Day 3 Recap

- Day 3:
  - Distance-based classifier (1703:10793)
  - Quantum optimization and adiabatic theorem
  - QAOA (1411.4028)
    - NP-hard problem
    - Variational Quantum Algorithms

Day 4 Plan

- Day 4:
  - ADAPT-QAOA (2103.17047)
  - Feedback-based ALgorithm Quantum Optimization (FALQON, 2103.08619)
  - Data re-uploading for a universal quantum classifier (1907.02085)

- Day 5: 1pm - 3:30pm
  - Quantum Fourier Transformation and Phase estimation
  - Error correction

  - Bernstein-Vazirani Algorithm and Simon's algorithm
  - Shor's algorithm, Grover's algorithm

Adaptive Derivative Assembled Problem Tailored - Quantum Approximate Optimization Algorithm (ADAPT-QAOA)

- https://arxiv.org/pdf/2005.10258.pdf

2005.10258

ADAPT-QAOA

$$\left|\psi_p(\vec{\gamma},\vec{\beta})\right\rangle = \left(\prod_{k=1}^{p}\left[e^{-iH_M\beta_k}e^{-iH_C\gamma_k}\right]\right)|\psi_{\text{ref}}\rangle \Rightarrow \left|\psi_p(\vec{\gamma},\vec{\beta})\right\rangle = \left(\prod_{k=1}^{p}\left[e^{-iA_k\beta_k}e^{-iH_C\gamma_k}\right]\right)|\psi_{\text{ref}}\rangle$$

$$\left|\psi^{(0)}\right\rangle = |\psi_{\text{ref}}\rangle = |+\rangle^{\otimes n}$$

n=number of qubits

mixer pool = set of A_j : $\{A_j\}$

Algorithm 1 ADAPT-QAOA

```
Initial state: $|\psi^{(0)}\rangle = |\psi_{\text{ref}}\rangle = |+\rangle^{\otimes n}$
Predefined: Number of layers $p$; Cost Hamiltonian $H_C$;
Initial parameter for optimization: $\gamma_0$; Operator pool with
$m$ operators $A_j$, $j \in [1, m]$
for $k = 1...p$ do
    //From operator pool select operator
    for $j = 1...m$ do
        //Get max measured gradient operator $A^{(k)}_{\text{max}}$:
        Set $\gamma_k = \gamma_0$
        Define $|\psi^{(k)}\rangle_t = e^{-iH_C\gamma_k}|\psi^{(k-1)}\rangle$
        $A^{(k)}_{\text{max}} = \text{argmax}\left(-i\ {}_t\langle\psi^{(k)}|[H_C, A_j]|\psi^{(k)}\rangle_t\right)$
    end for
    //Add $A^{(k)}_{\text{max}}$ to current ansatz:
    $|\psi^{(k)}\rangle = e^{-iA^{(k)}_{\text{max}}\beta_k}e^{-iH_C\gamma_k}|\psi^{(k-1)}\rangle$
    // Optimization
    $\min\langle\psi^{(k)}|H_C|\psi^{(k)}\rangle \to \vec{\beta}, \vec{\gamma}$
    output.add($\vec{\beta}, \vec{\gamma}, A^{(k)}_{max}, \min\langle\psi^{(k)}|H_C|\psi^{(k)}\rangle$)
end for
return output
```

2005.10258

ADAPT-QAOA

$$\left|\psi_p(\vec{\gamma},\vec{\beta})\right\rangle = \left(\prod_{k=1}^{p}\left[e^{-iH_M\beta_k}e^{-iH_C\gamma_k}\right]\right)|\psi_{\text{ref}}\rangle \Rightarrow \left|\psi_p(\vec{\gamma},\vec{\beta})\right\rangle = \left(\prod_{k=1}^{p}\left[e^{-iA_k\beta_k}e^{-iH_C\gamma_k}\right]\right)|\psi_{\text{ref}}\rangle$$

$$\left|\psi^{(0)}\right\rangle = |\psi_{\text{ref}}\rangle = |+\rangle^{\otimes n}$$

n=number of qubits

mixer pool = set of A_j : $\{A_j\}$

$$|\psi^{(k)}\rangle = e^{-i\beta A_j}e^{-iH_C\gamma_k}|\psi^{(k-1)}\rangle$$

$$|\psi^{(k)}\rangle_t = |\psi^{(k)}\rangle_t\Big|_{\beta=0}$$

$$\frac{\partial}{\partial\beta}{}_t\langle\psi^{(k)}|H_C|\psi^{(k)}\rangle_t = \frac{\partial}{\partial\beta}\langle\psi^{(k)}|H_C|\psi^{(k)}\rangle\Big|_{\beta=0} =$$

$$\frac{\partial}{\partial\beta}\langle\psi^{(k-1)}|e^{i\beta A_j}e^{iH_C\gamma_k}H_C e^{-i\beta A_j}e^{-iH_C\gamma_k}|\psi^{(k-1)}\rangle\Big|_{\beta=0}$$

Algorithm 1 ADAPT-QAOA

```
Initial state: $|\psi^{(0)}\rangle = |\psi_{\text{ref}}\rangle = |+\rangle^{\otimes n}$
Predefined: Number of layers $p$; Cost Hamiltonian $H_C$;
Initial parameter for optimization: $\gamma_0$; Operator pool with
$m$ operators $A_j$, $j \in [1,m]$
for $k = 1...p$ do
    //From operator pool select operator
    for $j = 1...m$ do
        //Get max measured gradient operator $A^{(k)}_{\max}$:
        Set $\gamma_k = \gamma_0$
        Define $|\psi^{(k)}\rangle_t = e^{-iH_C\gamma_k}|\psi^{(k-1)}\rangle$
        $A^{(k)}_{\max} = \text{argmax}\left(-i\ {}_t\langle\psi^{(k)}|[H_C, A_j]|\psi^{(k)}\rangle_t\right)$
    end for
    //Add $A^{(k)}_{\max}$ to current ansatz:
    $|\psi^{(k)}\rangle = e^{-iA^{(k)}_{\max}\beta_k}e^{-iH_C\gamma_k}|\psi^{(k-1)}\rangle$
    // Optimization
    $\min\langle\psi^{(k)}|H_C|\psi^{(k)}\rangle \to \vec{\beta},\vec{\gamma}$
    output.add($\vec{\beta},\vec{\gamma}, A^{(k)}_{max}, \min\langle\psi^{(k)}|H_C|\psi^{(k)}\rangle$)
end for
return output
```

2005.10258

ADAPT-QAOA

$$\left|\psi_p(\vec{\gamma},\vec{\beta})\right\rangle = \left(\prod_{k=1}^{p}\left[e^{-iA_k\beta_k}e^{-iH_C\gamma_k}\right]\right)|\psi_{\mathrm{ref}}\rangle$$

$$\left|\psi^{(0)}\right\rangle = |\psi_{\mathrm{ref}}\rangle = |+\rangle^{\otimes n}$$

mixer pool = set of A_j : $\{A_j\}$

$\mathcal{O}(1)$ elements $\quad P_{\mathrm{QAOA}} = \left\{\sum_{i\in Q} X_i\right\}$

$\mathcal{O}(n)$ elements $\quad P_{\mathrm{single}} = \cup_{i\in Q}\{X_i, Y_i\} \cup \left\{\sum_{i\in Q} Y_i\right\} \cup P_{\mathrm{QAOA}}$

$\mathcal{O}(n^2)$ elements $\quad P_{\mathrm{multi}} = \cup_{i,j\in Q\times Q}\{B_iC_j | B_i, C_j \in \{X,Y,Z\}\} \cup P_{\mathrm{single}}$

$$P_{\mathrm{QAOA}} \subset P_{\mathrm{single}} \subset P_{\mathrm{multi}} \longrightarrow \text{Best performance}$$

$$H_C = -\frac{1}{2}\sum_{i,j} w_{i,j}(I - Z_iZ_j)$$

- H_C has a Z_2 symmetry associated with the operator $F = \otimes_i X_i$. Since $[F, H_C] = 0$, one can show that the gradient is only nonzero for $[F, A_j] = 0$. The A_j that commutes with F are Pauli strings that have an even number of Y or Z operators.

ADAPT-QAOA

2005.10258

Zhu et al 2020

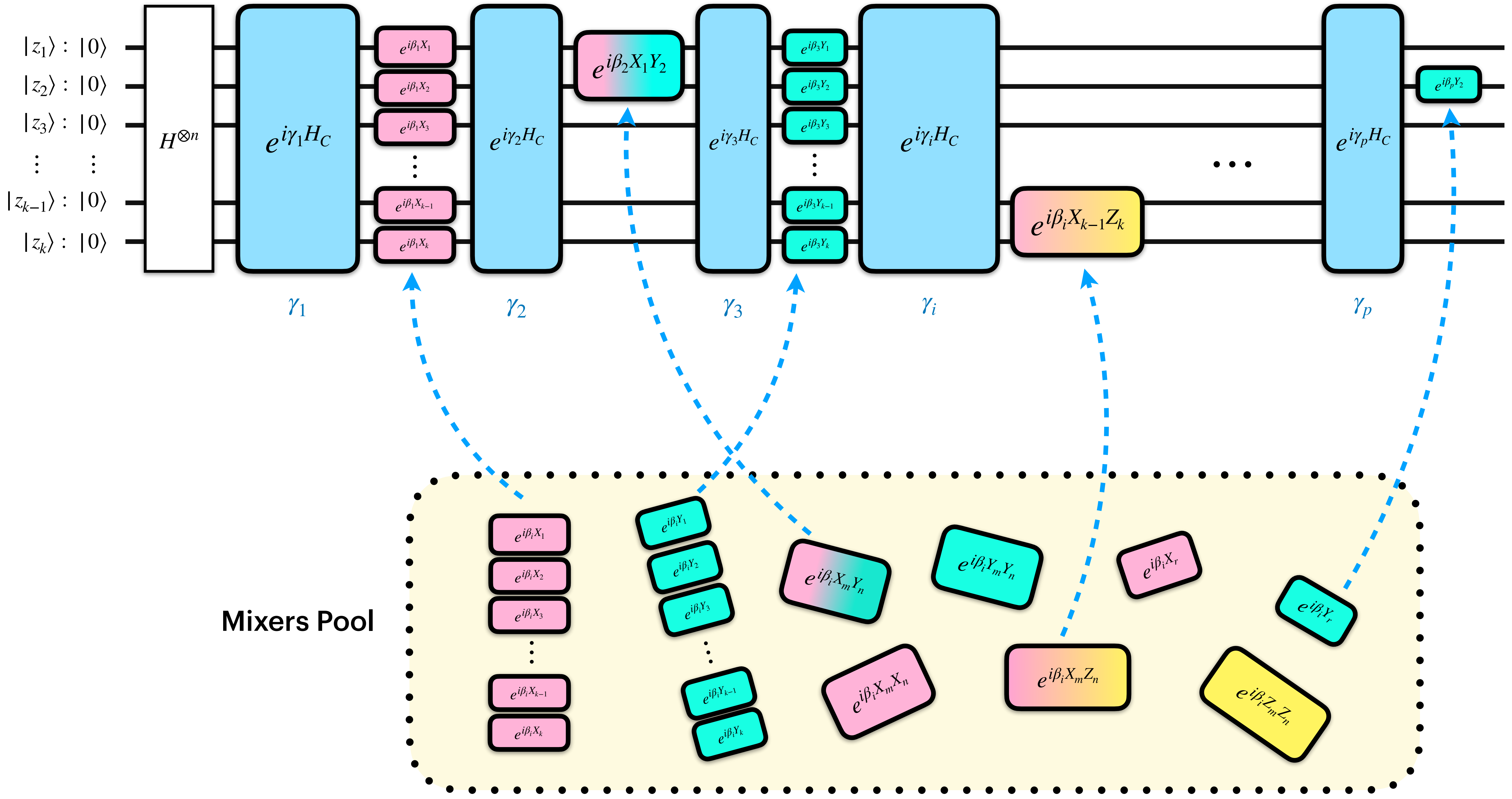


Figure taken from 2103.17047

2005.10258

ADAPT-QAOA applied to Mascot

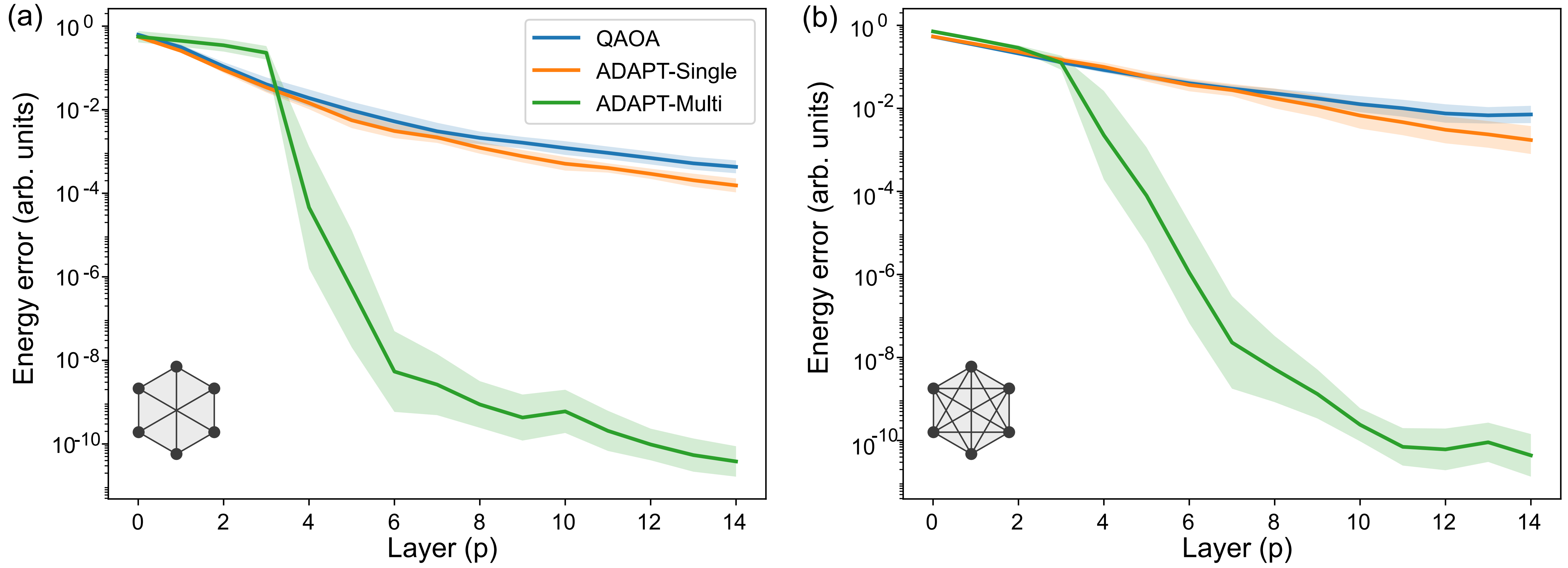


FIG. 1. Comparison of the performance of standard QAOA (blue) with ADAPT-QAOA for the single-qubit (orange) and multi-qubit (green) pools. The algorithms are run on the Max-Cut problem for the regular graphs shown in the figure, which have n=6 vertices and are of degree D=3 (a) and D=5 (b). The energy error (the difference between the energy estimate obtained by the algorithm and the exact ground state energy of H_C) is shown as a function of the number of layers in the ansatz. Results are shown for 20 different instances of edge weights, which are randomly sampled from the uniform distribution $U(0,1)$. The shaded regions indicate 95% confidence intervals.

Nelder-Mead for optimization
= downhill simplex method
= amoeba method
= polytope method

$\gamma_0 = 0.01$

- How much does the ADAPT-QAOA ansatz differ from the standard QAOA ansatz?
- When the single-qubit mixer pool is used, the single- qubit operators X_i are chosen instead of the standard mixer approximately 36.6% of the time for n=6,D=3 graphs and 25% of the time for n=6,D=5 graphs.
- For the multi-qubit mixer pool, the algorithm chooses operators other than the standard mixer approximately 75% of the time for n=6, D=3 graphs and 80% of the time for n=6, D=5 graphs.
- This trend supports the intuitive idea that a more connected graph requires more entanglement for a rapid convergence to the solution.

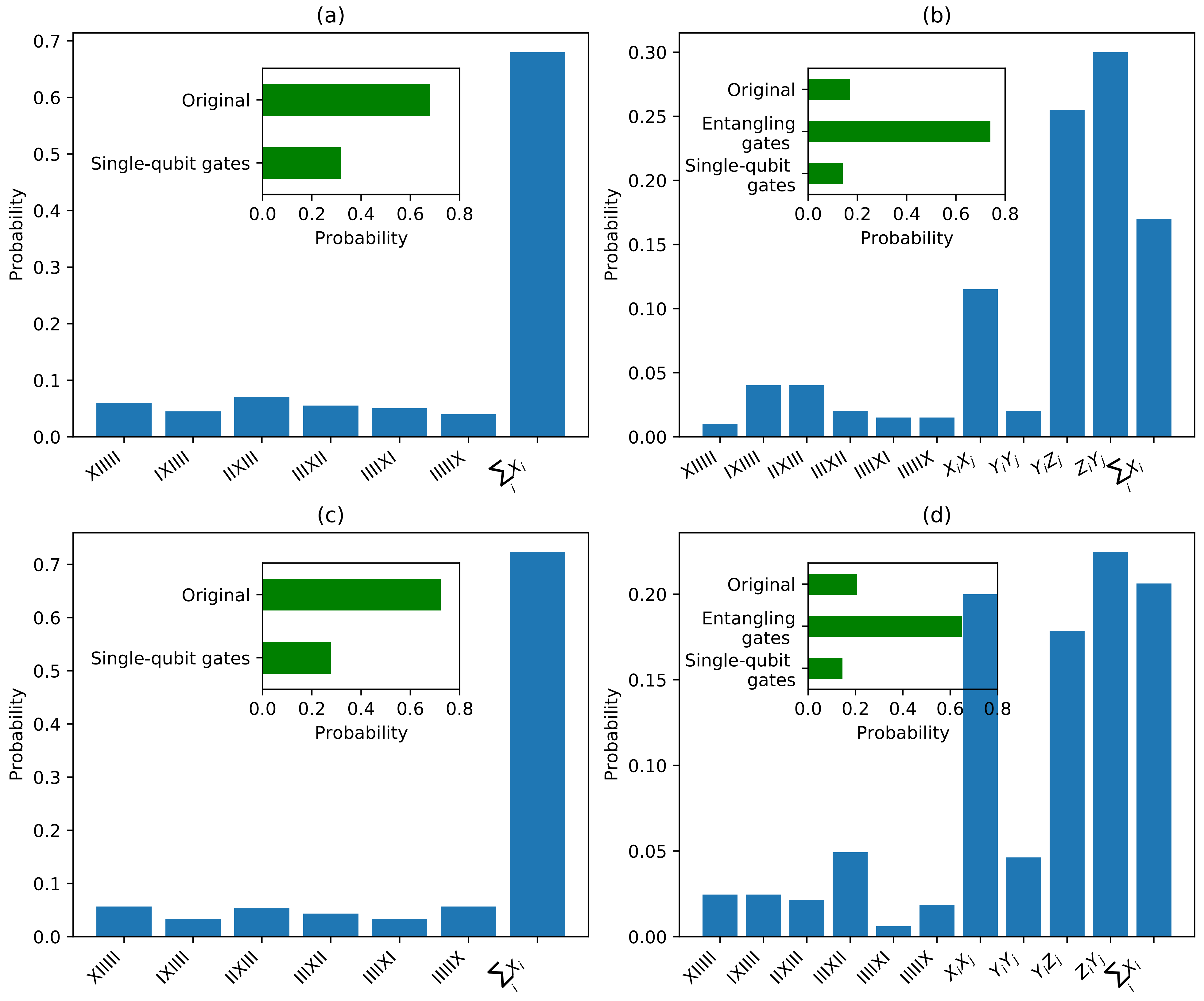
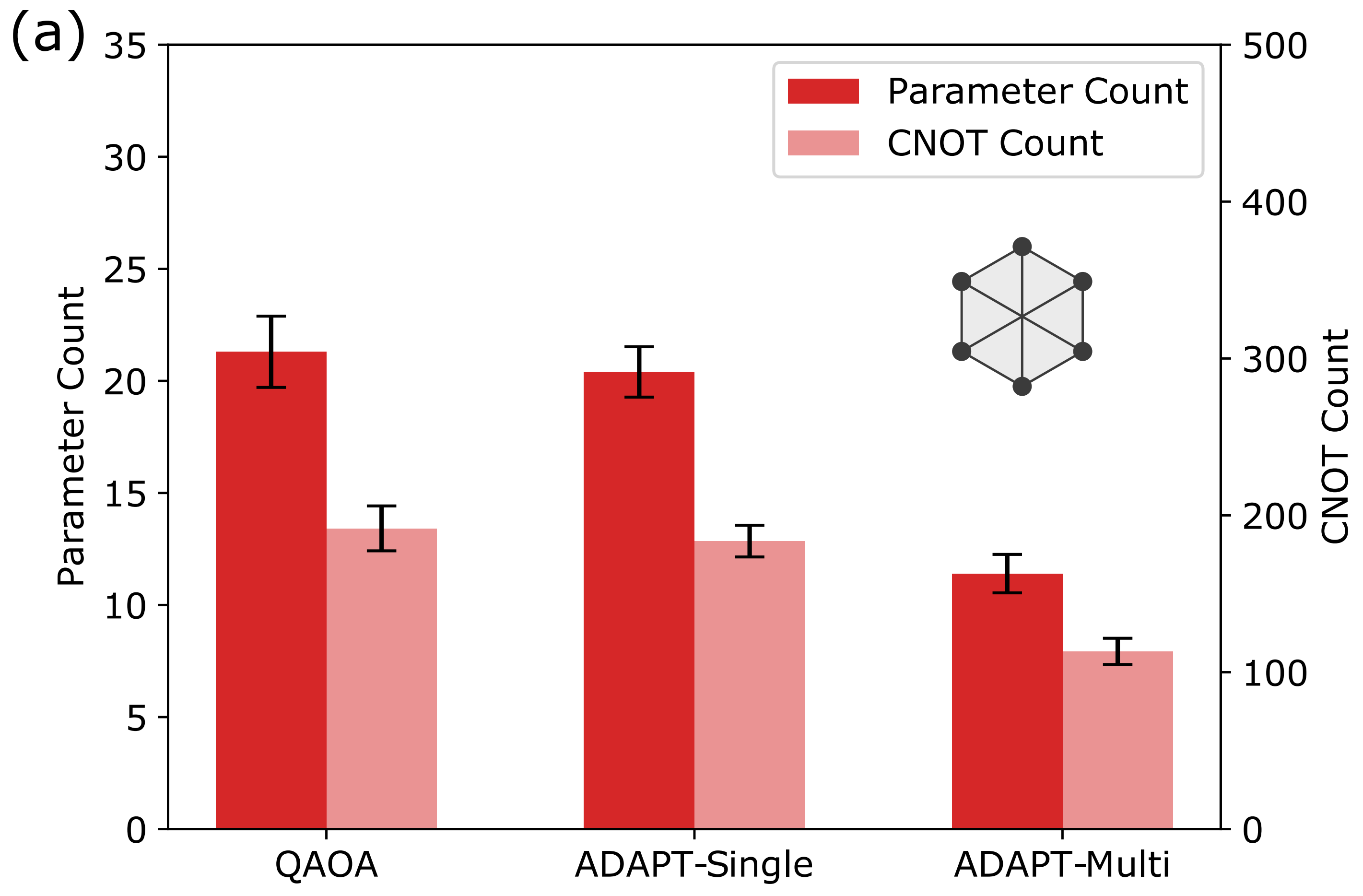
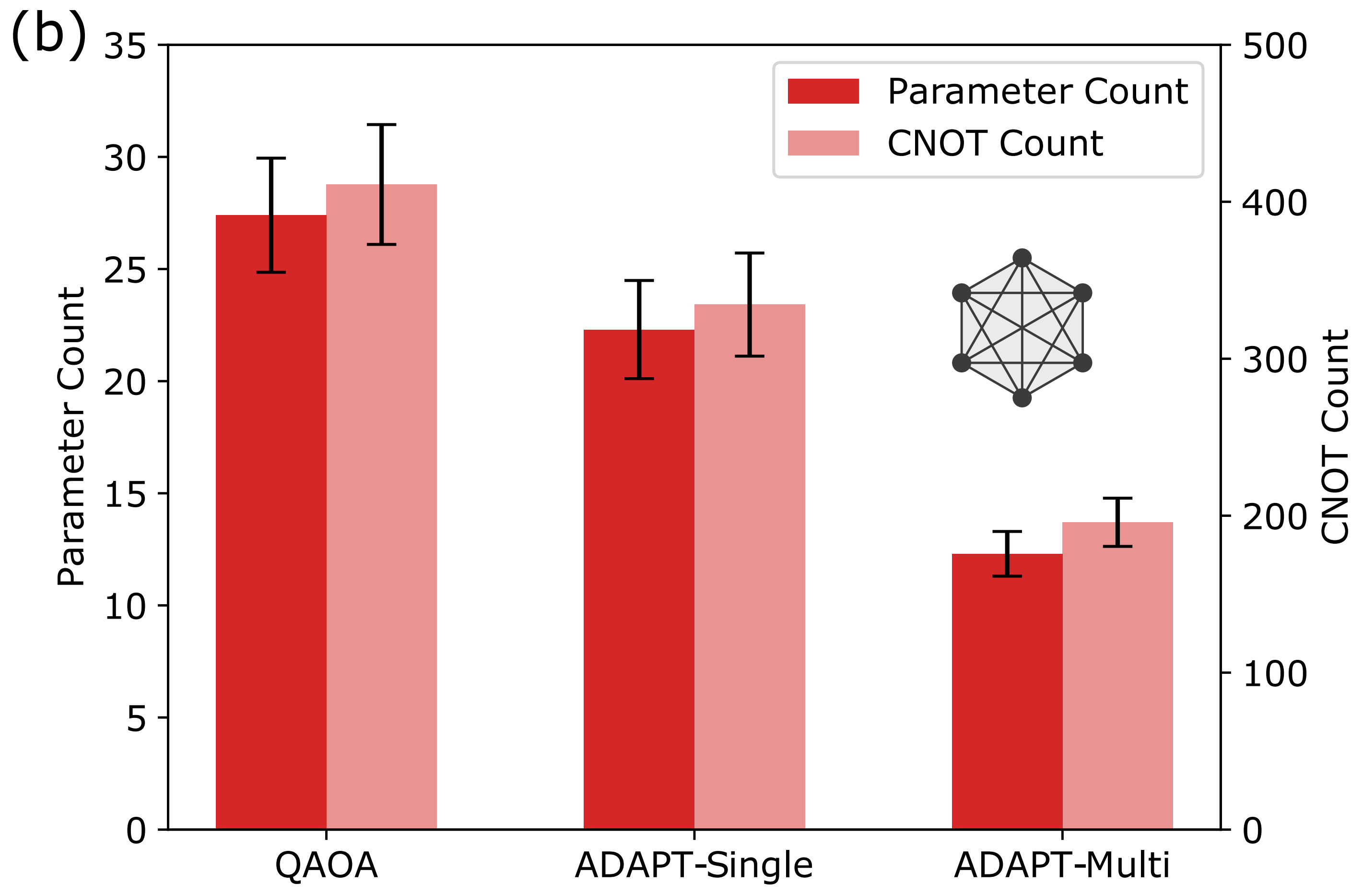


FIG. 6. Probability of operators picked by the original QAOA, ADAPT-QAOA with the single-qubit mixer and ADAPT-QAOA with multi-qubit pool for the Max-Cut problem on regular graphs with n=6 vertices with degree D=3 (a)(b) and D=5 (c)(d) with random edge weights sampled from a uniform distribution $U(0,1)$. The blue bars show the probability of each particular operator used for ansatz, and green bars show the probability of the original mixer, sum over all single-qubit gates and sum over all entangling gates used in ansatz. The results from 20 instances of random edge weights.

2005.10258

2005.10258





- ADAPT-QAOA provides a systematic way to both improve performance and reduce the number of parameters and CNOTs.

FIG. 2. Resource comparison of the standard QAOA, ADAPT-QAOA with the single-qubit mixer pool, and ADAPT-QAOA with the multi-qubit mixer pool for the Max-Cut problem on regular graphs with n=6 vertices and random edge weights. Panels (a) and (b) show the comparison for graphs of degree D=3 and D=5, respectively. For all cases except the standard QAOA applied to $D = 5$ graphs, we count the number of parameters and CNOTs needed to reach an energy error of $\delta E = 10^{-3}$. As standard QAOA for $D = 5$ graphs never reaches this error threshold, we instead count the CNOT gates and parameters at the end of the simulation (15 layers). The dark (light) red bars show variational parameter (CNOT gate) counts. The error bars show variances obtained by sampling over 20 different instances of edge weights.

Why ADAPT-QAOA performs better?

- Considering that the standard QAOA ansatz has a structure dictated by the adiabatic theorem, a possible explanation is related to Shortcuts to adiabaticity (STA).
- STA (counter-diabatic or transition less driving) was introduced by Demirplak and Rice [21] and later, independently, by Berry [22, 23].
- If we want to drive a system such that it remains in the instantaneous ground state at all times, then by adding a certain term H_{CD} to the Hamiltonian, we can achieve this without paying the price of a slow evolution.
- Although the instantaneous eigenstates of the original Hamiltonian only solve the time-dependent Schrodinger equation in the adiabatic limit, they become exact solutions when the Hamiltonian is updated to include H_{CD}.
- The advantage of STA is that the evolution can be achieved non-adiabatically.

Shortcuts to Adiabaticity (transitionless driving protocols)

$$i\,\partial_t\,|\psi\rangle = H(\theta(t))\,|\psi\rangle \quad \Longrightarrow \quad i\,\partial_t\,|\tilde{\psi}\rangle = (\tilde{H} - \dot{\theta}\tilde{A}_\theta)\,|\tilde{\psi}\rangle$$

$$\begin{aligned} |\psi\rangle &\longrightarrow |\tilde{\psi}\rangle = U^\dagger\,|\psi\rangle \\ H &\longrightarrow \tilde{H} = U^\dagger\,H\,U \\ i\partial_t &\longrightarrow i\partial_t - \dot{\theta}\tilde{A}_\theta \end{aligned} \qquad \begin{aligned} H_{CD} &= \dot{\theta}A_\theta \\ \tilde{A}_\theta &= iU^\dagger\partial_\theta\,U \\ A_\theta &= U\,\tilde{A}_\theta\,U^\dagger \end{aligned}$$

- Suppose that we consider a unitary transformation $U(\theta(t))$ to move the Hamiltonian $H(\theta(t))$ from the initial basis to its instantaneous eigenbasis, where $\tilde{H}(\theta) = U^\dagger(\theta)\,H(\theta)\,U(\theta)$ is diagonal at all times.
- The Schrodinger equation in the instantaneous eigenbasis is $i\,\partial_t\,|\tilde{\psi}\rangle = (\tilde{H} - \dot{\theta}\tilde{A}_\theta)\,|\tilde{\psi}\rangle$, where $\tilde{A}_\theta = iU^\dagger\,A_\theta\,U$ is the adiabatic gauge potential in the rotated frame. It is evident that the term $-\dot{\theta}\tilde{A}_\theta$ drives transitions between the energy levels of the original Hamiltonian H. Therefore, one can add the counterdiabatic term $H_{CD} = \dot{\theta}A_\theta$ to $H(\theta)$, with $A_\theta = U\,\tilde{A}_\theta\,U^\dagger$, to eliminate such transitions in the rotated frame. This is the core of transitionless driving protocols.
- Ref. [40] proposes an approximate gauge potential:

$$\mathcal{A}_\theta^{(p)} = i\sum_{k=1}^{p} a_k[\mathcal{H}, \partial_\theta\mathcal{H}]_{2k-1} \qquad [X,Y]_{k+1} = [X,[X,Y]]_k$$

Adaptive Derivative Assembled Problem Tailored QAOA (ADAPT-QAOA)

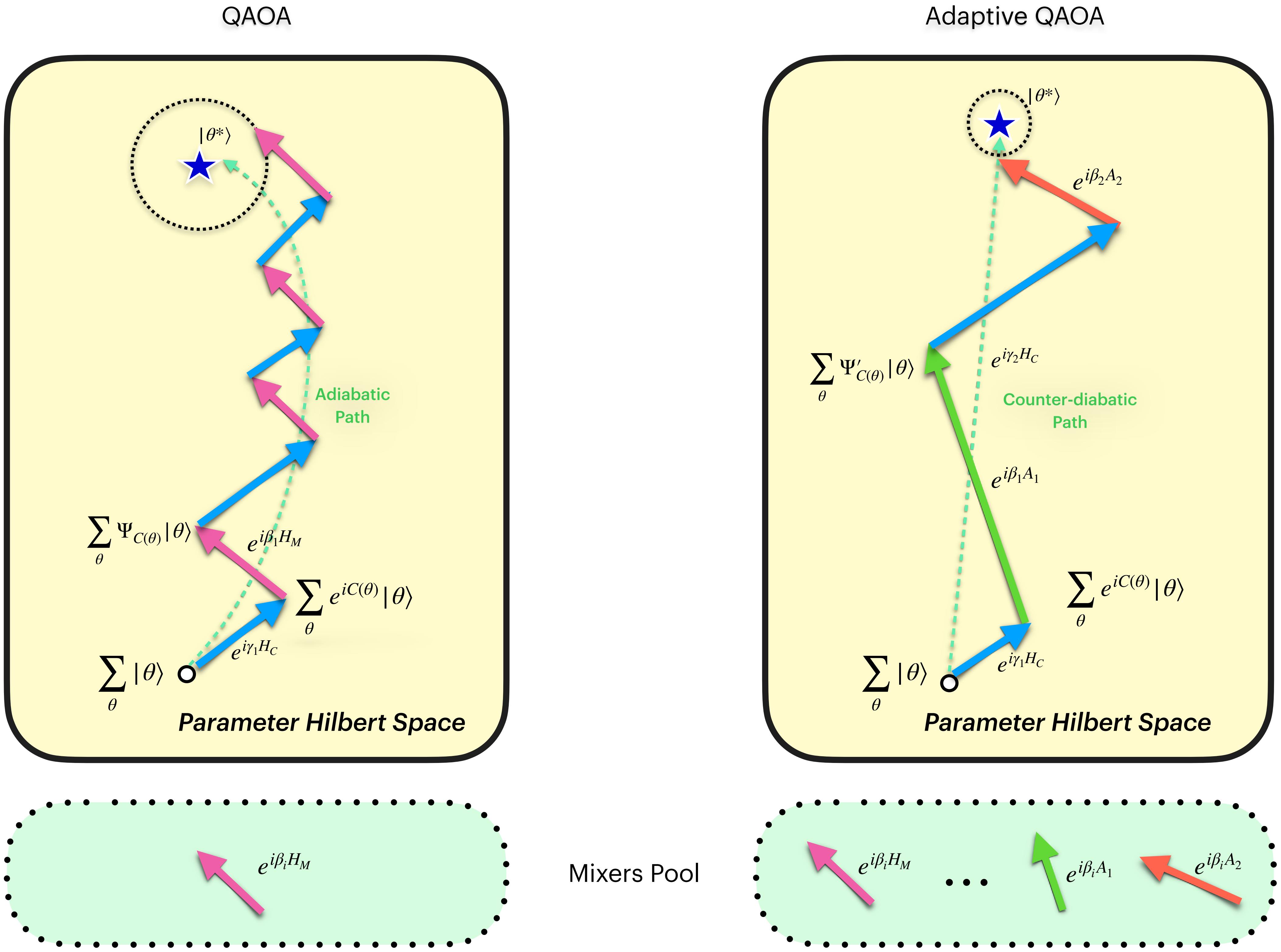


Figure taken from 2103.17047

Questions?

- Paper contains an interesting discussion on how to exploit non-adiabatic path.
- Paper contains evidence that ADAPT-QAOA is related to STA but not rigorous proof.
- Paper uses mixers with two Pauli matrices. What about 4, 6 or more?
- Non-Abelian shortcuts to adiabaticity on quantum simulation.
- There is an example code implemented in TensorFlowQuantum.

Quantum Neural Networks

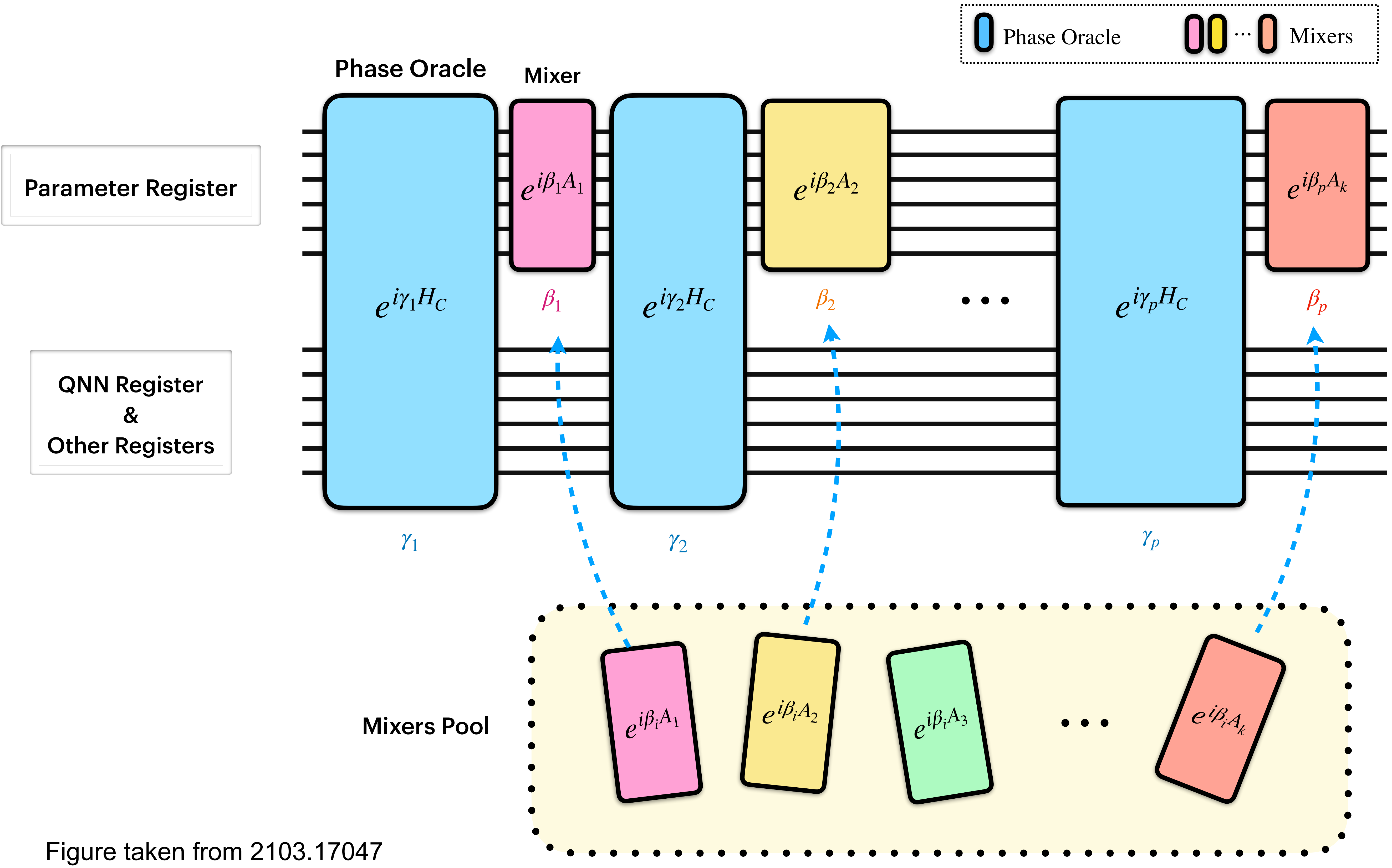
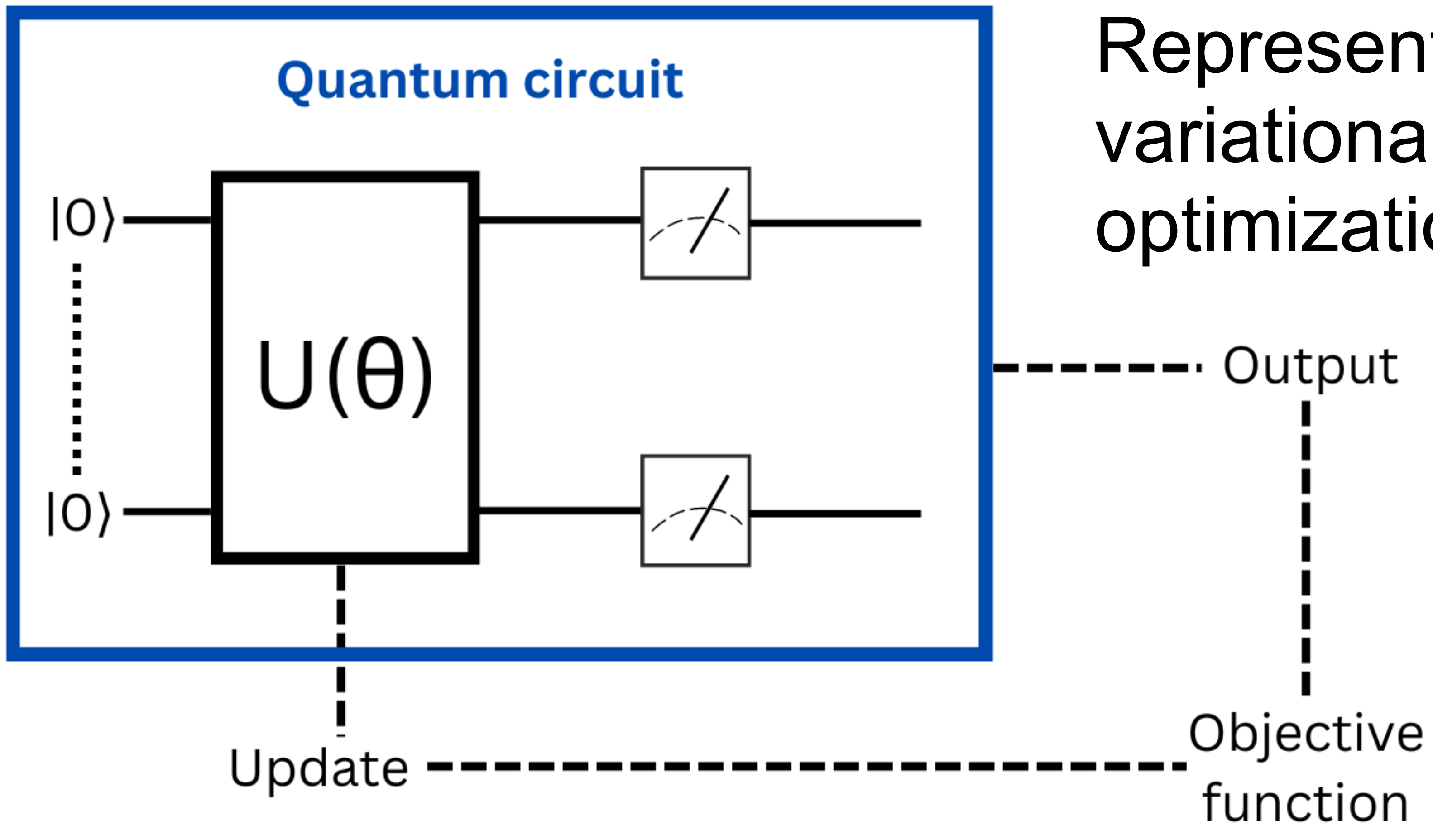


Figure taken from 2103.17047

- Training variational quantum algorithms is NP-hard
- https://arxiv.org/pdf/2101.07267.pdf



Representation of a variational quantum circuit optimization scheme.

Quantum Routines	QML Applications
HHL algorithm	QSVM [4,74] Q linear regression [8] Q least squares [75] QPCA [14]
Grover's algorithm	Q k-Means [10] Q K-Median [13] QKNN [6] Q Perceptron Models [76] Q Neural Networks [3]
Quantum phase estimation	Q k-Means [10]
Variational quantum circuit	Q decision tree [9] Circuit-centric quantum classifiers [77] Deep reinforcement learning [78]

An overview of current quantum machine learning algorithm.

Feedback-based ALgorithm Quantum Optimization (FALQON)

- 2103.08619, https://pennylane.ai/qml/demos/tutorial_falqon.html
- Consider a quantum system whose dynamics is governed by

$$i\frac{d}{dt}|\psi(t)\rangle = (H_\mathrm{p} + H_\mathrm{d}\beta(t))|\psi(t)\rangle$$

- Goal is to minimize: $\langle H_\mathrm{p}\rangle = \langle\psi(t)|H_\mathrm{p}|\psi(t)\rangle$

H_p : drift Hamiltonian

H_d : control Hamiltonian

$\beta(t)$: time – dependent control function

- One can minimize $\langle H_p\rangle$ by designing $\beta(t)$ such that

$$\frac{d}{dt}\langle\psi(t)|H_\mathrm{p}|\psi(t)\rangle(t) \leq 0, \quad \forall t \geq 0$$

$$\frac{d}{dt}\langle\psi(t)|H_p|\psi(t)\rangle = i\langle\psi(t)|(H_p + H_d\beta(t))H_p|\psi(t)\rangle - i\langle\psi(t)|H_p(H_p + H_d\beta(t))|\psi(t)\rangle$$

$$= \langle\psi(t)|\, i\,[H_d, H_p]\,|\psi(t)\rangle\,\beta(t) \equiv A(t)\,\beta(t)$$

Feedback-based ALgorithm Quantum Optimization (FALQON)

$$\frac{d}{dt}\langle\psi(t)\,|\,H_p\,|\,\psi(t)\rangle = \langle\psi(t)\,|\,i\,[H_d, H_p]\,|\,\psi(t)\rangle\,\beta(t) \equiv A(t)\,\beta(t)$$

- We can choose any $\beta(t)$.
- Consider $\beta(t) = -w\,f(t, A(t))$ for $w > 0$, where $f(t, A(t))$ is any continuous function with $f(t,0) = 0$ and $A(t)f(t, A(t)) > 0$ for all $A(t) \neq 0$.
- Take $w = 1$ and $f(t, A(t)) = A(t)$ such that $\beta(t) = -A(t)$ for simplicity.
- Consider alternating (rather than concurrent) applications of H_p and H_d, leading to a time evolution:

$$U = U_d(\beta_\ell)\,U_p \cdots U_d(\beta_1)\,U_p$$

$$U_p = e^{-iH_p\Delta t} \qquad k = 1, 2, \cdots, \ell \qquad \beta_k = \beta(k\tau - \Delta t)$$

$$U_d(\beta_k) = e^{-i\beta_k H_d \Delta t} \qquad \tau = 2\Delta t \qquad = \beta((k-1)\Delta t)$$

- For small Δt, this unitary evolution yields Trotterized approximation to the continuous time evolution of the system.

Feedback-based ALgorithm Quantum Optimization (FALQON)

- During the time evolution when H_p is applied, $\frac{d}{dt}\langle H_p \rangle = 0$, but eigenstate of H_p accumulates phase changes. (H_p is time-independent.)
- For the time evolution when H_d is applied, we recover $\frac{d}{dt}\langle H_p \rangle = A(t)\beta(t)$
- Set $\beta_{k+1} = -A_k$, where $A_k = \langle \psi_k | i\,[H_d, H_p]\,|\psi_k \rangle$
- In this setting, it is always possible to choose Δt small enough such that $\frac{d}{dt}\langle \psi(t) | H_p | \psi(t) \rangle \leq 0$. If Δt is chosen to be too large, the inequality will be violated.
- FALQON is a constructive, optimization free procedure for assigning values to each β_k according to a feedback law.
- By design, the quality of the solution to the combinatorial optimization problem improves monotonically with respect to depth of the circuit, k.

Feedback-based ALgorithm Quantum Optimization (FALQON)

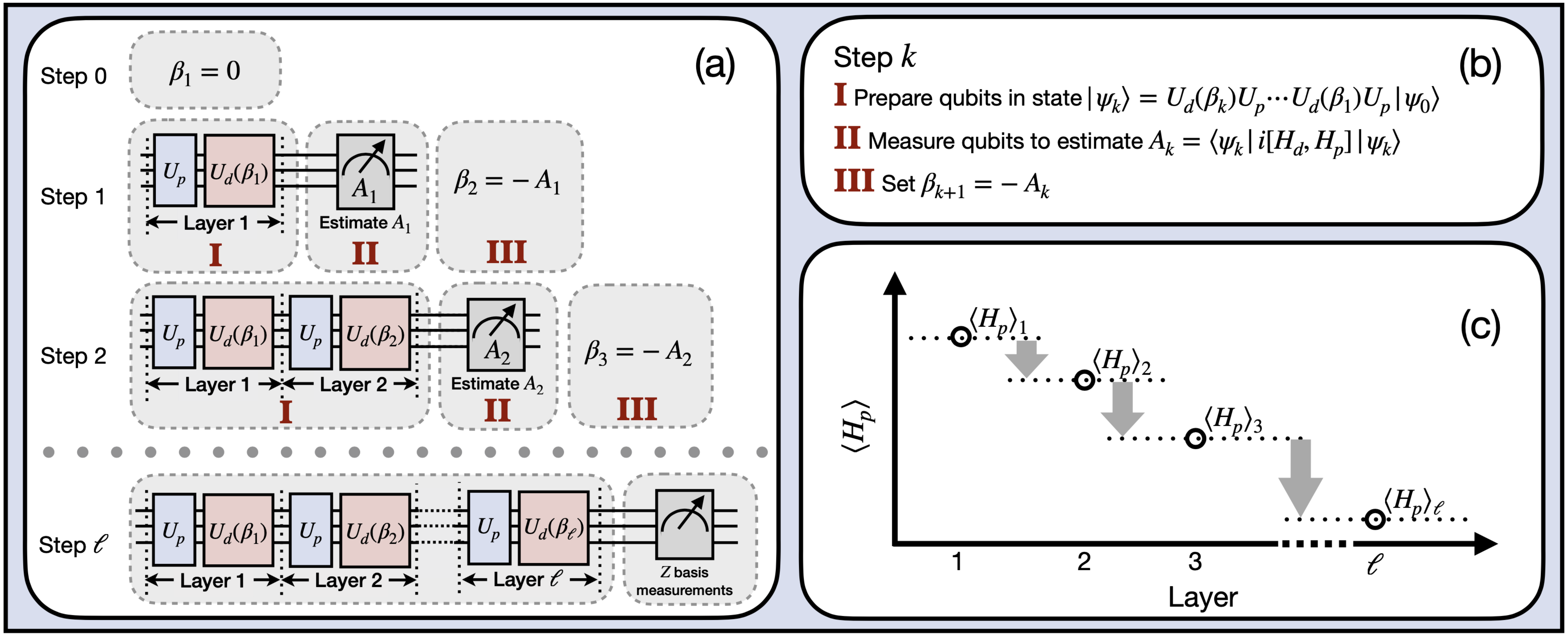


Figure 1. (a) The procedure for implementing FALQON. The initial step is to seed the procedure by setting $\beta_1 = 0$. The qubits are then initialized in the state $|\psi_0\rangle$, and a single FALQON layer is implemented to prepare $|\psi_1\rangle = U_\mathrm{d}(\beta_1)U_\mathrm{p}|\psi_0\rangle$. The qubits are then measured to estimate A_1, whose result is fed back to set $\beta_2 = -A_1$, up to sampling error. For subsequent steps $k = 2, \cdots, \ell$, the same procedure is repeated, as shown in (b): the qubits are initialized as $|\psi_0\rangle$, after which k layers are applied to obtain $|\psi_k\rangle = U_\mathrm{d}(\beta_k)U_\mathrm{p} \cdots U_\mathrm{d}(\beta_1)U_\mathrm{p}|\psi_0\rangle$, and then the qubits are measured to estimate A_k, and the result is fed back to set the value of β_{k+1}. This procedure causes $\langle H_\mathrm{p}\rangle$ to decrease layer-by-layer as per $\langle\psi_1|H_\mathrm{p}|\psi_1\rangle \geq \langle\psi_2|H_\mathrm{p}|\psi_2\rangle \geq \cdots \geq \langle\psi_\ell|H_\mathrm{p}|\psi_\ell\rangle$, as shown in (c), such that the quality of the solution to the combinatorial optimization problem monotonically improves with circuit depth. The protocol can be terminated when the value of $\langle H_\mathrm{p}\rangle$ converges or a threshold number of layers ℓ is reached. Then, after the final step, Z basis measurements on $|\psi_\ell\rangle$ can be used to determine a best candidate solution to the combinatorial optimization problem of interest, by repeatedly sampling from the probability distribution over bit strings induced by $|\psi_\ell\rangle$ and selecting the outcome associated with the best solution.

FALQON vs QAOA

- Circuits used in QAOA has the same alternative structure as those in FALQON with additional parameters $\vec{\gamma} = (\gamma_1, \cdots, \gamma_\ell)$ that enter into U_p such that $U_{QAOA} = U_d(\beta_\ell)U_p(\gamma_\ell)\cdots U_d(\beta_1)U_p(\gamma_1)$.
- Solution to the original combinatorial optimization is found by minimizing $\langle\psi(\vec{\gamma},\vec{\beta})|H_p|\psi(\vec{\gamma},\vec{\beta})\rangle$ over 2ℓ parameters, using classical optimization. ($|\psi(\vec{\gamma},\vec{\beta})\rangle = U_{QAOA}|\psi_0\rangle$)
- FALQON minimizes $\langle H_p\rangle$ over a sequence of quantum circuit layers, guided by qubit measurement-based feed back without classical optimization.

FALQON for MaxCut problem

- MaxCut: $H_p = -\sum_{(i,j)\in E} \frac{1}{2}\left(1 - Z_iZ_j\right)$ and $H_d = \sum_{j=1}^{n} X_j$
- $i\,[H_d, H_p] = \sum_{(i,j)\in E} Y_iZ_j + Z_iY_j$ where X_j, Y_j and Z_j are Pauli's matrices.

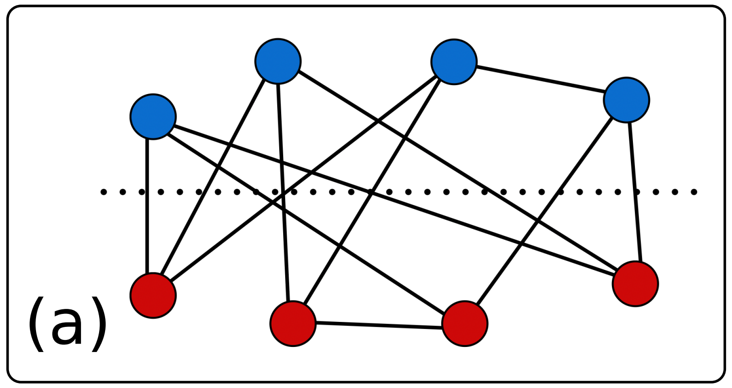
FALQON for MaxCut problem

Approximation ration:

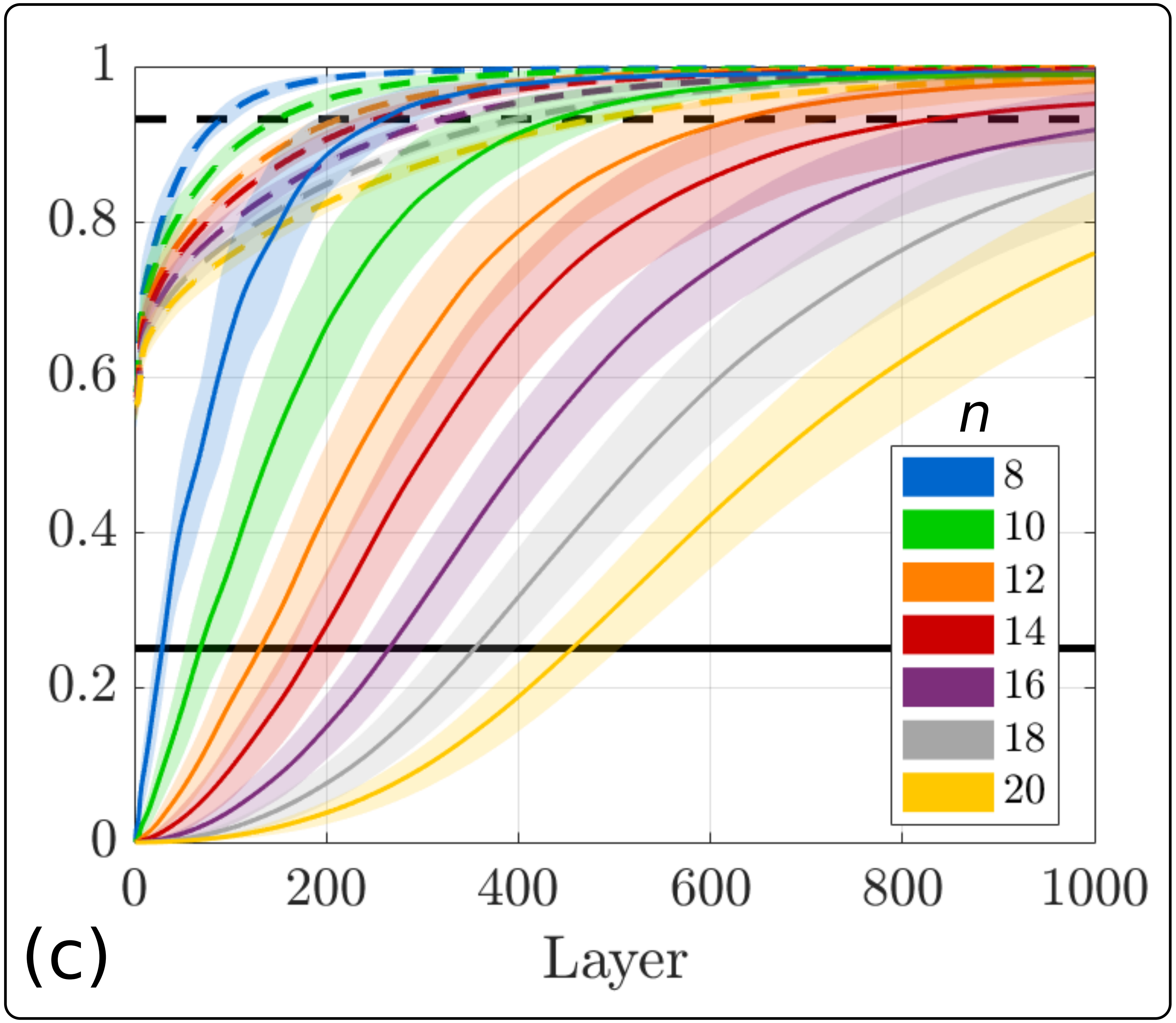
$$r_\mathrm{A} = \langle H_\mathrm{p} \rangle / \langle H_\mathrm{p} \rangle_\mathrm{min}$$

The largest known approximation ratio $r_A = 0.932$ by algorithm of Goemans and Williamson.

approximation ratio (dashed curves) and the success probability of measuring the degenerate ground state (solid curves)

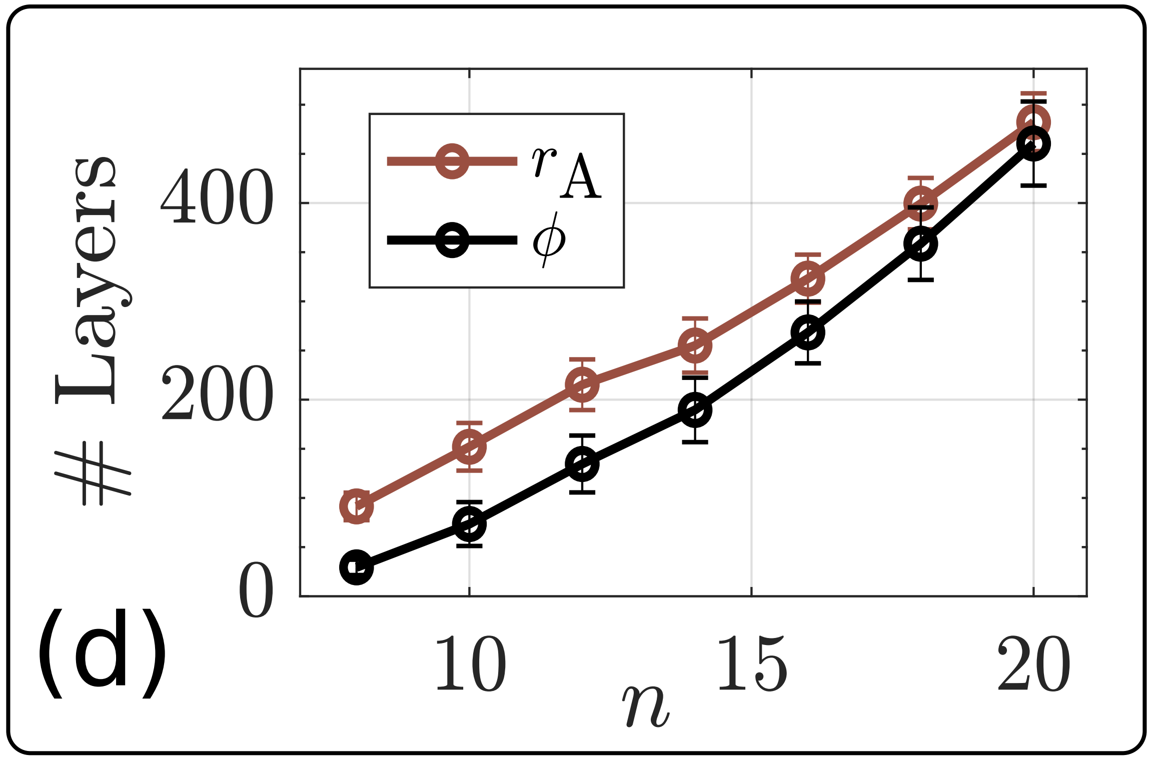


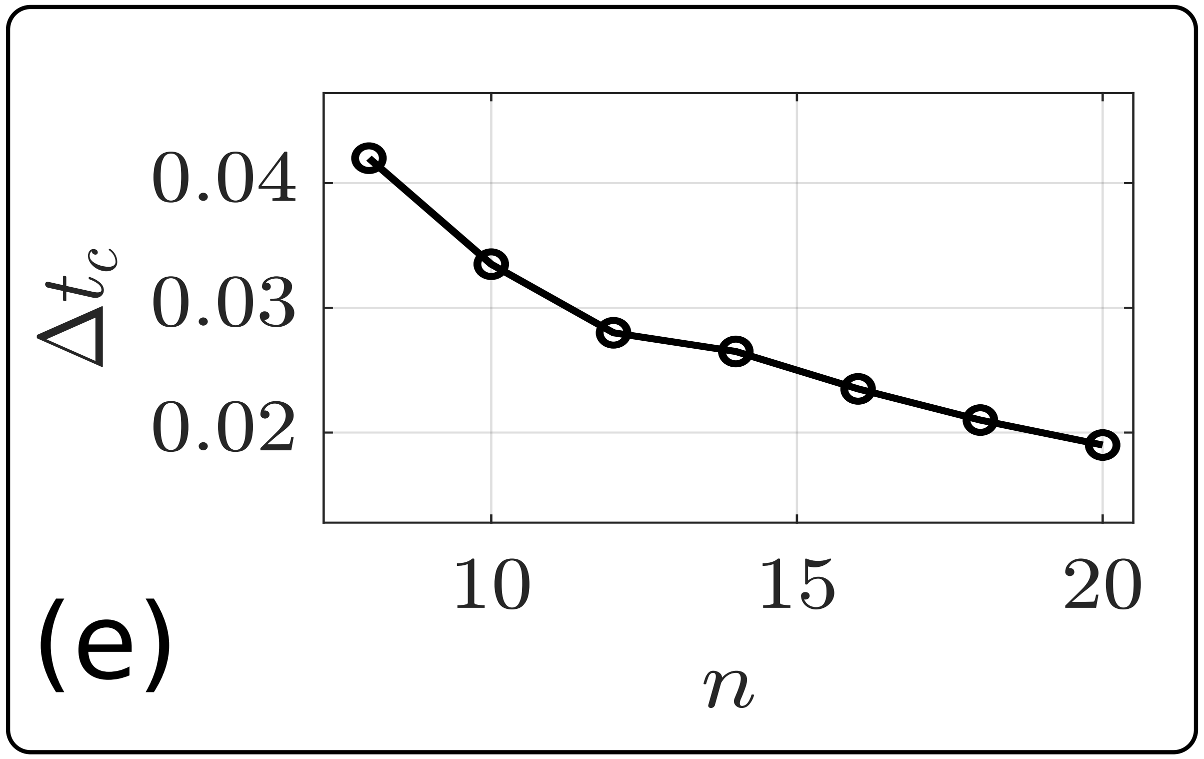
Pictorial representation of MaxCut on a 3-regular graph with 8 vertices.



n = the number of vertices

FALQON for MaxCut problem





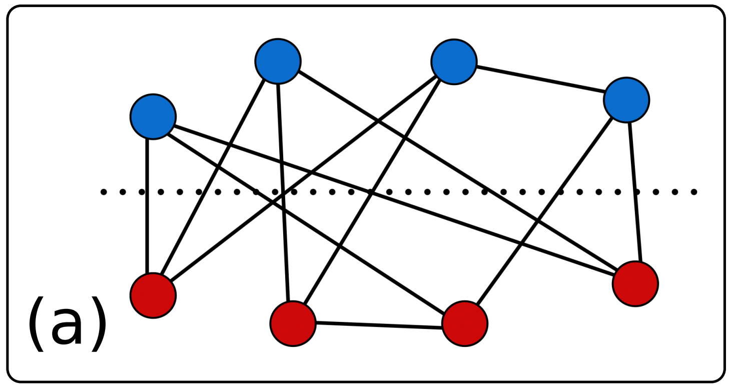
The mean number of layers needed to achieve the reference values of $r_A = 0.932$(dashed curve) and $\phi = 0.25$ (solid curve) is shown; error bars report the associated standard deviation.

The critical Δt values for different problem sizes are plotted.

The only free parameter is time step Δt, which is tuned to be as large as possible.

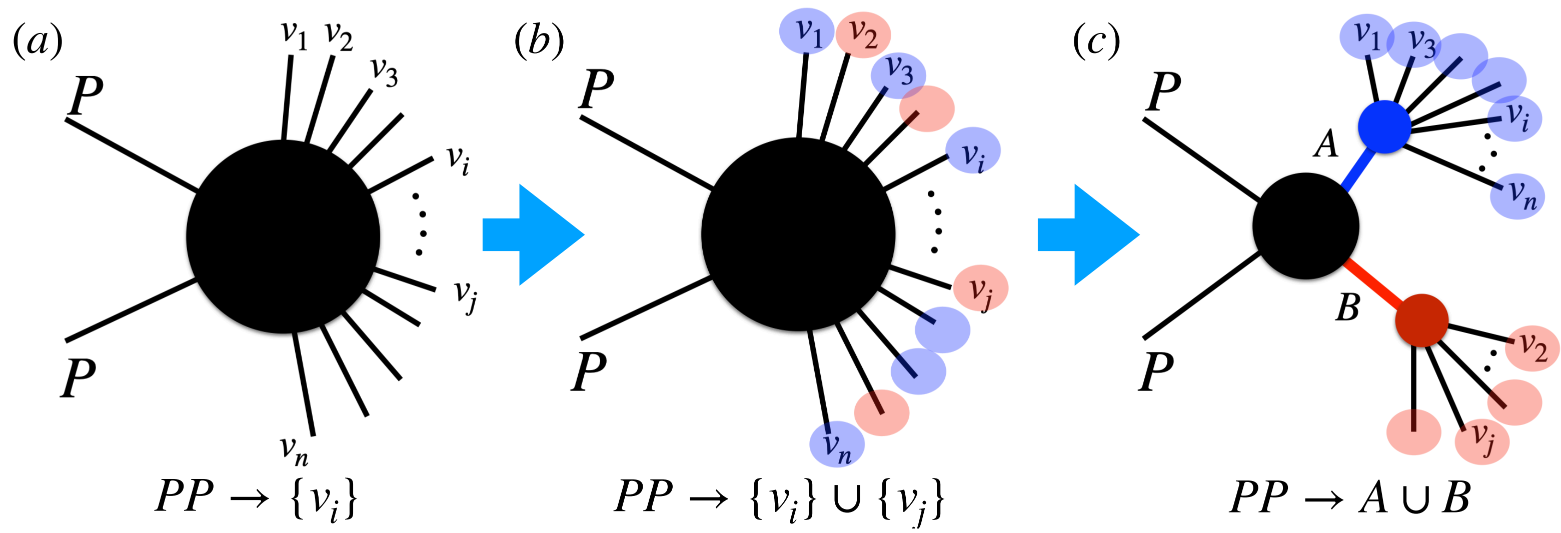
$r_{\mathrm{A}} = \langle H_{\mathrm{p}} \rangle / \langle H_{\mathrm{p}} \rangle_{\min}$ = approximation ratio

$\phi = \sum_i |\langle \psi | q_{0,i} \rangle|^2$ = the success probability of measuring the (potentially degenerate) ground state(s) $\{|q_{0,i}\rangle\}$,



Pictorial representation of MaxCut on a 3-regular graph with 8 vertices.

Combinatorial problems at the LHC



(a)
P
P
v_1 v_2 v_3 v_i v_j v_n
$PP \to \{v_i\}$
(b)
$PP \to \{v_i\} \cup \{v_j\}$
(c)
A
B
$PP \to A \cup B$

FIG. 1. (a) n-observed particles (b) Dividing n particles into two groups for $2 \to 2$ process (c) Identified event-topology with A and B.

- Assuming $2 \to 2$ production with subsequent decays, identification of an event-topology becomes a binary classification, with 2^n possibilities.
- Combinatorial problem: What would be an efficient way of assigning all observed particles in two decay chains?

p_i is the momentum of constituent of A if $x_i = 1$ $$P_1 = \sum_i p_i\, x_i$$

p_i is the momentum of constituent of B if $x_i = 0$ $$P_2 = \sum_i p_i\, (1 - x_i)$$

Minimize the mass difference: $H = \left(P_1^2 - P_2^2\right)^2$ for all possible combinations of x_i

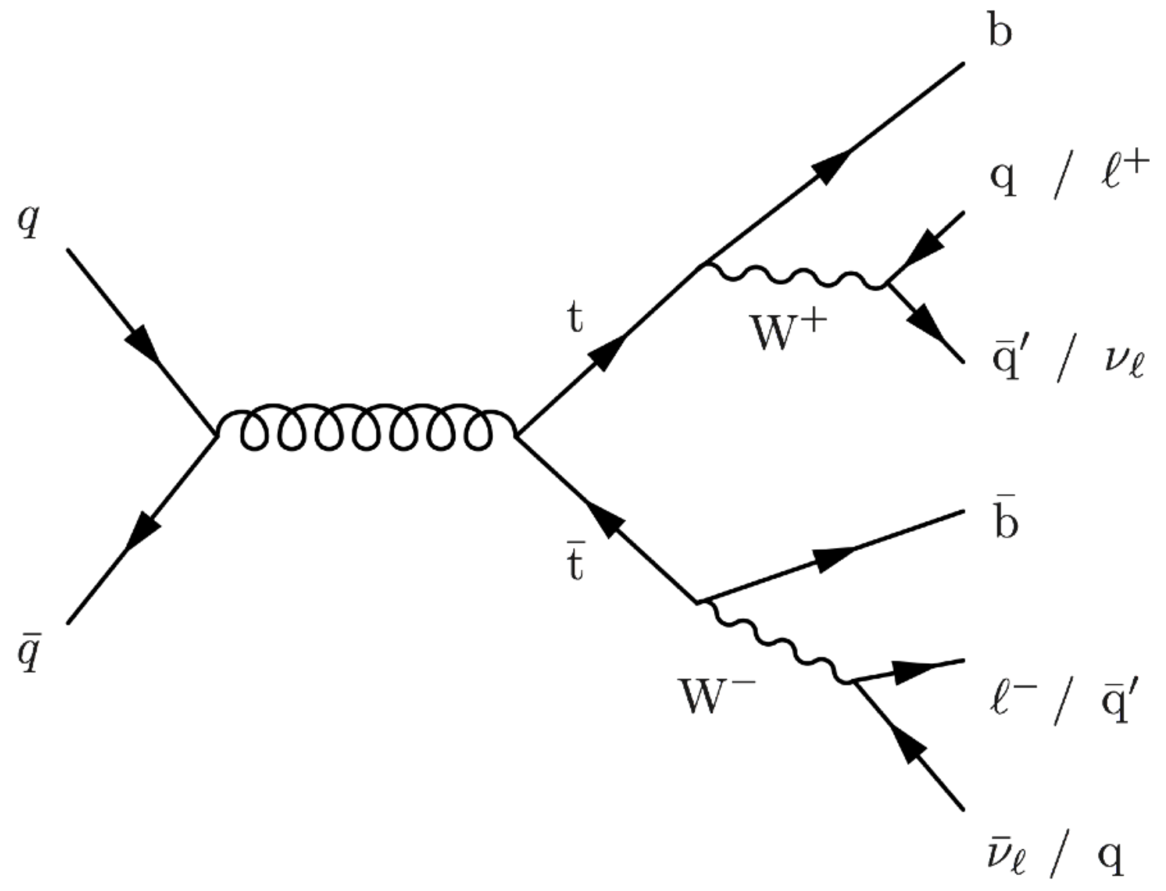
$$\boxed{H_{\rm QUBO} = \sum_{ij} J_{ij} s_i s_j + \sum_i h_i s_i}$$

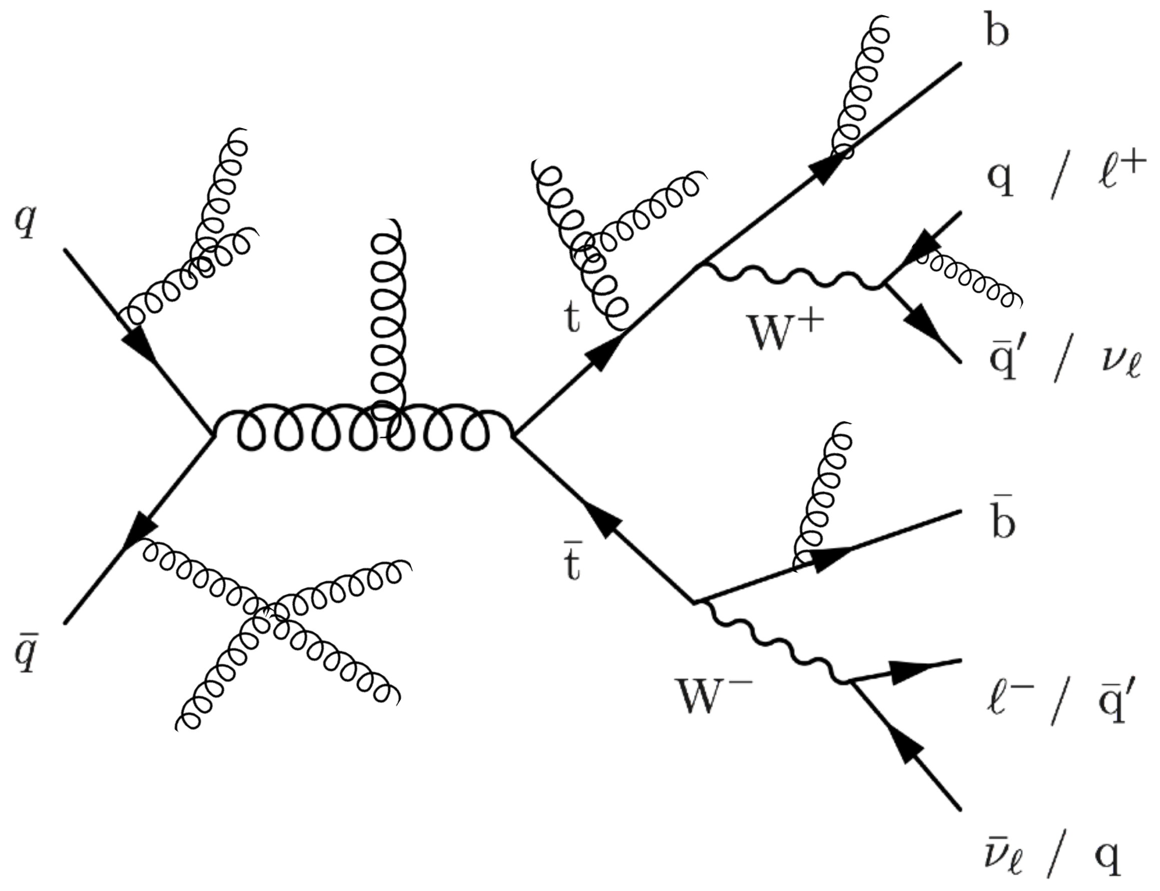
$$x_i = \frac{1 + s_i}{2}$$

$$J_{ij} = \sum_{k\ell} P_{ik} P_{j\ell},$$

$$h_i = 2 \sum_j [\sum_{k\ell} (P_{ik} P_{j\ell} - P_{k\ell} P_{ij})],$$

Combinatorial problems in the top quark production





- b-ℓ ambiguity (Semi-leptonic and dilepton)

Ht, mbl	CDF
pt, mbl	1009.2751, Rajaraman, Yu
MT2, mbl	1109.1563, Baringer, Kong, McCaskey, Noonan
MT2, mbl, MAOS, hybrid	1109.2201, Kim, Guadagnoli, Park
M2, mbl, hybrid	1706.04995, Debnath, Kim, Kim, Kong, Matchev
NN	2202.05849, Alhazmi, Dong, Huang, Kim, Kong, Shih

- Fully hadronic channel: $2^6 = 64$ possibilities for 6 particles in the final state. But in reality, 10-20 jets appear in the final state, leading to 2^{10}-2^{20} possibilities.

Combinatorial problems at the LHC

$$
\begin{aligned}
H_{\text{QUBO}} &\to H_{\text{QUBO}} + \lambda(P_1^2 + P_2^2)\\
&= H_{\text{QUBO}} + \lambda \sum_{ij} P_{ij}[s_i s_j + (1-s_i)(1-s_j)]\\
&= \sum_{ij} J'_{ij} s_i s_j + \sum_i h'_i s_i,
\end{aligned}
$$

for $m_A \neq m_B$

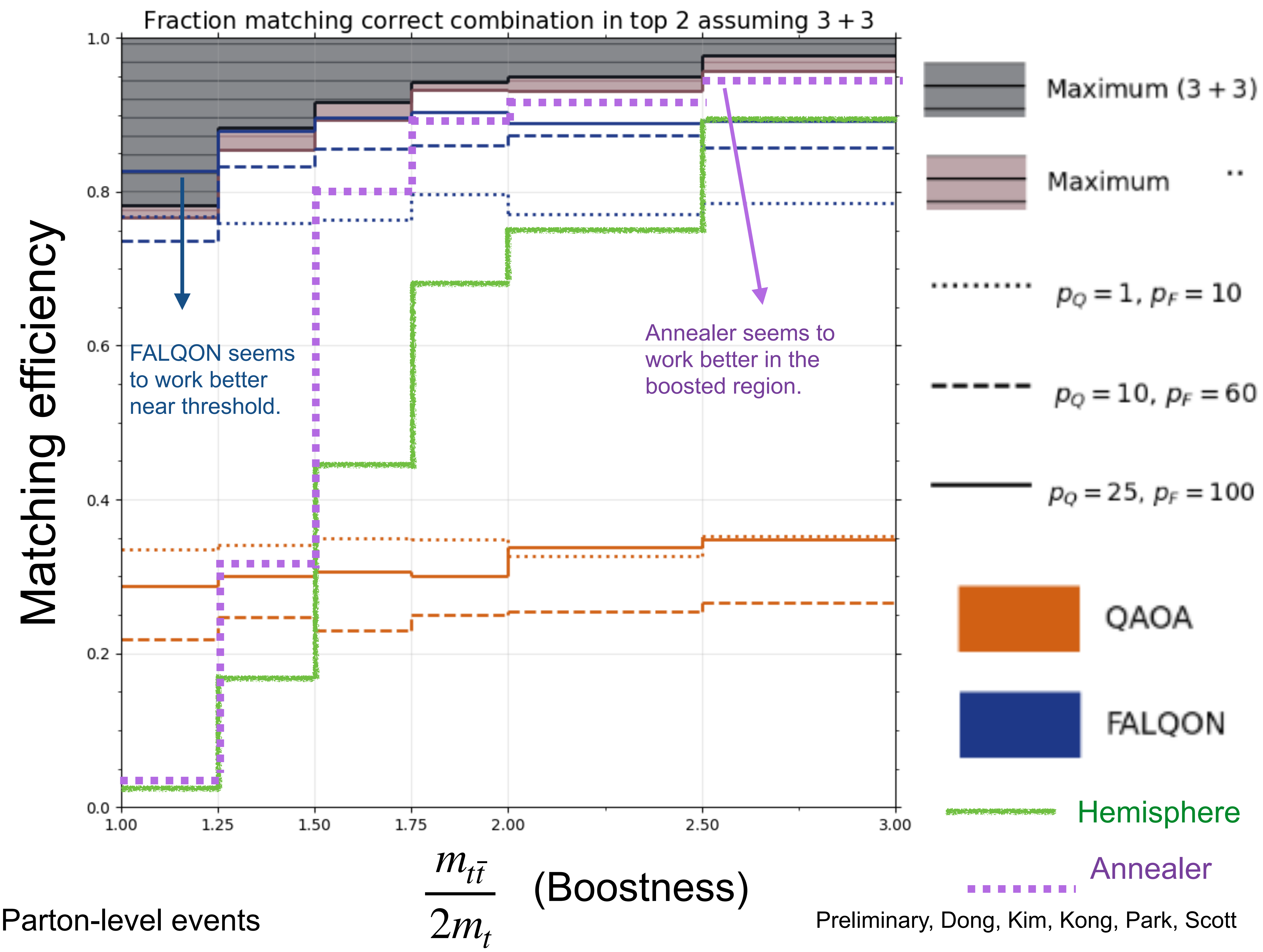
2111.07806, Kim, Ko, Park, Park

$$
\begin{aligned}
&pp \to t, \bar{t} \to \{j_1, j_2, j_3, j_4, j_5, j_6\},\\
&pp \to H, Z \to \{j_1, j_2, j_3, j_4, j_5, j_6\},\\
&pp \to \tilde{o}, \tilde{o}^* \to t, \bar{t}, t, \bar{t} \to \{j_1, j_2, j_3, \cdots, j_{11}, j_{12}\}.
\end{aligned}
$$

Process		$pp \to t\bar{t}$ Eq. (7a)	$pp \to HZ$ Eq. (7b)	$pp \to \tilde{o}\tilde{o}^*$ Eq. (7c)
Algorithm	QUBO	47.3%	89.5%	17.4%
	Hemishere	33.6%	86.2%	7.2%

Compared to the processing time of O(2^n) with the simplest but a robust brute-force scanning algorithm with a classical computer, a quantum annealer can have an enormous advantage in the computation complexity as

$$T_{\text{QUBO}}(n) \sim \mathcal{O}(n^2) \ll \mathcal{O}(2^n)$$

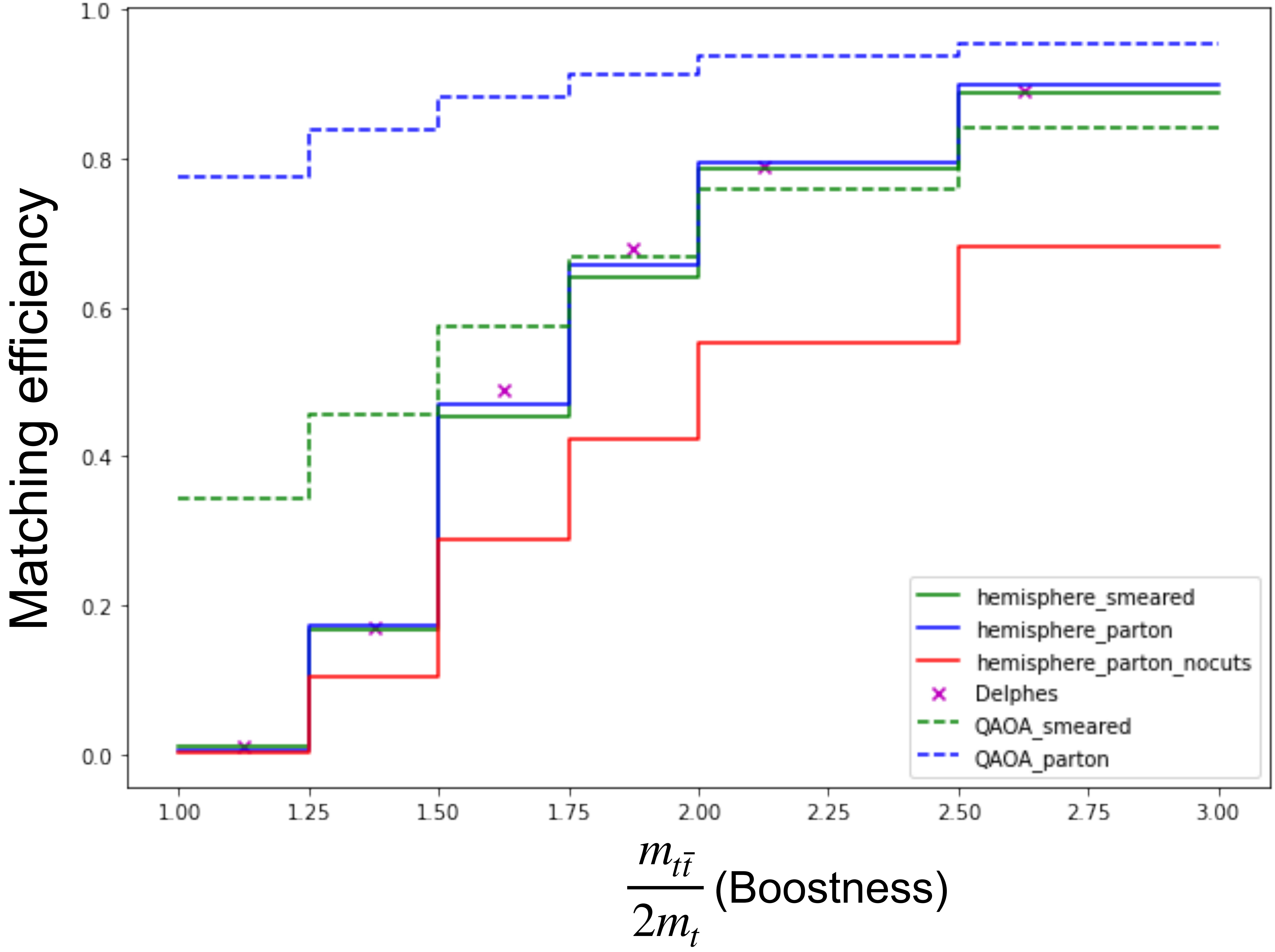


FALQON seems to work better near threshold.

Annealer seems to work better in the boosted region.

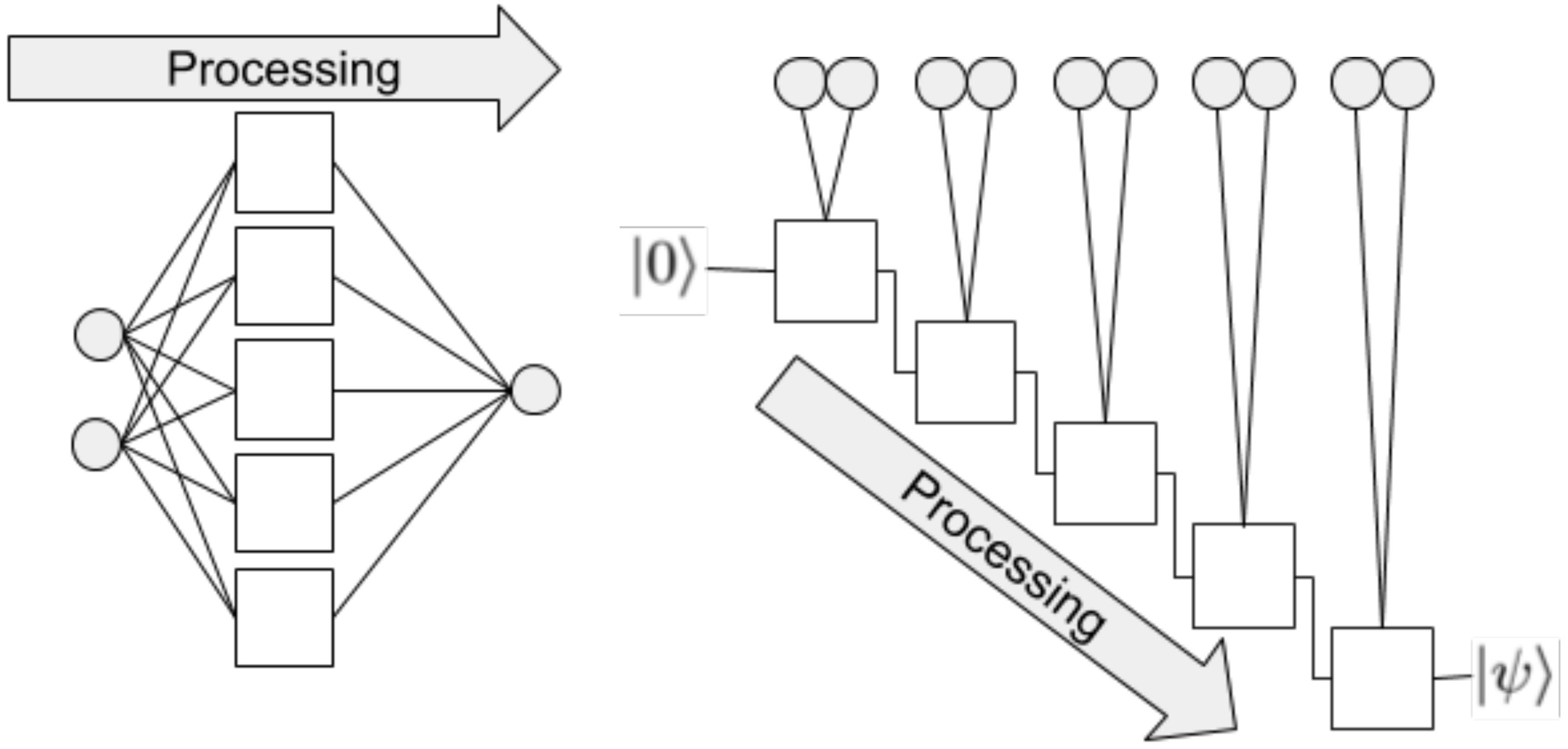
Parton-level events

Preliminary, Dong, Kim, Kong, Park, Scott


Matching efficiency
1.0
0.8
0.6
0.4
0.2
0.0
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
$\frac{m_{t\bar{t}}}{2m_t}$ (Boostness)
hemisphere_smeared
hemisphere_parton
hemisphere_parton_nocuts
Delphes
QAOA_smeared
QAOA_parton

Data re-uploading for a universal quantum classifier

1907.02085



(a) Neural network (b) Quantum classifier

- Universal approximation theorem
- We can approximate a function $F(\vec{x})$ with $f(\vec{x}, \vec{\theta})$, where $\vec{x}$ is an input feature and $\vec{\theta}$ is a learnable parameter.
- The cost function (ex. MSE) to be minimized is $\sum_{i=1}^{n} |F(\vec{x}_i) - f(\vec{x}_i, \vec{\theta})|^2$

Single qubit classifier using data re-uploading

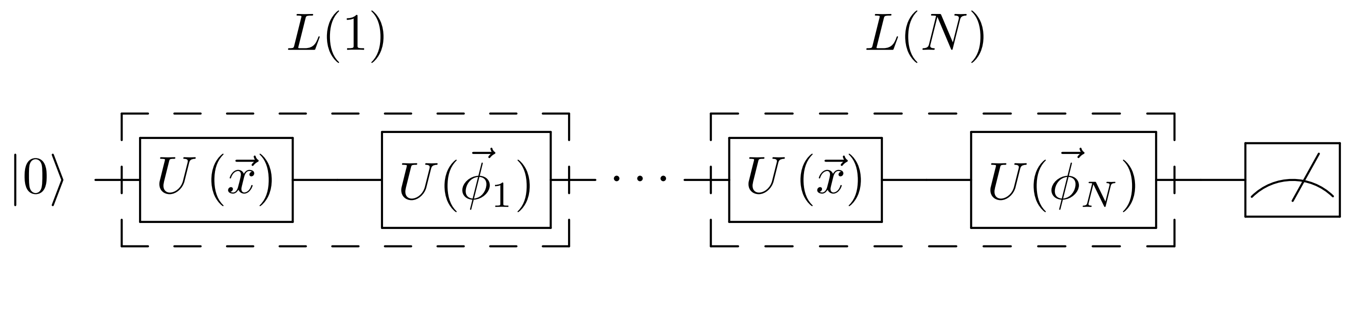
1907.02085

- Consider the three dimensional data, $\vec{x}$. (can be generalized.)
- Date can be re-uploaded using unitary transformation $U(\vec{x})$ rotating the qubit.
- The single-qubit classifier has the following structure:

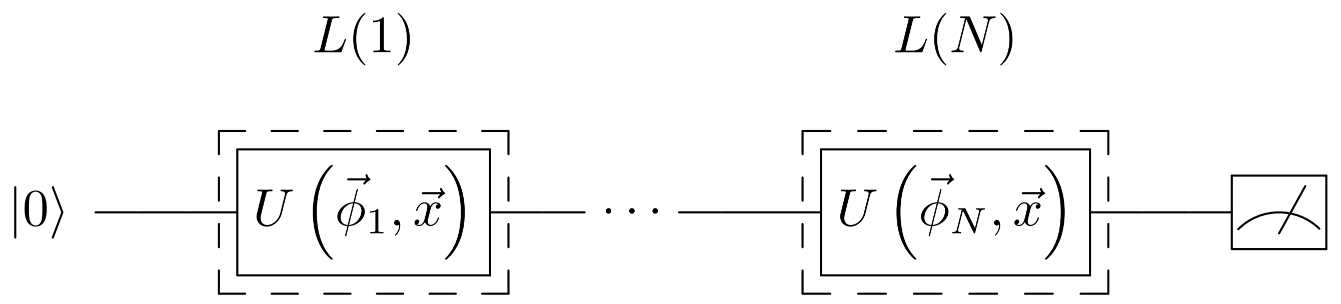
$$|\psi\rangle = \mathcal{U}(\vec{\phi}, \vec{x})|0\rangle$$

$$\mathcal{U}(\vec{\phi}, \vec{x}) = L(N) \dots L(1) \qquad L(i) \equiv U(\vec{\phi}_i)U(\vec{x}) \qquad \overrightarrow{\phi} = (\phi_1, \phi_2, \phi_3)$$

$$\mathcal{U}(\vec{\phi}, \vec{x}) \equiv U(\vec{\phi}_N)U(\vec{x}) \dots U(\vec{\phi}_1)U(\vec{x}) \qquad U(\phi_1, \phi_2, \phi_3) \in SU(2)$$



(a) Original scheme



$L(1)$
$L(N)$
$|0\rangle$
$U\left(\vec{\phi}_1, \vec{x}\right)$
$U\left(\vec{\phi}_N, \vec{x}\right)$

(b) Compressed scheme

$$L(i) = U\left(\vec{\theta}_i + \vec{w}_i \circ \vec{x}\right)$$

Hadamard product of $\overrightarrow{w}_i$ and $\vec{x}$:

$$\vec{w}_i \circ \vec{x} = \left(w_i^1 x^1, w_i^2 x^2, w_i^3 x^3\right)$$

$$L(i) = U\left(\vec{\theta}_i^{(k)} + \vec{w}_i^{(k)} \circ \vec{x}^{(k)}\right) \cdots U\left(\vec{\theta}_i^{(1)} + \vec{w}_i^{(1)} \circ \vec{x}^{(1)}\right)$$

Single qubit classifier: measurements

- The quantum circuit characterized by a series of processing angle $\{\theta_i\}$ and weights $\{w_i\}$ delivers a final state $|\psi\rangle$.

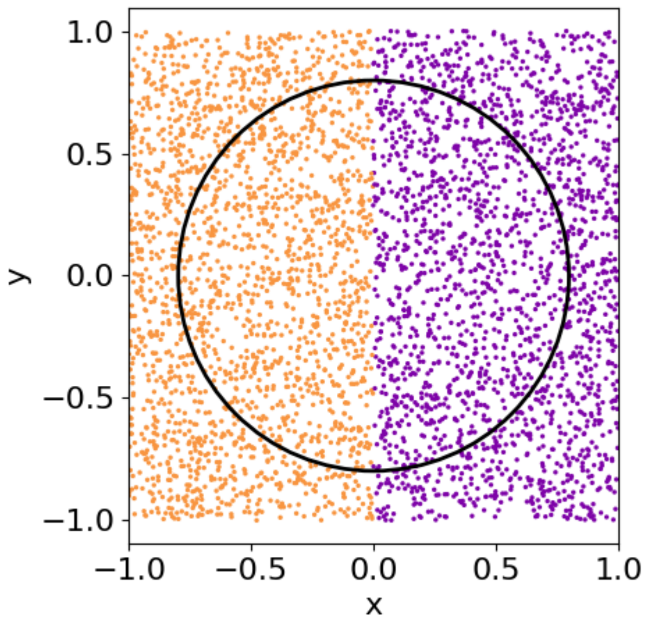
  $|\psi\rangle = \mathcal{U}(\vec{\phi}, \vec{x})|0\rangle$
- The critical point in the quantum measurement is to find an optimal way to associate outputs from the observations to target classes.
- This is easily established for a dichotomic classification, where one of two classes A and B have to be assigned to the final measurement of the single qubit.
- In such a case it is possible to measure the output probabilities $P(0)$ for $|0\rangle$ and $P(1)$ for $|1\rangle$. A given pattern could be classified into the A class if $P(0) > P(1)$ and into B otherwise.
- We may refine this criterium by introducing a bias. That is, the pattern is classified as A if $P(0) > \lambda$, and as B otherwise. The λ is chosen to optimize the success of classification on a training set.
- The assignment of classes to the output reading of a single qubit becomes an involved issue when many classes are present.
  - For example, one possible strategy consists on comparing the probability $P(0)$ to four sectors with three thresholds: $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq 1$. Then, the value of $P(0)$ will fall into one of them, and classification is issued.

1907.02085

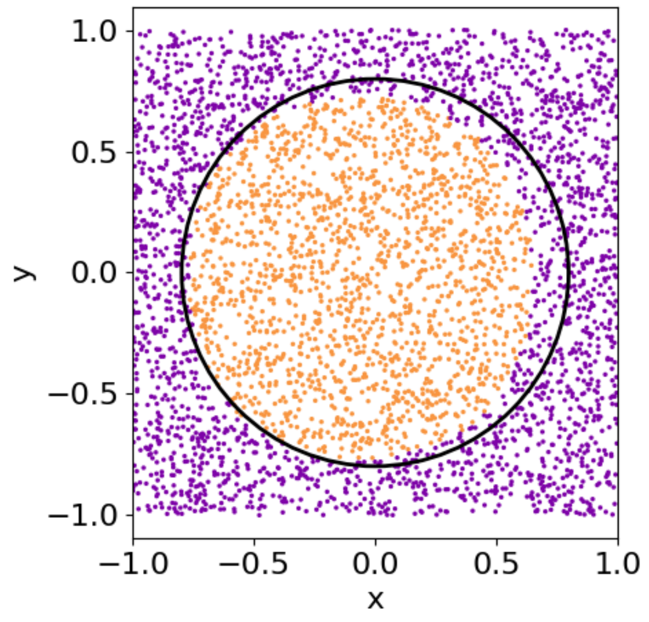
Single qubit classifier: cost function

- A fidelity cost function:
  - We want to force the quantum state (data state) $|\psi(\vec{\theta}, \vec{w}, \vec{x})\rangle$ to be as near as possible to one particular state (label state) on the Bloch sphere.
  - The angular distance between the label state and the data state can be measured with the relative fidelity between the two states.
  - Goal is to maximize the average fidelity
  - $\chi_f^2(\vec{\theta}, \vec{w}) = \sum_{\mu=1}^{M} \left(1 - \left|\langle \tilde{\psi}_s | \psi(\vec{\theta}, \vec{w}, \vec{x}_\mu)\rangle\right|^2\right)$ where $|\tilde{\psi}_s\rangle$ is the correct label state of the μ data point. (M = total number of training data)

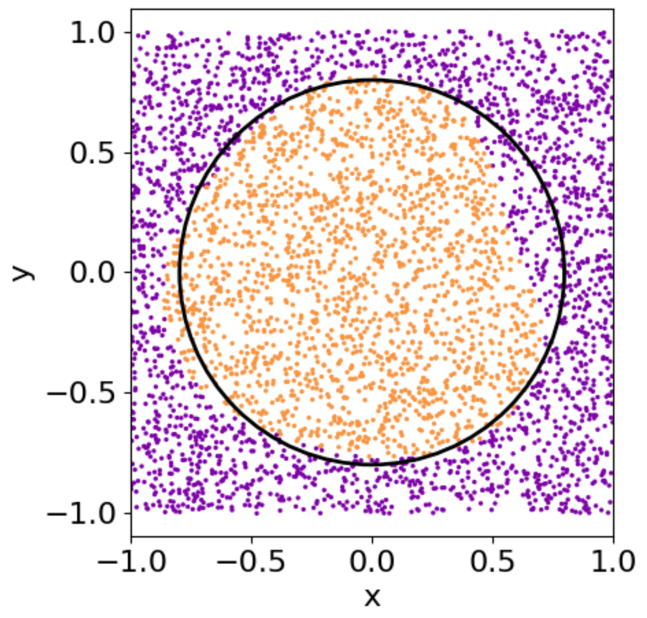
Single qubit classifier: example



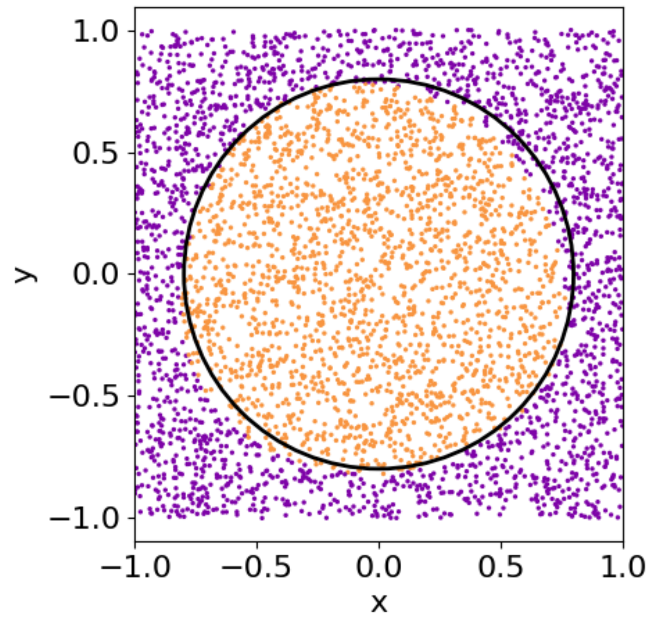
(a) 1 layer



(b) 2 layers



(c) 4 layers

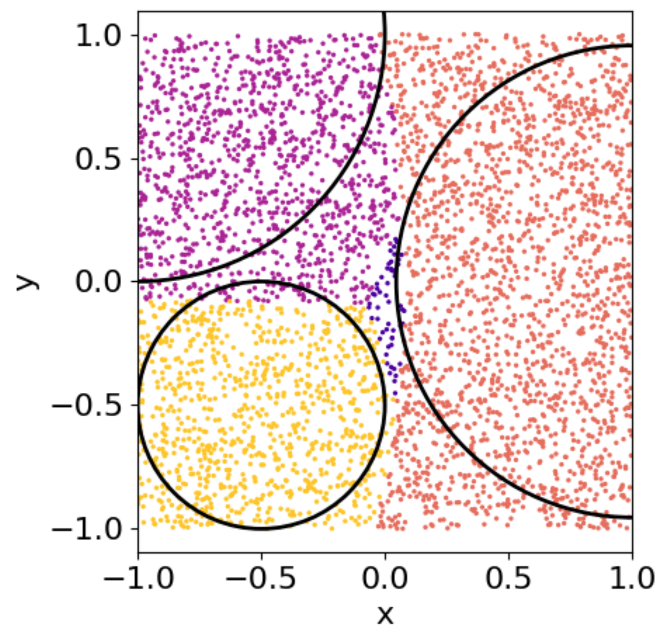


(d) 8 layers

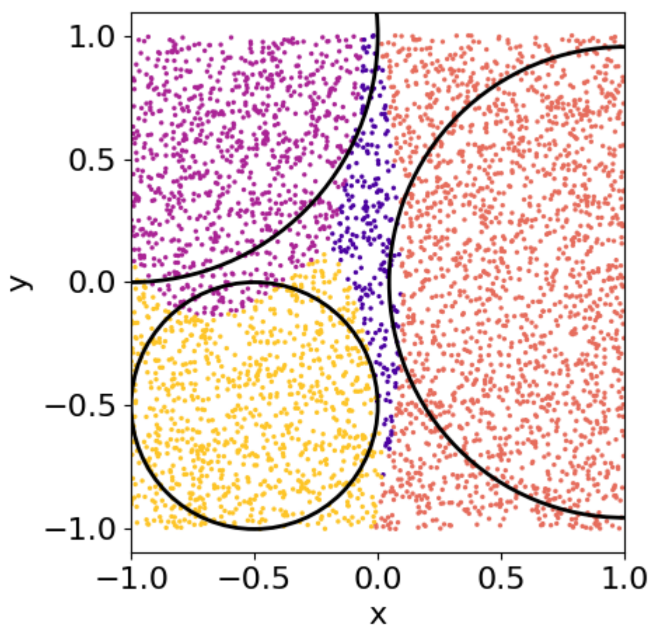
	χ^2_f		
Qubits	1	2	
Layers		No Ent.	Ent.
1	0.50	0.75	–
2	0.85	0.80	0.73
3	0.85	0.81	0.93
4	0.90	0.87	0.87
5	0.89	0.90	0.93
6	0.92	0.92	0.90
8	0.93	0.93	0.96
10	0.95	0.94	0.96

1907.02085

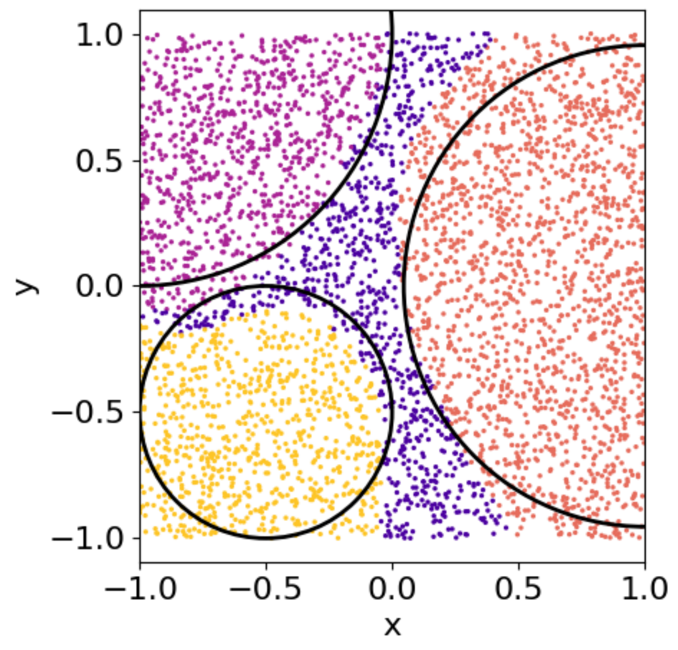
Single qubit classifier: example



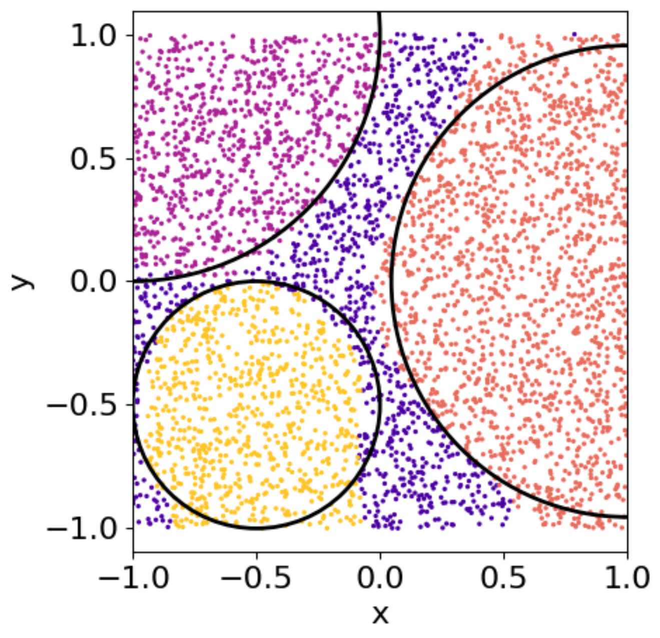
(a) 1 layer



(b) 3 layers



(c) 4 layers

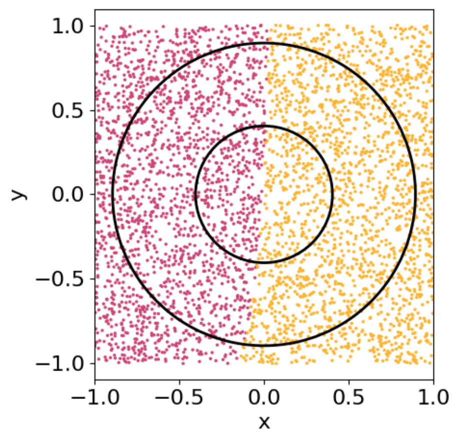


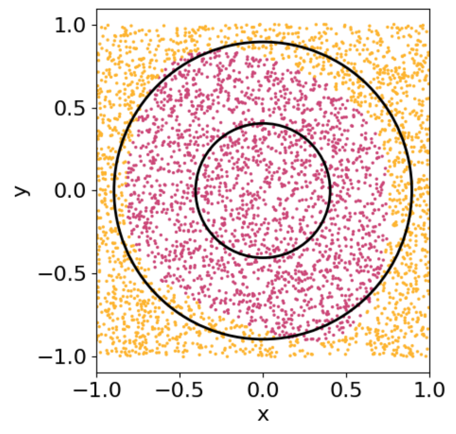
(d) 10 layers

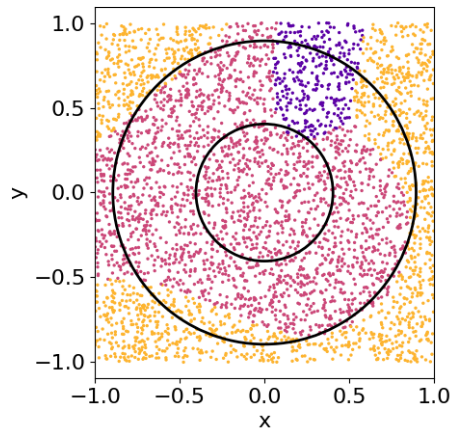
	χ_f^2		
Qubits	1	2	
Layers		No Ent.	Ent.
1	0.73	0.56	–
2	0.79	0.77	0.78
3	0.79	0.76	0.75
4	0.84	0.80	0.80
5	0.87	0.84	0.81
6	0.90	0.88	0.86
8	0.89	0.85	0.89
10	0.91	0.86	0.90

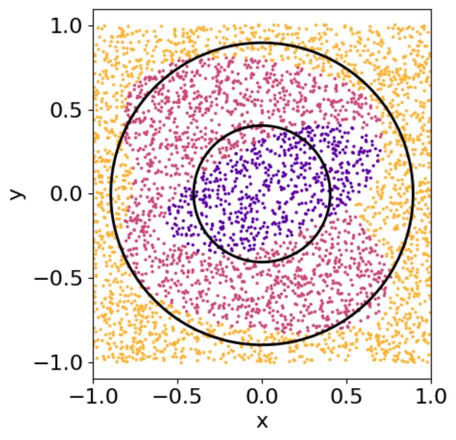
1907.02085

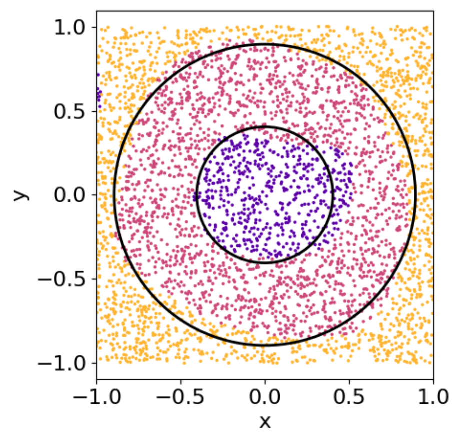
Single qubit classifier: example

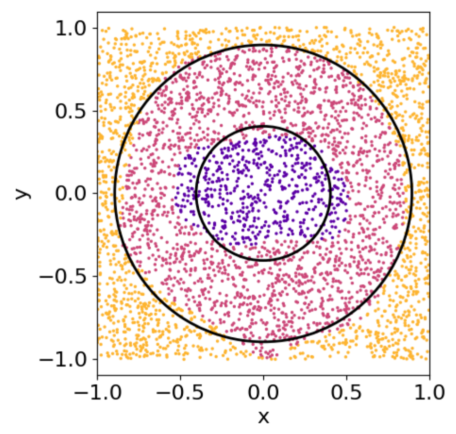

(a) 1 layer

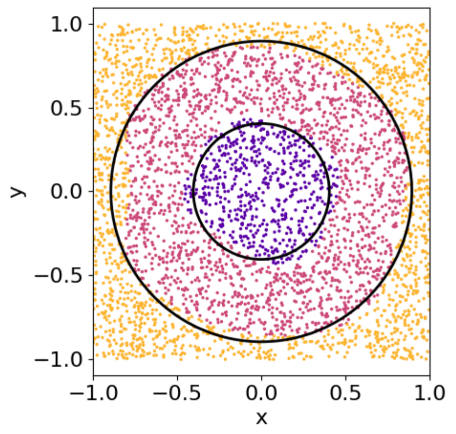

(b) 2 layers

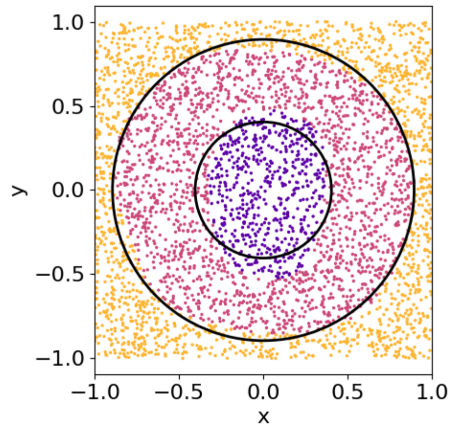

(c) 3 layers


(d) 4 layers


(e) 5 layers


(f) 6 layers


(g) 8 layers


(h) 10 layers

	χ_f^2		
Qubits	1	2	
Layers		No Ent.	Ent.
1	0.34	0.51	–
2	0.57	0.63	0.59
3	0.80	0.68	0.65
4	0.84	0.78	0.89
5	0.92	0.86	0.82
6	0.93	0.91	0.93
8	0.90	0.89	0.90
10	0.90	0.91	0.92

1907.02085

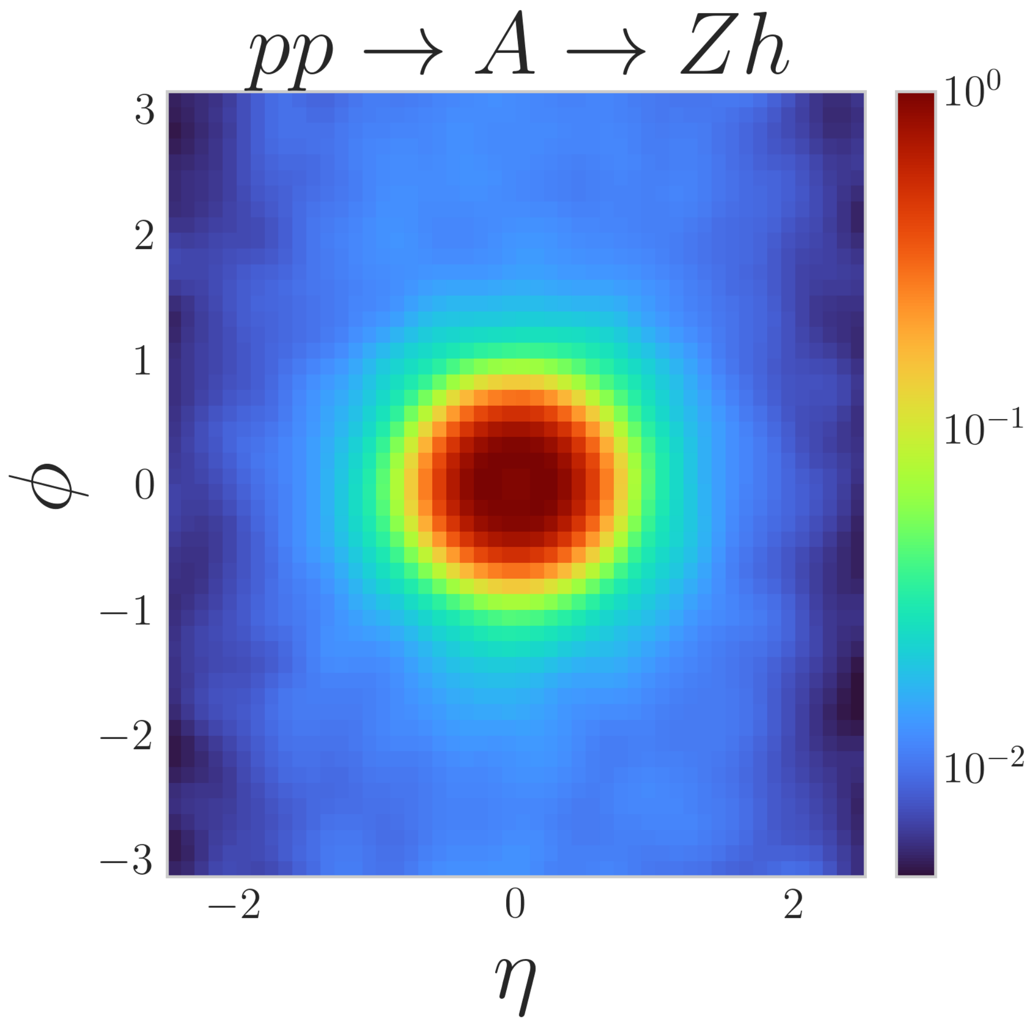
Single qubit classifier: example

1907.02085

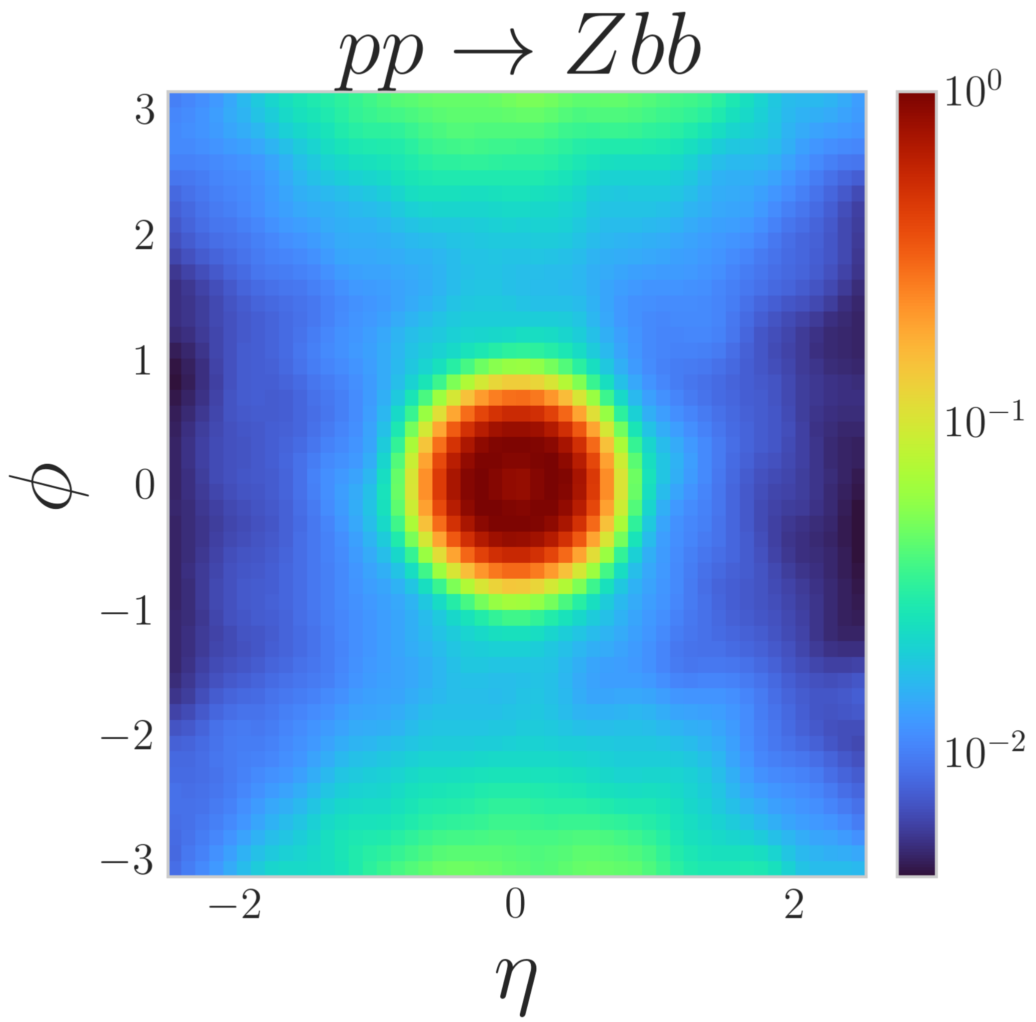
Problem	Classical classifiers		Quantum classifier	
	NN	SVC	χ^2_f	χ^2_{wf}
Circle	0.96	0.97	0.96	0.97
3 circles	0.88	0.66	0.91	0.91
Hypersphere	0.98	0.95	0.91	0.98
Annulus	0.96	0.77	0.93	0.97
Non-Convex	0.99	0.77	0.96	0.98
Binary annulus	0.94	0.79	0.95	0.97
Sphere	0.97	0.95	0.93	0.96
Squares	0.98	0.96	0.99	0.95
Wavy Lines	0.95	0.82	0.93	0.94

Comparison between single-qubit quantum classifier and two well-known classical classification techniques: a neural network (NN) with a single hidden layer composed of 100 neurons and a support vector classifier (SVC), both with the default parameters as defined in scikit-learn python package. We analyze nine problems: the first four are presented in Section 6 and the remaining five in Appendix B. Results of the single-qubit quantum classifier are obtained with the fidelity and weighted fidelity cost functions, $\chi^2{}_f$ and $\chi^2{}_{wf}$ defined in Eq. (7) and Eq. (9) respectively. This table shows the best success rate, being 1 the perfect classification, obtained after running ten times the NN and SVC algorithms and the best results obtained with single-qubit classifiers up to 10 layers.

Self-Supervised Quantum Metric Learning



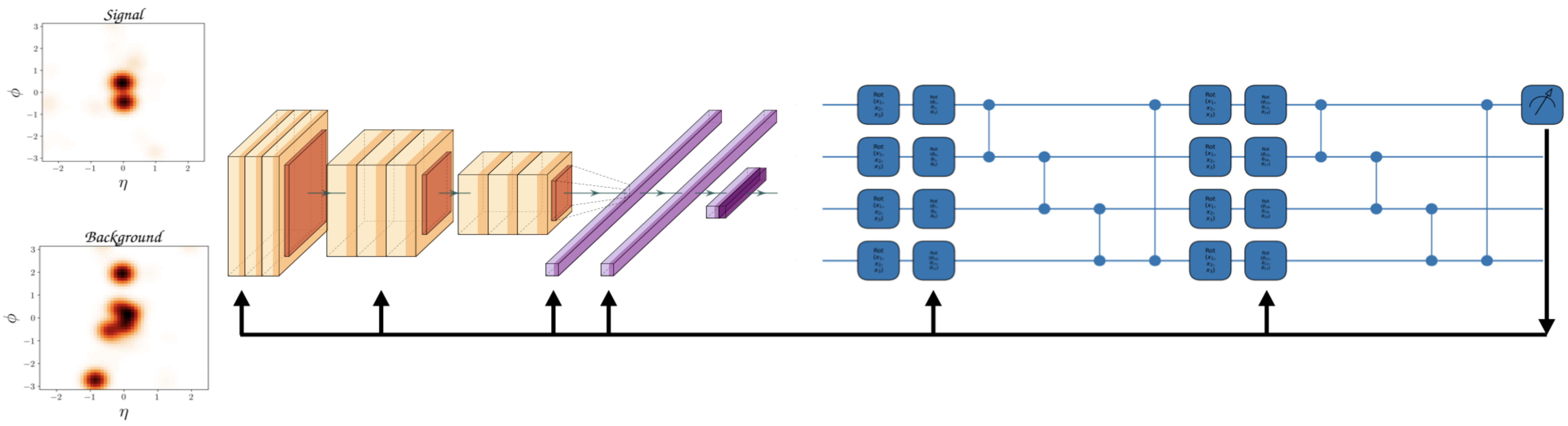
Jet image for 50k signal events

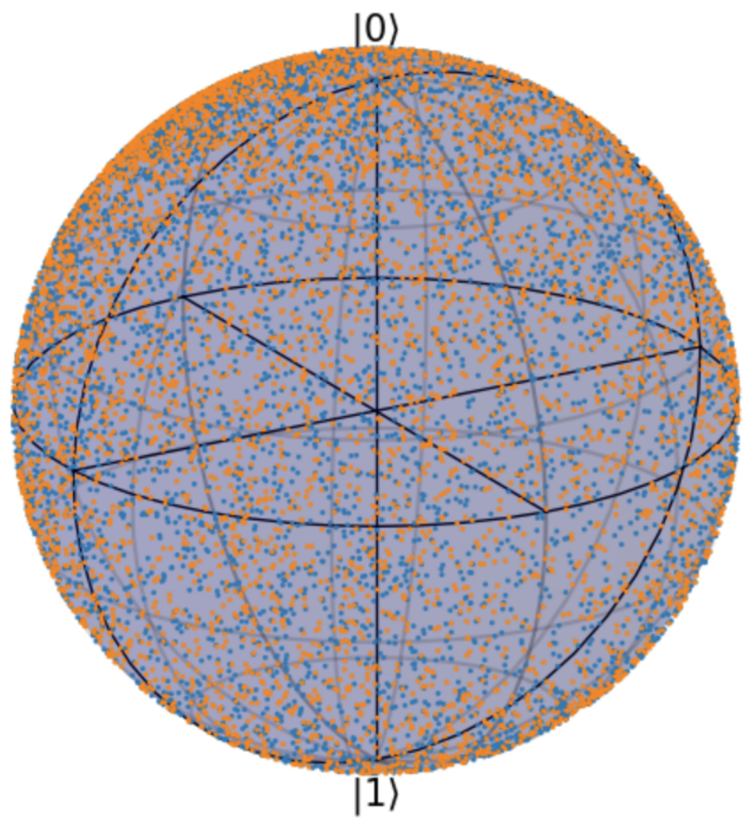


Jet image for 50k background events

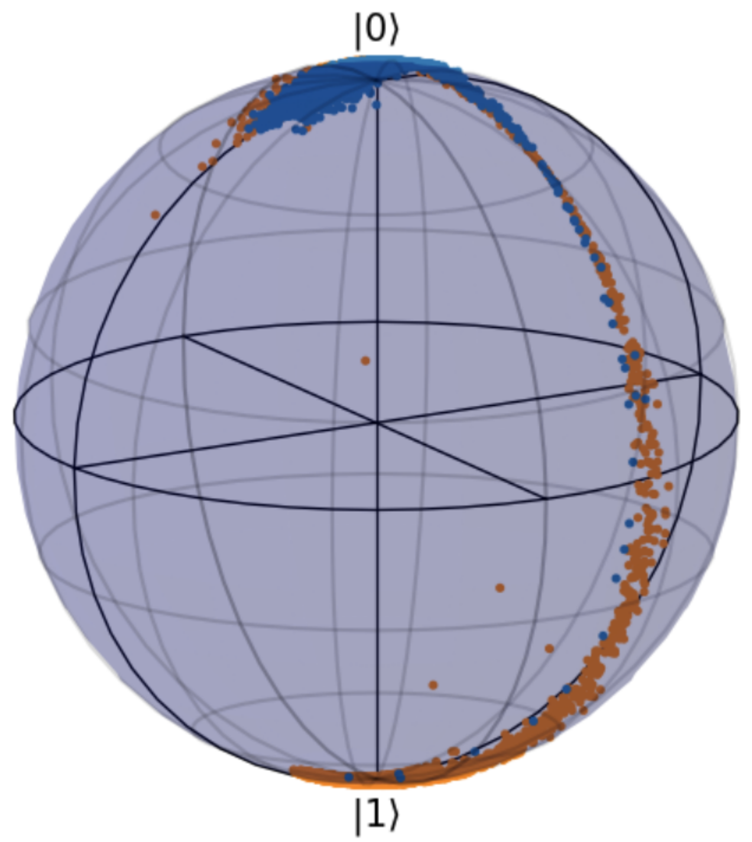
Preliminary, Hammad, Kong, Park, Shim

Preliminary, Hammad, Kong, Park, Shim

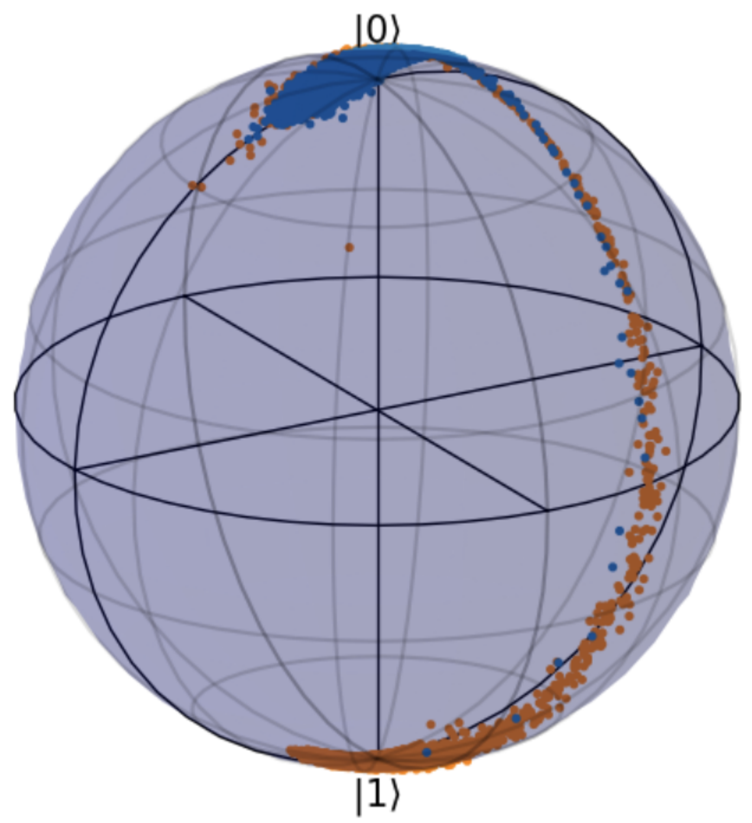




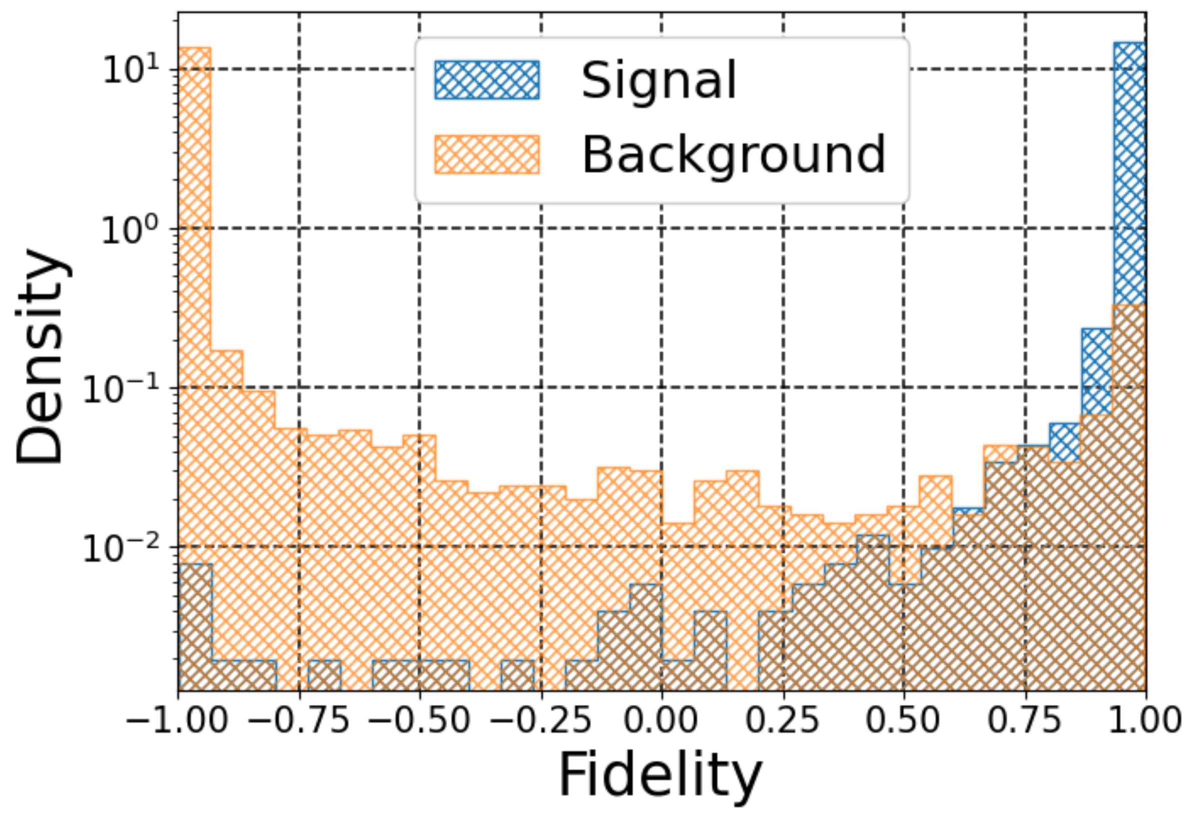
Data before training (10k events)

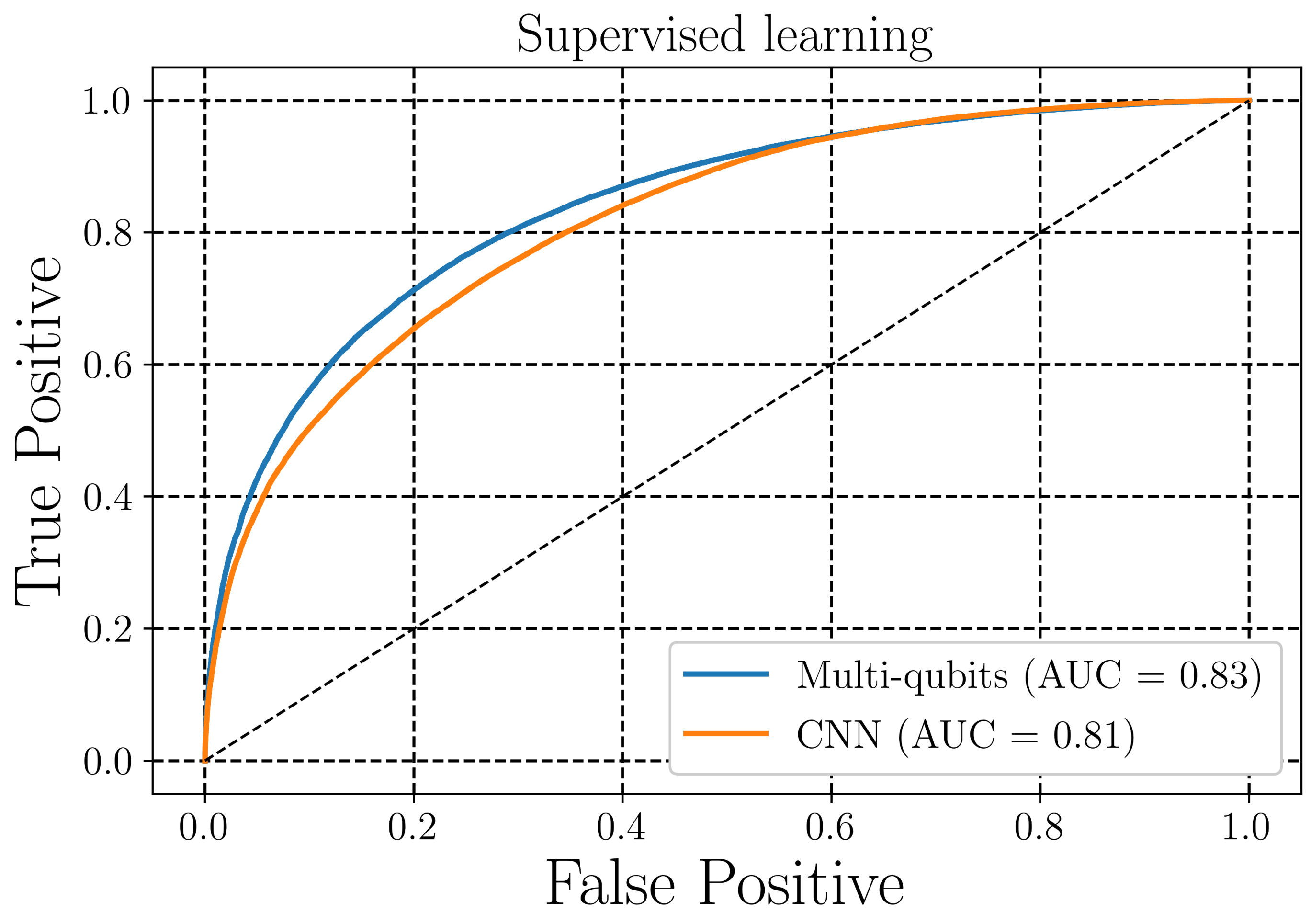


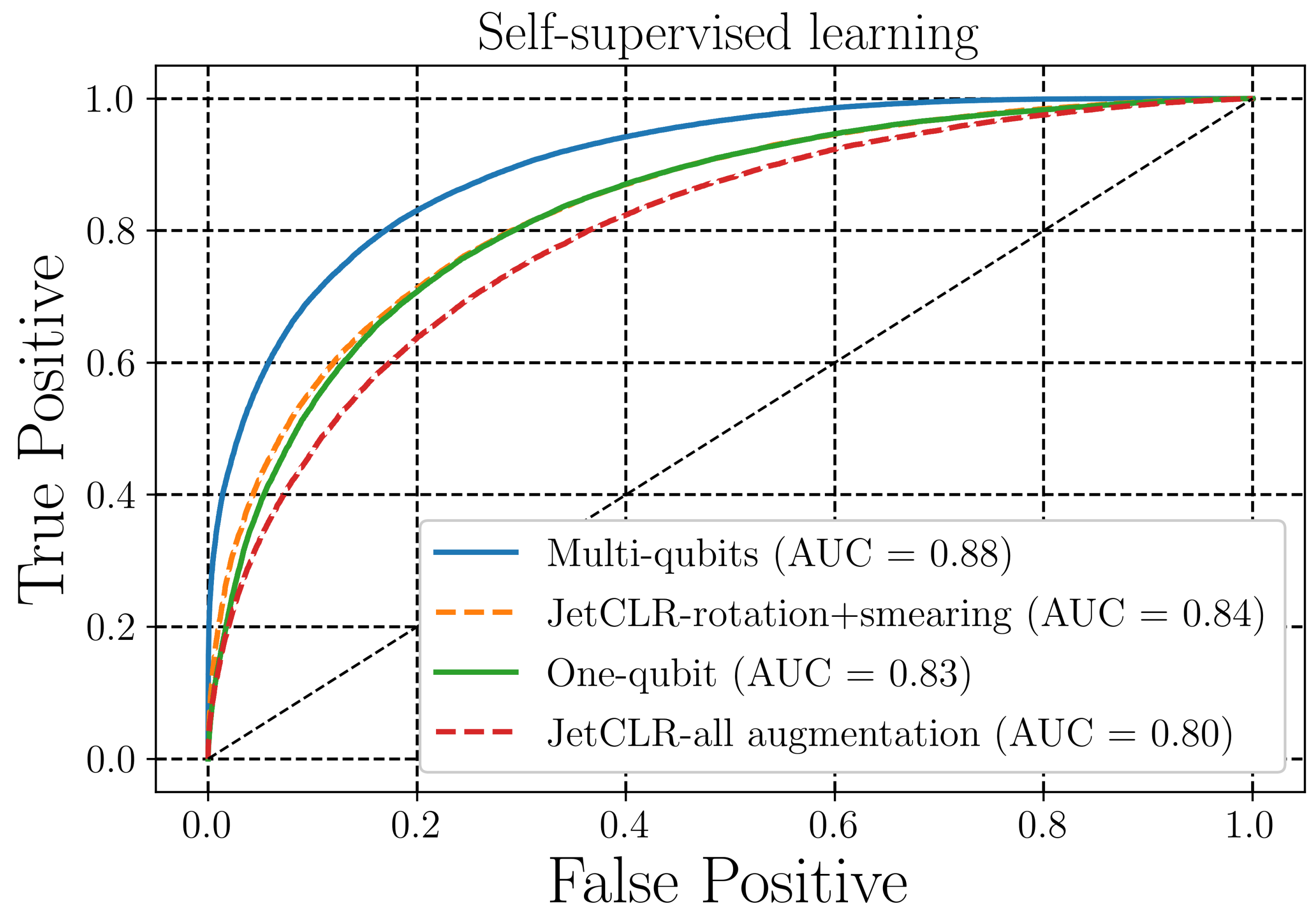
Training data after 50 epochs



Testing data (15k events)







Preliminary, Hammad, Kong, Park, Shim

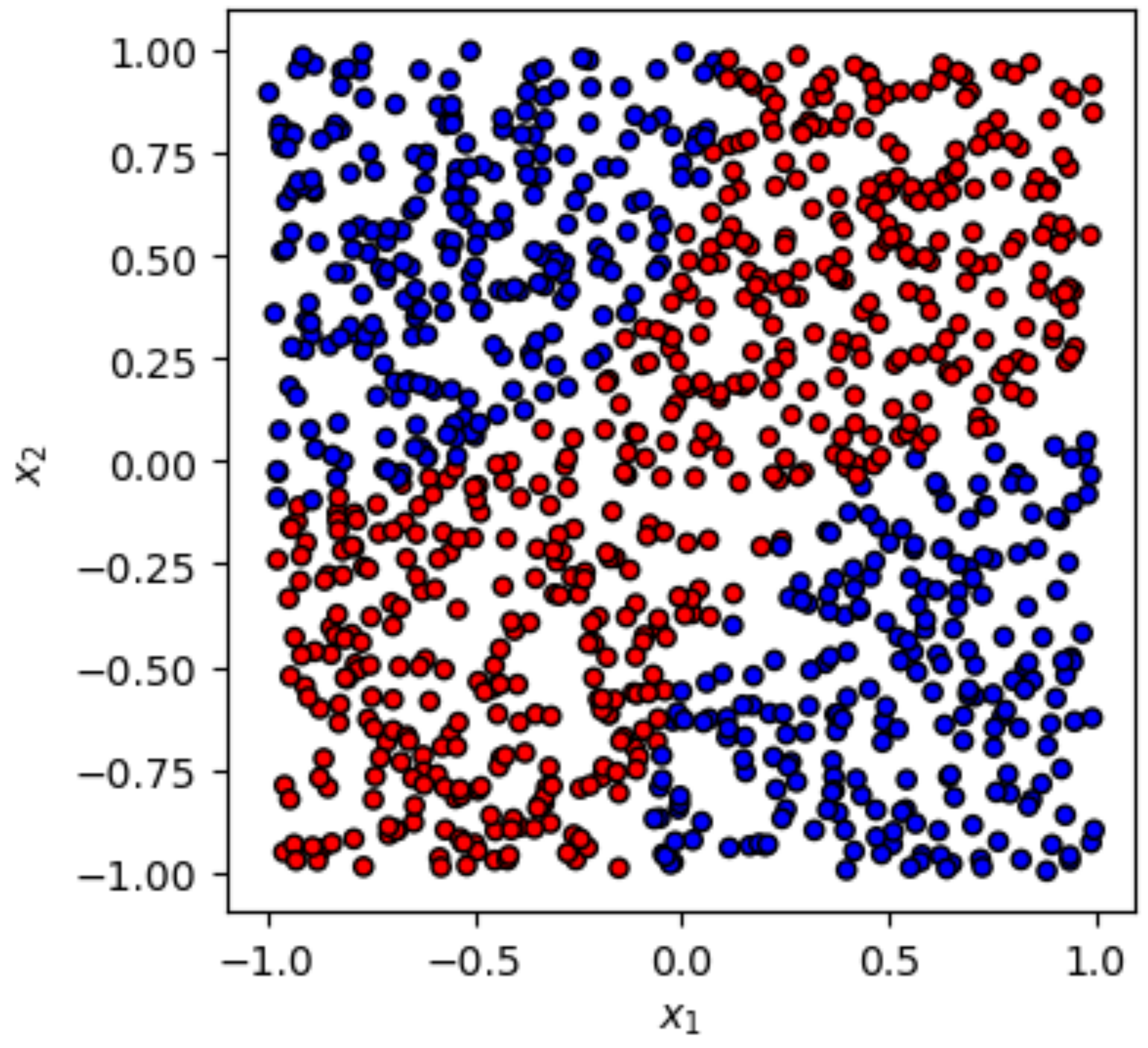
Equivariant Quantum Neural Networks

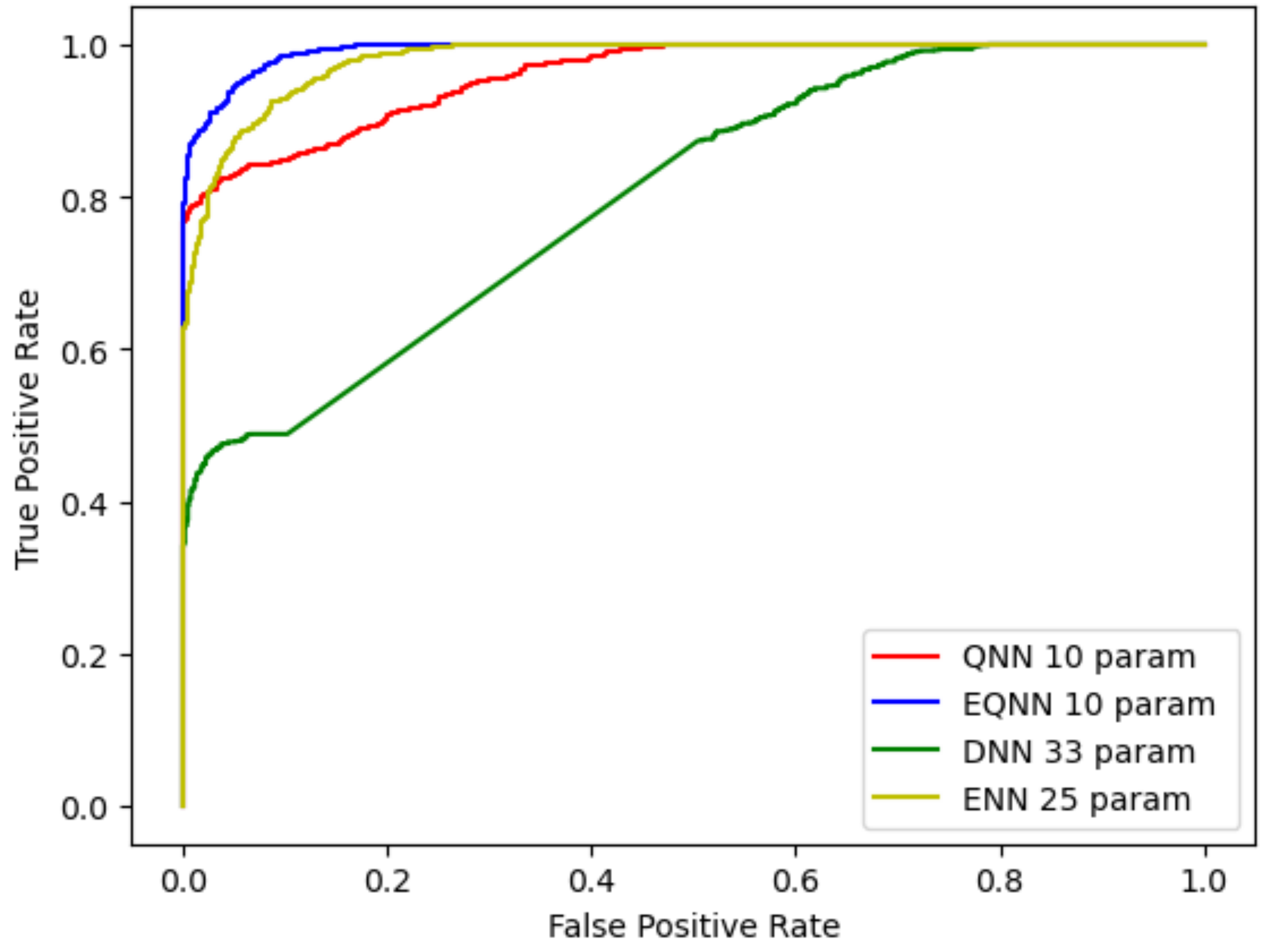
- Given some function $f(x)$, and a symmetry transformation on the input $\pi(x)$, equivariance means the following relation

$$f(\pi(x)) = \psi f(x)$$

- We hope that the output of the machine learning model will be invariant under this symmetry.
  $\mathbb{Z}_2 \times \mathbb{Z}_2$ example

Dong




True Positive Rate
False Positive Rate
QNN 10 param
EQNN 10 param
DNN 33 param
ENN 25 param