Quantum Machine Learning

Elizabeth Behrman

Wichita State University

behrman@math.wichita.edu

Thanks to: James Steck, Nam Nguyen, Nathan Thompson, Bill Ingle, Henry Elliott

Quantum Information: the New Frontier?

- Exponential increases in processing power
- Possibility of computing “impossible” things

For several decades, macroscopic quantum computers have been “ten years away”

Hardware problems of

- scaling
- noise
- decoherence

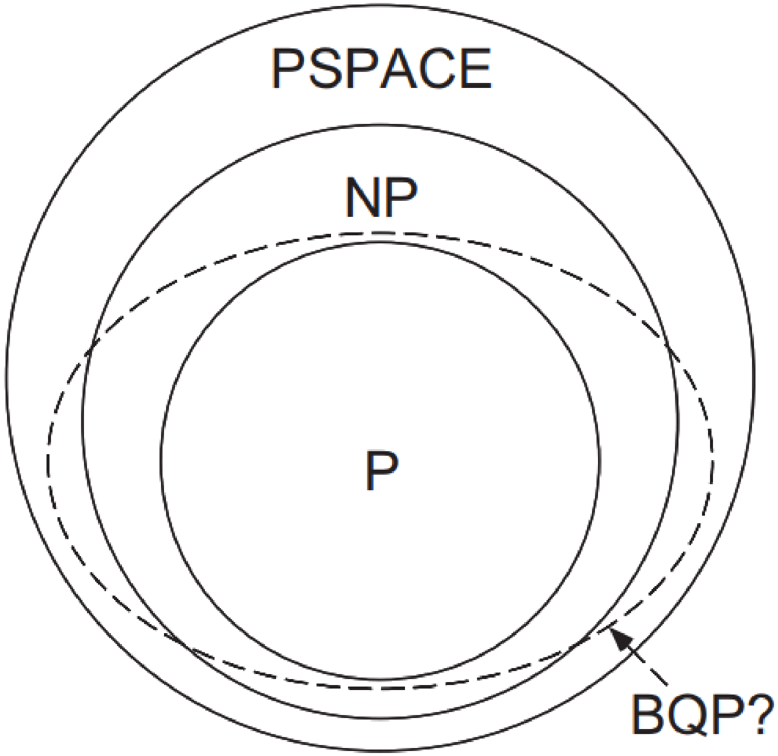
Software problems such as error correction

- Algorithm construction - not easy even for simple problems!

Standard Approaches

- “building block” strategy: a procedure is formulated as a sequence of steps (quantum gates)’ or alternatively
- Analog computing strategy in which the ground state of a physical system is the answer to a binary optimization problem

The Power of quantum computing?



Quantum machine learning

- Generate truly quantum “algorithms”
- Discover quantum advantage
- Resilience to disturbances/errors in models
- Resilience to decoherence
- Resilience to incomplete/damaged data
- Automatic scaleup
- Universality

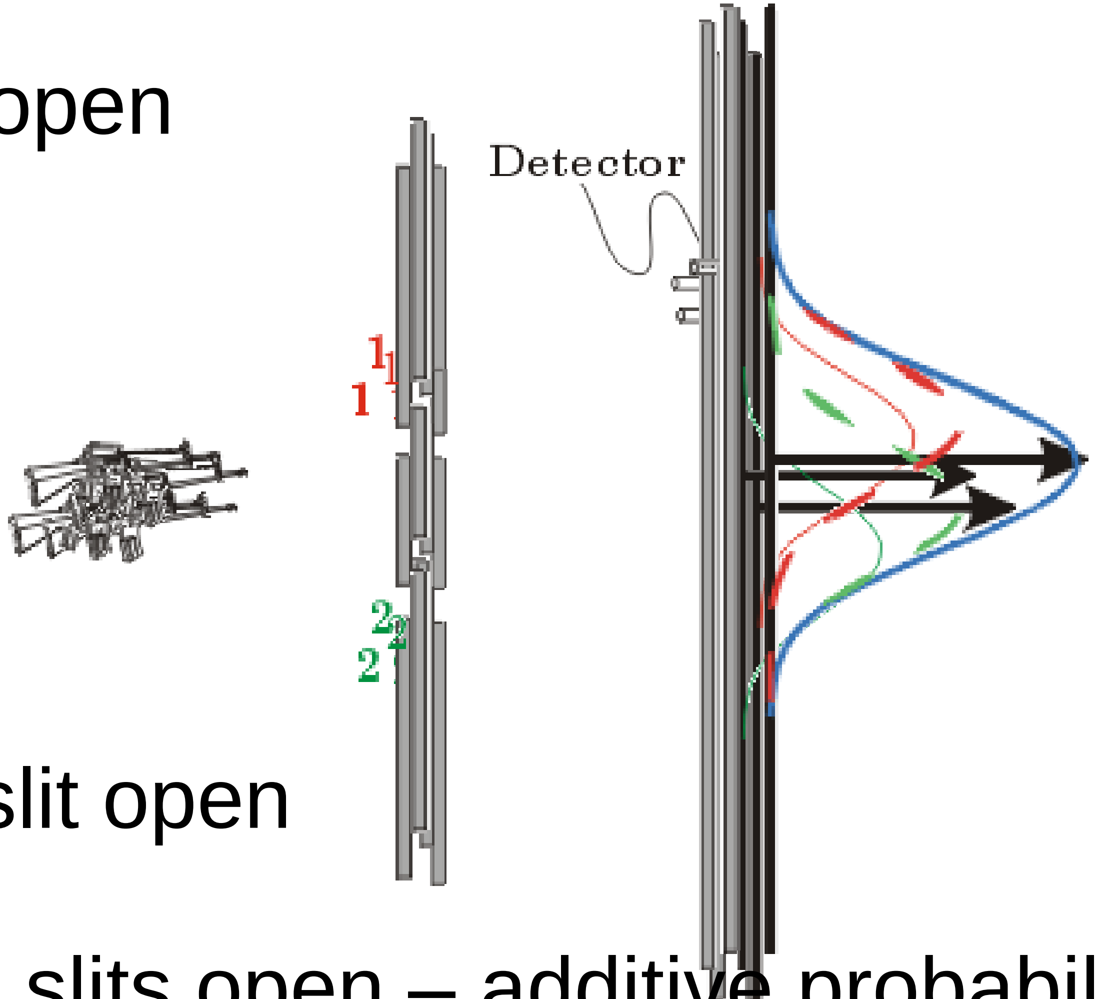
2-slit experiment: bullets

Top slit open

P_1

P_2

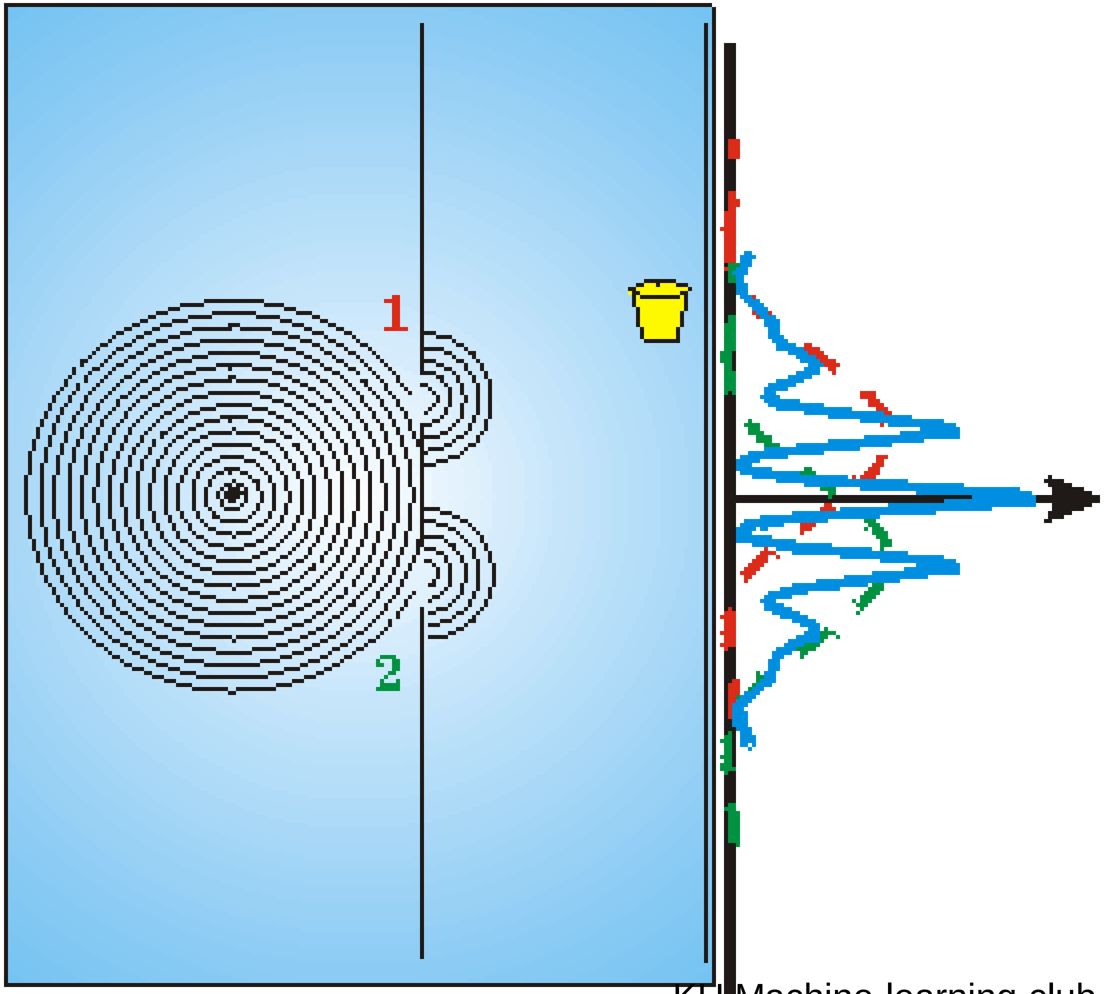
$P_{12} = P_1 + P_2$



Bottom slit open

Both slits open – additive probabilities for exclusive events - Laplace

2-slit water experiment - interference



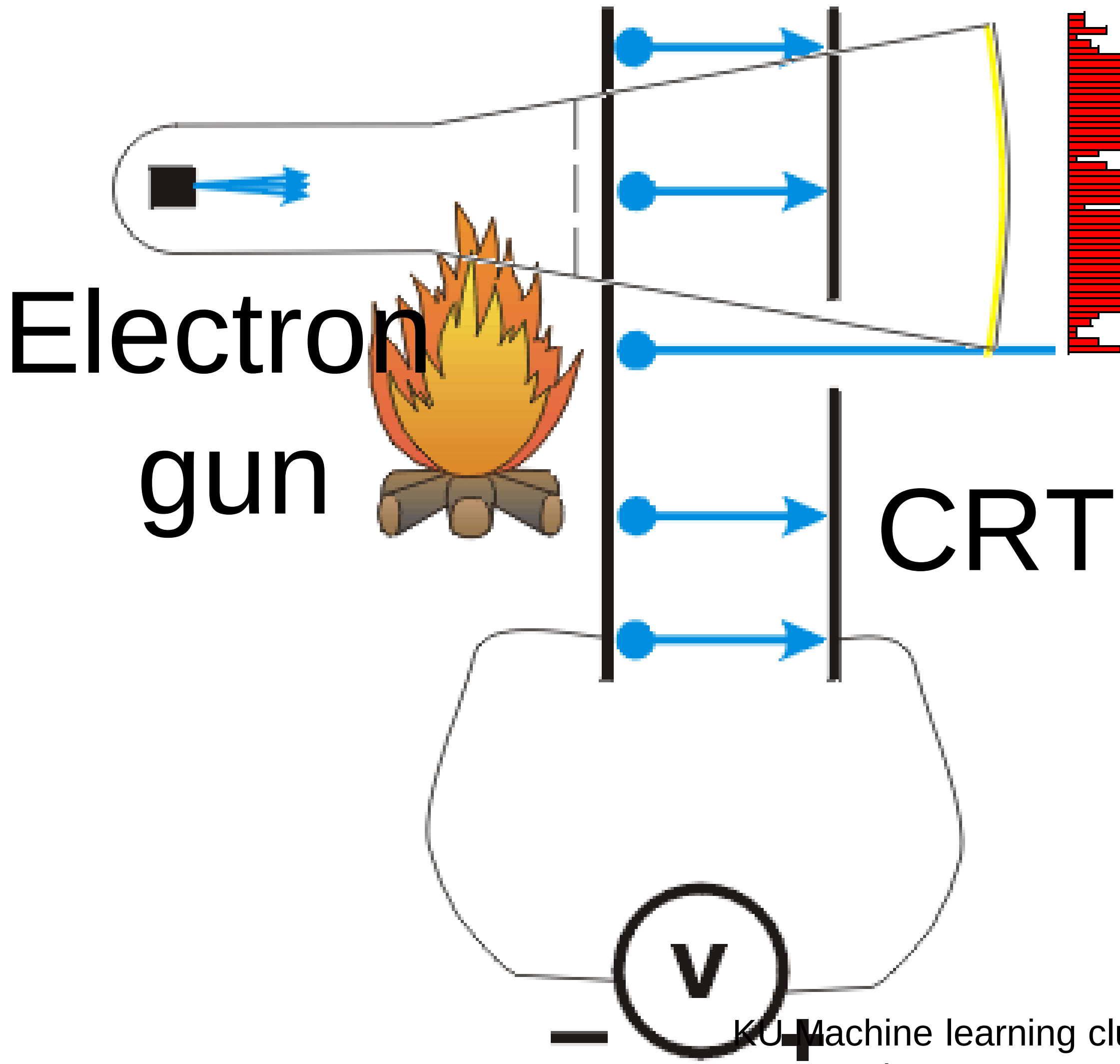
$$I = |\phi|^2$$

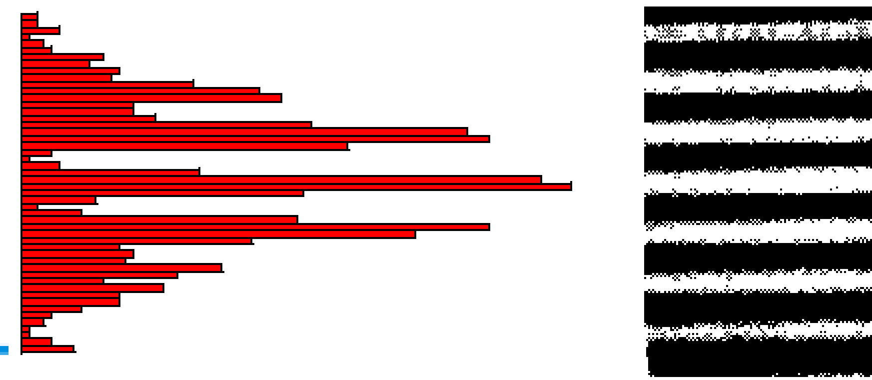
$$\phi = \phi_1 + \phi_2$$

$$I_1 = |\phi_1|^2; \quad I_2 = |\phi_2|^2$$

$$I_{12} = |\phi_1 + \phi_2|^2$$

How about one at a time?





$$P = |\phi|^2$$

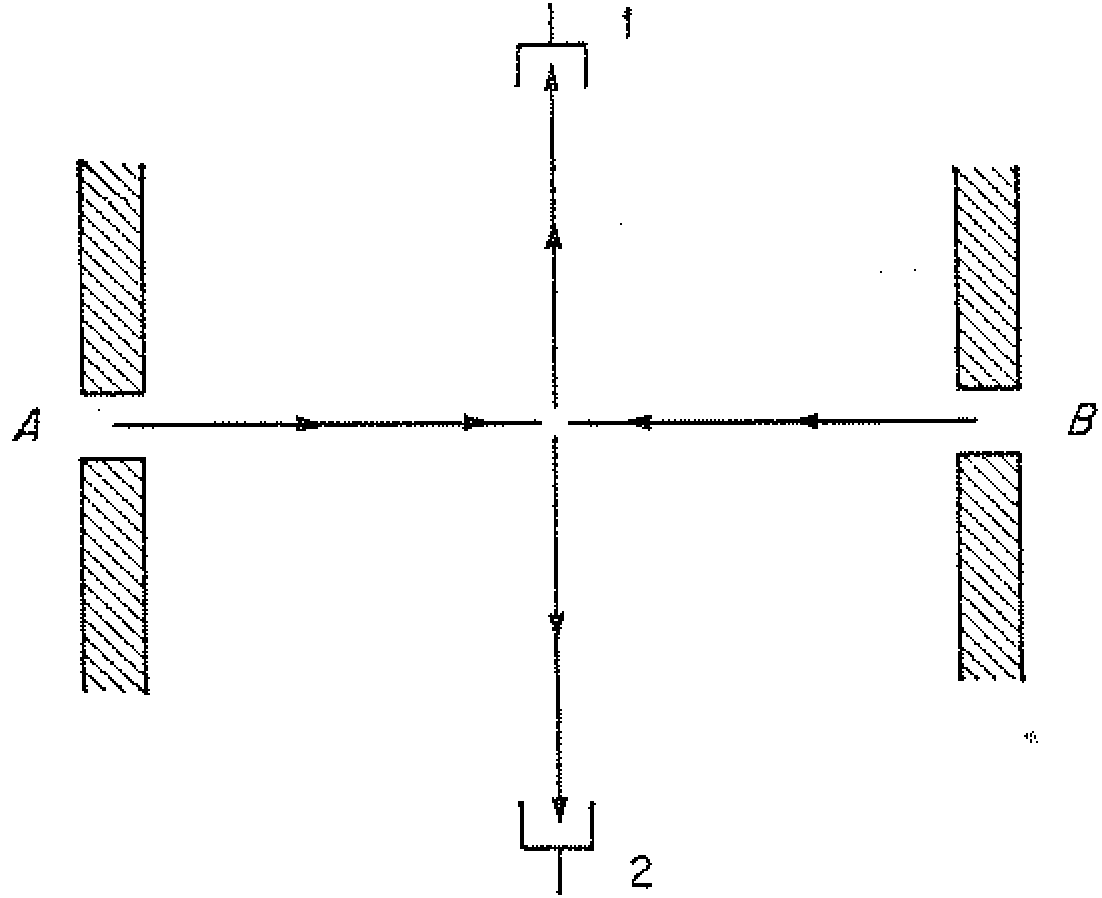
$$P_1 = |\phi_1|^2; \quad P_2 = |\phi_2|^2$$

$$\phi = \phi_1 + \phi_2$$

$$P_{12} = |\phi_1 + \phi_2|^2$$

Look f

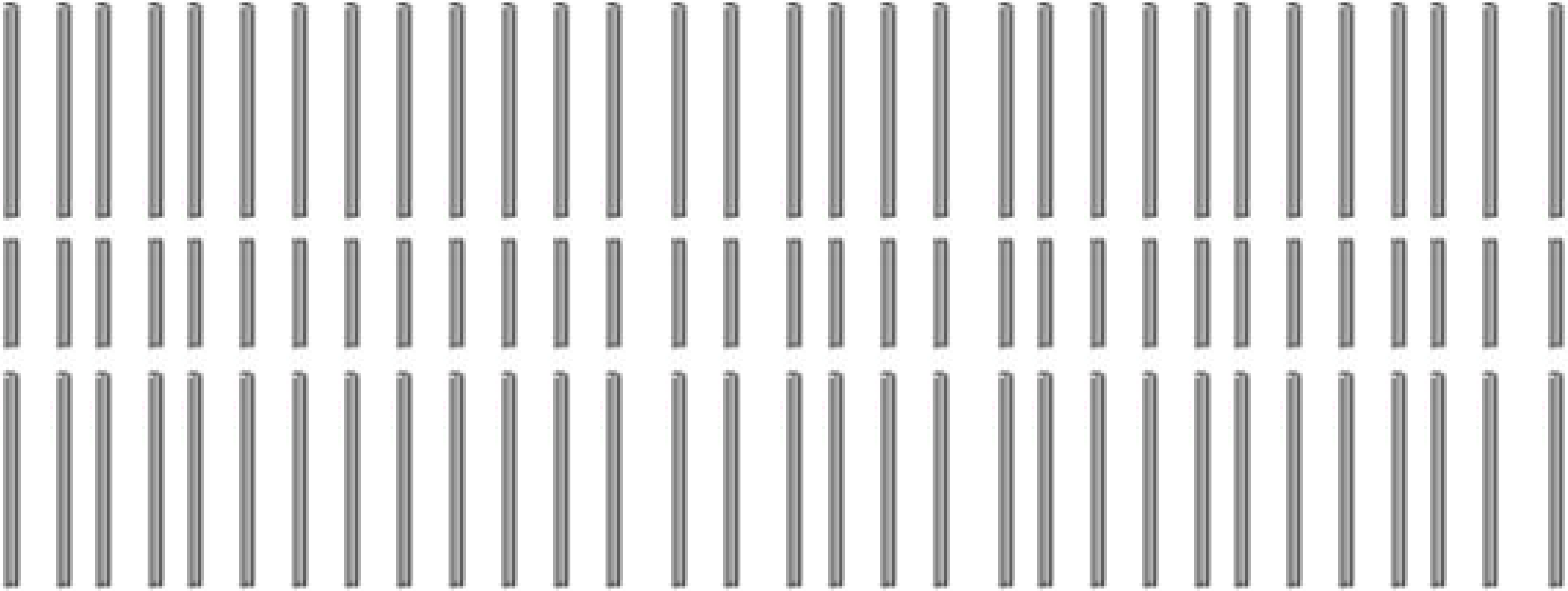
- See none: W
- See them all: P
- See some: W+F



- Worse – if we COULD look (but don't)! - P

$$p = \left|\Phi_{AB}(1,2)\right|^2$$

$$\left|\Phi_{AB}(1,2)\right|^2 + \left|\Phi_{AB}(2,1)\right|^2 = 2p \qquad \left|\Phi_{AB}(1,2) + \Phi_{AB}(2,1)\right|^2 = 4p$$



Path integrals

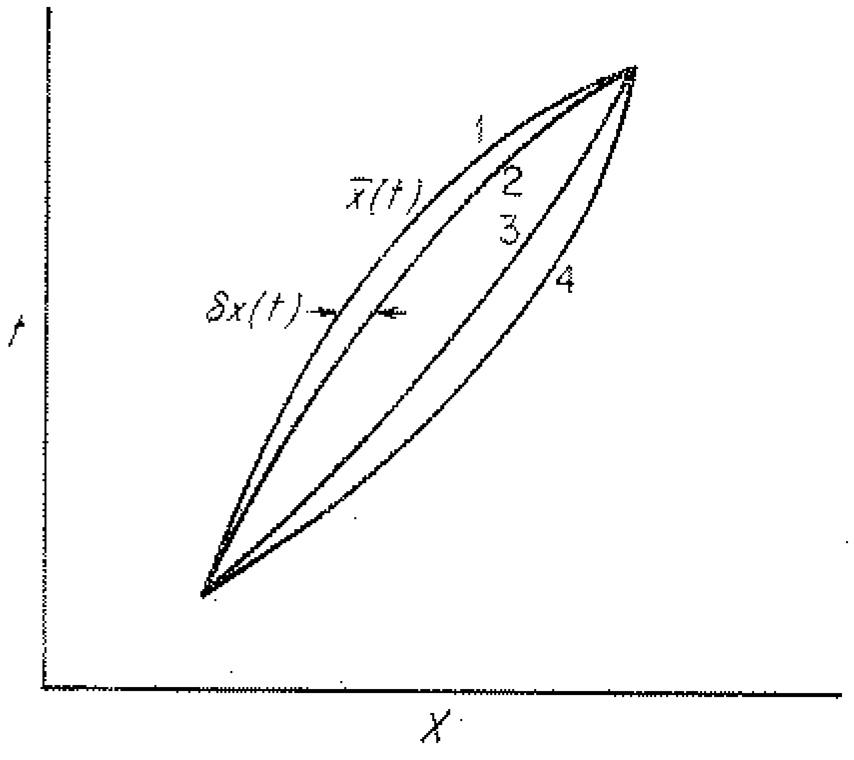
$$\phi(l,0;r,t) = \sum_{strings\{\alpha\}} \phi_N(l,0;\alpha_1,\frac{t}{N};\alpha_2,\frac{2t}{N};\alpha_3,\frac{3t}{N};...;\alpha_N,\frac{t(N-1)}{N};r,t)$$

$$= \sum_{strings\{\alpha\}} K(l,0;\alpha_1,\frac{t}{N})K(\alpha_2,\frac{2t}{N};\alpha_3,\frac{3t}{N})\cdots K(\alpha_N,\frac{t(N-1)}{N};r,t)$$

$$K = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{i}{\hbar}\begin{pmatrix} H_{rr} & H_{rl} \\ H_{lr} & H_{ll} \end{pmatrix}\Delta t$$

$$\phi(l,0;r,t) = \int_{x(0)=l}^{x(t)=r} D[x(t)]\; e^{\frac{-i}{\hbar}\int_0^t H(t)dt}$$

Classical mechanics



a few nearby paths

Schrodinger Algorithm

$$|\psi(t+\Delta t)> = [1 - iH\Delta t/\hbar]|\psi(t)>$$

$$\frac{|\psi(t+\Delta t)> - |\psi(t)>}{\Delta t} = \frac{-i}{\hbar} H |\psi(t)>$$

$$i\hbar \frac{\partial |\psi(t)>}{\partial t} = H |\psi(t)>$$

$$|\psi(t)> = e^{-iHt/\hbar} |\psi(0)>$$

How to do Quantum Mechanics

$$x \rightarrow x_{op}$$

$$x \rightarrow x_{op} = i\hbar \frac{\partial}{\partial p}$$

$$p \rightarrow p_{op} = \frac{\hbar}{i} \frac{\partial}{\partial x}$$

$$p \rightarrow p_{op}$$

i

$$\Delta x \Delta p \geq \frac{\hbar}{2}$$

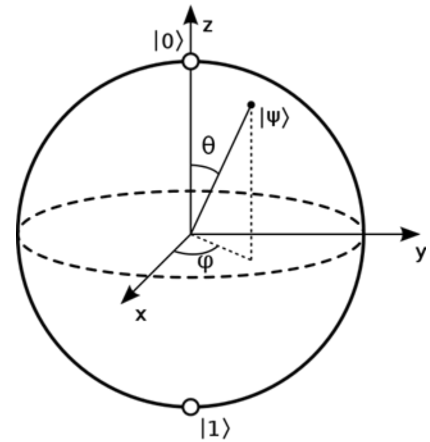
$$p = \frac{h}{\lambda}$$

$$[x, p] = i\hbar$$

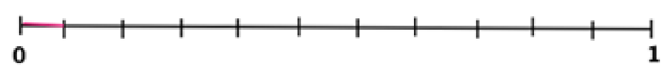
Qubits vs classical bits

$|0\rangle$ or $|1\rangle$ vector "in z basis": $\begin{pmatrix}1\\0\end{pmatrix}$ or $\begin{pmatrix}0\\1\end{pmatrix}$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \epsilon \mathbb{C}$$
$$= \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$



Compare: $\begin{bmatrix}a\\b\end{bmatrix}$



Multiple qubits

$|0\rangle|0\rangle$ $|0\rangle|1\rangle$ $|1\rangle|0\rangle$ $|1\rangle|1\rangle$, or $|00\rangle$ $|01\rangle$ $|10\rangle$ $|11\rangle$

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

compare: $\begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix}$

Superposition gives us entanglement!

Entanglement

Two possible states: $|\uparrow>, \ |\downarrow>$

Or: $\alpha\,|\uparrow> + \beta\,|\downarrow> = \cos(\theta)\,|\uparrow> + \sin(\theta)\,|\downarrow>$

Now suppose two electrons: $(\alpha\,|\uparrow> + \beta\,|\downarrow>)(\gamma\,|\uparrow> + \delta\,|\downarrow>)?$

$a\,|\uparrow\uparrow> + b\,|\uparrow\downarrow> + c\,|\downarrow\uparrow> + d\,|\downarrow\downarrow>$

Check: $\alpha\gamma\,|\uparrow\uparrow> + \alpha\delta\,|\uparrow\downarrow> + \beta\gamma\,|\downarrow\uparrow> + \beta\delta\,|\downarrow\downarrow>$

Bell state: $a\,|\uparrow\uparrow> + d\,|\downarrow\downarrow>$

Starting with the Schrödinger equation

$$\frac{\partial \rho}{\partial t} = \frac{1}{i\hbar}[H, \rho]$$

where ρ is the density matrix and H is the Hamiltonian. For an N-qubit system, the Hamiltonian H is:

$$H = \sum_{i=1}^{N} K_\alpha \sigma_{x\alpha} + \epsilon_\alpha \sigma_{z\alpha} + \sum_{\alpha \neq \beta = 1}^{N} \zeta_{\alpha\beta} \sigma_{z\alpha} \sigma_{z\beta}$$

where $\{\sigma\}$ are the Pauli operators corresponding to each qubits, $\{K\}$ are the tunneling amplitudes, $\{\epsilon\}$ are the biases, and $\{\zeta\}$ are qubit-qubit couplings. We encode the weights into these parameters.

Feedforward quantum temporal network

The general solution to the Schrodinger equation is mathematically isomorphic to the equation for information propagation in a neural network

Compare $\frac{\partial \rho}{\partial t} = -iL\rho$, where $L = \frac{2\pi}{ih}[\dots, H]$, to

$$\phi_i = \sum_j w_{ij} f_j(\phi_i)$$

$$\phi_{output} = F_W \phi_{input}$$

where ϕ_{output} and ϕ_{input} are the output and input vector of the networks.

F_W is the network operator, which depends on the neuron connectivity weight matrix W.

For our system,

- $\rho(0)$ = Input state vector
- $\rho(t_f)$ = Output state vector
- $\{K, \epsilon, \zeta\}$ = Weights of the network, which can be adjusted experimentally as functions of time for physical implementation

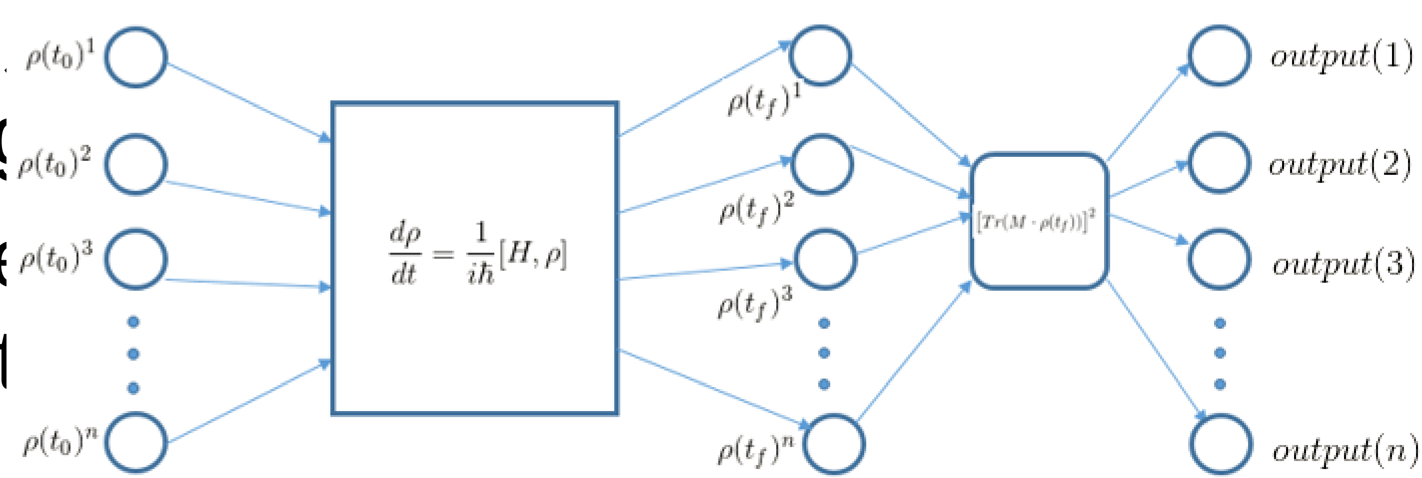
Our goal is to train the external parameters

$\{K, \epsilon$

usin puts

Once on

addi



Training paradigms

1. Quantum Backprop (quantum simulation in Matlab)
2. Reinforcement Learning of Fourier Parameters (quantum simulation in Matlab)
3. Reinforcement Learning of quantum circuit (quantum simulation in Qiskit - IBM, Q# - Microsoft)
4. Levenberg-Marquart
5. …

Quantum Backprop

Given an initial state, ρ(0), the system evolves in time according to the Schrödinger equation:

$$\frac{\partial \rho}{\partial t} = -\frac{i}{\hbar}\left[H, \rho\right]$$

We construct a Lagrangian to be minimized:

$$L = \frac{1}{2}\left[d - \langle M(T)\rangle\right] + \int_0^T \lambda^+(t)\left(\frac{\partial \rho}{\partial t} + \frac{i}{\hbar}\left[H, \rho\right]\right)\gamma(t)dt,$$

where we define the output measure M by:

$$\begin{aligned} Out &= \langle M(T)\rangle = tr(\rho(T)M) \\ &= \sum_i p_i \left|\psi_i(T)\right\rangle\left\langle\psi_i(T)\right| M \\ &= \sum_i p_i \left\langle\psi_i(T)\right| M \left|\psi_i(T)\right\rangle, \end{aligned}$$

We take the first variation of *L* with respect to *ρ*, set it equal to zero, then integrate by parts to give:

$$\lambda_i \frac{\partial \gamma_j}{\partial t} + \frac{\partial \lambda_i}{\partial t}\gamma_j - \frac{i}{\hbar}\sum_k \lambda_k H_{ki}\gamma_j + \frac{i}{\hbar}\sum_k \lambda_i H_{jk}\gamma_k = 0,$$

with the boundary conditions at the final time *T* given by

$$-\left[d - \langle M(T)\rangle\right] M_{ji} + \lambda_i(T)\gamma_j(T) = 0 \cdot$$

The gradient descent learning rule is given by

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w},$$

for each weight parameter *w*, where η is the learning rate and

$$\frac{\partial L}{\partial w} = \frac{i}{\hbar}\int_0^T \lambda^+(t)\left[\frac{\partial H}{\partial w}, \rho\right]\gamma(t)dt$$

$$= \frac{i}{\hbar}\int_0^T \sum_{i,j,k}\left(\lambda_i(t)\frac{\partial H_{ik}}{\partial w}\rho_{kj}\gamma_j - \lambda_i(t)\rho_{ik}\frac{\partial H_{kj}}{\partial w}\gamma_j\right)dt$$

Reinforcement Learning

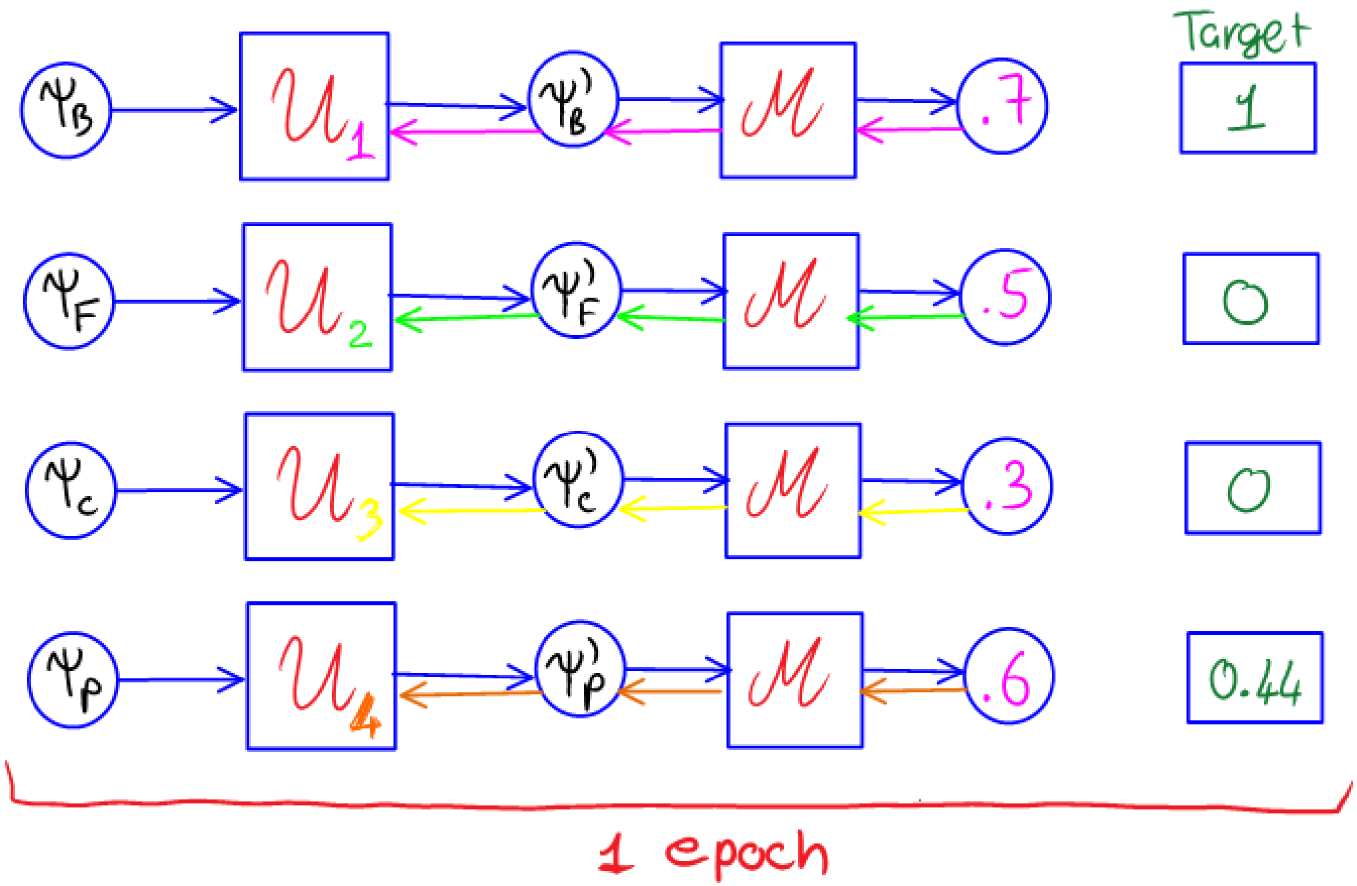
1. Initial error E computed
2. A parameter P is perturbed

$$P_{new} = P + \Delta P$$

3. New error computed with perturbed parameter
4. Error gradient computed

$$Grad = \frac{E_{new} - E}{\Delta P}$$

5. Parameter updated using the gradient and a learning rate $P = P - \eta Grad$ for each training pair and each parameter.


Target
Ψ_B
U_1
Ψ'_B
M
.7
1
Ψ_F
U_2
Ψ'_F
M
.5
0
Ψ_C
U_3
Ψ'_C
M
.3
0
Ψ_P
U_4
Ψ'_P
M
.6
0.44
1 epoch

- Can it do known classical neural network calculations?
  - Logic gates yes
  - Pattern recognition/classification yes
  - Function reproduction yes
- Can it do known quantum calculations? Yes – this is quantum universality -
- Can it do UNKNOWN quantum calculations?

Algorithm construction

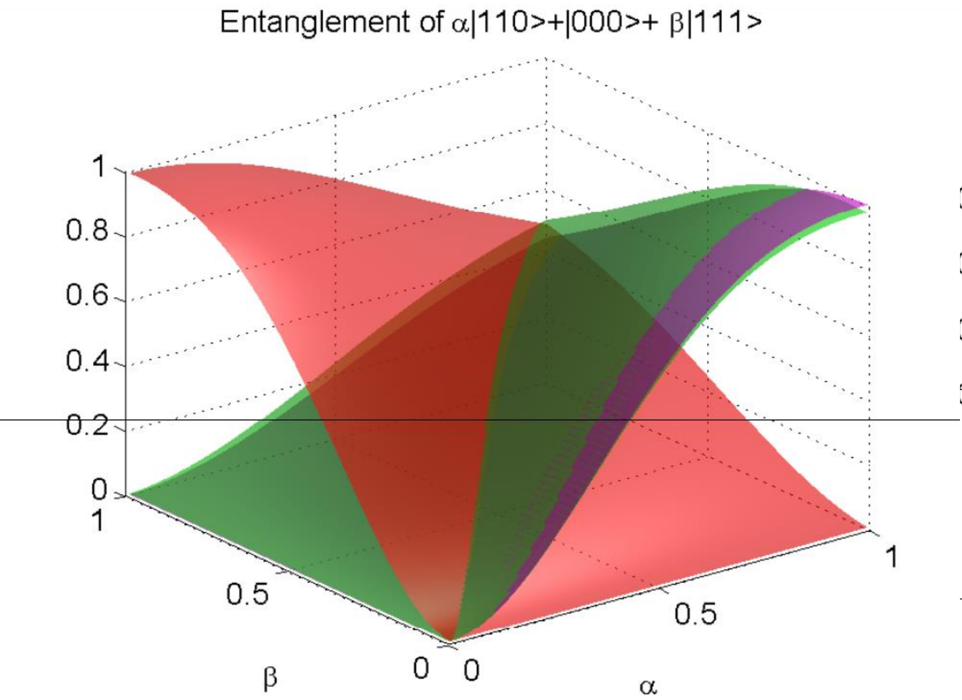
- What is essentially non-classical/quantum mechanical?
  - Superposition – a “qubit” can be in simultaneous mutually exclusive states
  - Entanglement – a consequence of superposition – crucial to most quantum computing/information applications
  - Separability is NP-hard (2003, Gurvitz)

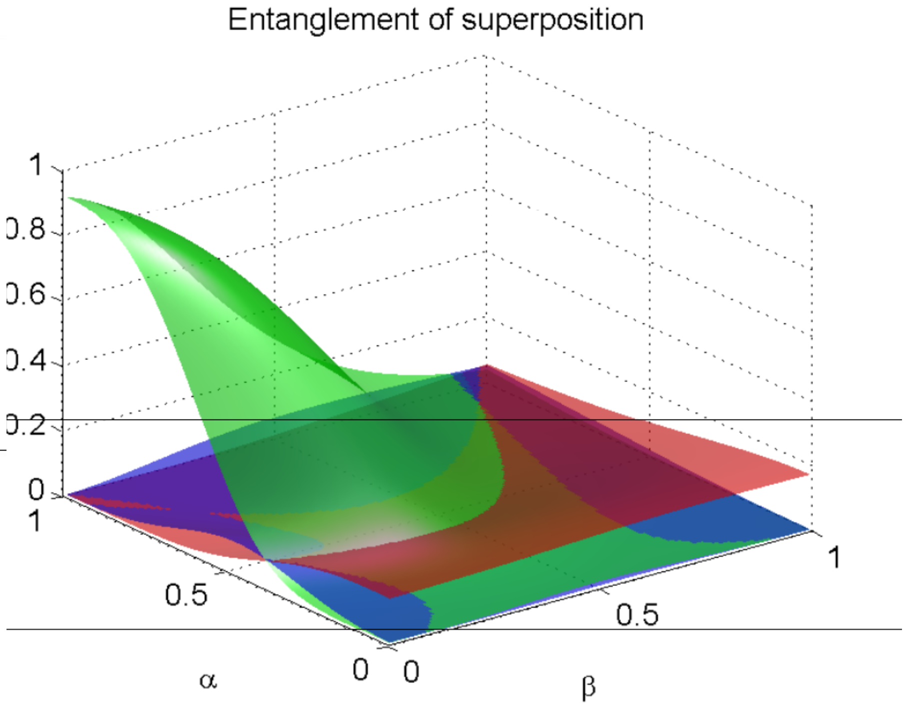
****
****

  - Experimental measure for entanglement?
  - Prepare a highly entangled state?

Entanglement indicator

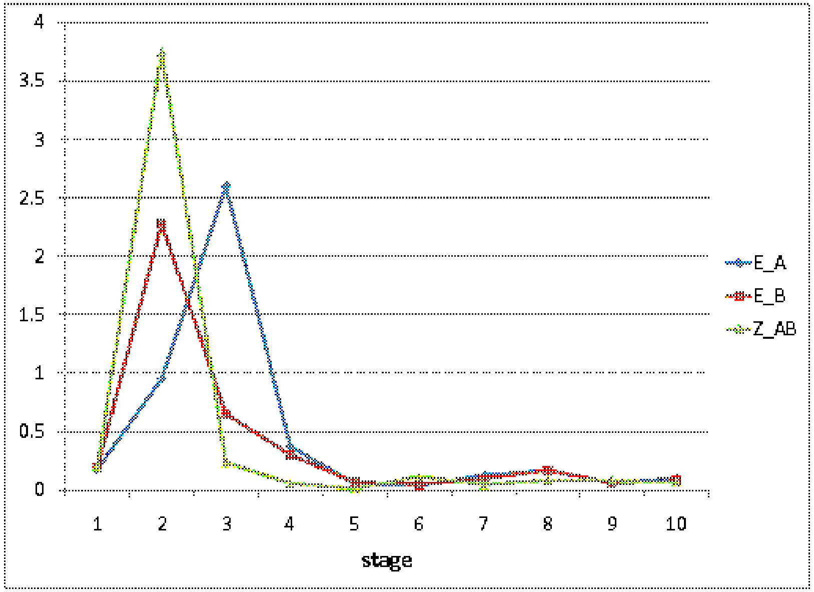
QNN learns to calculate its own entanglement

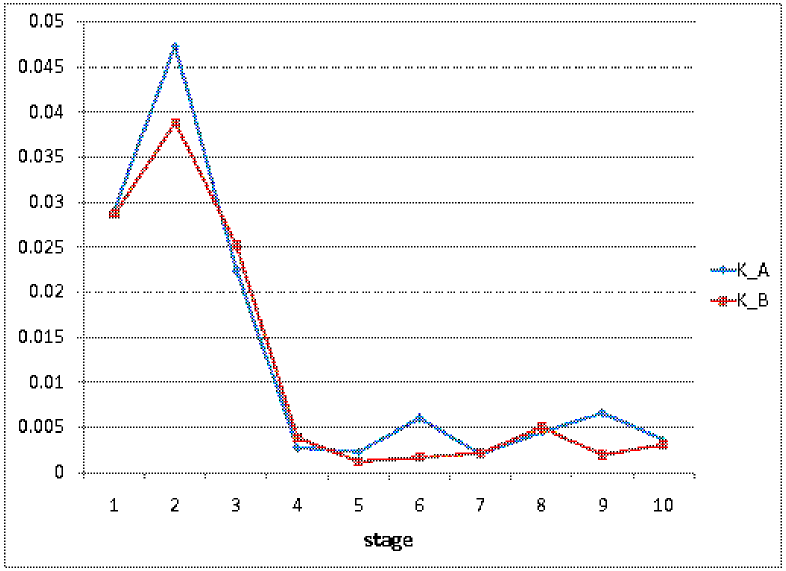




“Bootstrapping” to larger systems: a partial solution to the scaling problem

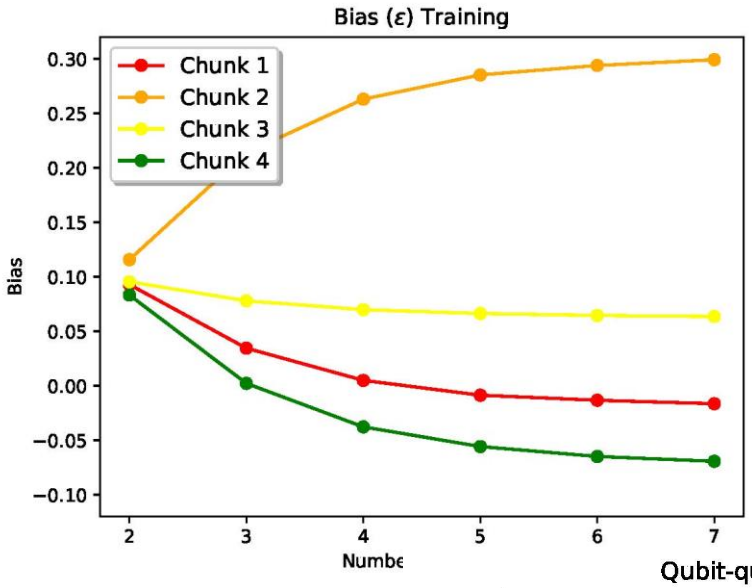
- Use trained parameter functions found for n as starting point for training functions for n+1
- Entanglement indicator: for n qubits, $\binom{n}{2}$ pairs; $\binom{n}{3}$ triplets, etc.

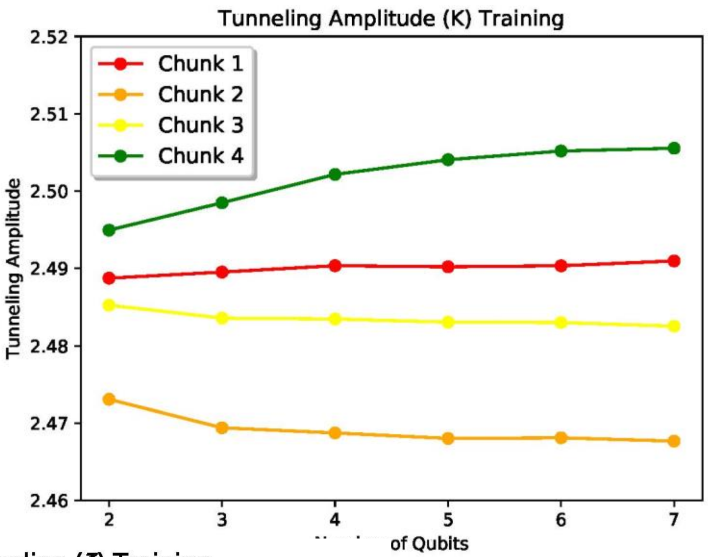


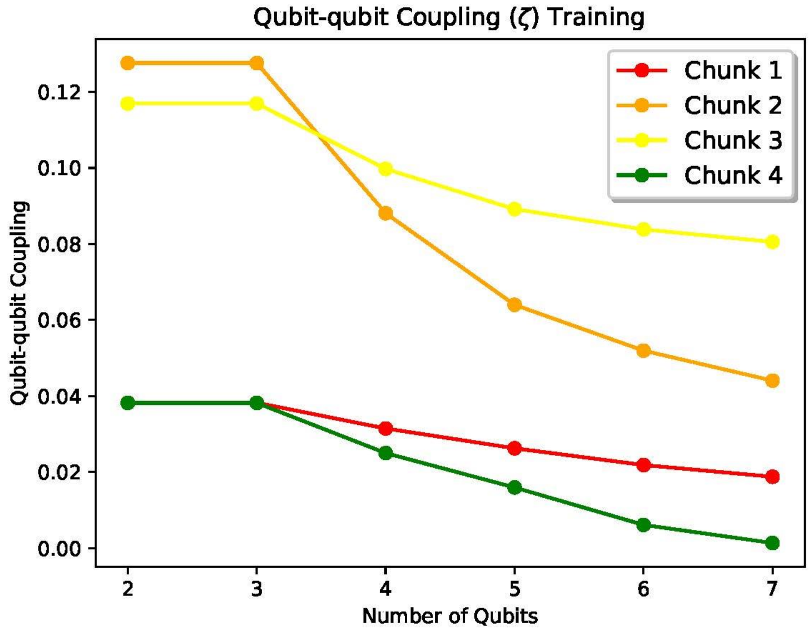


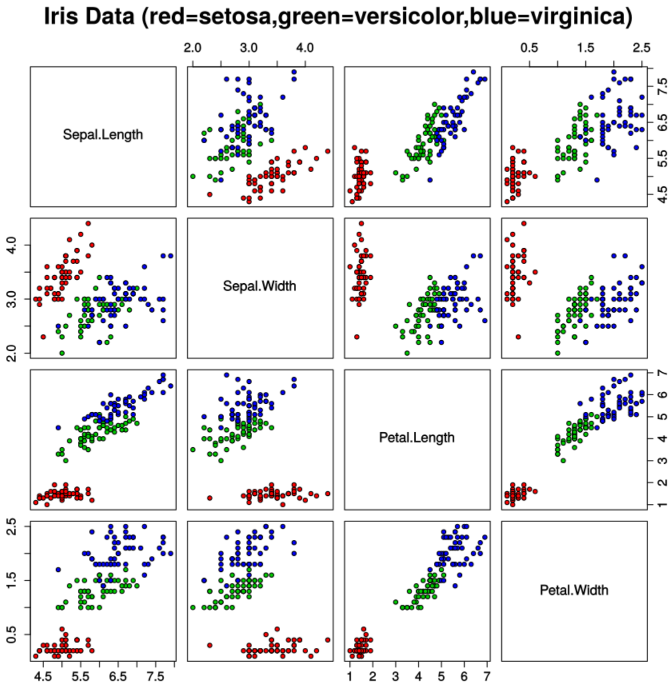
Power of interconnectivity

Asymptotic limit?









RVNN: 50,000 epochs
CVNN: 1000 epochs
QNN: 100 epochs

	Training RMS (%)			Testing RMS (%)			Testing Accuracy (%)		
Training Pairs	RVNN	CVNN	QNN	RVNN	CVNN	QNN	RVNN	CVNN	QNN
75	3.45	2.06	0.96	3.71	7.78	2.31	100	100	97.3
30	0.51	0.41	1.1	4.97	9.47	9.78	93.3	96.0	97.5
12	0.69	0.09	0.62	11.9	16.4	7.48	85.3	94.7	85.5

Benchmarking: classical problem

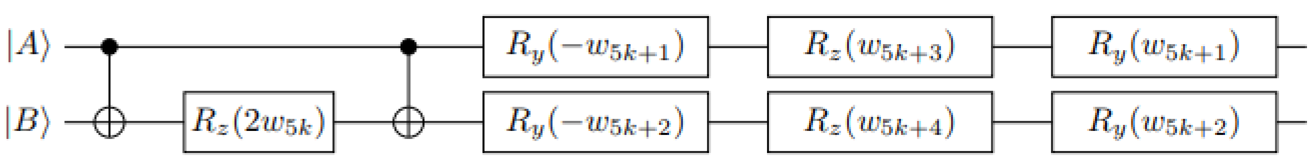
Benchmarking: Quantum problem

	Training RMS error (%)			
Training Pairs	N.Works	RVNN	CVNN	QNN
100	5.66	3.74	0.97	0.04
50	5.96	5.89	0.53	0.09
20	6.49	4.97	0.04	0.2
4	0.00	0.93	0.01	0.2

	Testing RMS error(%)			
Training Pairs	N.Works	ANN	CVNN	QNN
100	7.56	5.39	3.61	0.2
50	7.91	10.7	6.00	0.3
20	13.6	15.5	9.48	0.4
4	48.2	51.9	55.0	0.4

Reverse engineering

- Solovey-Kitaev Theorem: Any quantum operation can be approximated by a sequence of gates from any universal set (e.g., H, rotations, CNOT)
- A solution to one of the challenges in q computing: state preparation
- The parameters bootstrap experimentally providing a possible mesoscopic pairwise entanglement indicator

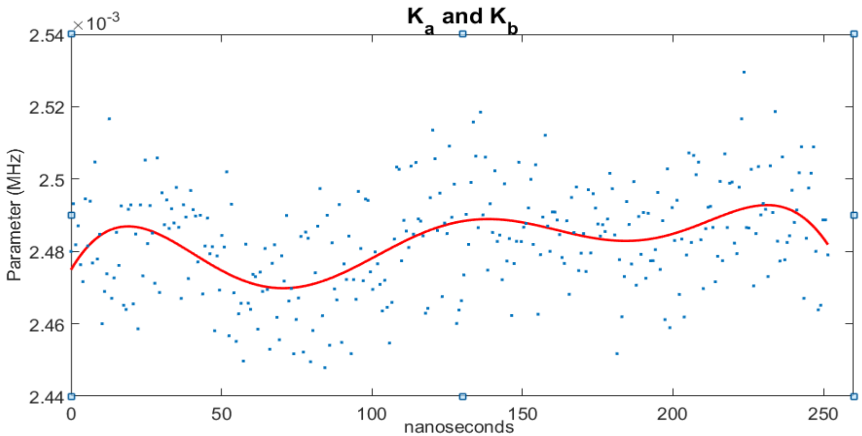


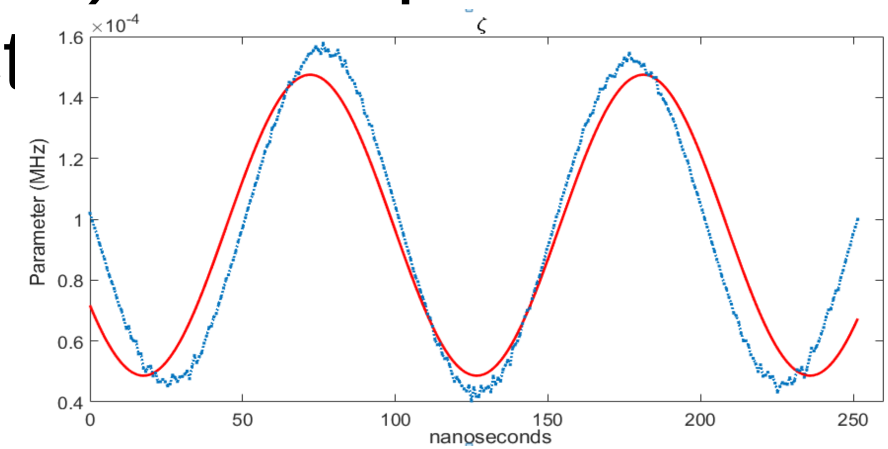
Noise and decoherence

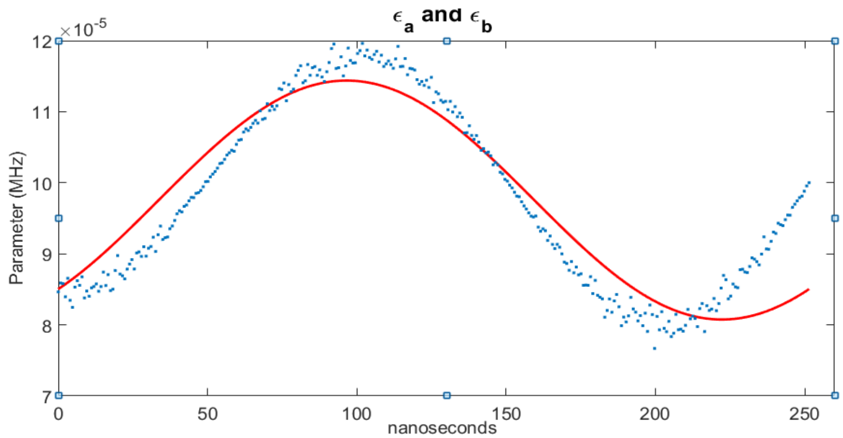
- Real physical systems are noisy
- Models are always approximate
- Real quantum systems also decohere
- Classically, neural nets are robust to noise and to incomplete and/or damaged data
- QNN? – yes!

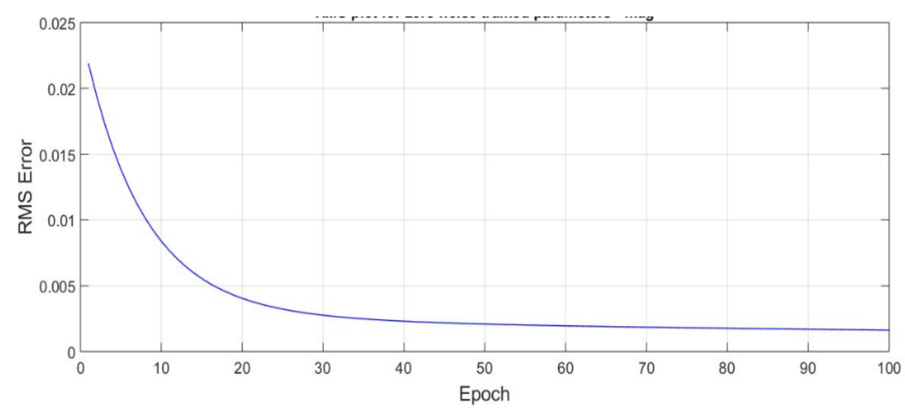
Training with (magnitude) noise- parameters

ct

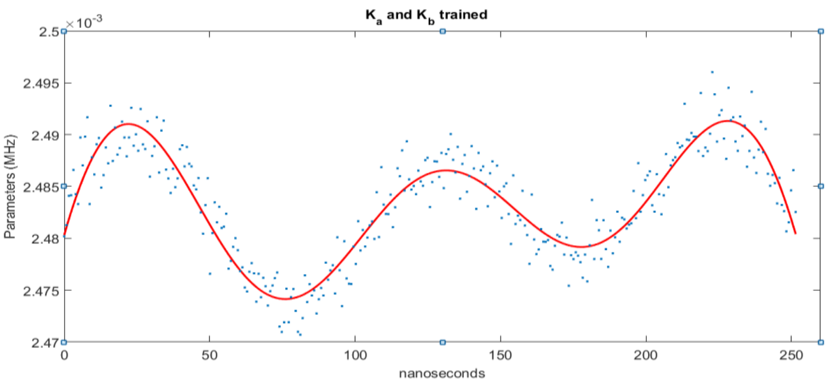


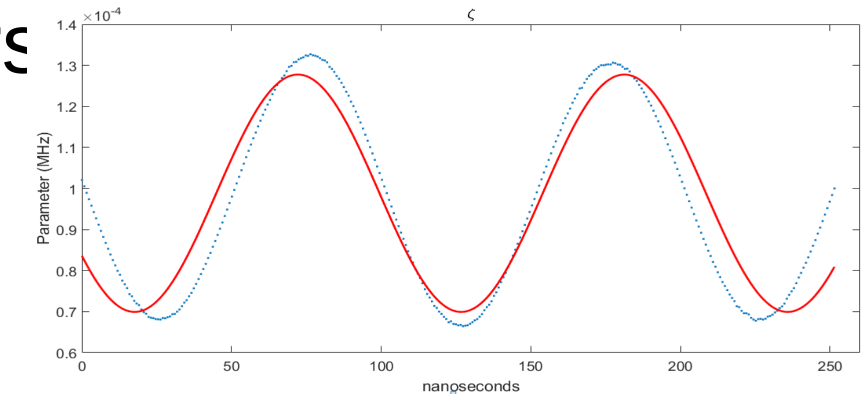


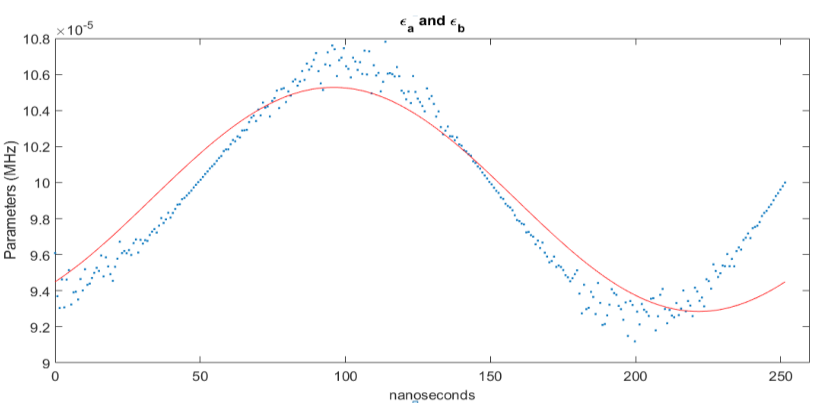


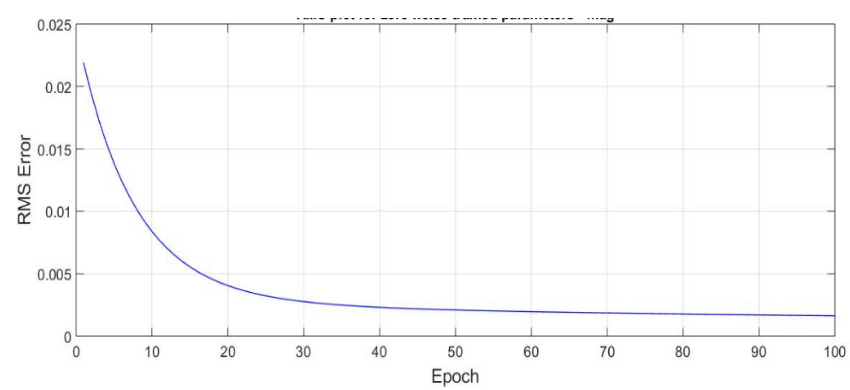


Training with (phase noise) decoherence-
ters

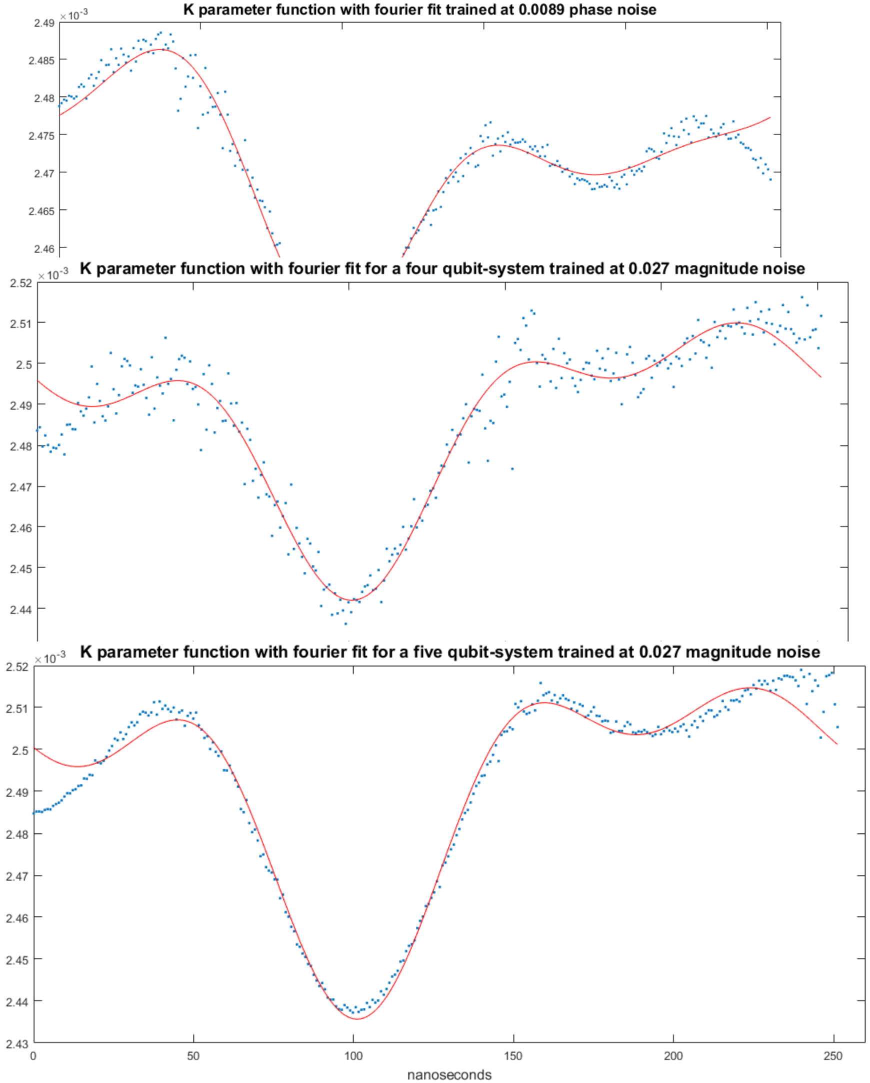








Robustness increases with size!



The marriage of quantum computing and machine learning can

1. Design algorithms
2. Provide robustness to noise and to decoherence
3. Provide automatic scaleup
4. Equivalent universality
5. Work in any experimental implementation
6. Take unknown factors into account

N.L. Thompson, N.H. Nguyen, E.C. Behrman, and J.E. Steck, “Experimental pairwise entanglement estimation for an N-qubit system: A machine learning approach for programming quantum hardware,” (to appear); arXiv:1902.07754

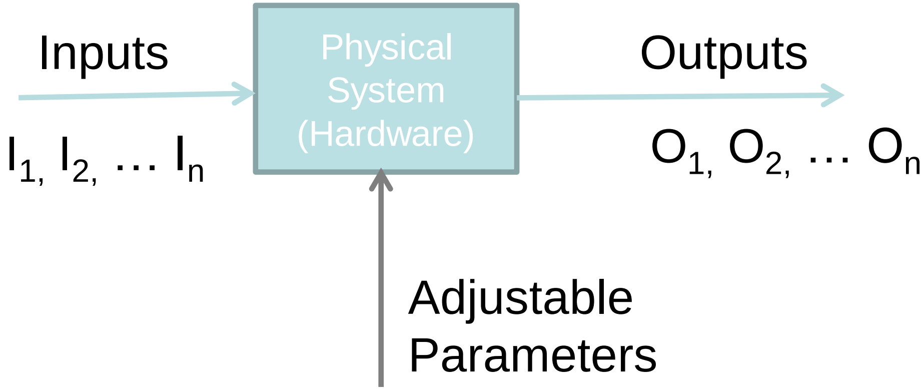
N.H. Nguyen, E.B., M.A. Moustafa and J.E. Steck, “Benchmarking neural networks for quantum computation,” *IEEE Transactions on Neural Networks and Learning Systems* **31**, 2522-2531 (2020); arXiv:1807.03253

E.B. and J.E. Steck, “Learning quantum annealing,” *Quantum Information and Computation* **17**, 0469-0487 (2017); arXiv:1603.01752v2

N.H. Nguyen, E.B., and J.E. Steck, “Quantum learning with noise and decoherence: A robust quantum neural network,” *Quantum Machine Intelligence* **2,** 5-15 (2020); arxiv.org/abs/1612.07593

Recent Relevant References

Our Contributions to Quantum Machine Learning



Learning: Choosing Adjustable Parameters from examples to get correct I->O map

- Quantum Dots – learned 2 input classical and quantum logic gates (1993-2002)
- Developed temporal and spatial architectures (1993-5)
- Learned 2, 3 qubit quantum gates (2005)
- Quantum processing and storage in spatial arrays (1993-2004)
- Genetic algorithm for finding pulse sequences for nmr computing (2009)
- Learned Entanglement Witness for 2,3, 4, 5, 6 qubits (2002-2008)
- Learned and corrected phase shifts in quantum states(2013)
- Learned Entanglement Indicator robust to Noise, Decoherence (2015)
- Learned Quantum Annealing to entangled states 2, 3, 4, 5, 6 qubits (2016)
- Complexity and power of qnn vs complex valued nn (2017)
- Bootstrapping (partial knowledge for inference to larger) (2018)
- Benchmarking against RVNN, CVNN (2019)
- Experimental entanglement witness for NISQ machines (2019)