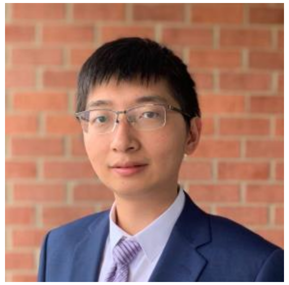
# Towards the Automatic Design of Quantum Neural Networks

**Weiwen Jiang, Ph.D.**
Assistant Professor
Electrical and Computer Engineering
George Mason University
wjiang8@gmu.edu
https://jqub.ece.gmu.edu

**Slides at https://jqub.ece.gmu.edu/categories/QF/**

# Speaker




Weiwen Jiang
Assistant Professor
Electrical and Computer Engineering (ECE)
George Mason University
Room3247, Nguyen Engineering Building
wjiang8@gmu.edu
(703)-993-5083
https://jqub.ece.gmu.edu/


- **Education Background**
  - **Chongqing University (2013-2019)**
  - **University of Pittsburgh (2017-2019)**
  - **University of Notre Dame (2019-2021)**
- **Research Interests**
  - **Automatic HW/SW Co-Design**
  - **Quantum Machine Learning**

## First HW/SW Co-Design Framework using NAS

**HW/SW Co-Design Framework**
FNAS
[DAC'19*]
[TCAD'20*]

**Application**

- **Medical Imaging**
  - NAS for Medical Image Seg. [MICCAI'20]
  - 3D Cardiac MRI Seg. [ICCAD'20]
- **NLP (Transformer)**
  - FPGA [ICCD'20]
  - Mobile [DAC'21]
  - GPU [GLSVLSI'21]
- **Graph-Based**
  - Social Net [GLSVLSI'21]
  - Drug Discovery [ICCAD'21]

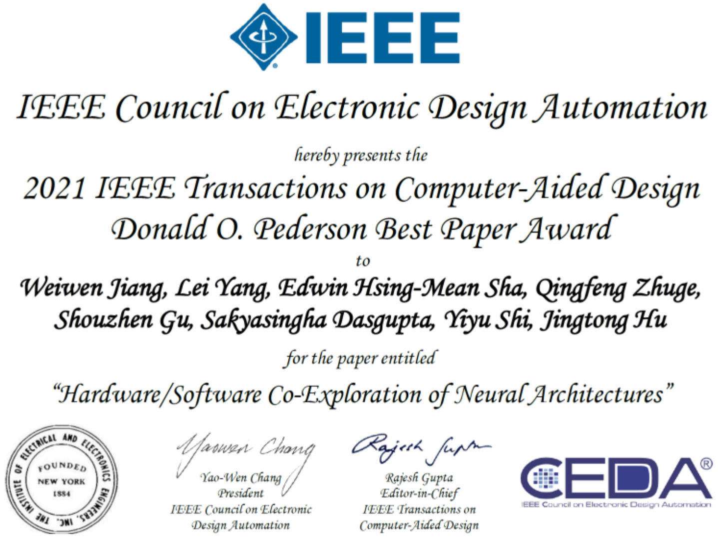
**Algorithm**

- **NAS Acc.**
  - HotNAS [CODES+ISSS'20]
- **Model Compression**
  - NAS for Quan. [ICCAD'19]
  - Compre.-Compilation [IJCAI'21]
- **Secure Infernece**
  - NASS [ECAI'20]
  - BUNET [MICCAI'20]

**Hardware**

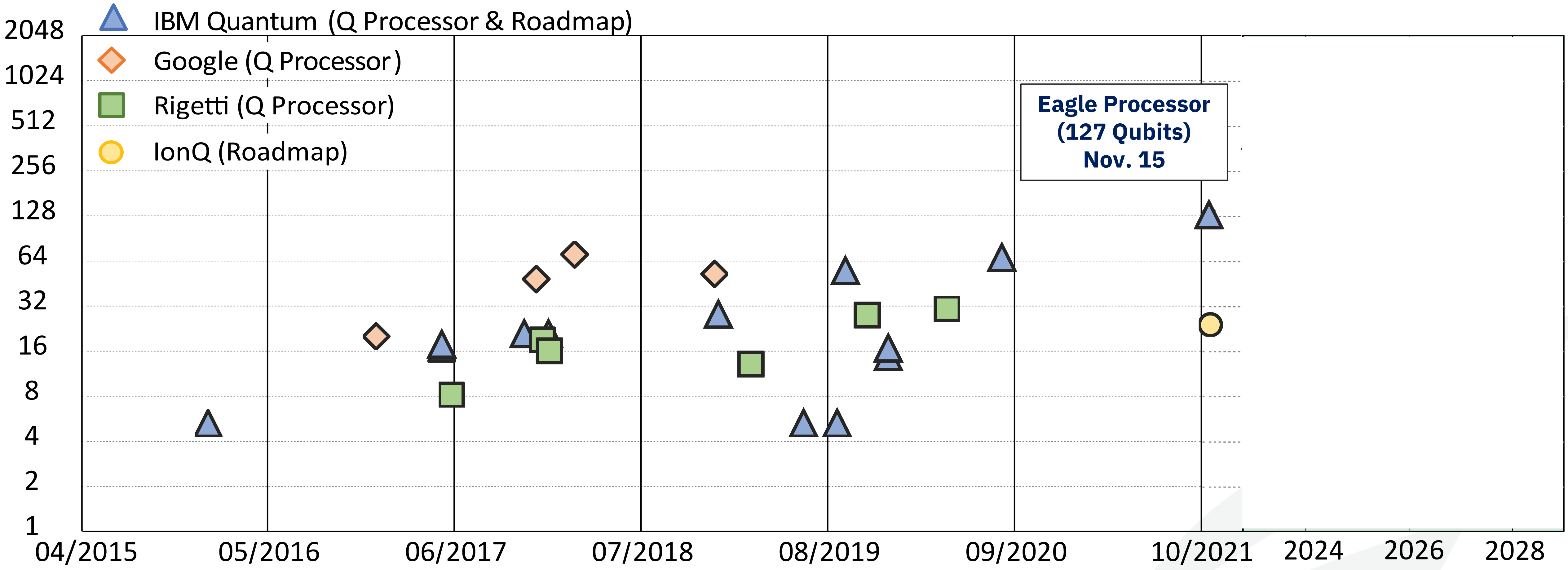
- **FPGA**
  - XFER [CODES+ISSS'19*]
- **ASIC**
  - NANDS [ASP-DAC'20*]
  - ASICNAS [DAC'20]
- **Computing-in-Memory**
  - Device-Circuit-Arch. [IEEE TC'20]

**Best Paper Award:**

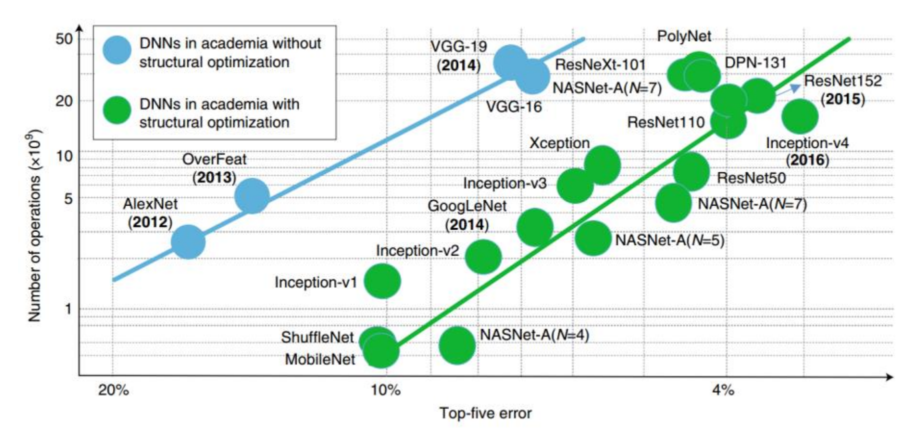

IEEE
IEEE Council on Electronic Design Automation
hereby presents the
2021 IEEE Transactions on Computer-Aided Design
Donald O. Pederson Best Paper Award
to
Weiwen Jiang, Lei Yang, Edwin Hsing-Mean Sha, Qingfeng Zhuge, Shouzhen Gu, Sakyasingha Dasgupta, Yiyu Shi, Jingtong Hu
for the paper entitled
"Hardware/Software Co-Exploration of Neural Architectures"
Yao-Wen Chang
President
IEEE Council on Electronic Design Automation
Rajesh Gupta
Editor-in-Chief
IEEE Transactions on Computer-Aided Design

**Best Paper Nominations:**

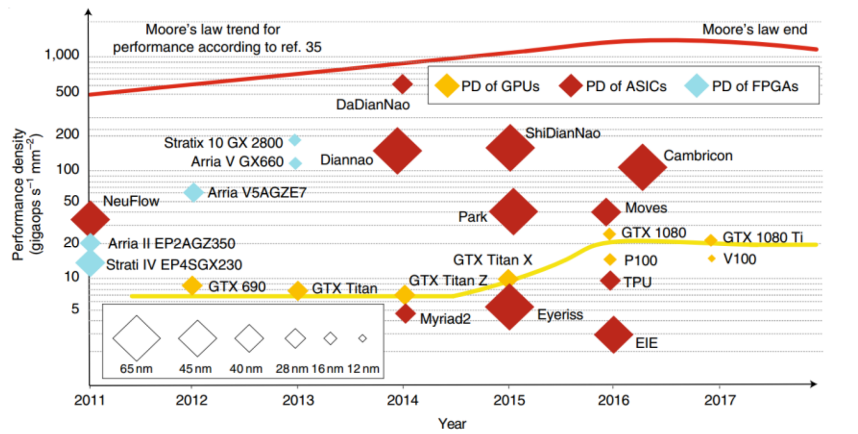
# Quantum Computers Have Come to Our Life



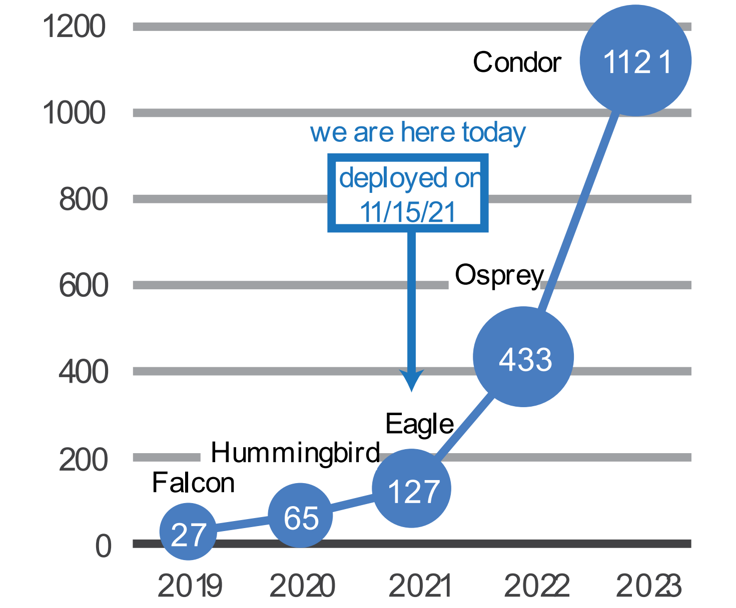
# Quantum Has Potential For Neural Network



Deep neural network grows exponentially



Perf. of classical computing stops increasing



Quantum computer grows exponentially

## Fundamental questions:

- Can we implement Neural Network on Quantum Computers?
- Can we achieve benefits in doing so?

## Further questions:

- **[Q1]** What is the best neural network architecture for quantum acceleration?
- **[Q2]** What is the problem for near-term quantum computing, i.e., in NISQ era?

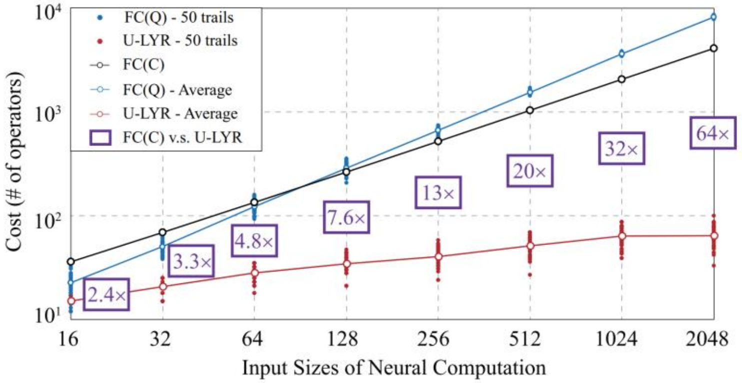
# Preliminary Results Answered to Fundamental Questions

**Fundamental questions:**

- Can we implement Neural Network on Quantum Computers?
- Can we achieve benefits in doing so?


QuantumFlow

$$O(N) \rightarrow O(log^3 N)$$




**Paper Published at:** nature COMMUNICATIONS

**Invited Contribution and Tutorial Talks at:**

IBM Quantum Summit
September 15-17, 2020

IEEE QUANTUM WEEK
IEEE International Conference on Quantum Computing and Engineering — QCE21

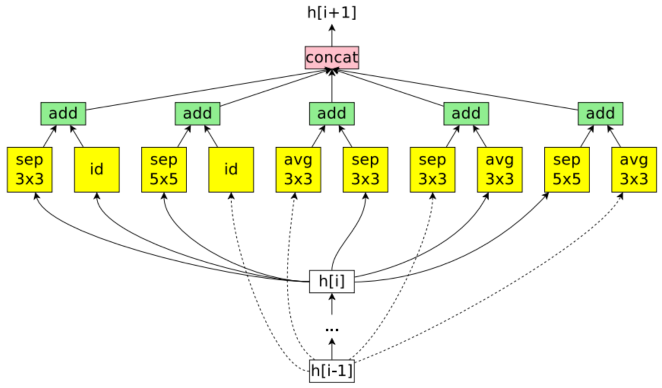
EMBEDDED SYSTEMS WEEK
OCTOBER 10-15, 2021 | VIRTUAL CONFERENCE

IEEE/ACM 2021 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN ICCAD
40th Edition

# Towards the Automatic Design of Quantum Neural Networks

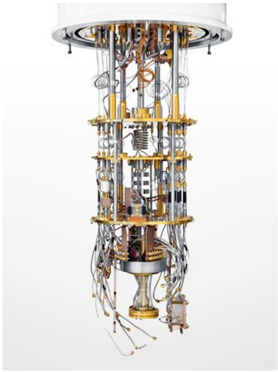
## Further questions:

- **[Q1]** What is the best neural network architecture for quantum acceleration?



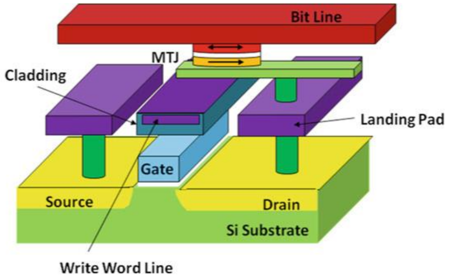
Convolution Cell

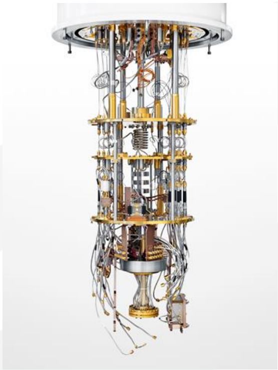
- $3 \times 3$ depthwise-separable conv
- $5 \times 5$ depthwise-separable conv
- $3 \times 3$ atrous conv with rate 2
- $5 \times 5$ atrous conv with rate 2
- $3 \times 3$ average pooling
- $3 \times 3$ max pooling
- skip connection
- no connection (zero)



?

- **[Q2]** What is the problem for near-term quantum computing, i.e., in NISQ era?



?

FPGA Error: $10^{-15}$

GPU Error: $10^{-15}$

STT-RAM Error: $10^{-9}$

Qubit Error: $10^{-4} \sim 10^{-2}$

# Outline

- Background
- **Co-Design: from Classical to Quantum**
- QuantumFlow for automatic design of quantum neural networks
  - Quantum Neurons
  - QF-Mixer for **[Q1]**
- Other Recent works and conclusion
  - QF-RobustNN for **[Q2]**
  - QFNN Library

# Co-Design

**Given:**

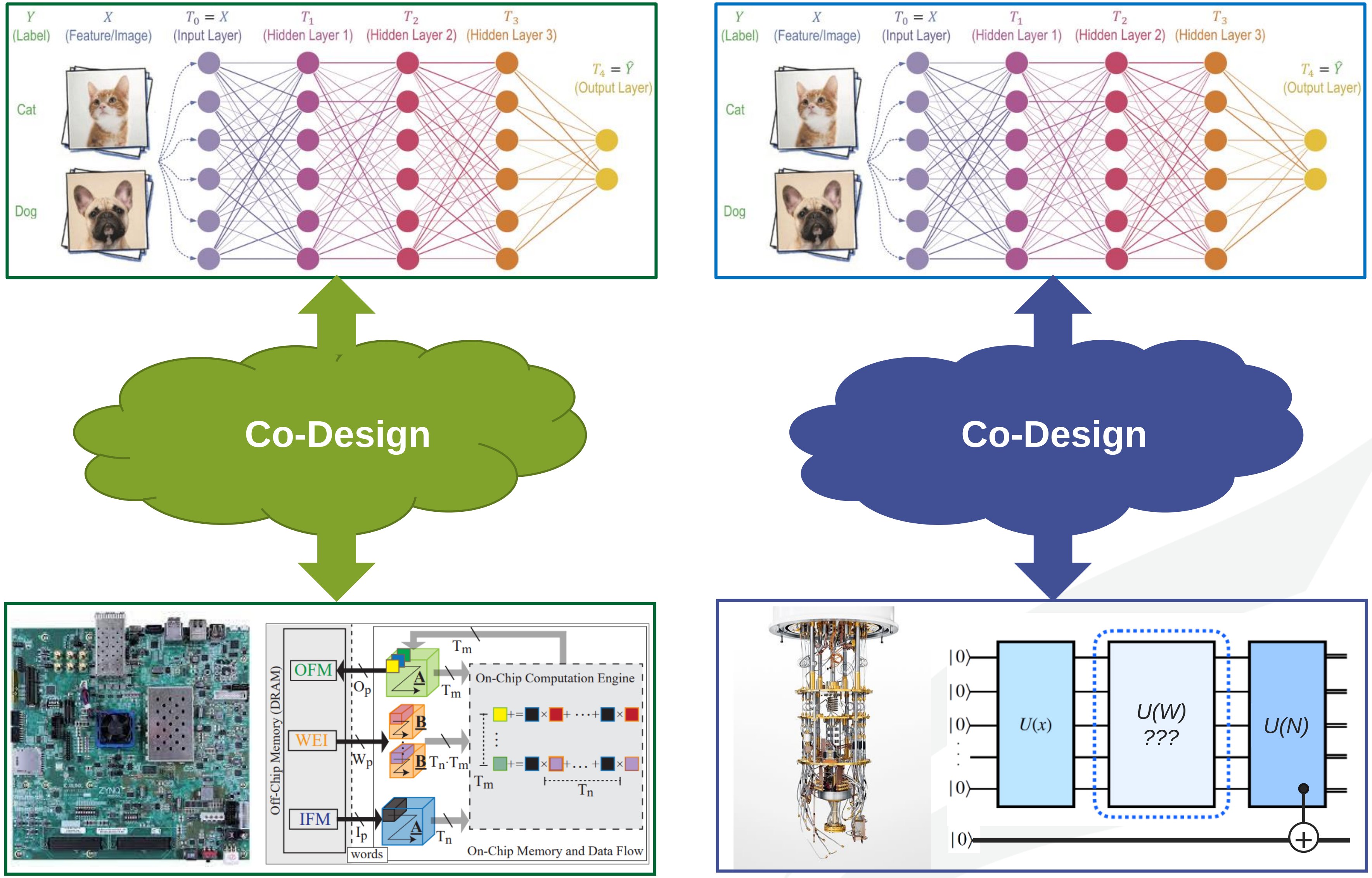
- Dataset (e.g., ImageNet)
- ML Task (e.g., classification)
- HW (e.g., FPGA spec.)

**Do:**

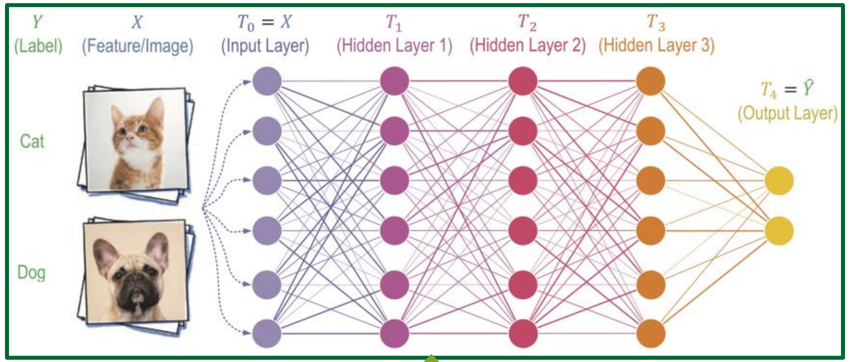
- Neural network design
- FPGA design

**Objective:**

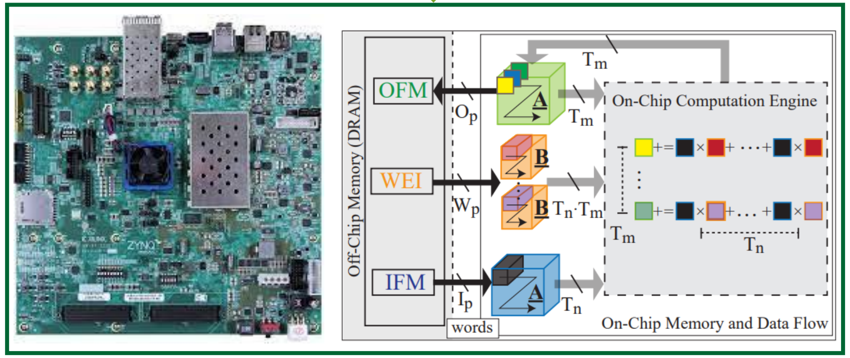
- Accuracy
- Latency
- Energy
- …



# My Previous Background: Co-Design of Neural “Architectures”

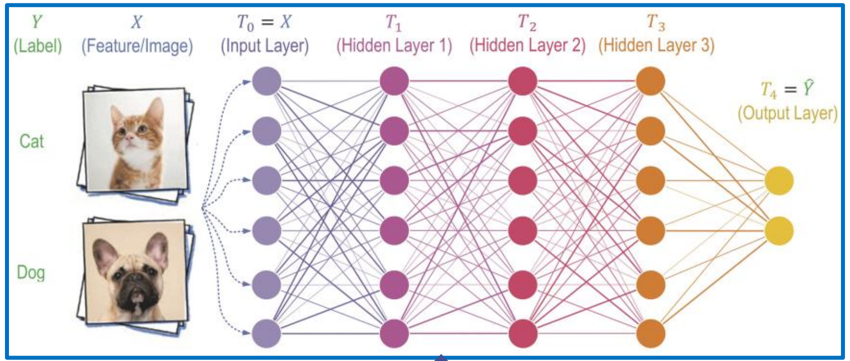



Co-Design

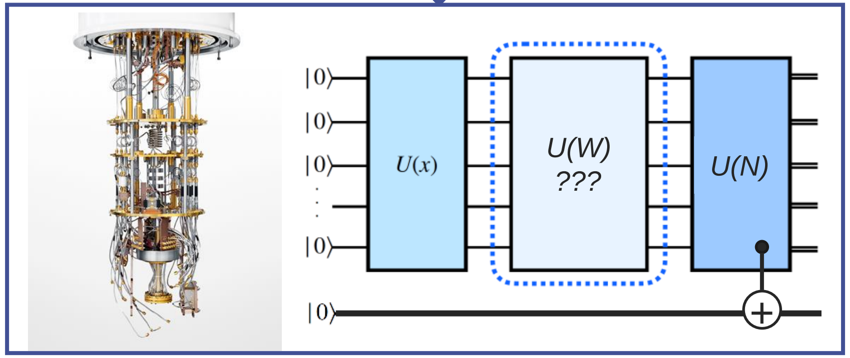



- What is the best **Neural Network Architecture** for FPGAs
- Model optimization (pruning and quantization)?
- Library
  - Co-Design Framework (e.g., Our FNAS)
    - Network exploration — NAS (Google)
    - Programming library — DNNBuilder (UIUC)
    - Place & Route — DNN on FPGA (UCLA)
- Mapping and scheduling?
- What is the best **FPGA Architecture** for neural networks

# Current Works: Co-Design of Neural Networks and Quantum Circuit




Co-Design




- What is the best **Neural Network Architecture** for QC?
- ......
- Library

**Co-Design Framework**
**QuantumFlow**

| | |
|---|---|
| Network exploration | QF-Mixer |
| Programming library | QFNN |
| Logic-physical Compile | QF-RobustNN |

- ......
- What is the best **QC design** for neural networks

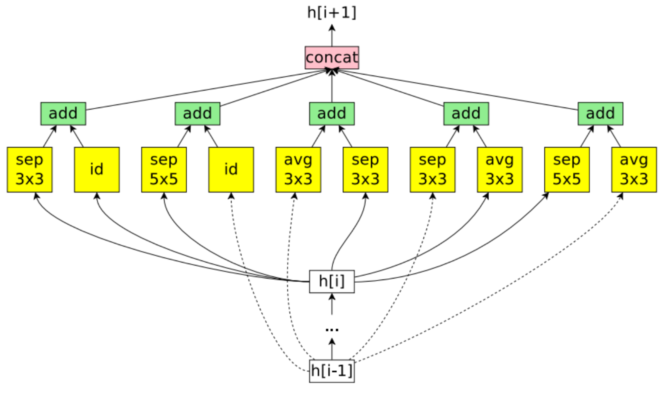
# Outline

- Background
- Co-Design: from Classical to Quantum
- **QuantumFlow for automatic design of quantum neural networks**
  - **Quantum Neurons**
  - **QF-Mixer for [Q1]**
- Other Recent works and conclusion
  - QF-RobustNN for **[Q2]**
  - QFNN Library

# Towards the Automatic Design of Quantum Neural Networks

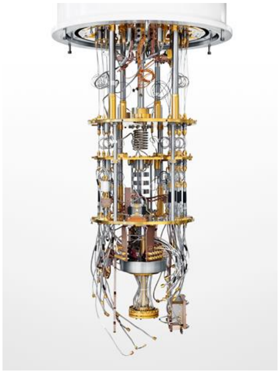
## Further questions:

- **[Q1]** What is the best neural network architecture for quantum acceleration?



Convolution Cell

- $3 \times 3$ depthwise-separable conv
- $5 \times 5$ depthwise-separable conv
- $3 \times 3$ atrous conv with rate 2
- $5 \times 5$ atrous conv with rate 2
- $3 \times 3$ average pooling
- $3 \times 3$ max pooling
- skip connection
- no connection (zero)


?

- **[Q2]** What is the problem for near-term quantum computing, i.e., in NISQ era?







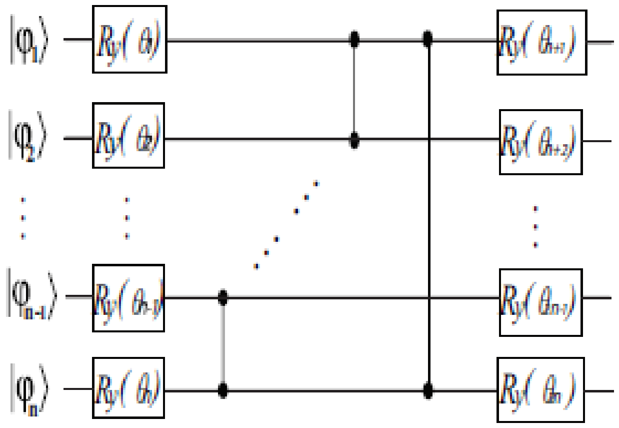



?

FPGA Error: $10^{-15}$ | GPU Error: $10^{-15}$ | STT-RAM Error: $10^{-9}$ | Qubit Error: $10^{-4}$~$10^{-2}$

# Existing Quantum Neuron Designs

- **Variational quantum circuit (VQC)-based neuron**



(b) V-NEU

**V-Neuron (V-NEU)**

- A widely used quantum neuron
- Reuse the input qubits as output qubits

Advantage

- **Real-valued weights**

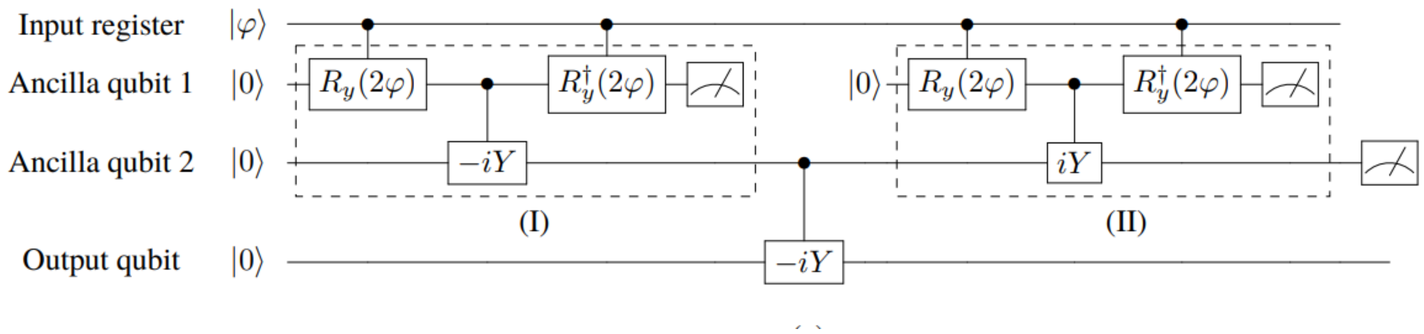
Disadvantage

- **Linear classifier**
- **Cannot be extended to multiple nonlinear layers with low cost**

- Make use of the entanglement from quantum computing to increase the model complexity

# Existing Quantum Neuron Designs

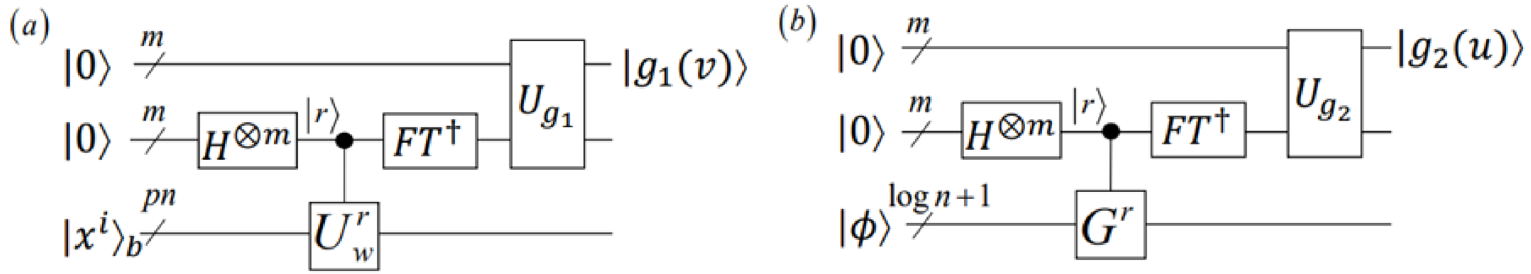
- **Q-Neuron**




Disadvantage

- **Data encoding: one-to-one mapping (almost impossible to achieve quantum advantage)**
- **Repeat-until-success to build non-linear function (Inefficient)**

- **Q-Non-Linear Neuron**



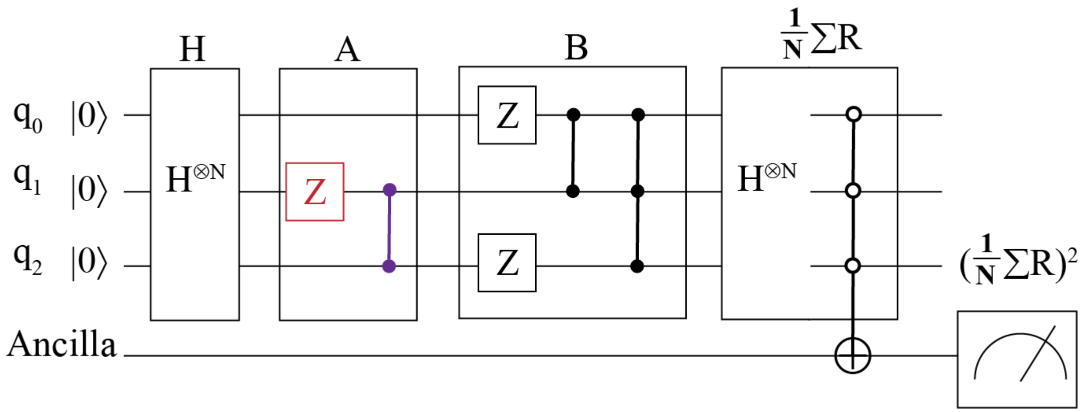

**Apply Boolean function to realize any non-linear function**

Disadvantage

- **Quantum advantage cannot be achieved**

# Existing Quantum Neuron Designs

- **Q-Artificial Neuron**




Implementing binary perceptron in quantum computer

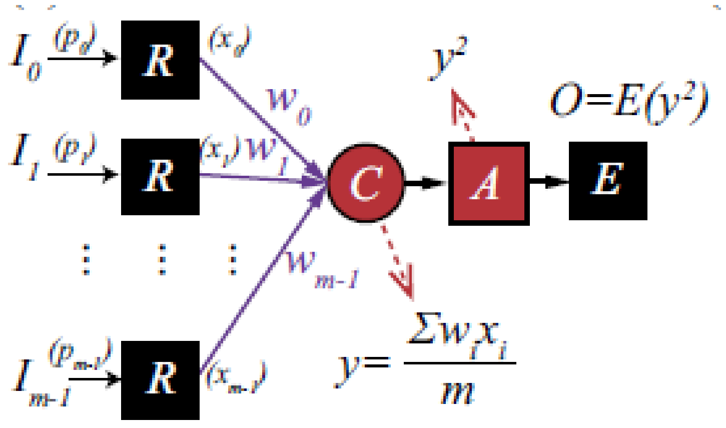
Disadvantage

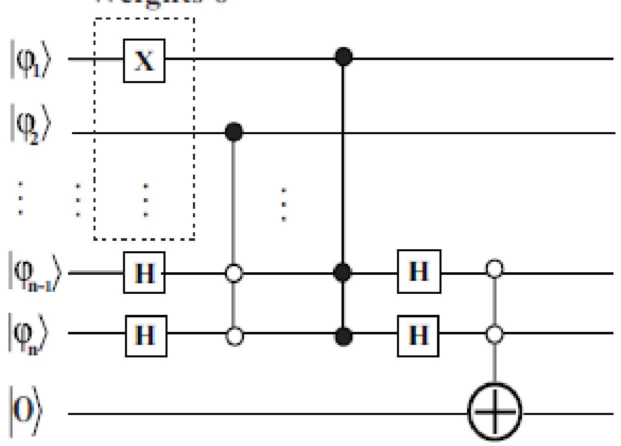
- **Both inputs and weights are binary**

Z. wang, et al. Exploration of Quantum Neural Architecture by Mixing Quantum Neuron Designs

# Existing Quantum Neuron Designs

- **Customized neurons of QuantumFlow**




(d) P-NEU

**P-Neuron (P-NEU)**

- Input encoding: *Probability encoding (Angle encoding)*
- Output encoding: *Probability encoding*

Advantage

- **Easy to be stacked to form multiple nonlinear layers**

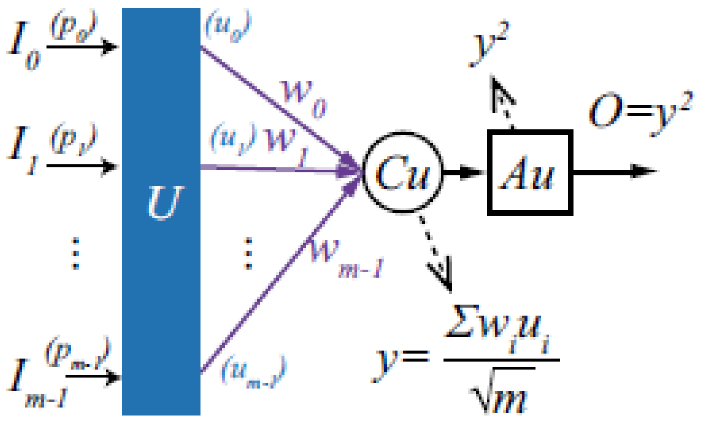
Disadvantage

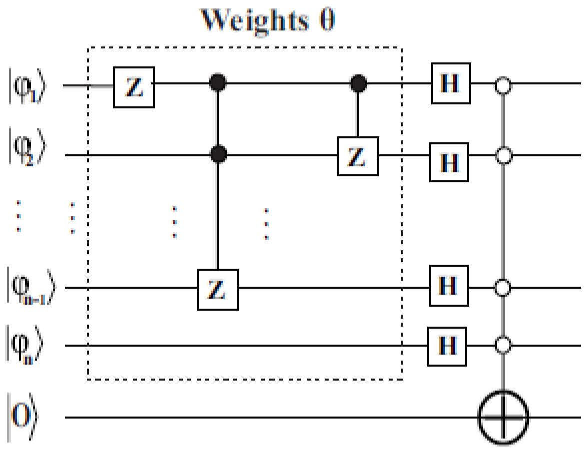
- **Binary weights**

[1] W. Jiang, et al. A Co-Design Framework of Neural Networks and Quantum Circuits Towards Quantum Advantage, Nature Communications

# Existing Quantum Neuron Designs

- **Customized neurons of QuantumFlow**




(e) U-NEU

**U-Neuron (U-NEU)**

- Input encoding: *Amplitude encoding*
- Output encoding: *Probability encoding*

| Advantage |
|---|
| • **It could be connected to P-Neuron seamlessly** <br> • **It achieves quantum advantage** |

| Disadvantage |
|---|
| • **Binary weights** |


[1] W. Jiang, et al. A Co-Design Framework of Neural Networks and Quantum Circuits Towards Quantum Advantage, Nature Communications

# Motivation

- **Mixing/connecting different neurons in an NN** could improve the performance
- For example: VQC and neurons of QuantumFlow are complementary

## VQC

**Advantage**

- Real-valued weights

**Disadvantage**

- Linear classifier
- Cannot be extended to multiple nonlinear layers with low cost

**Mixing**

**1+1 > 2**

## QuantumFlow

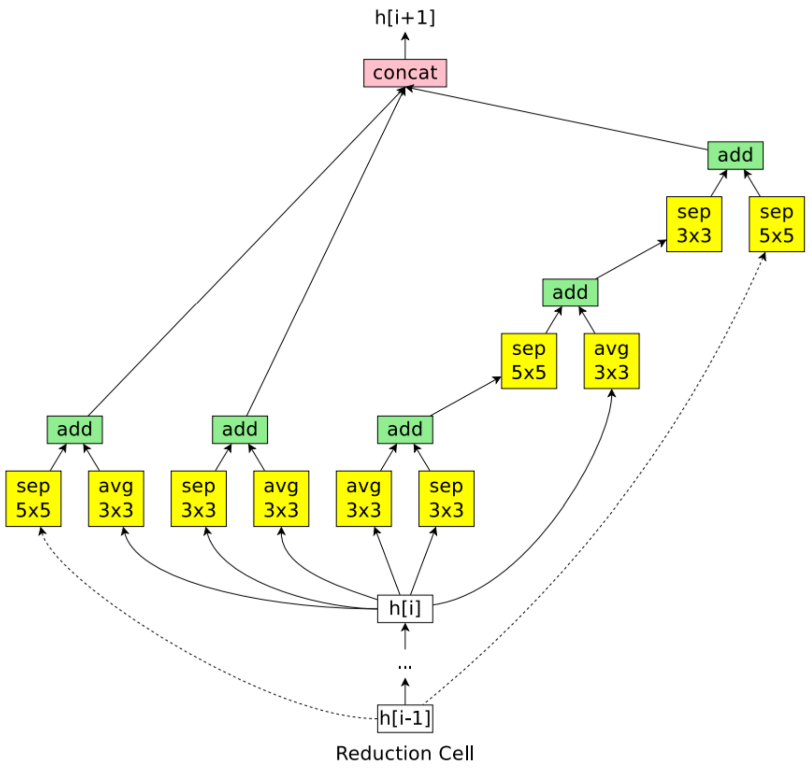
**Disadvantage**

- Binary weights

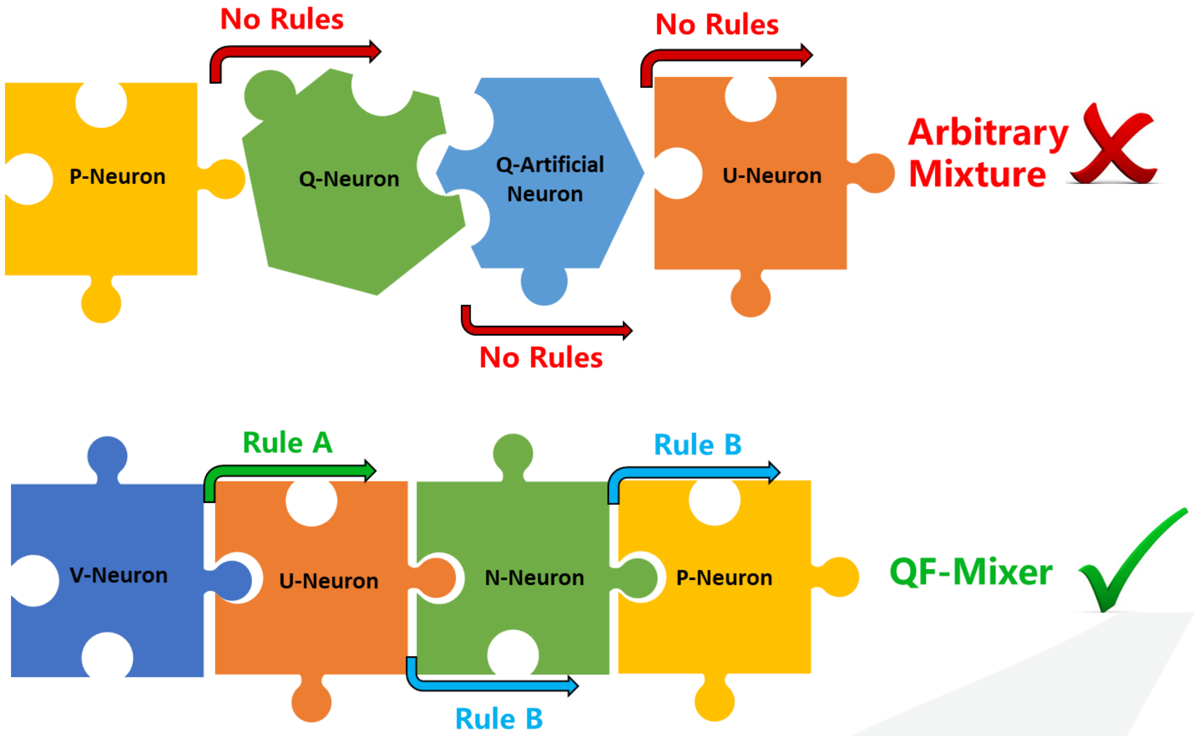
**Advantage**

- Easy to be extended to multiple nonlinear layers w/t measurement

# Challenges




**Different operators/neurons in classical computing can be connected seamlessly.**

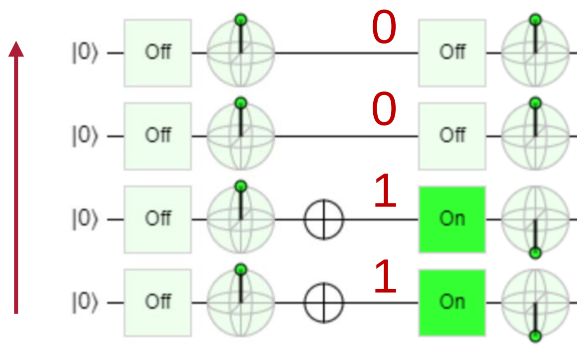



**Connect different quantum neurons may incur high overhead; may not be seamless.**

# Challenges: Designs May Base On Different Data Encoding

- **Can we encode an arbitrary number into quantum computer? Is it efficient?**

- **Yes / No**




**Yes, we can!**

Number: 12

Number: 1.5

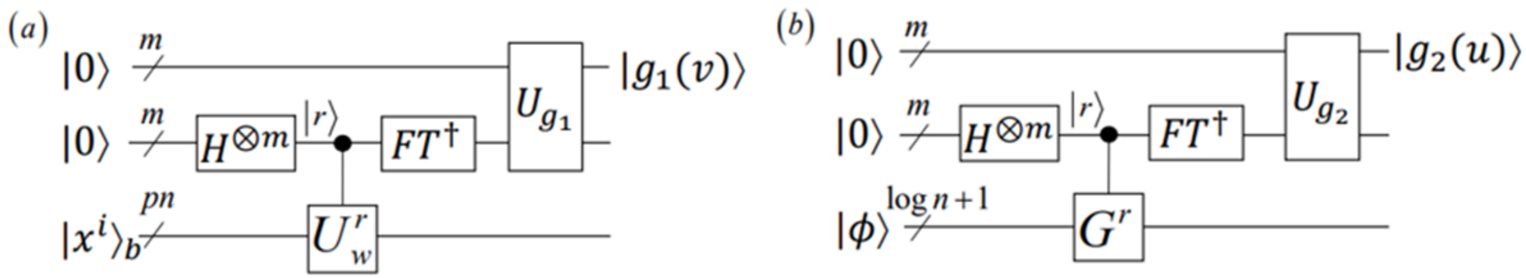
Number: 0.75

**No, because it uses too many qubits!**

This encoding is similar to classical bits, where each qubit is regarded as a binary number!

***1-to-N mapping! (Boolean Function)***

## Q-Non-Linear Neuron




**Apply Boolean function to realize any non-linear function**

# Challenges: Designs May Base On Different Data Encoding

- Can we encode an arbitrary number into quantum computer? Is it efficient?
  - Yes / No




**Yes, we can!**

Number: 12

Number: 1.5

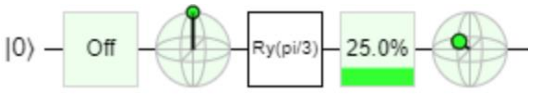
Number: 0.75

**No, because it uses too many qubits!**

This encoding is similar to classical bits, where each qubit is regarded as a binary number!

***1-to-N mapping! (Boolean Function)***

- **Can we take use of superposition of qubits to encode data? Is this solution perfect?**
  - **Yes / No**



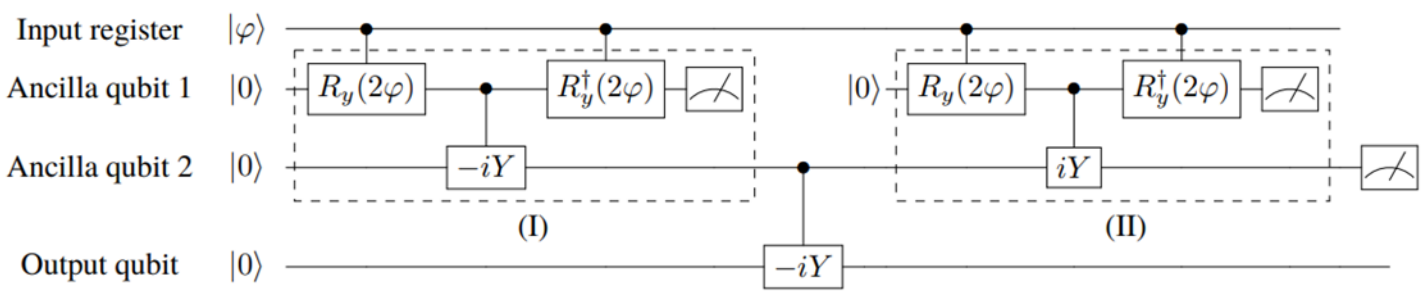

**Yes, we can!**

0.25 in one qubit

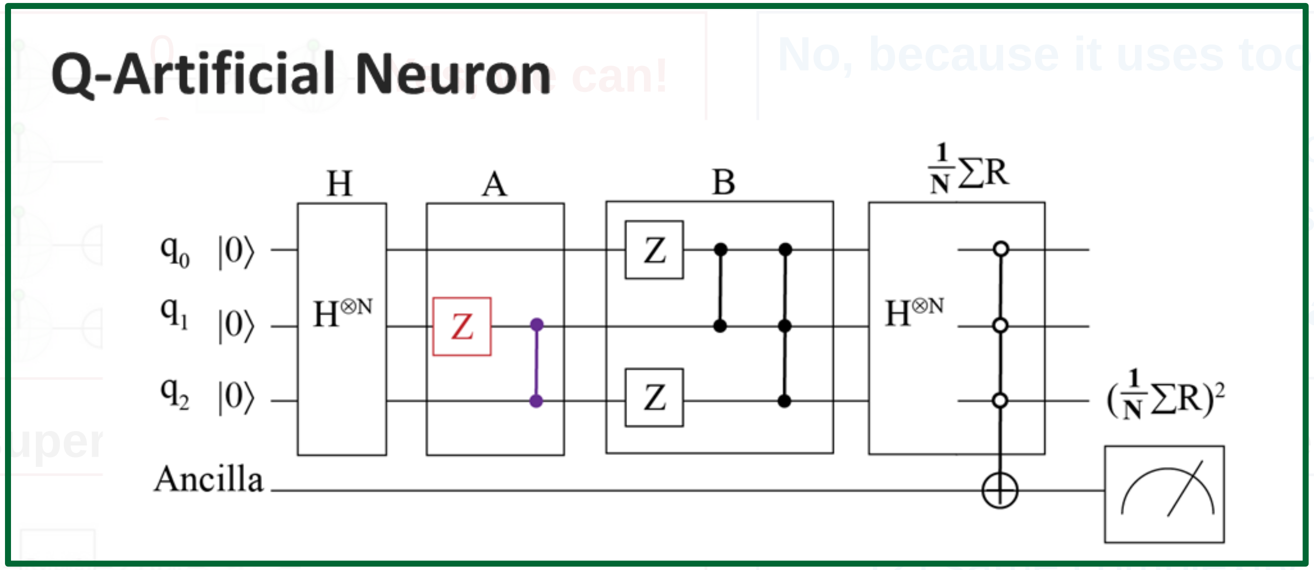
**No,** (1) data needs in the range of [0,1]!
(2) same complexity *O(1)* as classical

***1-to-1 mapping! (Probability Encoding)***

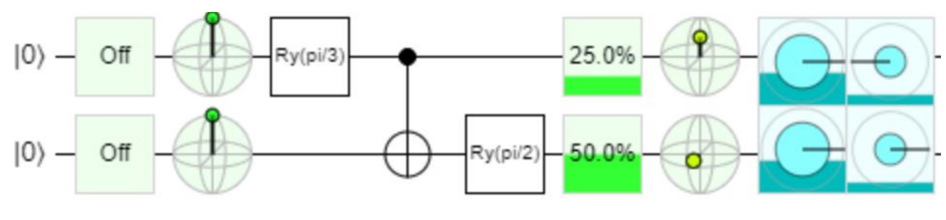
## Q-Neuron



# Challenges: Designs May Base On Different Data Encoding




- **Can we take use of entanglement of qubits to encode data? Is this solution perfect?**
  - **Yes** / **No**




**Yes,** 4 # in 2 qubits

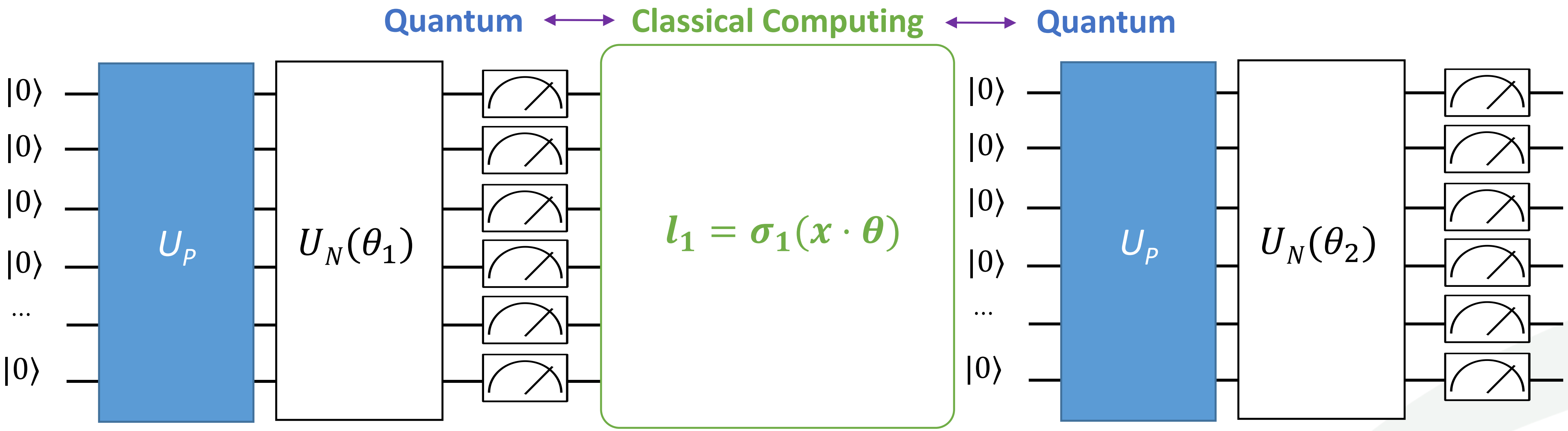
$$\begin{bmatrix} +0.61 & -0.35 \\ +0.61 & +0.35 \end{bmatrix}$$

**No,** (1) sum of the square of data need to be 1
(2) may have high cost to encode data

***N-to-logN mapping! (Amplitude Encoding)***

# Challenges

- **Inconsistent data encoding will lead to high-cost quantum-classical communication**

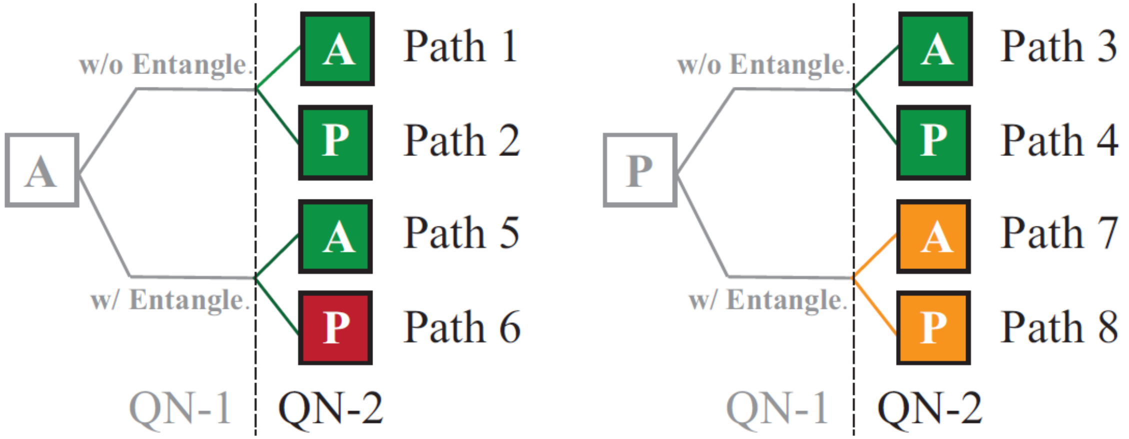


# QF-Mixer

# Encoding: Boolean *vs.* Probability *vs.* Amplitude

| Data Encoding | # of Qubit (C v.s. Q) | Data Limitation | Encoding Complexity |
|---|---|---|---|
| Boolean Encoding | O(1) v.s. O(N) | **Almost No!** | **Low** |
| Probability Encoding | O(1) v.s. O(1) | [0,+1] | **Low** |
| Amplitude Encoding | **O(N) v.s. O(*log*N)** | [-1,+1] and $\sum x^2 = 1$ | High |

# Design Principles




- **P: Probability encoding**
- **A: Amplitude encoding**

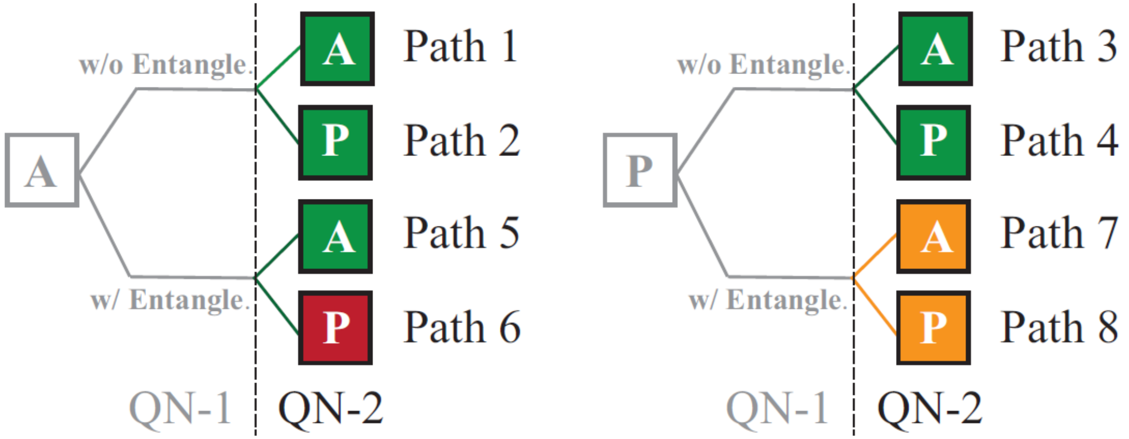
## Output qubits of QN-1 are not entangled

- **Principle 1 (*Path 1-4*)**
  - The output qubits from QN-1 are decoupled with the output qubits of its previous layers.
  - Conclusion: Feasible

## Output qubits of QN-1 are entangled

- **Principle 2 (*Path 5*)**
  - W/o probability encoding involved, there is no requirement on the decoupling
  - Conclusion: Feasible

# Design Principles

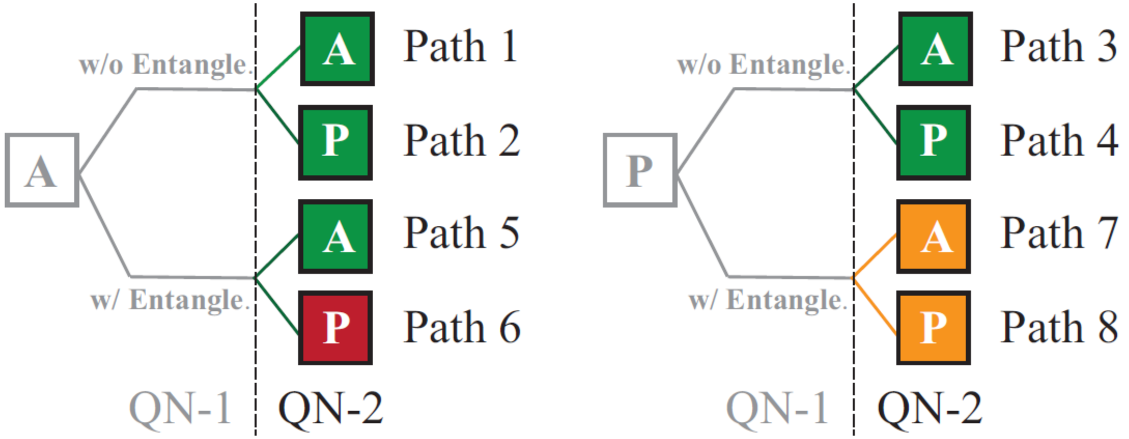



- **P: Probability encoding**
- **A: Amplitude encoding**

## Output qubits of QN-1 are entangled

- **Principle 3 (*Path 6*)**
  - When QN-2 is a neuron in the first layer of a QNN and uses probability encoding, the input qubits are required to be independent.
  - Based on the goal of consistency, when QN-1 is the neuron in other layers, independence requirement should also hold.
  - Conclusion: Infeasible

# Design Principles



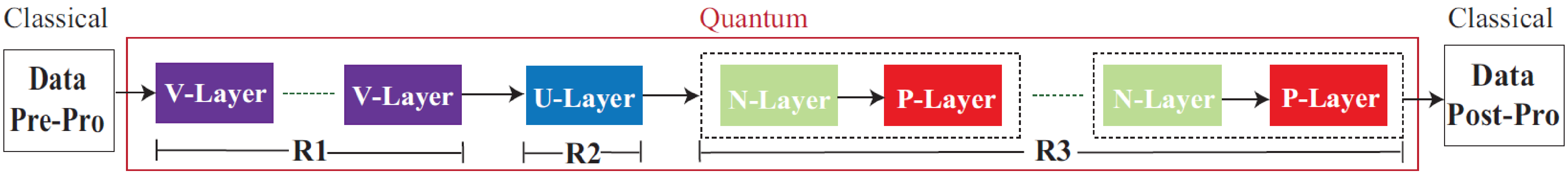

- **P: Probability encoding**
- **A: Amplitude encoding**

## Output qubits of QN-1 are entangled

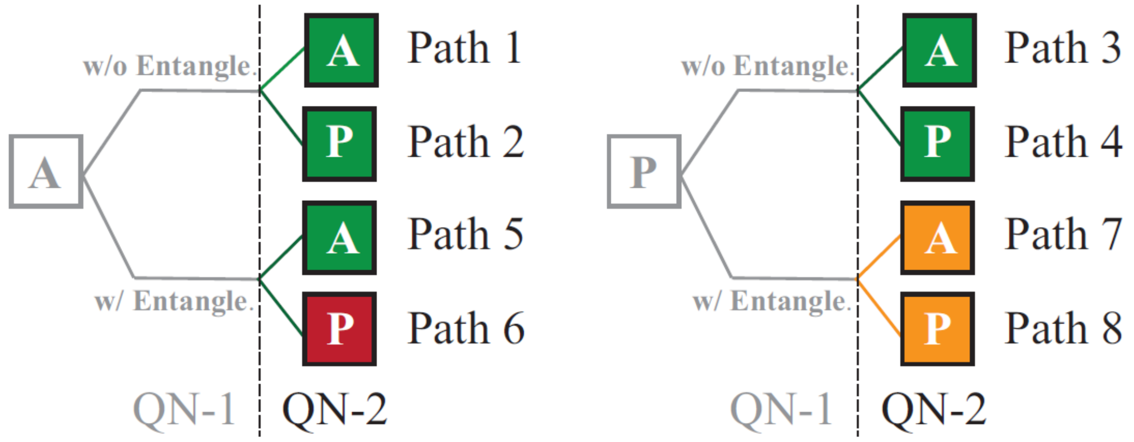
- **Principle 4 (*Path 7*)**
  - Conclusion: Conditional
  - Condition: The inputs qubits of QN-1 are reused by the output qubits, such as V-Layer.
- **Principle 5 (*Path 8*)**
  - Conclusion: Conditional
  - Condition:
    - Output qubits of QN-1 are used as control end without phase kickback
    - The operations on the output qubits of QN-1 only rotates them around X-axis

# QF-MixNN

- **Pure quantum architecture**
  - The neural computation is conducted purely on quantum devices
  - Data pre-processing and post-processing are on classical devices
- **V-Layer should be the first**
  - Applying amplitude encoding to the input data
  - The extreme case is V-Layers only
  - Larger *R1* provides more real-valued weights
- **Multi-layer QNN can be formed**
  - U-Layer provides the non-linearity to the V-Layers, which will be added if *R2 = 1*
  - Larger *R3* corresponds to more non-linear layers



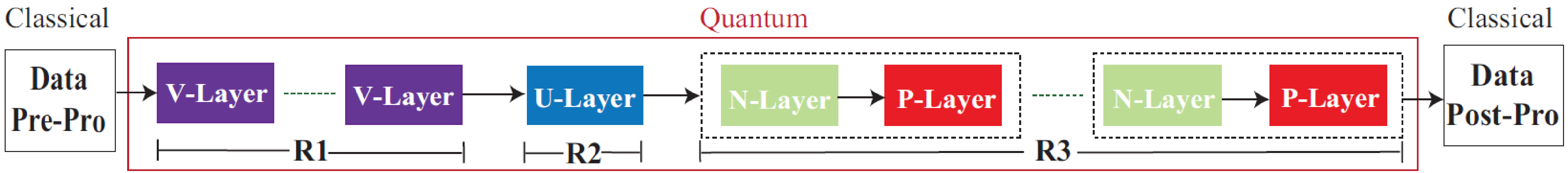
# The Design of QF-MixNN Follows the Principles




| Neuron Type | Input Encoding Method | Output Encoding Method |
|---|---|---|
| U-Neuron | Amplitude | Probability |
| V-Neuron | Amplitude | Amplitude/Probability |
| P-Neuron | Probability | Probability |
| N-Neuron | Probability | Probability |

- V-NEU to V-NEU: Path 5
- V-NEU to U-NEU: Path 5
- U-NEU to N-NEU: Path 8
- N-NEU to P-NEU: Path 8
- V-NEU to P-NEU: Path 8

**Feasible!**



# QF-MixNN Achieves the Best Accuracy on MNIST

TABLE I
EVALUATION OF QNNS WITH DIFFERENT NEURAL ARCHITECTURE

| Architecture | | MNIST-2† | MNIST-3† | MNIST-4‡ | MNIST-5‡ | MNIST§ |
|---|---|---|---|---|---|---|
| VQC (V×R1) | | **97.91%** | 90.09% | 93.45% | 91.35% | 52.77% |
| QuantumFlow | | 95.63% | 91.42% | 94.26% | 89.53% | 69.92% |
| QF-MixNN | V+U | 97.36% | **92.77%** | **94.41%** | **93.85%** | 88.46% |
| | V+U+P | 87.45% | 82.9% | 92.44% | 91.56% | **90.62%** |
| | V+P | 91.72% | 76.93% | 88.43% | 85.02% | 49.57% |

Input resolutions: † $4 \times 4$; ‡ $8 \times 8$; § $16 \times 16$;

- **Non-linearity is important**. A linear decision boundary is not sufficient for complicated tasks.
- **Real-valued weight is helpful**. It increases the representation capability of QNN significantly.

- QF-MixNN takes the advantage of both VQC-based QNN and QF-Net from Quantumflow.
- Achieve highest accuracy for full set of MNIST dataset

# Outline

- Background
- Co-Design: from Classical to Quantum
- QuantumFlow for automatic design of quantum neural networks
  - Quantum Neurons
  - QF-Mixer for [Q1]
- **Other Recent works and conclusion**
  - **QF-RobustNN for [Q2]**
  - **QFNN Library**

# Towards the Automatic Design of Quantum Neural Networks

## Further questions:

- **[Q1]** What is the best neural network architecture for quantum acceleration?

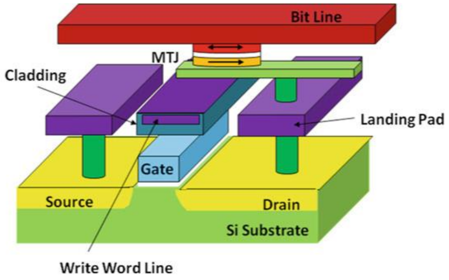


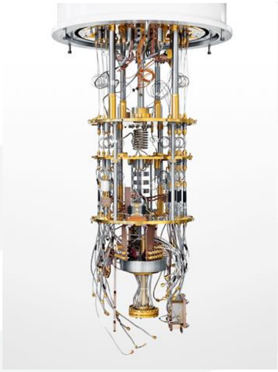

- $3 \times 3$ depthwise-separable conv
- $5 \times 5$ depthwise-separable conv
- $3 \times 3$ atrous conv with rate 2
- $5 \times 5$ atrous conv with rate 2
- $3 \times 3$ average pooling
- $3 \times 3$ max pooling
- skip connection
- no connection (zero)



?

- **[Q2]** What is the problem for near-term quantum computing, i.e., in NISQ era?



?

FPGA Error: $10^{-15}$

GPU Error: $10^{-15}$

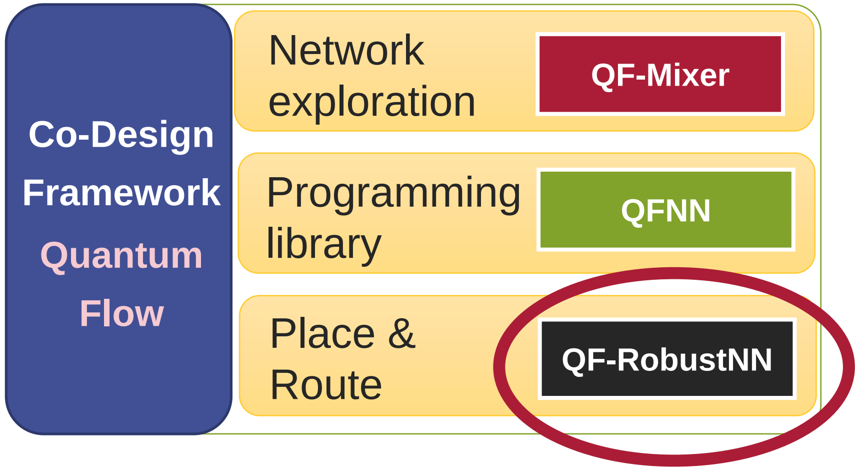
STT-RAM Error: $10^{-9}$

Qubit Error: $10^{-4}$~$10^{-2}$

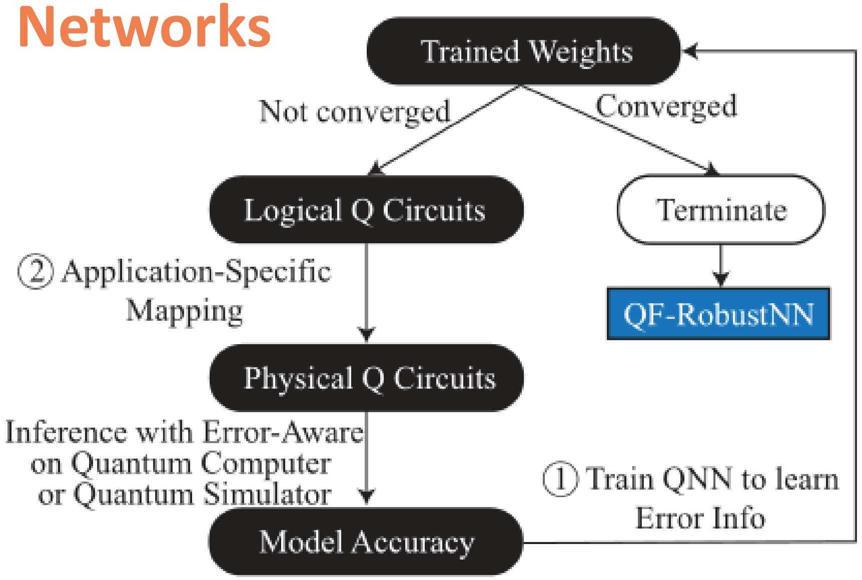
# On-Going Works in Building Quantum NN Co-Design Stack and Next

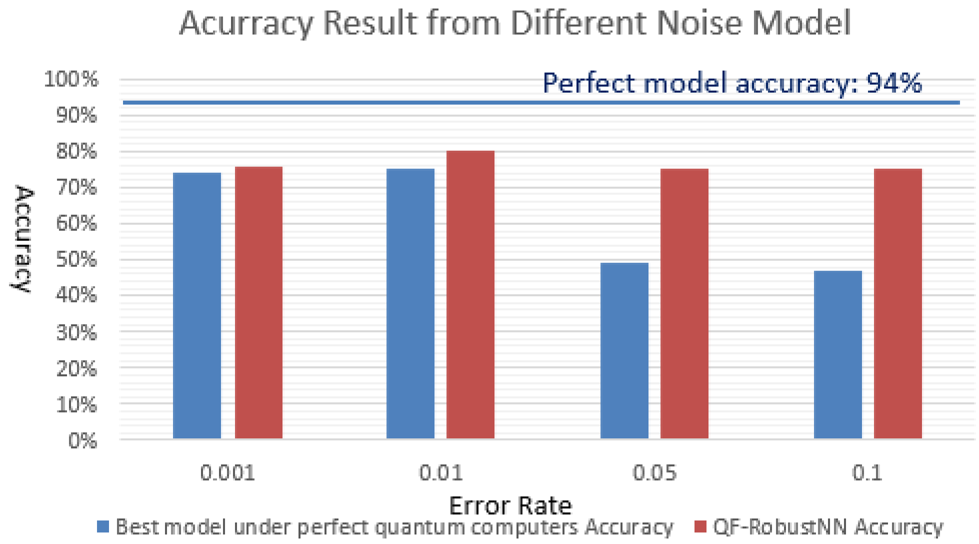
## Current works:

### Quatnum NN Co-Design Stack




## The first noise-aware training for Quantum Neural Networks




**Can Noise on Qubits Be Learned in Quantum Neural Network? A Case Study on QuantumFlow**

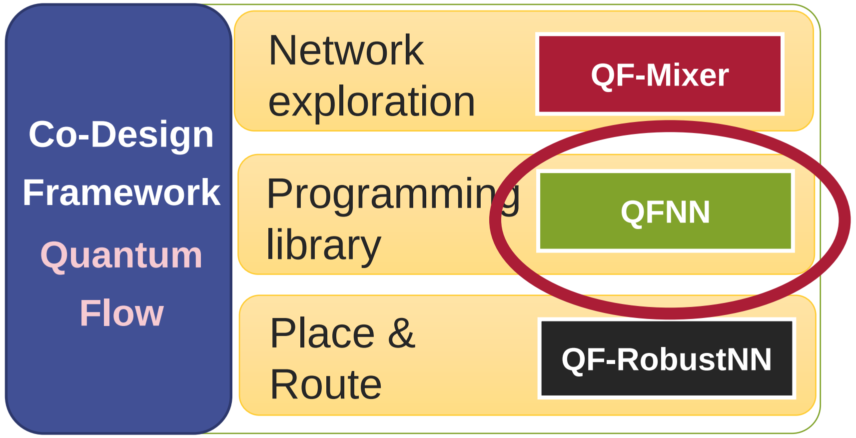
Z. Liang, Z. Wang, J. Yang, L. Yang, J. Xiong, Y. Shi, **W. Jiang**,

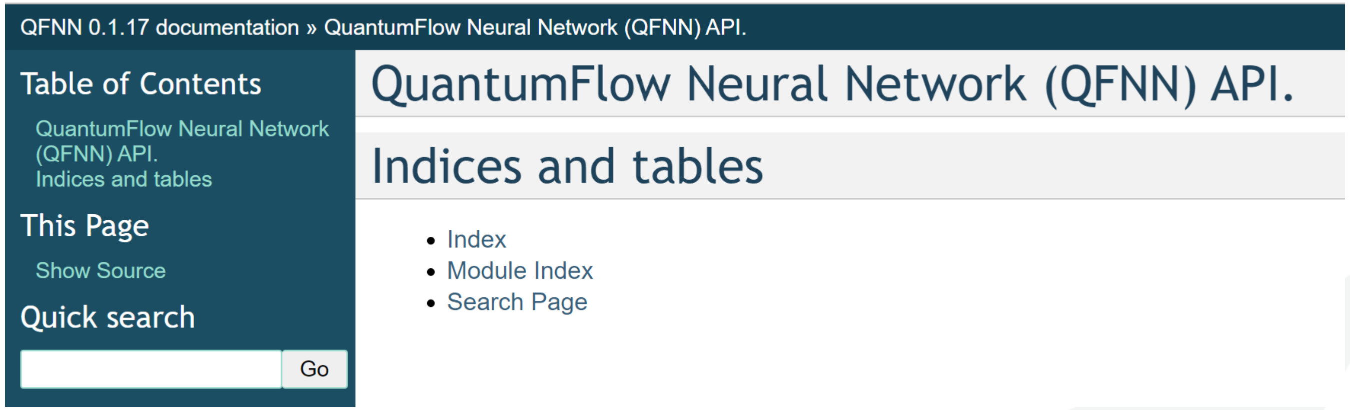
*Accepted by IEEE/ACM International Conference On Computer-Aided Design (**ICCAD**), Virtual, 2021.*

# On-Going Works in Building Quantum NN Co-Design Stack and Next

## Current works:

**Quatnum NN Co-Design Stack**




https://jqub.ece.gmu.edu/categories/QF/qfnn/index.html

https://github.com/jqub/qfnn

**QuantumFlow: An End-to-End Quantum Neural Network Acceleration Framework**
Zhirui Hu and **W. Jiang**
*IEEE International Conference on Quantum Computing and Engineering QCE 21 (**QuantumWeek**)*

# Conclusion & Resources

- **QF-Mixer** provides the fundamental design principles for the automatic design of quantum neural networks
- **QF-RobustNN** can learn the error in the quantum neural network
- **QFNN** provides interfaces for programming quantum neural networks

https://github.com/JQub/QuantumFlow_Tutorial (Source Code of All Hands-On in Tutorial)

https://github.com/JQub/qfnn (Source Code of QFNN API & Place to post Issues)

https://pypi.org/project/qfnn/ (Package of QFNN on PYPI)

https://libraries.io/pypi/qfnn/ (QFNN on Libraries.io)

https://www.nature.com/articles/s41467-020-20729-5

https://jqub.ece.gmu.edu (JQub Website)

https://jqub.ece.gmu.edu/categories/QF (News and **slides**)

https://jqub.ece.gmu.edu/categories/QF/qfnn/ (QFNN Documents)

https://arxiv.org/pdf/2012.10360.pdf

https://arxiv.org/pdf/2109.03806.pdf

https://arxiv.org/pdf/2109.03430.pdf

# Thank you!