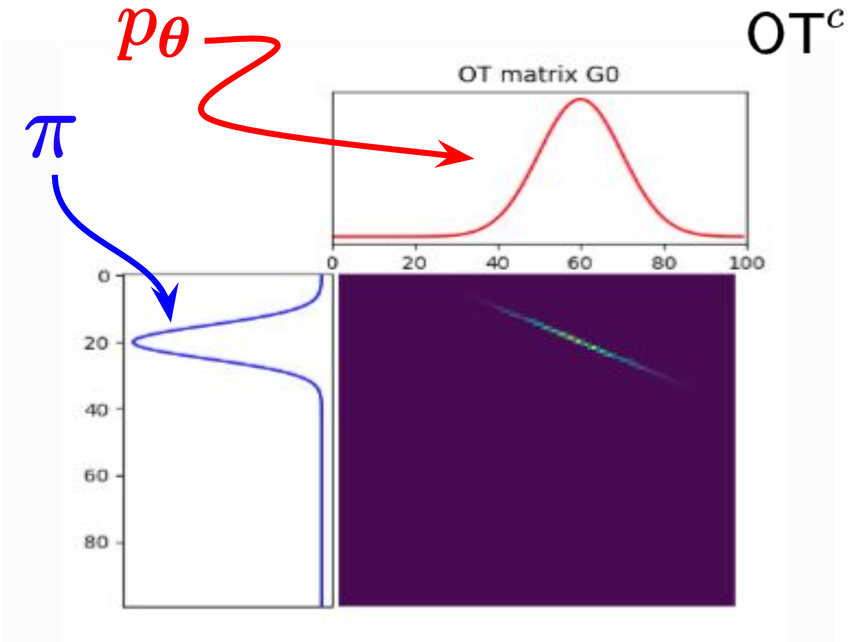
The Wasserstein metric is related to Optimal Transport ([Villani, 2009: Optimal transport, old and new.](#)) - a way to compare distributions by determining how to 'transport' one into the other



$$\mathrm{OT}^c(\boxed{p_\theta}, \pi) = \min_{U \in \mathcal{U}(p_\theta, \pi)} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}} c(\mathbf{x}, \mathbf{y}) U(\mathbf{x}, \mathbf{y})$$

Wasserstein + Optimal Transport?

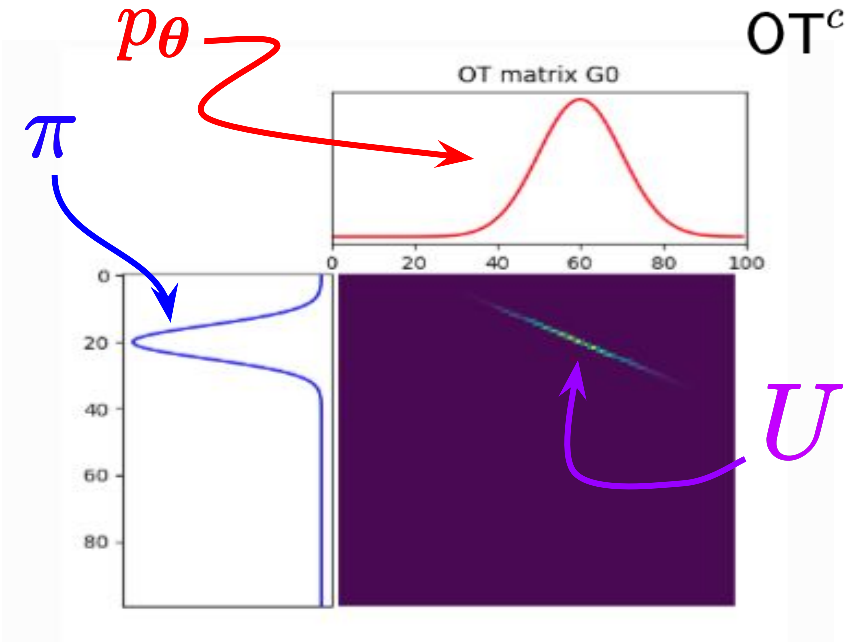
The Wasserstein metric is related to Optimal Transport (Villani, 2009: Optimal transport, old and new.) - a way to compare distributions by determining how to 'transport' one into the other



$$\mathrm{OT}^c\left(\boxed{p_\theta}, \boxed{\pi}\right) = \min_{U \in \mathcal{U}(p_\theta, \pi)} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}} c(\mathbf{x}, \mathbf{y}) U(\mathbf{x}, \mathbf{y})$$

Wasserstein + Optimal Transport?

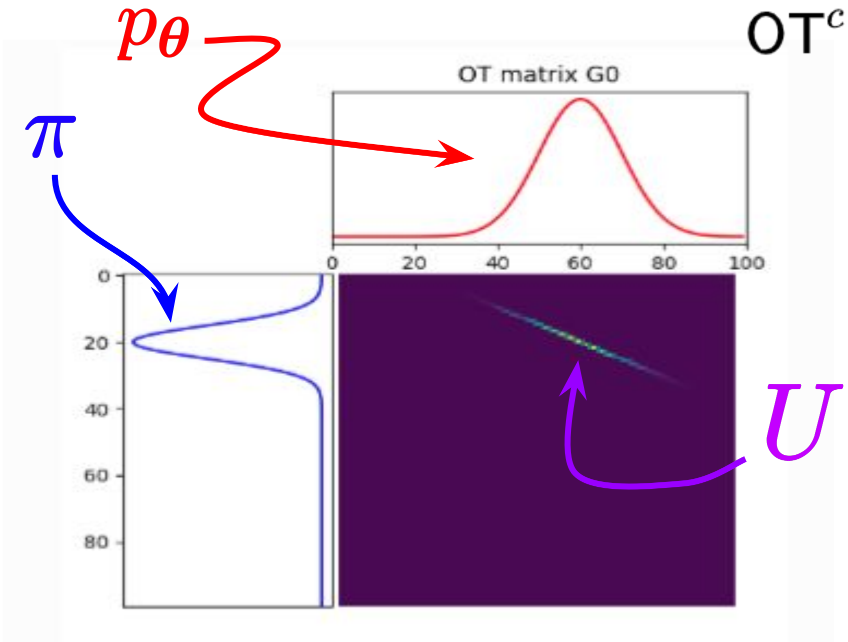
The Wasserstein metric is related to Optimal Transport ([Villani, 2009: Optimal transport, old and new.](#)) - a way to compare distributions by determining how to ‘transport’ one into the other



$$\mathrm{OT}^c(\boxed{p_\theta}, \boxed{\pi}) = \min_{\boxed{U}\in\mathcal{U}(p_\theta,\pi)} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{X}\times\mathcal{Y}} c(\mathbf{x},\mathbf{y})\boxed{U}(\mathbf{x},\mathbf{y})$$

Wasserstein + Optimal Transport?

The Wasserstein metric is related to Optimal Transport ([Villani, 2009: Optimal transport, old and new.](#)) - a way to compare distributions by determining how to 'transport' one into the other



$$\mathrm{OT}^c(\boxed{p_\theta},\boxed{\pi}) = \min_{\boxed{U}\in\mathcal{U}(p_\theta,\pi)} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{X}\times\mathcal{Y}} c(\mathbf{x},\mathbf{y})\boxed{U}(\mathbf{x},\mathbf{y})$$

OT 'cost' - Distance between points in p, q. - Take to be Hamming distance between bin strings.

$$c(\mathbf{x},\mathbf{y}) = d_H(\mathbf{x},\mathbf{y}) := |\mathbf{x}-\mathbf{y}|$$

Computation + Power

OT is hard to compute though arXiv:0901.2698...

$$|\mathsf{OT} - \widehat{\mathsf{OT}}| = \mathcal{O}\left(\frac{1}{M^{1/n}}\right)$$

But it does upper bound TV: On Choosing and Bounding Probability Metrics

$$\mathsf{TV}(p, q) \leq \mathsf{OT}^d(p, q)$$

Computation + Power

OT is hard to compute though arXiv:0901.2698...

$$|\mathrm{OT} - \widehat{\mathrm{OT}}| = \mathcal{O}\left(\frac{1}{M^{1/n}}\right)$$

But it does upper bound TV: On Choosing and Bounding Probability Metrics

$$\mathrm{TV}(p,q) \leq \mathrm{OT}^d(p,q)$$

$$|\sqrt{\mathcal{L}_{\mathrm{MMD}}} - \sqrt{\hat{\mathcal{L}}_{\mathrm{MMD}}}| = \mathcal{O}\left(\frac{1}{\sqrt{M}}\right)$$

$$TV(p,q) \geq \sqrt{\mathcal{L}_{\mathrm{MMD}}(p,q)}$$

Sinkhorn Divergence

Let's add a regularisation term to the optimal transport distance… While we're at it, let's also add symmetric terms to remove bias… => **Sinkhorn Divergence** Entropy 2017, 19(2), 47, arXiv:1810.08278, arXiv:1706.00292

$$\mathrm{OT}^c_\epsilon(p_\theta, \pi) := \min_{U \in \mathcal{U}(p_\theta, \pi)} \left(\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{X}^d \times \mathcal{Y}^d} c(\mathbf{x}, \mathbf{y}) U(\mathbf{x}, \mathbf{y}) + \epsilon \mathrm{KL}(U | p_\theta \otimes \pi) \right)$$

$$\mathcal{L}^\epsilon_{\mathrm{SHD}}(p_\theta, \pi) := \mathrm{OT}^c_\epsilon(p_\theta, \pi) - \frac{1}{2}\mathrm{OT}^c_\epsilon(p_\theta, p_\theta) - \frac{1}{2}\mathrm{OT}^c_\epsilon(\pi, \pi)$$

Regularised by the entropy term (KL divergence). Sinkhorn divergence interpolates between MMD and unregularised optimal transport (as a function of regulariser, ϵ) - gradient has same form as MMD.

Sinkhorn Divergence

Can be efficient arXiv:1810.02733

$$\mathbb{E}\left|\mathcal{L}^{\mathcal{O}(n^2)}_{\mathsf{SHD}} - \hat{\mathcal{L}}^{\mathcal{O}(n^2)}_{\mathsf{SHD}}\right| = \mathcal{O}\left(\frac{1}{\sqrt{M}}\right)$$

$$\left|\mathcal{L}^{\mathcal{O}(n^2)}_{\mathsf{SHD}} - \hat{\mathcal{L}}^{\mathcal{O}(n^2)}_{\mathsf{SHD}}\right| = \mathcal{O}\left(\frac{n}{\sqrt{M}}\log(1/\delta)^{1/2}\right)$$

(With prob. 1-δ)

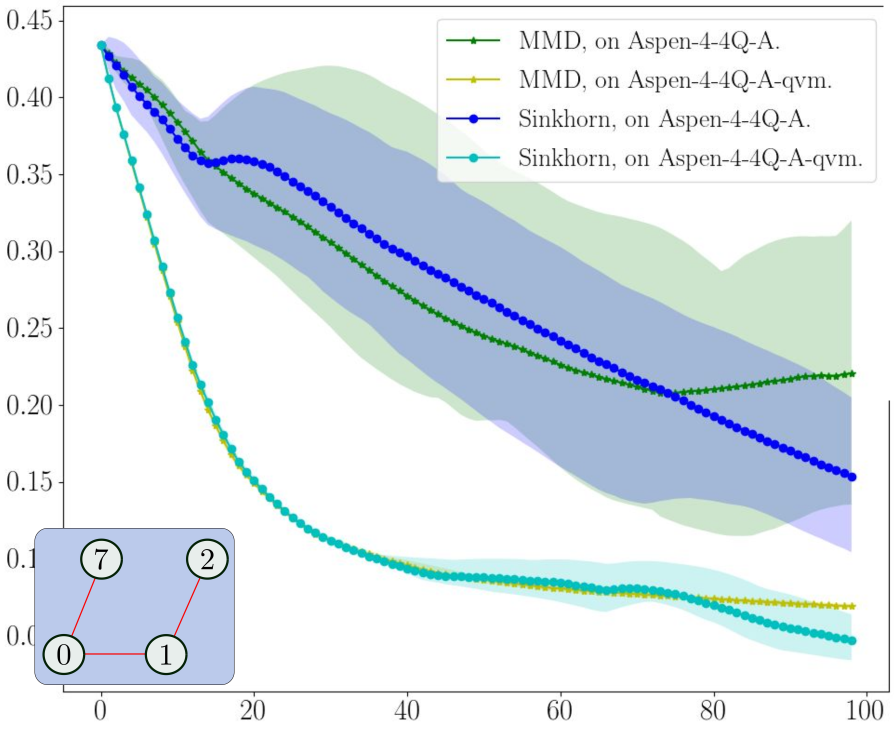
Can also be powerful:

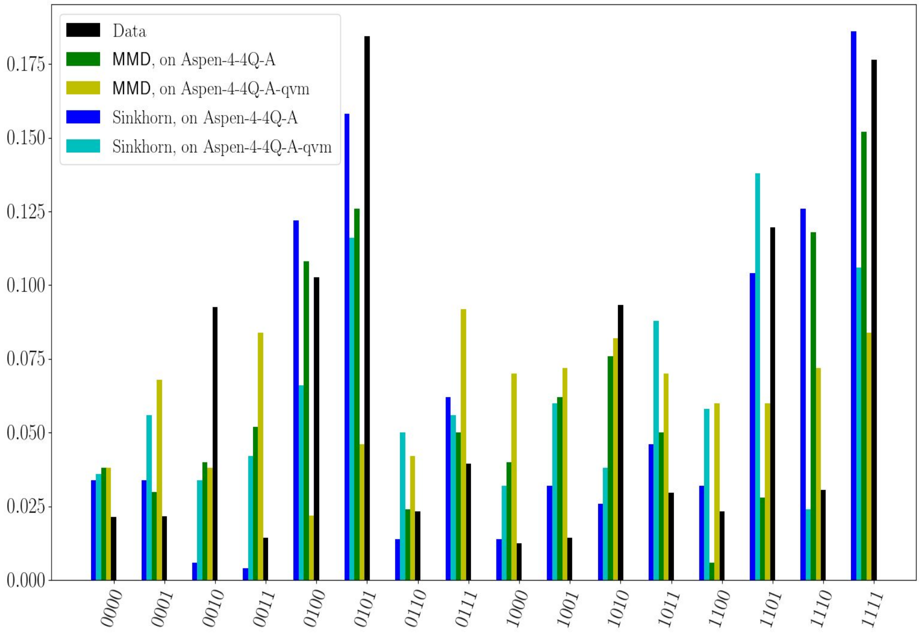
$$\mathsf{TV}(p_{\boldsymbol{\theta}}, \pi) \leq \mathsf{OT}^{d_H}_{0}(p_{\boldsymbol{\theta}}, \pi) \leq \mathsf{OT}^{d_H}_{\epsilon \leq ne^2}$$

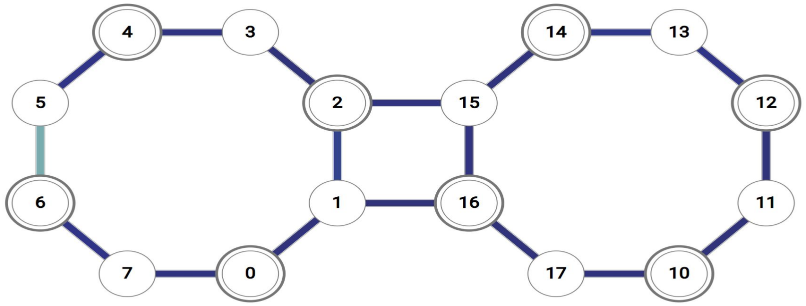
Un-regularised Optimal transport

Regularised Optimal transport

Sinkhorn vs MMD







The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine

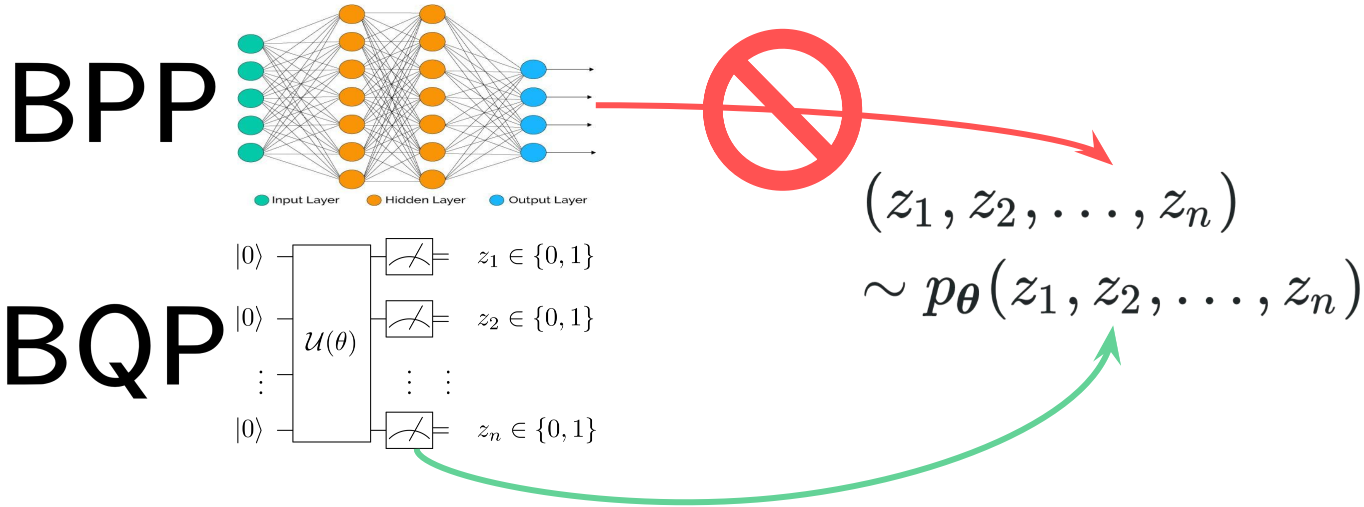
The Born Supremacy:

Quantum Advantage and Training of an Ising Born Machine

Quantum Advantage of Training

“Quantum supremacy is the potential ability of quantum computing devices to solve problems that classical computers practically cannot.” John Preskill

The simplest example of such a problem is a sampling problem (or at least that we have evidence for)



Quantum Advantage of Training

Part 1 - Hardness of Simulation:

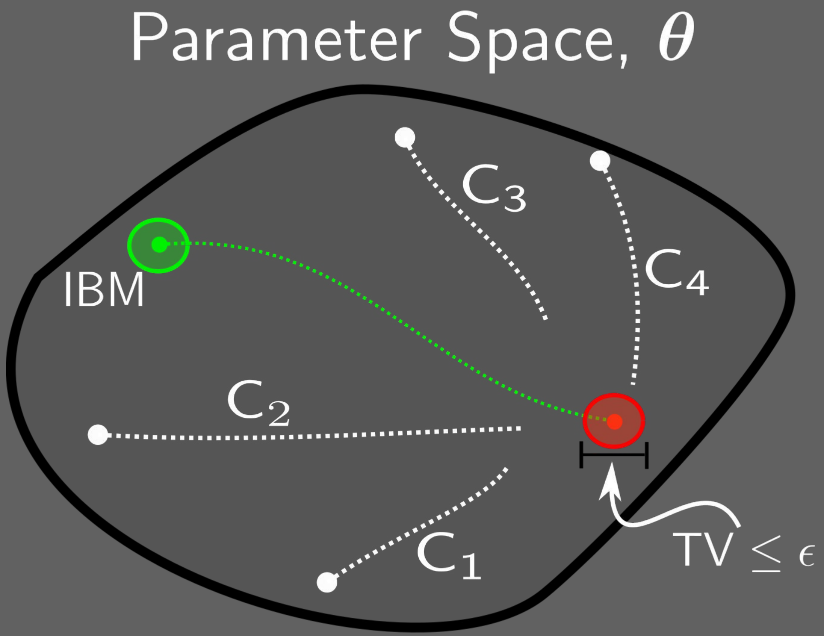
- Hardness of simulating the IBM (Ising Born machine) can be retained through training by enforcing parameter updates in a particular way.

BUT: This doesn't say the model is able to actually outperform all classical algorithms in a learning task - Hardness of simulation does not imply usefulness!

Quantum Advantage of Training

Part 2 - Advantage in learning (??):

- Can we find examples which can be reached by quantum models, but cannot by any classical models?



The Order of Definitions

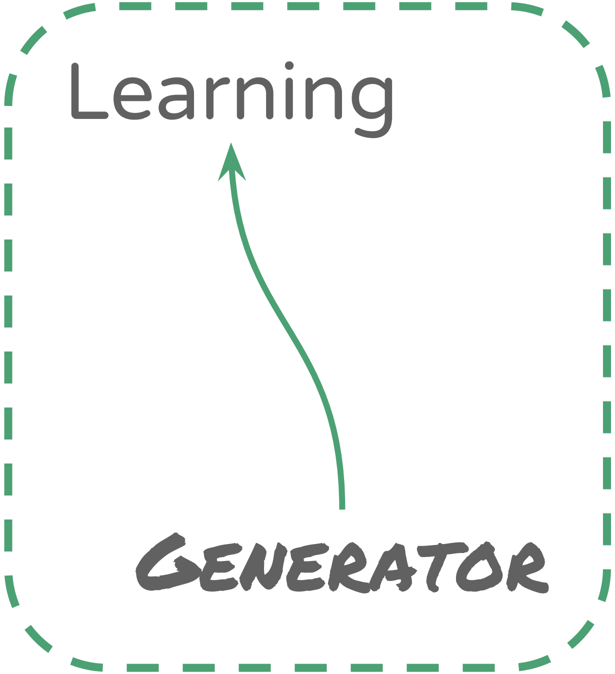
Supremacy

The Order of Definitions

Learning

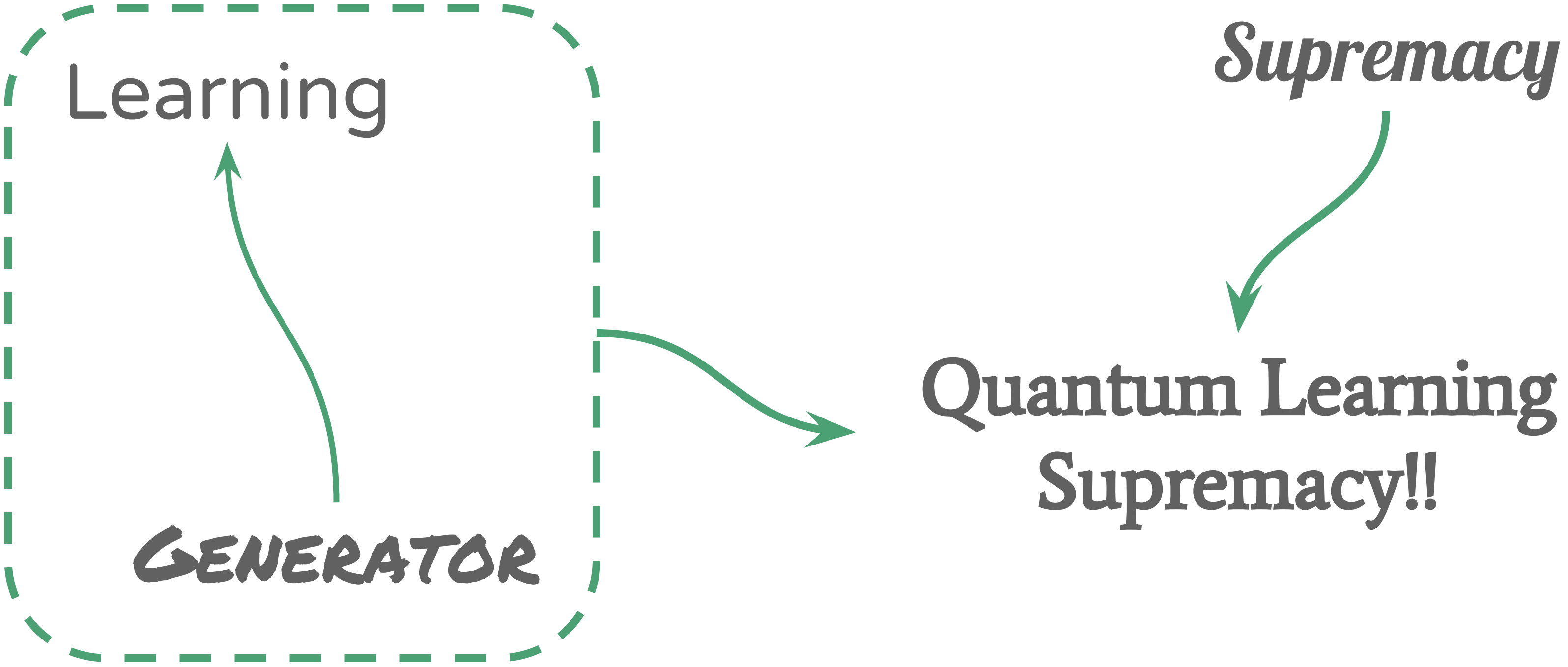
Supremacy

The Order of Definitions

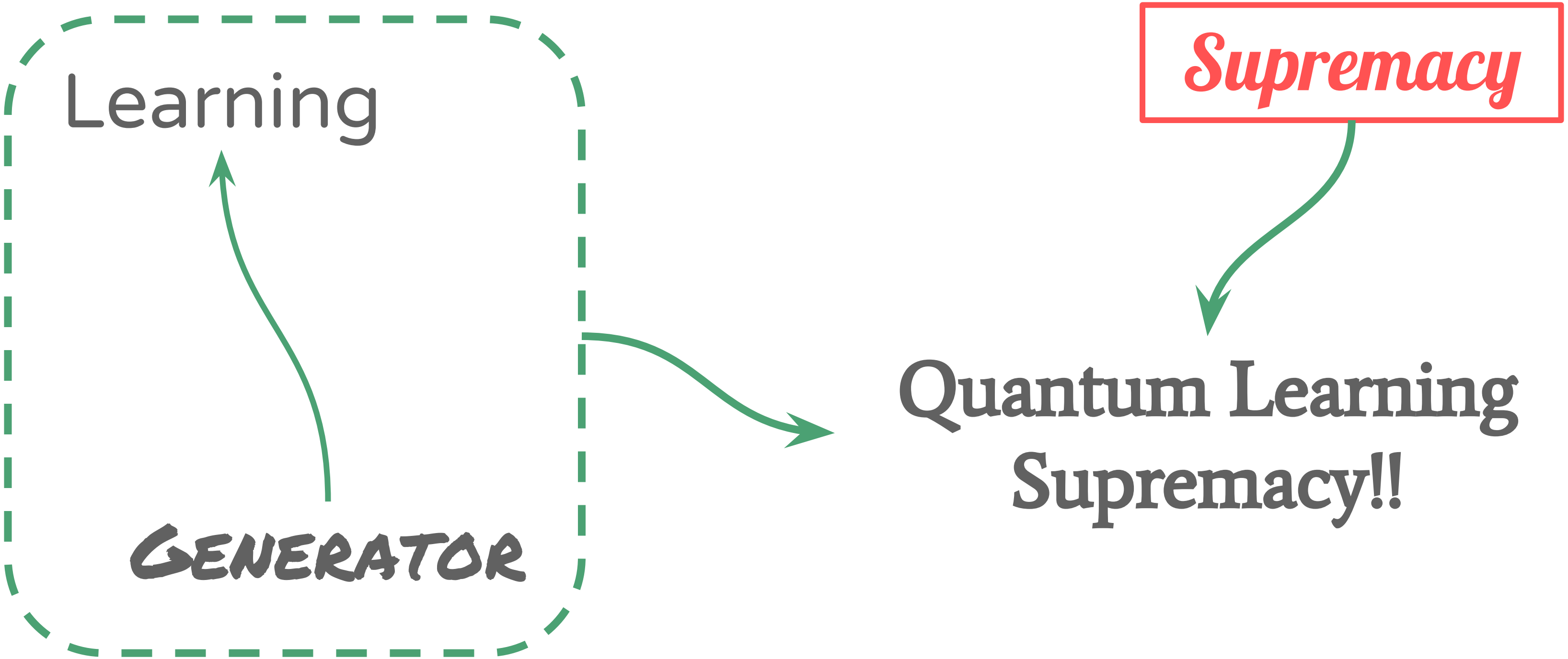


Supremacy

The Order of Definitions



The Order of Definitions

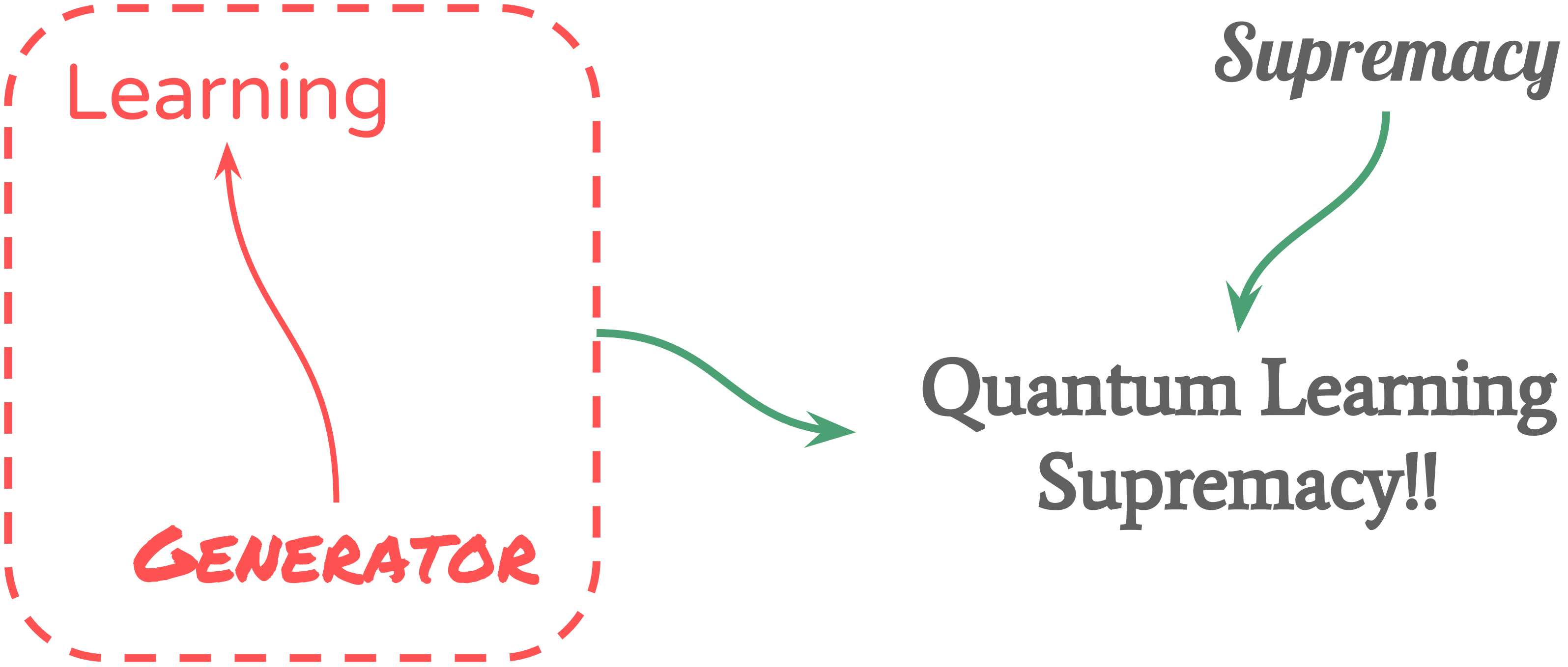


Supremacy

Supremacy null hypothesis:

The output of this computation was arrived at by a classical computer

The Order of Definitions



Generator

$$D \in \mathcal{D}$$

Collection of probability distributions

$$bits \in \{0,1\}^{r(n)} \longrightarrow \boxed{GEN_D} \longrightarrow x \sim D$$

Generator

Samples from
distribution in collection

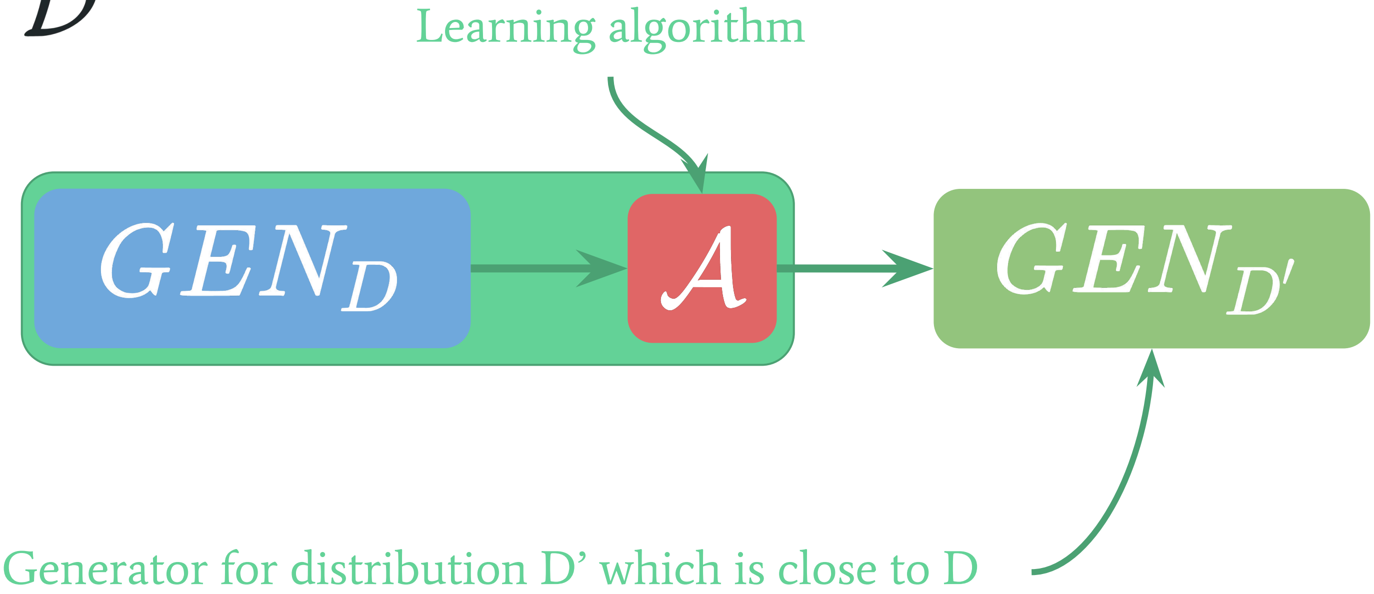
Generator

An ‘Approximate Generator’, adapted from Kearns '94 - On the learnability of discrete distributions:

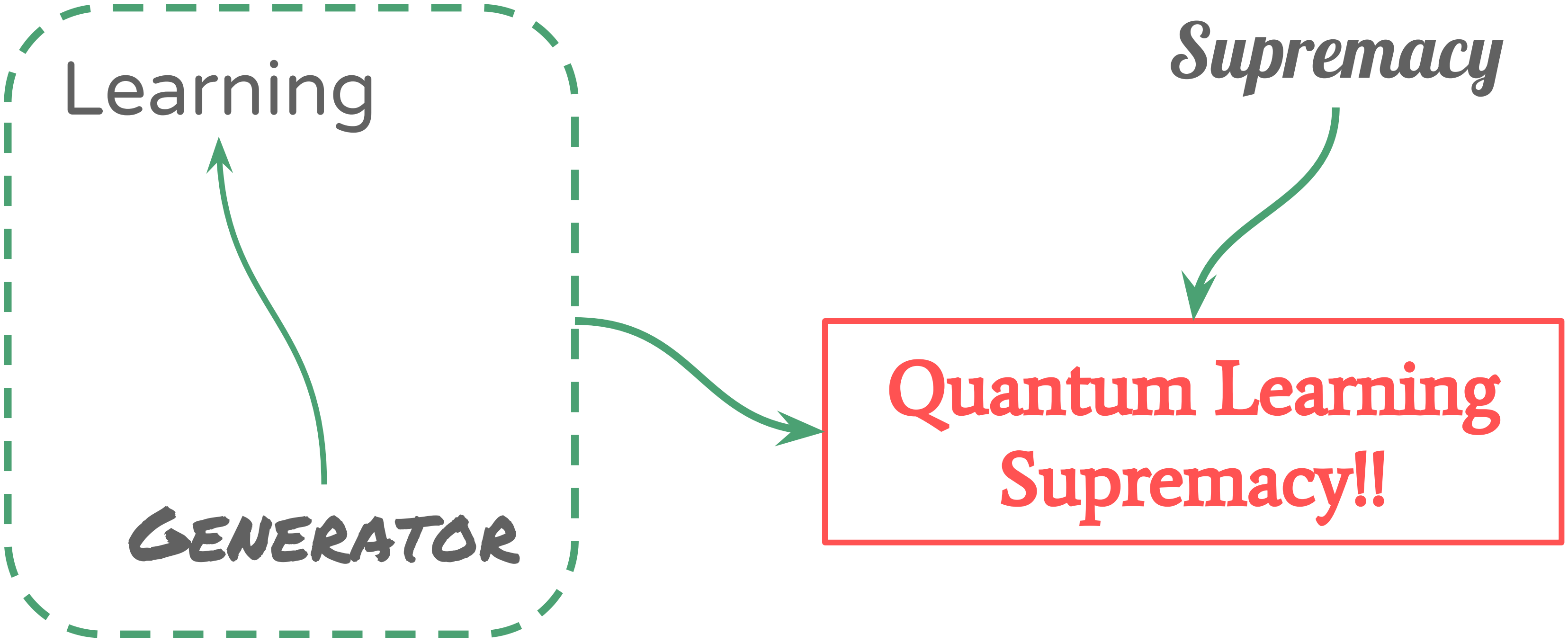
$$GEN_{D'} \longrightarrow \mathbf{z} \sim D',\, d(D, D') \leq \epsilon$$

Learning

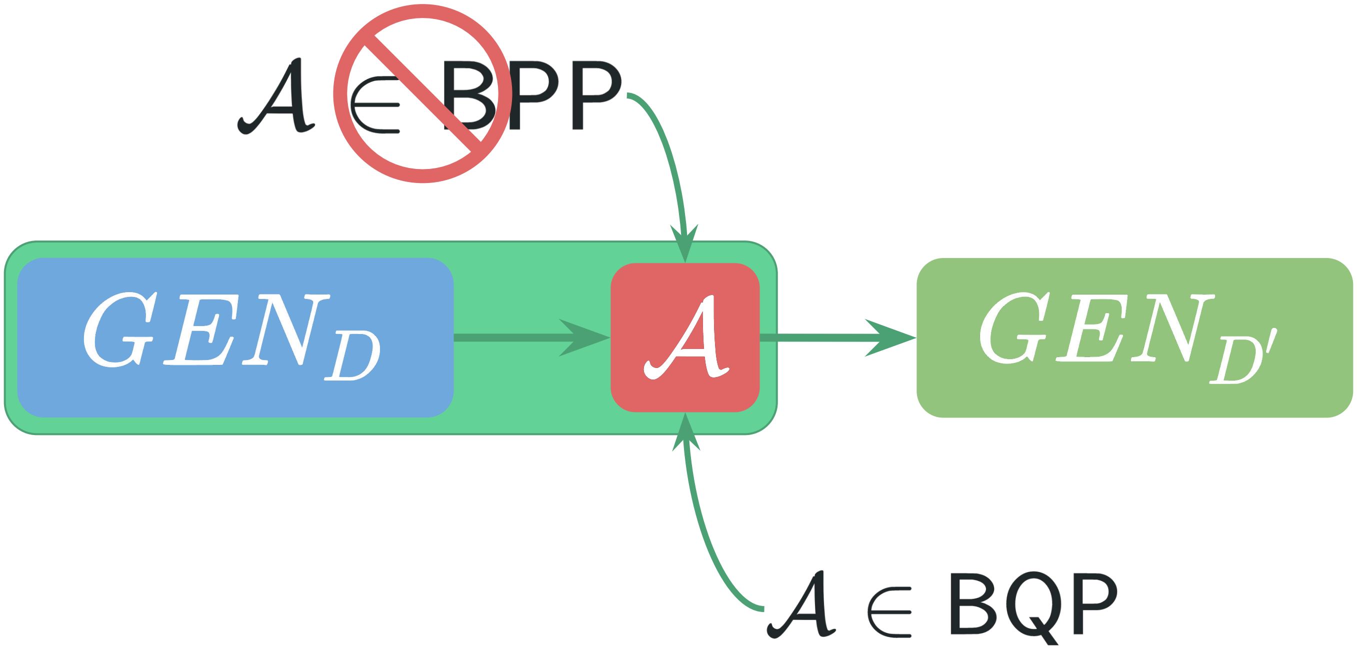
$$D \in \mathcal{D}$$



The Order of Definitions



Quantum Learning Supremacy



Conclusions

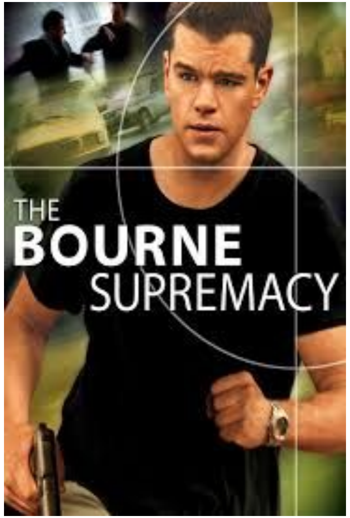
- We defined the Ising Born Machine.
- We used new gradient training methods - Sinkhorn Divergence - which is ‘stronger’ than the MMD, but efficiently computable.
- Quantum Advantage - By connecting to IQP and QAOA, the model is hard to sample from and can remain hard during training. We defined a framework for a provable advantage for generative modelling, potentially in the near term.

Conclusions

The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine: arXiv:1904.02214

Conclusions

The Born Supremacy: Quantum Advantage and Training of an Ising Born Machine: arXiv:1904.02214



Additional Plot