Quantum Compiling

CPSC 4470/5470

Introduction to Quantum Computing

Instructor: Prof. **Yongshan Ding**

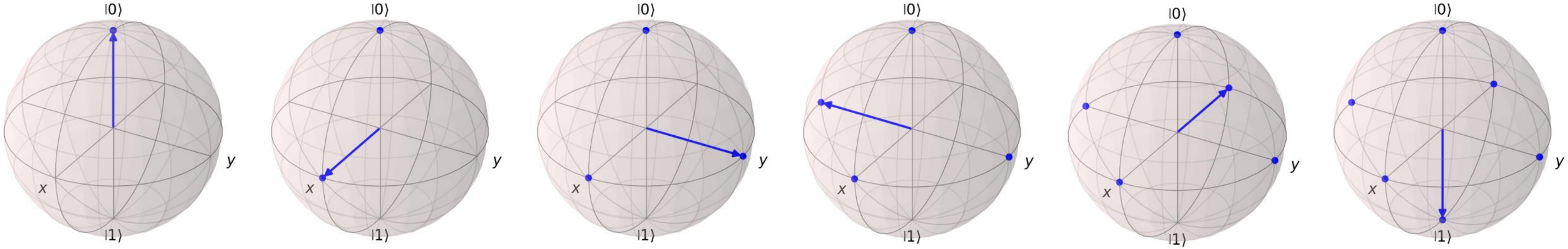
Computer Science, Applied Physics, Yale Quantum Institute

A Tour on the Bloch Sphere

Carlton M. Caves: *“Hilbert space is a big space.”*

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

$|0\rangle$ $\quad SH|0\rangle \quad S^{\dagger}HSH|0\rangle$

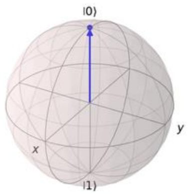


$H|0\rangle \quad HSH|0\rangle \quad HS^{\dagger}HSH|0\rangle$

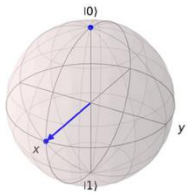
Performing gate sequence: $H, S, H, S^{\dagger}, H$ from the $|0\rangle$ state.

We **cannot** get to an arbitrary point on the Bloch sphere from $|0\rangle$ using only H and S gates.

H and S Gates: Changing Principal Axes

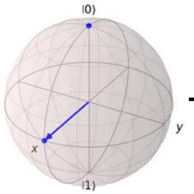


$\xrightarrow{H}$

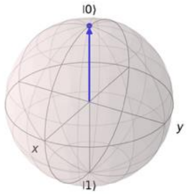


$HZH = X$

Z axis X axis

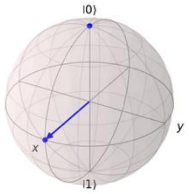


$\xrightarrow{H}$

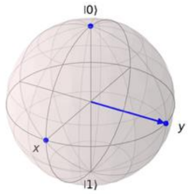


$HXH = Z$

X axis Z axis

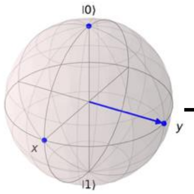


$\xrightarrow{S}$

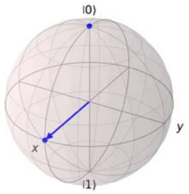


$SXS^\dagger = Y$

X axis Y axis

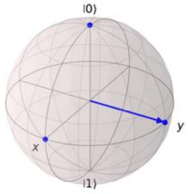


$\xrightarrow{S^\dagger}$

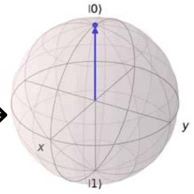


$S^\dagger Y S = X$

Y axis X axis

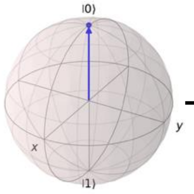


$\xrightarrow{HS^\dagger}$

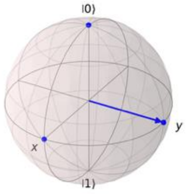


$HS^\dagger Y \left(HS^\dagger\right)^\dagger = Z$

Y axis Z axis



$\xrightarrow{SH}$



$SHZHS^\dagger = Y$

Z axis Y axis

Clifford Gates

Definition (Clifford group):

On a single qubit, the Clifford group is the set of unitaries that maps Pauli's to Pauli's under conjugation.
A unitary U is a Clifford gate if:

$$UPU^{\dagger} \in \mathcal{P}$$

for every $P \in \mathcal{P}$, where $\mathcal{P} = \{\mathrm{I}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$ (up to global phase $\pm 1, \pm i$).

Generator: H, S. (Any single-qubit Clifford operation can be expressed as a sequence of H, S.)

Definition (n-qubit Clifford group):

"Pauli string": $X \otimes I \otimes Y$

On n qubits, the Clifford group is the set of unitaries that maps Pauli string to Pauli string under conjugation.
A unitary U is a Clifford gate if:

$$UPU^{\dagger} \in \mathcal{P}_n$$

for every $P \in \mathcal{P}_n$, where $\mathcal{P}_n = \{\mathrm{I}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\}^{\otimes n}$ (up to global phase $\pm 1, \pm i$).

Generator: H, S, CNOT. (Any n-qubit Clifford operation can be expressed as a sequence of H, S, CNOT.)

What about Arbitrary Gates?

Single-Qubit Unitary $U \in \mathbb{C}^{2\times 2}$:

$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix} = \begin{bmatrix} e^{i\alpha}\cos\theta & e^{i\beta}\sin\theta \\ -e^{i\beta}\sin\theta & -e^{-i\alpha}\cos\theta \end{bmatrix}$$

3 independent real numbers

n-Qubit Unitary $U \in \mathbb{C}^{2^n \times 2^n}$:

- Total degrees of freedom: $2^{2n} - 1$ independent real numbers.

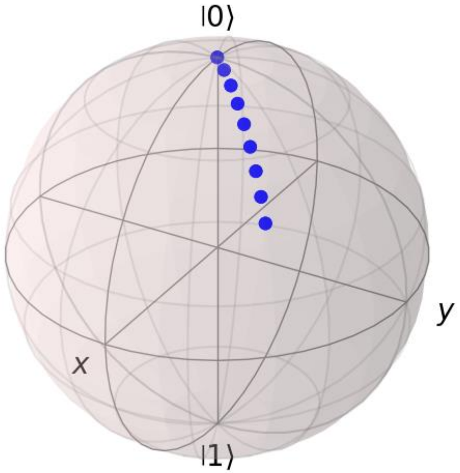
In practice: rather than realizing a unitary *exactly*,
we usually ask whether we can *approximate* it to within error tolerance ϵ.

Universal for Single-Qubit Unitaries: $R_x(\theta), R_z(\theta)$ gates

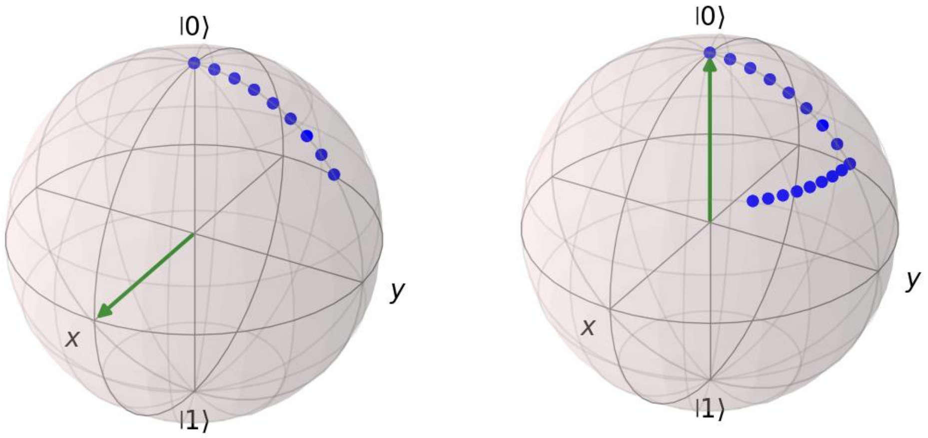
For single qubit, $R_x(\theta), R_z(\theta)$ gates are "**universal**":

- **State preparation**: Any single-qubit state can be prepared by a sequence of $R_x(\theta), R_z(\theta)$ gates from $|0\rangle$.
- **Unitary synthesis**: Any single-qubit unitary can be synthesized by a sequence of $R_x(\theta), R_z(\theta)$ gates.

Let's consider the *state* version: How do we get to an **arbitrary point on Bloch sphere** from $|0\rangle$?



Geodesic path from $|0\rangle$.

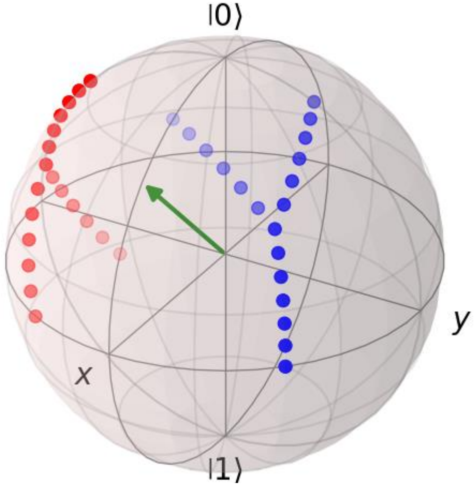


$|0\rangle$ — $\boxed{R_x(\alpha)}$ — $\boxed{R_z(\beta)}$ —

$R_z(\beta)\ R_x(\alpha)$

Universal for Single-Qubit Unitaries: $R_x(\theta)$, $R_z(\theta)$ gates

Now consider the *unitary* version: How do we get an **arbitrary rotation** by a sequence of $R_x(\theta), R_z(\theta)$ gates?

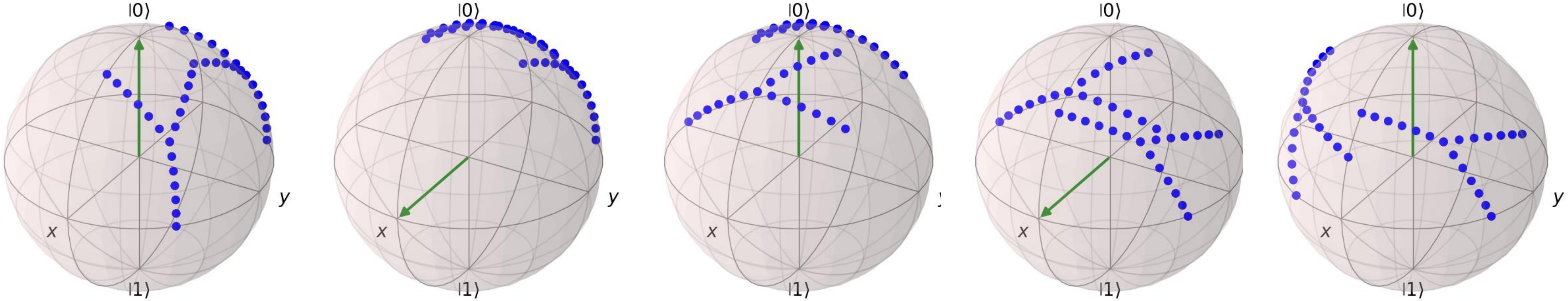


Example: How to implement a rotation gate $R_{\hat{n}}(\pi)$ along the axis $\hat{n} = \frac{\hat{x}+\hat{z}}{\sqrt{2}}$?

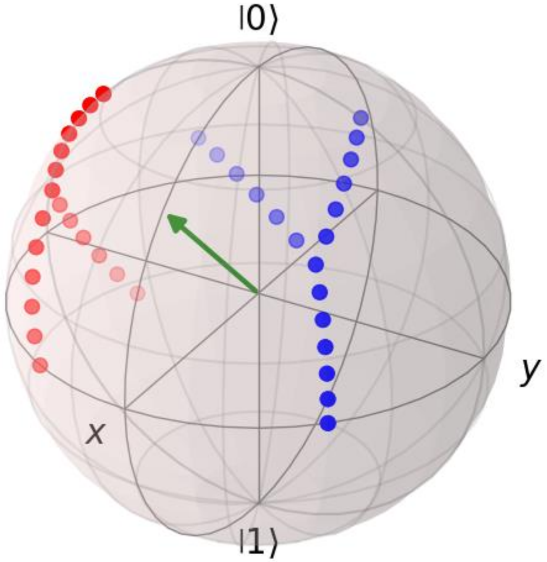
$|\psi\rangle$ — $\boxed{R_z}$ — $\boxed{R_x}$ — $\boxed{R_z}$ — $\boxed{R_x}$ — $\boxed{R_z}$ —

$$R_z(-\alpha)\ R_x(-\beta)\ R_z(\gamma) R_z(\beta)\ R_z(\alpha)$$

$$R_{\hat{n}}(\pi) = R_z\left(-\frac{\pi}{2}\right) R_x\left(-\frac{\pi}{4}\right) R_z(\pi) R_x\left(\frac{\pi}{4}\right) R_z\left(\frac{\pi}{2}\right)$$



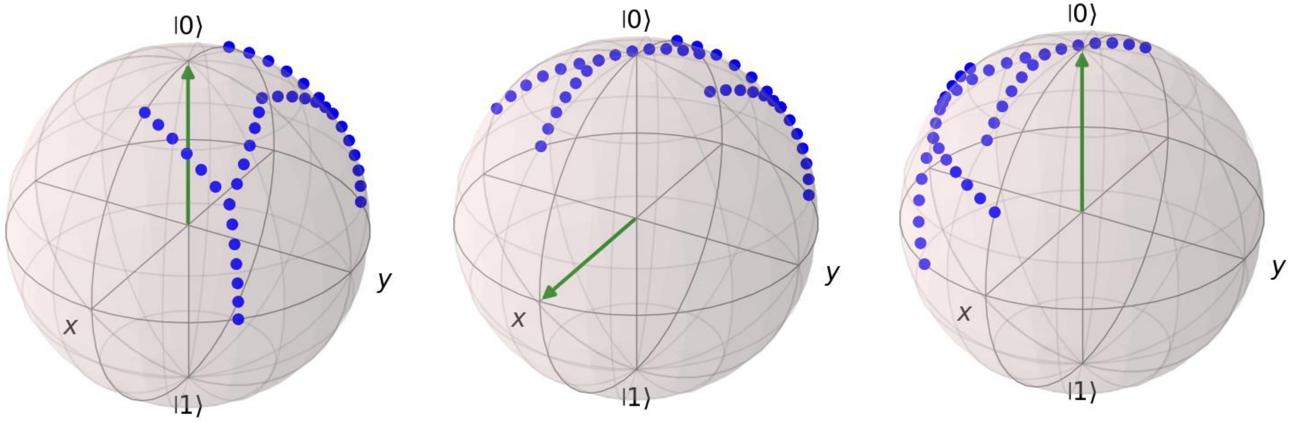
Universal for Single-Qubit Unitaries: $R_x(\theta), R_z(\theta)$ gates



For single qubit, *three* rotations are enough.

$|\psi\rangle$ — $\boxed{R_z(\alpha)}$ — $\boxed{R_x(\beta)}$ — $\boxed{R_z(\gamma)}$ — $R_z(\gamma)\ R_x(\beta)\ R_z(\alpha)$

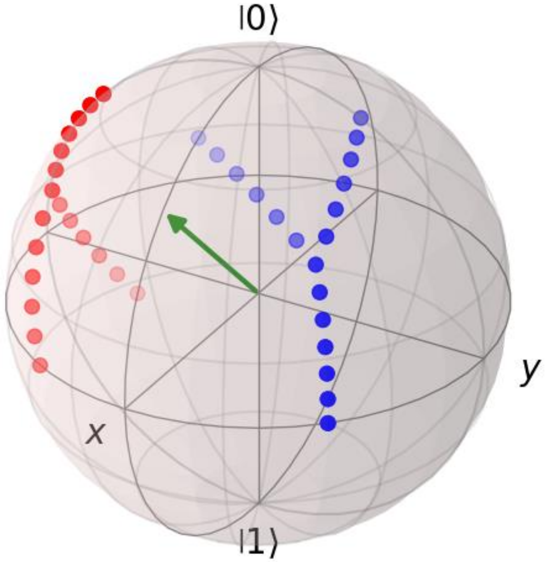
Example: How to implement Hadamard gate $R_{\hat{n}}(\pi), \hat{n} = \frac{\hat{x}+\hat{z}}{\sqrt{2}}$?



$$R_{\hat{n}}(\pi) = R_z\left(\frac{\pi}{2}\right) R_x\left(\frac{\pi}{2}\right) R_z\left(\frac{\pi}{2}\right)$$

Is Hadamard also a Rotation?

Example: $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{X+Z}{\sqrt{2}}$



Recall from lecture 5:
H is a **reflection** about $|h\rangle = \cos\pi/8\,|0\rangle + \sin\pi/8\,|1\rangle$

$$2|h\rangle\langle h| - I = H$$

In fact, H is also a **rotation** about axis $|h\rangle$ by 180°: $R_{\hat{n}}(\pi)$

Rotation axis: $\hat{n} = \frac{\hat{x}+\hat{z}}{\sqrt{2}}$

Rotation matrix (derive on board):

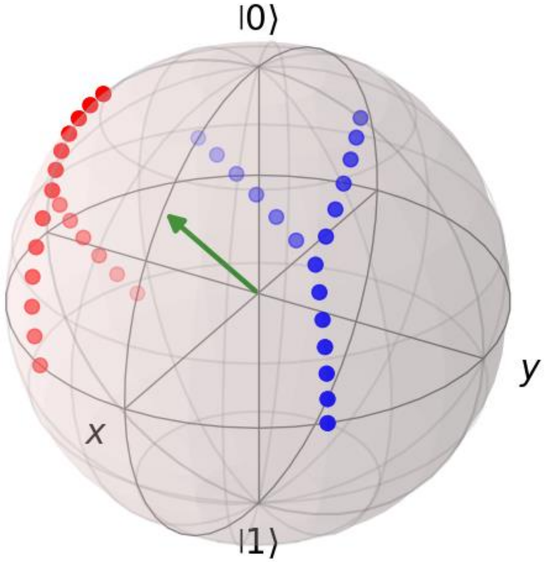
$$R_{\hat{n}}(\pi) = e^{-\frac{i\pi}{2}\left(\frac{X+Z}{\sqrt{2}}\right)} = -iH$$ (Up to global phase.)

In practice, H gate is often an operation that is natively supported by hardware.

Implementing H gate

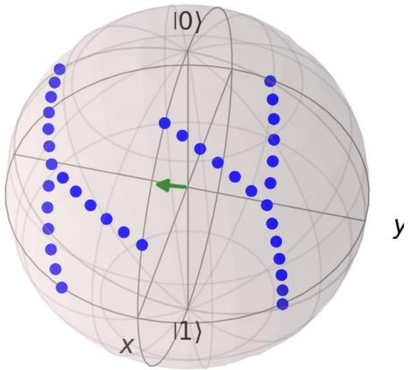
Also works: $H = R_y\left(\frac{\pi}{4}\right) Z R_y\left(-\frac{\pi}{4}\right)$

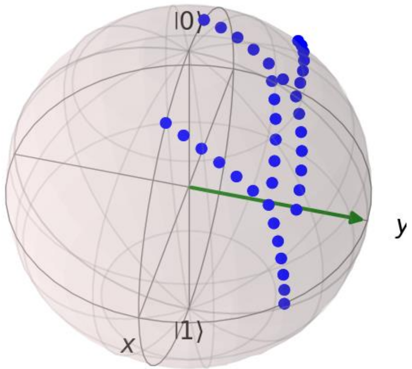
$|\psi\rangle$ —$\boxed{R_y(\alpha)}$—$\boxed{R_z(\beta)}$—$\boxed{R_y(\gamma)}$— $R_y(\gamma)\, R_z(\beta)\, R_y(\alpha)$

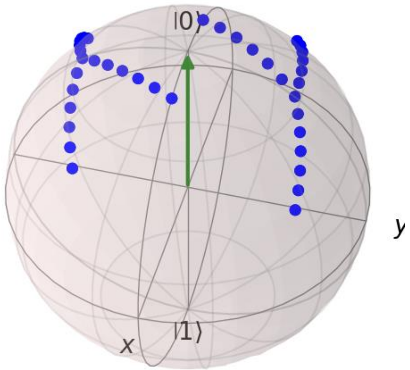


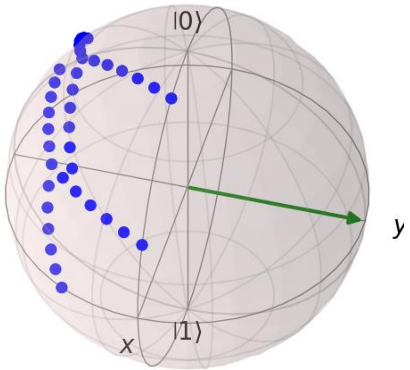
Example: Implementing Hadamard gate $R_{\hat{n}}(\pi), \hat{n} = \frac{\hat{x}+\hat{z}}{\sqrt{2}}$

Camera view change









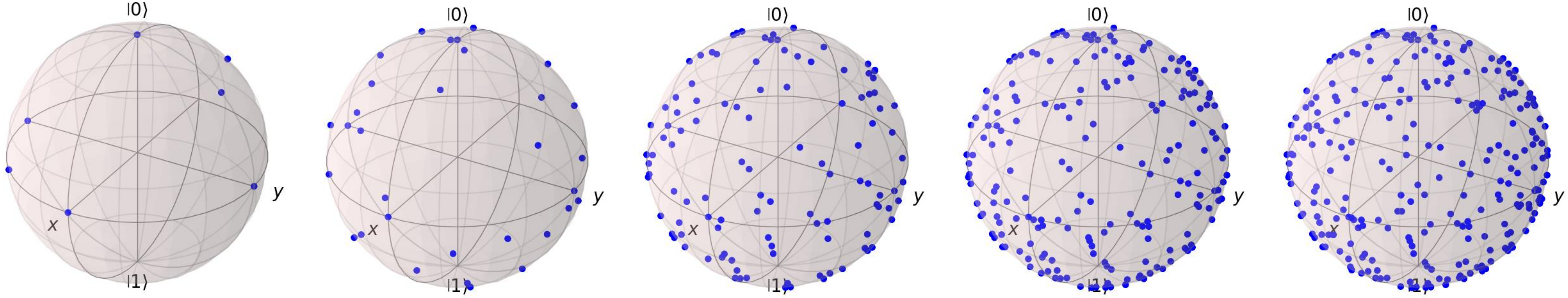
Universal for Single-Qubit Unitaries: H, T gates

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix}$$

For a single qubit, H, T gates are universal!

Synthesizing arbitrary z rotations using H, S, and T:

- Solovay-Kitaev: $\#T = O(\log^c(1/\epsilon)), c \approx 3$
- Ross-Selinger: $\#T = O(\log(1/\epsilon) + \log\log(1/\epsilon))$



After H,S,H,T,H,T gates.

Gate sequence: H,S,H,T,H,T, (states in Fig (a)) H,T,H,T,H,T,H,T,H,T,S,H,T,H,T,H,T,H,T,S,H,T,H,T,H, (states in Fig (b)) T,H,T,H,T,S,H,T,S,H,T,H,T,H,T,H,T,H,T,S,H,T,H,T,H,T,S,H,T,S,H,T,H,T,S,H,T,S,H,T,S,H,T,H,T,H,T,H,T,S,H,T,H,T,H,T,H,T,H,T,S,H,T,S,H,T,H,T,S,H,T,H,T,S,H,T,S,H, (states in Fig (c)) T,S,H,T,S,H,T,H,T,S,H,T,S,H,T,S,H,T,S,H,T,H,T,H,T,S,H,T,S,H,T,S,H,T,H,T,H,T,H,T,S,H,T,H,T,S,H,T,H,T,H,T,S,H,T,H,T,H,T,H,T,S,H,T,H,T,S,H,T,H,T,S,H,T,S,H,T,S,H,T,H,T, (states in Fig (d)) H,T,H,T,H,T,H,T,H,T,S,H,T,H,T,S,H,T,H,T,H,T,S,H,T,S,H,T,H,T,H,T,S,H,T,S,H,T,S,H,T,H,T,S,H,T,H,T,H,T,H,T,S,H,T,S,H,T,S,H,S,S,S (states in Fig (e))