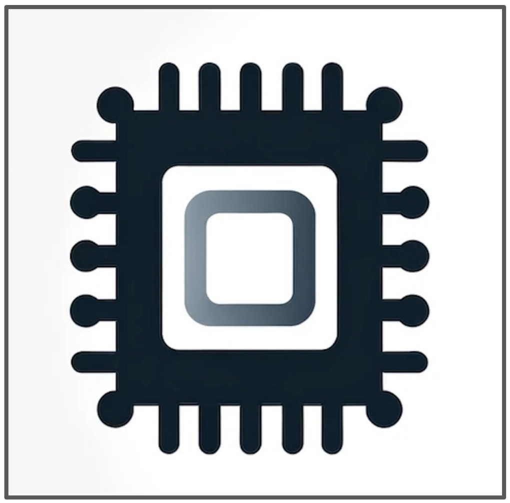
# Quantum-Inspired Computing for Cybersecurity

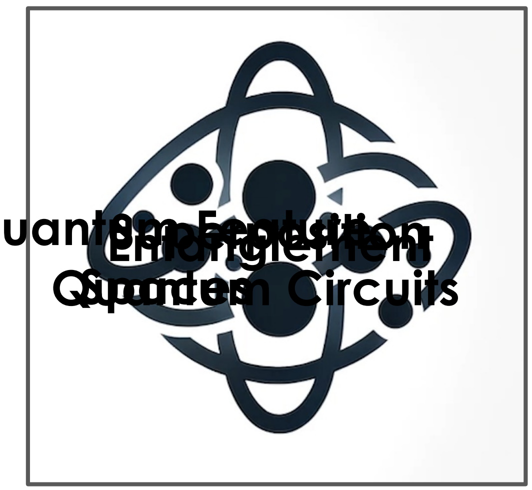
[1,2] Larry Huynh, [1] Jin Hong, [1] Ajmal Mian,
[2] Hajime Suzuki, [2] Harish Vallury, [2] Seyit Camtepe

[1] University of Western Australia
[2] CSIRO Data61

# What is "*Quantum-Inspired*"?

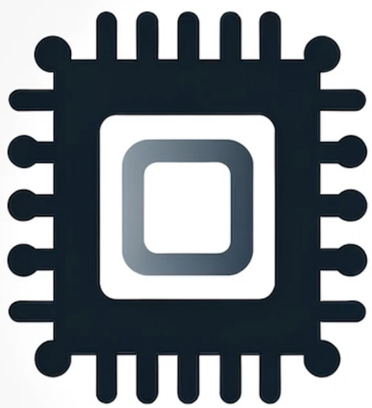


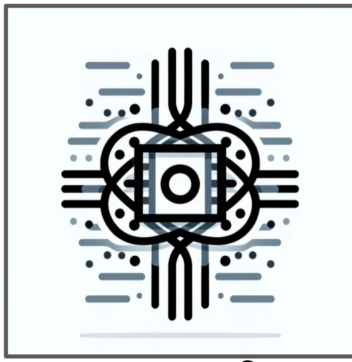
Classical Computing




Quantum Computing

# What is "*Quantum-Inspired*"?





ment

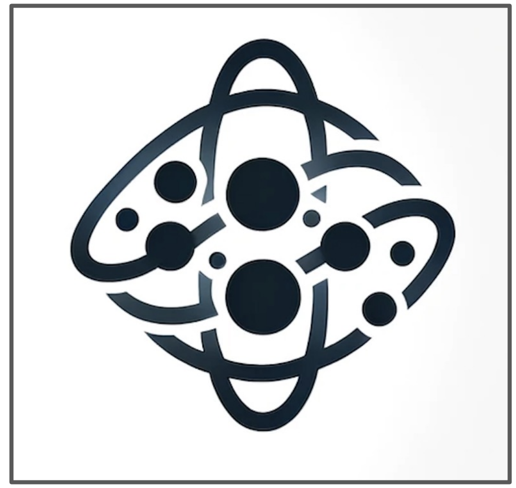
sition

Feature

Spaces

Quantum-Inspired
Computing

Quantum Circuits



Classical Computing

Quantum Computing

# Quantum-Inspired Computing

THE UNIVERSITY OF WESTERN AUSTRALIA

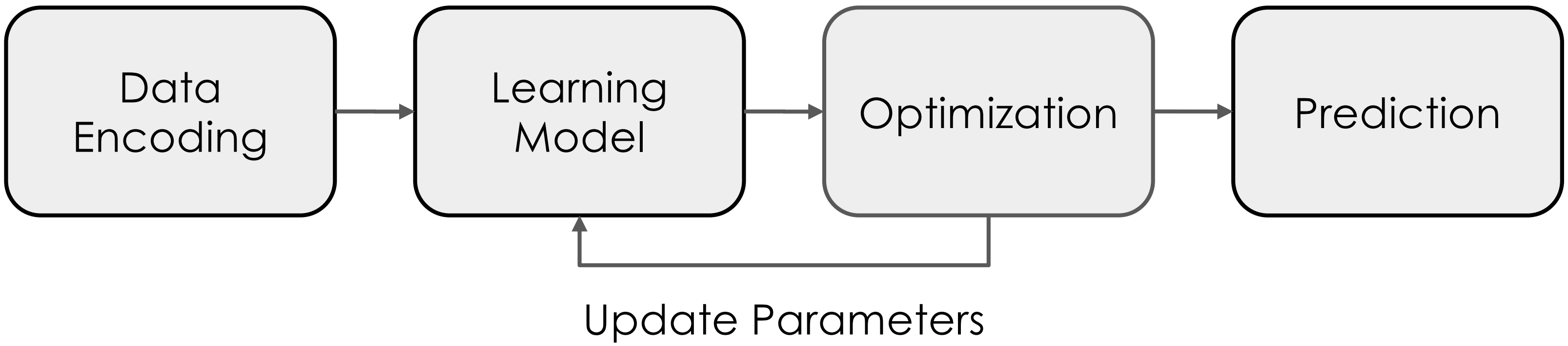
**Methods in:**
- Optimization
- Search Algorithms
- Machine Learning
- …

**Applications in:**
- Finance
- Medicine
- Cybersecurity
- ...

**Advantages over Classical Computing:**
- Richer search spaces, greater expressivity
- Possible speed-ups and performance improvements
- Reduced model parameters

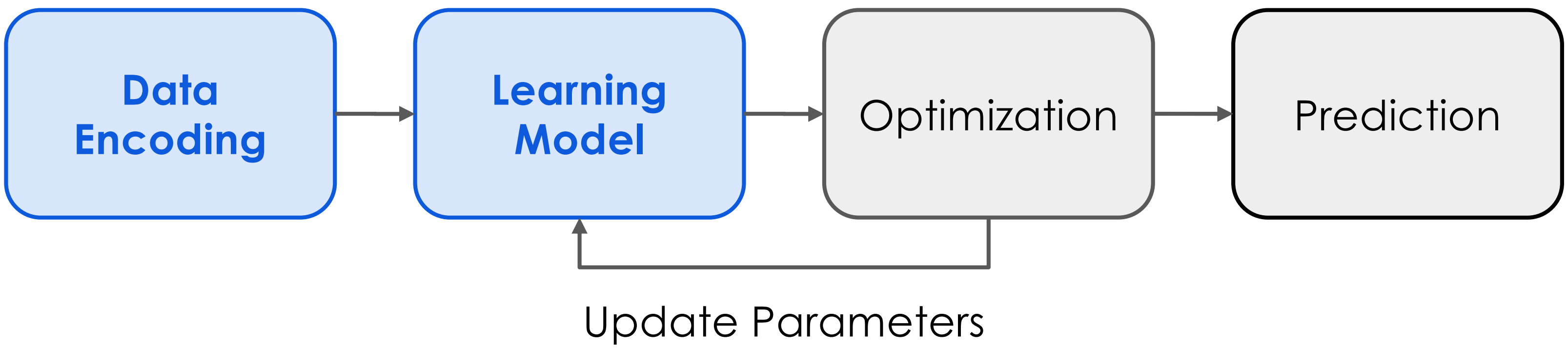
# QIML for Cybersecurity

**Classical ML Framework**



# QIML for Cybersecurity

### QI Modifications to ML Framework

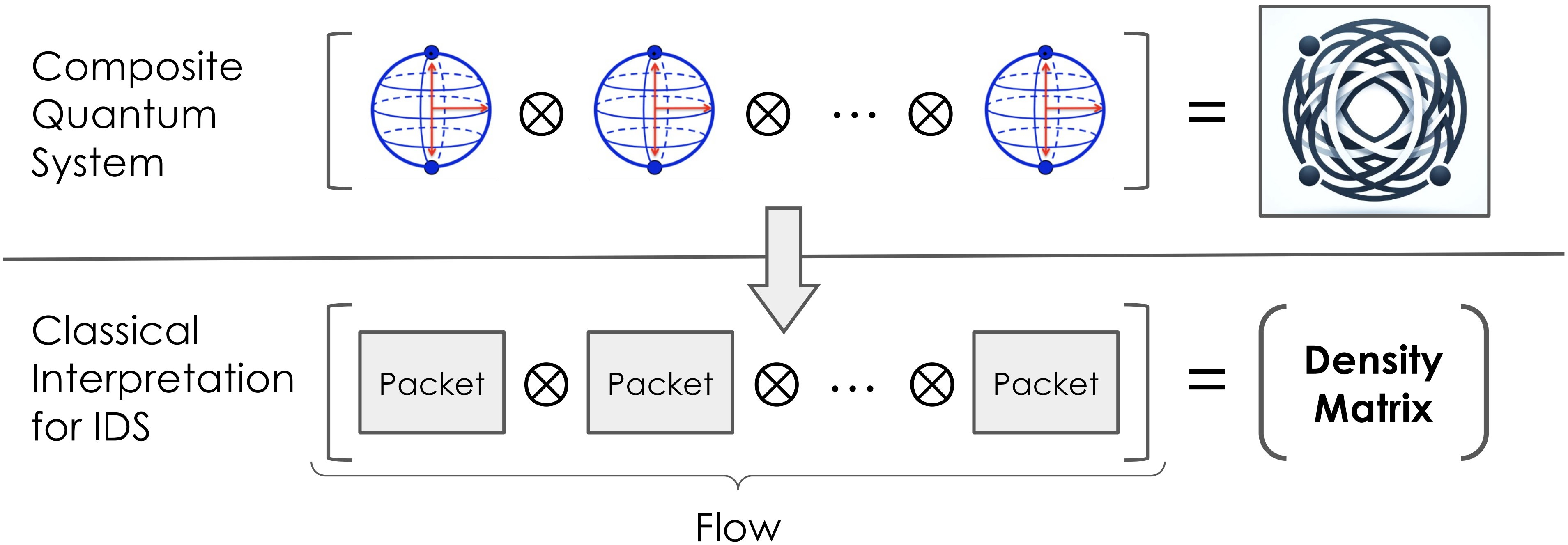



- **Density Matrix Encodings**
  - Class Separability
  - Enhanced performance
  - Adversarial robustness
- **Quantum Circuit Design**
  - Optimizing circuit models
  - Steeper convergence

# QIML: Density Matrix Encodings



- Model network packet flows as quantum systems



# QIML: Density Matrix Encodings

THE UNIVERSITY OF WESTERN AUSTRALIA

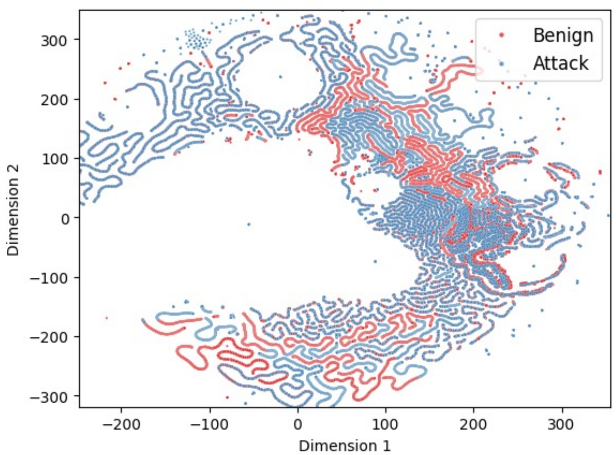
## 1. Greater class separability

- Can induce distinct clusters within data
- Simpler, more accurate ML classification

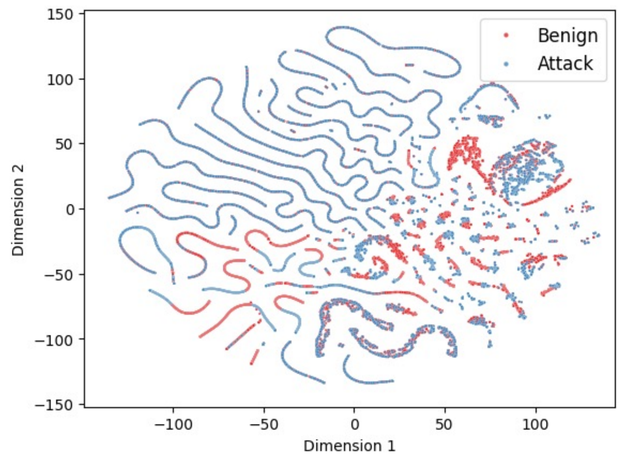
T-SNE Visualisations of the Mirai IDS dataset

Benign Label

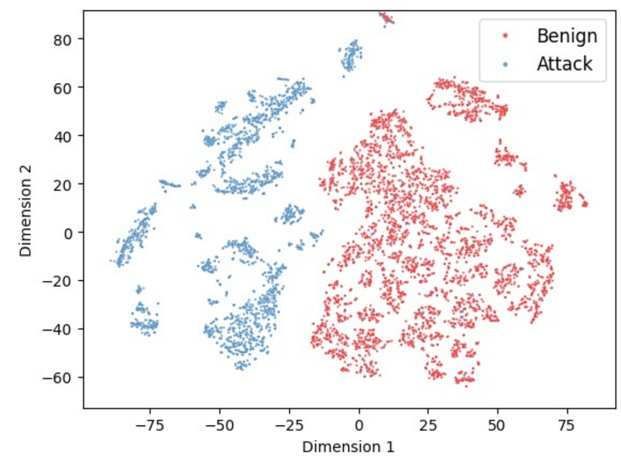
Attack Label




Raw IDS Data




Flow IDS Data




**DM Encoded**

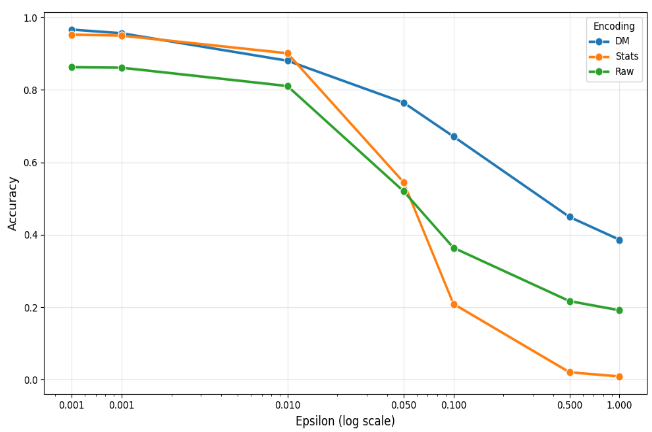
# QIML: Density Matrix Encodings

## 2. Strong performing, and adversarially robust

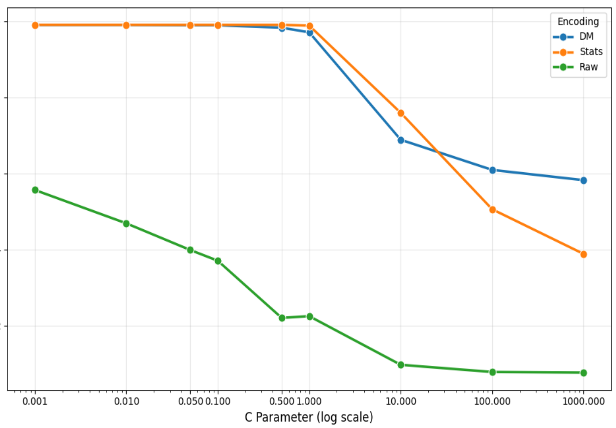
- Competitively high performance compared to common IDS representations.
- Maintains resilience to adversarial attacks; excels at high attack strength.

Accuracy vs. Attack Strength

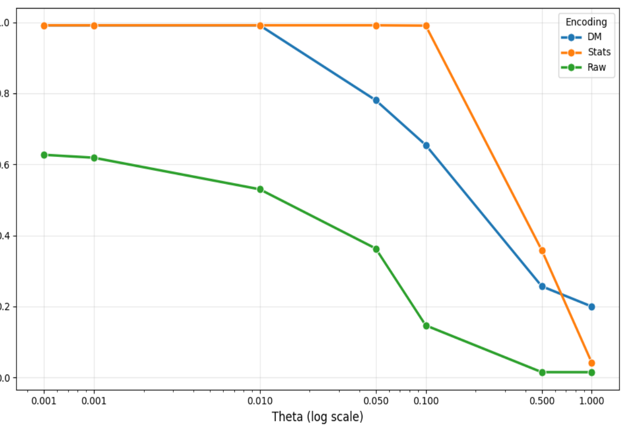
**DM Enc.** Stats. Enc. Raw




FGSM (ε)

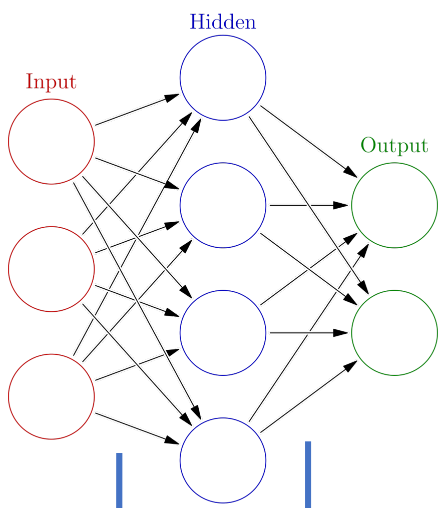



C&W (C)

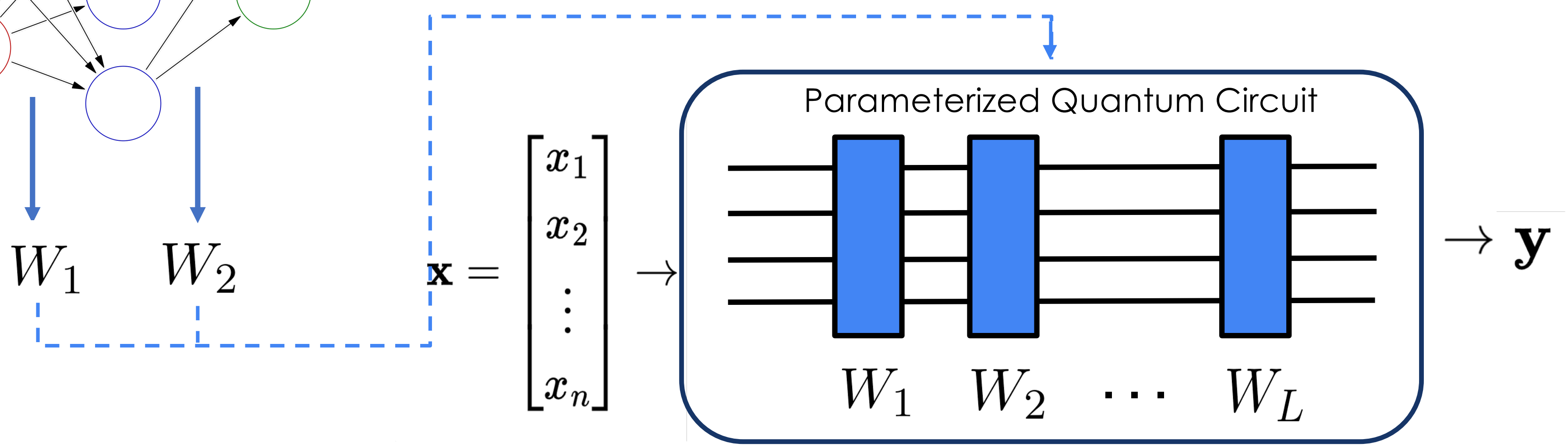



JSMA (θ)

# QIML: Weight-Informed Circuit Design




- Classical NN **weights and biases** represent a learned "pathway" from input to output.
- Leverage to inform quantum circuit (model) design:
  - Parameter initialization
  - Entanglement (correlation) structure

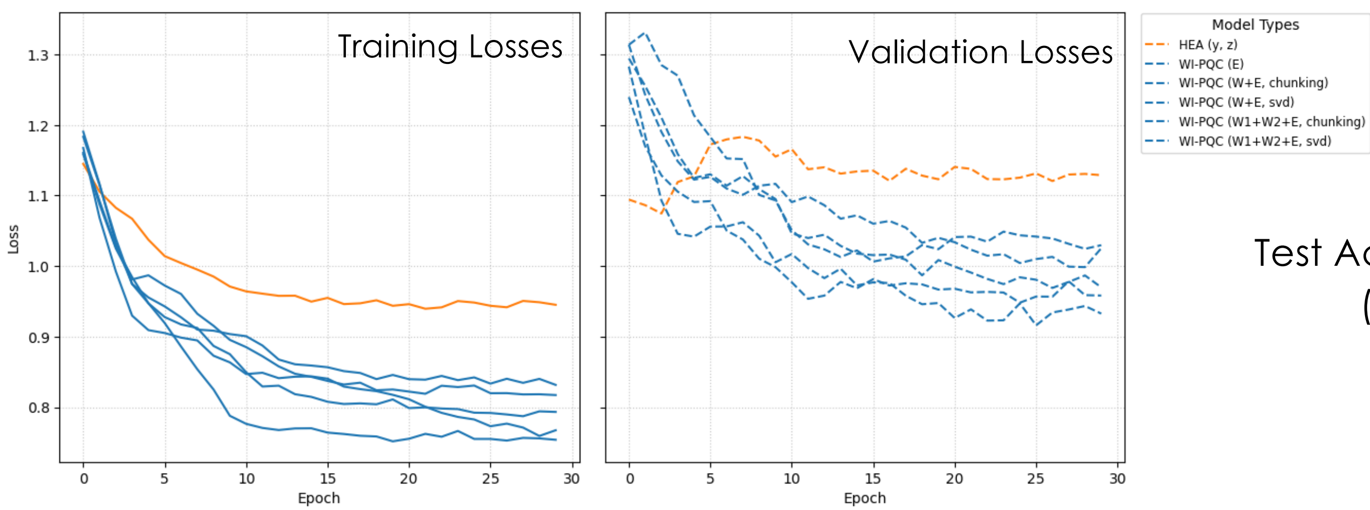


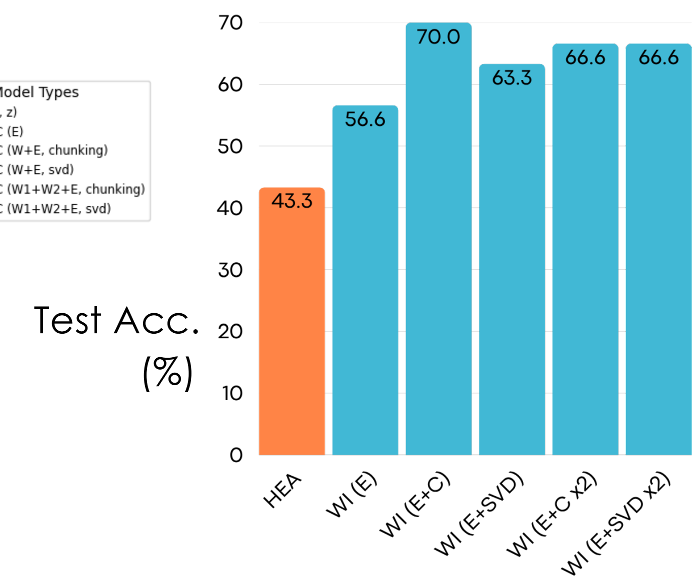
# QIML: Weight-Informed Circuit Design

## Good performance, with faster convergence

- Compared to best performing circuit architectures, e.g. Hardware-Efficient Ansatz (HEA).
- Consistent across various parameter init. and entanglement schemes

Iris Dataset (4 Qubits)

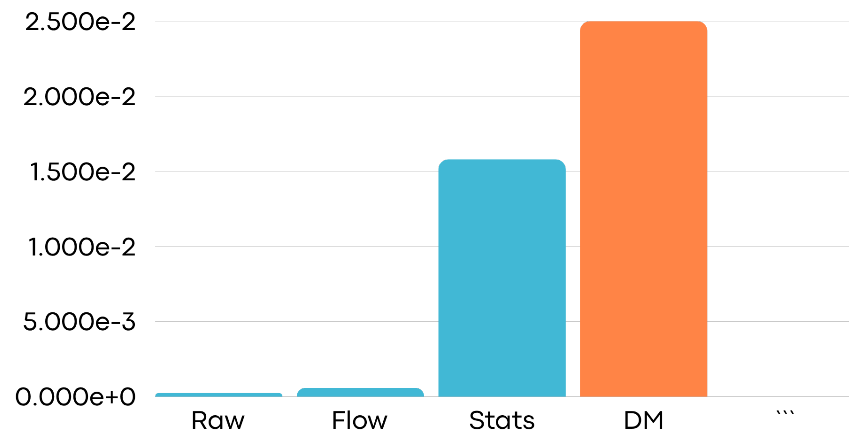


# Quantum-Inspired Computing: Drawbacks

- **Inefficiencies**
  - Density matrices require $O(2^n)$ memory, $O(nm^2)$ time (n = #datapoints, m = #features); classical-to-quantum transformation adds preprocessing cost.
  - Quantum circuits can be exponentially inefficient when run classically.
    - More expressivity mean more computation.
    - Barren plateaus: vanishing gradients across exponential parameter space.

Ave. Encoding Time per Instance for Mirai Dataset (s)



# Drawbacks

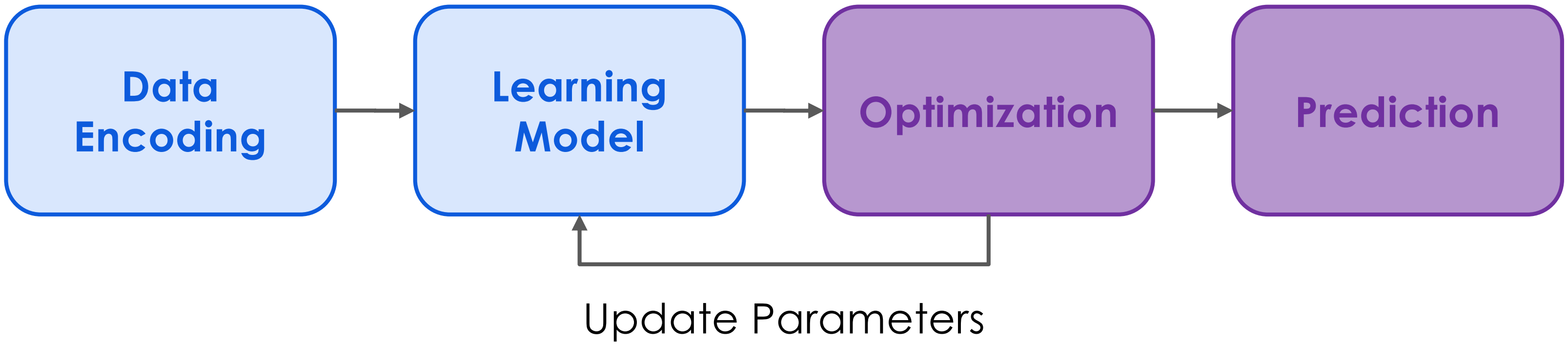
- **Inefficiencies**
  - Density matrices require $O(2^n)$ memory, $O(nm^2)$ time (n = #datapoints, m = #features); classical-to-quantum transformation adds preprocessing cost.
  - Quantum circuits can be exponentially inefficient when run classically.
    - More expressivity mean more computation.
    - Barren plateaus: vanishing gradients across exponential parameter space.
- **Performance Walls**
  - Classical methods still superior: Traditional ML consistently outperforms QIML on standard benchmarks.
  - Theoretical speedups rarely translate to practical gains on real problems.

Niches do exist; finding them is hard!

# Future Directions

THE UNIVERSITY OF WESTERN AUSTRALIA

**QI Modifications to ML Framework**

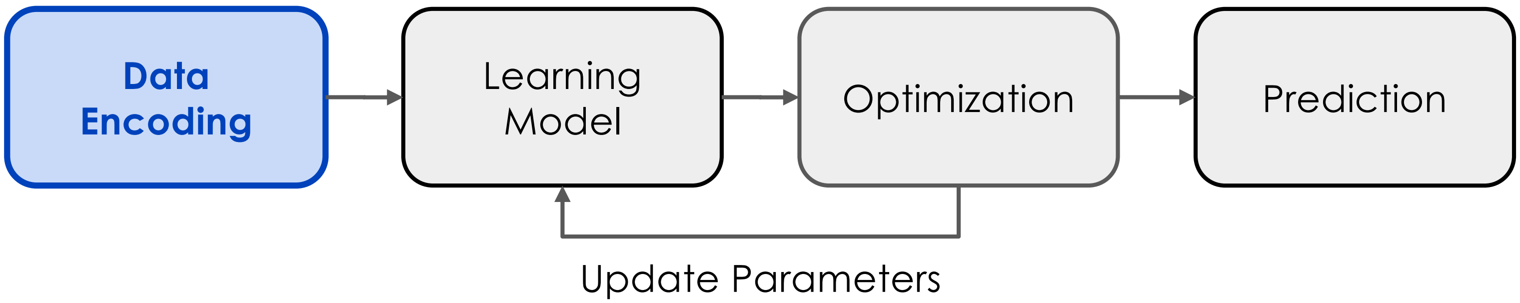



- Quantum-inspired **optimizers** that help alleviate barren plateaus.
- Enhancing **prediction** routines with superposition-based uncertainty quantification.
- **Further solidifying current understanding**;
  - black-box adversarial scenarios for DMs,
  - fine-tuning circuit optimization pipeline.

THE UNIVERSITY OF WESTERN AUSTRALIA

Q&A

# What does QiML Look Like?



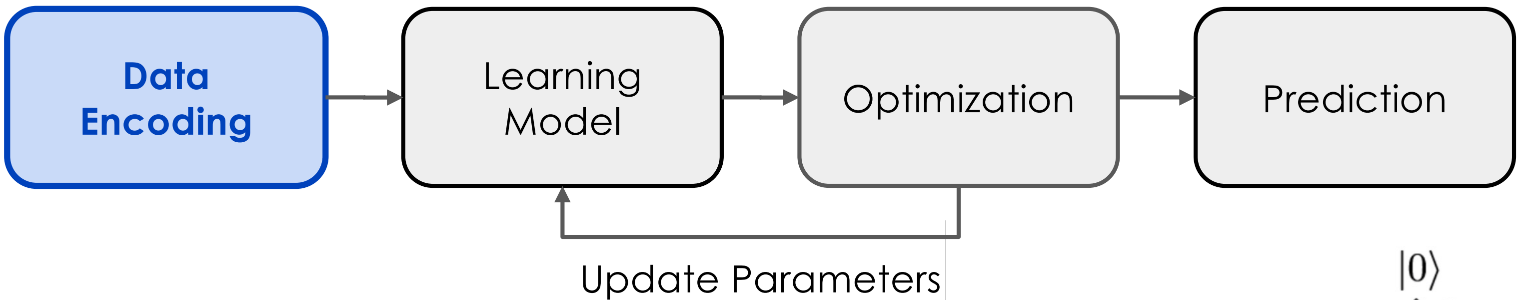

**Classical Feature Vectors**

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

**QiML Feature Vectors**

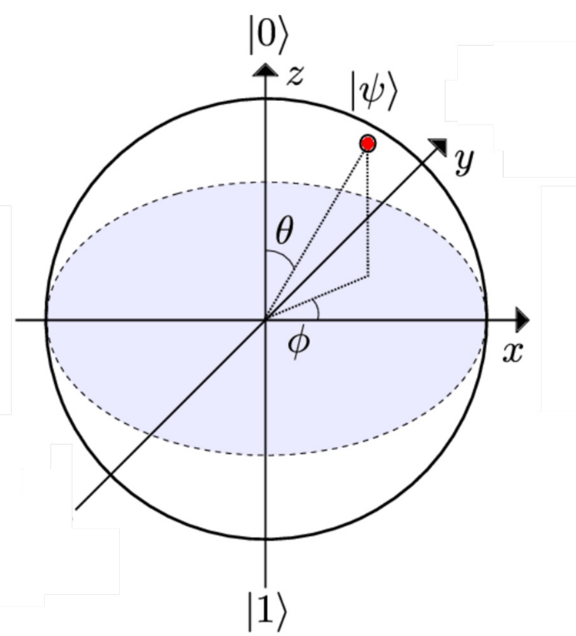
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \rightarrow E_{(\alpha_1, \ldots, \alpha_m)}(\mathbf{x}) \rightarrow |\psi\rangle$$

# What does QiML Look Like?



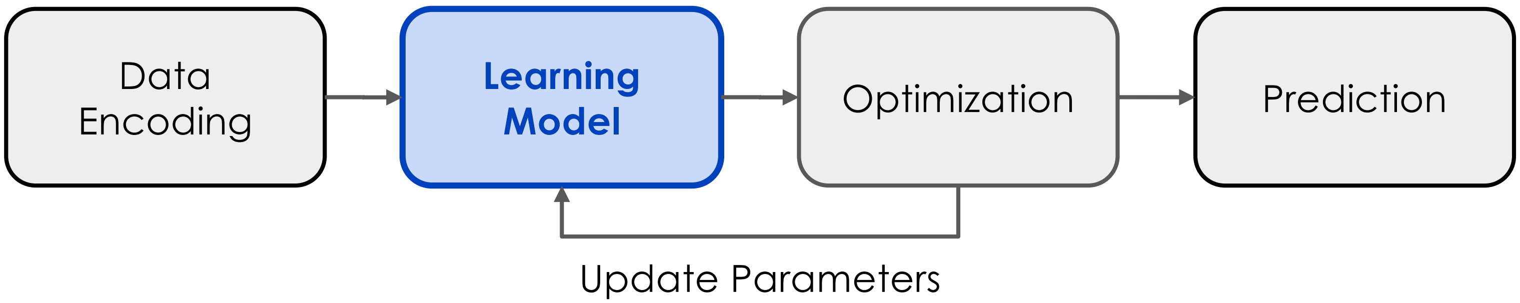

**QiML Feature Vectors**

$$E_{(\alpha_1,\ldots,\alpha_m)}(\mathbf{x}) = \begin{bmatrix} R(\alpha_1) & R(\alpha_2) & \cdots & R(\alpha_m) \\ R(\alpha_1) & R(\alpha_2) & \cdots & R(\alpha_m) \\ \vdots & \vdots & \ddots & \vdots \\ R(\alpha_1) & R(\alpha_2) & \cdots & R(\alpha_m) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$



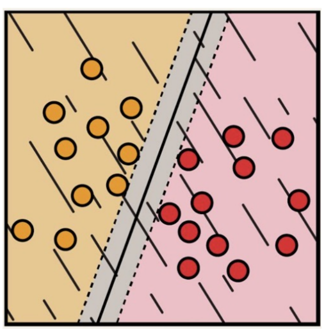
# What does QiML Look Like?



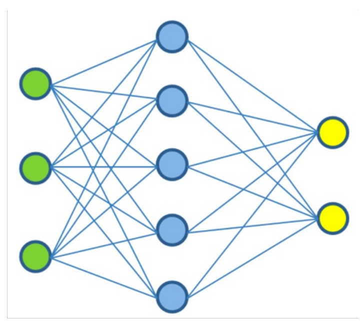



**Classical models used…**

SVM



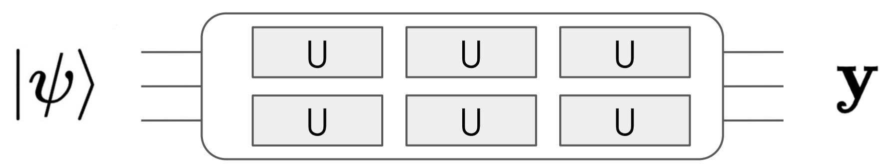
Neural Network



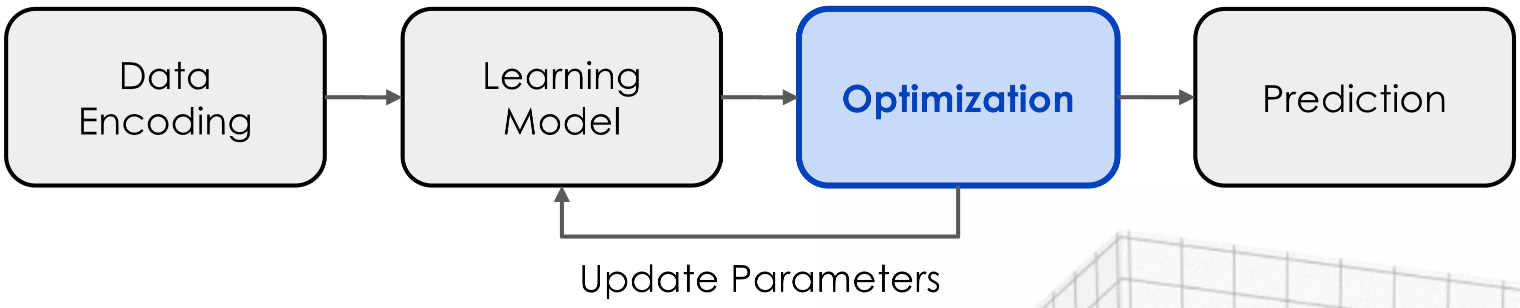
**…and also QiML-specific models**

Decompositons

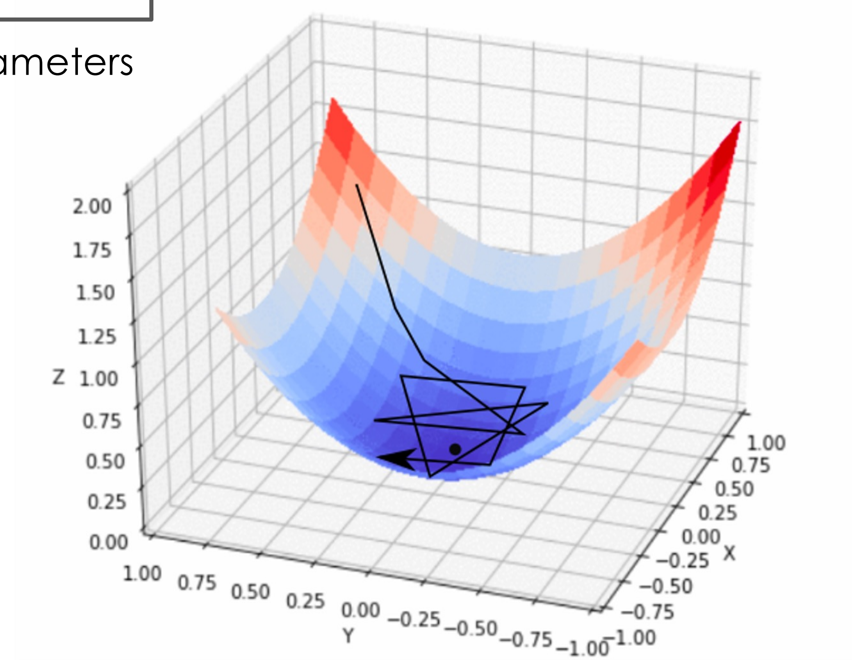

Circuit Learning



# What does QiML Look Like?

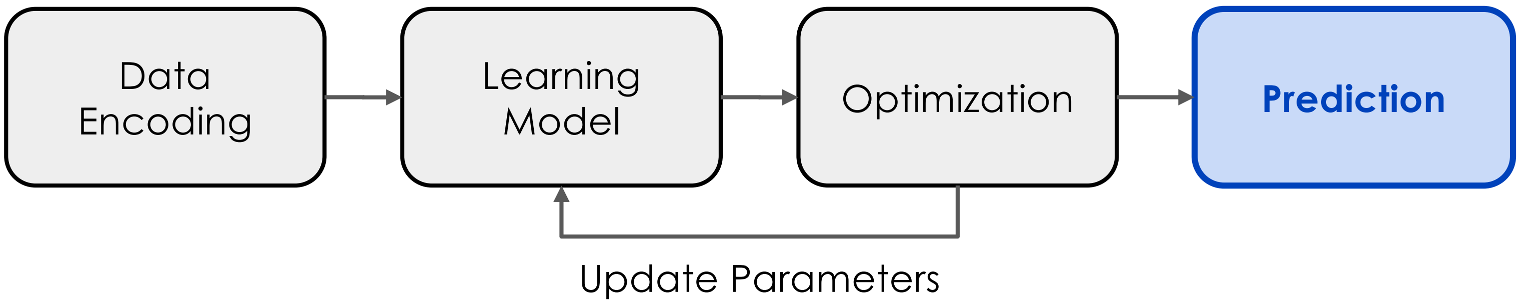



Classical **gradient descent** is still the main optimization method

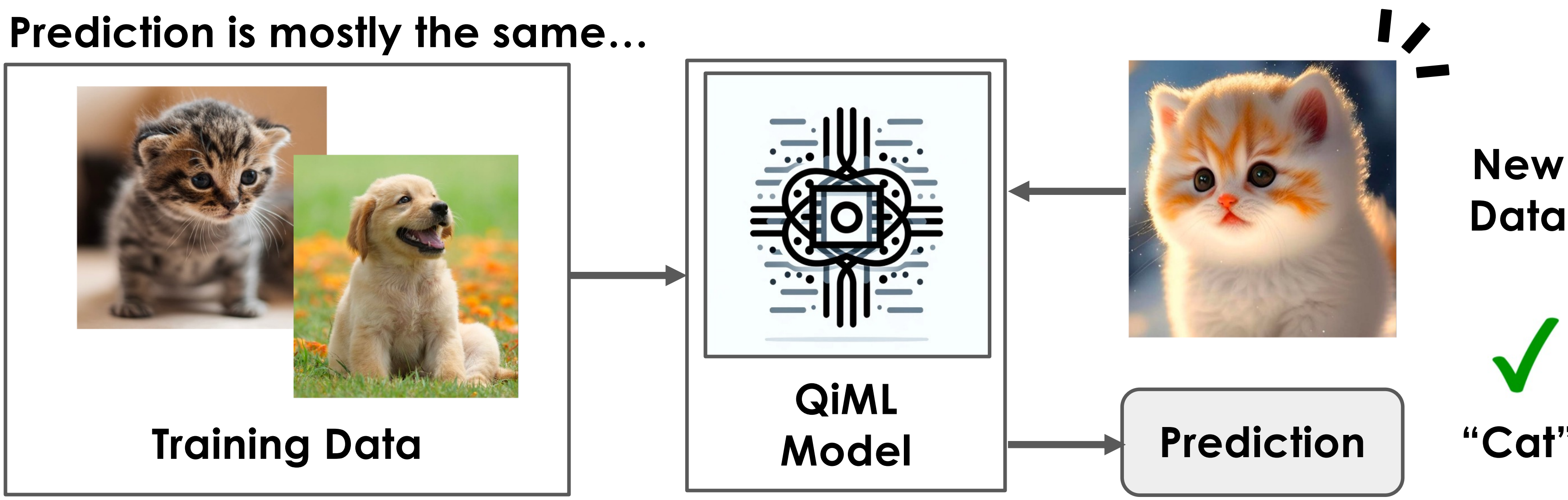


# What does QiML Look Like?



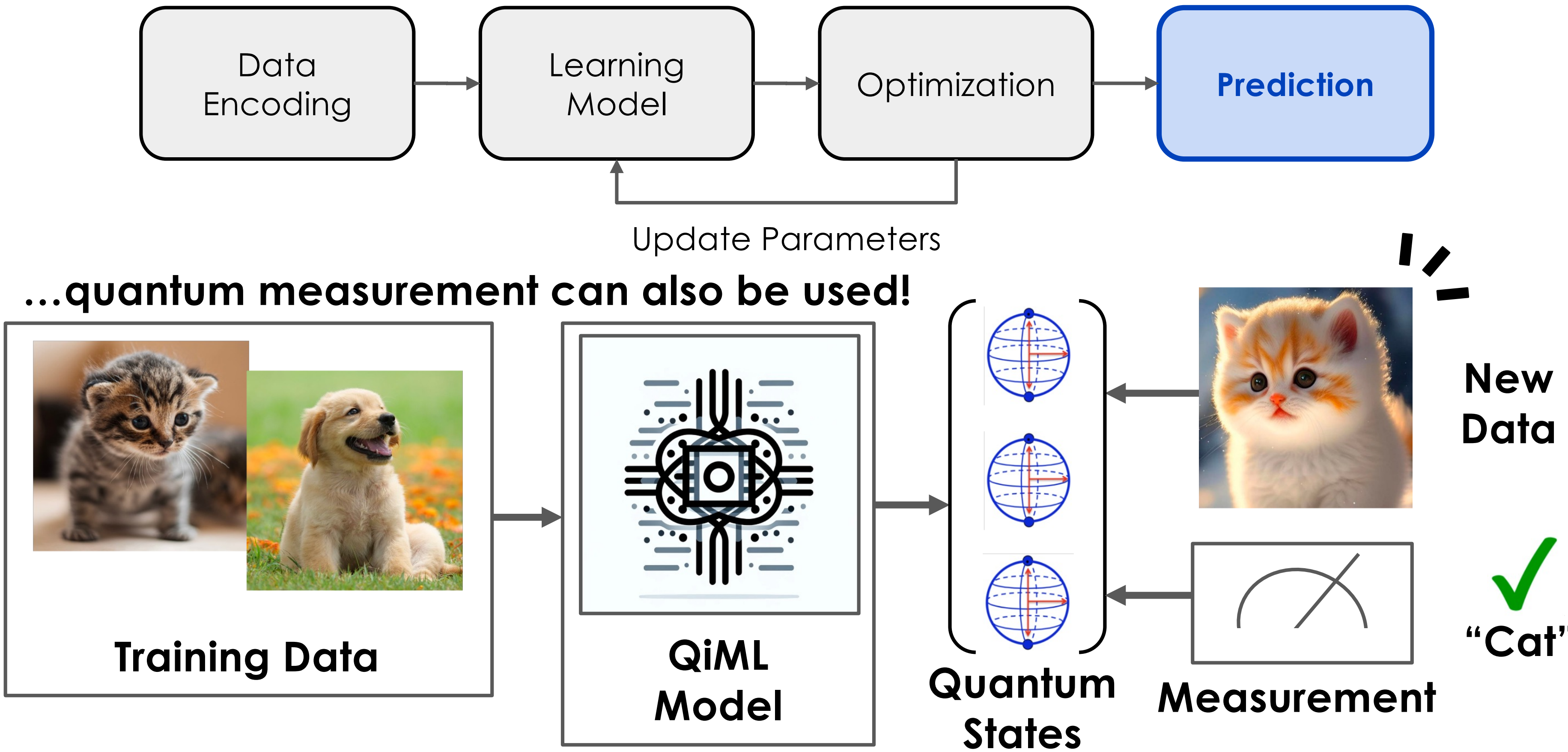



**Prediction is mostly the same…**



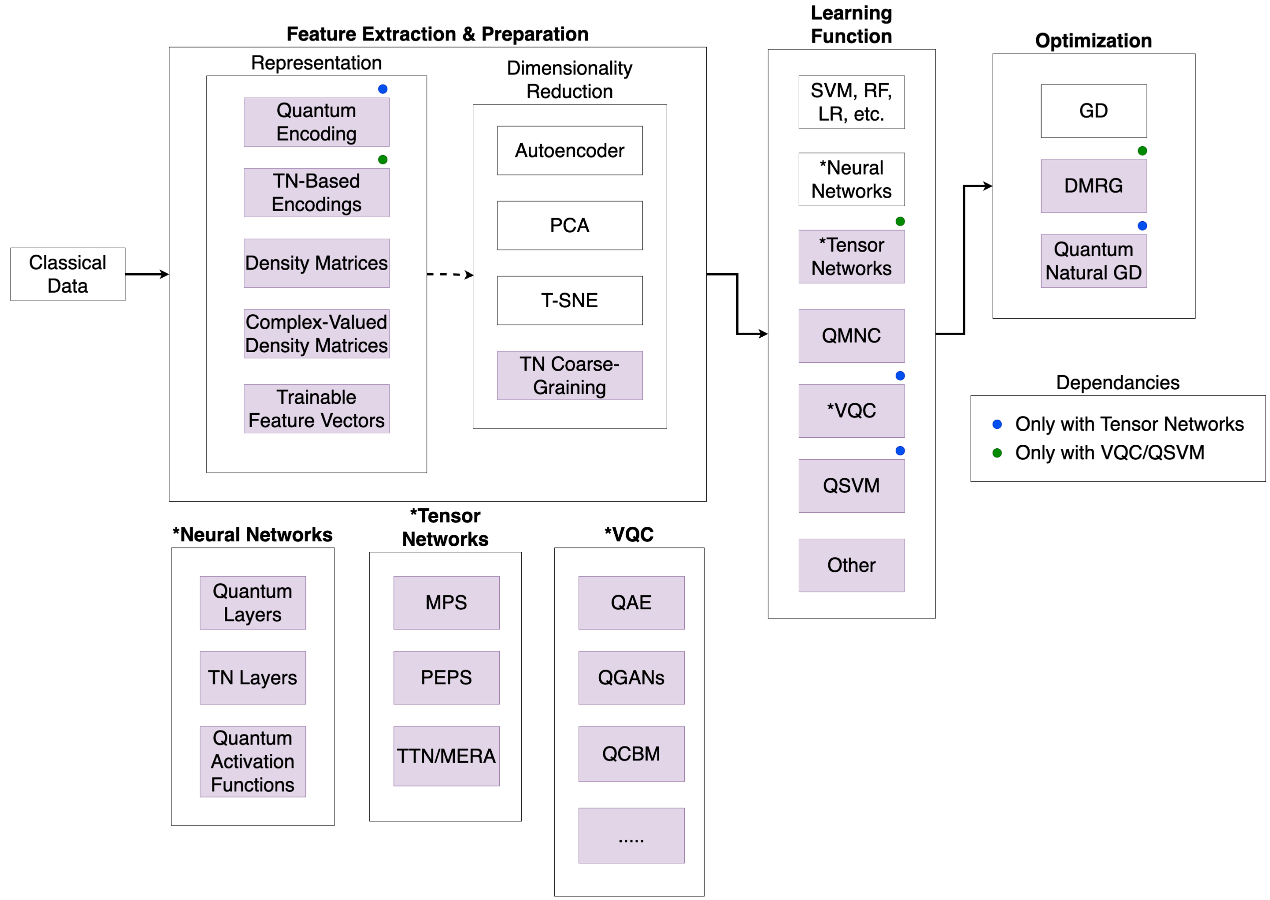
# What does QiML Look Like?

THE UNIVERSITY OF WESTERN AUSTRALIA



# QiML IDS: Previous Works

## Representation



# QiML IDS: Previous Works

THE UNIVERSITY OF WESTERN AUSTRALIA

- Very few works using QIML for IDS [1-3]
  - Limited methodologies: only explore quantum circuit learning methods
  - Their results and decisions are not well explained.
  - Performance and training times are same or worse than classical methods.

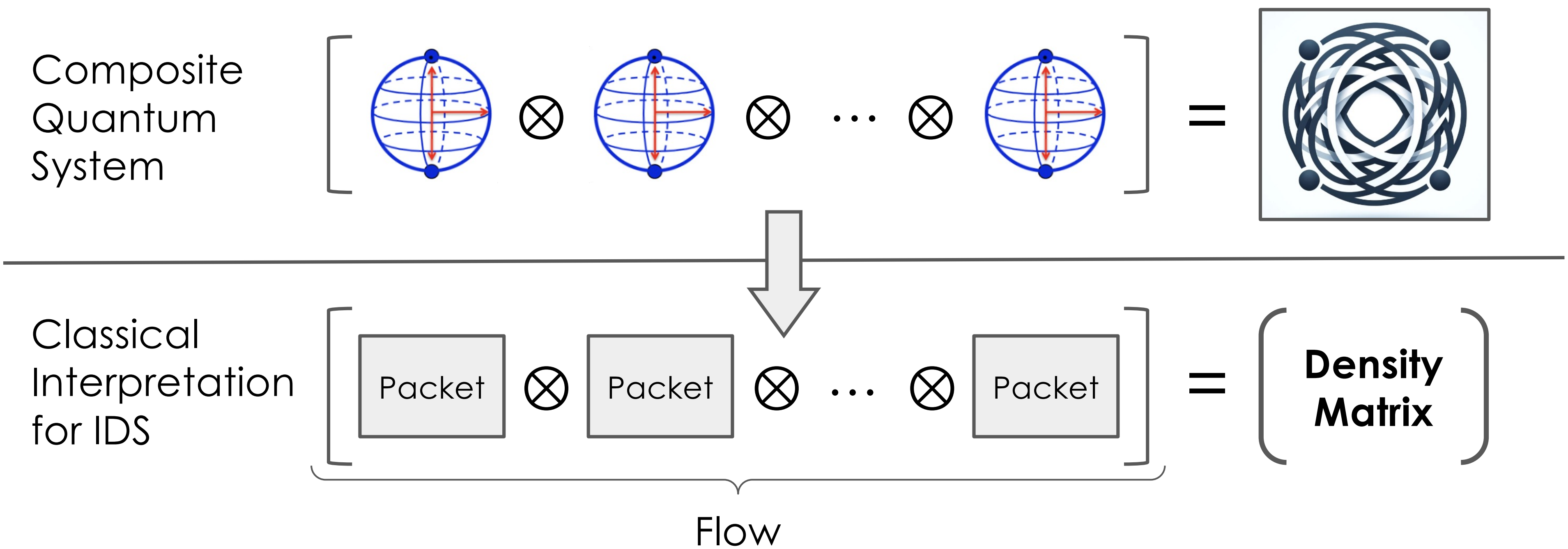
# QiML IDS: Previous Works

- **IDS wants to:**
  - Quickly and accurately detect attacks;
  - Detect new, unseen attacks early; and
  - Handle high throughput network traffic

- **We want to explore:**
  - How can QIML enhance IDS?
  - What sorts of QIML methods can apply to IDS?

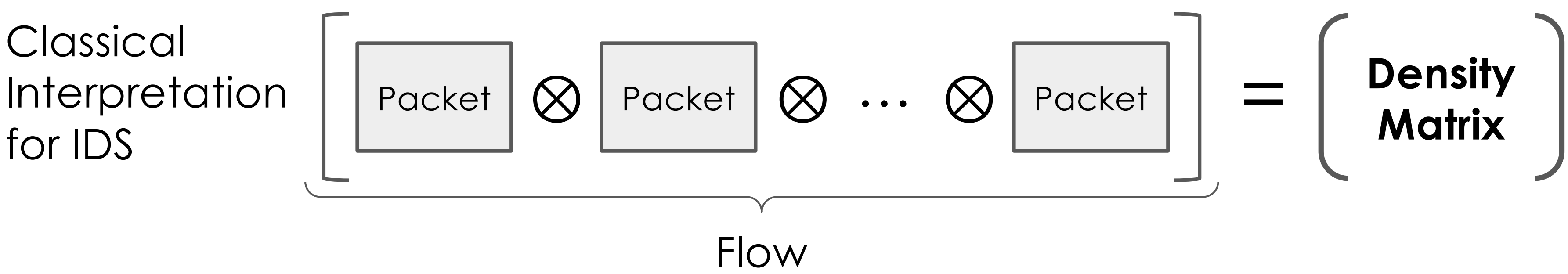
# QiML IDS: Density Matrices

- **Start with the Encodings:**
  - Think of packets within flows as a quantum system?

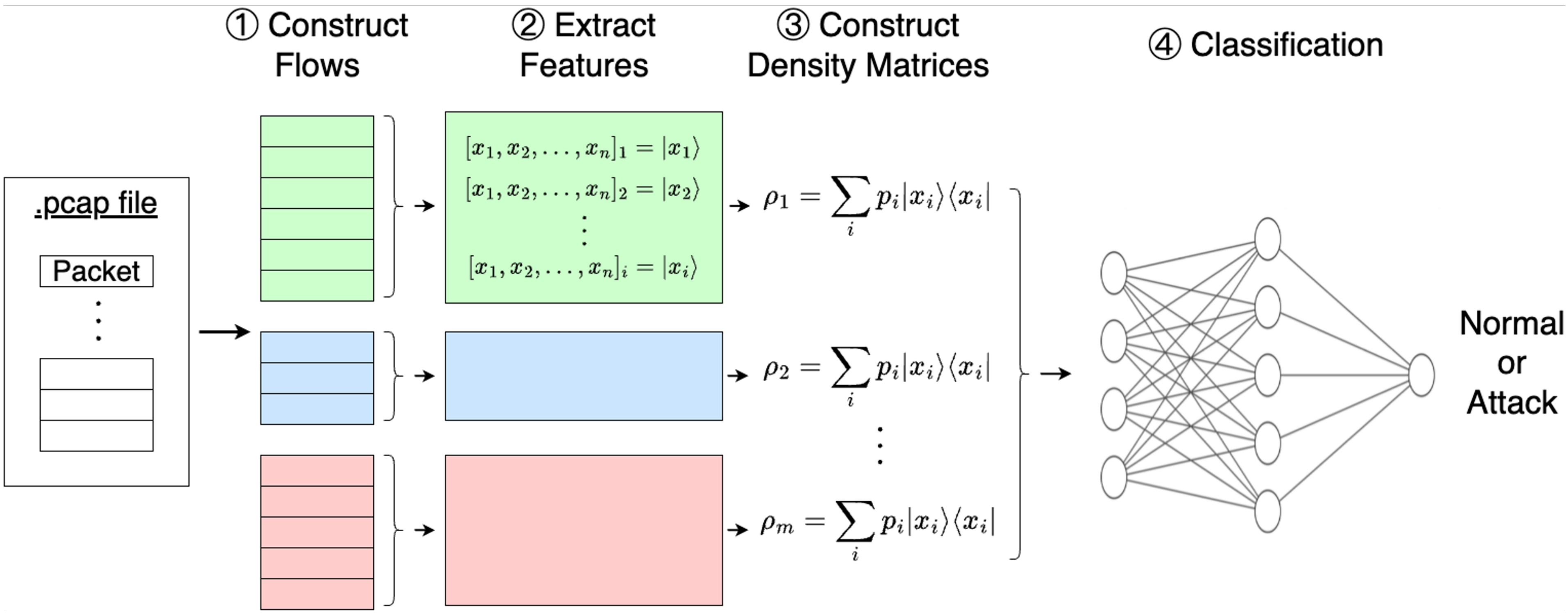


# QiML IDS: Density Matrices

- **Start with the Encodings:**
  - Think of packets within flows as a quantum system?
- **Density Matrices:**
  - Represent packet data as a mixture of outcomes, based on some probability
  - Capture correlations between packets within flows

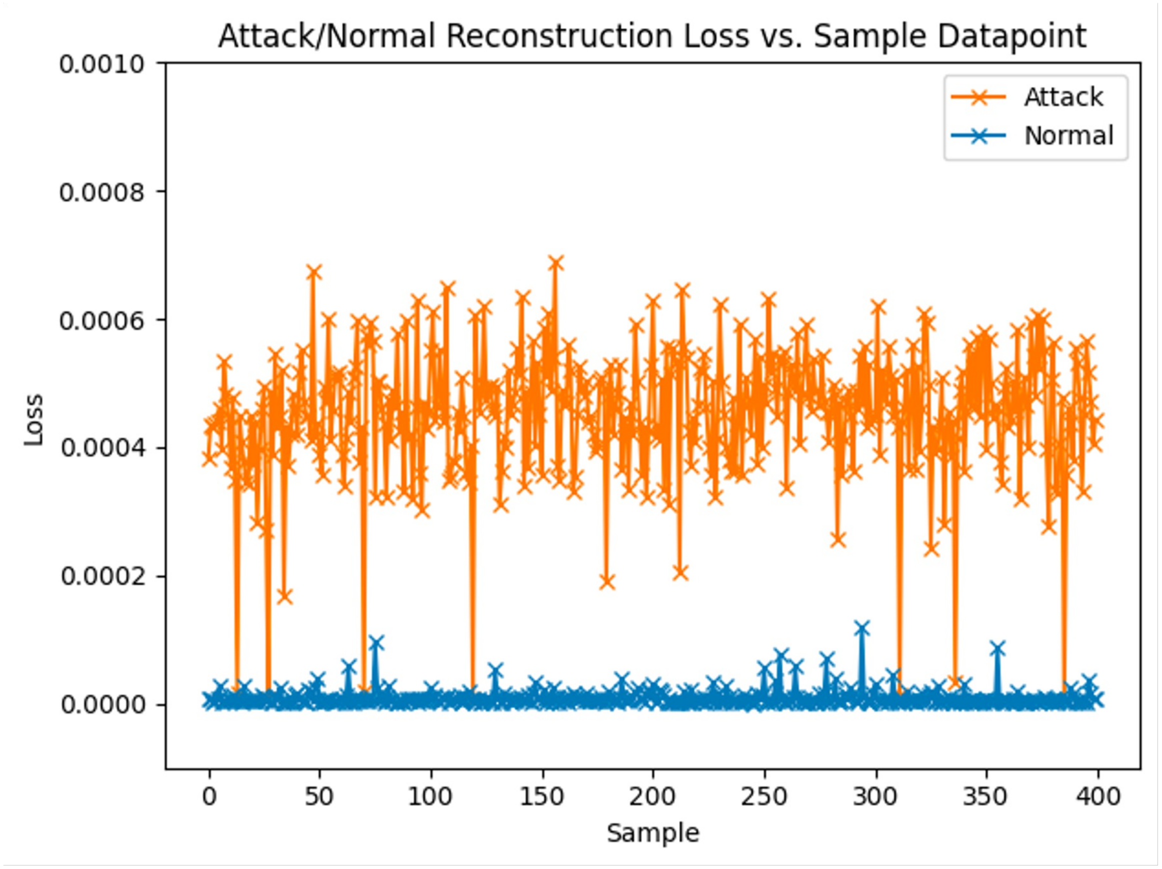


# QiML IDS: Density Matrices



# QiML IDS: Results

- Good performance
  - F1 Score: 98.35%
  - AUC > 0.99
- Improvements over packet-based IDS in:
  - Performance
  - Training time



# QiML IDS: Further Research

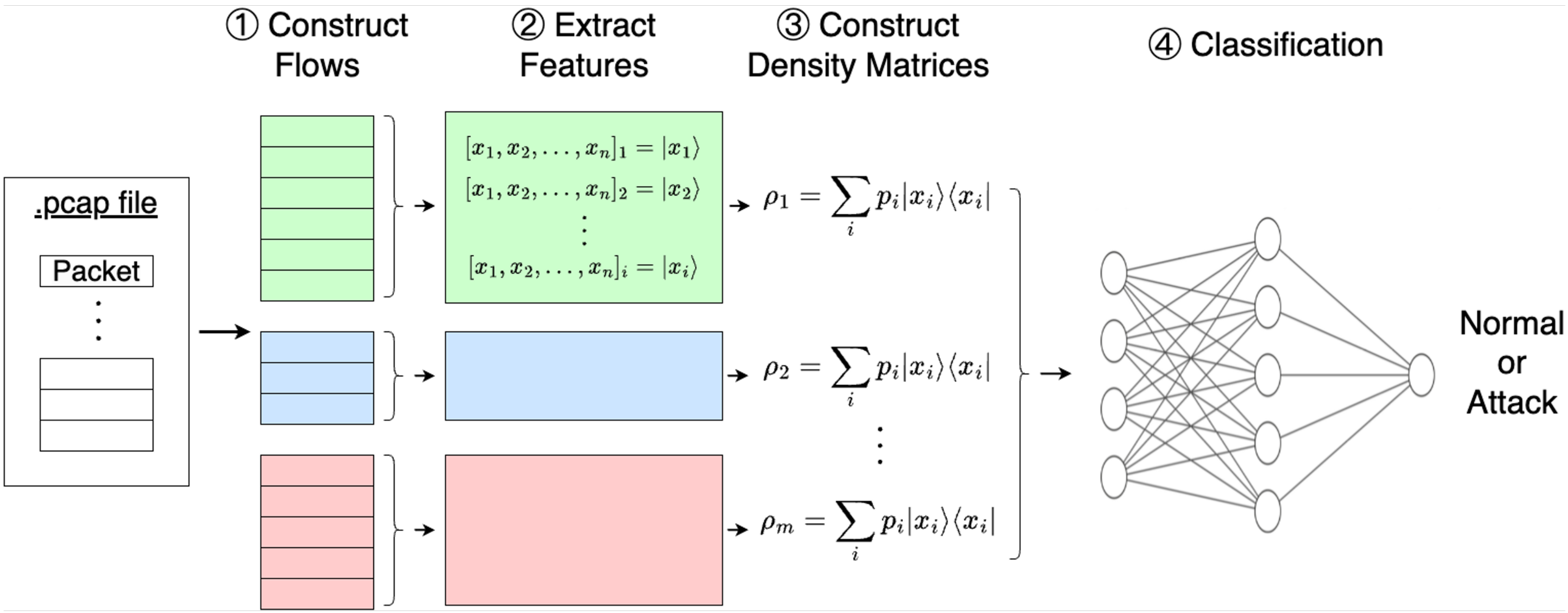
- Still in early stages, many things to explore:
  - Comparison against flow-based IDS
  - Better understand the effect of inducing correlations between packets within flows
  - Further exploit introduced quantum aspects
  - Explore additional encoding and learning methods

# References

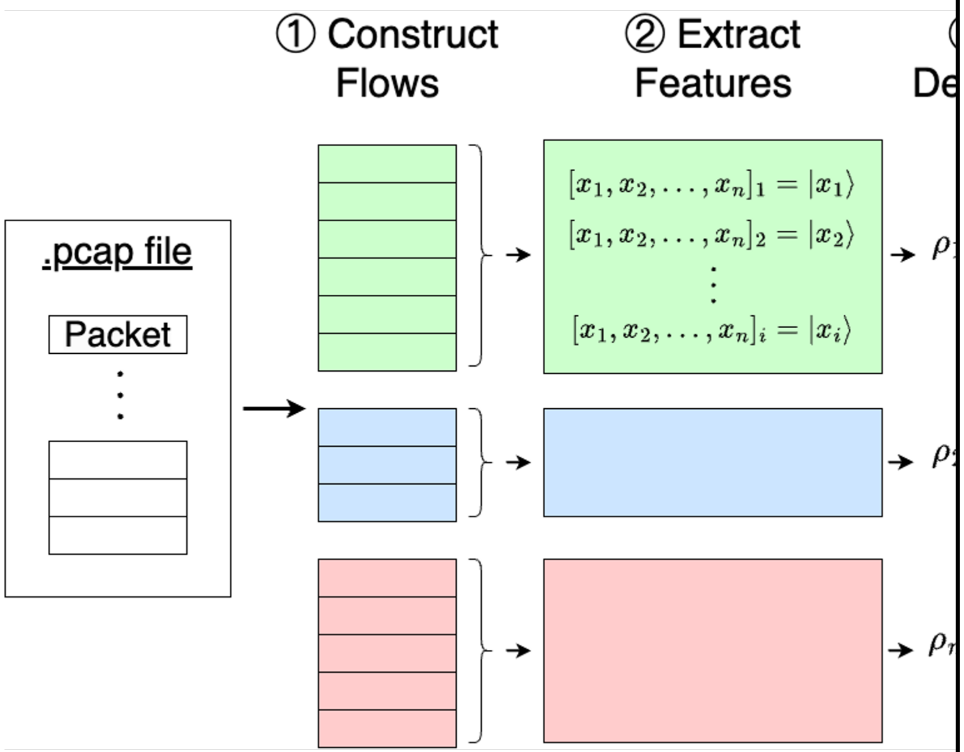

1. Payares, E. D., & Martínez-Santos, J. C. (2021). Quantum machine learning for intrusion detection of distributed denial of service attacks: a comparative overview. Quantum Computing, Communication, and Simulation, 11699, 35-43.
2. Laxminarayana, N., Mishra, N., Tiwari, P., Garg, S., Behera, B. K., & Farouk, A. (2022). Quantum-assisted activation for supervised learning in healthcare-based intrusion detection systems. IEEE Transactions on Artificial Intelligence.
3. Gouveia, A., & Correia, M. (2020, November). Towards quantum-enhanced machine learning for network intrusion detection. In 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA) (pp. 1-8). IEEE.

# Appendix

# Quantum-Inspired IDS: Density Matrices

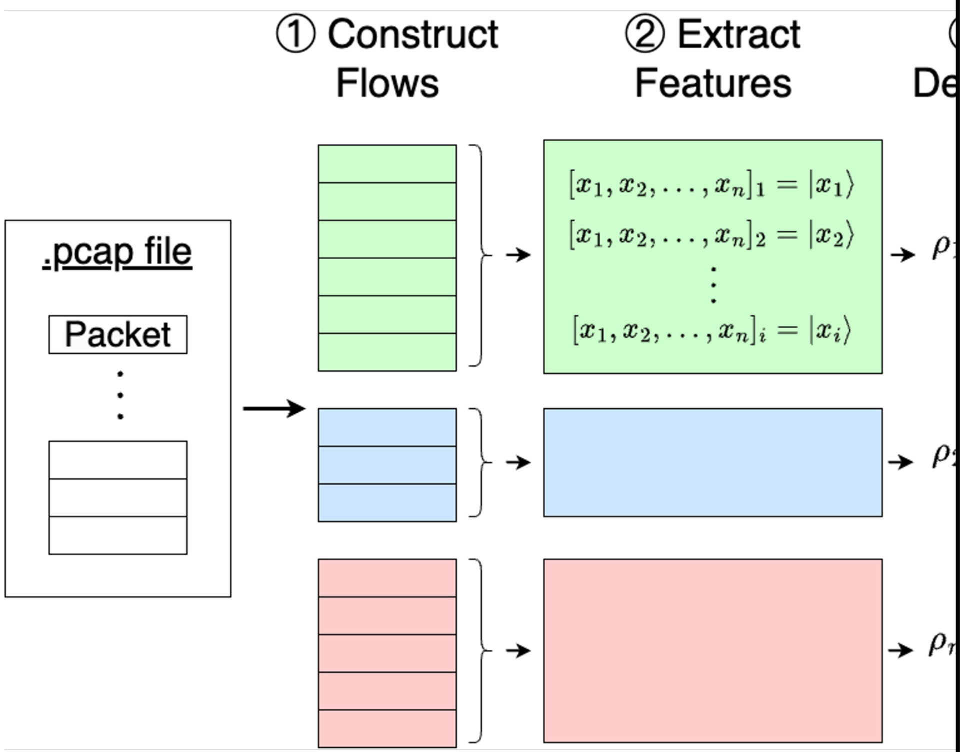


# Quantum-Inspired IDS: Density Matrices




- **① How is a "Flow" Determined?**
  - Packets-per-time interval (10ms, 1sec, 10sec, etc.)
  - Packets-per-communication channel (from source to dest.)

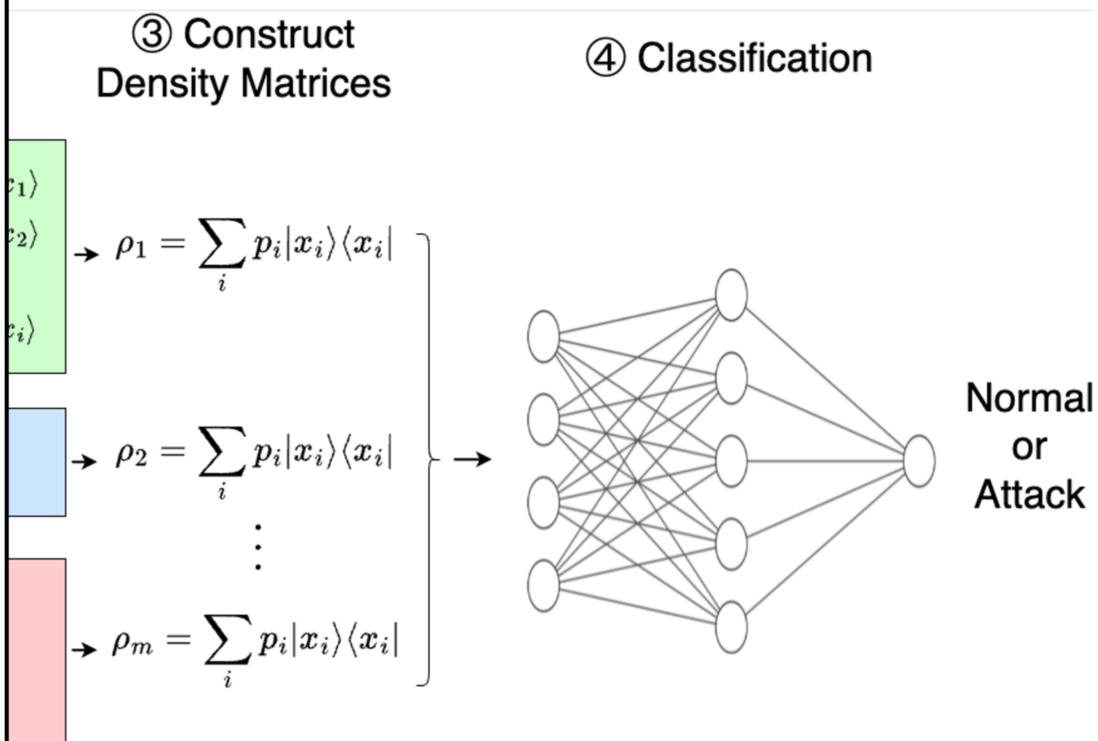
# Quantum-Inspired IDS: Density Matrices




- **② What Features are Used?**
  - Packet Header information (IP src/dst, port src/dst, TCP/UDP information, etc.)
  - First N-bytes also commonly used: to be explored

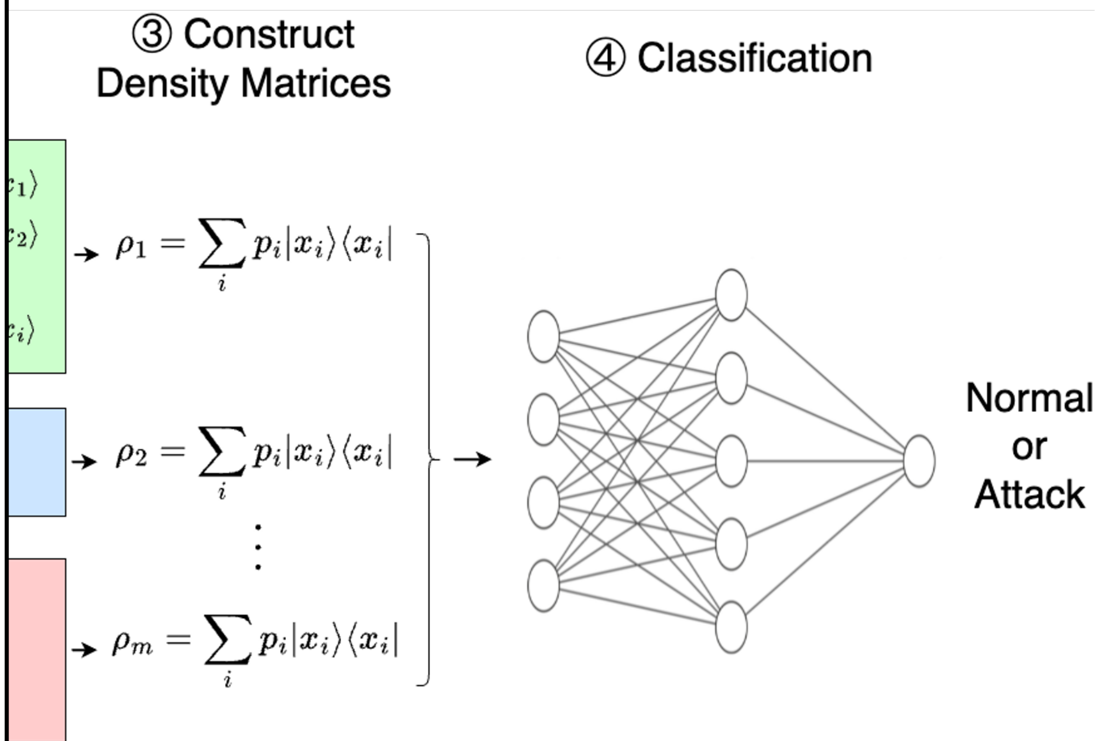
# Quantum-Inspired IDS: Density Matrices

- **③ What are the Probabilities $p_i$?**
  - Based on local and global protocol frequency
  - Several other viable choices: to be explored.

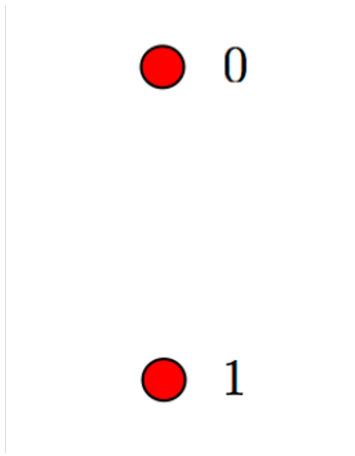


# Quantum-Inspired IDS: Density Matrices

- ④ **What is the Learning Model?**
  - Neural network autoencoder
  - Several other viable choices: to be explored.

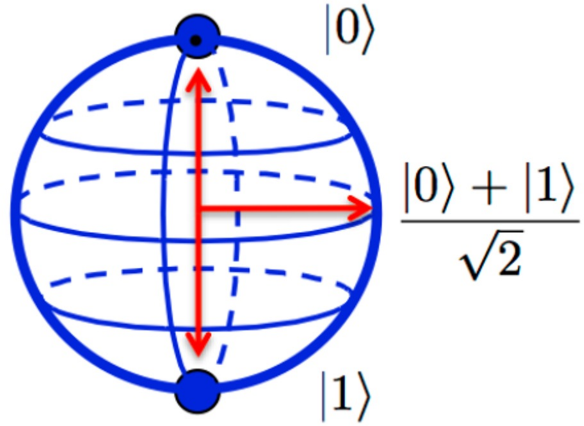


# Background: Qubits




**Classical Bit:**

0 or 1



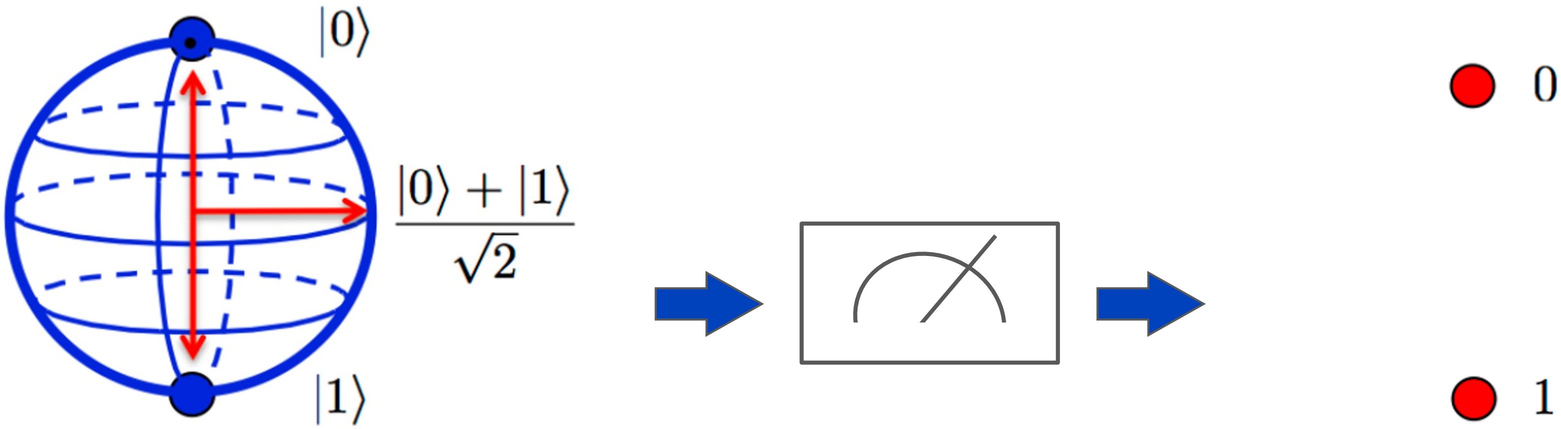

**Quantum Bit (Qubit):**

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

[2]

# Background: Measurement





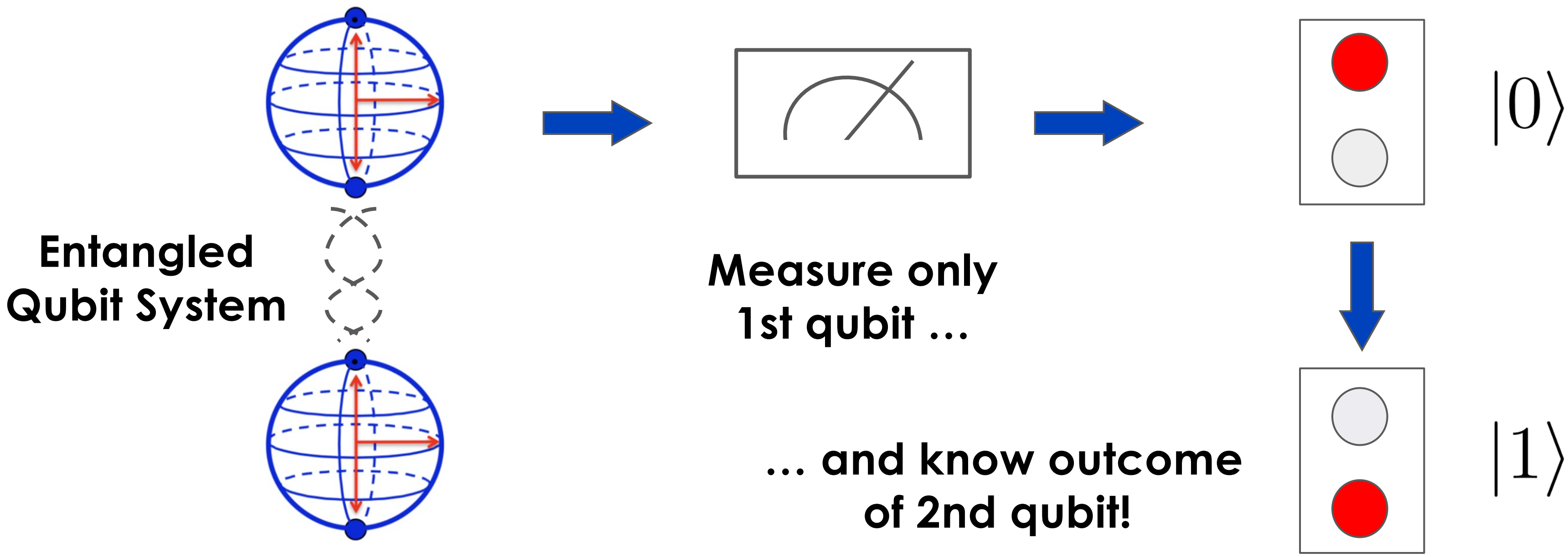

**Qubit:**

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

**Measurement**

**Classical Information**

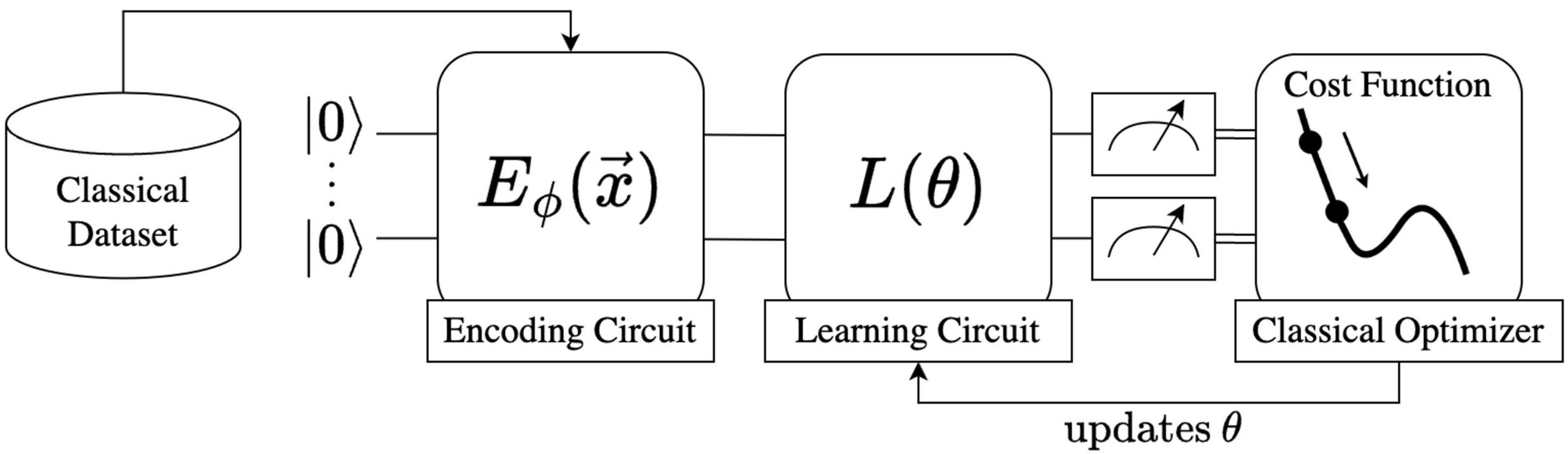
# Background: Entanglement




- Entanglement represents the correlation between qubits in a system.
- Measurement on one part of the system can give information about other parts.

# QiML IDS: Previous Works

- Very few works using QIML for IDS [1-3]:
  - They only use simulated quantum circuit learning algorithms

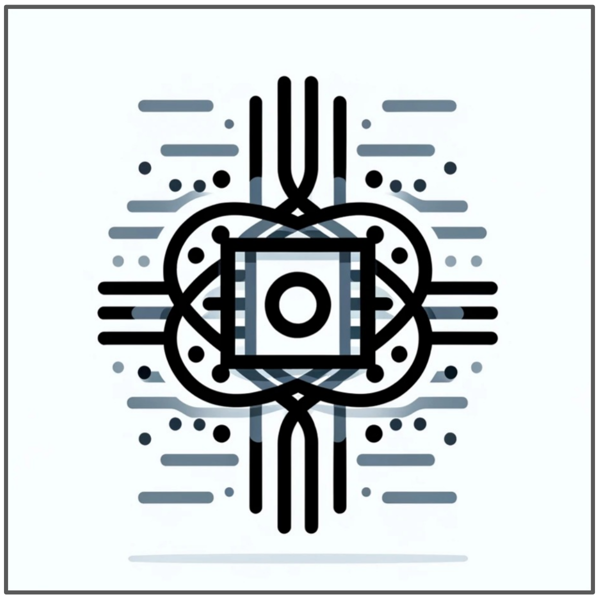


# QiML IDS: Previous Works

- **However!**
  - Their results and decisions are not well explained.
  - Performance and training times are same or worse than classical methods.
  - Many other QIML, and encoding methods exist.

# Quantum-Inspired Computing

THE UNIVERSITY OF WESTERN AUSTRALIA



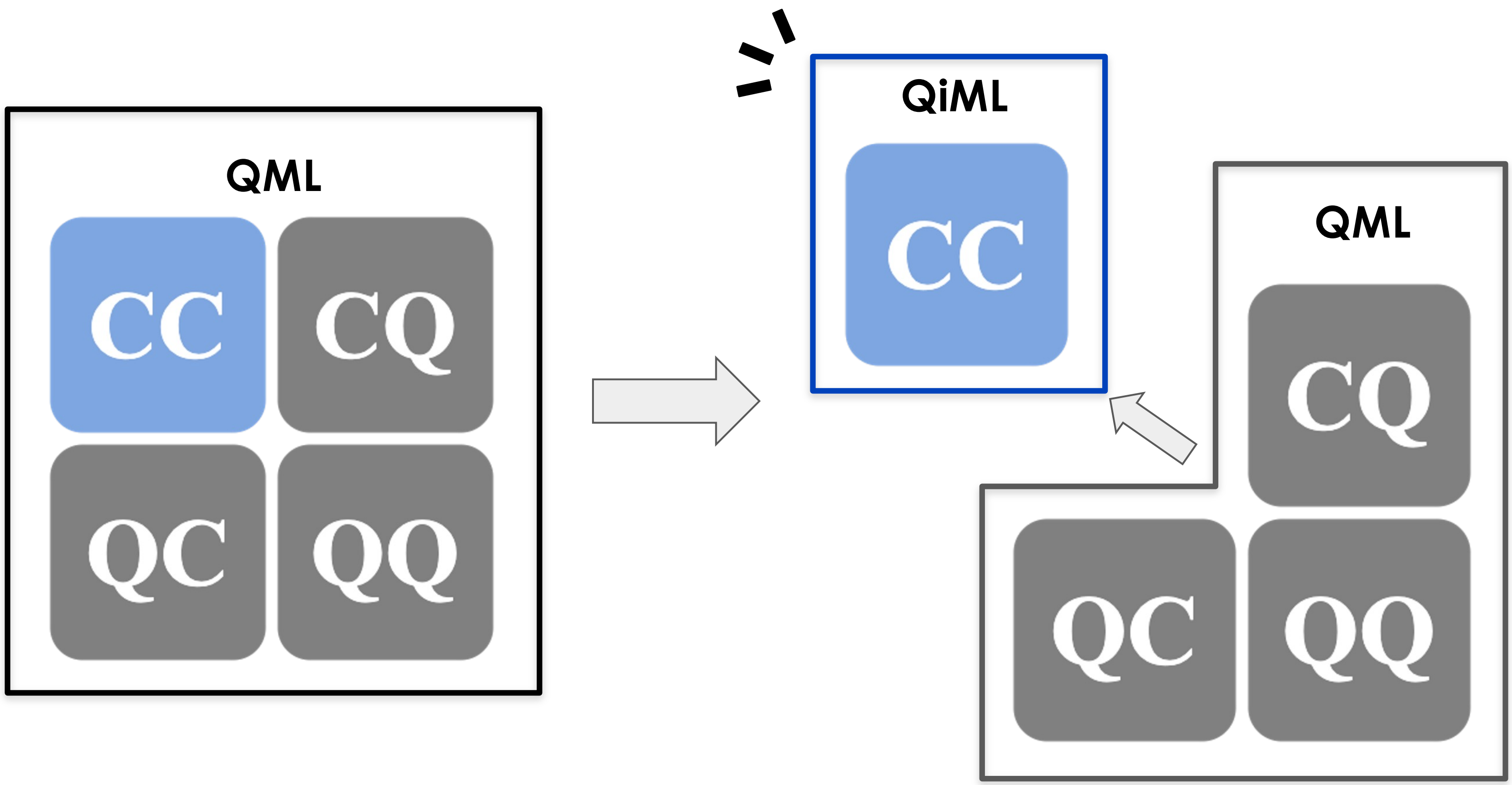
**Methods in:**

- Optimization
- Search Algorithms
- **Machine Learning**
- ...

**Applications in:**

- Finance
- Medicine
- **Cybersecurity**
- ...

# What is Quantum-Inspired Machine Learning (QiML)?



# Quantum-Inspired Computing Methods

1. Tensor Network-based Learning Methods
2. Quantum Variational Algorithm Simulation
3. Other QiML Methods
4. Dequantized Algorithms

# QiML Methods

1. **Tensor Network-based Learning Methods**
2. Dequantized Algorithms
3. Quantum Variational Algorithm Simulation
4. Other QiML Methods

# QiML Methods: Tensor Networks

- Quantum wavefunction $|\psi\rangle$ = big tensor
- Scales **exponentially** with number of qubits
- Decompose as a **tensor network**
- Now scales **linearly** with qubits!

$$|\psi\rangle = T \approx t_1 \otimes t_2 \otimes \cdots t_N$$

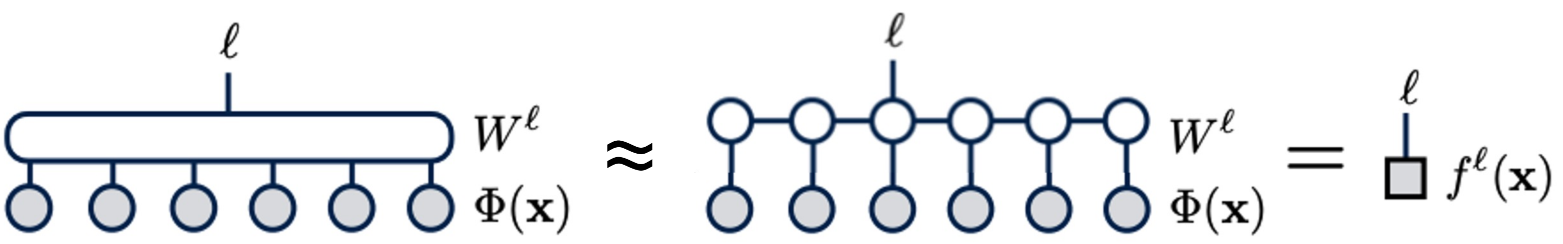
[3]

# QiML Methods: Tensor Networks

- **Supervised Learning:**
  - Treat the weight tensor *W* as a wavefunction, and decompose as a tensor network!

$$f^l(x) = W^l \cdot \Phi(\mathbf{x})$$

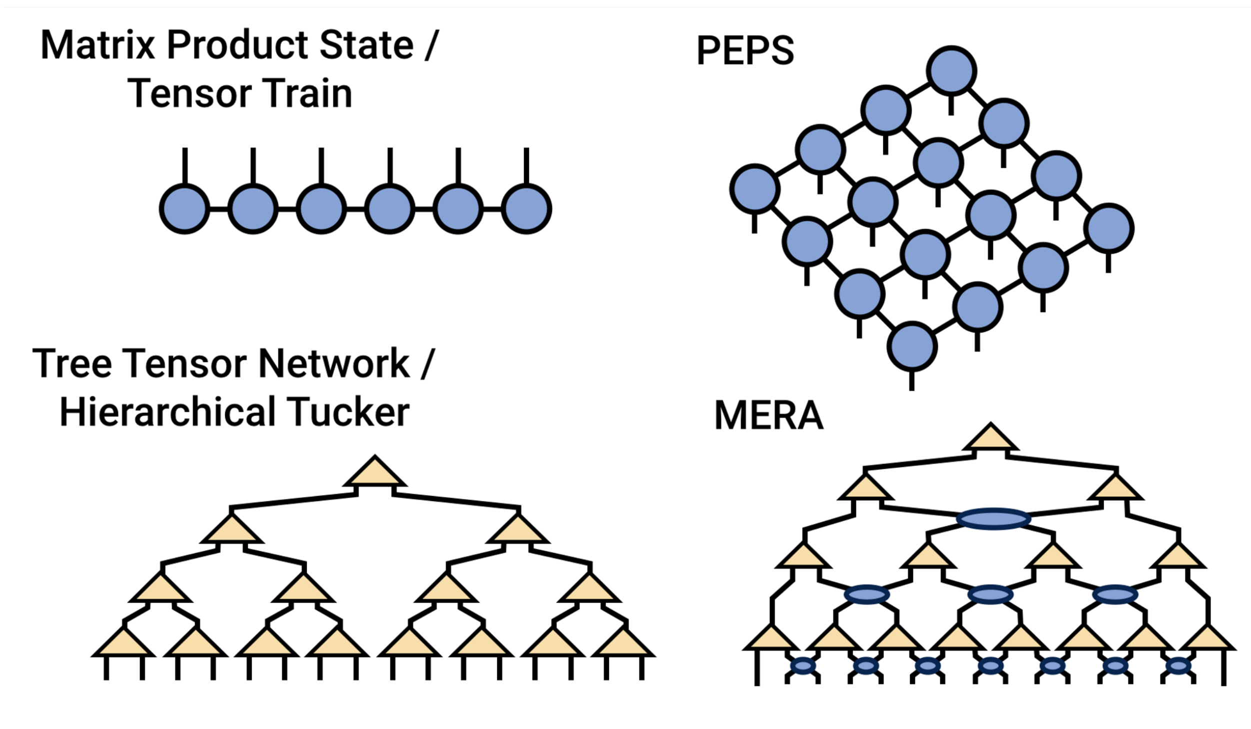
Learning Function — Weight Tensor — Kernelled Input Data




[3]

# QiML Methods: Tensor Networks

- Common Tensor Network Decompositions:




[4]

# QiML Methods

1. Tensor Network-based Learning Methods
2. **Quantum Variational Algorithm Simulation**
3. Other QiML Methods
4. Dequantized Algorithms

# QiML Methods: Q. Variational Alg. Simulation

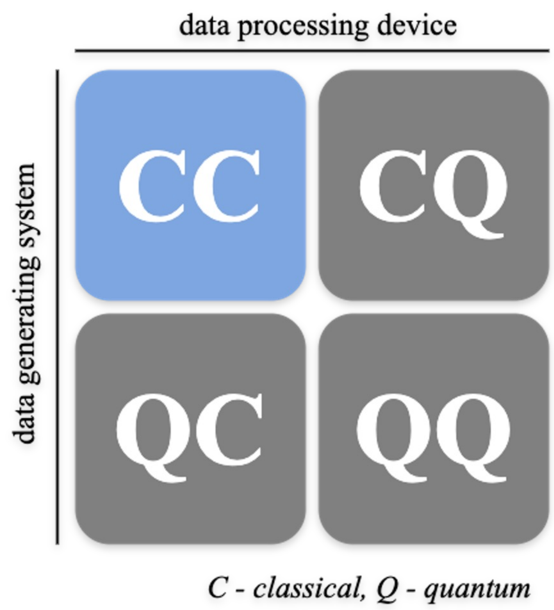
THE UNIVERSITY OF WESTERN AUSTRALIA

- Recall:
  - **CC:** Classical data and classical processing
  - **CQ:** Classical data and quantum processing

- **QML = CQ** (and QC, QQ)
- **QiML = CC**

  = Classical ML drawing inspiration from quantum mechanics/quantum computing, without need for quantum processing.

- If you can simulate QML classically, then this is also QiML!




[1]

# QiML Methods: Q. Variational Alg. Simulation

THE UNIVERSITY OF WESTERN AUSTRALIA

- **Quantum Kernel Estimation (QKE)**
  - Support vector machine (SVM) — dual formulation

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

# QiML Methods: Q. Variational Alg. Simulation

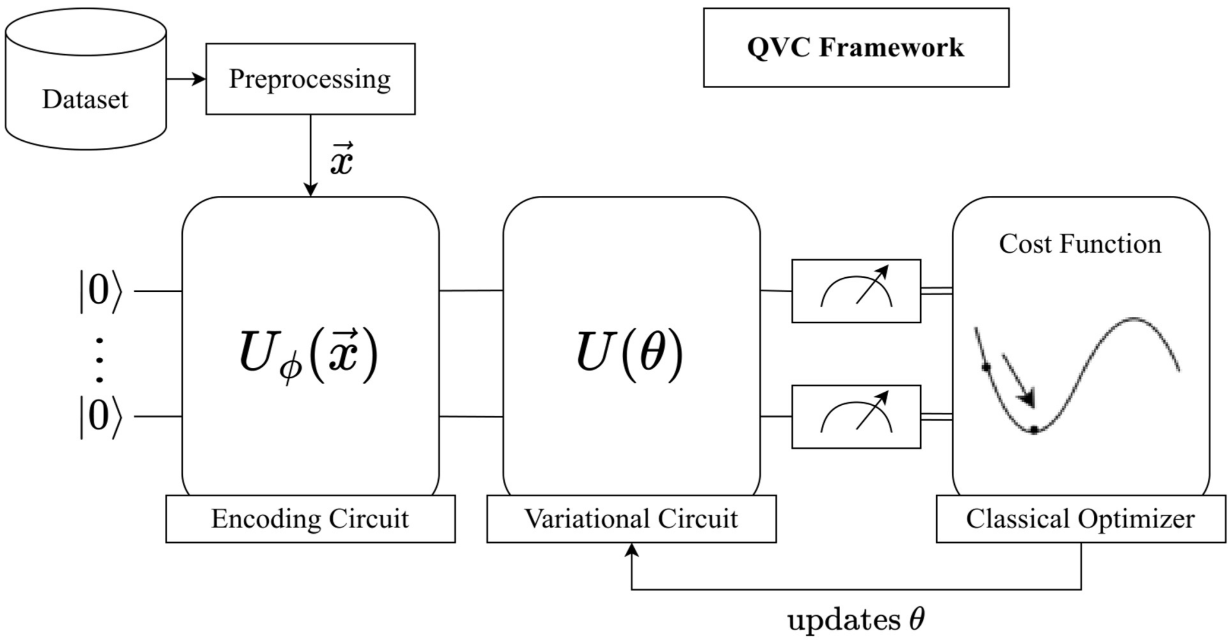
- **Quantum Kernel Estimation (QKE)**
  - Leverage quantum feature maps to perform the kernel trick

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

- Classical Kernel: $$K_{ij} = k(\vec{x}_i, \vec{x}_j)$$
- **Quantum Kernel:** $$K_{ij} = \left|\langle \phi(\vec{x}_i) \mid \phi(\vec{x}_j) \rangle\right|^2$$

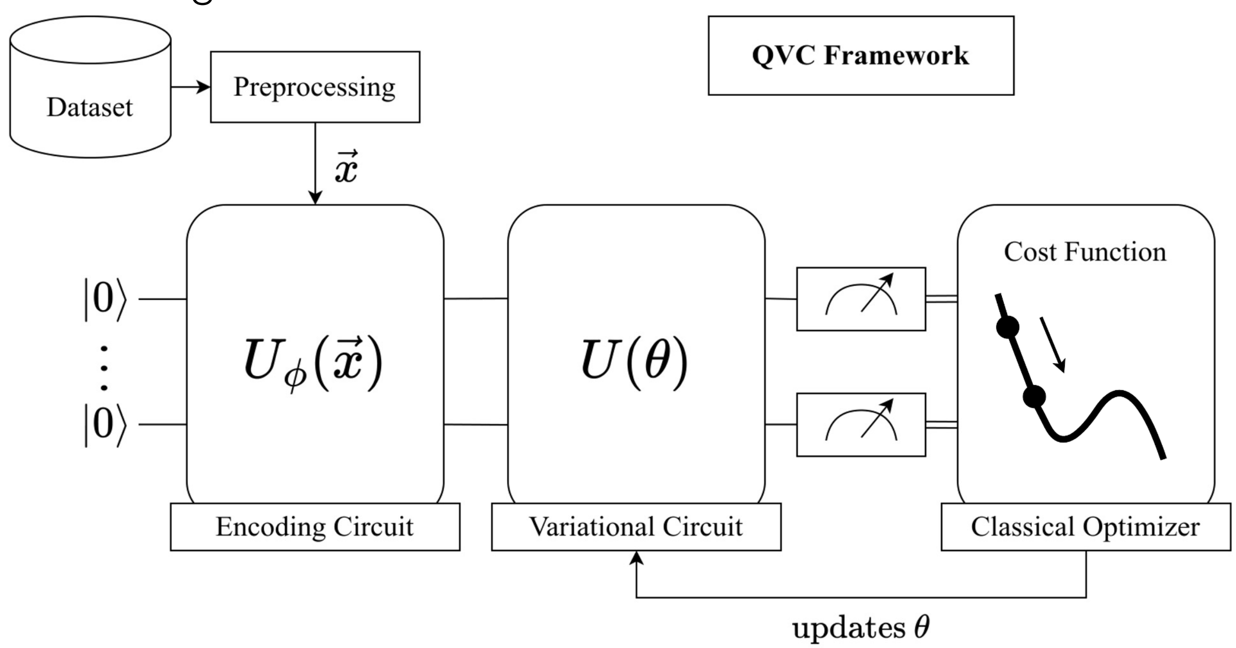
# QiML Methods: Q. Variational Alg. Simulation

- **Quantum Variational Circuits (QVC)**
  - Hybrid quantum-classical approach
  - Classical optimizer adjusts the parameters of a quantum circuit
  - Quantum analogues of neural networks



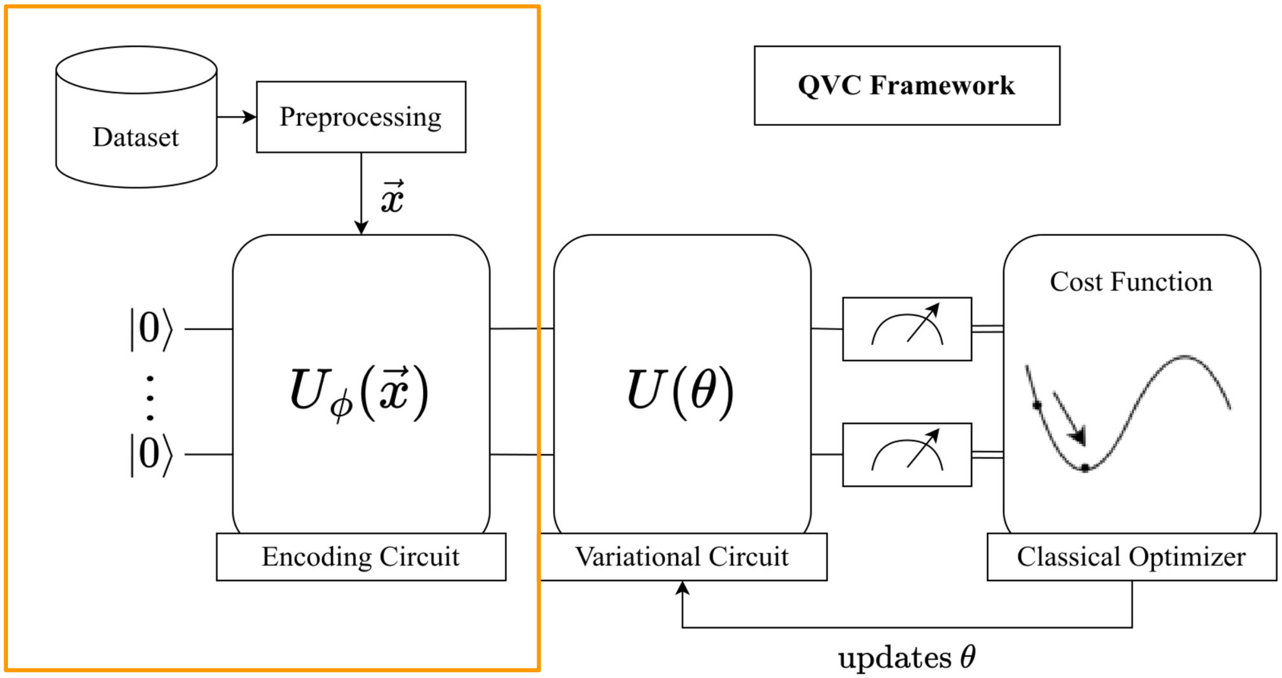
# QiML Methods: Q. Variational Alg. Simulation

- **Quantum Variational Circuits (QVC)**
  - Hybrid quantum-classical approach
  - Classical optimizer adjusts the parameters of a quantum circuit
  - Quantum analogues of neural networks



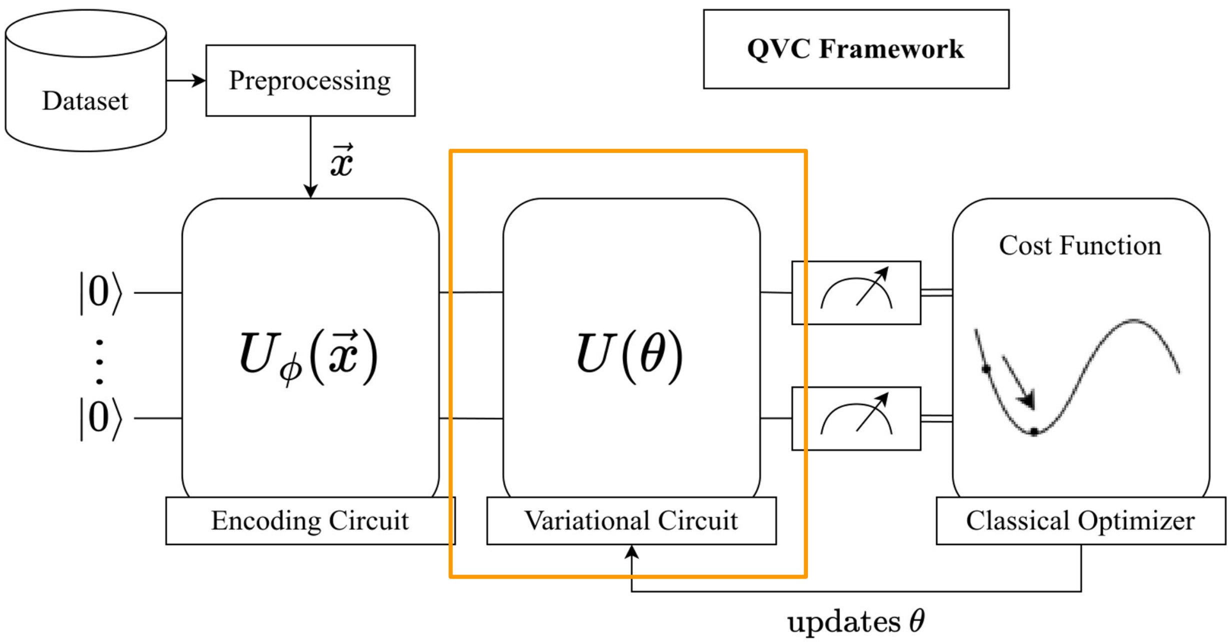
# QiML Methods: Q. Variational Alg. Simulation

- **1. Encoding Circuit**
  - Encodes classical data into quantum state space using a non-linear feature map $\phi$
  - Defined by circuit $U_\phi(\vec{x})$; induces qubit state based on input data $\vec{x}$



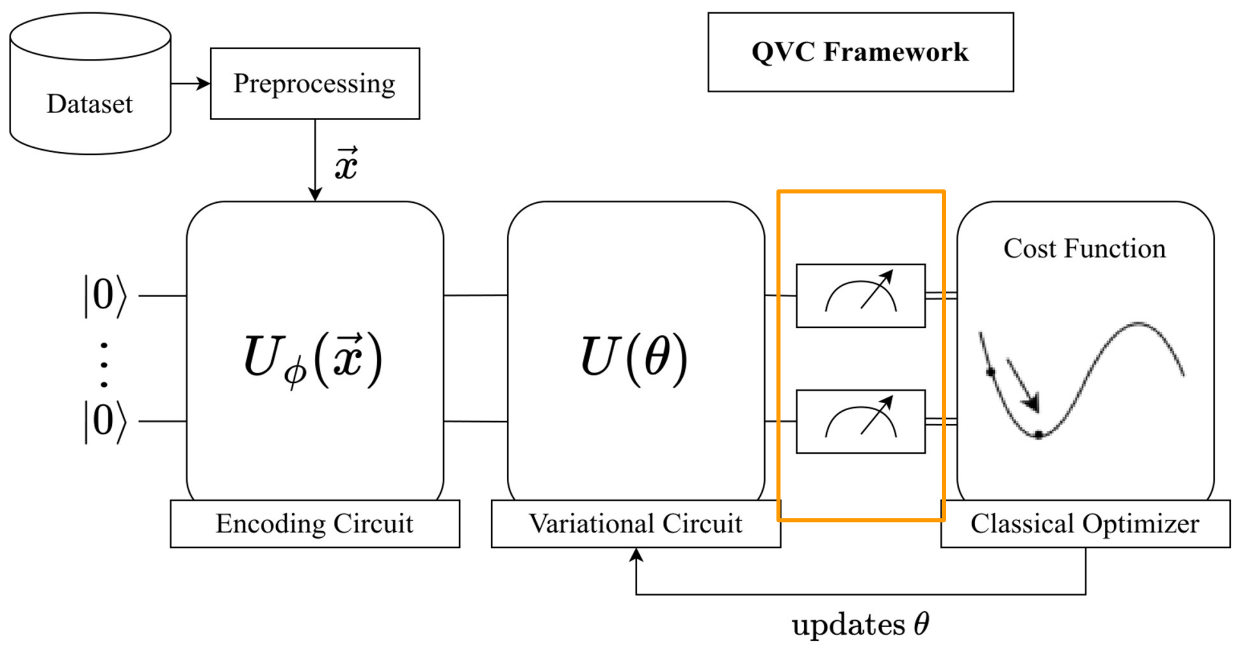
# QiML Methods: Q. Variational Alg. Simulation

- **2. Variational Circuit**
  - Quantum circuit learns a generalized representation of the data
  - Layers of quantum gates parameterized by a set of "free parameters" $\theta$



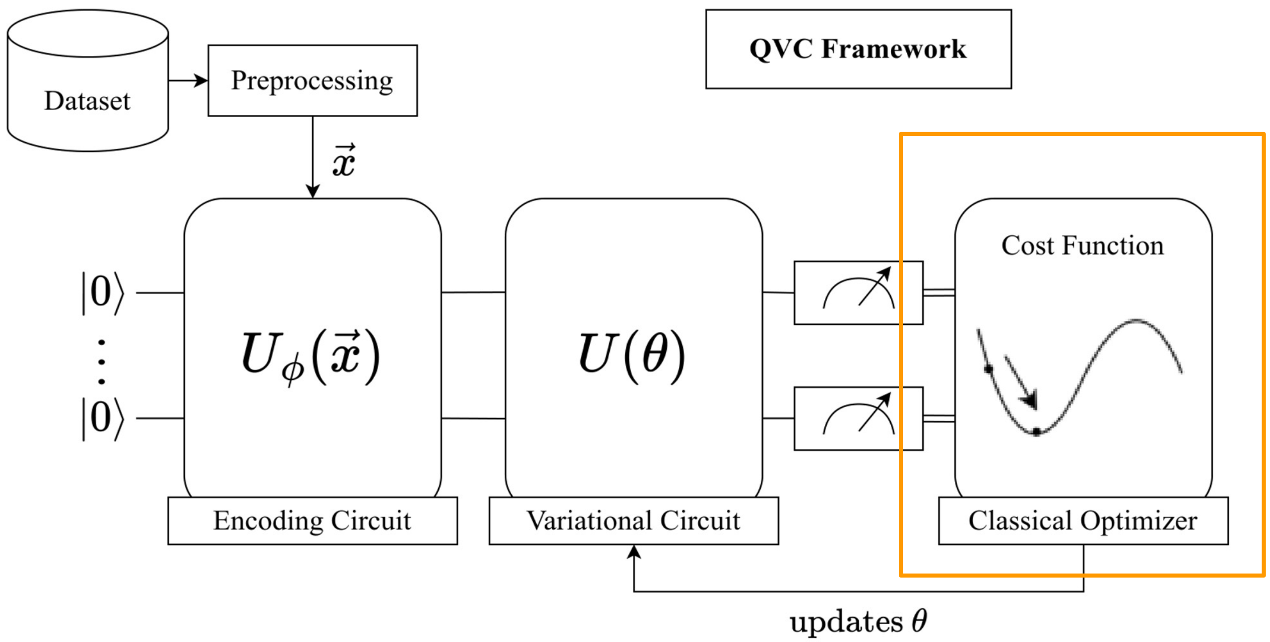
# QiML Methods: Q. Variational Alg. Simulation

- **3. Measurement**
  - Collapsing the resulting state into classical information
  - Expectation values → scalar cost function



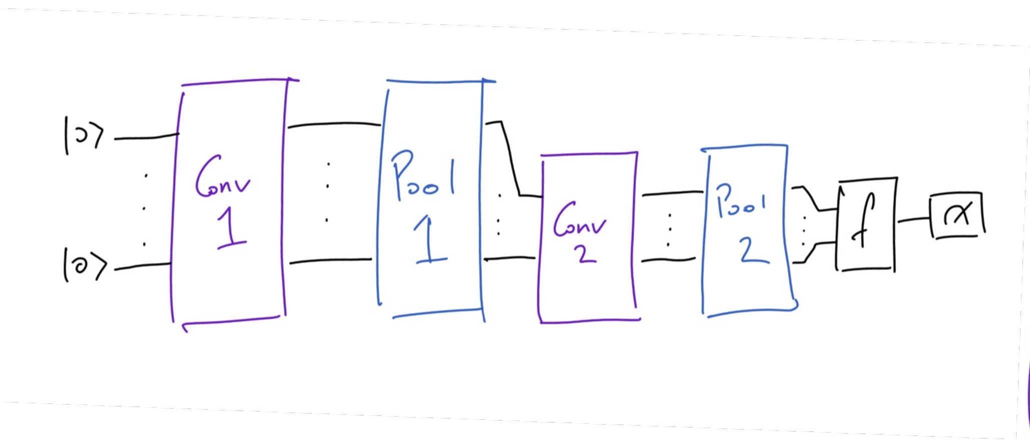
# QiML Methods: Q. Variational Alg. Simulation

- **4. Classical Optimization**
  - Cost function optimized via gradient descent on classical computer, adjusting parameters $\theta$

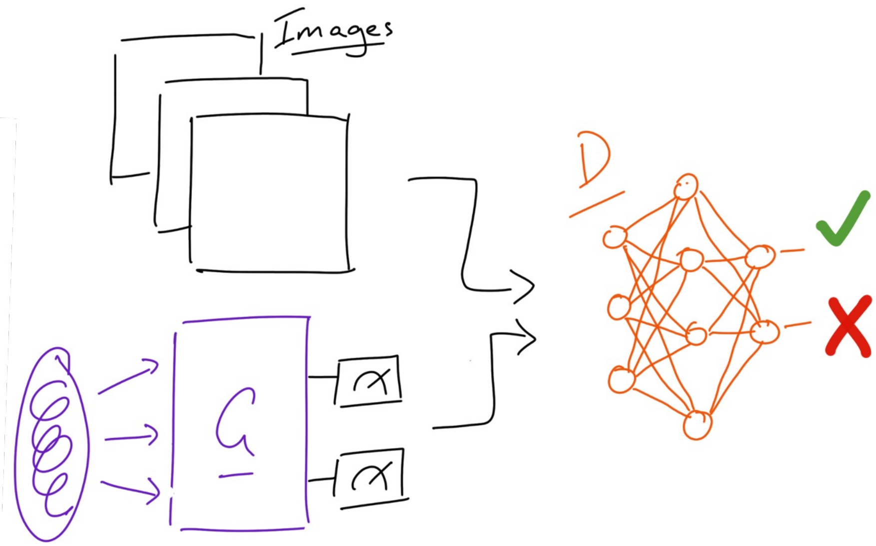


# QiML Methods: Q. Variational Alg. Simulation

- QVC framework as a basis for more complex models:
  - Quantum Convolutional Neural Networks (QCNN),
  - Quantum Generative Adversarial Networks (QGAN),
  - Quantum Autoencoder (QAE), ...




QCNN




QGAN

# QiML Methods:

1. Tensor Network-based Learning Methods
2. Dequantized Algorithms
3. **Other QiML Methods**
4. Quantum Variational Algorithm Simulation

# QiML Methods: Other Methods

- Quantum inspiration in classical machine learning:
  - Quantum-Inspired Nearest Mean Classifiers
  - Density Matrix-based Feature Representations
  - Quantum Formalisms in Neural Networks
  - ...
- Primarily takes advantage of the larger quantum feature space

# QiML: Strengths

- Utilization of quantum feature spaces = greater expressivity

- Strengths over classical ML — a mixed bag:
    - Inductive biases
    - Model size

# QiML: Limitations

THE UNIVERSITY OF WESTERN AUSTRALIA

- Constraints on data that are not present in classical ML
  - Dequantized algorithms: low rank, sometimes well conditioned input matrix
  - Tensor network: low bond dimension
  - Quantum circuits: small datasets, small feature sets
- Models scale poorly
- Speed and performance issues
  - In general, comparable, or worse than classical ML

# ML in Cybersecurity

- Learning Threat Patterns from Data
  - Intrusion detection systems
  - Software vulnerability detection
  - Malware detection
  - Spam filtering
  - ...

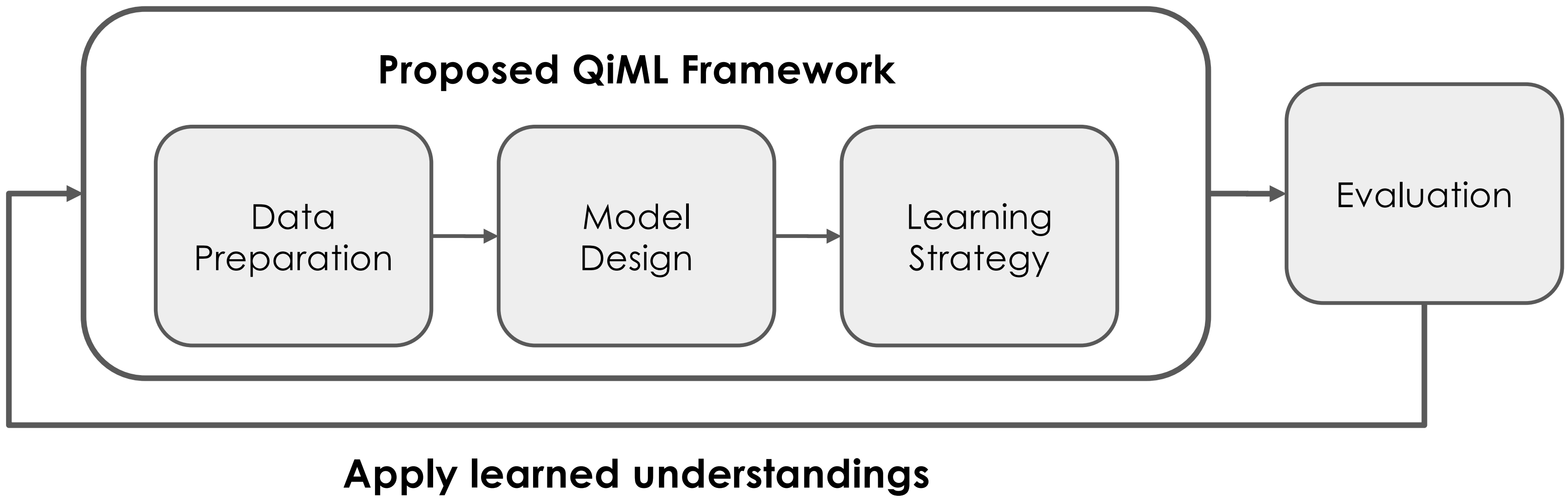
# QiML in Cybersecurity

- Tensor Networks:
  - Anomaly detection [6]
- Quantum Variational Algorithm Simulation:
  - DDoS detection [7]
  - Malware detection [7]
  - Source code vulnerability analysis [8]
  - Botnet detection [9]
  - Credit card fraud [10]
- However…
  - Small datasets and feature sets used
  - Needs excessive training time

# Research Objectives

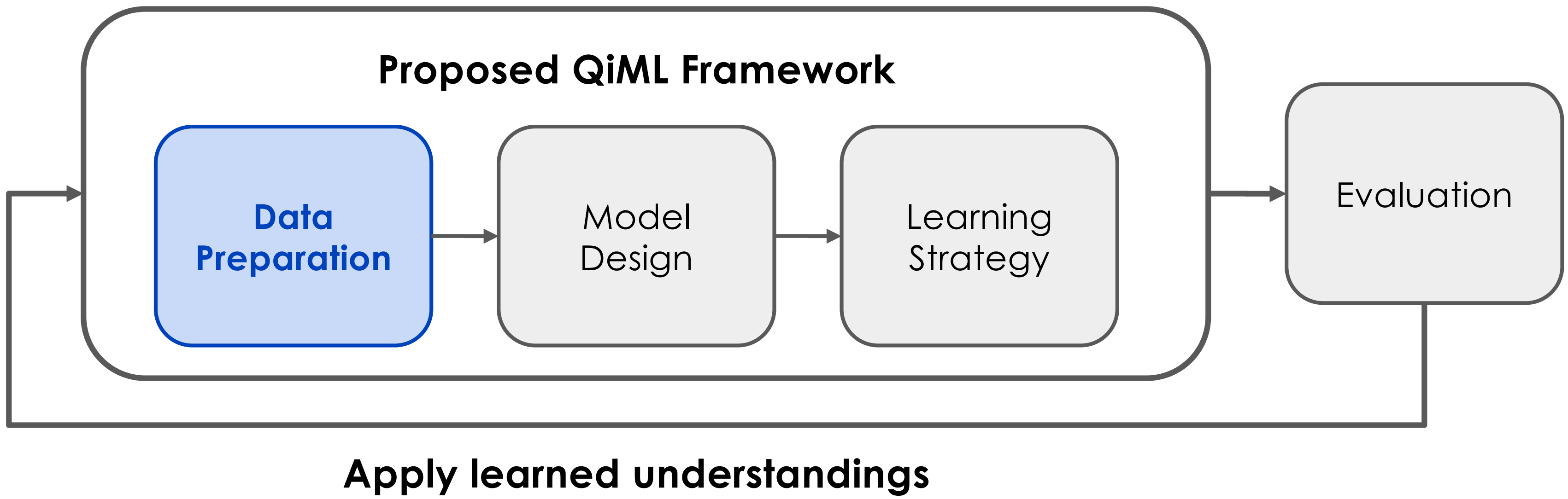
1. Deepen the understanding of how QiML can enhance cybersecurity
2. Explore QiML techniques and their impact on cybersecurity applications
3. Formulate advanced QiML strategies for enhanced cybersecurity

# Methodology



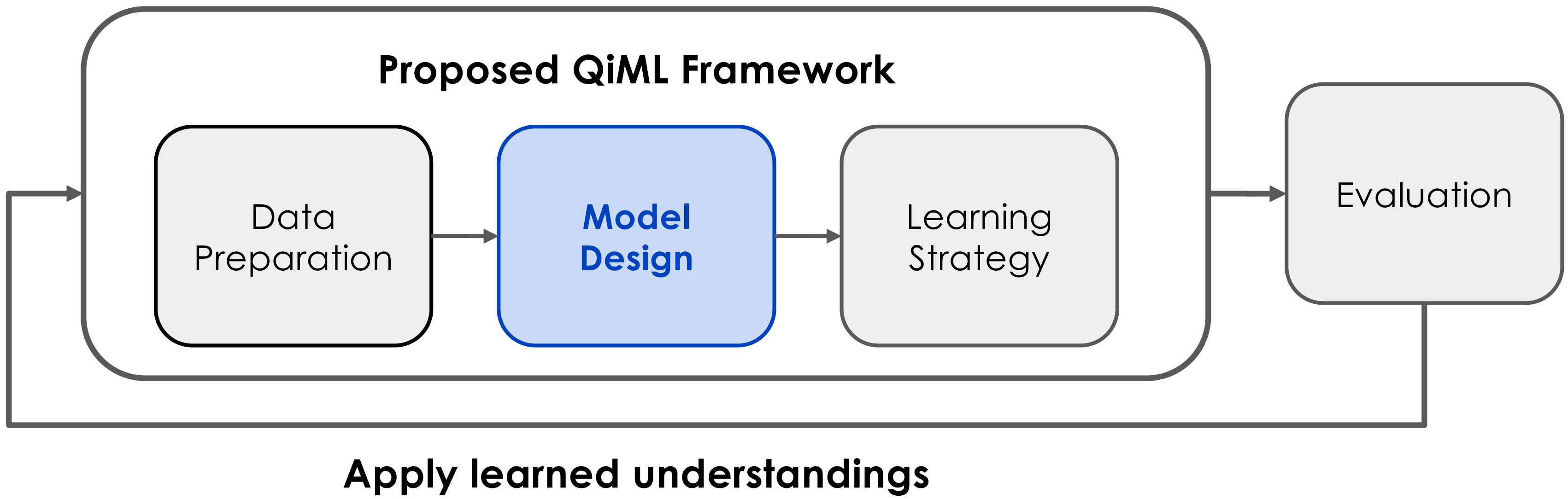

- Systematic approach to exploration of QiML applicability to IDS/SVD

# Methodology



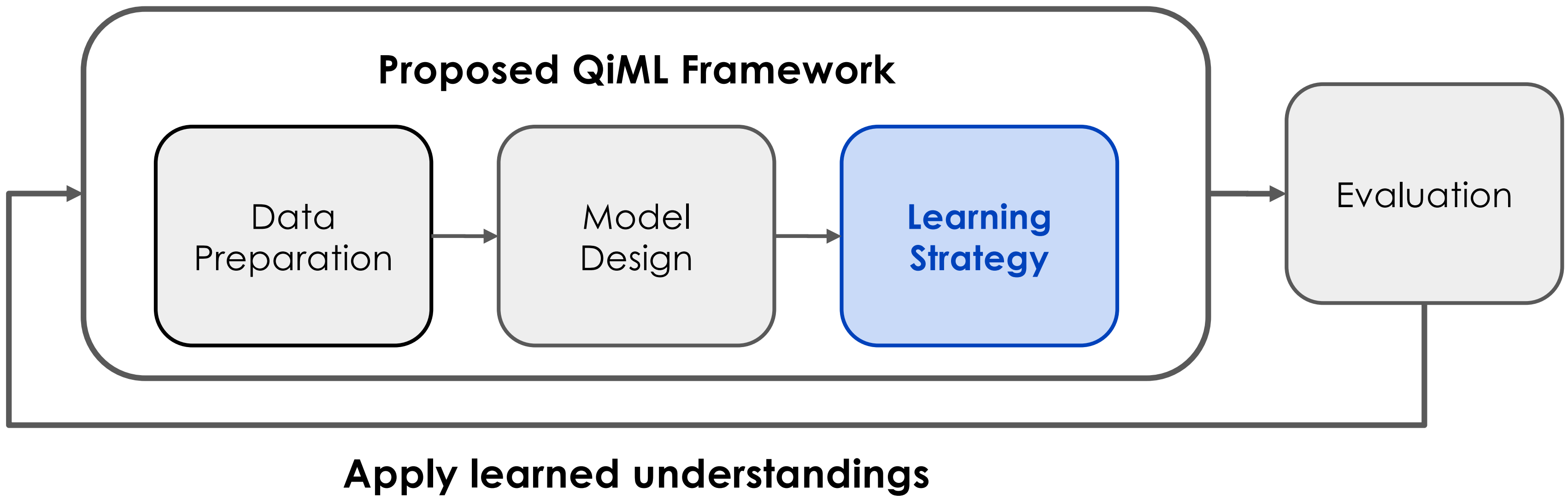

- Understanding the data: do quantum feature spaces help?
- Investigate suitable encoding schemes for the data.

# Methodology




- Explore various architectures (tensor networks, QVC) and investigate their applicability.

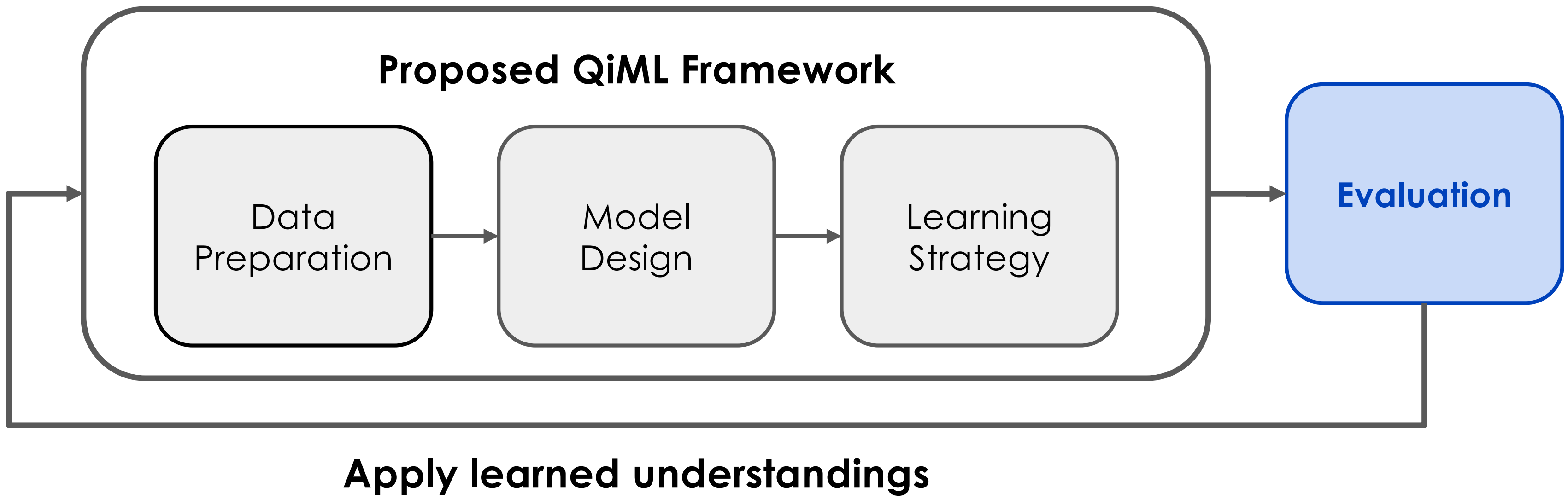
# Methodology




- Formulate learning strategies tailored for these models.

# Methodology

THE UNIVERSITY OF WESTERN AUSTRALIA




- Evaluate methods and refine solutions based on findings

# Evaluation

- **Datasets:** Benchmark IDS and SVD datasets

| Dataset | Year | No. of Features | Data Type |
|---|---|---|---|
| KDD Cup99 | 1998 | 41 | Emulated Traffic |
| NSL-KDD | 1998 | 41 | Emulated Traffic |
| ISOT | 2010 | 49 | Emulated Traffic |
| ISCX 2012 | 2012 | 8 | Emulated Traffic |
| UNSW-NB15 | 2015 | 42 | Emulated Traffic |
| KYOTO | 2015 | 24 | Real Traffic |
| CIC-IDS2017 | 2017 | 84 | Emulated Traffic |

Table 2: Publicly Available IDS Datasets [33]

| Dataset | Year | No. of Functions | % of Vulnerabilities |
|---|---|---|---|
| Big-Vul | 2020 | 188,636 | 5.78 |
| Devign | 2019 | 27,318 | 45.61 |
| D2A | 2021 | 1,295,623 | 1.44 |
| Juliet | 2012 | 253,002 | 36.77 |

Table 3: Publicly Available Software Vulnerability Detection Datasets [20]

# Evaluation

- **Metrics**:
  - Model Performance:
    - Accuracy,
    - Precision,
    - Recall,
    - F1
  - Computational Efficiency:
    - Complexity analysis (big-O)
    - Empirical assessment (running time)
  - Model Size:
    - Number of parameters

# Facilities & Costs

- **Facilities**:
  - Use of supercomputing (Pawsey) and HPC (CSIRO) if necessary

- **Costs:**
  - No estimated costs

# Confirmation of Candidature

THE UNIVERSITY OF WESTERN AUSTRALIA

- QiML survey paper completed – draft chapter in thesis
- Investigating QiML techniques for IDS – possible $2^{nd}$ paper & draft chapter

# Candidature Plan

| | Year 1 | | | | | | | | | | | | Year 2 | | | | | | | | | | | | Year 3 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2023 | | | | | | | | | | 2024 | | | | | | | | | | | | 2025 | | | | | | | | | | | | 2026 | |
| **TASK** | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 1 | 2 |
| Literature Review | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| First Paper (Survey) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proposal Submission | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Apply baseline QiML to IDS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Second Paper | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Substantial Piece of Writing | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Confirmation of Candidature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Apply baseline QiML to SVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Third Paper | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Enhancing IDS Methods | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mid candidature progress review 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Enhancing SVD Methods | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fourth Paper | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Thesis Writing | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mid candidature progress review 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Thesis Submission | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

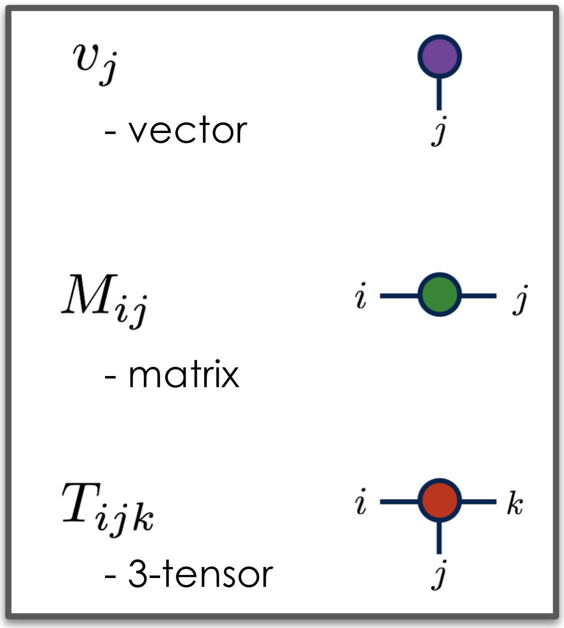
# References

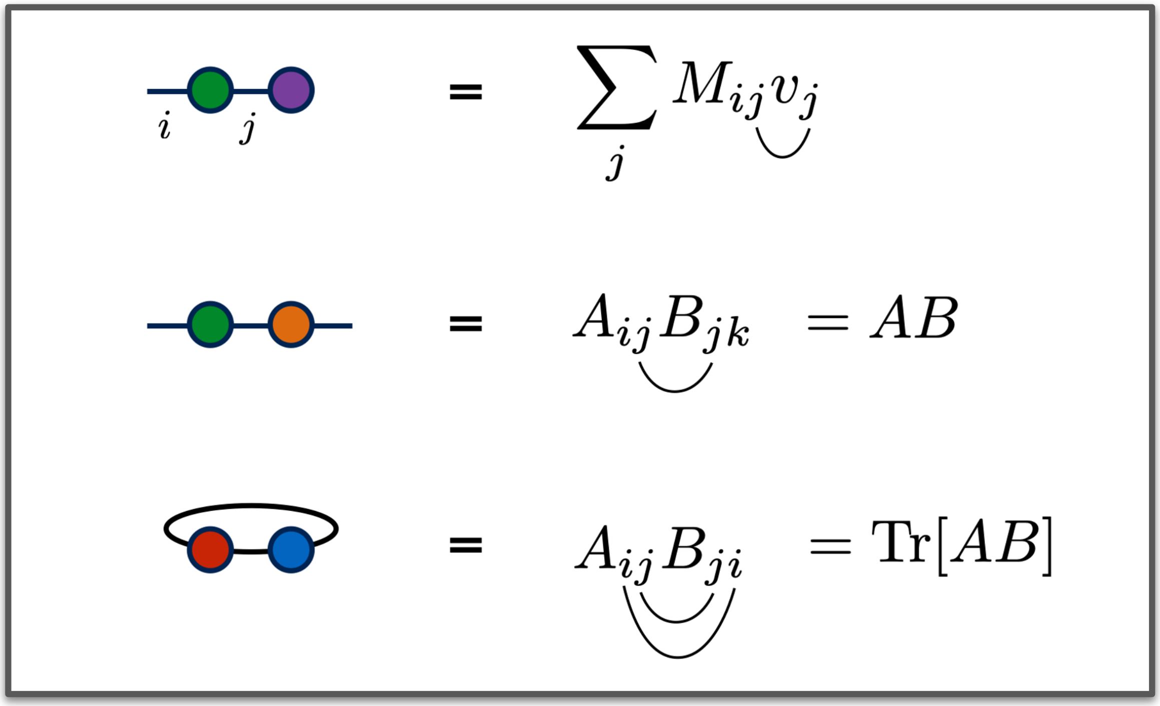

1. Schuld, M., & Petruccione, F. (2021). Machine learning with quantum computers. Berlin: Springer.
2. Prateek Joshi. (2013). Quantum Encryption And Black Holes - Part 2/2 - Perpetual Enigma.
3. Stoudenmire, E., & Schwab, D. J. (2016). Supervised learning with tensor networks. Advances in neural information processing systems, 29.
4. Tensor Network. (2023). Tensor Network.
5. Kerenidis, I., & Prakash, A. (2016). Quantum recommendation systems. arXiv preprint arXiv:1603.08675.
6. Wang, J., Roberts, C., Vidal, G., & Leichenauer, S. (2020). Anomaly detection with tensor networks. arXiv preprint arXiv:2006.02516.
7. Payares, E., & Martinez-Santos, J. (2021). Quantum machine learning for intrusion detection of distributed denial of service attacks: a comparative overview. Quantum Computing, Communication, and Simulation, 11699, 35–43.
8. Masum, M., Nazim, M., Faruk, M., Shahriar, H., Valero, M., Khan, M., Uddin, G., Barzanjeh, S., Saglamyurek, E., Rahman, A., & others (2022). Quantum Machine Learning for Software Supply Chain Attacks: How Far Can We Go?. In 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 530–538).
9. Suryotrisongko, H., & Musashi, Y. (2022). Evaluating hybrid quantum-classical deep learning for cybersecurity botnet DGA detection. Procedia Computer Science, 197, 223–229.
10. Herr, D., Obert, B., & Rosenkranz, M. (2021). Anomaly detection with variational quantum generative adversarial networks. Quantum Science and Technology, 6(4), 045004.

# Appendix

# QiML Methods: Tensor Networks

- Tensor Arithmetic - Tensor Diagram Notation



# QiML Methods: Tensor Networks

- Many-body quantum wavefunction:

$$|\Psi\rangle = \sum_{s_1 s_2 \cdots s_N} \Psi^{s_1 s_2 \cdots s_N} |s_1 s_2 \cdots s_N\rangle$$

- Decompose as a tensor network - **Matrix Product State (MPS)**:
  - Tensor with N sites, each of dimension d: $\mathbf{d^N}$ parameters.
  - MPS with bond dimension m: $\mathbf{Ndm^2}$ parameters; now scales linearly with N!

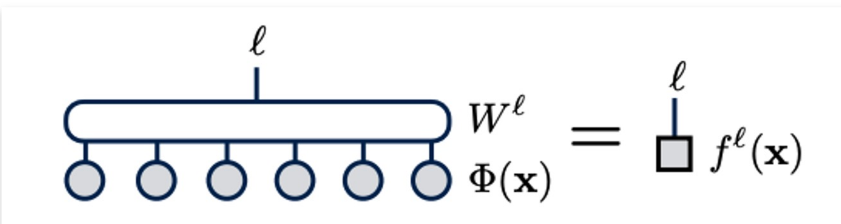
$$T_{s_1 s_2 s_3 s_4 s_5 s_6} \approx \sum_{\alpha} A^{s_1}_{\alpha_1} A^{s_2}_{\alpha_1 \alpha_2} A^{s_3}_{\alpha_2 \alpha_3} A^{s_4}_{\alpha_3 \alpha_4} A^{s_5}_{\alpha_4 \alpha_5} A^{s_6}_{\alpha_5}$$

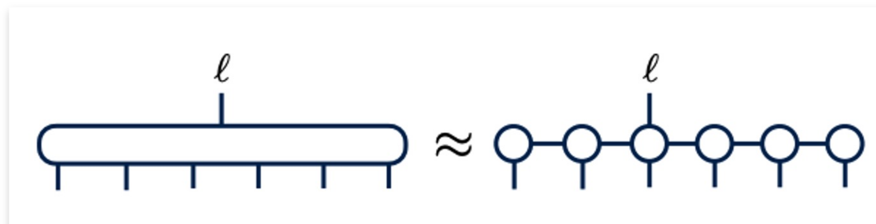
# QiML Methods: Tensor Networks

- Supervised Learning:
  - Treat the weight vector *W* as a wavefunction, and decompose as a tensor network!

$$f^l(x) = W^l \cdot \Phi(\mathbf{x})$$

$$W^l_{s_1 s_2 \ldots s_N} = \sum_{\{a\}} A^{\alpha_1}_{s_1} A^{\alpha_1 \alpha_2}_{s_2} \ldots A^{l:\alpha_j \alpha_{j+1}}_{s_j} \ldots A^{\alpha_{N-1}}_{s_N}$$



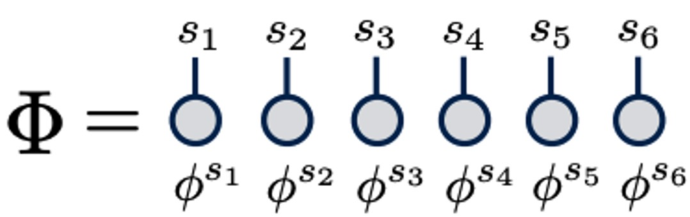
# QiML Methods: Tensor Networks

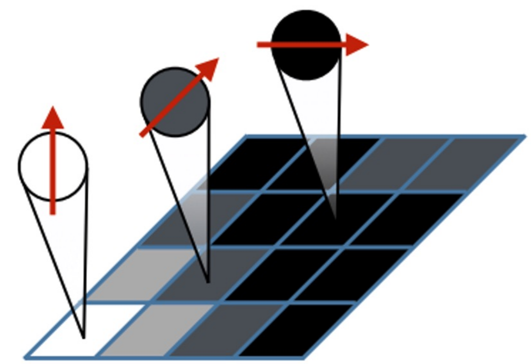
- Supervised Learning:
    - Input data as a tensor, with some local feature mapping

$$f^l(x) = W^l \cdot \Phi(\mathbf{x})$$

$$\Phi(\mathbf{x}) = \phi(x_1) \otimes \phi(x_2) \otimes \cdots \otimes \phi(x_N)$$

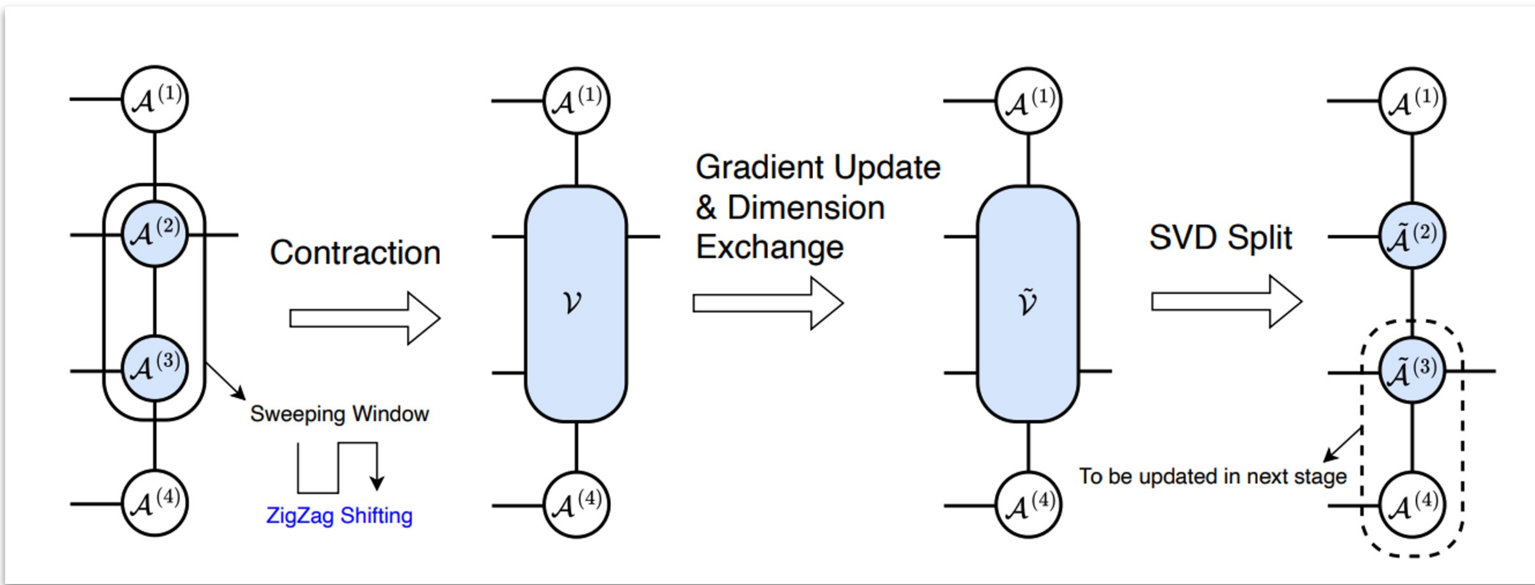
$$\phi(x_j) = \left[\cos\left(\frac{\pi}{2}x_j\right), \sin\left(\frac{\pi}{2}x_j\right)\right]$$



# QiML Methods: Tensor Networks

- Optimization:
  - Gradient descent-based methods (mostly batch or stochastic GD)
  - Density Matrix Renormalization Group (DMRG) “sweeping” algorithm



# QiML Methods: Tensor Networks

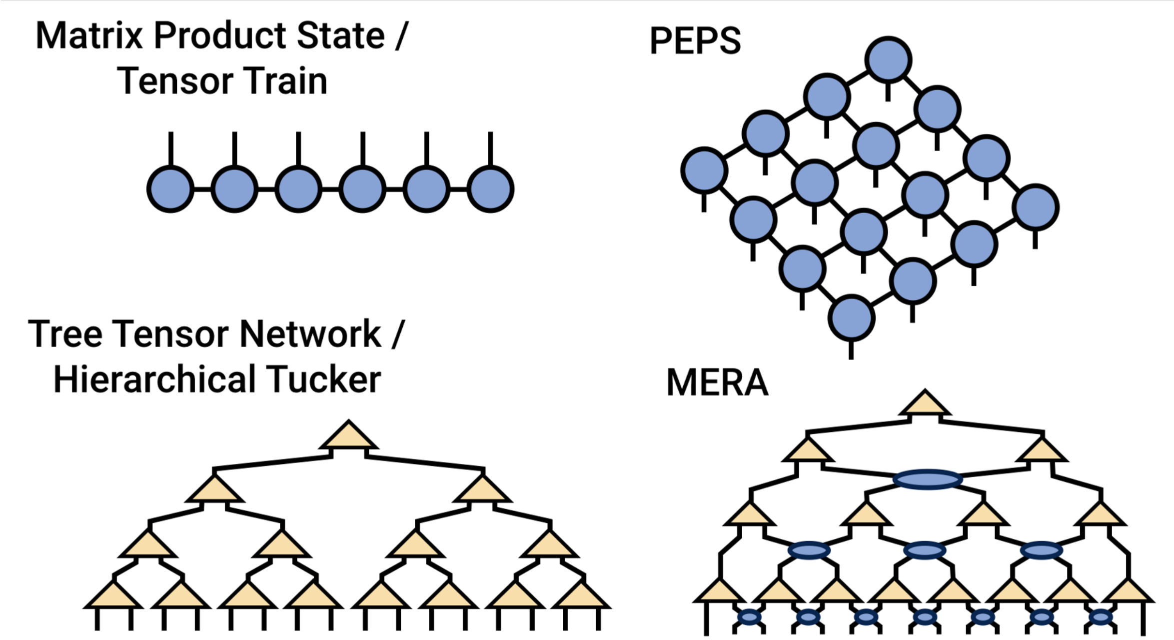
- Unsupervised Learning:
  - Encode some probability distribution into a wavefunction $\Psi(x)$, modelled by:

$$P(x) = \frac{|\Psi(x)|^2}{\sum |\Psi(x)|^2}$$

  - Decompose $\Psi(x)$ via some tensor network
  - Adjust parameters of the wavefunction such that the distribution given above is as close as possible to the data distribution in $x$.
    - Negative log-likelihood (NLL) typically used as cost function

# QiML Methods: Tensor Networks

- Common Tensor Network Decompositions:



# QiML Methods: Dequantized Algorithms

THE UNIVERSITY OF WESTERN AUSTRALIA

- Classical algorithms that **scrutinize** notions of *"quantum supremacy"*
- *"quantum supremacy":*
  - quantum computing's ability to strictly outperform classical systems
  - I.e. quantum algorithms are exponentially faster than classical ones

- *"Are QML algorithms inherently more powerful, or can this be attributed to* ***strong assumptions regarding I/O state preparation****?"*
- *"How to* ***compare the speed*** *of quantum algorithms with quantum I/O to classical algorithms with classical I/O?"*

# QiML Methods: Dequantized Algorithms

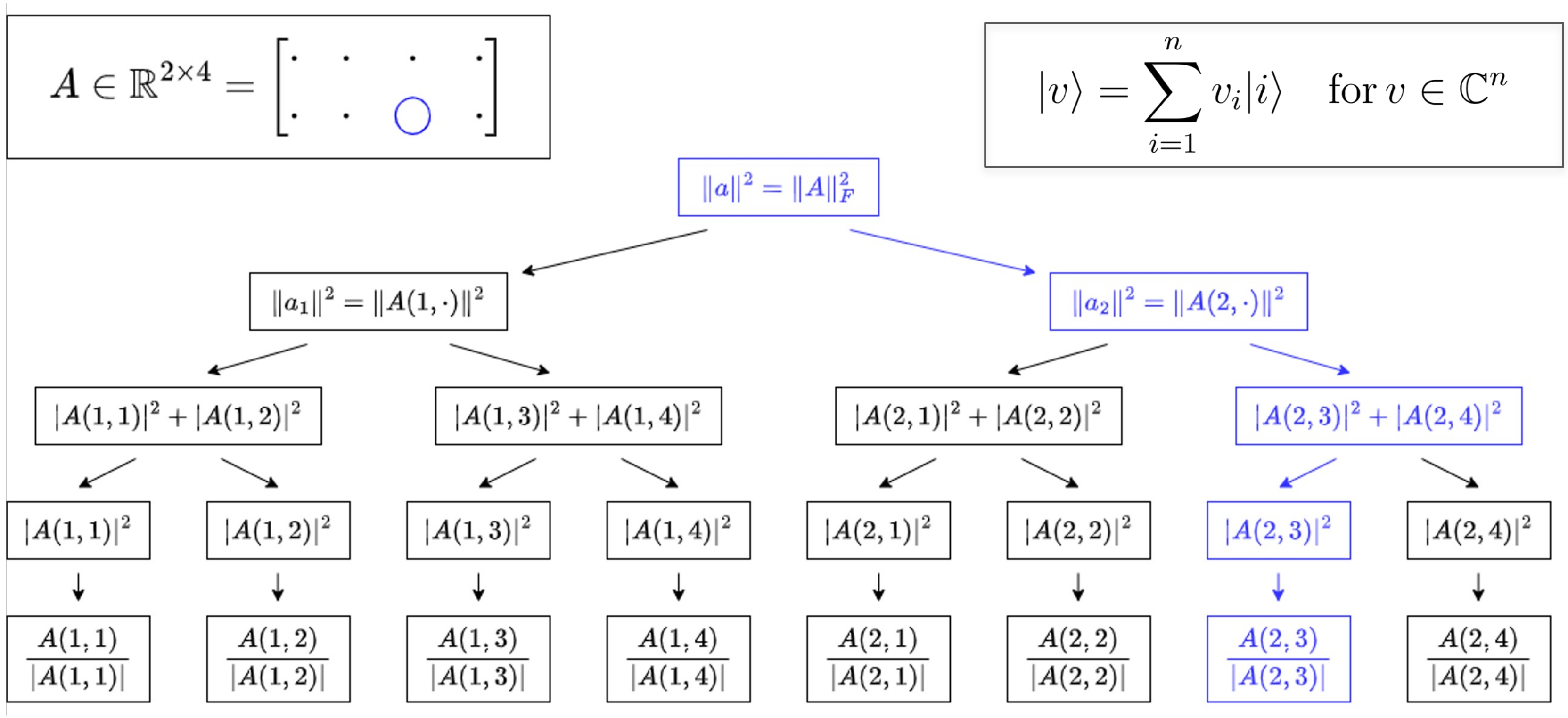
- *"Are QML algorithms inherently more powerful, or can this be attributed to **strong assumptions regarding I/O state preparation**?"*

- Prevailing assumptions in QML; either:
  - computing $|v\rangle$ from some input vector $v$ is **arbitrarily fast**, or;
  - the necessary quantum states come into the system **already prepared**.

- The cost of state preparation is **non-trivial!**
- Quantum supremacy is only apparent if state preparation is performed in **poly-logarithmic time!**

# QiML Methods: Dequantized Algorithms

- Kerenidis and Prakash: explicit I/O quantum state preparation routine



# QiML Methods: Dequantized Algorithms

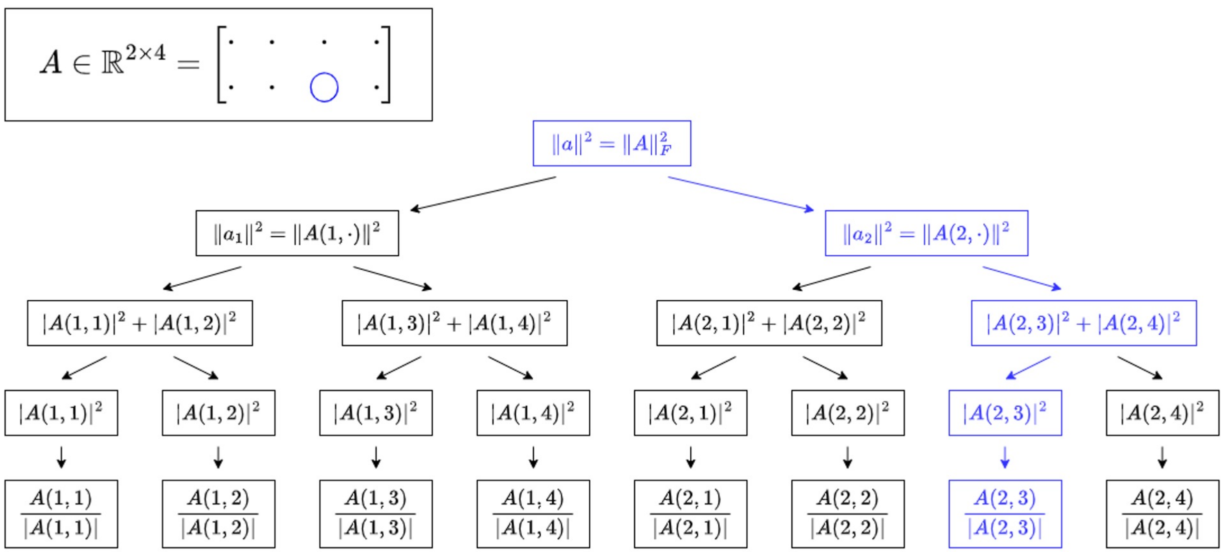
- Kerenidis and Prakash: explicit I/O quantum state preparation routine

- Assumes quantum access to this data structure with prepared quantum states.
- Promising candidate for demonstrably exponential improvement.

- **However!**
  - This structure actually also fulfills **classical L2-norm sampling assumptions**.
  - Imposing similar constraints on I/O allows for **fairer comparison between quantum and classical algorithms.**

# QiML Methods: Dequantized Algorithms

- *"Sample and Query Access"* - Classical L2-norm sampling assumptions
- For a vector $v \in \mathbb{C}^N$, we have $SQ(v)$ if, in $\text{polylog}(N)$ time, we can:
  - **Sample**: sample independently $v_i$ from $v$ with prob. $x_i^2/\|x\|$
  - **Query**: output entries $v_i$ of $v$
  - **Norm**: determine $\|v\|$



# QiML Methods: Dequantized Algorithms

- Successfully Dequantized ML Routines:
  - Recommendations Systems
  - Supervised Clustering
  - Matrix Inversion
  - Principal Component Analysis
  - Support Vector Machines
  - Semi-definite Programming
  - **Quantum Singular Value Transformation (QSVT)**
  - Hamiltonian Simulation
  - Discriminant Analysis

# QiML Methods: Dequantized Algorithms

- Briefly describe qsvt, or rec. Sys. not sure which one will be more digestible?

# QiML Methods: Dequantized Algorithms

- Dequantized Algorithm - Landscape and Complexities Overview

| | Quantum Algorithm | Dequantized Algorithms | | | | | |
|---|---|---|---|---|---|---|---|
| **Rec. Systems** [129], [250], [36],[35], [13] | $\frac{\|A\|_F}{\sigma}$ | $\frac{\|A\|_F^{24}}{\sigma^{24}\varepsilon^{12}}$, | $\frac{\|A\|_F^6\|A\|^{10}}{\sigma^{16}\varepsilon^6}$, | $\frac{\|A\|_F^6}{\sigma^6\varepsilon^6}$, | $\frac{\|A\|_F^4}{\sigma^9\varepsilon^2}$ | | |
| **Supervised Clustering** [152], [251], [36] | $\frac{\|M\|_F^2\|w\|^2}{\varepsilon}$ | $\frac{\|M\|_F^4\|w\|^4}{\varepsilon^2}$, | $\frac{\|M\|_F^4\|w\|^4}{\varepsilon^2}$ | | | | |
| **PCA** [153], [251], [36] | $\frac{\|X\|_F\|X\|}{\lambda_k\varepsilon}$ | $\frac{\|X\|_F^{36}}{\|X\|^{12}\lambda_k^{12}\eta^6\varepsilon^{12}}$, | $\frac{\|X\|_F^6}{\|X\|^2\lambda_k^2\eta^6\varepsilon^6}$ | | | | |
| **Matrix Inversion** [91], [89], [36],[90], [35], [232] [13] | $\frac{\|A\|_F}{\sigma}$ | $\frac{k^6\|A\|_F^6\|A\|^{16}}{\sigma^{22}\varepsilon^6}$, | $\frac{\|A\|_F^6\|A\|^{22}}{\sigma^{28}\varepsilon^6}$, | $\frac{\|A\|_F^6\|A\|^6}{\sigma^{12}\varepsilon^4}$, | $\frac{\|A\|_F^4\log(c)}{\sigma^8\varepsilon^4}$, | $\frac{\|A\|_F^6\|A\|^2}{\sigma^8\varepsilon^2}$, | $\frac{\|A\|_F^4}{\sigma^{11}\varepsilon^2}$ |
| **SVM** [204], [63], [36] | $\frac{1}{\lambda^3\varepsilon^3}$ | $\text{poly}\left(\frac{1}{\lambda}, \frac{1}{\varepsilon}\right)$, | $\frac{1}{\lambda^{28}\varepsilon^6}$ | | | | |
| **SDP** [263], [37], [36] | $\frac{\|A^{(\cdot)}\|_F^7}{\varepsilon^{7.5}} + \frac{\sqrt{m}\|A^{(\cdot)}\|_F^2}{\varepsilon^4}$ | $\frac{mk^{57}}{\varepsilon^{92}}$, | $\frac{\|A^{(\cdot)}\|_F^{22}}{\varepsilon^{46}} + \frac{\sqrt{m}\|A^{(\cdot)}\|_F^{14}}{\varepsilon^{28}}$ | | | | |
| **QSVT** [91], [36], [126], [13] | $\frac{d\|A\|_F\|b\|}{p^{(QV)}(A)b}$ | $\frac{d^{22}\|A\|_F^6}{\varepsilon^6}$, | $\frac{\|A\|_F^6\kappa^{20}(d^2+\kappa)}{\varepsilon^6}$, | $\frac{d^{11}\|A\|_F^4}{\varepsilon^2}$ | | | |
| **HS** [91], [36], [13] | $\|H\|_F$ | $\frac{\|H\|_F^6\|H\|^{16}}{\varepsilon^6}$, | $\frac{\|H\|_F^4\|H\|^9}{\varepsilon^2}$ | | | | |
| **DA** [47], [36] | $\frac{\|B\|_F^7}{\varepsilon^3\sigma^7} + \frac{\|W\|_F^7}{\varepsilon^3\sigma^7}$ | $\frac{\|B\|_F^6\|B\|^4}{\varepsilon^6\sigma^{10}} + \frac{\|W\|_F^7\|W\|^{10}}{\varepsilon^6\sigma^{16}}$ | | | | | |

# QiML Methods: Dequantized Algorithms

- Dequantized Algorithm - Landscape and Complexities Overview

| | Quantum Algorithm | Dequantized Algorithms | | | | | |
|---|---|---|---|---|---|---|---|
| **Rec. Systems** [129], [250], [36],[35], [13] | $\frac{\|A\|_F}{\sigma}$ | $\frac{\|A\|_F^{24}}{\sigma^{24}\varepsilon^{12}}$, | $\frac{\|A\|_F^{6}\|A\|^{10}}{\sigma^{16}\varepsilon^{6}}$, | $\frac{\|A\|_F^{6}}{\sigma^{6}\varepsilon^{6}}$, | $\frac{\|A\|_F^{4}}{\sigma^{9}\varepsilon^{2}}$ | | |
| **Supervised Clustering** [152], [251], [36] | $\frac{\|M\|_F^{2}\|w\|^{2}}{\varepsilon}$ | $\frac{\|M\|_F^{4}\|w\|^{4}}{\varepsilon^{2}}$, | $\frac{\|M\|_F^{4}\|w\|^{4}}{\varepsilon^{2}}$ | | | | |
| **PCA** [153], [251], [36] | $\frac{\|X\|_F\|X\|}{\lambda_k \varepsilon}$ | $\frac{\|X\|_F^{36}}{\|X\|^{12}\lambda_k^{12}\eta^{6}\varepsilon^{12}}$, | $\frac{\|X\|_F^{6}}{\|X\|^{2}\lambda_k^{2}\eta^{6}\varepsilon^{6}}$ | | | | |
| **Matrix Inversion** [91], [89], [36],[90], [35], [232] [13] | $\frac{\|A\|_F}{\sigma}$ | $\frac{k^6\|A\|_F^{6}\|A\|^{16}}{\sigma^{22}\varepsilon^{6}}$, | $\frac{\|A\|_F^{6}\|A\|^{22}}{\sigma^{28}\varepsilon^{6}}$, | $\frac{\|A\|_F^{6}\|A\|^{6}}{\sigma^{12}\varepsilon^{4}}$, | $\frac{\|A\|_F^{4}\log(c)}{\sigma^{8}\varepsilon^{4}}$, | $\frac{\|A\|_F^{6}\|A\|^{2}}{\sigma^{8}\varepsilon^{2}}$, | $\frac{\|A\|_F^{4}}{\sigma^{11}\varepsilon^{2}}$ |
| **SVM** [204], [63], [36] | $\frac{1}{\lambda^3\varepsilon^3}$ | $\text{poly}\left(\frac{1}{\lambda}, \frac{1}{\varepsilon}\right)$, | $\frac{1}{\lambda^{28}\varepsilon^{6}}$ | | | | |
| **SDP** [263], [37], [36] | $\frac{\|A^{(\cdot)}\|_F^{7}}{\varepsilon^{7.5}} + \frac{\sqrt{m}\|A^{(\cdot)}\|_F^{2}}{\varepsilon^{4}}$ | $\frac{mk^{57}}{\varepsilon^{92}}$, | $\frac{\|A^{(\cdot)}\|_F^{22}}{\varepsilon^{46}} + \frac{\sqrt{m}\|A^{(\cdot)}\|_F^{14}}{\varepsilon^{28}}$ | | | | |
| **QSVT** [91], [36], [126], [13] | $\frac{d\|A\|_F\|b\|}{p^{(QV)}(A)b}$ | $\frac{d^{22}\|A\|_F^{6}}{\varepsilon^{6}}$, | $\frac{\|A\|_F^{6}\kappa^{20}(d^2+\kappa)}{\varepsilon^{6}}$, | $\frac{d^{11}\|A\|_F^{4}}{\varepsilon^{2}}$ | | | |
| **HS** [91], [36], [13] | $\|H\|_F$ | $\frac{\|H\|_F^{6}\|H\|^{16}}{\varepsilon^{6}}$, | $\frac{\|H\|_F^{4}\|H\|^{9}}{\varepsilon^{2}}$ | | | | |
| **DA** [47], [36] | $\frac{\|B\|_F^{7}}{\varepsilon^3\sigma^7} + \frac{\|W\|_F^{7}}{\varepsilon^3\sigma^7}$ | $\frac{\|B\|_F^{6}\|B\|^{4}}{\varepsilon^6\sigma^{10}} + \frac{\|W\|_F^{7}\|W\|^{10}}{\varepsilon^6\sigma^{16}}$ | | | | | |

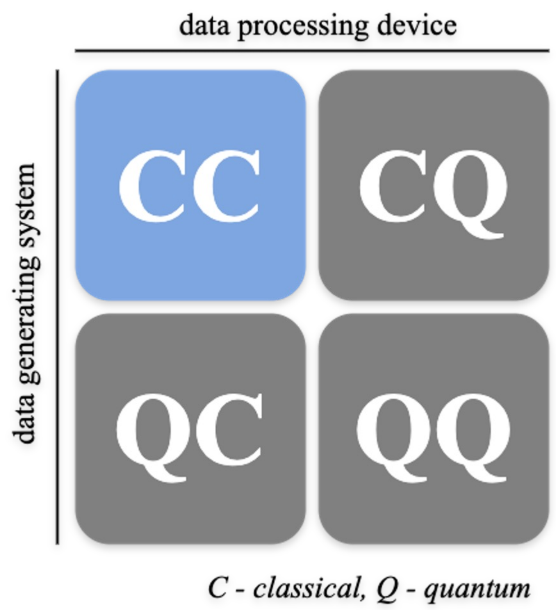
# QiML Methods: Q. Variational Alg. Simulation

- Recall:
  - **CC:** Classical data and classical processing
  - **CQ:** Classical data and quantum processing

- **QML = CQ** (and QC, QQ)
- **QiML = CC**

  = Classical ML drawing inspiration from quantum mechanics/quantum computing, without need for quantum processing.

- If you can simulate QML classically, then this is also QiML!



# QiML Methods: Q. Variational Alg. Simulation

THE UNIVERSITY OF WESTERN AUSTRALIA

- Simulating Quantum Computation - Challenges:
  - Quantum state spaces grow exponential with number of qubits
  - Quantum phenomena (superposition, entanglement, interference) requires the storage of all amplitudes exactly

- PC with 16GB GPU memory ≈ 30 qubits
- >50 qubits requires HPC/supercomputing
- However, low-qubit simulations have shown comparable results

# QiML Methods: Q. Variational Alg. Simulation

- **Quantum Kernel Estimation (QKE)**
  - Dual representation of the support vector machine (SVM)

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

# QiML Methods: Q. Variational Alg. Simulation



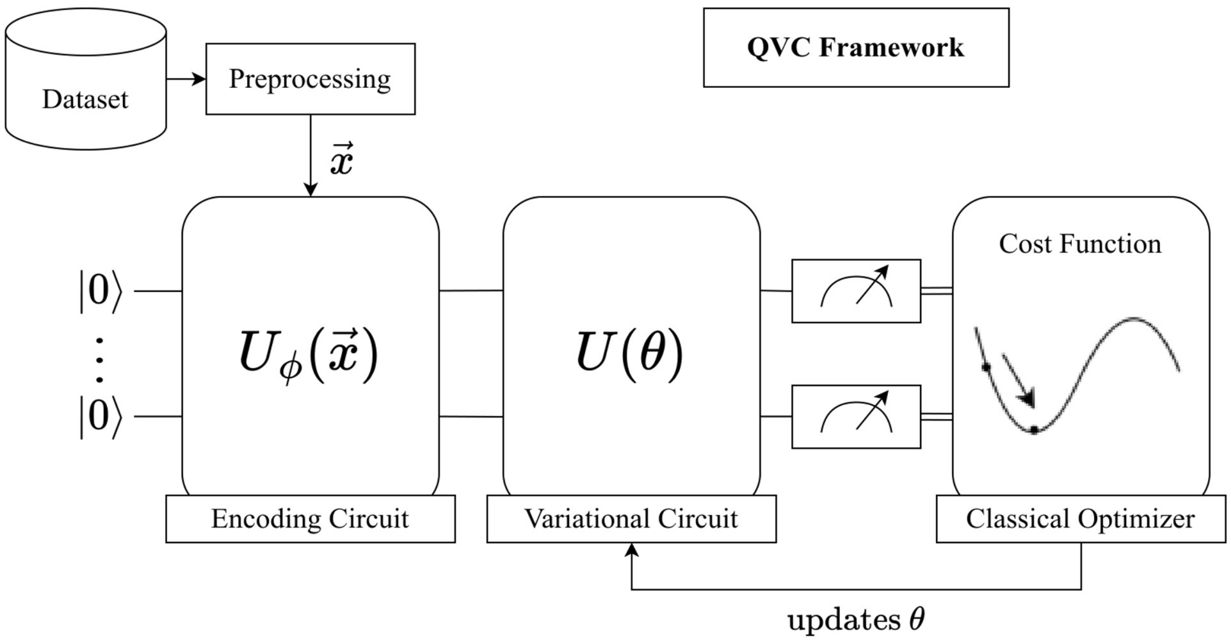
- **Quantum Kernel Estimation (QKE)**
  - Leverage quantum feature maps to perform the kernel trick

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boxed{K(\mathbf{x}_i, \mathbf{x}_j)}$$

- Classical Kernel: $$K_{ij} = k(\vec{x}_i, \vec{x}_j)$$
- **Quantum Kernel:** $$K_{ij} = \left|\langle \phi(\vec{x}_i) \mid \phi(\vec{x}_j) \rangle\right|^2$$

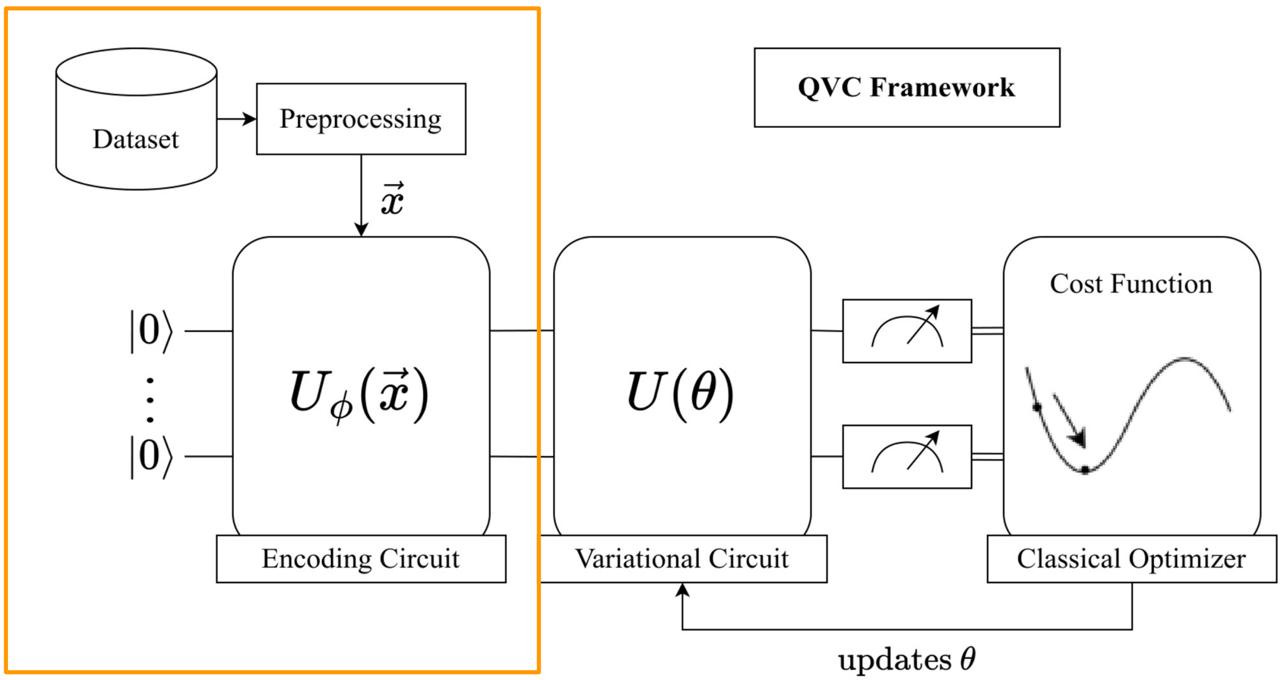
# QiML Methods: Q. Variational Alg. Simulation

- **Quantum Variational Circuits (QVC)**
  - Hybrid quantum-classical approach
  - Classical optimizer adjusts the parameters of a quantum circuit
  - Quantum analogues of neural networks



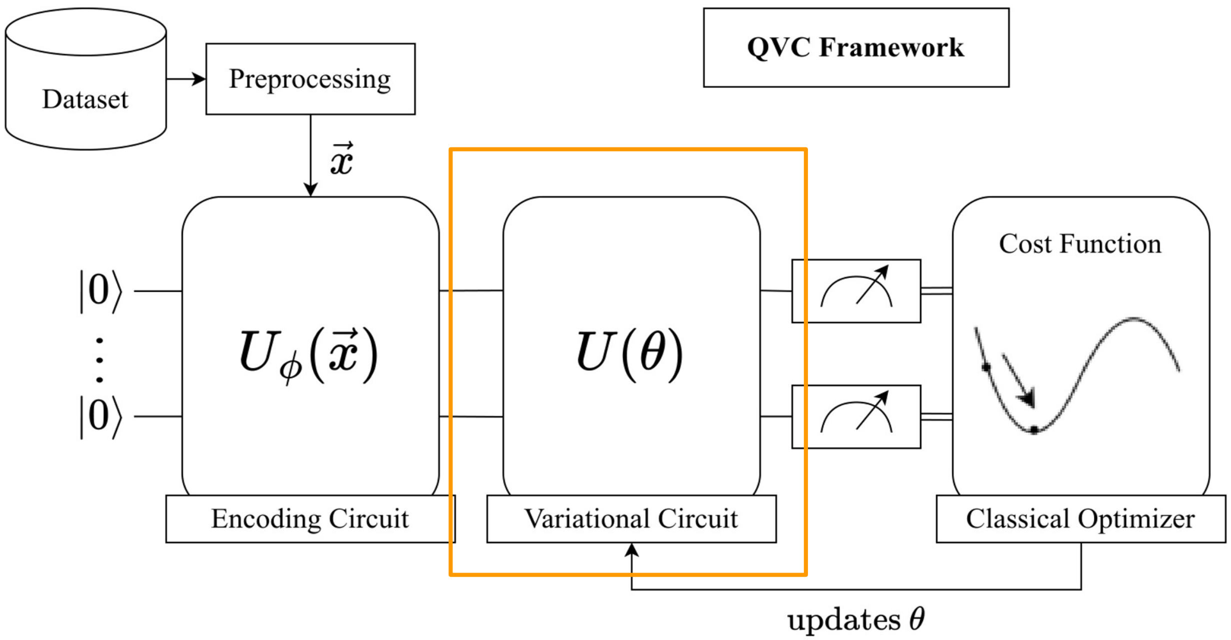
# QiML Methods: Q. Variational Alg. Simulation

- **1. Encoding Circuit**
  - Encodes classical data into quantum state space using a non-linear feature map $\phi$
  - Defined by circuit $U_\phi(\vec{x})$, and acts on data: $\vec{x} \rightarrow U_\phi(\vec{x})|0\rangle^{\otimes n}$



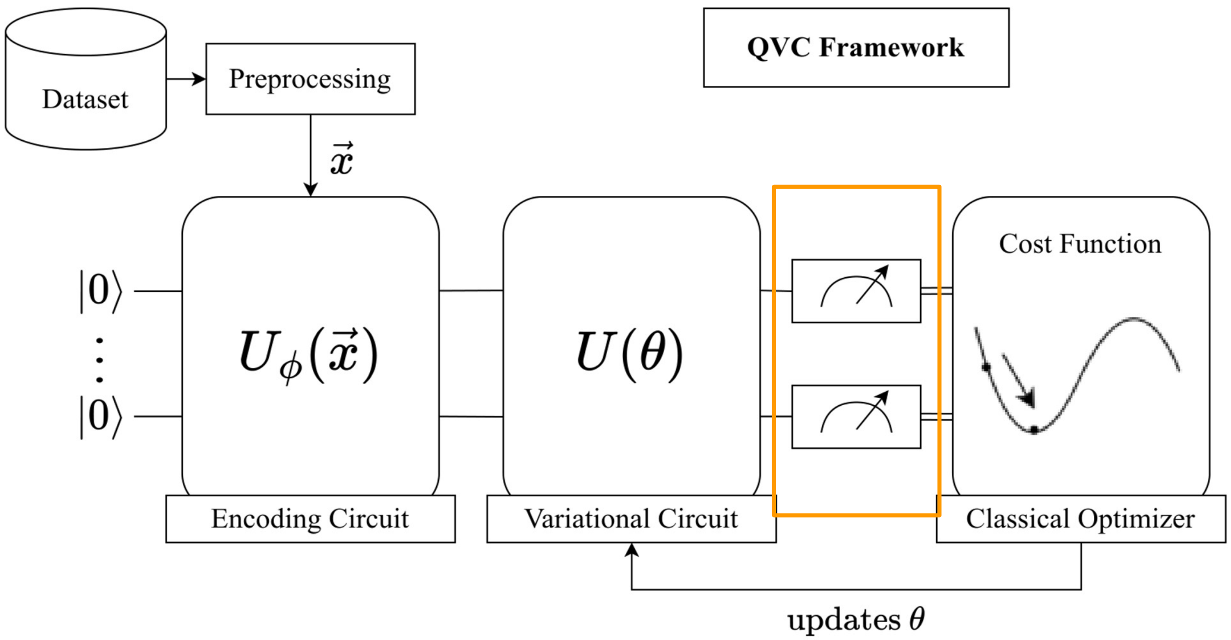
# QiML Methods: Q. Variational Alg. Simulation

- **2. Variational Circuit**
  - Quantum circuit that represents and approximates a target function for the given task
  - Layers of quantum gates parameterized by a set of "free parameters" $\theta$



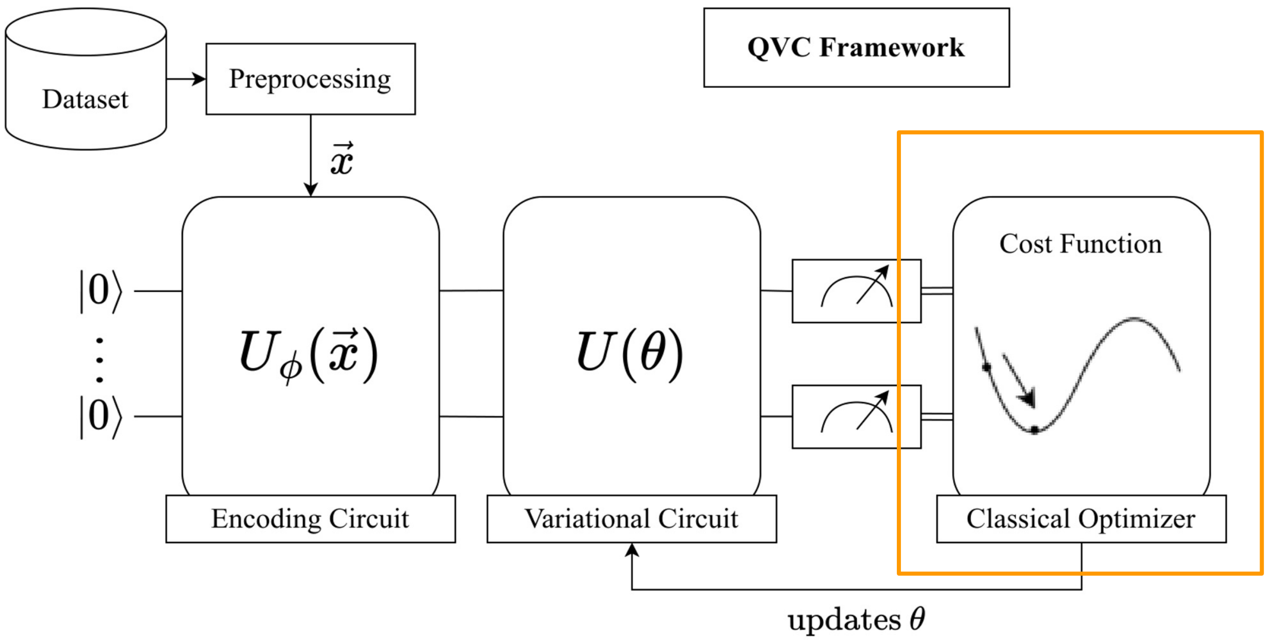
# QiML Methods: Q. Variational Alg. Simulation

- **3. Measurement**
  - Collapsing the resulting state into classical information, based on chosen basis
  - Expectation value of observable $M$: $f(\theta) = \langle 0|U^{\dagger}(\theta)MU(\theta)|0\rangle$ → scalar cost function



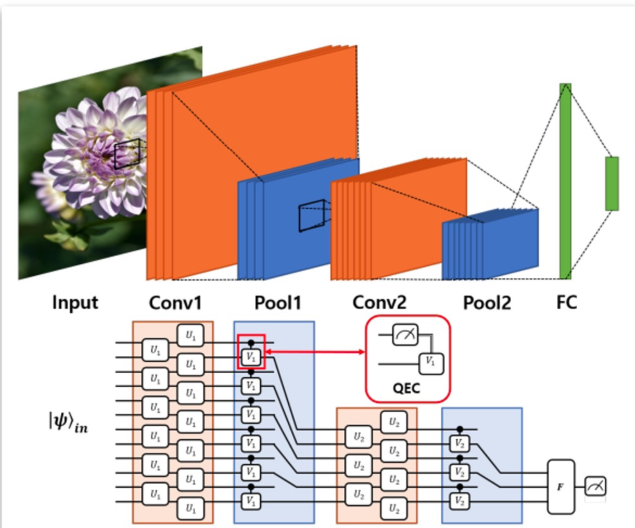
# QiML Methods: Q. Variational Alg. Simulation

- **4. Classical Optimization**
  - $f(\theta)$ optimized via gradient descent, adjusting parameters $\theta$

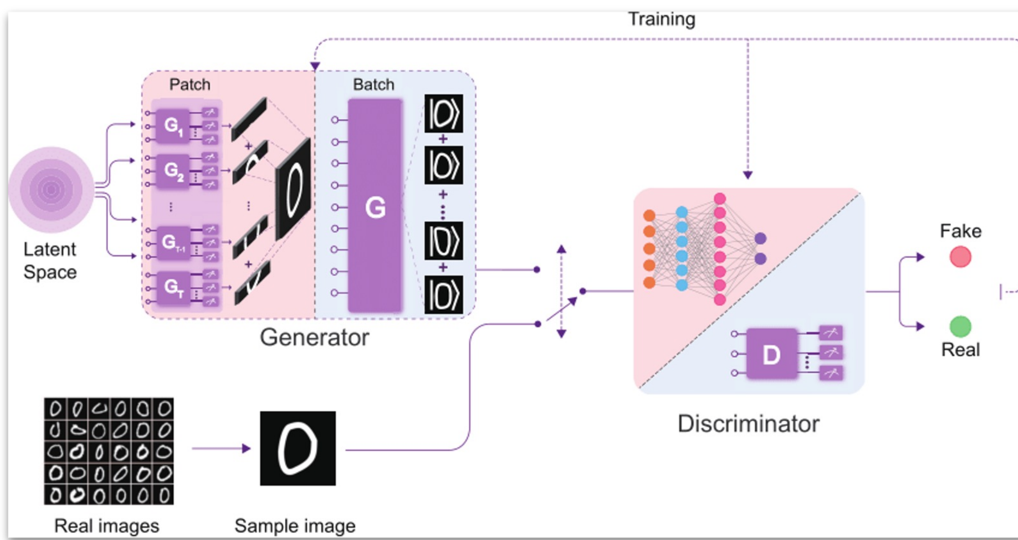


# QiML Methods: Q. Variational Alg. Simulation

- QVC framework as a basis for more complex frameworks:
  - Quantum Convolutional Neural Networks (QCNN),
  - Quantum Generative Adversarial Networks (QGAN),
  - Quantum Circuit Born Machines (QCBM), ...




QCNN




QGAN