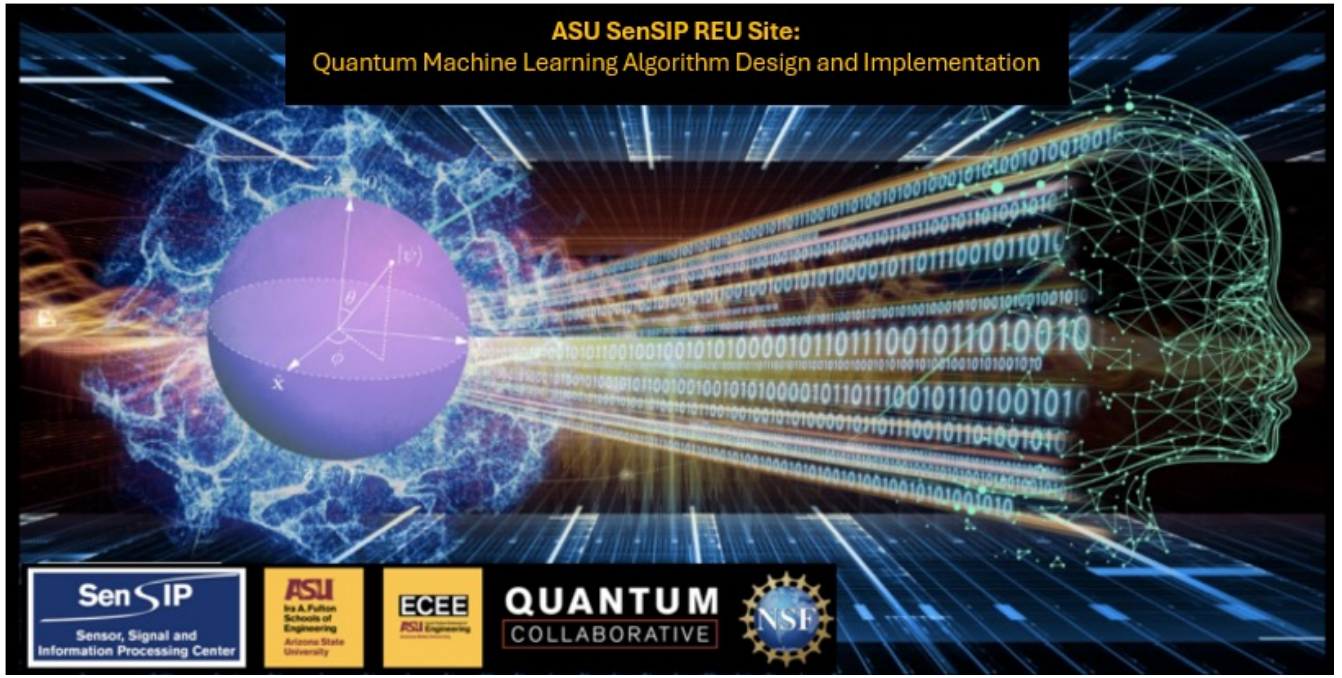


QML REU 2024 – Presentation

Chris Su

Carnegie Mellon University
Mentors: Greg Vetaw, Glen Uehara,
Dr. Suren Jayasuriya, Dr. Andreas Spanias

NSF Award 2349567

https://sensip.engineering.asu.edu/sensip-quantum-dsp-ai-reu/

Carnegie Mellon University

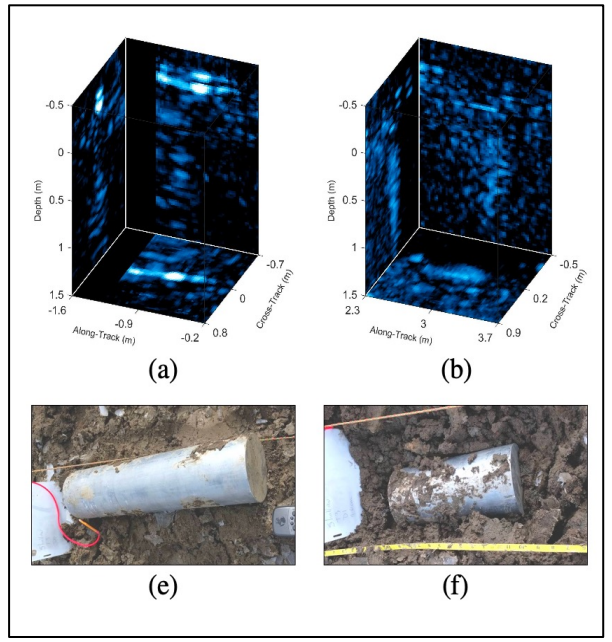
Research Background

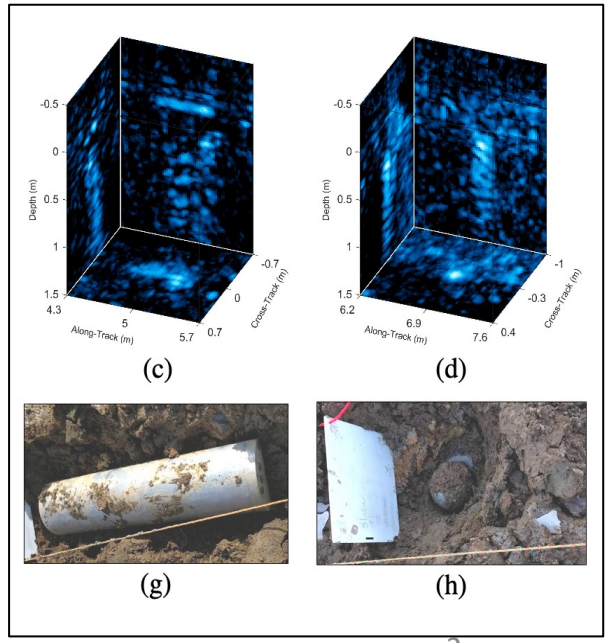
- 3D Synthetic Aperture Sonar Data Cubes targets that mimic underwater ordnances

- Want to classify them, but several challenges:
  - High dynamic range -> Classic range compression techniques may lose important information
  - Targets return weaker acoustic signals when compared to clutter

- Became sort of an “exemplar” problem for a bigger picture of 3D data processing.
  - 3D data processing is understudied, most algorithms are tailored for 2D data
  - Why?
    - 3D data more expensive to procure, more expensive compute

- How can we make it more efficient? (in terms of memory usage, computational efficiency, etc?)

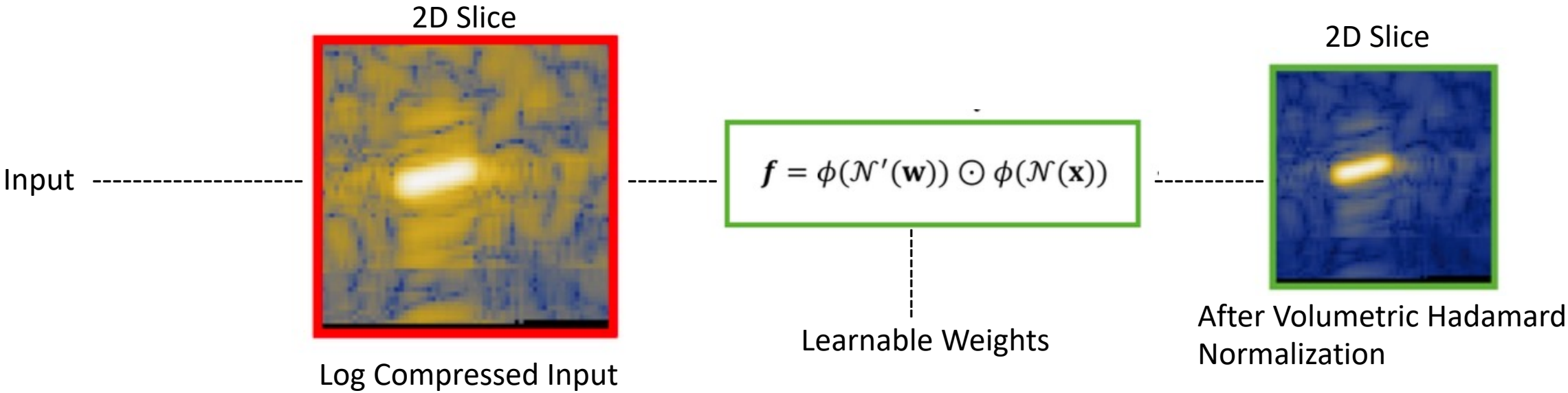




https://sensip.engineering.asu.edu/sensip-quantum-dsp-ai-reu/

Research Background: Classical Preprocessing (VHN)

(Past works) Intuition / Hypothesis: Tone mapping might help the CNN learn the targets from clutter, so why not make the tone mapping learnable? Hence, learnable weights are used to tone map (and showed increase in AUC precision recall metric) [*]

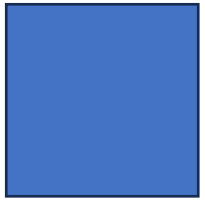


*Gregory Vetaw, Benjamin Cowen, Daniel C. Brown, David P. Williams, Suren Jayasuriya, "Learning-based Tone Mapping to Improve 3D SAS ATR", *International Geoscience and Remote Sensing Symposium (IGARSS) 2023*

Research Objectives

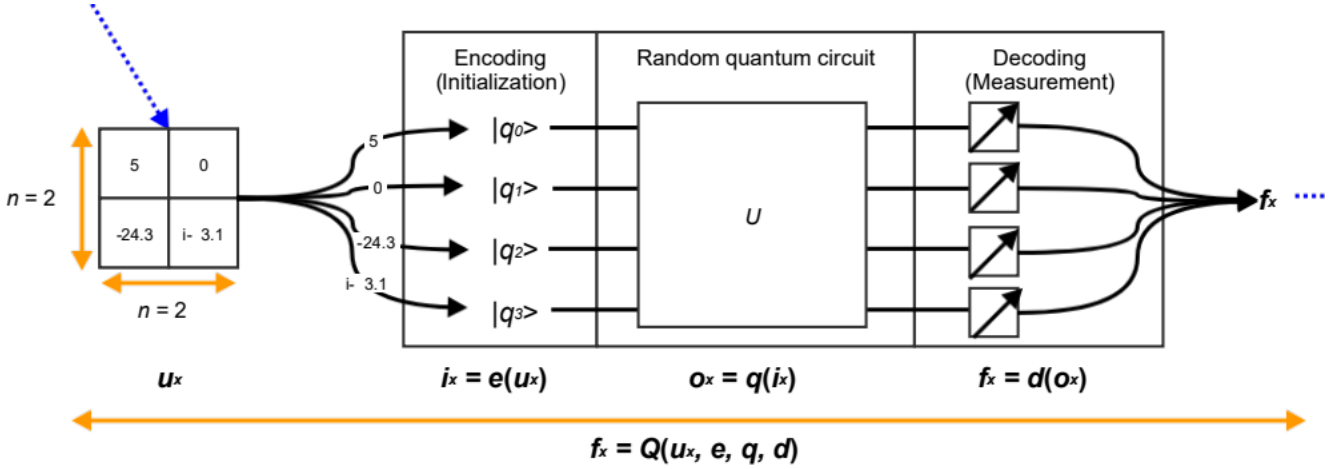
1) Can we make a Quantum VHN? (QVHN)

$$f = \phi(\mathcal{N}'(\mathbf{w})) \odot \phi(\mathcal{N}(\mathbf{x}))$$

<--------> 

Quantum Circuit

2) Quantum Convolutional Kernels* for 3D Data?



*Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, Tristan Cook. *"Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits.", Arxiv*

https://sensip.engineering.asu.edu/sensip-quantum-dsp-ai-reu/

Making a Quantum VHN: Hadamard Product

-The Hadamard product is an elementwise multiplication of matrices and can be thought of as a ”naïve matrix multiplication”

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & 8 & -2 \end{bmatrix} \circ \begin{bmatrix} 3 & 1 & 4 \\ 7 & 9 & 5 \end{bmatrix} = \begin{bmatrix} 2 \times 3 & 3 \times 1 & 1 \times 4 \\ 0 \times 7 & 8 \times 9 & -2 \times 5 \end{bmatrix} = \begin{bmatrix} 6 & 3 & 4 \\ 0 & 72 & -10 \end{bmatrix}$$

-Consequently, this generalizes to 3-D and n-D matrices, if they are of the same shape.

-At the root of the Hadamard product is multiplication!

-For our purposes, given our input data cube (represented as a matrix) which is of dimension 64 x 64 x 101 we have a learnable weight matrix of the same dimension which we apply the Hadamard product to

-We use the Hadamard product as a tone mapping operation!

-Can we do this more efficiently in quantum???

Image from wikipedia.com/hadamard_product

Making a Quantum VHN: Shortcomings of Big O in Practice

Big O notation hides constants in the term.

(proof)

Recall: $f(n) \in O\big(g(n)\big) \Leftrightarrow \exists C, \exists x_0, \forall x \geq x_0, f(x) \leq Cg(x)$

Intuitively this means that eventually, f(x) is always smaller than Cg(x).

But now we want to show that $cf(x) \in O\big(f(x)\big), \forall f(x) \geq 0$

Choose $C = c$, trivially, $cf(x) \leq cf(x)$ (for arbitrary x)

Making a Quantum VHN: Shortcomings of Big O in Practice

But what does this mean?

Does $cf(x) \in O(f(x))$ have any implications in **practice**?

Yes!

EXAMPLE:

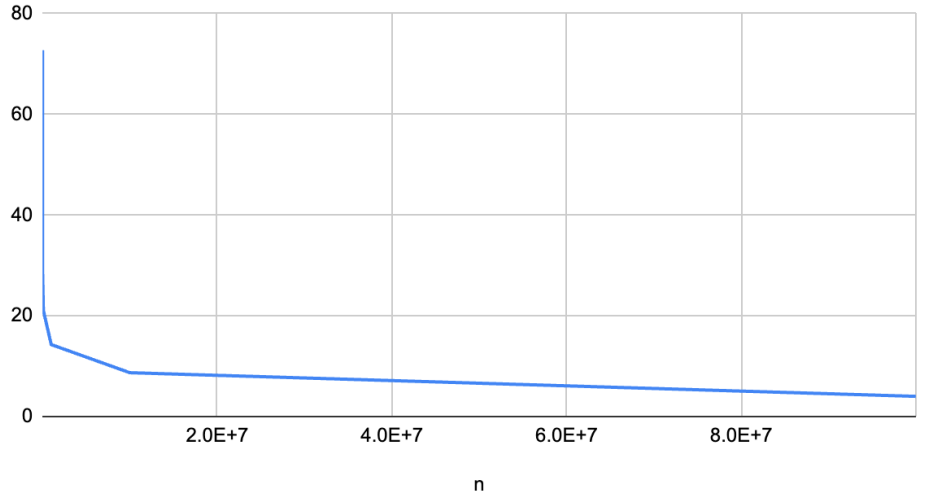
Consider algorithm A that runs in $O(n^2)$ and algorithm B that runs in $O(n^3)$

Now suppose A hides a large constant $c = 10^{10}$, while B does not hide a constant c (i.e, $c = 1$) in the big O.

Per the figure, algorithm B that is "theoretically slower" than algorithm A is practically faster than algorithm A until $n = 10^{10}$

.

Algorithm B vs Algorithm A w.r.t Data input length



Making a Quantum VHN: Shortcomings of Big O in Practice

Takeaway:

Big O may not be the most informative metric of computational efficiency in practice.

In practice, an “efficient” quantum VHN with applications in ML asks whether we can parallelize multiplication using quantum circuit. We tentatively concluded that such an algorithm is not suitable for a NISQ era but more so when we are able to test practical algorithmic behavior as opposed to comparing complexities.

TLDR: Making a quantum VHN is kind of like comparing apples and oranges

NAME	DATATYPE	TYPE	RUNTIME	FFT/ QFFT?	PROB.?	GAL.?
Grade School scalar	Scalar	Classical	O(n^2)	N	N	N
Karatsuba	Scalar	Classical	O(n^1.58)	N	N	N
Schonhage-Strassen	Scalar	Classical	O(nlogn* loglogn)	Y	N	Y
Harvey-Hoeven	Scalar	Classical	O(nlogn)***	Y	N	Y
Ramezani et. al. [1]	Scalar	Quantum	O(n^(½)(logn)^2)	Y	Y	N
Grade school matrix	Matrix	Classical	O(n^3)	N	N	N
Strassen	Matrix	Classical	O(n^(2.807))	N	N	Y
Li et. al. [2]	Matrix	Quantum	O(n^2logn)	N	Y	N/A
Randall [3]	Matrix	Classical	O((logn)^2)	N	N	N

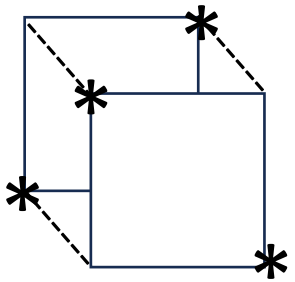
KEY:
DATATYPE: The datatype in question
TYPE: Classical / Quantum / Hybrid
RUNTIME: Asymptotic runtime in big O
FFT/QFFT: Whether the algorithm uses any sort of FFT or QFFT transform
PROB: Whether the algorithm is probabilistic / randomized or not. For probabilistic or randomized algorithms, the runtime is **AFTER** boosting. (?)
GAL: Whether the algorithm is galactic (see WIKI: Galactic algorithm - Wikipedia). In essence, whether the algorithm is practically implemented or not. (some algorithms hide big constants in big O notation since c*f(n) = O(f(n))

Turning to Quantum Convolutional Kernels

- A paper proposed ‘quantum convolutional kernels’ that were a quantum analogue to classical convolutions.*

- The intuition:
  - Quantum kernels may have an easier time accessing higher-dimensional spaces and more complex convolutions that may be classically intractable to compute

- Our research for now focuses on randomizing these circuits.
- ”Lattice” Quantum Kernels:
  - A quantum kernel of size $k * k * k$ needs $k * k * k$ qubits
  - This is impractical to simulate for $k > 2$
  - We used a kernel of size 2,stride 2 and used half of the points in the cube, thereby only needing 4 qubits to simulate a quantum convolutional kernel



*Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, Tristan Cook. *“Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits.”, Arxiv*

SVSS Dataset and CNN Architecture

We used a subset of the original dataset produced by Sediment Volume Search Sonar (SVSS).

Number of classes	2
Number of targets (Train, Test)	500, 120
Number of clutter (Train, Test)	500, 120
Dimensions of cube	64 x 64 x 101
CPU	Apple M2 arm64
OS	macOS
Memory	16GB Unified memory

CNN 1, CNN 1 + VHN (Used in classical, Brown et. al*, Vetaw et. al)

CNN 2 (Used with quantum kernels)

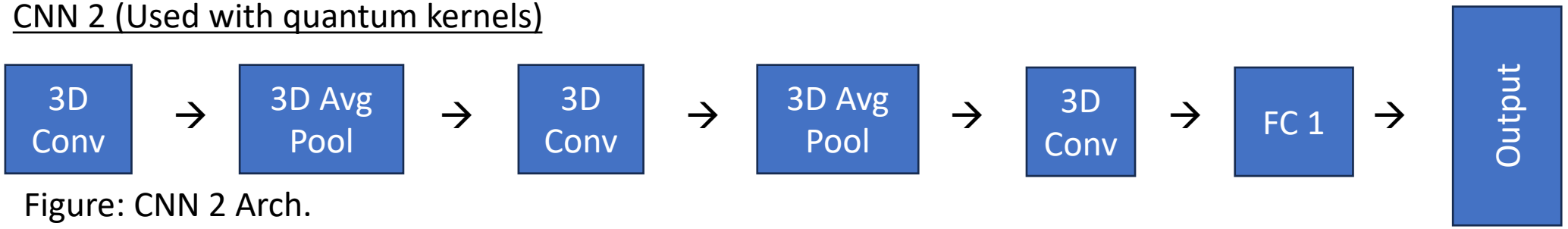


Figure: CNN 2 Arch.

*Williams and D. Brown, "Three-dimensional convolutional neural networks for target classification with volumetric sonar data," *Proc. of Meetings on Acoustics*, vol. 44, p. 070005, Aug. 2021.

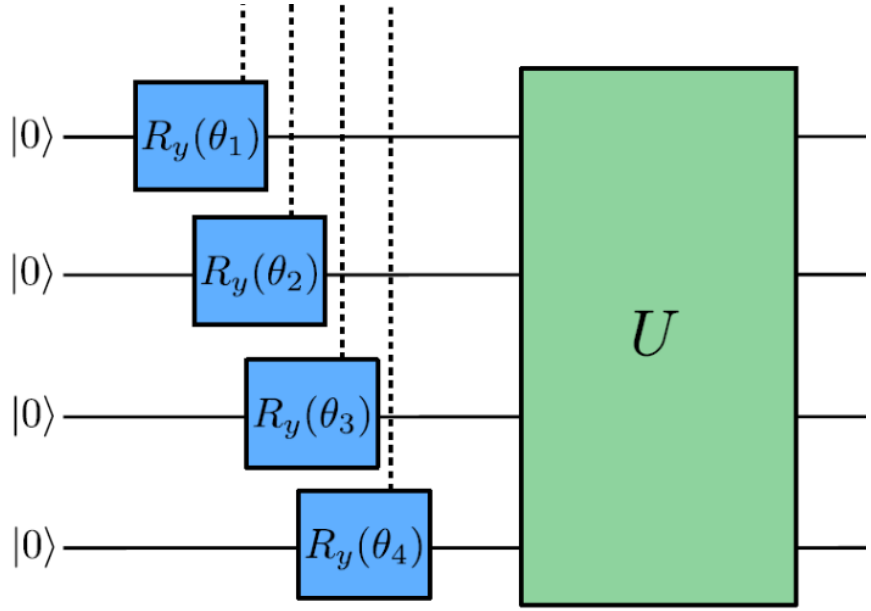
Research Results (Subset of full dataset)

	VHN + CNN	Q Conv + CNN	CNN
AUCPR	86 ± 1	83 ± 1	83 ± 1
Number of parameters	8.2M	2,845	2,342
Avg. Inference Time	0.01952s	0.00226s	0.01134s
Training Epochs	150	150	150
Optimizer	Adam	Adam	Adam
Learning Rate	0.0008	0.0005	0.0004
Betas	0.5, 0.9	0.5, 0.9	0.9, 0.99

- Used the Area Under Curve of Precision Recall metric that helps with imbalanced classes and scores how well the model identifies true positives
- A **quantum convolutional kernel achieves comparable performance with the baseline CNN**, and **reduces inference computation time by a factor of 5**
- **Requires less parameters** than the VHN + CNN, and **reduces inference computation time by a factor of 10**

Concluding Remarks / Future Research

- Train on larger data with imbalanced data
- Use more filters in quantum kernel
  - Using more filters in convolution neural networks usually increases performance
  - This was also shown for quantum kernels in the original paper
  - Our research only used 1 filter so increasing number of filters is the next step



*Images from pennylane, zdnet

https://sensip.engineering.asu.edu/sensip-quantum-dsp-ai-reu/

Acknowledgements

- Special thanks to my mentors Greg Vetaw, Glen Uehara, Dr. Suren Jayasuriya, Dr. Andreas Spanias
- Special thanks to Ms. Sayed
- Special thanks to NSF for sponsoring the program.