QML

A Simple Quantum Machine Learning Guide

Wenjuan Gong

UPC
Qingdao, China

October 12, 2022

Section 1: Foundamentals

I: Fundamentals of Theoretical Physics

II: Fundamentals of Experimental Physics

1. Coplanar Waveguide Resonator
2. Circuit Quantum Electrodynamics (Circuit QED)
3. Errors, Error Corrections, Calibrations, and Error Simulation

1. Hilbert space is the mathematical space for quantum problem formulation. (Waves can be formulated as points in Hilbert space.)

Hilbert space

Hilbert spaces (named after David Hilbert) allow generalizing the methods of linear algebra and calculus from (finite-dimensional) Euclidean vector spaces to spaces that may be infinite-dimensional. A Hilbert space is a vector space equipped with an inner product which defines a distance function for which it is a complete metric space. Hilbert spaces arise naturally and frequently in mathematics and physics, typically as function spaces. --------Wikipedia

III: Fundamentals of Mathematics

- The state of a vibrating string can be modeled as a point in a Hilbert space. The decomposition of a vibrating string into its vibrations in distinct overtones is given by the projection of the point onto the coordinate axes in the space. [1]

[1] https://en.wikipedia.org/wiki/Hilbert_space

IV: Algorithms and Quantum Implementations

1. Deutsch Oracle
2. Deutsch Jozsa Algorithm
3. Grover's algorithm

Section 2: Logical Quantum Bits

1. Quantum circuit is an ordered sequence of quantum gates, measurements and resets, which may be conditioned on real-time classical computation.
2. A set of quantum gates is said to be universal if any unitary transformation of the quantum data can be efficiently approximated arbitrarily well as sequence of gates in the set.
3. Any quantum program can be represented by a sequence of quantum circuits and classical near-time computation.

II: Multi-qubit systems with noise

1. Multi-qubit circuit fidelity, and ultimately the path to a fully fault tolerant architecture, is impeded by the tradeoff between crosstalk and gate speed[2]. This tradeoff is implicit in the canonical cQED Hamiltonian for two transmons with fixed coupling(i = 0, 1),

$$H = \sum_{i=0,1} (\omega_i \hat{a}_i^\dagger \hat{a}_i + \frac{\alpha_i}{2} \hat{a}_i^\dagger \hat{a}_i [\hat{a}_i^\dagger \hat{a}_i - 1]) + J(\hat{a}_0^\dagger + \hat{a}_0)(\hat{a}_1^\dagger + \hat{a}_1), \quad (1)$$

with frequecies ω_i, anharmonicities α_i and coupling strength J that can be engineered by a common bus resonator[3] or direct capacitance[4].

[2] A. Kandala, et al. Demonstration of a High-Fidelity CNOT for Fixed-Frequency Transmons with Engineered ZZ Suppression.

[3] L. DiCarlo, et al. Demonstration of two-qubit algorithms with a superconducting quantum processor, Nature 460, 240 (2009).

[4] R. Barends, et al. Superconducting quantum circuits at the surface code threshold for fault tolerance, Nature 508, 500 (2014).

2. The entanglement rate is set by J for a number of two-qubit gates[3], and so, a large J is desirable for fast two-qubit entangling gates. This maximizes gate fidelity given finite qubit coherence. However, in this Hamiltonian, the dressed energy levels have a two-qubit frequency shift (to second order in J)[5]

$$\begin{aligned} ZZ &= \omega_{11} - \omega_{01} - \omega_{10} + \omega_{00}, \\ &= 2J^2 \frac{\alpha_0 + \alpha_1}{(\Delta + \alpha_0)(\Delta - \alpha_1)}, \end{aligned} \tag{2}$$

where Δ is the qubit-qubit detuning. For fixed couplings, this interaction is an always-on source of error and is referred to as the static ZZ. It limits its multi-qubit circuit performance[6,7,8,9,10], and is an impediment for realizing quantum error detection.

[5] E. Magesan and J. M. Gambetta. Effective hamiltonian models of the cross-resonance gate, Phys. Rev. A 101, 052308 (2020).

[6] N. Sundaresan, et al. Reducing unitary and spectator errors in cross resonance with optimized rotary echoes, arXiv preprint arXiv:2007.02925 (2020).

[7] P. Jurcevic, et al. Demonstration of quantum volume 64 on a superconducting quantum computing system, arXiv preprint arXiv:2008.08571 (2020).

[8] D. C. McKay, et al. Three-qubit randomized benchmarking, Phys. Rev. Lett. 122, 200502 (2019).

[9] S. Krinner, et al. Benchmarking coherent errors in controlled-phase gates due to spectator qubits, Phys. Rev. Applied 14, 024042 (2020).

[10] K. X. Wei, et al. Verifying multipartite entangled greenberger-horne-zeilinger states via multiple quantum coherences, Phys. Rev. A 101, 032343 (2020).

I: Multi-qubit systems with noise

3. More recent approaches have directly focused on suppressing the static *ZZ* interaction by engineering the two-qubit level spacings. As seen from Eqn. (2), this can be achieved by coupling qubits with opposite signs of anharmonicity[11,12].
4. This effect can also be achieved by employing multiple coupling paths[13,14,15,16,17]

[11] P. Zhao, et al. High-contrast zz in- teraction using multi-type superconducting qubits, arXiv preprint arXiv:2002.07560 (2020).

[12] J. Ku, et al. Suppression of unwanted zz interactions in a hybrid two-qubit system, arXiv preprint arXiv:2003.02775 (2020).

[13] P. Mundada, et al. Suppression of qubit crosstalk in a tunable coupling superconducting circuit, Physical Review Applied 12, 054023 (2019).

[14] F. Yan, et al. Tunable coupling scheme for implementing high-fidelity two- qubit gates, Physical Review Applied 10, 054062 (2018).

[15] M. C. Collodo, et al. Advantages of versatile neural-network decoding for topological codes, (2020), arXiv:2005.08863.

[16] Y. Xu, et al. High- fidelity, high-scalability two-qubit gate scheme for superconducting qubits, (2020), arXiv:2006.11860.

[17] Y. Sung, et al. Realization of high-fidelity cz and zz-free iswap gates with a tunable coupler, (2020), arXiv:2011.01261.

5. The effective Hamiltonian can be reduced to a form[18] identical to that of the CPB (Copper pair box)[19] system:

$$\hat{H} = 4E_C(\hat{n} - n_g)^2 - E_J \cos\hat{\varphi}, \qquad (3)$$

which describes the effective circuit below without the coupling to the transmission line (L_r and C_r), $\hat{n}$ and $\hat{\varphi}$ denote the number of Cooper pairs transferred between the islands and the gauge-invariant phase difference between the superconductors, respectively.

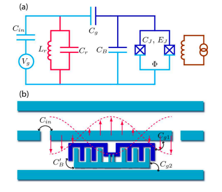


FIG. 1: (Color online) (a) Effective circuit diagram of the transmon qubit. The two Josephson junctions (with capacitance and Josephson energy C_J and E_J) are shunted by an additional large capacitance C_B, matched by a comparably large gate capacitance C_g. (b) Simplified schematic of the transmon device design (not to scale), which consists of a traditional split Cooper-pair box, shunted by a short ($L \sim \lambda/20$) section of twin-lead transmission line, formed by extending the superconducting islands of the qubit. This short section of line can be well approximated as a lumped-element capacitor, leading to the increase in the capacitances C_{g1}, C_{g2} and C'_B and hence in the effective capacitances C_B and C_g in the circuit.

[18] J. Koch, et al. Charge insensitive qubit design derived from the Cooper pair box.

[19] In conclusion, the transom is a CPB operated in $E_J/E_C \gg 1$ regime with charge fluctions of the the order of unity

6. The use of Thévenin's theorem[20] allows for the reduction of the capacitance network to a few effective capacitances, like in the previous figure. Here, the effect of the resonator can be modeled by a local LC oscillator. Following the standard quantization procedure for circuits, we obtain:

$$\begin{aligned} \hat{H} &= \frac{\hat{\phi}_r^2}{2L_r} + \frac{(C_B + C_g)\hat{Q}_r^2}{2C_*} \\ &+ \frac{(C_g + C_{in} + C_r)\hat{Q}_J^2}{2C_*^2} - E_J \cos\left(\frac{2\pi}{\hbar}\hat{\phi}_J\right) \\ &+ \frac{C_g \hat{Q}_r \hat{Q}_J}{C_*^2} + \frac{(C_B C_{in} + C_g C_{in})\hat{Q}_r V_g + C_g C_{in} \hat{Q}_J V_g}{C_*^2}, \end{aligned} \tag{4}$$

where C_B is a shunting capacitance (并联电容器), C_g is a gate capacitance, V_g is the original gate voltage, Q_i and ϕ_j are the charge and potential associated with a conducting island i. In Eq. 4, the first two terms describes the local oscillator of the resonator, the two terms in the second line capture the qubit's degrees of freedom and the terms in line 3 give the coupling between the two of them and the coupling to the gate electrode. Taking into account that $\hat{V} = V_{rms}^0(\hat{a} + \hat{a}^\dagger)$ and assuming that $C_r \gg C_B, C_{in}, C_g$, we recover the Hamiltonian:

[20] By Thévenin's theorem, any single-port linear network of impedances and voltage sources can be substituted by an equivalent circuit consisting of one voltage source V' and one impedance.

I: Multi-qubit systems with noise

$$\hat{H} = 4E_C(\hat{n} - n_g)^2 - E_J \cos\hat{\varphi} + \hbar\omega_r \hat{a}^\dagger \hat{a} + 2\beta e V^0_{rms} \hat{n}(\hat{a} + \hat{a}^\dagger), \quad (5)$$

where $\omega_r = 1/\sqrt{L_r C_r}$ denotes the resonator frequency, $\hat{a}(\hat{a}^\dagger)$ annihilates (creates) one photon in the transmission line. The root mean square voltage of the local oscillator is denoted by $V^0_{rms} = \sqrt{\hbar\omega_r/2C_r}$. The parameter β is defined as the ratio of the gate capacitance and the total capacitance, $\beta = C_g/C_{\sum}$. And $C_R \gg C_{\sum}$.

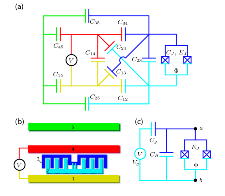


FIG. 12: (Color online) (a) Full capacitance network for the transmon device. (b) Simplified schematic of the transmon device design (not to scale). (c) Reduced network.

7. In close analogy to the situation of the CPB, embedding the transmon in a superconducting transmission line resonator opens up the possibility of control and readout of the qubit state-a scenario that has been termed circuit QED. The quantizaiton of the circuit is described by Eq. 5.

Section 3: Physical Quantum Bits

1.

II: Gate Errors

1.

III: Error Mitigation–Qiskit Ignis

1. complete_meas_cal has the following functions.
   1.1 Every quantum qubit is applied a X-gate, so that the qubit is flipped;
   1.2 The new quantum circuit is measured and stored in cal_matrix variable of the CompleteMeasFitter class, e.g.,

```
qr=qiskit.QuantumRegister(5)
qubit_list=[2,3,4]
meas_calibs,state_labels=complete_meas_cal(qubit_list,qr=qr,circlabel='mcal')
#Qiskit/qiskit-ignis/qiskit/ignis/mitigation/measurement/circuits.py-L25
```

2. CompleteMeasFitter counts the occurrences of all measurements, e.g.,

```
meas_fitter=CompleteMeasFitter(cal_results,state_labels,circlabel='mcal')
#Qiskit/qiskit-ignis/qiskit/ignis/mitigation/measurement/fitters.py-Line38
```

3. Apply filter:

```
raw_counts=results.get_counts()
meas_filter=meas_fitter.filter
mitigated_results=meas_filter.apply(results)
#Qiskit/qiskit-ignis/qiskit/ignis/mitigation/measurement/fitters.py-Line101
#Qiskit/qiskit-ignis/qiskit/ignis/mitigation/measurement/filters.py/-Line78
```

- The codes that actually carry out filtering using least squared optimization method:

```
#filters.py-Line78
# Apply the correction
for data_idx, _ in enumerate(raw_data2):

if method == 'pseudo_inverse':
  raw_data2[data_idx] = np.dot(pinv_cal_mat, raw_data2[data_idx])

elif method == 'least_squares':
  nshots = sum(raw_data2[data_idx])

  def fun(x):
    return sum((raw_data2[data_idx] - np.dot(self._cal_matrix, x))**2)
  x0 = np.random.rand(len(self._state_labels))
  x0 = x0 / sum(x0)
  cons = ({'type': 'eq', 'fun': lambda x: nshots - sum(x)})
  bnds = tuple((0, nshots) for x in x0)
  res = minimize(fun, x0, method='SLSQP', constraints=cons, bounds=bnds, tol=1e-6)
  raw_data2[data_idx] = res.x

else:
  raise QiskitError("Unrecognized method.")
```

IV: Random Benchmarking–Qiskit Ignis[23]

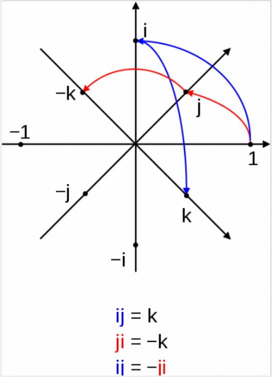
1. Randomized benchmarking[21] has shown that long sequences of quantum gates sampled uniformly at random from the Haar measure on the group SU(d) would lead to an exponential decay at a rate that was uniquely fixed by the error model.
2. Qiskit textbook explaination about random benchmarking[22].
3. The special unitary group of degree n, denoted SU(n), is the Lie group of $n \times n$ unitary matrices with determinant 1; The orthogonal group in dimension n, denoted O(n), is the group of distance-preserving transformations of a Euclidean space of dimension n that preserve a fixed point, where the group operation is given by composing transformations. Unit quaternions provide a convenient mathematical notation for representing spatial orientations and rotations of elements in three dimensional space.

[21] Emerson, Joseph; Alicki, Robert; Zyczkowski, Karol (2005). "Scalable noise estimation with random unitary operators". Journal of Optics B: Quantum and Semiclassical Optics. 7 (10): S347.

[22] https://qiskit.org/textbook/ch-quantum-hardware/randomized-benchmarking.html

[23] Note that this only works in online environment

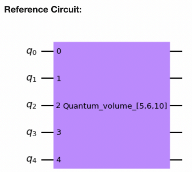
3. SO(3) = Quaternion

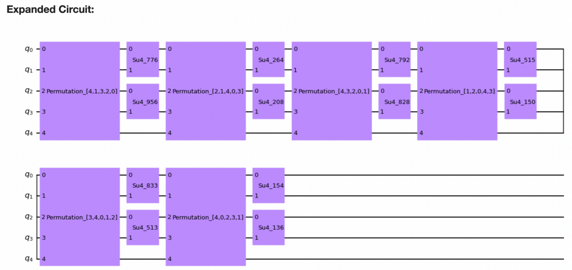


Graphical representation of products of quaternion units as 90° rotations in the planes of 4-dimensional space spanned by two of $\{1, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$. The left factor can be viewed as being rotated by the right factor to arrive at the product. Visually $\mathbf{i} \cdot \mathbf{j} = -(\mathbf{j} \cdot \mathbf{i})$.

- In *blue*:
  - $1 \cdot \mathbf{i} = \mathbf{i}$ (1/**i** plane)
  - $\mathbf{i} \cdot \mathbf{j} = \mathbf{k}$ (**i**/**k** plane)
- In *red*:
  - $1 \cdot \mathbf{j} = \mathbf{j}$ (1/**j** plane)
  - $\mathbf{j} \cdot \mathbf{i} = -\mathbf{k}$ (**j**/**k** plane)

1. A quantum volume model circuit.
2. The model circuits are random instances of circuits used to measure the Quantum Volume metric, as introduced in[24].
3. The model circuits consist of layers of Haar random elements of SU(4) applied between corresponding pairs of qubits in a random bipartition.
4. The procedure[25].





[24] A. Cross et al. Validating quantum computers using randomized model circuits, Phys. Rev. A 100, 032328 (2019).

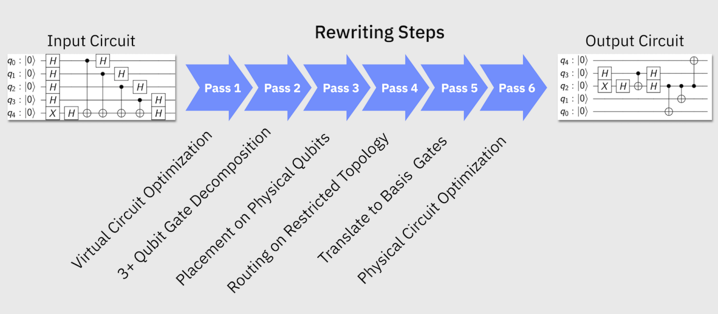
[25] https://qiskit.org/textbook/ch-quantum-hardware/measuring-quantum-volume.html

VI: SYNDROME MEASUREMENT

1. The syndrome measurement provides information about the error that has happened, but not about the information that is stored in the logical qubit—as otherwise the measurement would destroy any quantum superposition of this logical qubit with other qubits in the quantum computer, which would prevent it from being used to convey quantum information.

Section 4: Logical <-> Physical

1. Transpiler(编译器)Transpilation is the process of rewriting a given input circuit to match the topology of a specific quantum device, and/or to optimize the circuit for execution on present day noisy quantum systems.



2. DAGCircuit(有向图量子电路)

Section 5: Application Related

I: QOTP: Quantum One-time Pad

- QOTP (or random compilation)
- Accreditation Protocol–Trap circuit(classically simulable circuits?)

Section 6: Others

1. Hamiltonian parameters–Qiskit tutorial[26]
2. Gate Errors–Qiskit tutorial

[26] dressed basis 缀饰基. The term light dressed state refers to a quantum state of an atomic or molecular system interacting with a laser light. Dispersive shift 色散移位: as you move the qubit, both resonator and qubit frequency move.

II: Others

1. Qiskit Providers:

```
import qiskit
simulator =qiskit.Aer.get_backend('qasm_simulator')

# https://github.com/Qiskit/qiskit-terra/blob/main/qiskit/providers/providerutils.py----L20
def filter_backends(backends, filters=None, **kwargs):
"""Return the backends matching the specified filtering.
```

2. Qiskit Execute Experiments: First, transpile the circuit; Second, run on backend.

```
import qiskit
from qiskit import execute
simulator =qiskit.Aer.get_backend('qasm_simulator')
job=execute(circuit_list,simulator,shots=1)

# https://github.com/Qiskit/qiskit-terra/blob/main/qiskit/execute_function.py----L38
def execute(experiments,backend,basis_gates=None,coupling_map=None, # circuit transpile options
            backend_properties=None,initial_layout=None,seed_transpiler=None,optimization_level=None,
            pass_manager=None,qobj_id=None,qobj_header=None,shots=None, # common run options
            memory=None,max_credits=None,seed_simulator=None,default_qubit_los=None,default_meas_los=None,
            # schedule run options
            qubit_lo_range=None,meas_lo_range=None,schedule_los=None,meas_level=None,meas_return=None,
            memory_slots=None,memory_slot_size=None,rep_time=None,rep_delay=None,parameter_binds=None,
            schedule_circuit=False,inst_map=None,meas_map=None,scheduling_method=None,init_qubits=None,
            **run_config):
"""Execute a list of :class:`qiskit.circuit.QuantumCircuit` or :class:`qiskit.pulse.Schedule` on a
    backend.
```

2. Qiskit Execute Experiments: Run on backend.

```
#qiskit-aer/qiskit/providers/aer/backends/aerbackend.py----L123
#https://github.com/Qiskit/qiskit-aer/blob/6ca7aac12bb0acf1071217aa8d4a1b70c0ba208b/qiskit/providers/aer/
      backends/aerbackend.py----L123
def run(self,circuits,validate=False,parameter_binds=None,**run_options):
"""Run a qobj on the backend."""

qobj = self._assemble(circuits, parameter_binds=parameter_binds, **run_options)

aer_job = AerJob(self, job_id, self._run, experiments, executor)
aer_job.submit()

# L286
def _run(self, qobj, job_id='', format_result=True):
"""Run a job"""

output = self._execute(qobj) #nothing in _execute

#qiskit.providers.aer.backends.qasm_simulator
def _execute(self, qobj):
"""Execute a qobj on the backend"""
return cpp_execute(self._controller, qobj)

#https://github.com/Qiskit/qiskit-aer/blob/main/qiskit/providers/aer/backends/backend_utils.py----109
def cpp_execute(controller, qobj):
"""Execute qobj on C++ controller wrapper"""
qobj.config.library_dir = LIBRARY_DIR
return controller(qobj)
```

Thank you!

Appendices

```
Examples::

    lhs.compose(rhs, qubits=[3, 2], inplace=True)

    .. parsed-literal::
```