# Developing Quantum Programming Software

Jin-Long Huang(黄金龙)

Supervised by Professor Man-Hong Yung(翁文康)

For the Fulfillment of National Training Program of Innovation and Entrepreneurship for Undergraduates.
(国家级大学生创新创业训练计划)

Southern University of Science and Technology

September 21, 2017

# Outline

- Quantum Programming Compiler: Quipper
  - Quantum Computer Architecture
  - Introduction to Quipper
  - Elementary Quantum Algorithms in Quipper
- Quantum SVM to Recognize Handwritten 6 and 9
  - Classical SVM
  - Quantum SVM
  - Characterization of 6 an 9
- Quantum Supremacy: Boson Sampling

# Why Quantum Software?

**Quantum Algorithms:**

Quantum Fourier Transform
Shor’s Algorithm
Eigenvalues Estimation
Grover’s Algorithm
HHL
Quantum SVM
Quantum Reinforcement Learning
…

**Experimental Realization:**

Superconducting
Ion Trap
NMR
Cavity QED
Optical
Topological
Diamond-based
…

How can we test the performance of quantum algorithms in real situation?

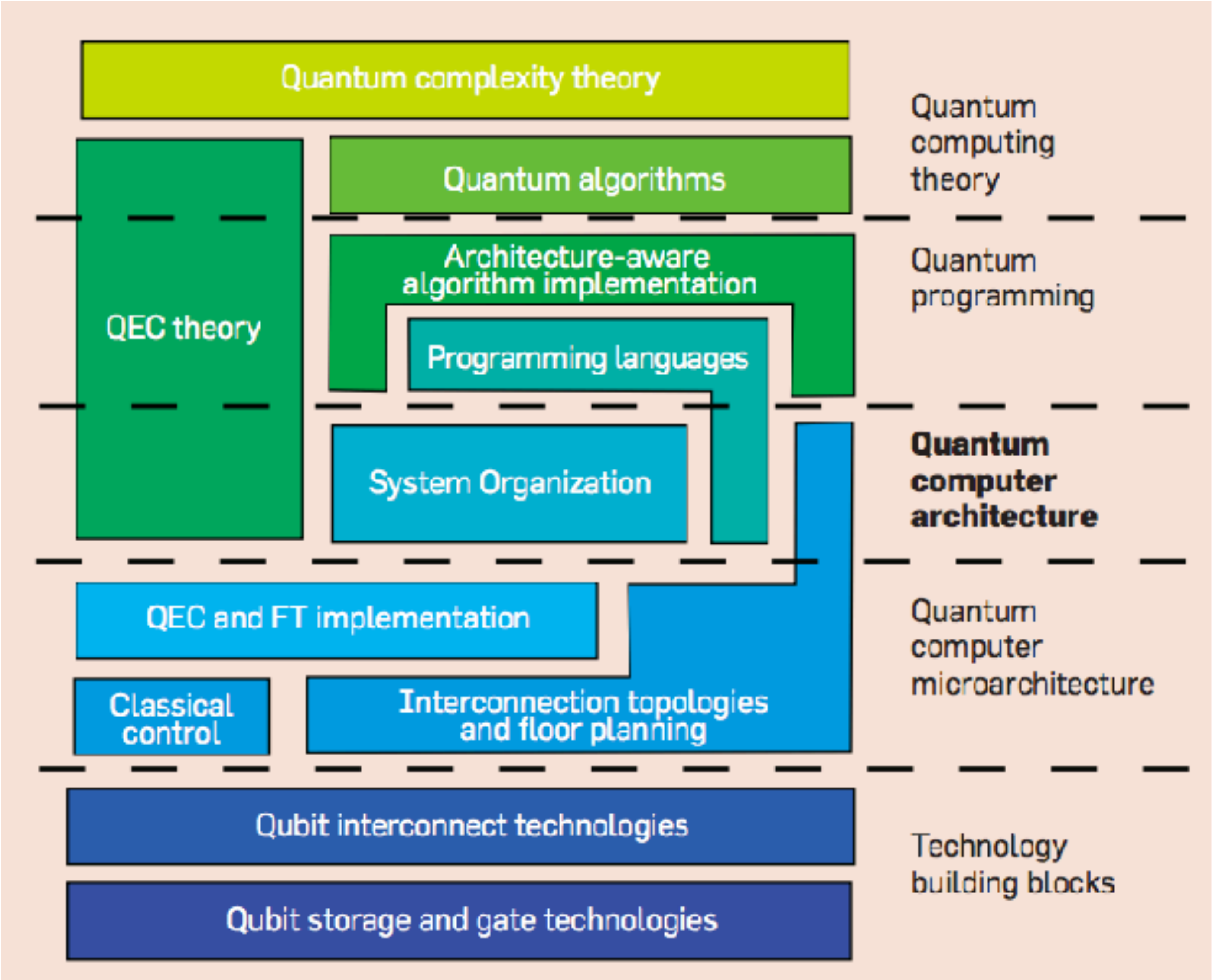
What can we do when we have 100/1000 qubits?

How to design a quantum computer architecture for best performance?

…

# Quantum Computer Architecture

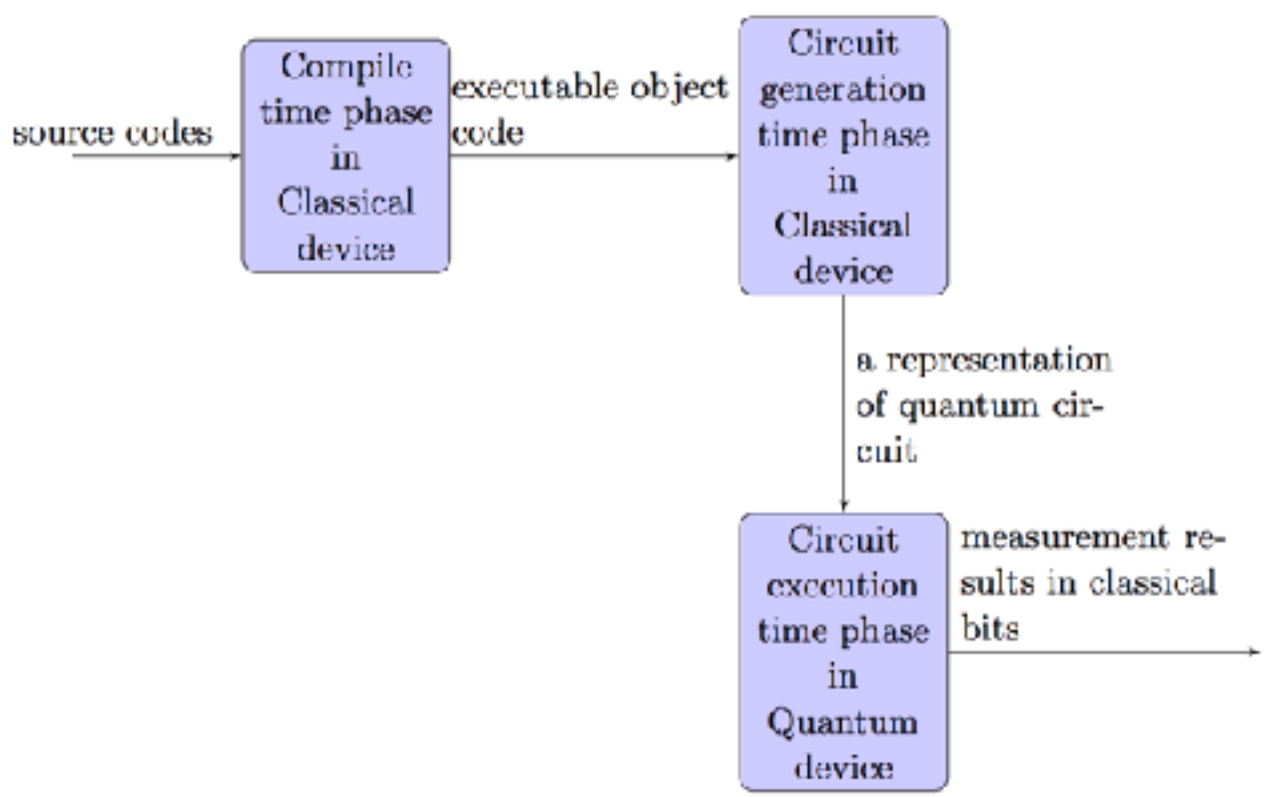
A Blueprint




Van Meter, Rodney, and Clare Horsman. "A blueprint for building a quantum computer." *Communications of the ACM* 56.10 (2013): 84-93.

# Introduction to Quipper

- A unified general-purpose programming framework for quantum computation.
- Quantum version of Java — A high-level circuit description language. Module.
- Compiled to single qubit gates and two qubits gates. Optimized. Reverse.
- Generate quantum circuits and simulate on classical back-ends.
- Analysis cost of resource of quantum algorithms.
- Embedded on Haskell, a pure functional classical programming language.




Siddiqui, Safat, Mohammed Jahirul Islam, and Omar Shehab. "Five Quantum Algorithms Using Quipper." *arXiv preprint arXiv:1406.4481* (2014).

# Elementary Quantum Algorithms in Quipper

- Hello Quantum World
- Entanglement Generation
- Deutsch algorithm
- Deutsch-Jozsa algorithm
- Grover’s Search
- Linear Combination of Unitary

# Elementary Quantum Algorithms in Quipper

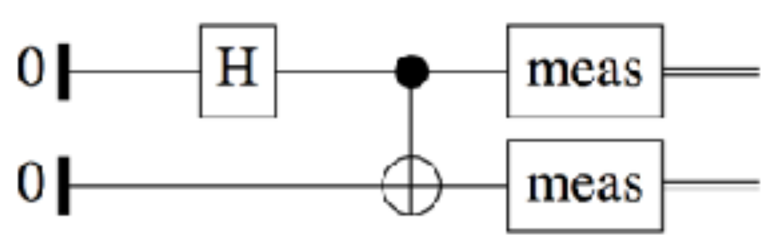
## Hello Quantum World

```
import Quipper

hello_quantum_world :: Bool -> Circ Qubit
hello_quantum_world var = do
  qbit <- qinit var
  label (qbit) ("<| Hello Quantum World! |>")
  return qbit

main = print_simple Preview (hello_quantum_world False)
```


## Entanglement Generation




```
➜  myQuipper ./entanglement
(True,True)
(False,False)
(True,True)
(False,False)
(False,False)
(False,False)
(False,False)
(True,True)
(True,True)
(False,False)
```

# Elementary Quantum Algorithms in Quipper

The Deutsch Problem:

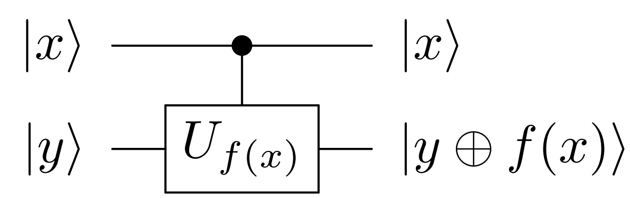
Input: Oracle $U_f$ where $f : \{0,1\} \to \{0,1\}$.

Problem: Calculate $f(0) \oplus f(1)$.

Classical: 2 queries

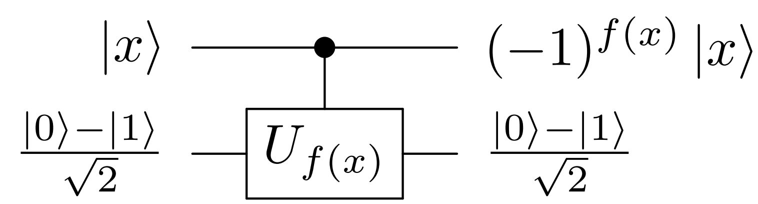
Oracle:

$$U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$$




Phase Kick-Back:

$$U_f : |x\rangle|\frac{|0\rangle - |1\rangle}{\sqrt{2}}\rangle \mapsto (-1)^{f(x)}|x\rangle|\frac{|0\rangle - |1\rangle}{\sqrt{2}}\rangle$$




## Deutsch Algorithm:

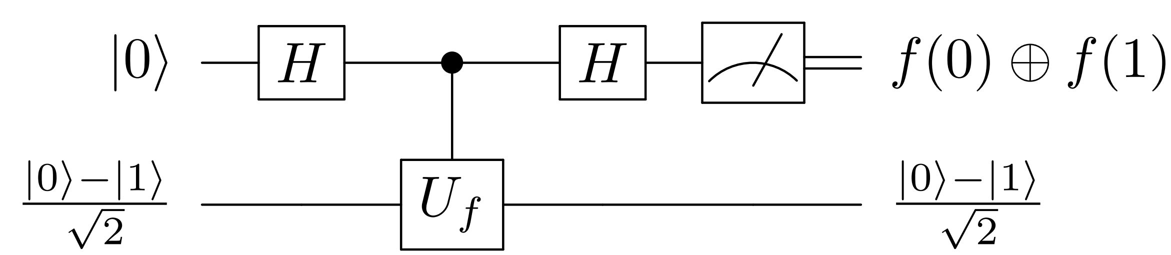
The first qubit:

$$|0\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{f} \frac{(-1)^{f(0)}\,|0\rangle + (-1)^{f(1)}\,|1\rangle}{\sqrt{2}}$$

$$= (-1)^{f(0)}\frac{|0\rangle + (-1)^{f(0)+f(1)}\,|1\rangle}{\sqrt{2}}$$

$$\xrightarrow{H} (-1)^{f(0)}\,|f(0) \oplus f(1)\rangle$$




Quantum: 1 query

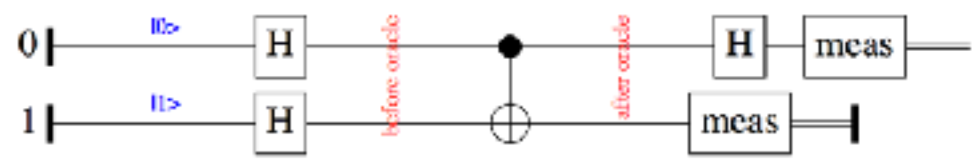
# Elementary Quantum Algorithms in Quipper

## Deutsch algorithm

```
-- | initialize oracle function f(x)
constant_oracle_function_1 :: (Qubit, Qubit) -> Circ (Qubit, Qubit)
constant_oracle_function_1 (x, y) = do
  -- f(0) = 0; f(1) = 0
  return (x, y)
```

```
-- | initialize oracle function f(x)
balanced_oracle_function_1 :: (Qubit, Qubit) -> Circ (Qubit, Qubit)
balanced_oracle_function_1 (x, y) = do
  -- f(0) = 1; f(1) = 0
  qnot_at y `controlled` x
  return (x, y)
```


Constant Oracle




Balanced Oracle

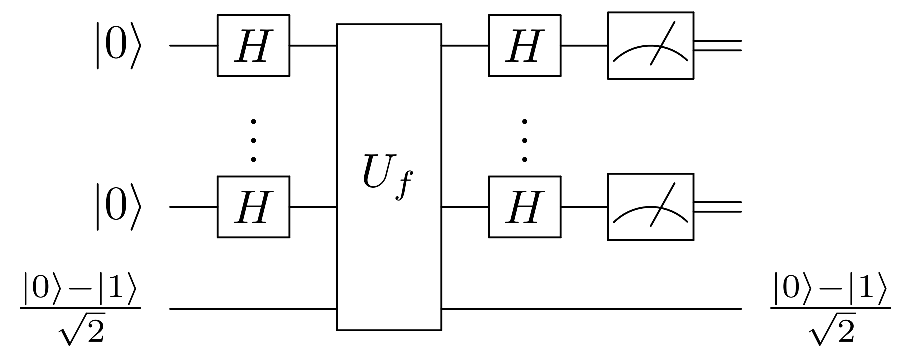
```
➜  myQuipper ./deutsch
Given oracle is Constant.
Given oracle is Constant.
Given oracle is Balanced.
Given oracle is Balanced.
```

# Elementary Quantum Algorithms in Quipper

A boolean function $f$ is constant meaning $f(x)$ is the same for all $x$.
$f$ is balanced meaning $f(x) = 0$ for exactly half of the input strings $x$, and $f(x) = 1$ for the other half of the inputs.

The Deutsch-Jozsa Problem:
Input: Oracle $U_f$ where $f : \{0,1\}^n \to \{0,1\}$.
Promise: $f$ is either constant or balanced.
Problem: Determine whether $f$ is constant or balanced.

Classical: $2^n/2 + 1$ queries




First n qubits: $$|0\rangle^{\otimes n} \xrightarrow{H} \frac{1}{\sqrt{2^n}} \sum_{\vec{x} \in \{0,1\}^n} |\vec{x}\rangle$$

$$\xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{\vec{x} \in \{0,1\}^n} (-1)^{f(\vec{x})} |\vec{x}\rangle$$

$$\Downarrow$$

$$H^{\otimes n} |\vec{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\vec{y} \in \{0,1\}^n} (-1)^{\vec{x} \cdot \vec{y}} |\vec{y}\rangle \quad \Rightarrow \quad \frac{1}{2^n} \sum_{\vec{x} \in \{0,1\}^n} \left[ (-1)^{f(\vec{x})} \cdot \sum_{\vec{y} \in \{0,1\}^n} (-1)^{\vec{x} \cdot \vec{y}} |\vec{y}\rangle \right]$$

$$\Downarrow$$

$$\text{Amplitude of } |00 \cdots 0\rangle : \frac{1}{2^n} \sum_{\vec{x} \in \{0,1\}^n} (-1)^{f(\vec{x})}$$

$$\Downarrow$$

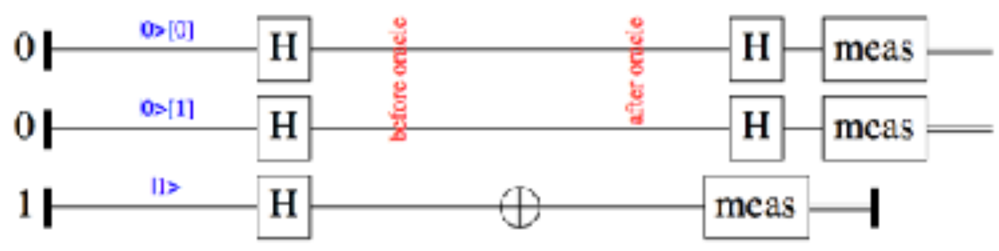
$f$ is constant: $\pm 1$

$f$ is balanced: $0$

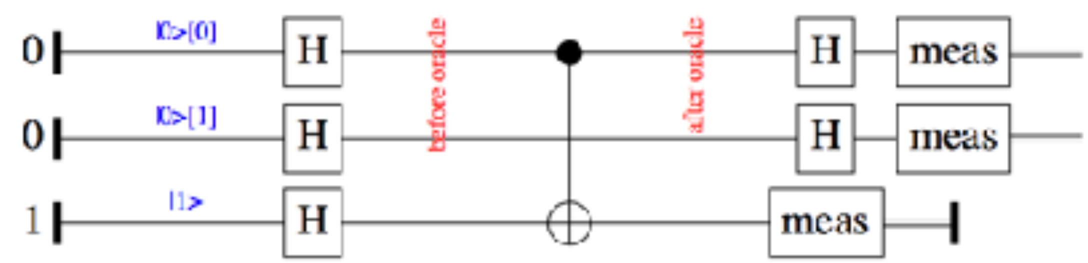
Quantum: 1 query

# Elementary Quantum Algorithms in Quipper

## Deutsch-Jozsa algorithm




Constant Oracle




Balanced Oracle

➜ myQuipper ./deutsch_jozsa
Given oracle is Constant.
Given oracle is Constant.
Given oracle is Balanced.
Given oracle is Balanced.
Given oracle is Balanced.
Given oracle is Balanced.
Given oracle is Balanced.
Given oracle is Balanced.

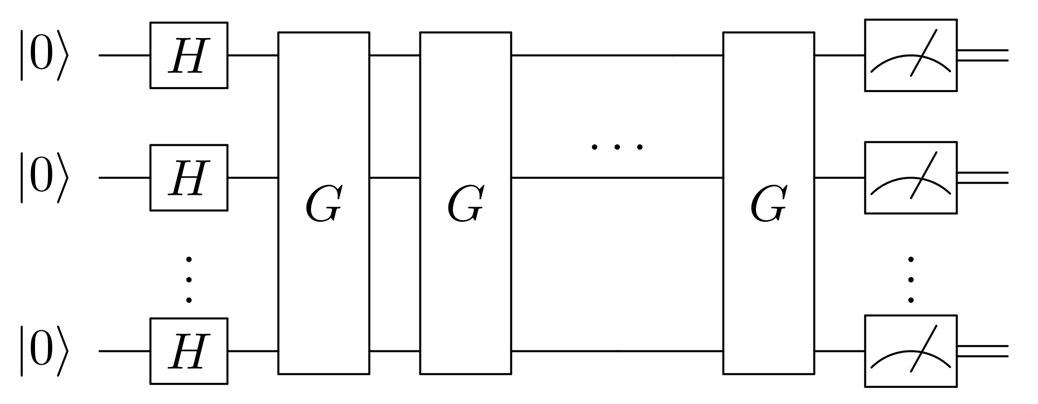
# Elementary Quantum Algorithms in Quipper

The Search Problem:

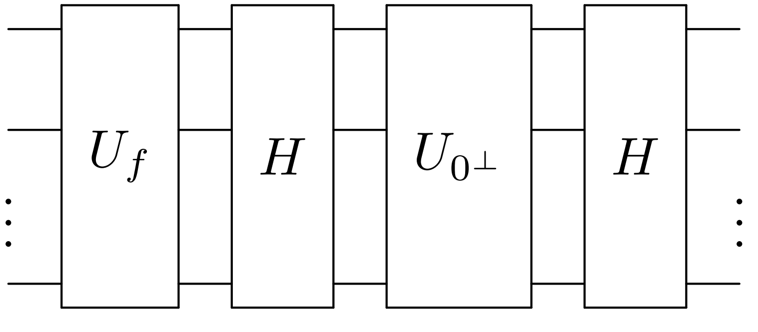
Input: Oracle $U_f$ where $f : \{0,1\}^n \to \{0,1\}$.

Problem: Find an input $x \in \{0,1\}^n$ such that $f(x) = 1$.

The Grover Search Circuit:




Where G is the Grover iterator:



# Elementary Quantum Algorithms in Quipper

## Grover's Search

$$\text{Probability: } \frac{1}{2^n} \xrightarrow{2^n - 1 \text{ queries}} 1$$

$$\text{Amplitude: } \frac{1}{\sqrt{2^n}} \xrightarrow[\uparrow \frac{1}{\sqrt{2^n}}]{\sqrt{2^n} \text{ queries}} 1$$

$$|\varphi\rangle \equiv H|00\cdots 0\rangle = \frac{1}{2^n}\sum_{x=0}^{2^n-1}|x\rangle = \frac{1}{\sqrt{2^n}}|\omega\rangle + \sqrt{\frac{2^n - 1}{2^n}}|\psi_{\text{bad}}\rangle$$

$$\Downarrow$$

$$U_f|\varphi\rangle = \cos(2\theta)|\varphi\rangle - \sin(2\theta)|\varphi\rangle$$

$$U_{\varphi^\perp}U_f|\varphi\rangle = \sin(3\theta)|\omega\rangle + \cos(3\theta)|\psi_{\text{bad}}\rangle$$

$$(U_{\varphi^\perp}U_f)^k|\varphi\rangle = \sin((2k+1)\theta)|\omega\rangle + \cos((2k+1)\theta)|\psi_{\text{bad}}\rangle$$

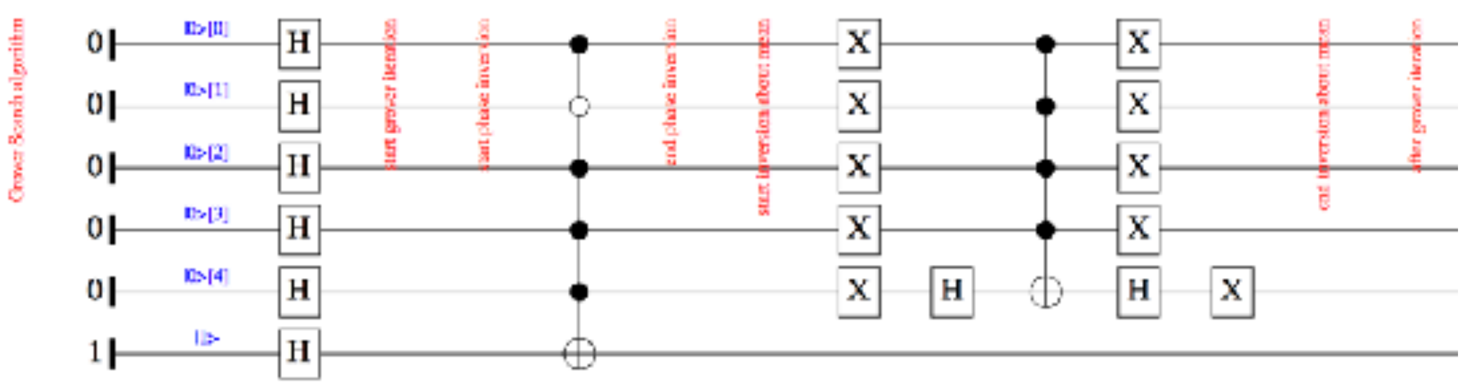
$$\Downarrow$$

$$(2k+1)\theta \approx \tfrac{\pi}{2} \Rightarrow k \approx \tfrac{\pi}{4}\sqrt{2^n}$$

# Elementary Quantum Algorithms in Quipper

## Grover’s Search

```
-- | initialize n_qubit_oracle’s function
n_qubit_oracle_function :: ([Qubit],Qubit) -> Circ ([Qubit],Qubit)
n_qubit_oracle_function (controlled_qubit, target_qubit) = do
  qnot_at target_qubit `controlled` controlled_qubit .==. [1,0,1,1,1]
  return (controlled_qubit, target_qubit)
```



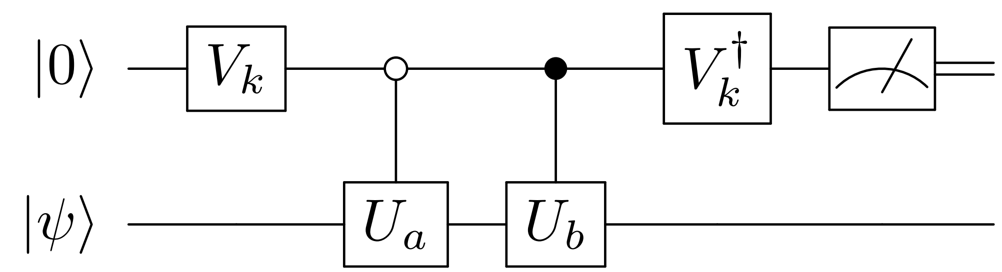
➜ myQuipper ./grovers_search
[True,False,True,True,True]

# Elementary Quantum Algorithms in Quipper

## Linear Combination of Unitary Algorithm

How can we implement $\kappa U_a + U_b$?

$$V_k := \begin{pmatrix} \sqrt{\frac{\kappa}{\kappa+1}} & \frac{-1}{\sqrt{\kappa+1}} \\ \frac{1}{\sqrt{\kappa+1}} & \sqrt{\frac{\kappa}{\kappa+1}} \end{pmatrix}$$




$$\begin{aligned}
|0\rangle |\psi\rangle &\xrightarrow{V_k} \left( \sqrt{\frac{\kappa}{\kappa+1}} |0\rangle + \frac{1}{\sqrt{\kappa+1}} |1\rangle \right) |\psi\rangle \\
&\xrightarrow{U_a, U_b} \left( \sqrt{\frac{\kappa}{\kappa+1}} |0\rangle U_a |\psi\rangle + \frac{1}{\sqrt{\kappa+1}} |1\rangle U_b |\psi\rangle \right) \\
&\xrightarrow{V_k^\dagger} |0\rangle \left( \frac{\kappa}{\kappa+1} U_a + \frac{1}{\kappa+1} U_b \right) |\psi\rangle + |1\rangle \frac{\sqrt{\kappa}}{\kappa+1} (U_b - U_a) |\psi\rangle
\end{aligned}$$

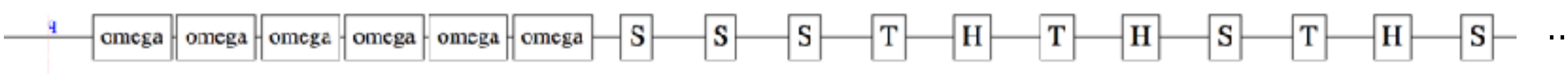
The probability of failure: $P_1 = \frac{||U_b - U_a||^2 \kappa}{(\kappa+1)^2} \leq \frac{4\kappa}{(\kappa+1)^2}$

# Elementary Quantum Algorithms in Quipper

## Linear Combination of Unitary Algorithm

```
-- | define the first Vk gate
linear_combination_circuit :: IO ()
linear_combination_circuit = do
  -- get a random number
  g1 <- newStdGen
  let circ1 = approximate_synthesis_u2 True prec op1 g1
  -- call linear_combination_circuit2
  linear_combination_circuit2
  -- print the circuit for Vk1 in pdf
  print_simple Preview circ1
  where
  prec = 10 * digits
  op1 = matrix2x2 (0.9486832980505138, -0.3162277660168379)
                  (0.3162277660168379,  0.9486832980505138)
```

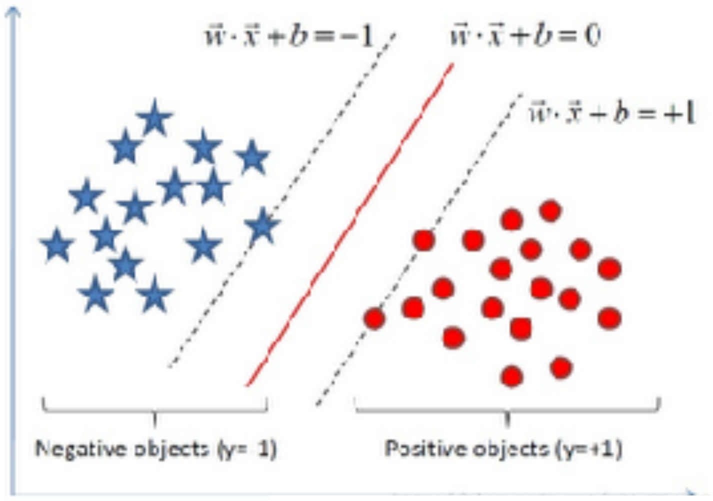
Decomposition of Vk gate:



# Quantum SVM to Recognize Handwritten 6 and 9

- Classical SVM
  - Basic Model
  - Kernel Function
  - Least Square SVM
- Quantum SVM
  - Quantum Fourier Transform(QFT)
  - Eigenvalues Estimation
  - Controlled-Rotation
  - SWAP_Test
  - Non-sparse Matrix Exponentiation
- Characterization of 6 an 9

# Classical SVM




## Basic Model:

Classifier function : $f(x) = \text{sign}\ (w^T x + b)$.

Geometric distance: $\gamma_i = \frac{|w^T x_i + b|}{||w||} = \frac{y(w^T x_i + b)}{||w||} = \frac{y f(x_i)}{||w||}$

Objective function: $\max_{w,b} \min_i \gamma_i = \max_{w,b} \frac{1}{||w||}$

Dual problem: $\min_{w,b} \max_{\alpha_i \geq 0} \mathcal{L}(w, b, \alpha)$ to $\max_{\alpha_i \geq 0} \min_{w,b} \mathcal{L}(w, b, \alpha)$

Final Lagrangian function: $\mathcal{L}(w, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j$. It's a quadratic programming problem.

Andrew NG, Lecture notes, Support Vector Machines.

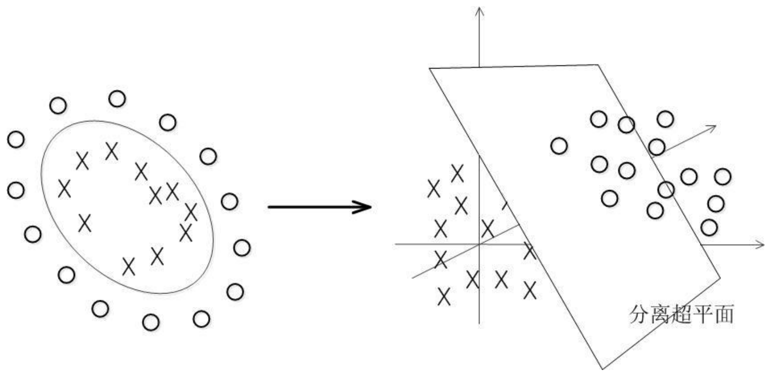
# Classical SVM

## Kernel:

$$a_1X_1 + a_2X_1^2 + a_3X_2 + a_4X_2^2 + a_5X_1X_2 + a_6 = 0$$

$$\Downarrow \quad \phi(X_1, X_2) = (X_1, X_1^2, X_2, X_2^2, X_1X_2)^T$$

$$\sum_{i=1}^{5} a_i Z_i + a_6 = 0$$




Objective Function:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \langle x_i | x \rangle + b$$

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \langle \phi(x_i) | \phi(x) \rangle + b$$

Let: $x_1 = (\eta_1, \eta_2)^T$, $x_2 = (\xi_1, \xi_2)^T$

We can see that:

$$\langle \phi(x_1) | \phi(x_2) \rangle = \eta_1\xi_1 + \eta_1^2\xi_1^2 + \eta_2\xi_2 + \eta_2^2\xi_2^2 + \eta_1\eta_2\xi_1\xi_2$$

$$(\langle x_1 | x_2 \rangle + 1)^2 = 2\eta_1\xi_1 + \eta_1^2\xi_1^2 + 2\eta_2\xi_2 + \eta_2^2\xi_2^2 + 2\eta_1\eta_2\xi_1\xi_2 + 1$$

$$\phi(X_1, X_2) = (\sqrt{2}X_1, X_1^2, \sqrt{2}X_2, X_2^2, \sqrt{2}X_1X_2, 1)^T$$

Kernel Function: $\kappa(x_1, x_2) = (\langle x_1 | x_2 \rangle + 1)^2$

# Classical SVM

## Least Square SVM:

Objective function: $\frac{1}{2}||w||^2 + \frac{\gamma}{2}\sum_{i=1}^{n} e_k^2$

Constrains:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i \varphi(x_i)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial \mathcal{L}}{\partial e_i} = 0 \Rightarrow \alpha_i = \gamma e_i, i = 1, ..., n$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \Rightarrow y_i(w^T\varphi(x_i) + b) - 1 + e_i = 0, i = 1, ..., n$$

Which is equivalent to:

$$\begin{pmatrix} 0 & -Y^T \\ Y & ZZ^T + \gamma^{-1}I \end{pmatrix} \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{1} \end{pmatrix}$$

where $Z = (\varphi(x_1)^T y_1, \varphi(x_2)^T y_2, ..., \varphi(x_n)^T y_n)^T, Y = (y_1, ..., y_n)^T, \vec{1} = (1, ..., 1)^T, e = (e_1, e_2, ..., e_n)^T, \alpha = (\alpha_1, ..., \alpha_n)^T$.

Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." Neural processing letters 9.3 (1999): 293-300.

# Quantum SVM

Quantum Fourier Transform(QFT):

$$\text{Discrete Fourier Transform: } x_j \to \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} x_k e^{2\pi ijk/2^n}$$

$$\text{Quantum Fourier Transform: } \sum_{j=0}^{2^n-1} x_j \left|j\right\rangle \to \sum_{k=0}^{2^n-1} y_k \left|k\right\rangle$$

$$\text{where } \left|j\right\rangle \to \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n} \left|k\right\rangle$$

Classical: $O(n2^n)$ Quantum: $O(n^2)$

# Quantum SVM

## Eigenvalues Estimation:

Assume $A\,|\psi_i\rangle = \lambda_i\,|\psi_i\rangle$ and $e^{iAT}$ can be implemented efficiently. Given $|\psi_i\rangle$,

$$|\psi_i\rangle\,|0\rangle \xrightarrow{\text{Eigenvalue Estimation}} |\psi_i\rangle\left|\tilde{\lambda}_i\right\rangle \tag{1}$$

where $|\tilde{\lambda}_i - \lambda_i| \leq \varepsilon$

## SWAP-test:

Given two states $|\varphi_1\rangle$ and $|\varphi_2\rangle$ as input, SWAP-test can estimate $|\langle\varphi_1|\varphi_2\rangle|^2$.

$$\begin{aligned}|+\rangle\,|\varphi_1\rangle\,|\varphi_2\rangle &\xrightarrow{\text{control-SWAP}} \frac{|0\rangle\,|\varphi_1\rangle\,|\varphi_2\rangle + |1\rangle\,|\varphi_2\rangle\,|\varphi_1\rangle}{\sqrt{2}} \\ &\xrightarrow{\text{measure 1st qubit on } |\pm\rangle} \frac{|\varphi_1\rangle\,|\varphi_2\rangle \pm |\varphi_2\rangle\,|\varphi_1\rangle}{2}\end{aligned}$$

The probability to get the outcome $|\pm\rangle$ is

$$|\tfrac{|\varphi_1\rangle|\varphi_2\rangle\pm|\varphi_2\rangle|\varphi_1\rangle}{2}|^2 = \tfrac{1\pm|\langle\varphi_1|\varphi_2\rangle|^2}{2}$$

# Quantum SVM

## Controlled-Rotation:

$$|\theta\rangle|0\rangle \to |\theta\rangle e^{-i\theta\sigma_y}|0\rangle = |0\rangle(\cos\theta|0\rangle + \sin\theta|1\rangle)$$

which can be generalized to

$$|x\rangle|0\rangle \xrightarrow{\text{control-rotation}} |x\rangle\left(f(x)|0\rangle + \sqrt{1-f^2(x)}|1\rangle\right)$$

## Non-sparse Matrix Exponentiation:

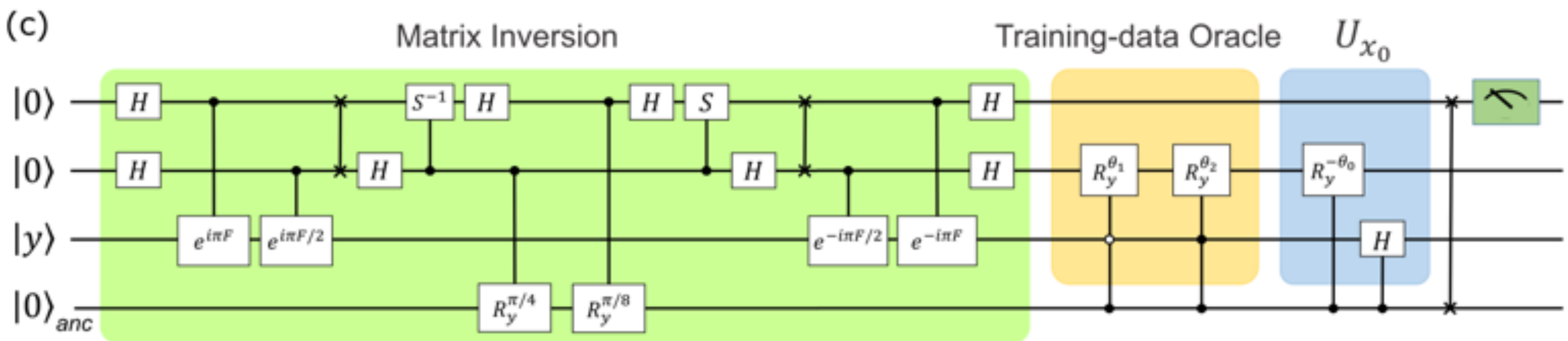
Suppose the density matrix $\rho$ is in register 1 and $\sigma$ is in register 2, and $S$ is the SWAP operation

$$\begin{aligned}\mathrm{Tr}_1 e^{-iS\delta t}(\rho\otimes\sigma)e^{iS\delta t} &= \sigma - i[\rho,\sigma]\delta t + O(\delta t^2)\\ &= e^{-i\rho\delta t}\sigma e^{i\rho\delta t} + O(\delta t^2)\end{aligned}$$

The first step is due to

$$\mathrm{Tr}_1(\rho\otimes\sigma)S = \sigma\rho$$
$$\mathrm{Tr}_1 S(\rho\otimes\sigma) = \rho\sigma$$

# Characterization of 6 an 9




Feature vector for training data are $x_1 = (0.987, 0.159)$ for standard 6 and $x_2 = (0.354, 0.935)$ for standard 9.

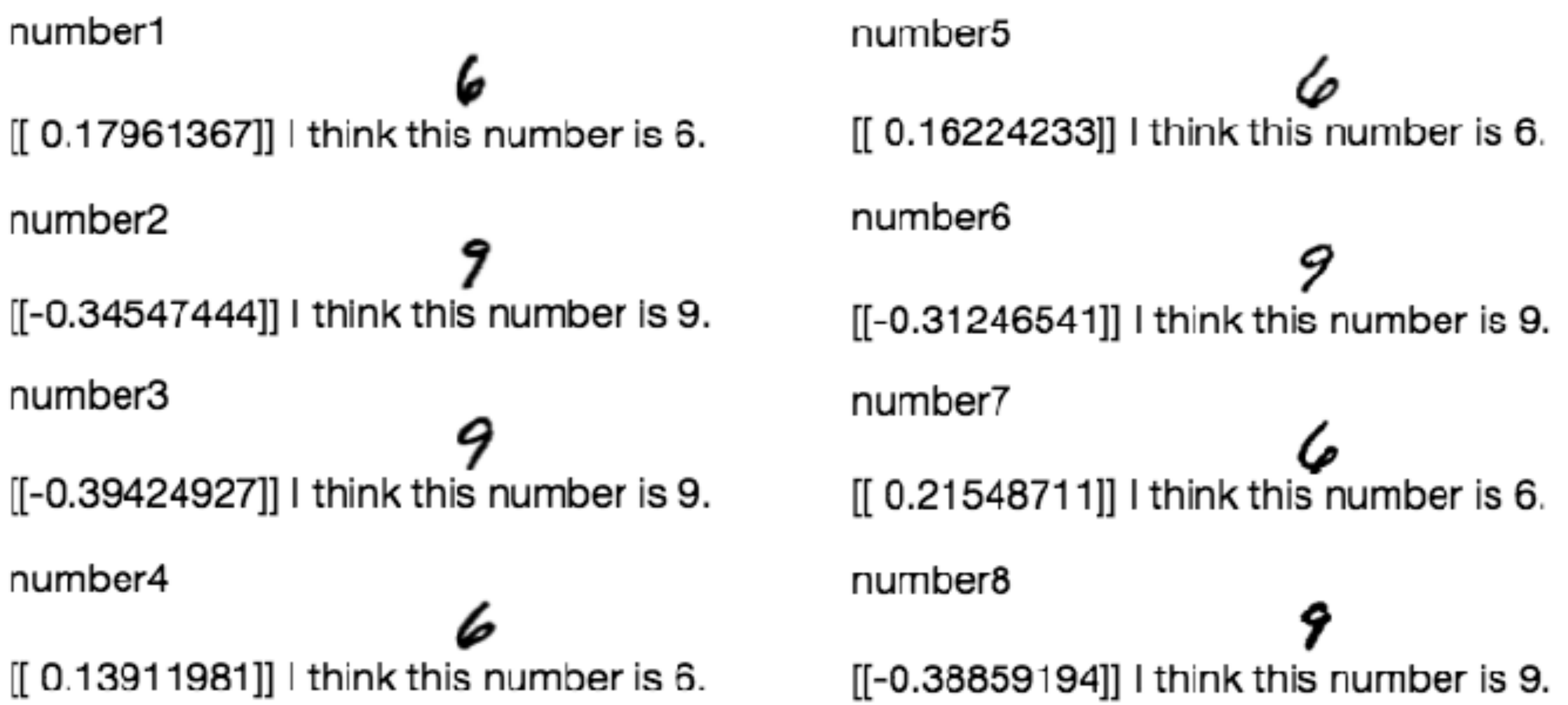
Encode the training data and test data into quantum states using rotation around Y-axis about $\theta_i = \text{arccot}[(x_i)_1/(x_i)_2], i = 0, 1, 2$.

$$R_y^\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \end{pmatrix} = \begin{pmatrix} (x_i)_1 \\ (x_i)_2 \end{pmatrix}$$

Li, Zhaokai, et al. "Experimental realization of a quantum support vector machine." Physical review letters 114.14 (2015): 140504.

# Characterization of 6 an 9

Demonstration of Results:

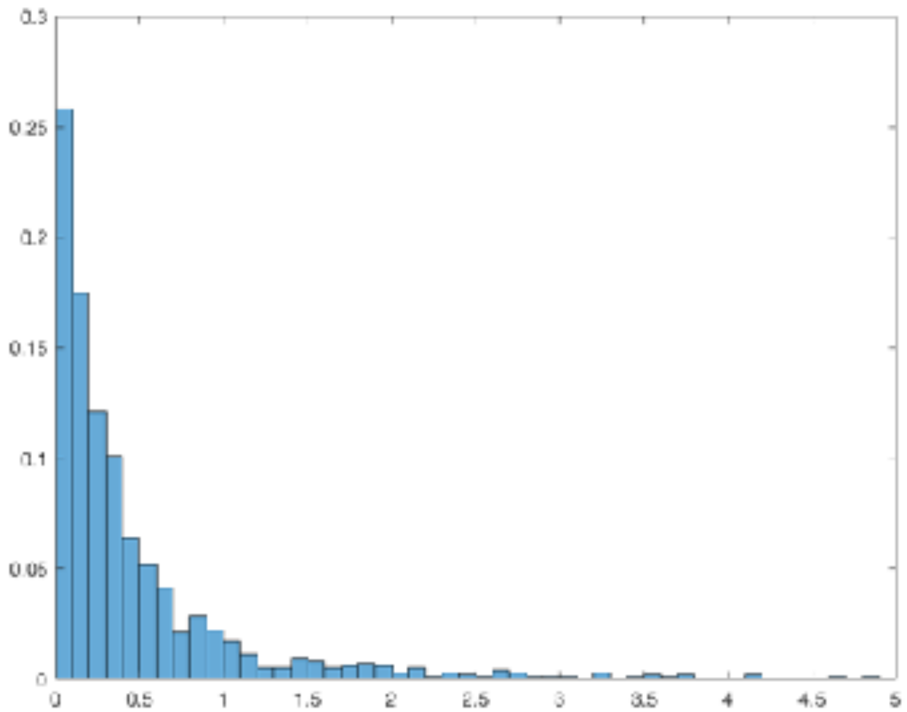


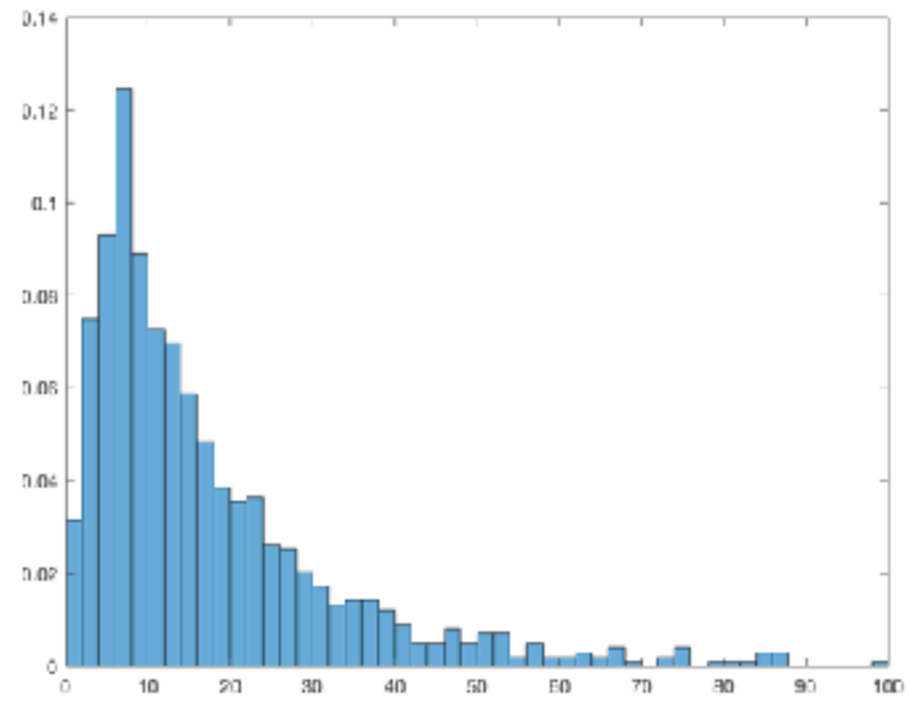

If the number is positive, the machine recognizes it as 6, otherwise 9.

The first quantum machine learning program targeting computer vision task, at least to our knowledge.

https://github.com/JinlongHuang/quantum-SVM

# Quantum Supremacy: Boson Sampling




Boson Sampling 16 photons

Glynn’s Algorithm

Gaussian Matrix 20x20

Gurvits’s Algorithm

Thank You.

Q & A?