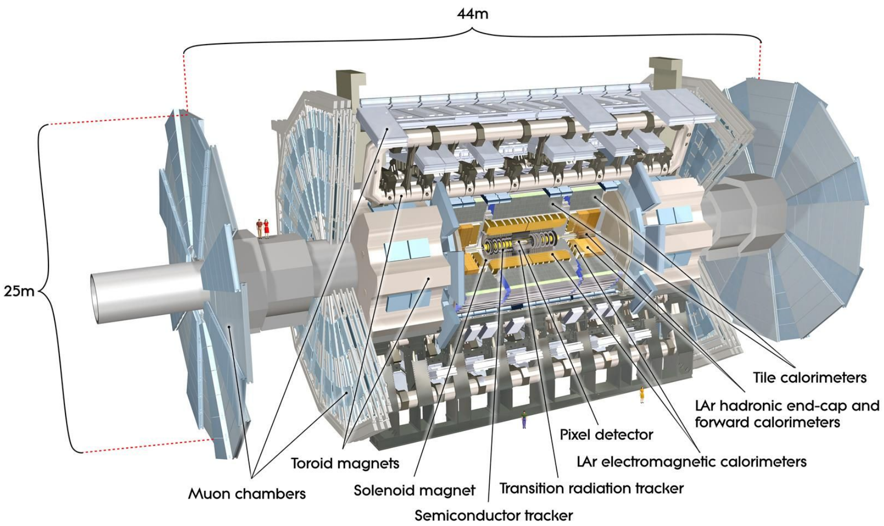
Quantum Hough Transform

for Charged-Particle Track Reconstruction

Annual LBNL ATLAS Meeting
01-07-2020

Amitabh Yadav
(amitabhyadav@lbl.gov)
Heather Gray

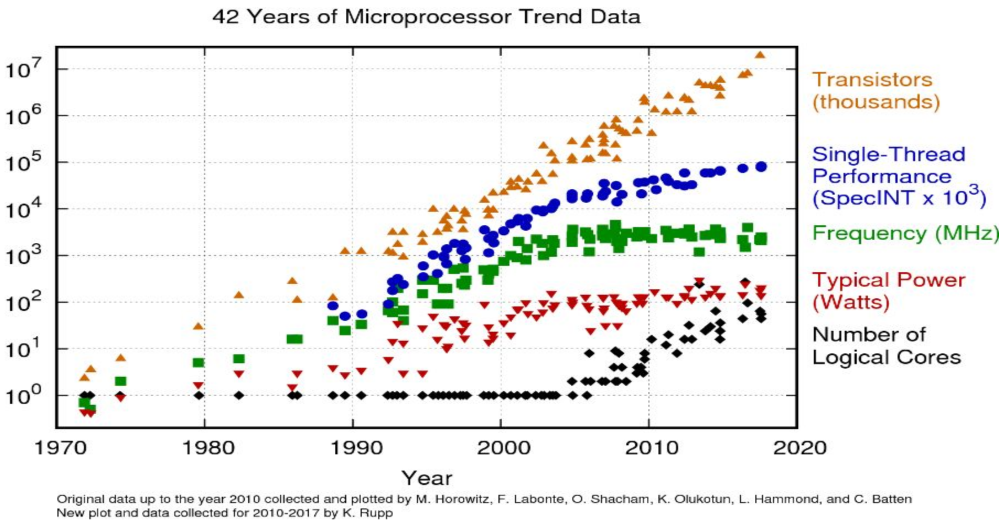
HEP Quantum Pattern Recognition - https://hep-qpr.lbl.gov/



BERKELEY LAB
Lawrence Berkeley National Laboratory

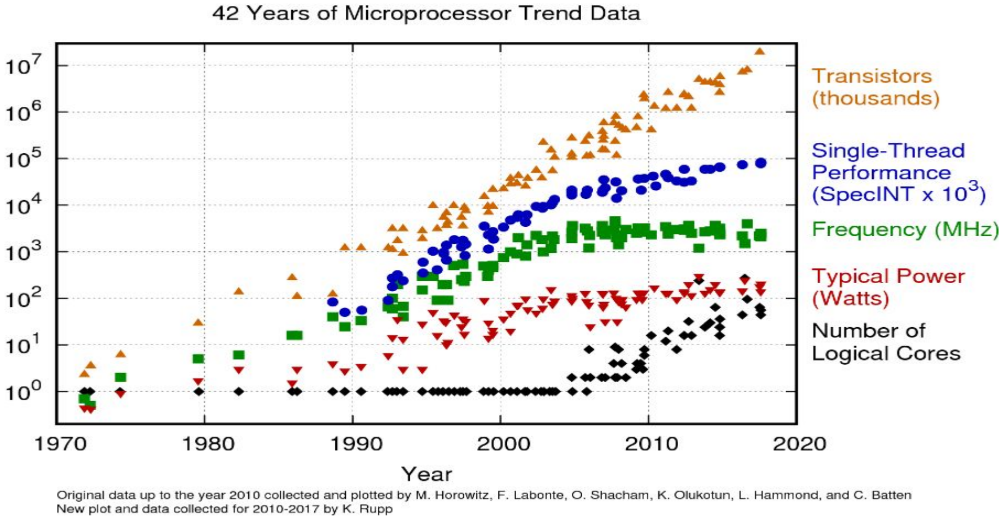
ATLAS EXPERIMENT

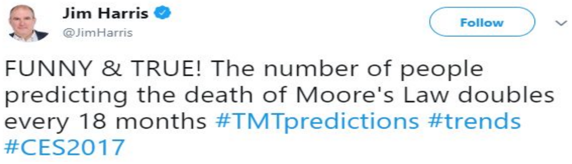
The Perspective



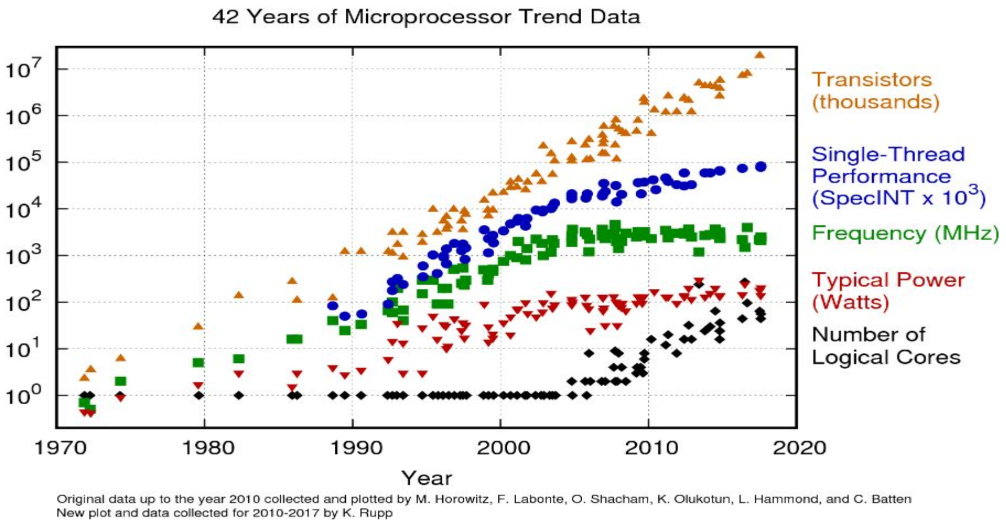
+

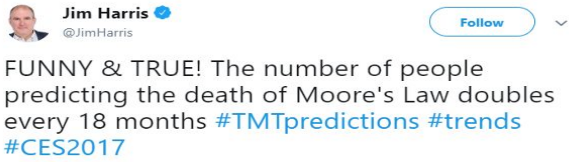
The Perspective





The Perspective





Host CPU

Graphics Processing Unit

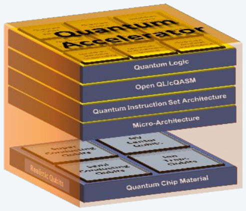
Field-Programmable Gate Array

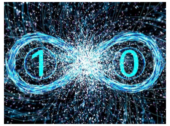
Digital Signal Processor

Neural Processing Unit

Quantum Accelerator

map problem to quantum:
do:
run Q Algorithm



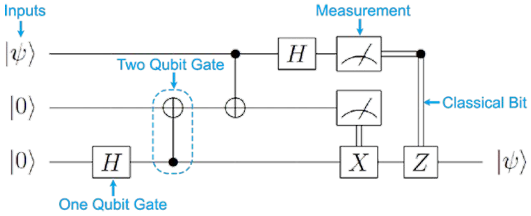


\+

- assess answer
- save measurement result/statistics
- interpret classical answer

The Idea

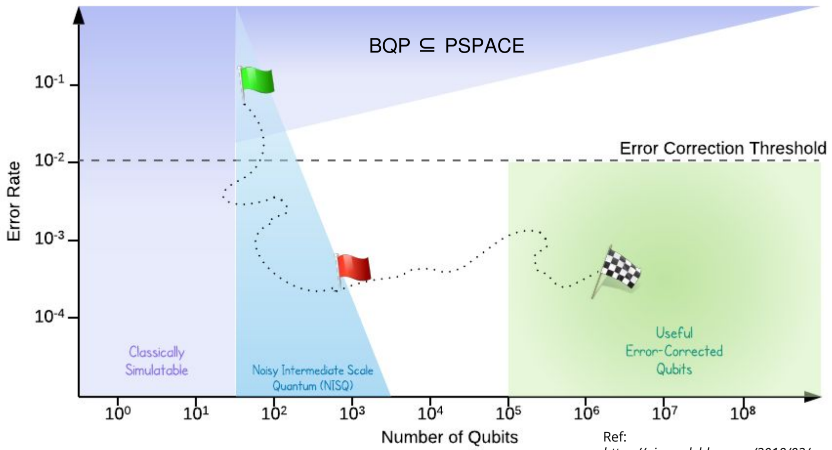
- **Circuit Model QC**
  - Solovay-Kitaev Theorem
  - Universal Quantum Computing



The Idea

- **Circuit Model QC**
  - Solovay-Kitaev Theorem
  - Universal Quantum Computing

- **Quantum Computing Applications Approach**
  - Perfect Qubits
  - Erroneous Qubits in NISQ-era
    - Test algorithmic bounds of error-tolerance,
    - Pauli Frames,
    - Temporary Quantum Registers and SWAPS,
    - Small (distance) EC Codes etc.



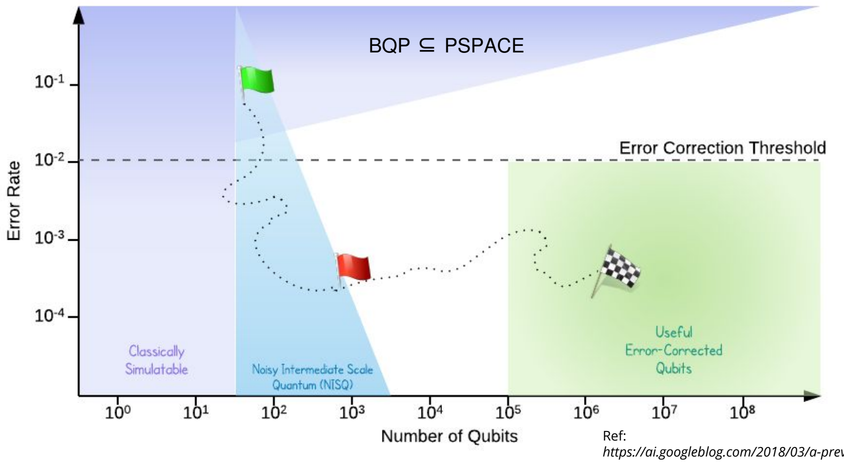
Ref: *https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html*

The Idea

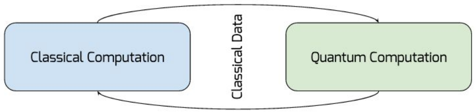
- **Circuit Model QC**
  - Solovay-Kitaev Theorem
  - Universal Quantum Computing

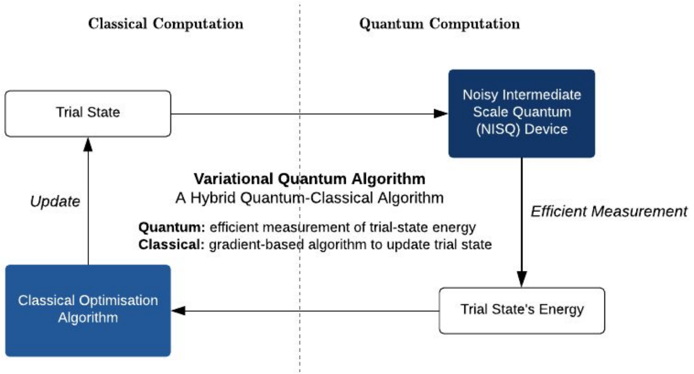
- **Quantum Computing Applications Approach**
  - Perfect Qubits
  - Erroneous Qubits in NISQ-era
    - Test algorithmic bounds of error-tolerance,
    - Pauli Frames,
    - Temporary Quantum Registers and SWAPS,
    - Small (distance) EC Codes etc.

- **QPU vs Simulators platform**
  - Quantum accelerator model?
    - Kind of but Not really.
  - In-memory computing model.



Ref: *https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html*



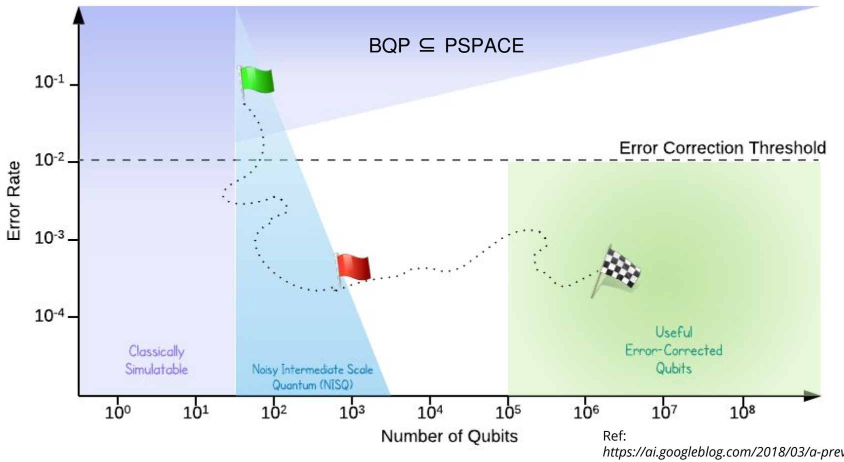


The Idea

- **Circuit Model QC**
  - Solovay-Kitaev Theorem
  - Universal Quantum Computing

- **Quantum Computing Applications Approach**
  - Perfect Qubits
  - Erroneous Qubits in NISQ-era
    - Test algorithmic bounds of error-tolerance,
    - Pauli Frames,
    - Temporary Quantum Registers and SWAPS,
    - Small (distance) EC Codes etc.

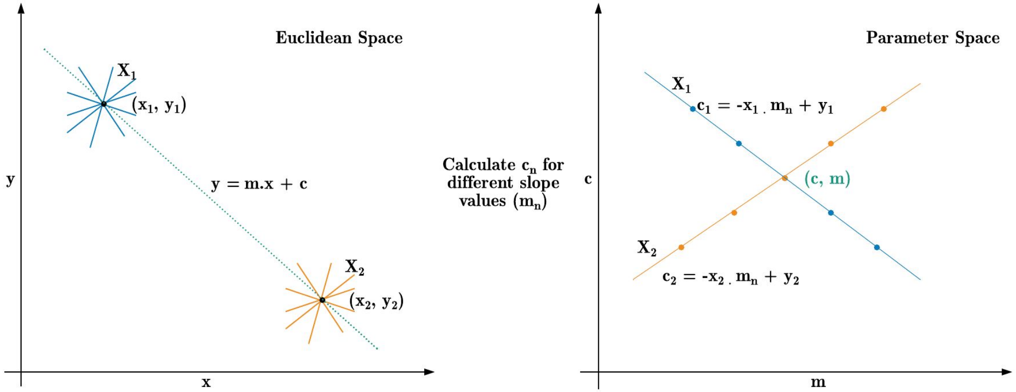
- **QPU vs Simulators platform**
  - Quantum accelerator model?
    - Kind of but Not really.
  - In-memory computing model.
  - Good QC Packages out there for modern QC methods:
    (QISKIT, Forest, D-Wave Ocean, Xanadu PennyLane, Entropica, QuEST, quantumSim, Project-Q, QX… to name a few)



Ref: *https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html*

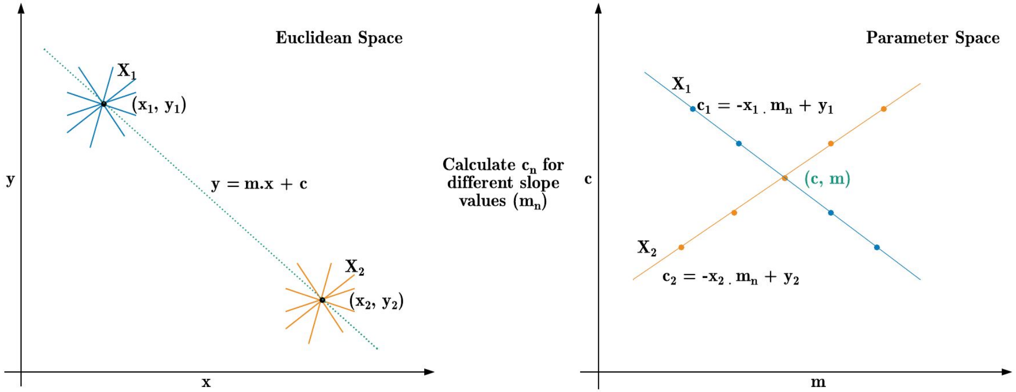
Hough Transform

- Feature Extraction Technique commonly used in Image Processing/Computer Vision.
- Identification of lines and other shapes such as circles or ellipses.
- Used for HEP Tracking Application.
- A line is represented as a point (b, m) in parameter space and, (r, θ) in Hough Space.

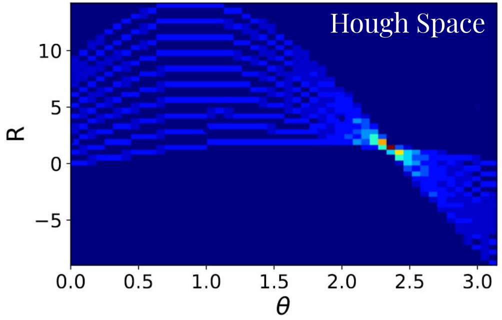


Hough Transform

- Feature Extraction Technique commonly used in Image Processing/Computer Vision.
- Identification of lines and other shapes such as circles or ellipses.
- Used for HEP Tracking Application.
- A line is represented as a point (b, m) in parameter space and, (r, θ) in Hough Space.



$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$



- In the Hough Space, we calculate r for each θ.
- All the points on the line are represented by a local maximum.

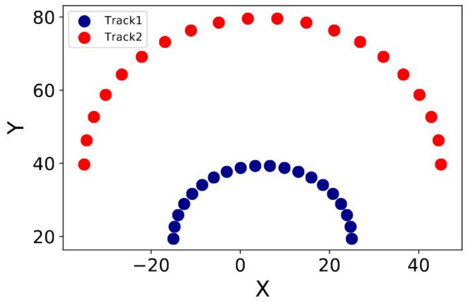
Hough Transform

for Track Reconstruction

TrackML Dataset

```
/* filetype: hits  */ hit_id,x,y,z,volume_id,layer_id,module_id
/* filetype: truth */ hit_id,particle_id,tx,ty,tz,tpx,tpy,tpz,weight
/* filetype: cells */ hit_id,ch0,ch1,value
/* filetype: particles */ particle_id,vx,vy,vz,px,py,pz,q,nhits
```

$$(x-a)^2 + (y-b)^2 = R^2$$



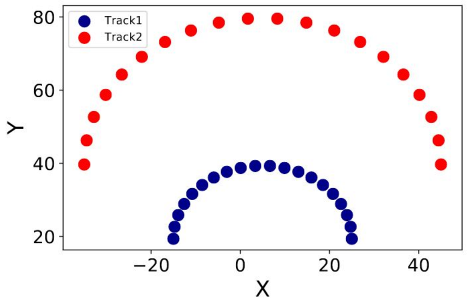
Ref.: *Alipour Tehrani, Niloufar "Simulation and tracking studies for a drift chamber at the FCC-ee experiment" (2019). cds.cern.ch*

Hough Transform
for Track Reconstruction

TrackML Dataset

```
/* filetype: hits  */ hit_id,x,y,z,volume_id,layer_id,module_id
/* filetype: truth */ hit_id,particle_id,tx,ty,tz,tpx,tpy,tpz,weight
/* filetype: cells */ hit_id,ch0,ch1,value
/* filetype: particles */ particle_id,vx,vy,vz,px,py,pz,q,nhits
```

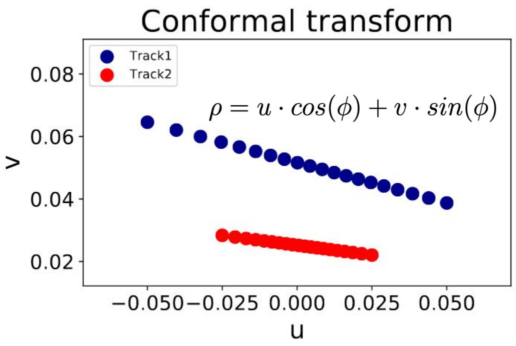
$$(x-a)^2 + (y-b)^2 = R^2$$



The conformal mapping maps a circle to a line if and only if the circle passes from the origin or following the condition:

$$a^2 + b^2 = R^2$$

$$v = \frac{1}{2b} - u\frac{a}{b}$$



Ref.: *Alipour Tehrani, Niloufar "Simulation and tracking studies for a drift chamber at the FCC-ee experiment" (2019). cds.cern.ch*

Hough Transform
for Track Reconstruction

TrackML Dataset

```
/* filetype: hits  */ hit_id,x,y,z,volume_id,layer_id,module_id
/* filetype: truth */ hit_id,particle_id,tx,ty,tz,tpx,tpy,tpz,weight
/* filetype: cells */ hit_id,ch0,ch1,value
/* filetype: particles */ particle_id,vx,vy,vz,px,py,pz,q,nhits
```

The conformal mapping maps a circle to a line if and only if the circle passes from the origin or following the condition:

$$a^2 + b^2 = R^2$$

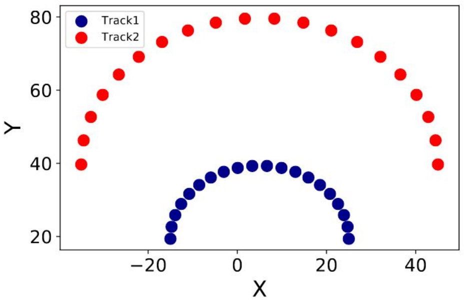
$$v = \frac{1}{2b} - u\frac{a}{b}$$

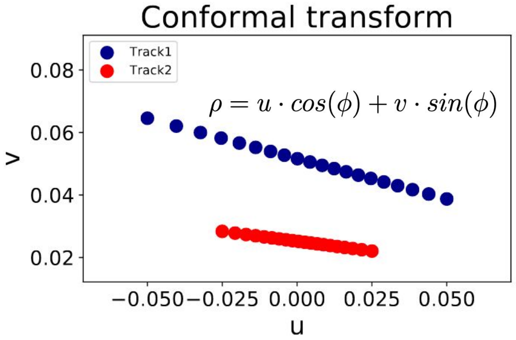
The center of the circle is extracted from the Hough transformation using

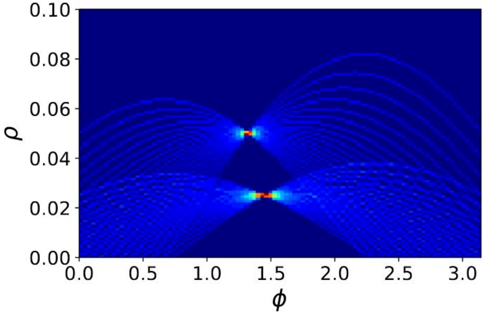
$$a = \frac{cos(\phi)}{2 \cdot \rho}, \quad b = \frac{sin(\phi)}{2 \cdot \rho}$$

Pattern Recognition is performed by finding the local maxima in the Hough space.

$$(x-a)^2 + (y-b)^2 = R^2$$

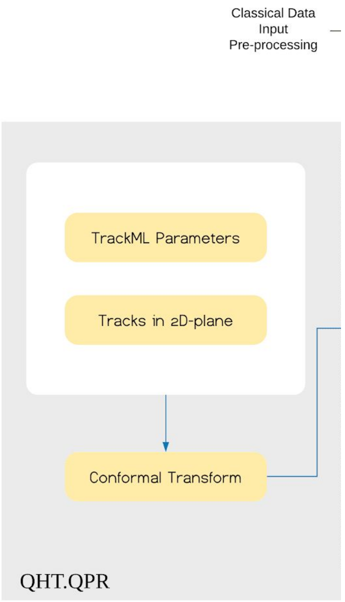




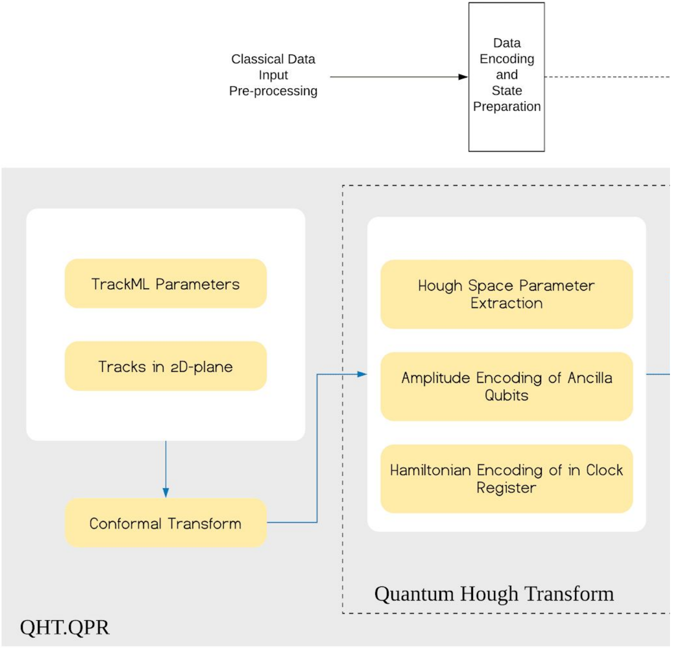


Ref.: *Alipour Tehrani, Niloufar "Simulation and tracking studies for a drift chamber at the FCC-ee experiment" (2019). cds.cern.ch*

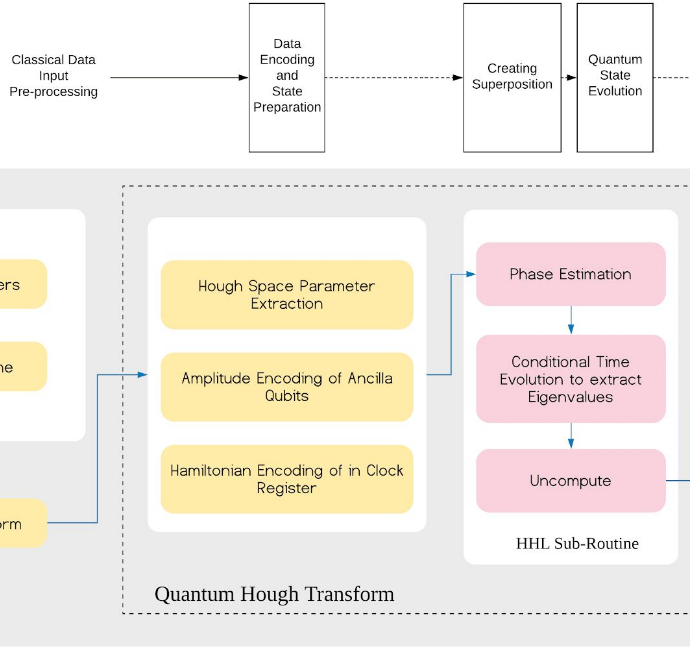
Quantum Hough Transform



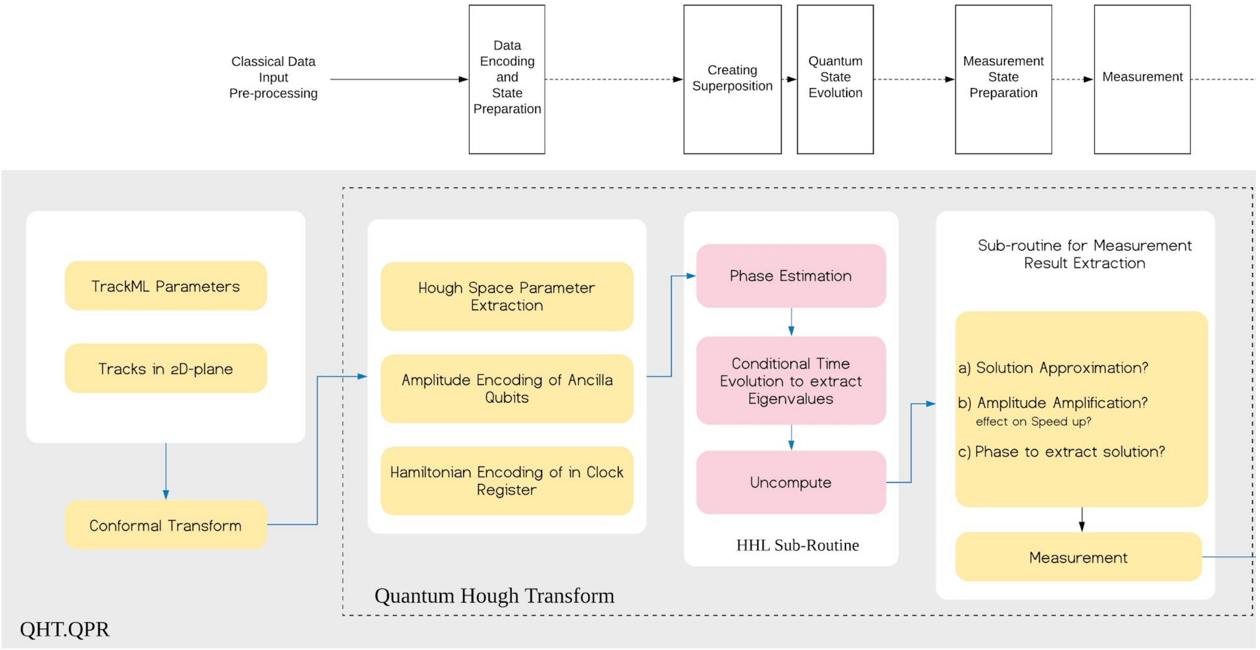
Quantum Hough Transform



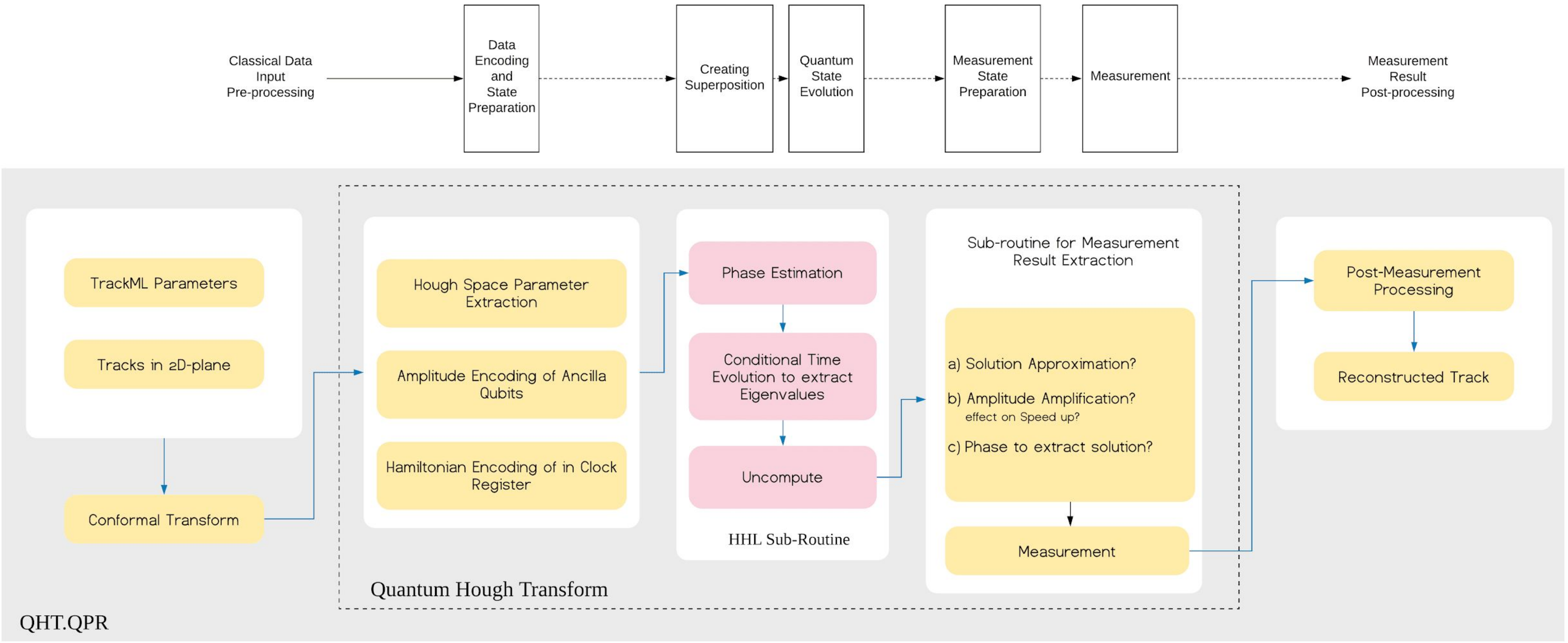
Quantum Hough Transform



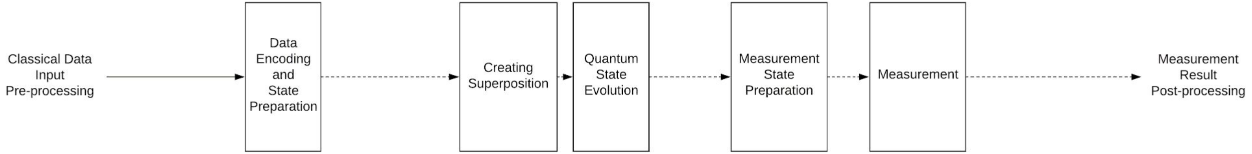
Quantum Hough Transform

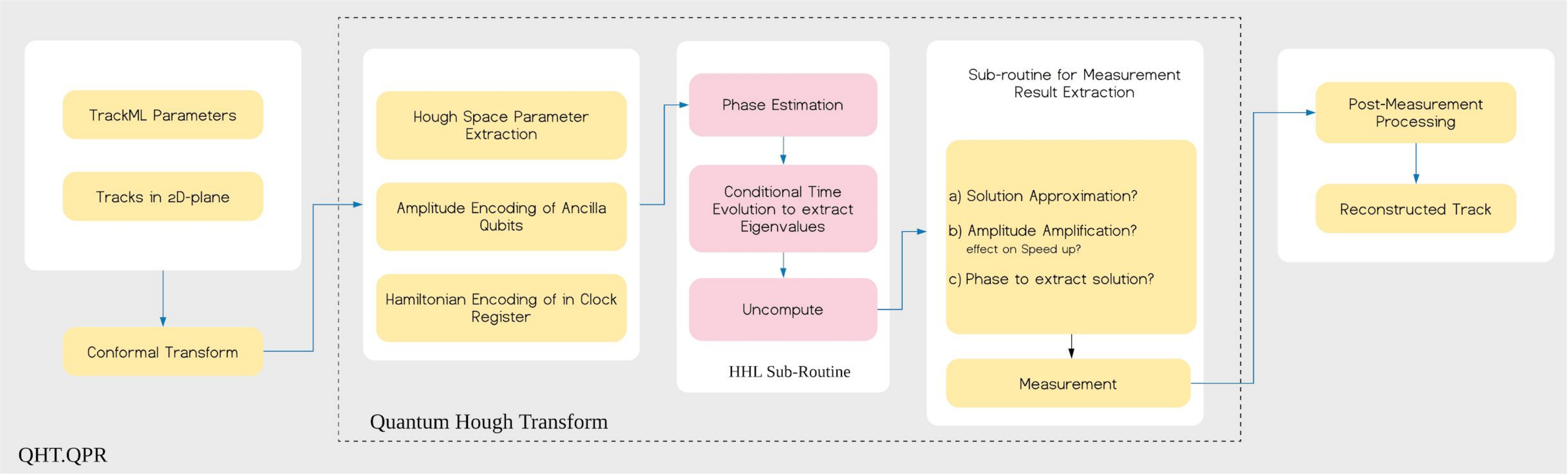


Quantum Hough Transform



Quantum Hough Transform





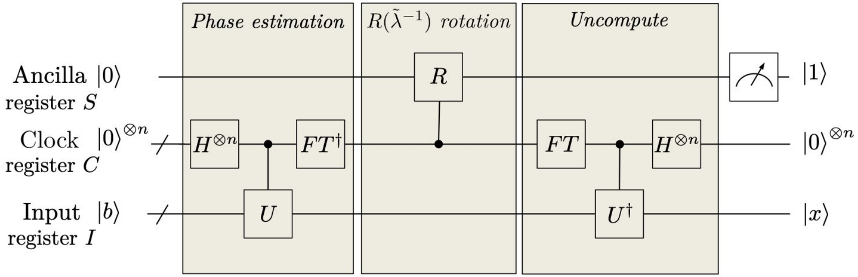
Key Challenges:

- HHL Algorithm Scaling
- Hamiltonian Encoding
- Measurement Result Interpretation

Problem	Algorithm	Runtime Complexity
LSP	CG [She94]	$\mathcal{O}\left(Ns\kappa\log(1/\epsilon)\right)$
QLSP	HHL [HHL09]	$\mathcal{O}\left(\log(N)s^2\kappa^2/\epsilon\right)$
QLSP	VTAA-HHL [Amb10]	$\mathcal{O}\left(\log(N)s^2\kappa/\epsilon\right)$
QLSP	Childs *et. al.* [CKS17]	$\mathcal{O}\left(s\kappa\,\mathrm{polylog}(s\kappa/\epsilon)\right)$
QLSP	QLSA [WZP18]	$\mathcal{O}\left(\kappa^2\,\mathrm{polylog}(n)\Vert A\Vert_F/\epsilon\right)$

Constructing QHT/QLSA Sub-Routine

https://github.com/amitabhyadav/QHT.QPR



What?

Given $A\vec{x} = \vec{b}$ for a Matrix A and vector $\vec{b}$, solve for $\vec{x}$.

Conditions:

A is Hermitian matrix i.e. $A = A^\dagger$. A must be sparse and low-rank.

We have to represent $\vec{x}$ and $\vec{b}$ as quantum states, $|x\rangle$ and $|b\rangle$. So, we normalize them, i.e. $|x| = |b| = 1$

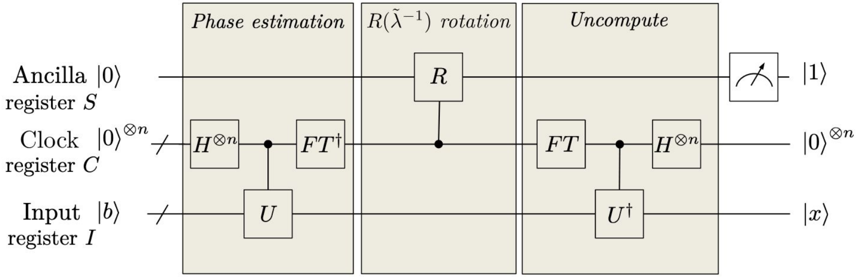
Qubits Required:

- *1* ancilla qubit
- *n* qubits to store eigen values of A in binary format --- with precision up to *n* bits.
- a memory of *O(log(N))*, that initially stores $|b\rangle$ and later $|x\rangle$.

Any quantum algorithm, proceeds as follows:
Initialization ⟶ Superposition ⟶ Unitary Transformation ⟶ Post Processing ⟶ Measurement

Constructing QHT/QLSA Sub-Routine

https://github.com/amitabhyadav/QHT.QPR



What?

Given $A\vec{x} = \vec{b}$ for a Matrix A and vector $\vec{b}$, solve for $\vec{x}$.

Conditions:

A is Hermitian matrix i.e. $A = A^\dagger$. A must be sparse and low-rank.

We have to represent $\vec{x}$ and $\vec{b}$ as quantum states, $|x\rangle$ and $|b\rangle$. So, we normalize them, i.e. $|x| = |b| = 1$

Qubits Required:

- *1* ancilla qubit
- *n* qubits to store eigen values of A in binary format --- with precision up to *n* bits.
- a memory of *O(log(N))*, that initially stores $|b\rangle$ and later $|x\rangle$.

Any quantum algorithm, proceeds as follows:

Initialization ⟶ Superposition ⟶ Unitary Transformation ⟶ Post Processing ⟶ Measurement

HHL Algorithm:

1. **Initialization** : $|0\rangle_a, |0\rangle_r, |b\rangle_m$ = for *ancilla*, *register* and *memory*
   Initial state, therefore is: $|0\rangle_a|0\rangle_r|b\rangle_m$
2. **(Step 1)** Perform quantum phase estimation using the unitary transformation e^{iAt}, where *t* is the time.
3. Map the eigenvalues λ_j into the register (in binary form), so the state: $|0\rangle_a|0\rangle_r|b\rangle_m$ becomes:
$$|0\rangle_a|0\rangle_r|b\rangle_m \Rightarrow \sum_{j=1}^{N} \beta_j |0\rangle_a|\lambda_j\rangle_r|u_j\rangle_m$$
4. **(Step 2)** Rotate the ancilla qubit $|0\rangle_a$ to $\sqrt{1-\frac{C^2}{\lambda_j^2}}|0\rangle_a + \frac{C}{\lambda_j}|1\rangle_a$ for each λ_j. This is done by controlled rotation on ancilla qubit $|0\rangle_a$.
5. After Step 2, the state becomes:
$$\sum_{j=1}^{N} \beta_j \left(\sqrt{1-\frac{C^2}{\lambda_j^2}}|0\rangle_a + \frac{C}{\lambda_j}|1\rangle_a\right)|\lambda_j\rangle_r|u_j\rangle_m$$
6. **(Step 3)** Uncompute. Reverse Step 1, the state we get is:
$$\sum_{j=1}^{N} \beta_j \left(\sqrt{1-\frac{C^2}{\lambda_j^2}}|0\rangle_a + \frac{C}{\lambda_j}|1\rangle_a\right)|0\rangle_r|u_j\rangle_m$$
   This step does inverse phase estimation and disentanglement. This restores the register to $|0\rangle$.
7. **Measurement** : By selecting the $|1\rangle$ in ancilla qubit, we get the following on the memory qubit:
   In the memory qubit, we get: $|x\rangle \approx \sum_{j=1}^{N} C\left(\frac{\beta_j}{\lambda_j}\right)|u_j\rangle$

The output of the algorithm:
Expectation values of Pauli observables of $|x\rangle$

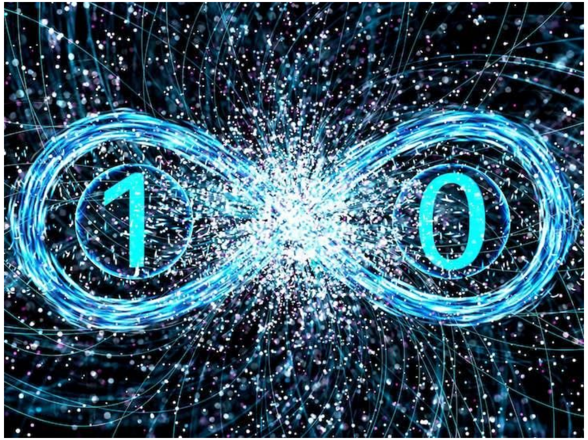
Note that, the accuracy of the result depends on the following factors:

- Register size
- Choice of parameters C and t

Current Work on QHT/QLSA

https://github.com/amitabhyadav/QHT.QPR

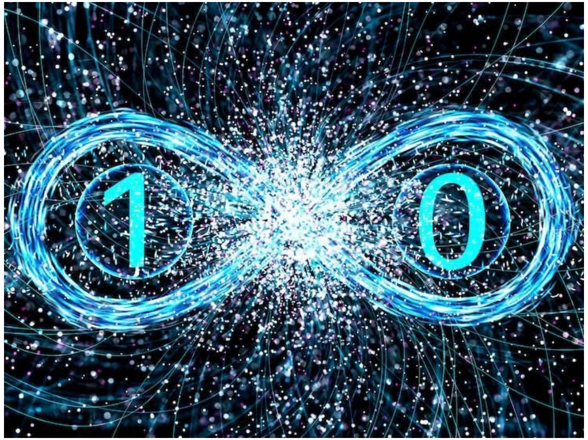
- Algorithmic Pipeline Design
- Optimal Data Encoding and solution for arbitrary sizes of matrices.
  - Can we do more with larger sizes? Larger Sizes ≠ More Parallelism.
- Post HHL Measurement Result Extraction Sub-routine.



Current Work on QHT/QLSA

https://github.com/amitabhyadav/QHT.QPR

- Algorithmic Pipeline Design
- Optimal Data Encoding and solution for arbitrary sizes of matrices.
    - Can we do more with larger sizes? Larger Sizes ≠ More Parallelism.
- Post HHL Measurement Result Extraction Sub-routine.



- Future Work:
    - Benchmarking and Precision.
    - Design for Dense Matrices.
    - Computational Complexity Analysis.
    - Evaluation of fault-resilience.
    - Low Circuit-Depth implementation.

Conclusion

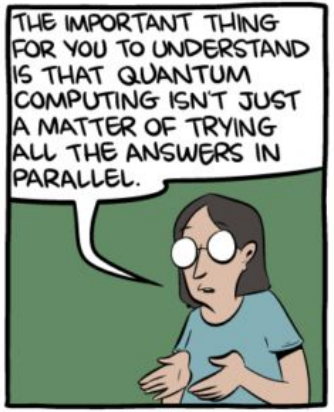
- HHL Algorithm for Linear Equation Solver Subroutine.

- Implementation for 2 x 2, 3 x 3 for low-rank sparse matrices and derivation of matrix sizes to fit 5 parameters of particle Tracks.

- Hough Transform uses linear set of equations.
  - HHL offers guaranteed exponential speed-up for solving linear equations.
  - Speedup for measurement extraction? TBD.

- Algorithmic pipeline development using Quantum Hough Transform using HHL.

- QHT.QPR Implementation on NISQ devices.



Modern Quantum Algorithmic Landscape

for Gate Model QC (...lots to Explore and Invent!)

- **Simulation**
  - Variational Quantum Eigensolver (VQEs) and α-VQEs
  - QITE/QLanczos
  - Thermal Averages: METTS Algorithm, Quantum Importance Sampling
  - Unitary Quantum Dynamics,
  - Field Theory Simulation, HEP Simulations etc.

Modern Quantum Algorithmic Landscape

for Gate Model QC (...lots to Explore and Invent!)

- **Simulation**
  - Variational Quantum Eigensolver (VQEs) and α-VQEs
  - QITE/QLanczos
  - Thermal Averages: METTS Algorithm, Quantum Importance Sampling
  - Unitary Quantum Dynamics,
  - Field Theory Simulation, HEP Simulations etc.

- **Scientific Computing/Mathematical Methods**
  Finite Element Methods, Partial Differential Equations, Poisson Solver (ref)
  - Quantum Linear System Algorithms/HHL
  - Quantum Support Vector Machine
  - Quantum Phase Estimation
  - Quantum Fourier Transform

Modern Quantum Algorithmic Landscape

for Gate Model QC (...lots to Explore and Invent!)

- **Simulation**
  - Variational Quantum Eigensolver (VQEs) and α-VQEs
  - QITE/QLanczos
  - Thermal Averages: METTS Algorithm, Quantum Importance Sampling
  - Unitary Quantum Dynamics,
  - Field Theory Simulation, HEP Simulations etc.

- **Scientific Computing/Mathematical Methods**
  Finite Element Methods, Partial Differential Equations, Poisson Solver (ref)
  - Quantum Linear System Algorithms/HHL
  - Quantum Support Vector Machine
  - Quantum Phase Estimation
  - Quantum Fourier Transform

- **Quantum Machine Learning**
  - Bayesian Deep Learning
  - Quantum Tensor Networks
  - Quantum Neural Networks
  - Quantum Principal Component Analysis

Modern Quantum Algorithmic Landscape

for Gate Model QC (...lots to Explore and Invent!)

- **Simulation**
  - Variational Quantum Eigensolver (VQEs) and α-VQEs
  - QITE/QLanczos
  - Thermal Averages: METTS Algorithm, Quantum Importance Sampling
  - Unitary Quantum Dynamics,
  - Field Theory Simulation, HEP Simulations etc.

- **Scientific Computing/Mathematical Methods**
  Finite Element Methods, Partial Differential Equations, Poisson Solver (ref)
  - Quantum Linear System Algorithms/HHL
  - Quantum Support Vector Machine
  - Quantum Phase Estimation
  - Quantum Fourier Transform

- **Quantum Machine Learning**
  - Bayesian Deep Learning
  - Quantum Tensor Networks
  - Quantum Neural Networks
  - Quantum Principal Component Analysis

- **Quantum Graph Theory/Other Approaches**
  - Quantum Random Walks
  - Quantum Monte Carlo
  - Quantum Probably Approximately Correct
  - Grover’s Search

Modern Quantum Algorithmic Landscape

for Gate Model QC (...lots to Explore and Invent!)

- **Simulation**
  - Variational Quantum Eigensolver (VQEs) and α-VQEs
  - QITE/QLanczos
  - Thermal Averages: METTS Algorithm, Quantum Importance Sampling
  - Unitary Quantum Dynamics,
  - Field Theory Simulation, HEP Simulations etc.

- **Scientific Computing/Mathematical Methods**
  Finite Element Methods, Partial Differential Equations, Poisson Solver (ref)
  - Quantum Linear System Algorithms/HHL
  - Quantum Support Vector Machine
  - Quantum Phase Estimation
  - Quantum Fourier Transform

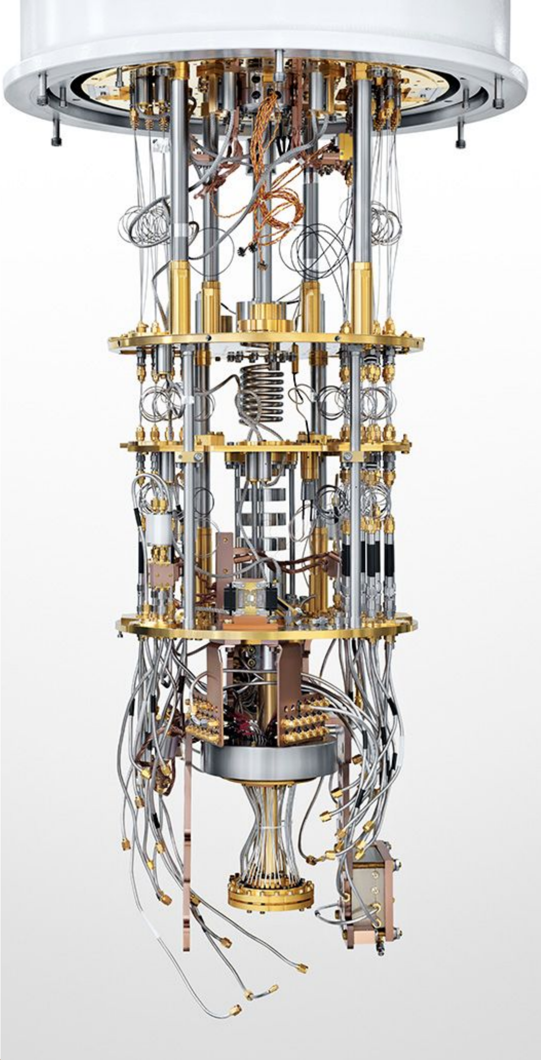
- **Quantum Machine Learning**
  - Bayesian Deep Learning
  - Quantum Tensor Networks
  - Quantum Neural Networks
  - Quantum Principal Component Analysis

- **Quantum Graph Theory/Other Approaches**
  - Quantum Random Walks
  - Quantum Monte Carlo
  - Quantum Probably Approximately Correct
  - Grover's Search

- **Optimization**
  - Quantum Alternating Ansatz Algorithms (QAAA)
    - QAOA and Hamiltonian-based QAOA
  - Quantum Annealing (on D-Wave)

- **Quantum-Inspired Classical Algorithms**
  - Recommendation Systems

Thank you :)



HEP.QPR Project
https://hep-qpr.lbl.gov/

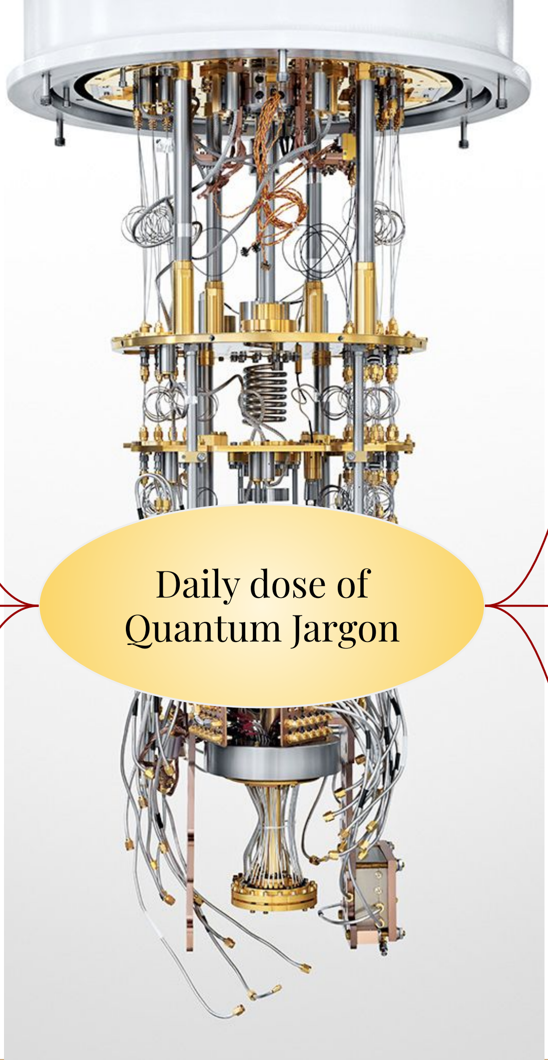
Link to Slides and
Project Repository

BERKELEY LAB ATLAS EXPERIMENT

Thank you :)

HEP.QPR Project
https://hep-qpr.lbl.gov/

Link to Slides and
Project Repository



Daily dose of Quantum Jargon

Weak Quantum Value
Improving solution of any problem, using a QC (possibly in combination with classical methods) so that the results are better than any available purely classical solution

Quantum Value
improving solution of a valuable problem, using a QC (possibly in combination with classical methods) so that the results are better than any available purely classical solution

Weak Quantum Supremacy
solution of any problem using a QC, faster or better than any available classical solution

Quantum Advantage
Solution of a valuable problem using a QC, faster or better than any available classical solution

Quantum Supremacy
Mathematical proof that any problem has a super-polynomial separation w.r.t. assumptions (e.g. P ≠ NP) between any possible quantum algorithm and any possible classical algorithm. The exhibition of the solution of this problem by a QC at a performance (size, speed, or efficiency) that is infeasible with any available classical computer.

Strong Quantum Advantage
Q supremacy + Q advantage (e.g. breaking 2048 RSA with Shor's algorithm)

BERKELEY LAB

ATLAS EXPERIMENT