

Jet Clustering and Matching with QAOA

Dong Liu (THU), Yunheng Ma (BAQIS), Chen Qian (BAQIS),
Manqi Ruan (IHEP), Weifeng Zhuang (BAQIS), Yongfeng Zhu (PKU), Chen Zhou (PKU)

Quantum Computing and Machine Learning Workshop

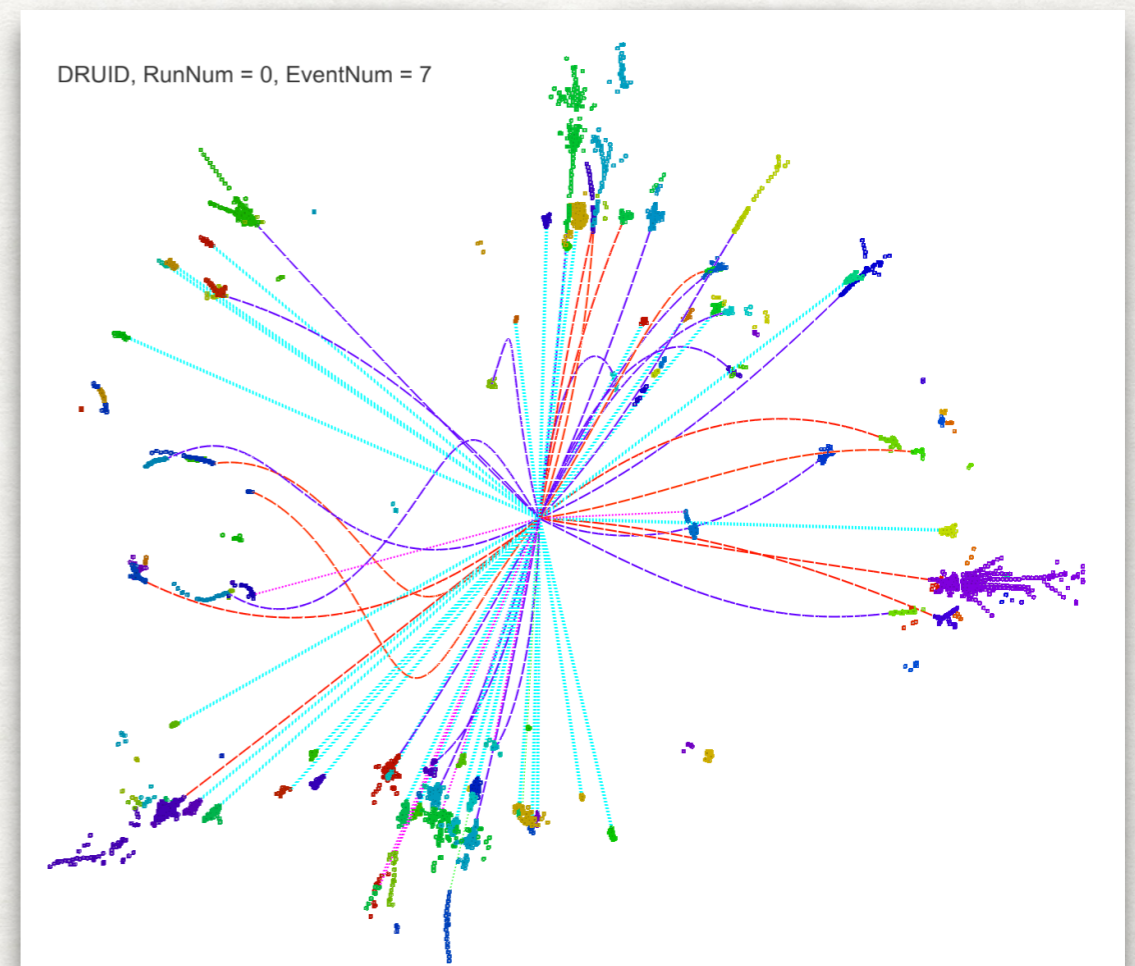
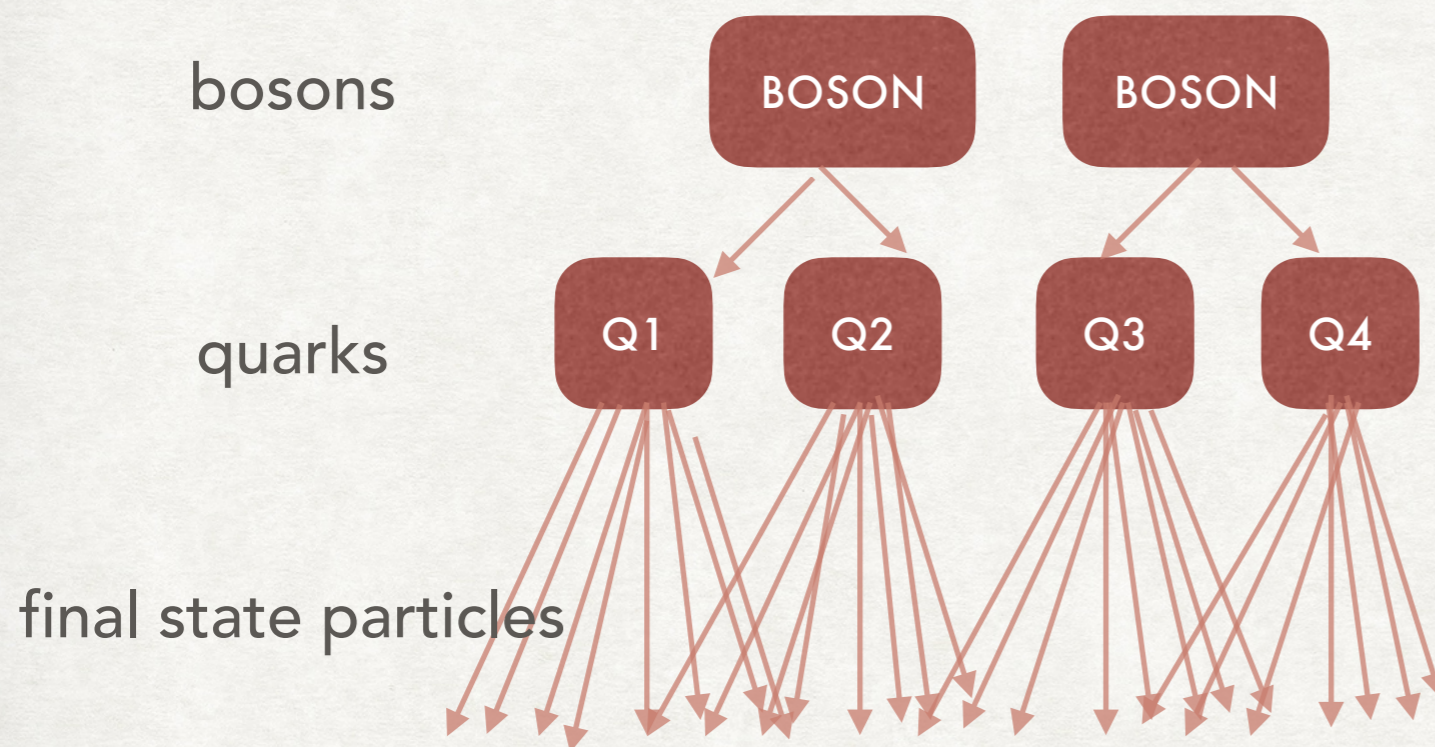
Contents :

- Motivation
- Max-Cut problem
- Variational quantum algorithm (VQA)
- Quantum Approximate Optimization Algorithm (QAOA)
- the results

Motivation :

<https://doi.org/10.1140/epjc/s10052-019-6719-2>

- $e^+e^- \rightarrow WW \text{ or } ZZ$, each boson decay to two quarks, further to two jets, each jet consists of several particles
- goal : reconstruct each boson, then separate WW/ZZ events



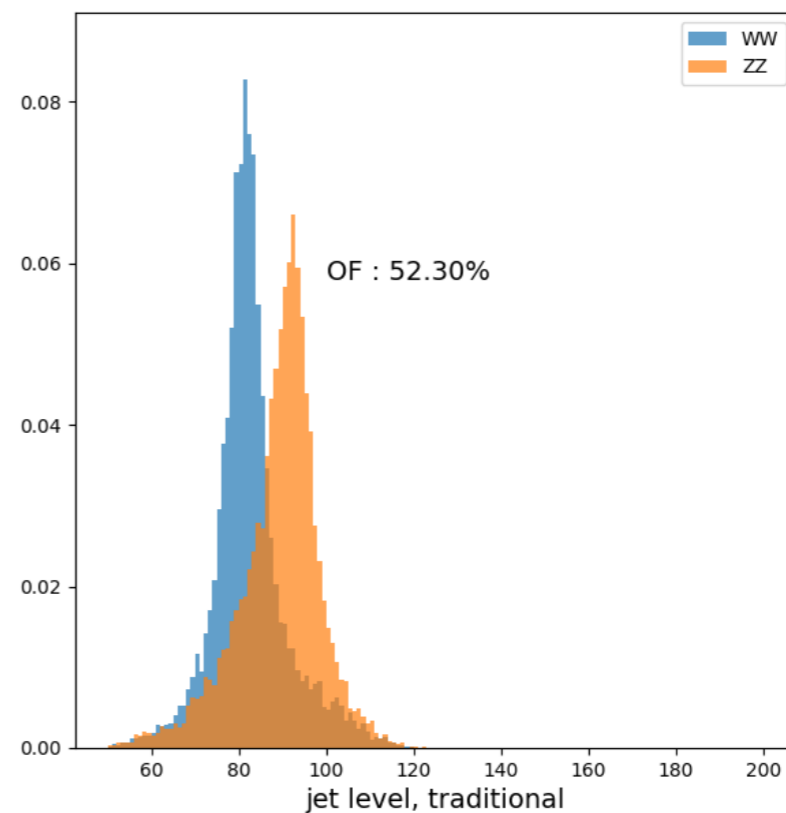
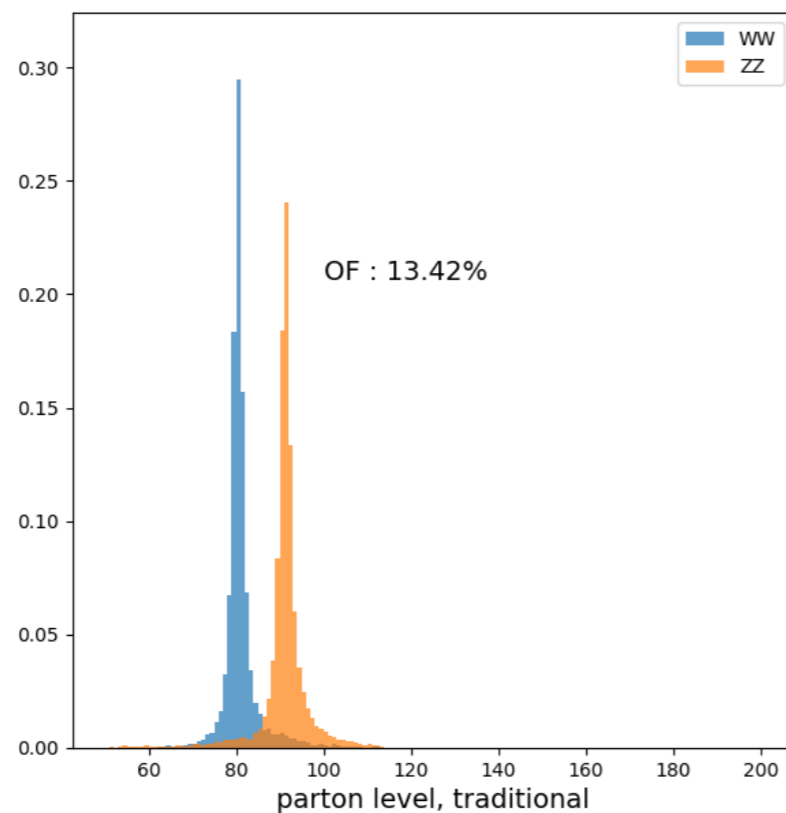
traditional method :

1, reconstruct four jets J_1, J_2, J_3, J_4 with jet clustering algorithm

2, pair four jets to two di-jet systems with method: $\chi^2 = (M_{12} - M_B)^2 + (M_{34} - M_B)^2$, where $M_B : M_W$ or M_Z . There are three pairing conditions : 12 34, 13 24, 14 23, we select the condition with minimal χ^2

3, evaluation criteria : the overlapping fraction of two histograms $0.5 \times (M_{12} + M_{34})$

samples : $e^+e^- \rightarrow WW \rightarrow c\bar{u}s\bar{d}$ and $e^+e^- \rightarrow ZZ \rightarrow \text{down quarks}$

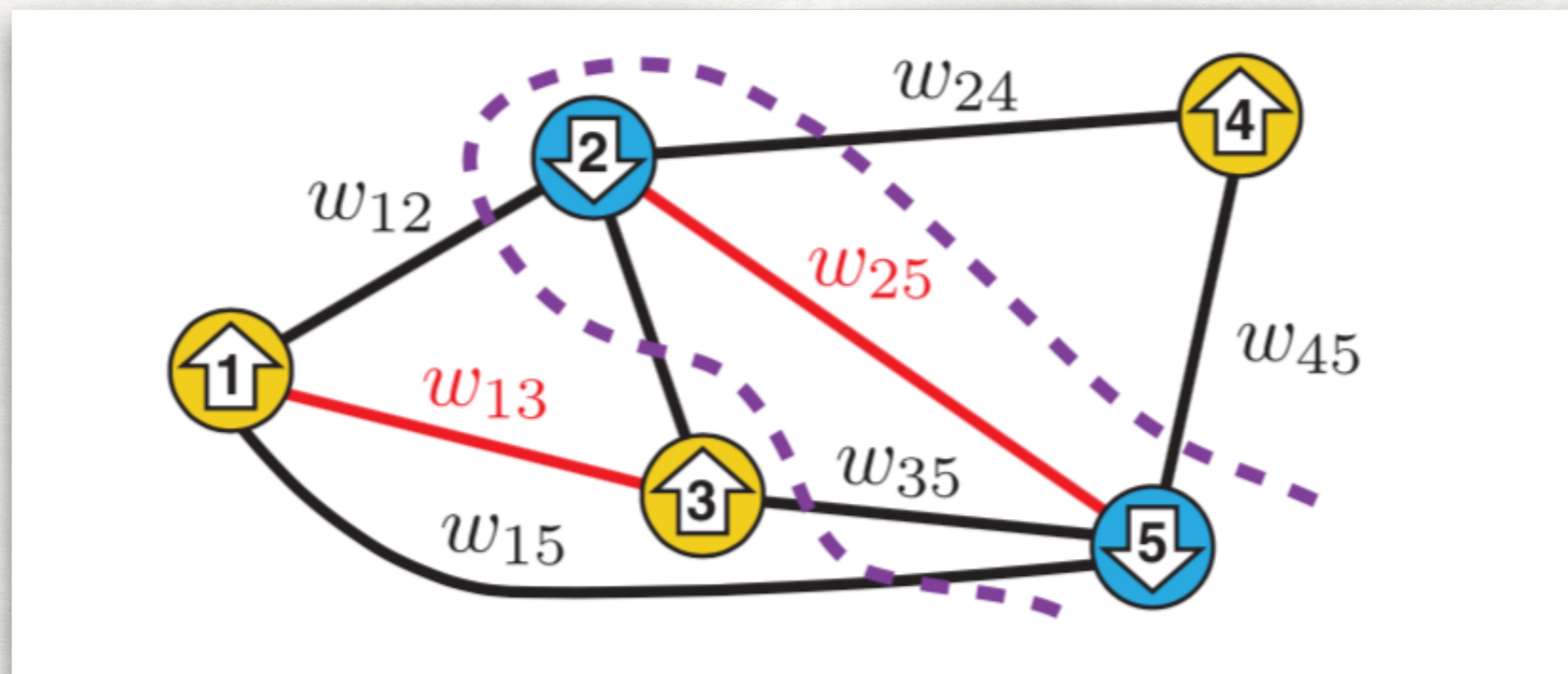


MaxCut :

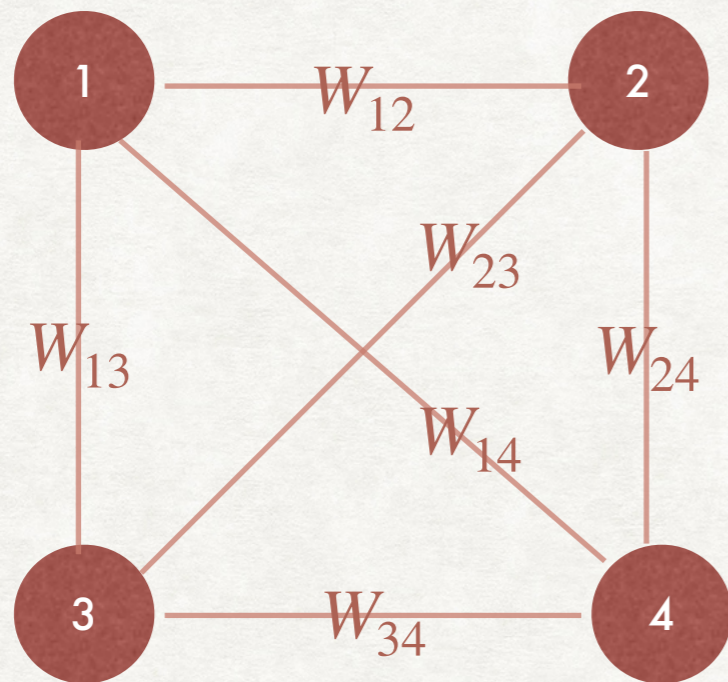
Graph : Set of vertices or nodes V connected by weighted edges E

Cut : Partition of vertices into two disjoint subsets.

To maximize the sum of the weight of the edges with two nodes partitioned.



Map our problem to MaxCut.

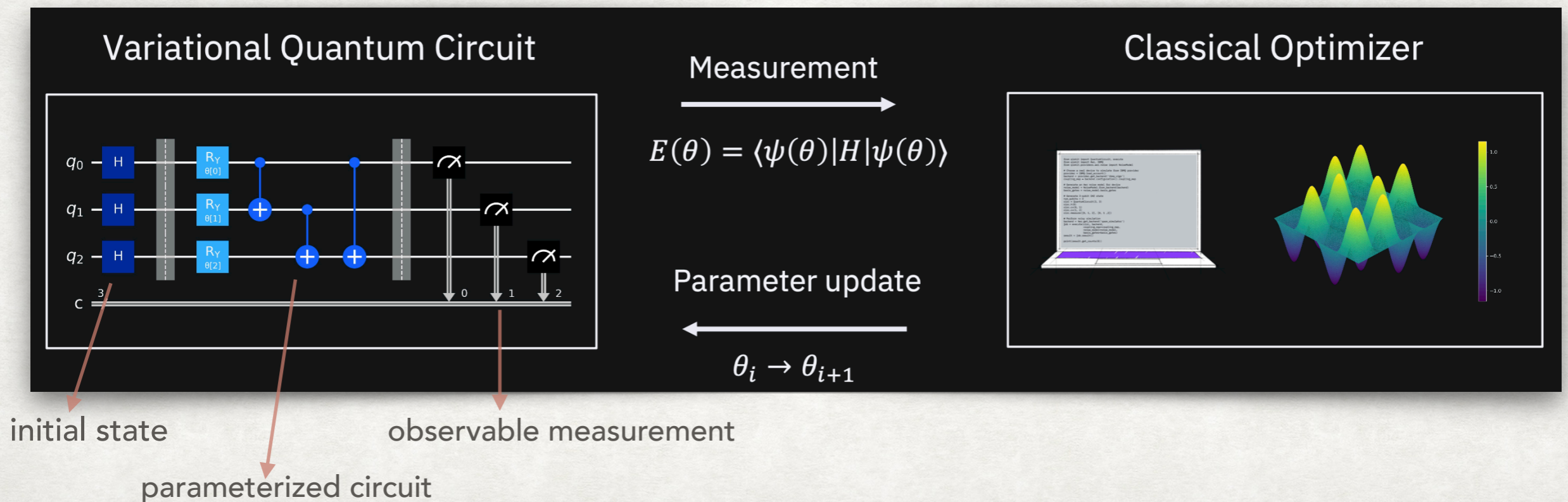


$$W_{ij} = \min(\text{abs}(M_{ij} - M_W), \text{abs}(M_{ij} - M_Z))$$

Variational quantum algorithms :

approximating ground state $|\psi^*\rangle$ and lowest energy E_{min} of a quantum system

- (1) prepare initial state $|\psi_0\rangle$, which is easy to construct
- (2) construct parameterized quantum circuit $U(\theta)$ based on your problem
- (3) get the state $|\psi(\theta)\rangle$ evolved from $|\psi_0\rangle$ through quantum circuit, $|\psi(\theta)\rangle = U(\theta)|\psi_0\rangle$
- (4) calculate $E(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$
- (5) Vary parameters θ to minimize the energy value $E(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$ to get $\theta^* = \underset{\theta}{\operatorname{argmin}} E(\theta)$
- (6) $E(\theta^*)$ is an upper bound for E_{min} and $|\psi(\theta^*)\rangle$ approximates $|\psi^*\rangle$



VQA v.s. ML

parameters update

input

construct

output

cost function

features of samples

model

prediction

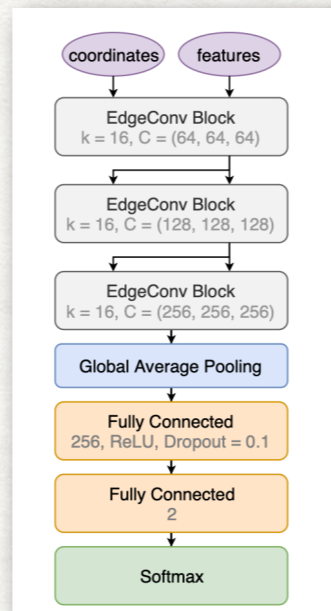
classification
or
regression

$f(y, y')$

ML

- [part_isElectron, null]
- [part_isMuon, null]
- [part_isNeutralHadron, null]
- [part_isPhoton, null]
- [part_d0, null]
- [part_d0err, 0, 1, 0, 1]
- [part_dz, null]
- [part_dzerr, 0, 1, 0, 1]
- [part_deta, null]
- [part_dphi, null]

- [part_pt_log, -1.5, 1.0]
- [part_e_log, -0.687, 1.0]
- [part_logptrel, -4.7, 1.0]
- [part_logerel, -4.473, 1.0]
- [part_deltaR, 2.1, 2.3]
- [part_charge, null]
- [part_isChargedHadron, null]



y'

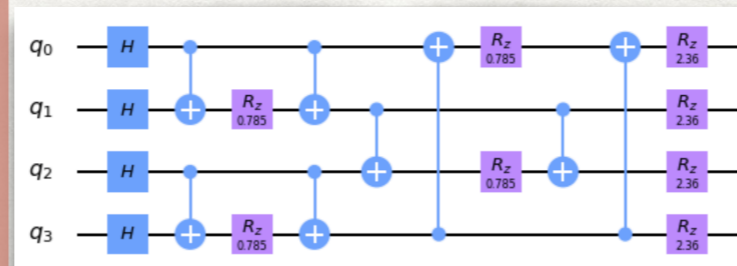
initial state

quantum circuit

final state

VQA

$|\psi(0)\rangle$



$|\psi(\theta)\rangle$

$E(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle$

QAOA for Max-cut problem :

the driver Hamiltonian $H_D : B = \sum_{j=1}^n \sigma_j^x$

the problem Hamiltonian $H_P : C = \frac{1}{2} \sum_{(i,j) \in E} W_{ij} (1 - \sigma_i^z \sigma_j^z)$

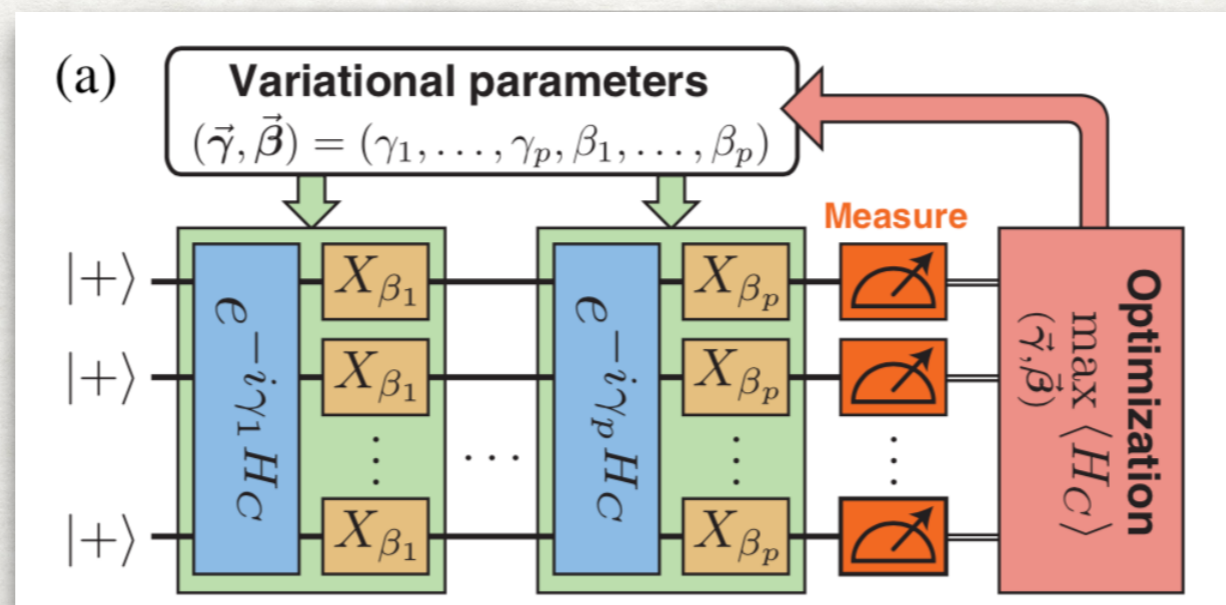
process : 1, letting the system evolve under $H(t) = (1 - s(t))H_D + s(t)H_P$, where $s(0) = 0, s(T) = 1$

adiabatic quantum algorithm to QAOA  see backup

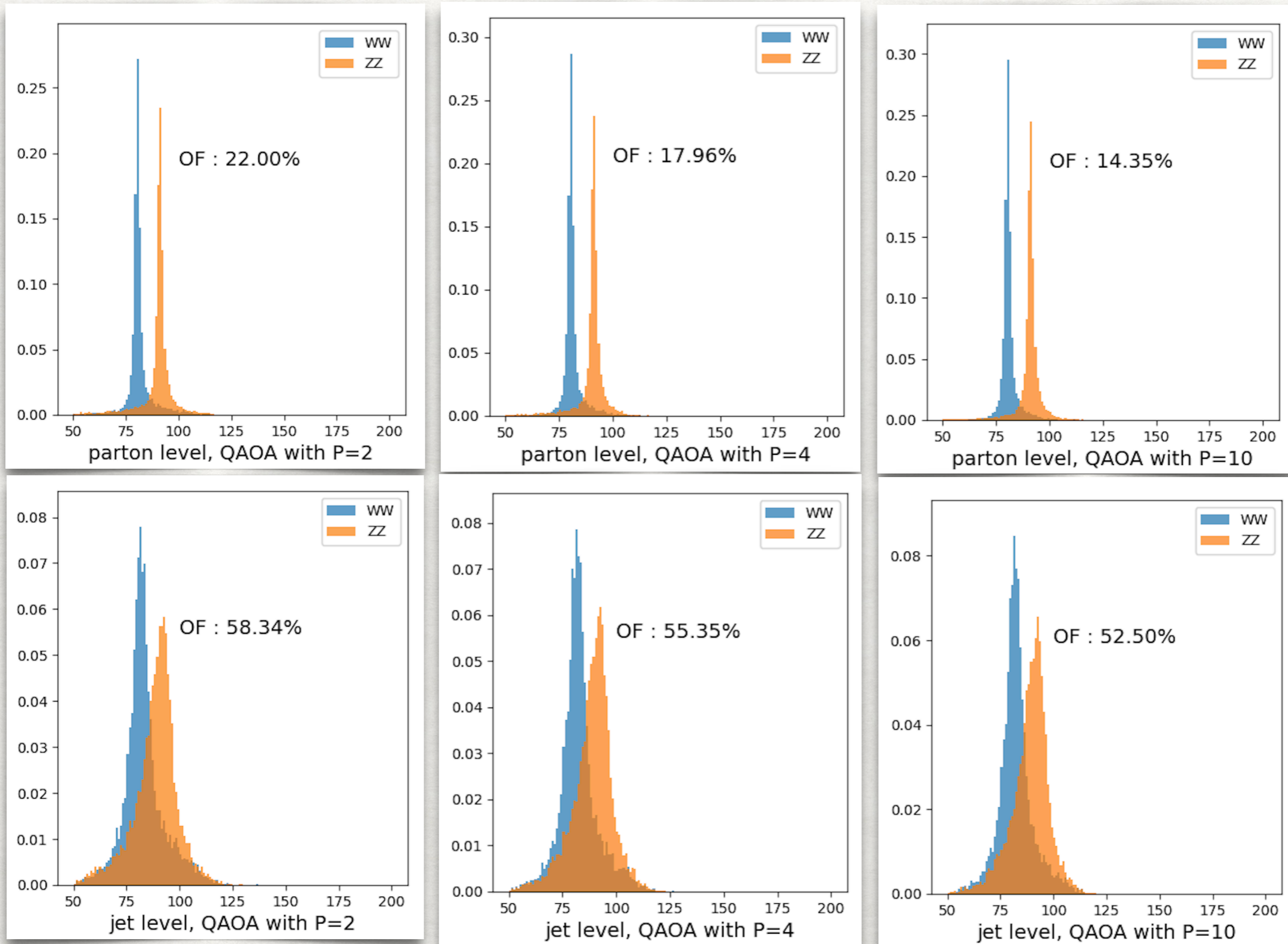
2, letting the system evolve under C for some γ amount of time and under B for some β amount of time respectively.

3, prepare the quantum state $|s\rangle$ and compute $|\gamma, \beta\rangle = U(B, \beta_p)U(C, \gamma_p) \dots U(B, \beta_1)U(C, \gamma_1) |s\rangle$ as $P \rightarrow \infty$, we perfectly replicate AQC

4, measure $|\gamma, \beta\rangle$ in the computational basis to get some bit-string z , and evaluate $C(z)$



comparison of different P values



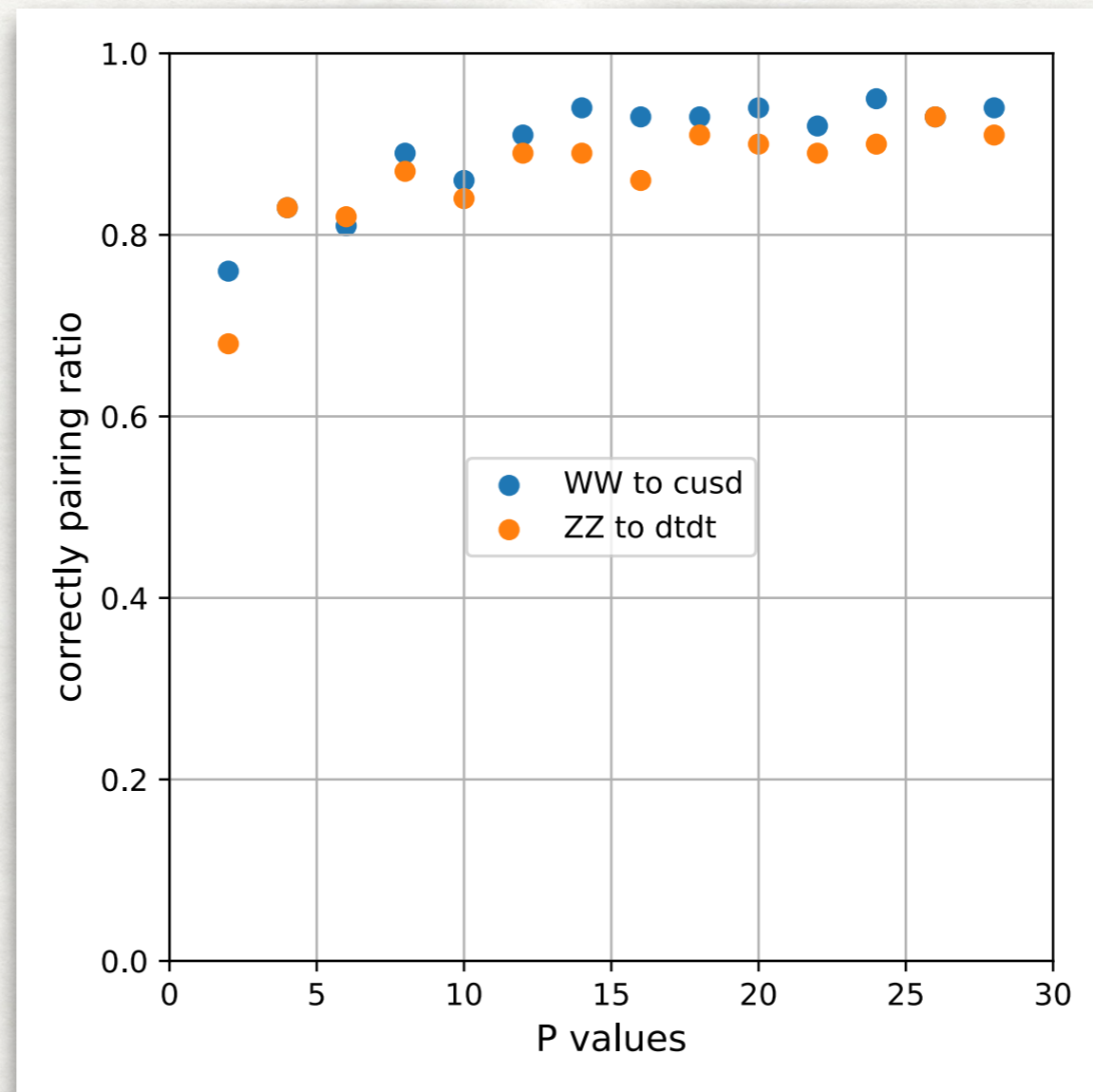
P= 2

P= 4

P= 10

how to determine the P values

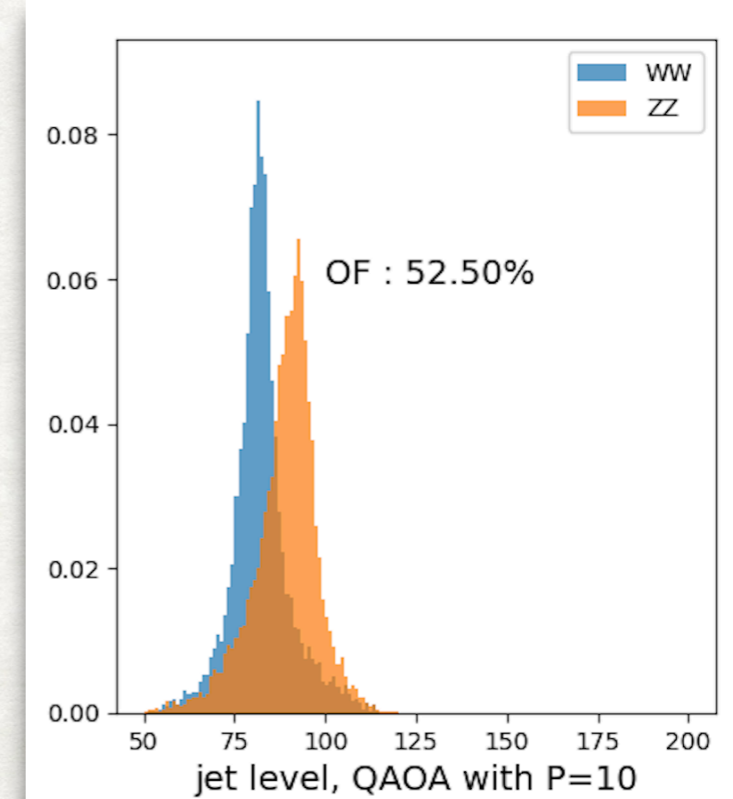
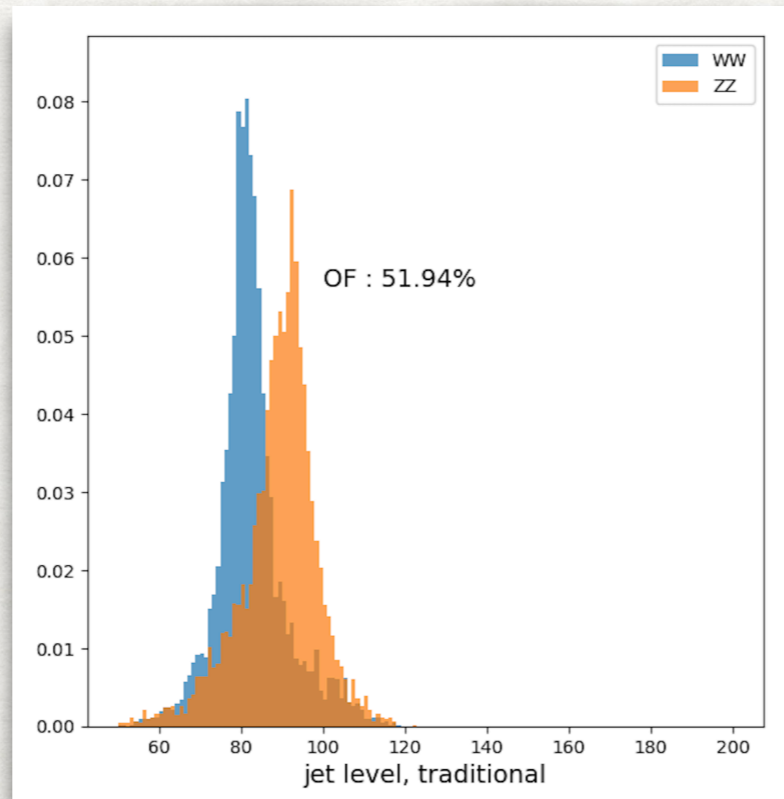
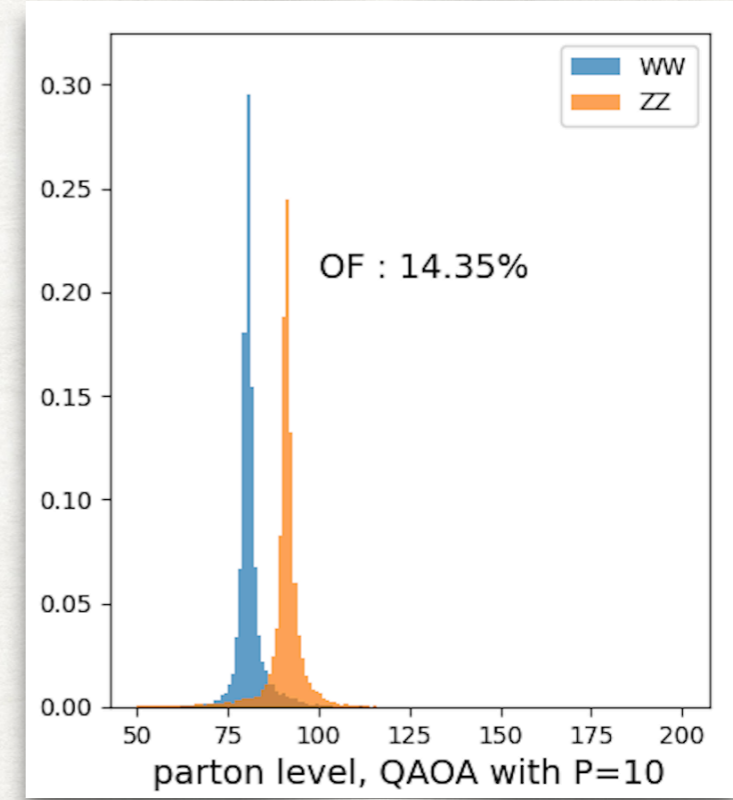
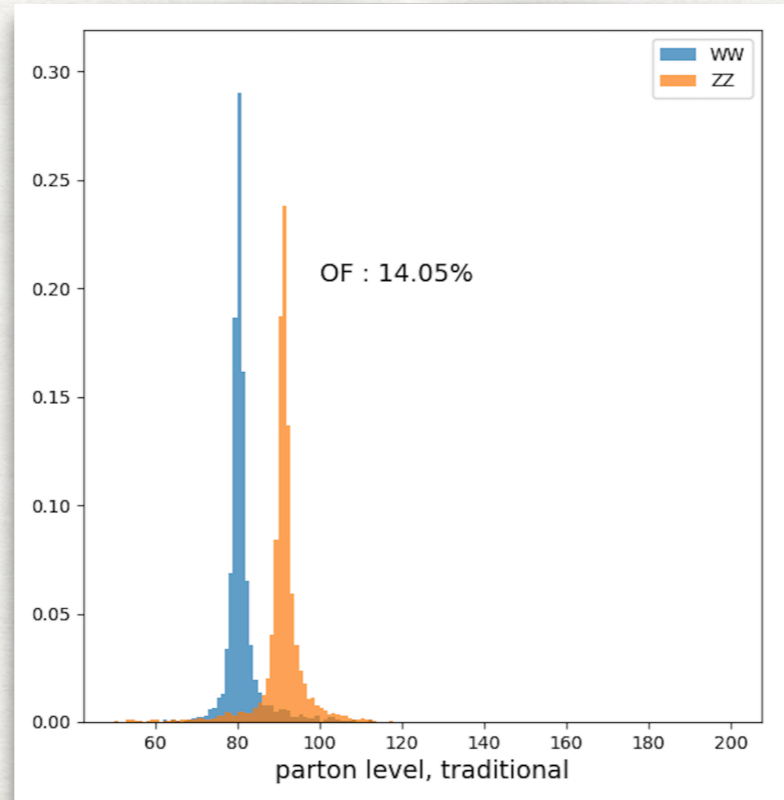
the correctly pairing ratio versus P values :



traditional v.s. QAOA

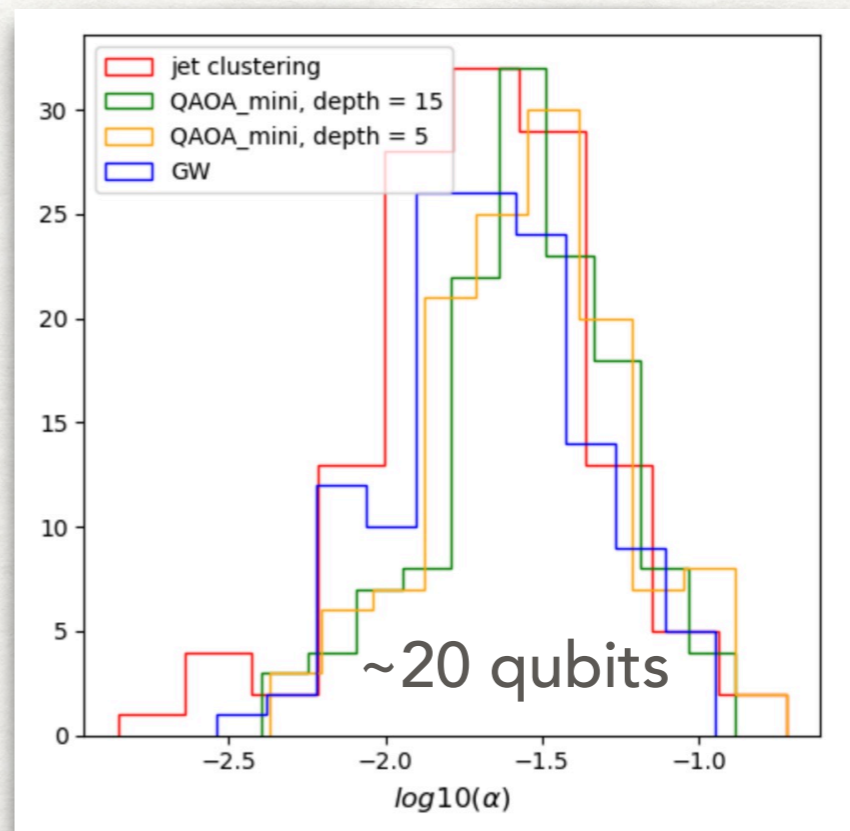
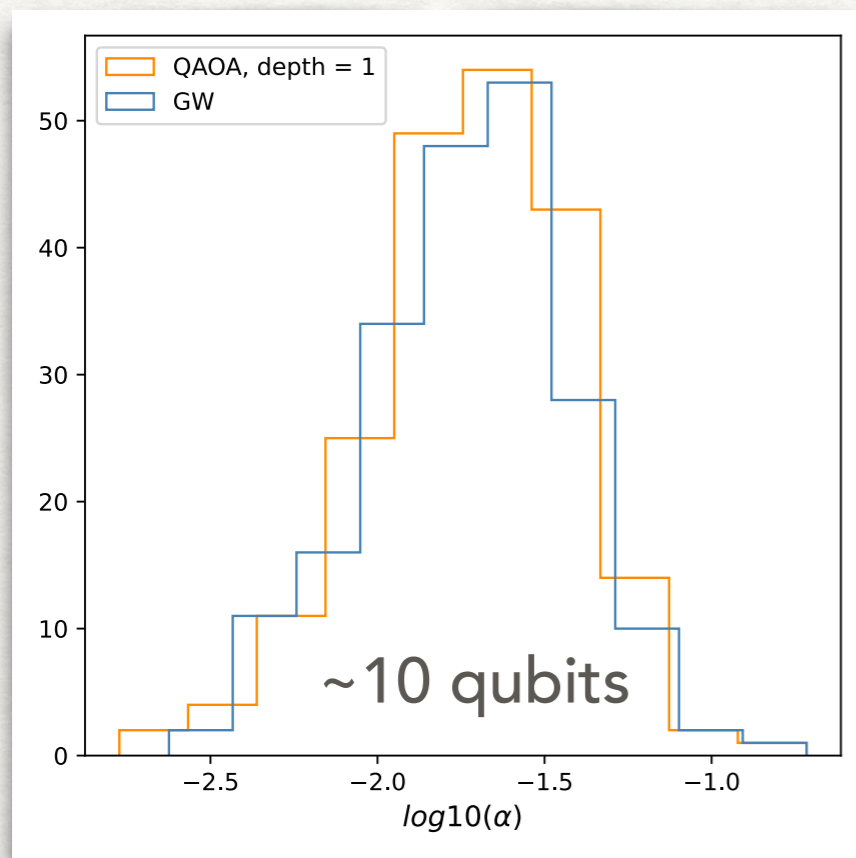
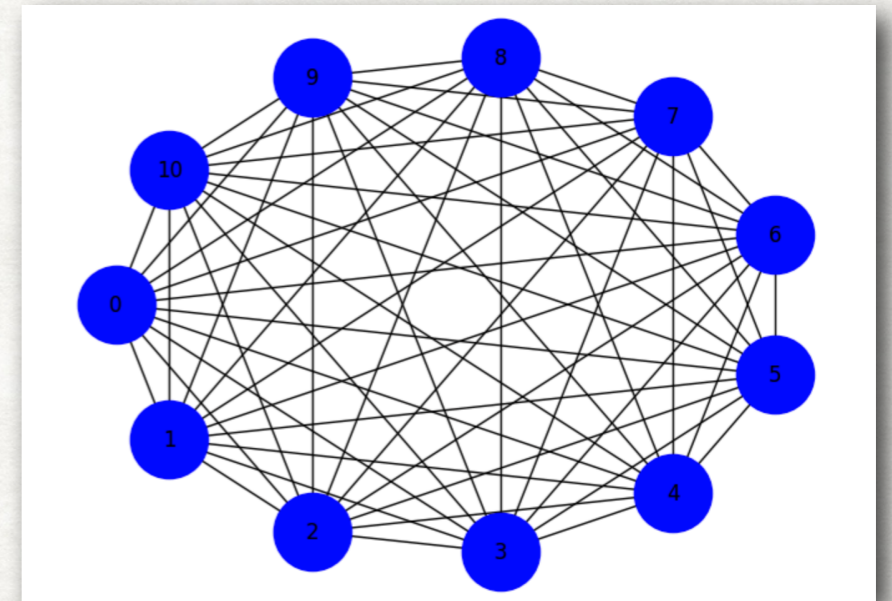
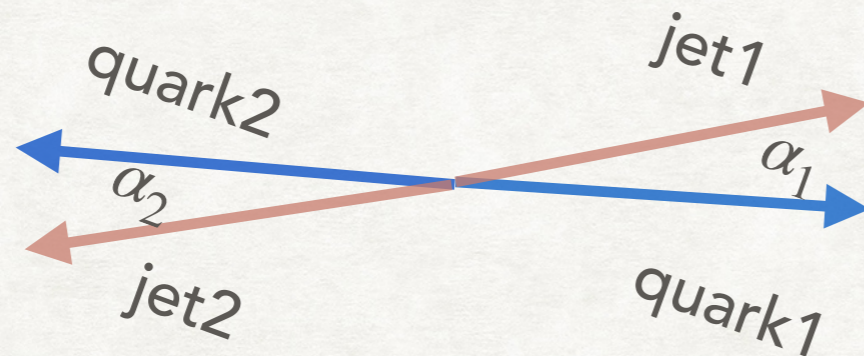
traditional

QAOA with depth =10



jet clustering on $Z \rightarrow u\bar{u}$

$$d_{ij} = 2\min(E_i^2, E_j^2)(1 - \cos\theta_{ij})$$



Summary:

- QAOA can be used to do jet clustering and jet matching.
- With increasing depth of QAOA, the jet matching performance is being better.
- Whether QAOA can surpass traditional approach in jet matching need to be explored further.
- For jet clustering, the performance of QAOA can reach traditional approach on small qubits' events, while it is hard to improve on large qubits' events.

To do:

Sample more times and run on the real hardware.

Thank you !

Backup

Introduction of adiabatic quantum computation and the evolution from AQC to QAOA

AQC : a theoretical framework

components : 1, the driver Hamiltonian (H_D) encodes some quantum state that is easy to prepare its ground state 2, the problem Hamiltonian (H_P) encodes a quantum state we are interested in as its ground state

the idea underline the AQC : we start with a ground state that is easy to prepare and wish to end up with the quantum state we are interested in. This transition is accomplished via the adiabatic theorem, which states that a system in the ground state of some Hamiltonian will remain in the ground state if the Hamiltonian is changed slowly enough.

the process of AQC : 1, define the Hamiltonian: $H(t) = (1 - s(t))H_D + s(t)H_P$ and let our quantum system evolve under it. Unfortunately, time evolution under this time-dependent Hamiltonian involves very messy integral that is hard to evaluate : $U(t) = \tau e^{\frac{-i}{\hbar} \int_0^t H(T)dT}$

2, We discretize $U(T)$ into intervals of Δt small enough that the Hamiltonian is approximately constant over each interval.

3, Let $U(b, a)$ represent time evolution from time a to time b

$$U(T,0) = U(T, T - \Delta t)U(T - \Delta t, T - 2\Delta t) \dots U(\Delta t, 0) = \prod_{j=1}^P U(j\Delta t, (j-1)\Delta t) \approx \prod_{j=1}^P e^{-iH(j\Delta t)\Delta t}$$

since $H(j\Delta t) = (1 - s(j\Delta t))H_D + s(j\Delta t)H_P$ we get $U(T,0) \approx \prod_{j=1}^P e^{-i(1-s(j\Delta t))H_D\Delta t} e^{-is(j\Delta t)H_P\Delta t}$ Thus

we can approximate AQC by repeatedly letting the system evolve under H_P for some $s(j\Delta t)\Delta t$ and then H_D for some small $(1 - s(j\Delta t))\Delta t$

$$H(t) = (1 - s(t))H_D + s(t)H_P$$

$$U(t) = \tau e^{\frac{-i}{\hbar} \int_0^t H(T) dT}$$

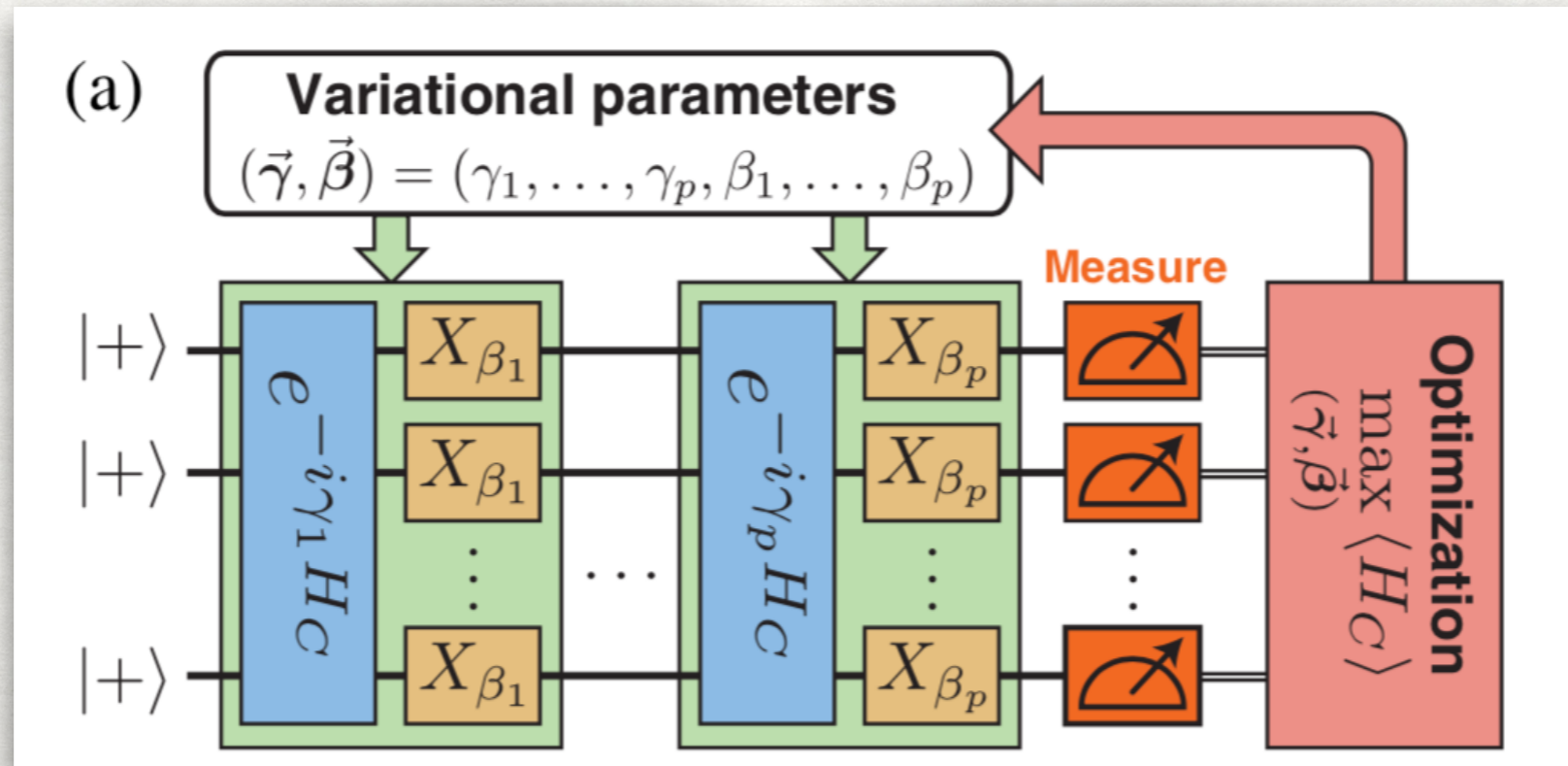
AQC

$$U(T,0) = U(T, T - \Delta t)U(T - \Delta t, T - 2\Delta t) \dots U(\Delta t, 0) = \prod_{j=1}^P U(j\Delta t, (j-1)\Delta t) \approx \prod_{j=1}^P e^{-iH(j\Delta t)\Delta t}$$

$$H(j\Delta t) = (1 - s(j\Delta t))H_D + s(j\Delta t)H_P$$

QAOA

$$U(T,0) \approx \prod_{j=1}^P e^{-i(1-s(j\Delta t))H_D\Delta t} e^{-is(j\Delta t)H_P\Delta t} = \prod_{j=1}^P U((1 - s(j\Delta t))\Delta t, H_D)U(s(j\Delta t)\Delta t, H_P) = \prod_{j=1}^P U(\beta_j, B)U(\gamma_j, C)$$



coding of max-cut in qiskit :

basics : the rotation gate

$$R_x(\theta) = e^{-i\theta X/2}$$

$$R_y(\theta) = e^{-i\theta Y/2}$$

$$R_z(\theta) = e^{-i\theta Z/2}$$

only consider one layer : $P = 1$

$$\sum_{\langle ij \rangle} C_{\langle ij \rangle} | U(B, \beta_1) U(C, \gamma_1) | s \rangle = \sum_{\langle ij \rangle} C_{\langle ij \rangle} | e^{-i\beta_1(\sigma_i^x + \sigma_j^x)} e^{-i\gamma_1 W_{ij}(\sigma_i^z \sigma_j^z)} | s \rangle, \text{ where } | s \rangle = H | 0 \rangle^{\otimes n}$$

prepare initial state $| + \rangle^{\otimes n}$

for i in n_nodes:

qc.h(i)

apply $U(C, \gamma)$ and $U(B, \beta)$ to qubits (we only consider weight on edges):

for edge in edges:

qc.rzz($2 \cdot \gamma_1 \cdot W_{ij}$, q_i , q_j)

for node in nodes:

qc.rx($2 \cdot \beta_1$, q_i)