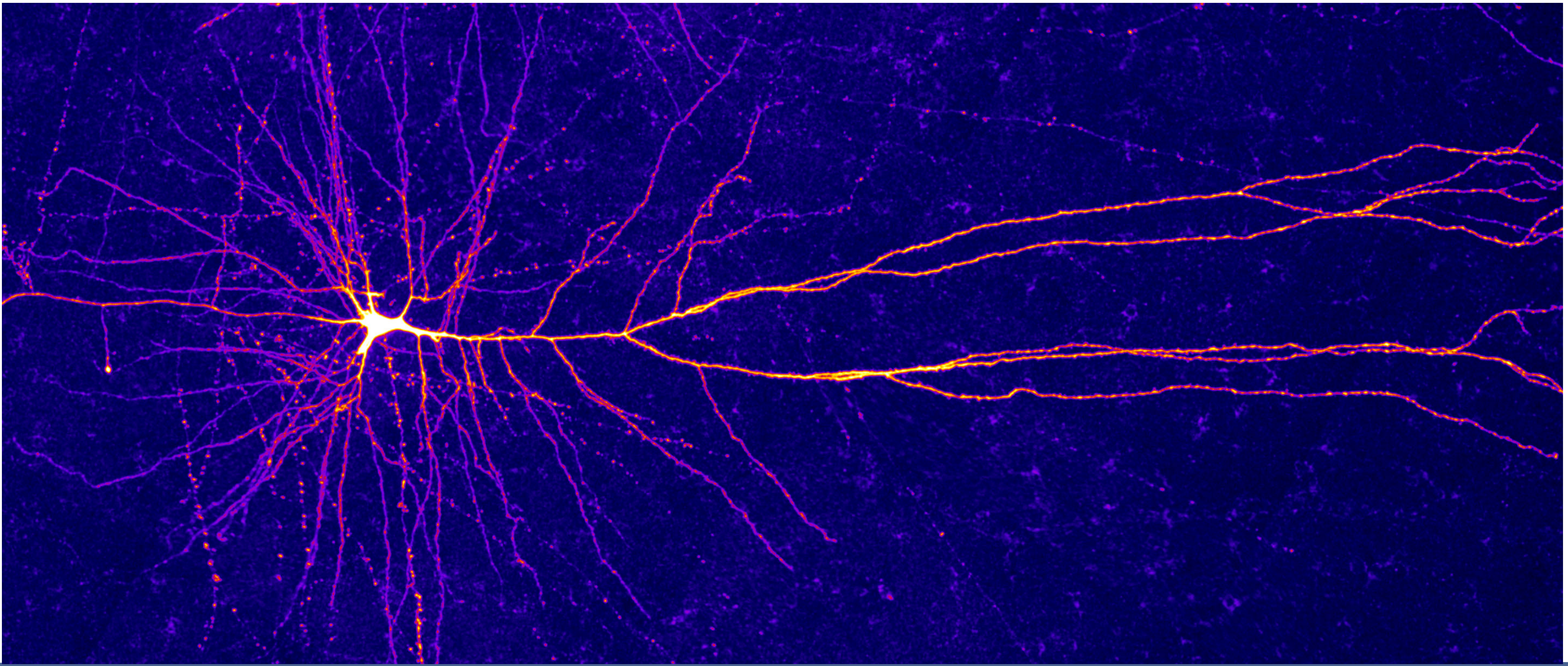
Seminar: Reinforcement Learning Quantum Error Correction

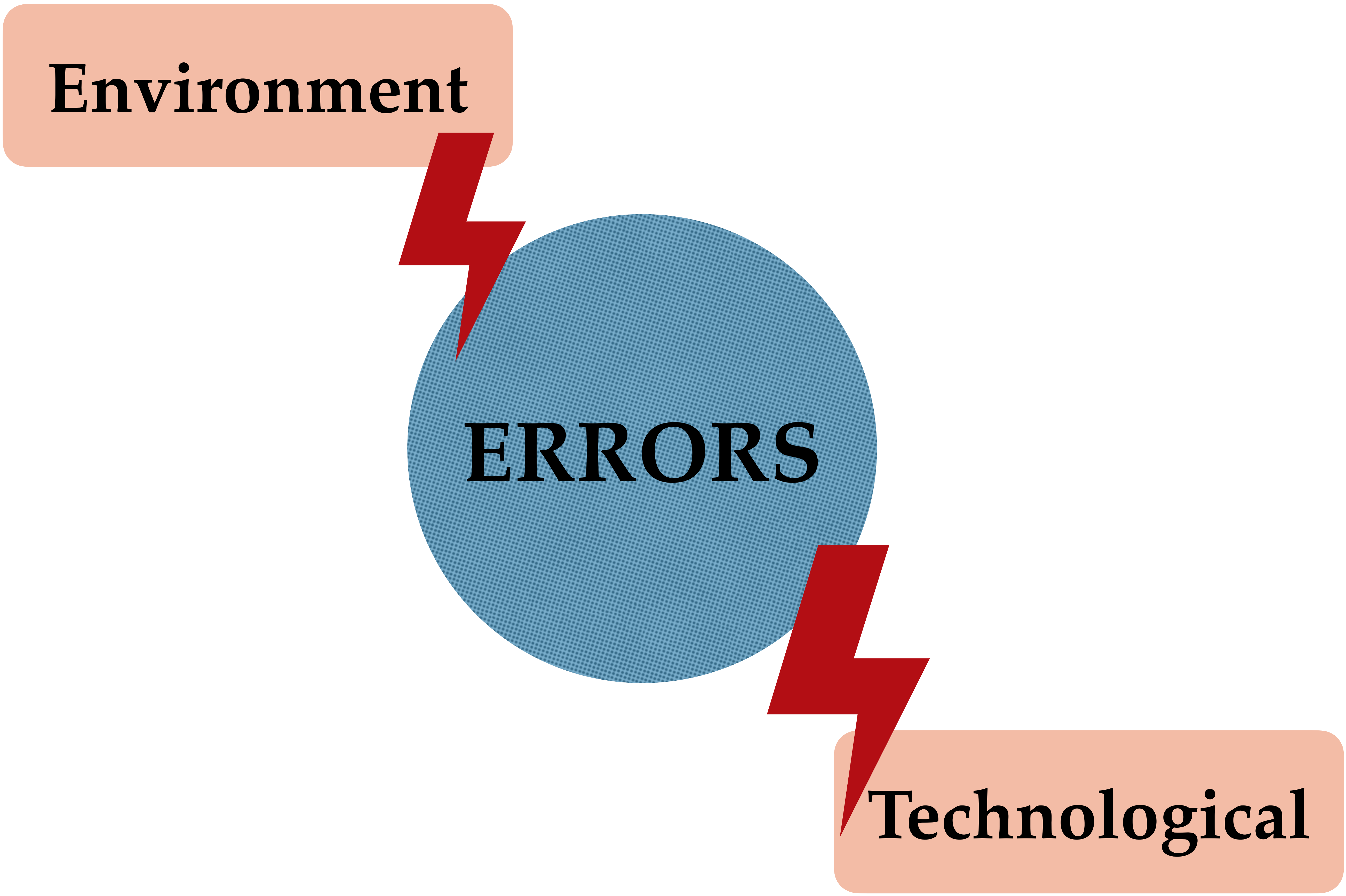


Sakshi Pahujani
Supervised by: Kai Meinerz
University of Cologne

Contents

Quantum error correction

Sources of error and why we need correction



- Huge registers
- Multiple gates
- High Speed >> Faster error propagation

Classical error correction

- N-BIT REPETITION CODE

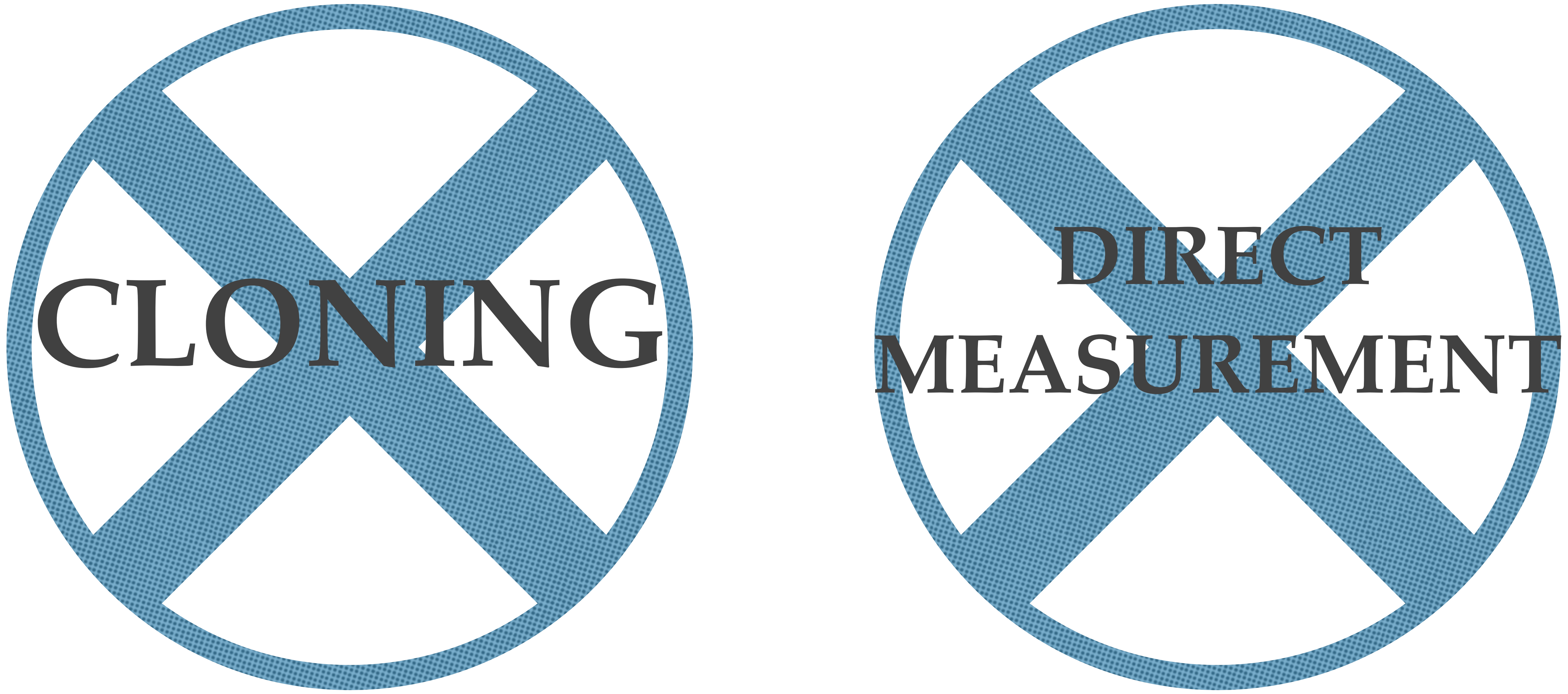
Exploiting redundancy

0 → 000 → 001

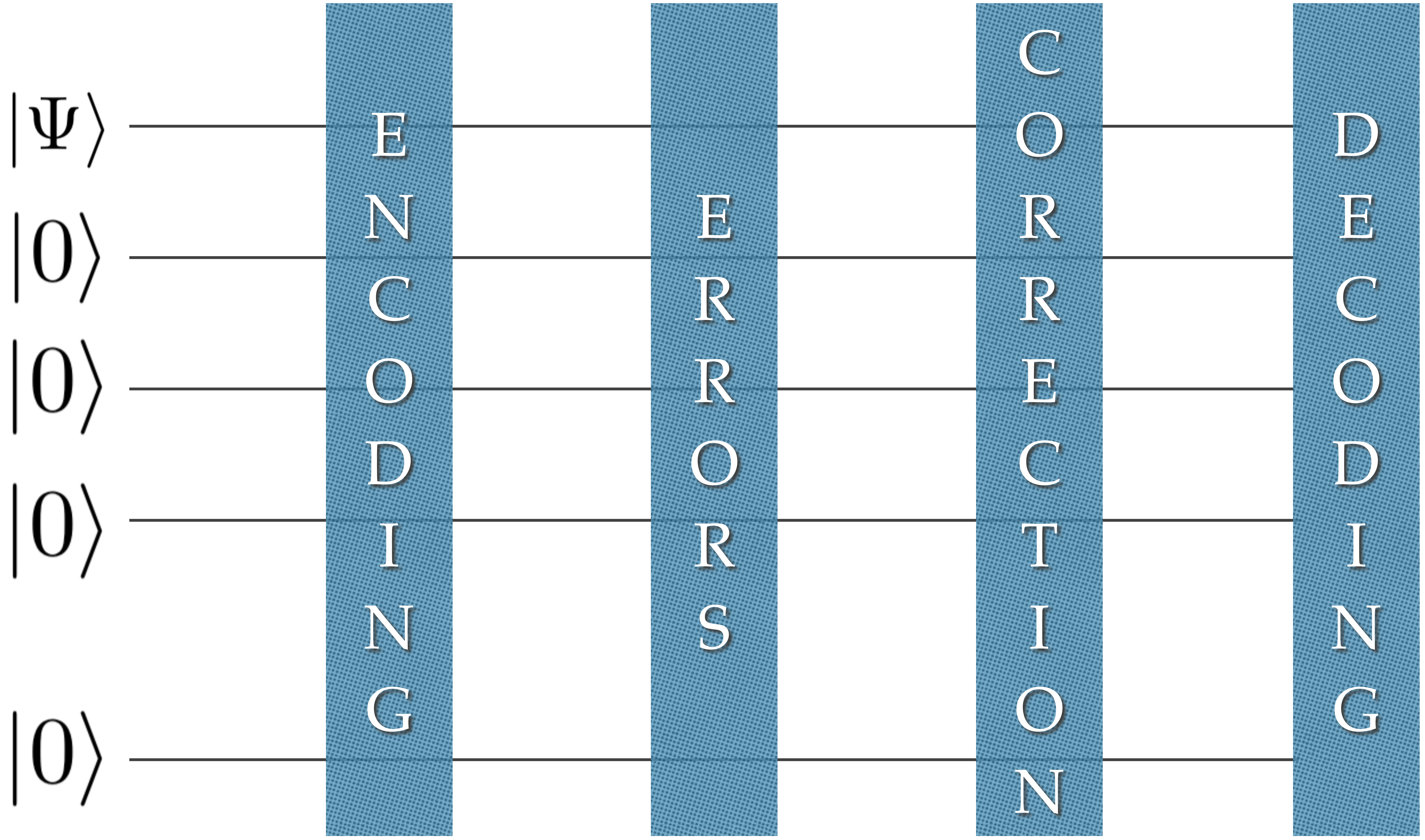
1 → 111 → 110

Error detection by majority rule

Extension to QEC…?!

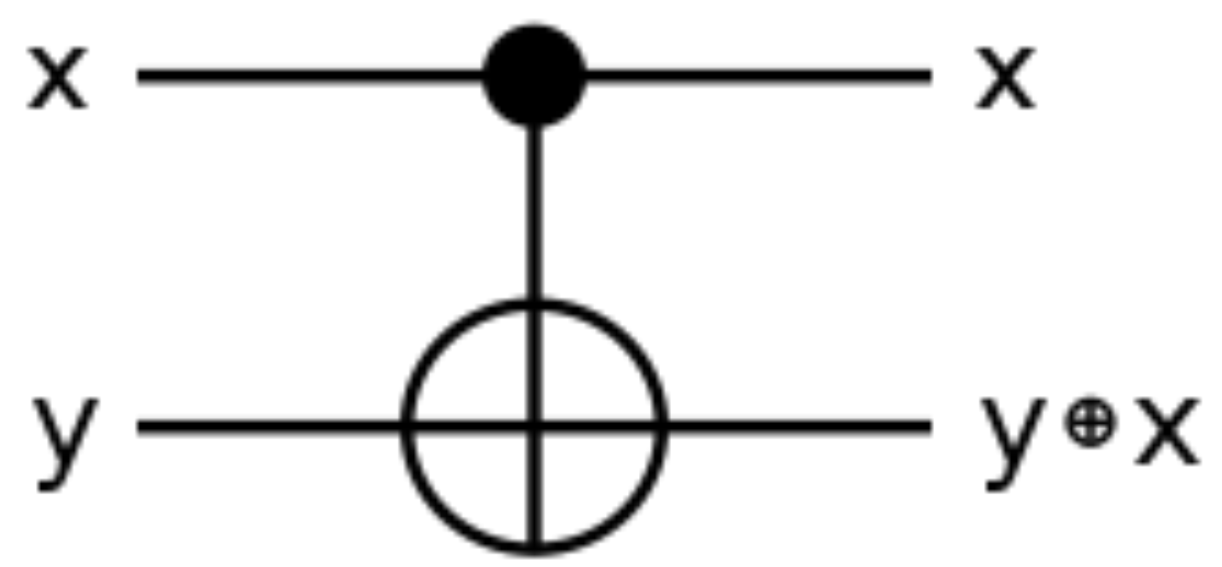


Conceptual setting of a QEC code



CNOT gate

CNOT gate



input		output	
x	y	x	y+x
$\lvert 0\rangle$	$\lvert 0\rangle$	$\lvert 0\rangle$	$\lvert 0\rangle$
$\lvert 0\rangle$	$\lvert 1\rangle$	$\lvert 0\rangle$	$\lvert 1\rangle$
$\lvert 1\rangle$	$\lvert 0\rangle$	$\lvert 1\rangle$	$\lvert 1\rangle$
$\lvert 1\rangle$	$\lvert 1\rangle$	$\lvert 1\rangle$	$\lvert 0\rangle$

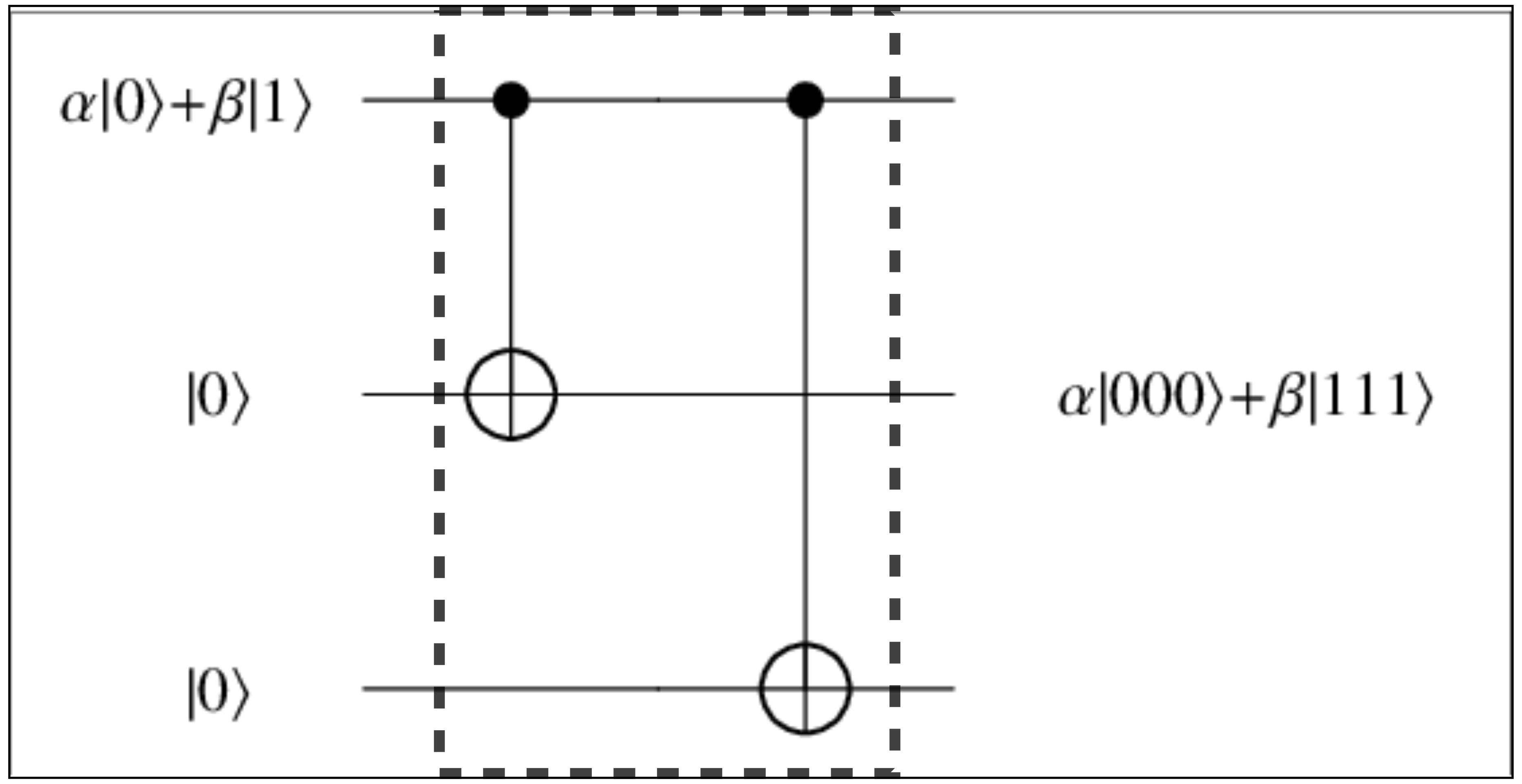
$$\mathbb{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \text{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

3-Qubit bit flip code

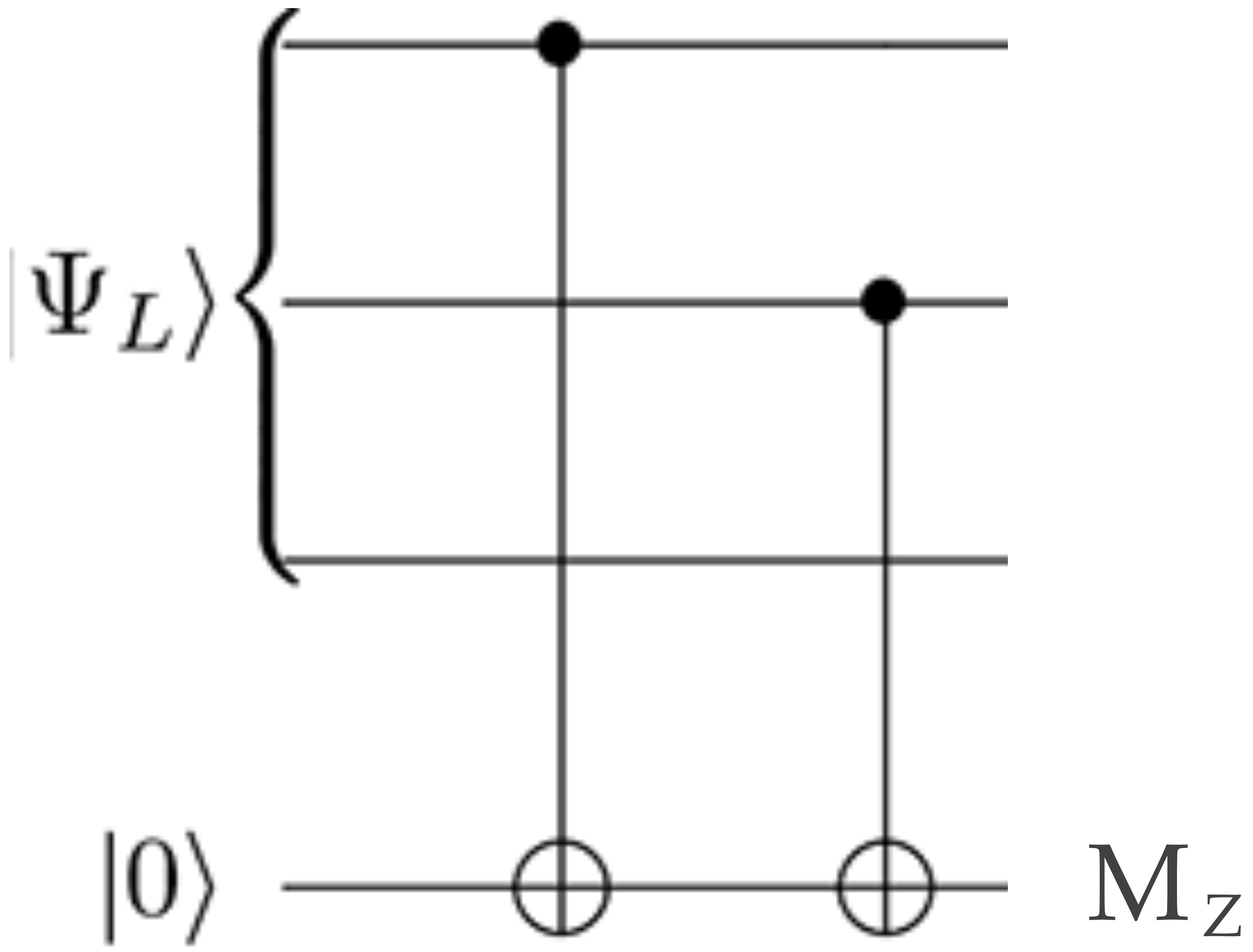
- Idea : Distribute logical information (an arbitrary quantum state) over entangled state of three qubits.

Encoding circuit



$$(\alpha|0\rangle+\beta|1\rangle)|0\rangle|0\rangle \xrightarrow{C_1NOT_2} (\alpha|00\rangle+\beta|11\rangle)|0\rangle \xrightarrow{C_1NOT_3} (\alpha|000\rangle+\beta|111\rangle) \tag{1}$$

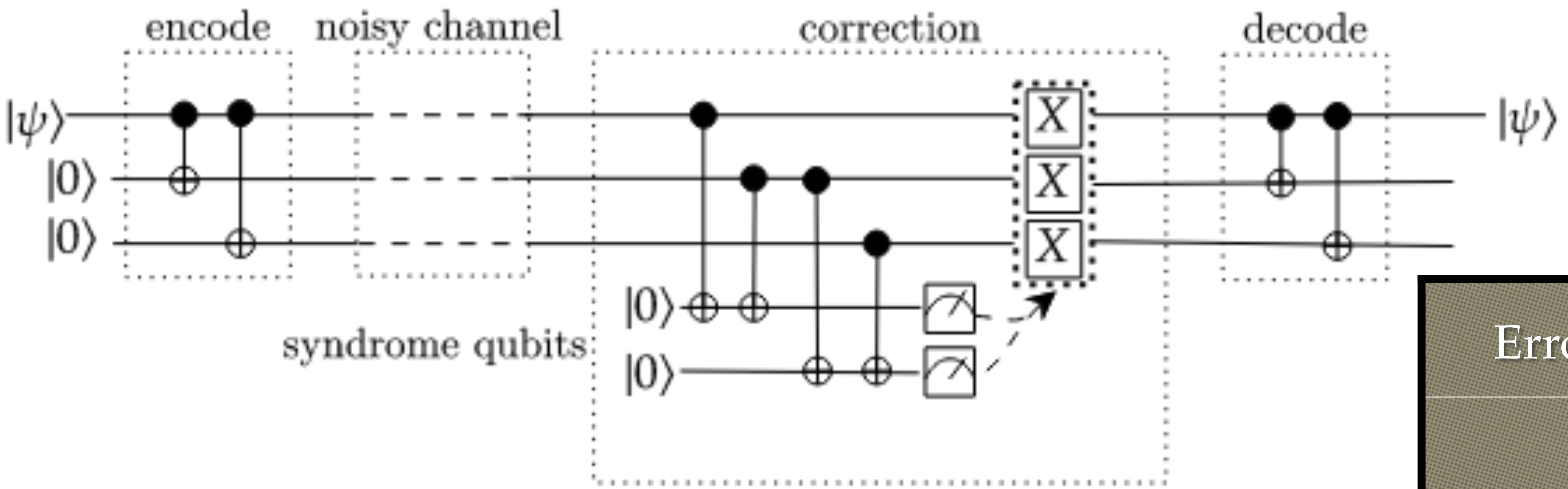
Indirect measurement



Input state	Ancillar	Measurement
$\|00\rangle$	$\|0\rangle$	1
$\|01\rangle$	$\|1\rangle$	-1
$\|10\rangle$	$\|1\rangle$	-1
$\|11\rangle$	$\|0\rangle$	1

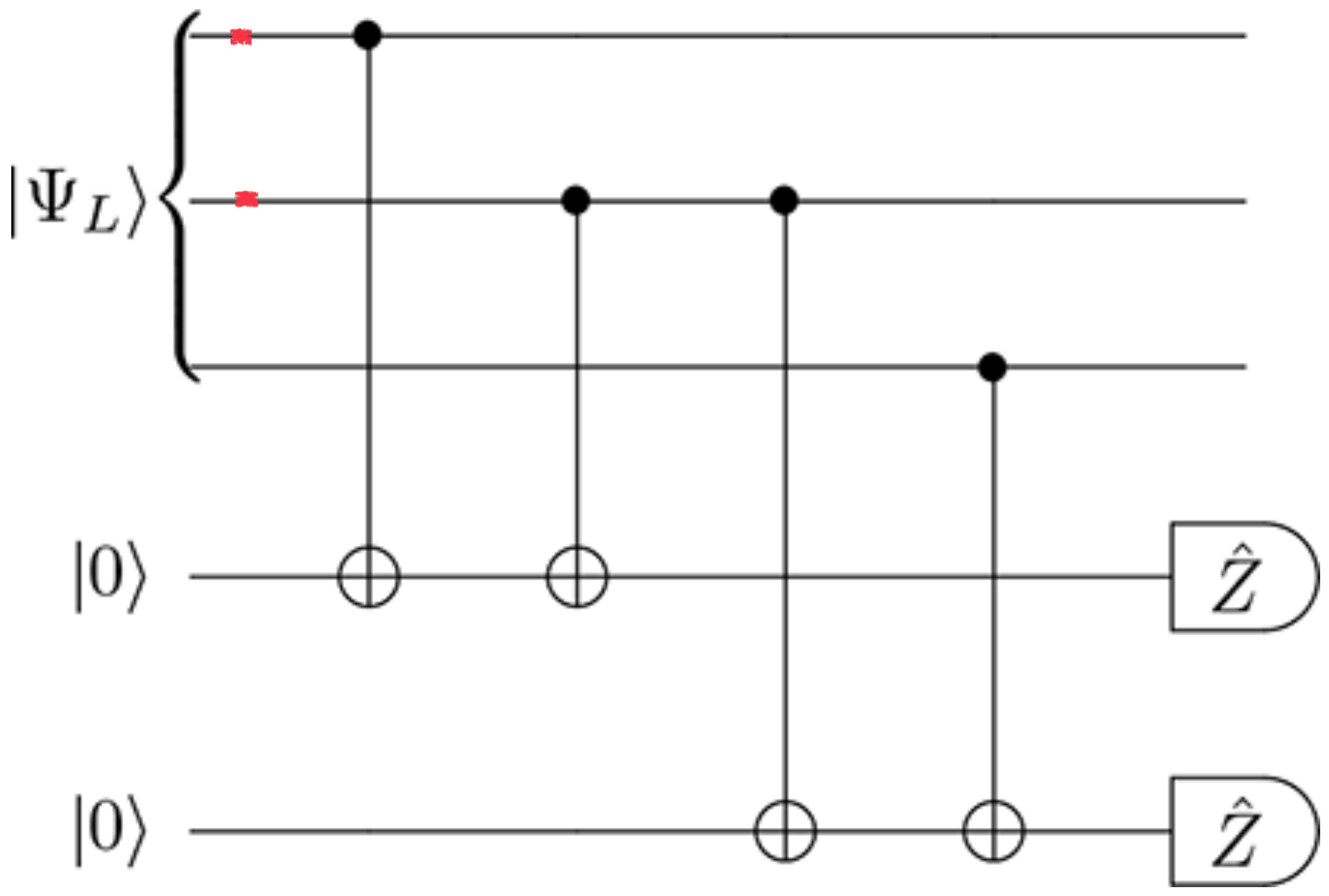
- 1: Even parity —> parallel orientation
- -1: Odd parity —> anti-parallel orientation

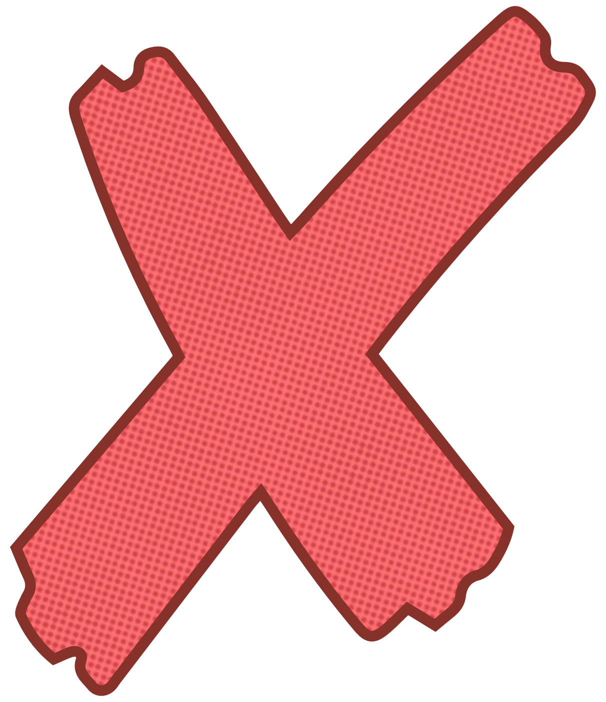
Correction



Error	Syndrome		Correction
	$M_Z^{(1)} = Z_1 Z_2$	$M_Z^{(2)} = Z_1 Z_2$	
1	1	1	1
X_1	-1	1	X_1
X_2	-1	-1	X_2
X_3	1	-1	X_3

Fails for more than one error!
Fails for phase flip errors!

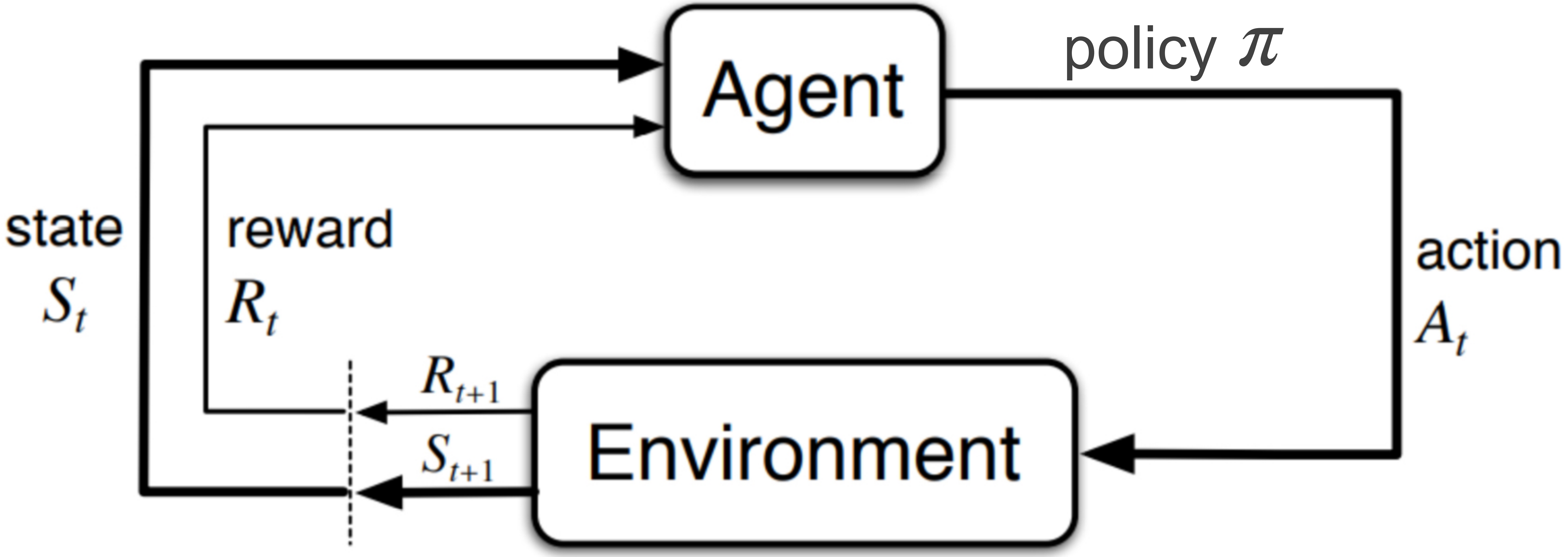




More sophisticated QEC codes: 9-bit Shor's code, Toric code etc.

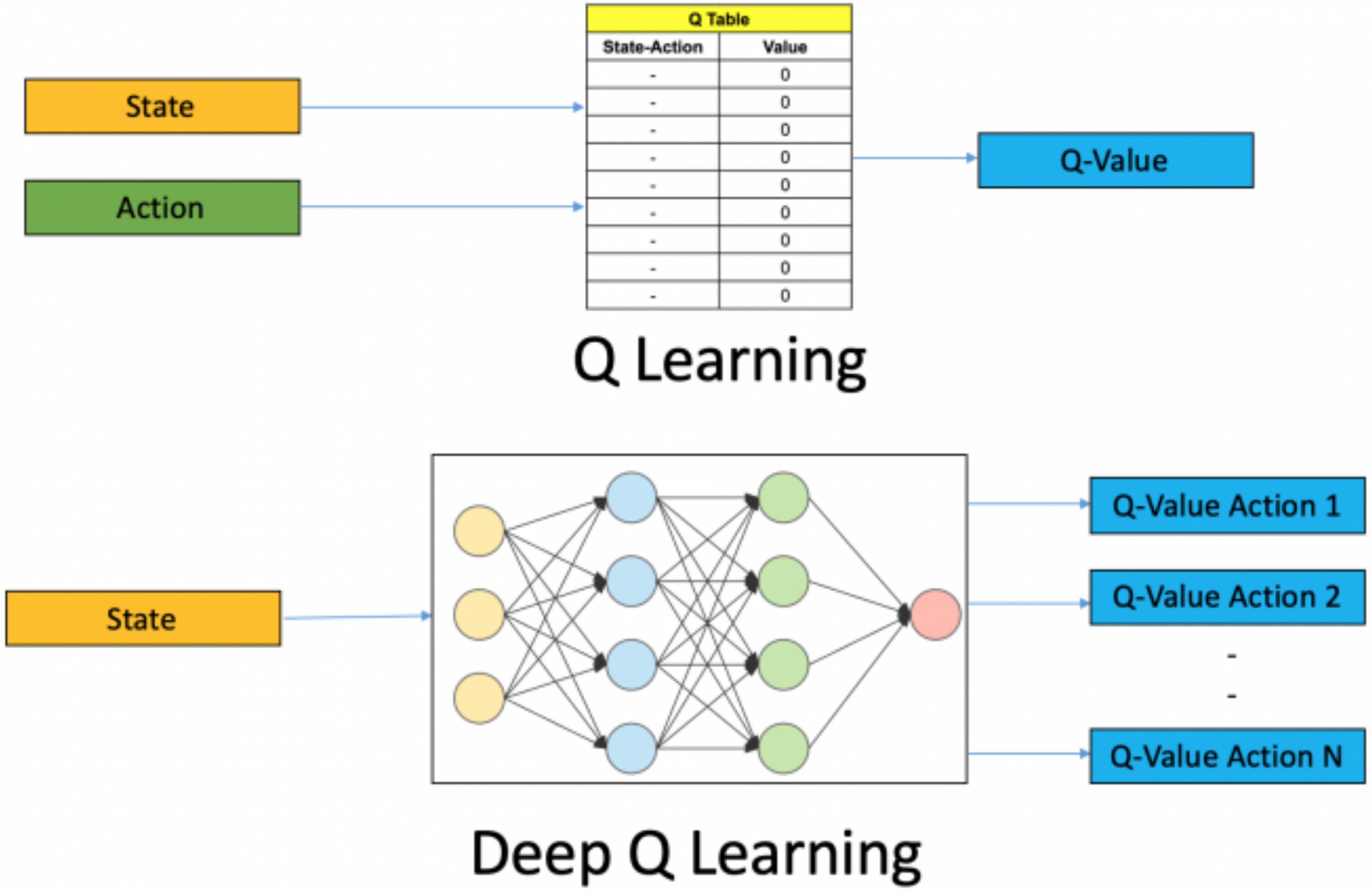
Reinforcement Learning

Setting



Agent
policy π
state
S_t
reward
R_t
action
A_t
R_{t+1}
S_{t+1}
Environment

Deep Q



- Neural network used to approximate the Q-value function
- The state is given as input and the Q-value of all possible actions is generated as output

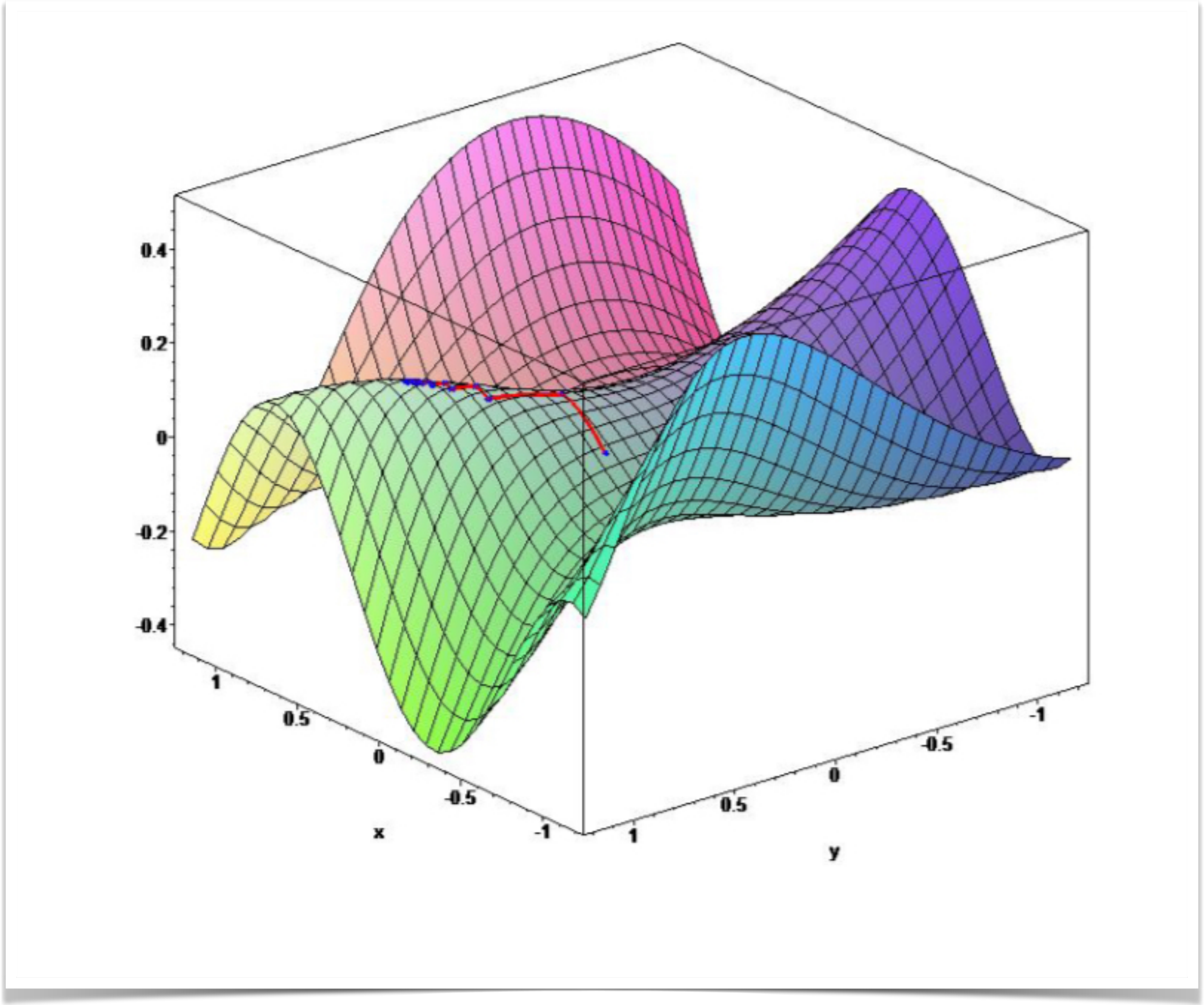
Policy Gradient

$$\delta\theta = \alpha \nabla_\theta J(\theta)$$

$$\nabla_\theta \pi_\theta(s,a) = \pi_\theta(s,a)\frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)}$$

$$= \pi_\theta(s,a)\,\nabla_\theta log \pi_\theta(s,a)$$

- θ : networks weights and biases.
- J : Objective function
- α : Learning rate parameter
- π_θ : Policy



Objective Function

- Starting in state s ~ d(s)
- Terminating after one time-step with reward r

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{\pi_\theta}[r] \\
&= \sum_{s \in \mathcal{S}} d(s) \sum_{a \in \mathcal{A}} \pi_\theta(s, a) \mathcal{R}_{s,a} \\
\nabla_\theta J(\theta) &= \sum_{s \in \mathcal{S}} d(s) \sum_{a \in \mathcal{A}} \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a) \mathcal{R}_{s,a} \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) r]
\end{aligned}
$$

- Expected return maximised by applying the policy gradient update rule:

$$\delta\theta = \alpha \nabla_\theta J(\theta)$$

Reinforcement learning with neural networks for quantum feedback : Setup

PHYSICAL REVIEW X **8**, 031084 (2018)

Reinforcement Learning with Neural Networks for Quantum Feedback

Thomas Fösel, Petru Tighineanu, and Talitha Weiss
Max Planck Institute for the Science of Light, Staudtstraße 2, 91058 Erlangen, Germany

Florian Marquardt
Max Planck Institute for the Science of Light, Staudtstraße 2, 91058 Erlangen, Germany
and Physics Department, University of Erlangen-Nuremberg, Staudtstraße 5, 91058 Erlangen, Germany

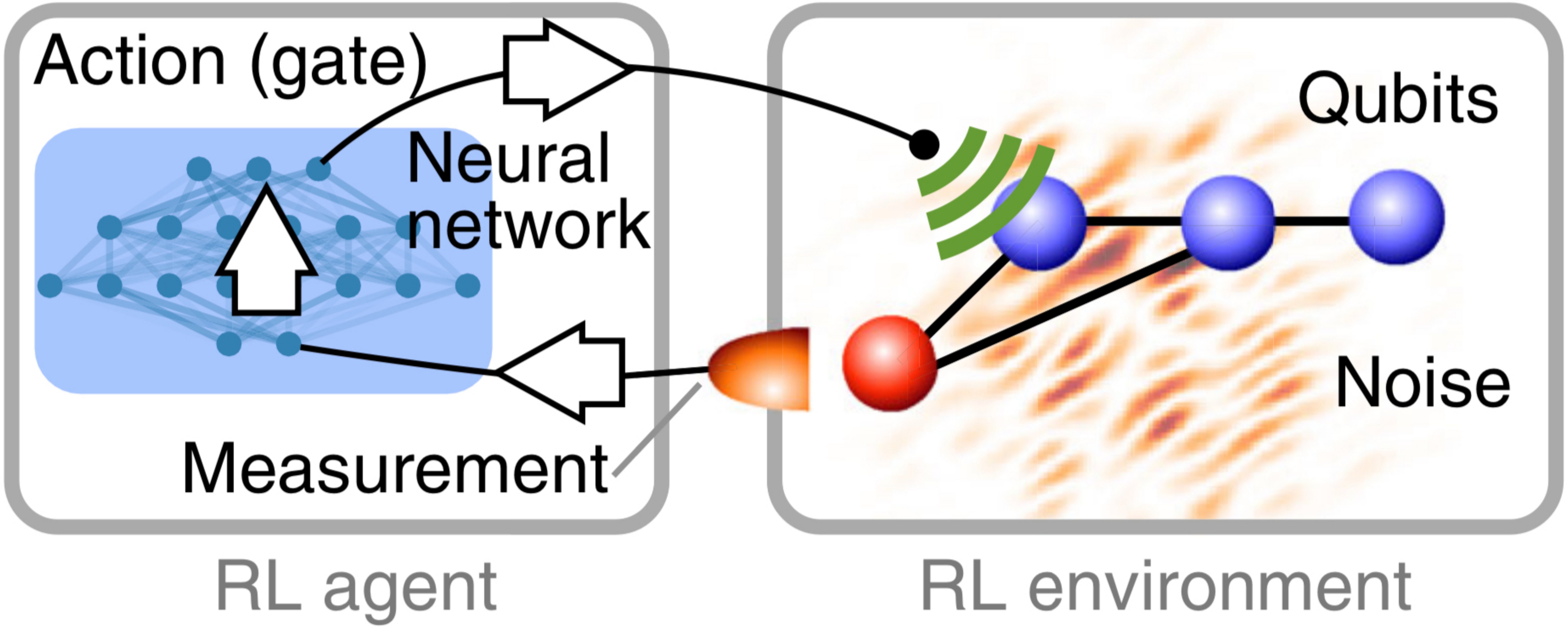
(Received 23 April 2018; revised manuscript received 12 June 2018; published 27 September 2018)

Machine learning with artificial neural networks is revolutionizing science. The most advanced challenges require discovering answers autonomously. In the domain of reinforcement learning, control strategies are improved according to a reward function. The power of neural-network-based reinforcement learning has been highlighted by spectacular recent successes such as playing Go, but its benefits for physics are yet to be demonstrated. Here, we show how a network-based "agent" can discover complete quantum-error-correction strategies, protecting a collection of qubits against noise. These strategies require feedback adapted to measurement outcomes. Finding them from scratch without human guidance and tailored to different hardware resources is a formidable challenge due to the combinatorially large search space. To solve this challenge, we develop two ideas: two-stage learning with teacher and student networks and a reward quantifying the capability to recover the quantum information stored in a multiqubit system. Beyond its immediate impact on quantum computation, our work more generally demonstrates the promise of neural-network-based reinforcement learning in physics.

DOI: 10.1103/PhysRevX.8.031084

Subject Areas: Computational Physics, Interdisciplinary Physics, Quantum Information

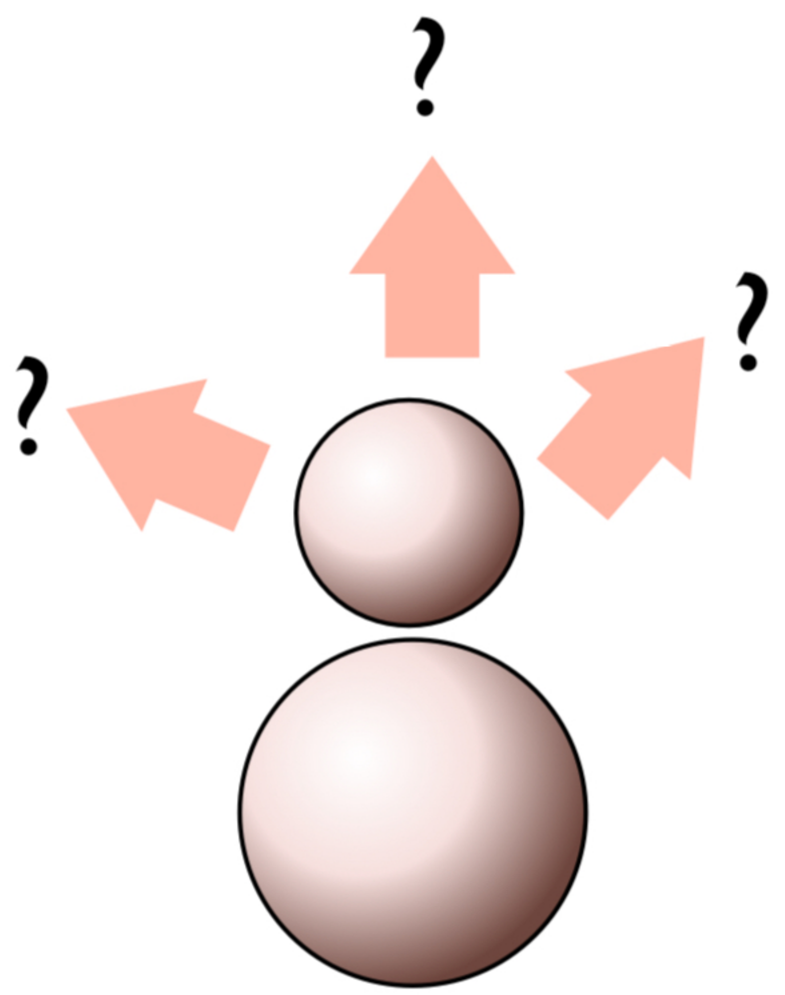
RL setting of the QEC problem



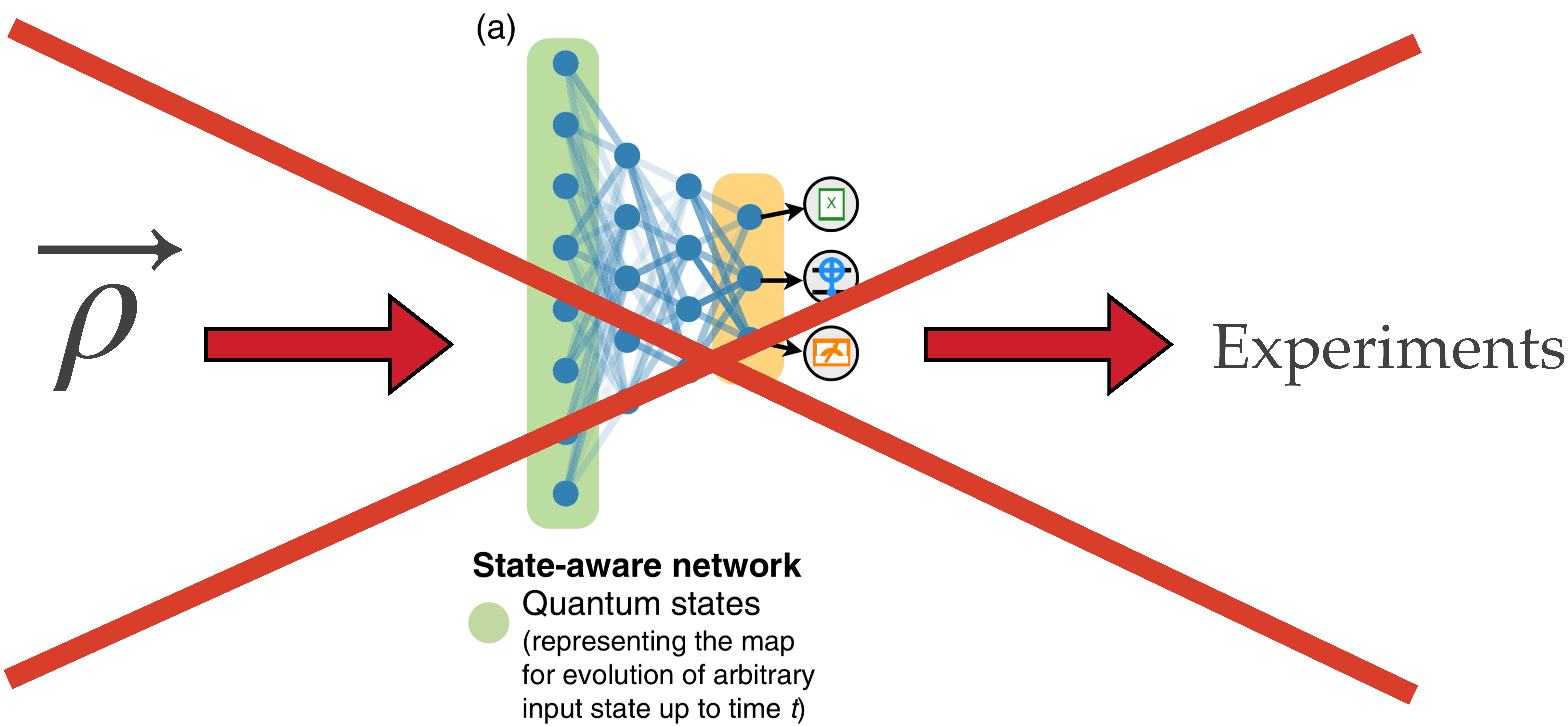
Task: To preserve an arbitrary quantum state initially stored in qubits

Why RL?

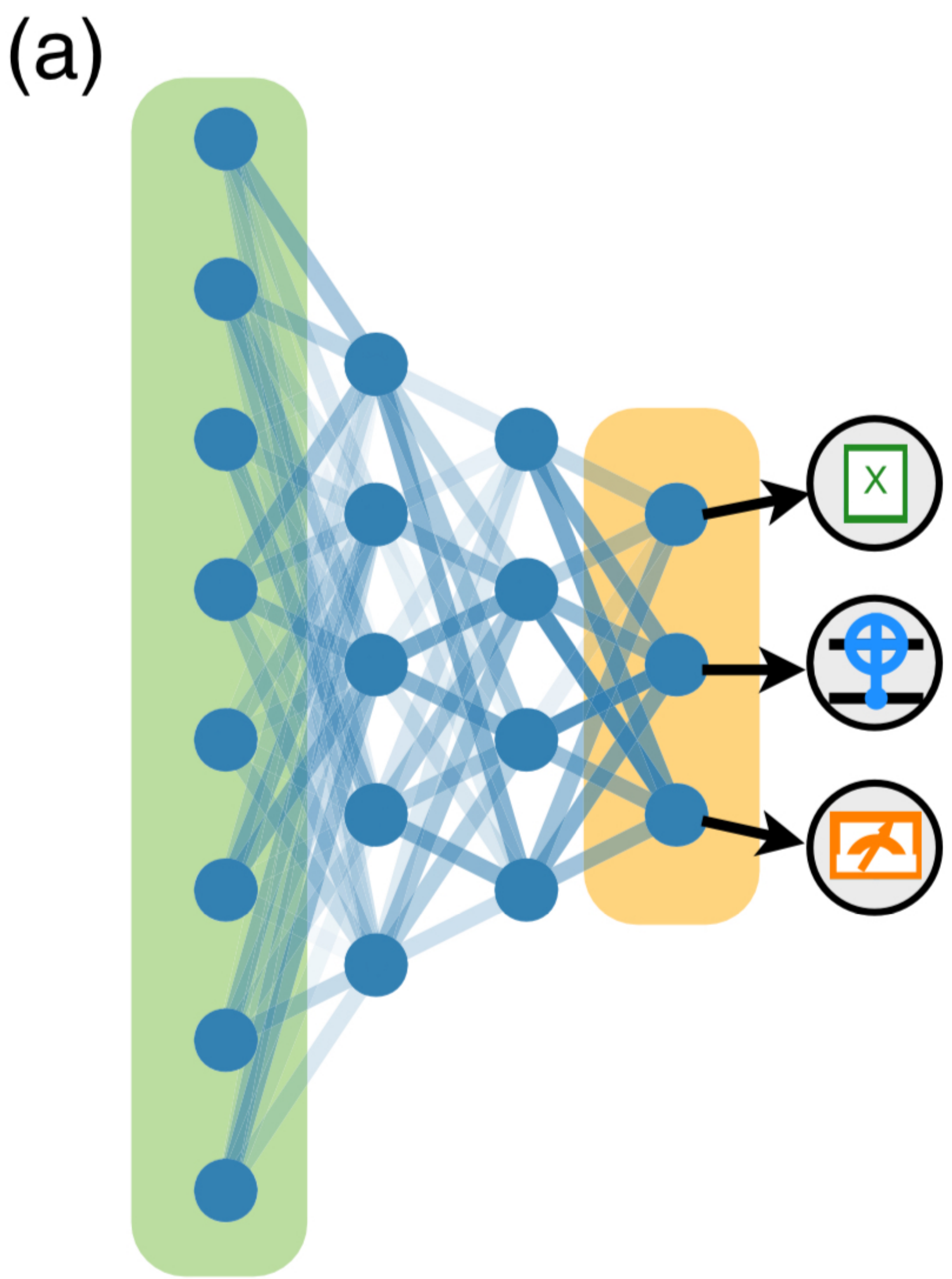
- Autonomous!
  - No requirement of a model
- Feedback based control optimal for QEC



Failure of RL

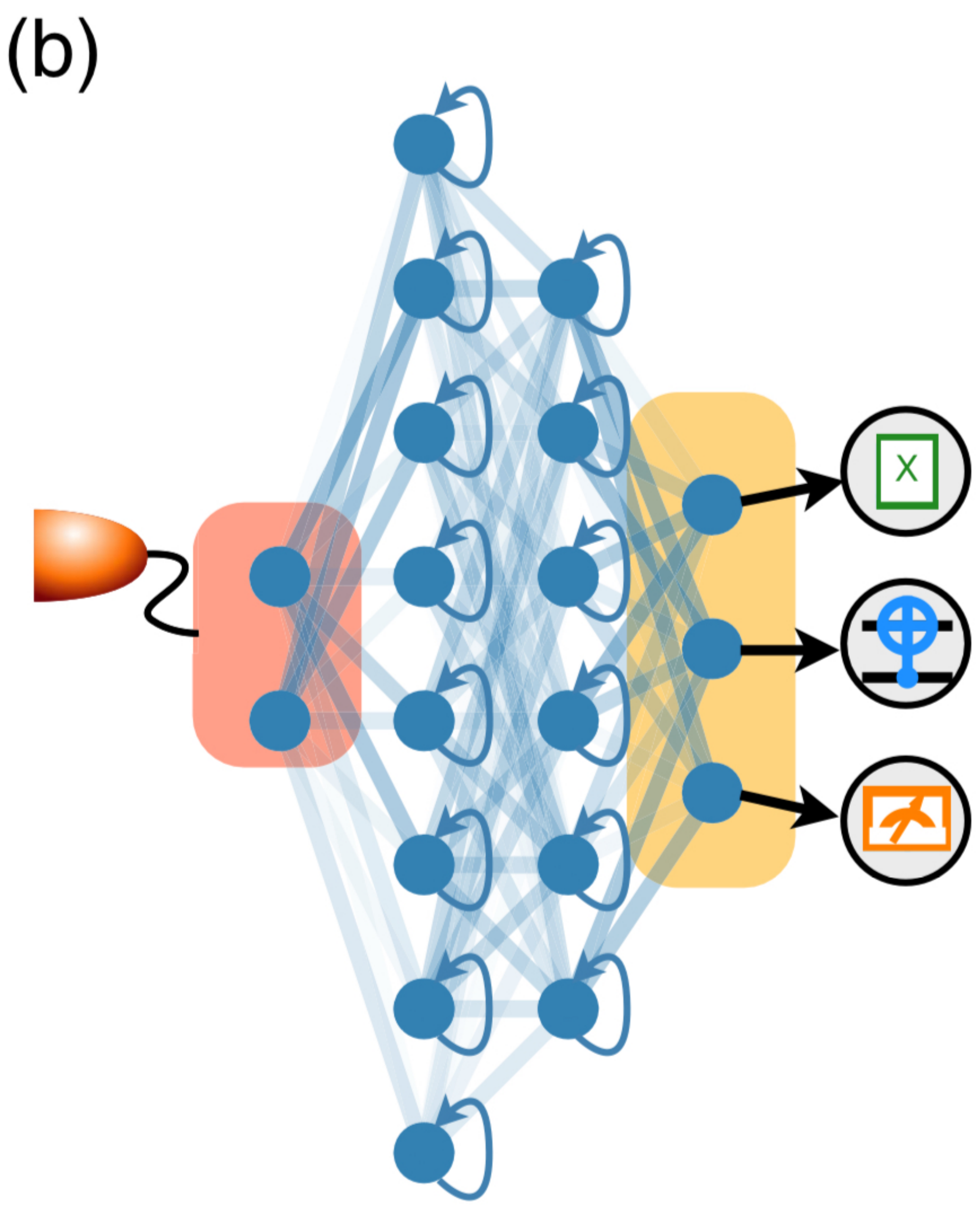


Two stage learning



State-aware network

Quantum states (representing the map for evolution of arbitrary input state up to time *t*)



Recurrent network

Measurement results

Action probabilities

State aware network : Input

$$\hat{\rho}_0 = \frac{1}{2}\left[\hat{\rho}_{\vec{e}_j}(0) + \hat{\rho}_{-\vec{e}_j}(0)\right]$$

$$\delta\hat{\rho}_j = \frac{1}{2}\left[\hat{\rho}_{\vec{e}_j}(0) - \hat{\rho}_{-\vec{e}_j}(0)\right]$$

Agent $\xrightarrow{\text{policy } \pi}$ **Action : Measurement/Gate Operation**

$$\hat{\rho}_0$$
$$\hat{\rho}_1 := \hat{\rho}_0 + \delta\hat{\rho}_x$$
$$\hat{\rho}_2 := \hat{\rho}_0 + \delta\hat{\rho}_y$$
$$\hat{\rho}_3 := \hat{\rho}_0 + \delta\hat{\rho}_z$$

State aware network : Time evolution

Agent

policy π $\rightarrow$

Action : Measurement/Gate Operation

$\downarrow$

$$\hat{\rho}(t_f) = \frac{\phi[\hat{\rho}_0(t_i)]}{tr\{\phi[\hat{\rho}_0(t_i)]\}}$$

$$\delta\hat{\rho}(t_f) = \frac{\phi[\delta\hat{\rho}_0(t_i)]}{tr\{\phi[\hat{\rho}_0(t_i)]\}}$$

$$\phi[\hat{\rho}] = e^{\Delta t D}(\hat{U}\hat{\rho}\hat{U}^{\dagger})$$

- $\hat{U}$: Unitary operation(projection operators/ gate operators)
- D : Dissipative part from the error model employed

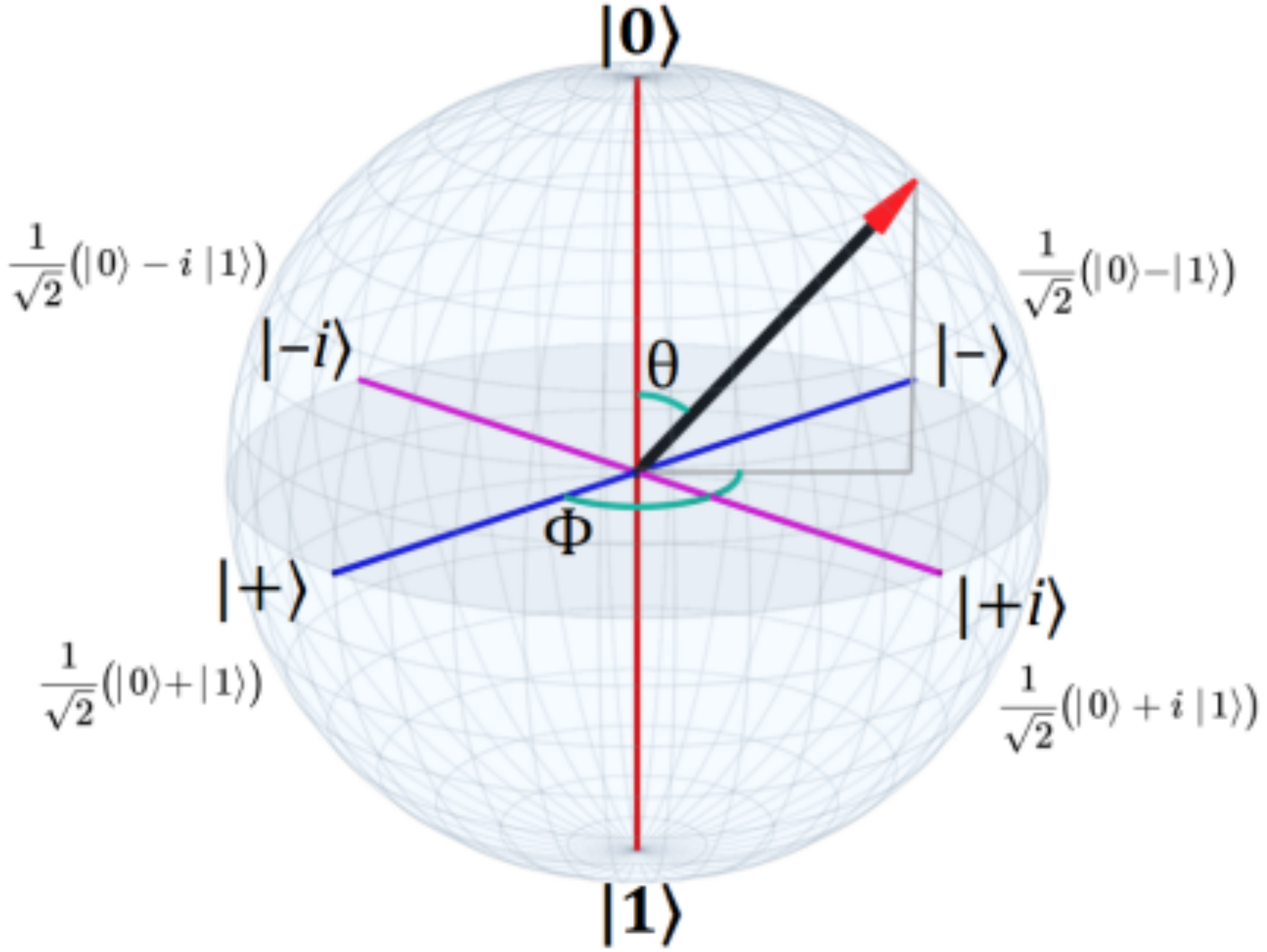
E.g. for bit flip:

$$D_{BF}\hat{\rho} = \frac{\sum_j (\hat{\sigma}_x\hat{\rho}\hat{\sigma}_x - \hat{\rho})}{T_{decay}}$$

State aware network : Recoverable Quantum Information

Remaining information in evolved state?

Success in distinguishing antipodal logical states?



Reward (based on R)

Recoverable Quantum Information

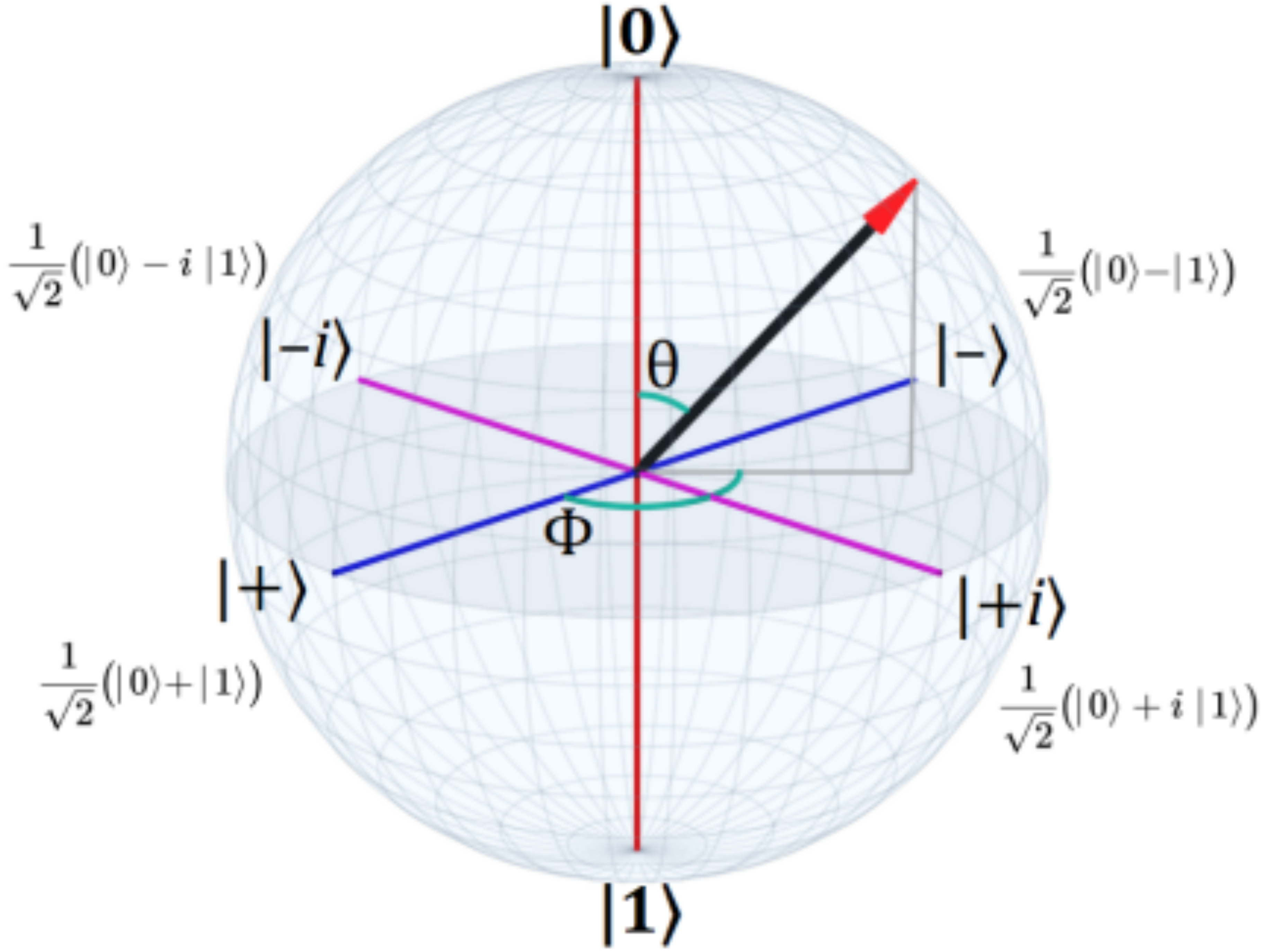
$$R_Q(t) = \frac{1}{2} min_{\vec{n}} \parallel \hat{\rho}_{\vec{n}}(t) - \rho_{-\vec{n}}(t) \parallel$$

Time evolved states

State aware network : Recoverable Quantum Information

Remaining information in evolved state?

Success in distinguishing antipodal logical states?



Reward (based on R)

Recoverable Quantum Information

$$R_Q(t) = min_{\vec{n}} \left\| \sum_j n_j \delta\hat{\rho}_j(t) \right\|$$

Time evolved states

State aware network : Reward Function

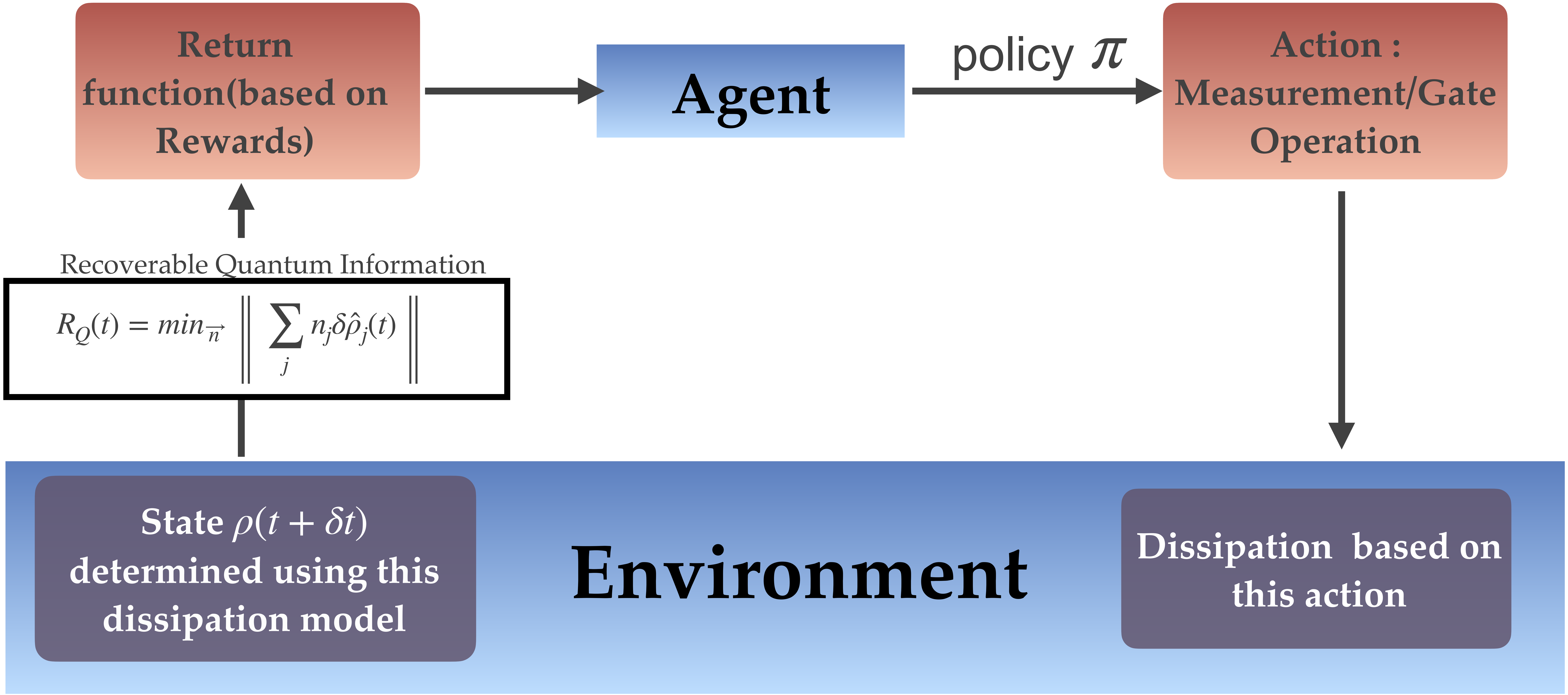
Return function(based on Rewards)

State $\rho(t+\delta t)$

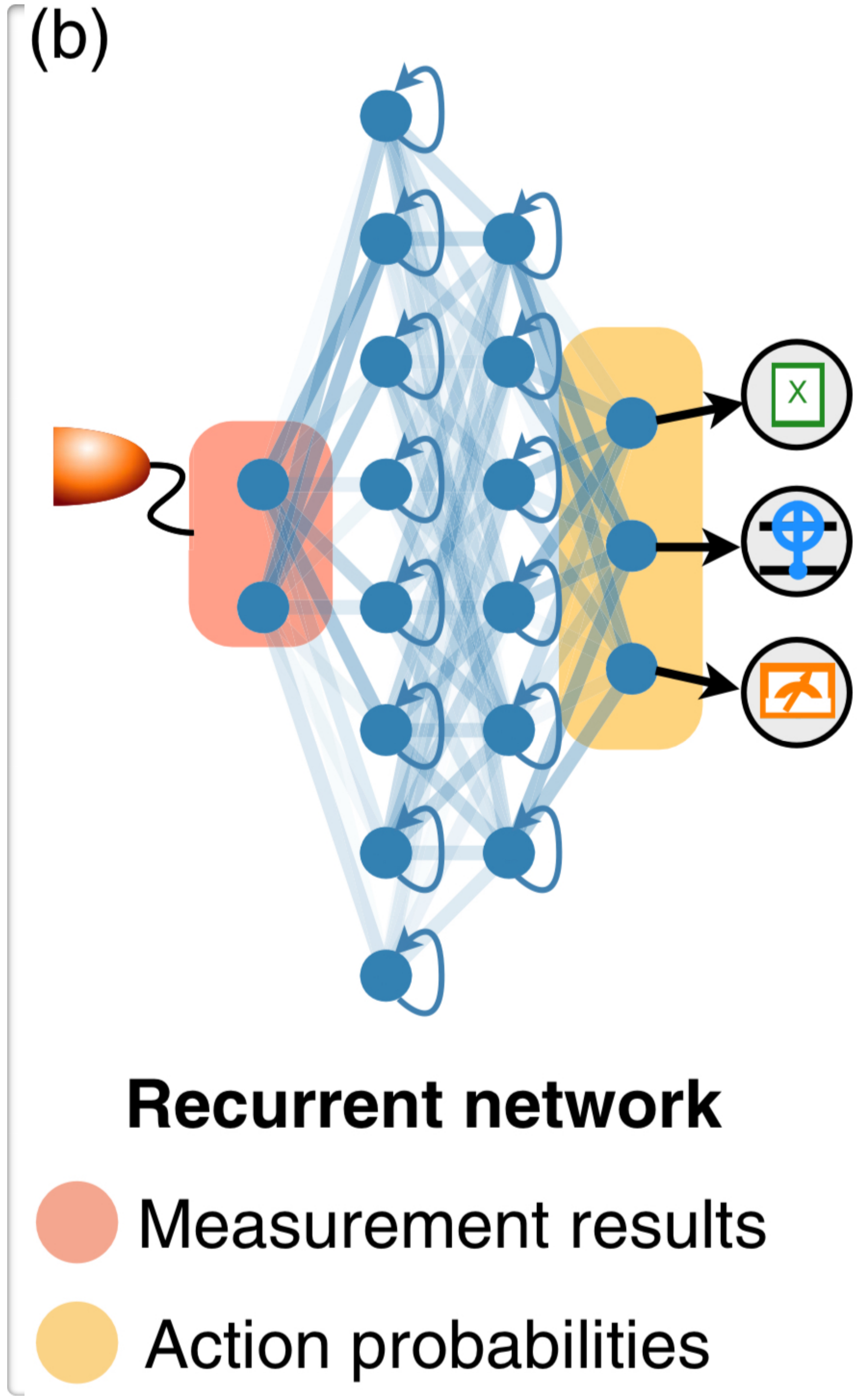
$$r_t = \begin{cases} 1+\frac{\bar{\mathcal{R}}_Q(t+\Delta t)-\mathcal{R}_Q(t)}{2\Delta t/T_{\text{single}}} & +0 \quad \text{if } \bar{\mathcal{R}}_Q(t+\Delta t)>0, \\ 0 & -P \quad \text{if } \mathcal{R}_Q(t)\neq 0 \\ & \qquad \wedge\, \mathcal{R}_Q(t+\Delta t)=0, \\ \underbrace{\qquad 0 \qquad}_{=:r_t^{(1)}} & \underbrace{+0}_{=:r_t^{(2)}} \quad \text{if } \mathcal{R}_Q(t)=0 \end{cases}$$

Recoverable Quantum Information

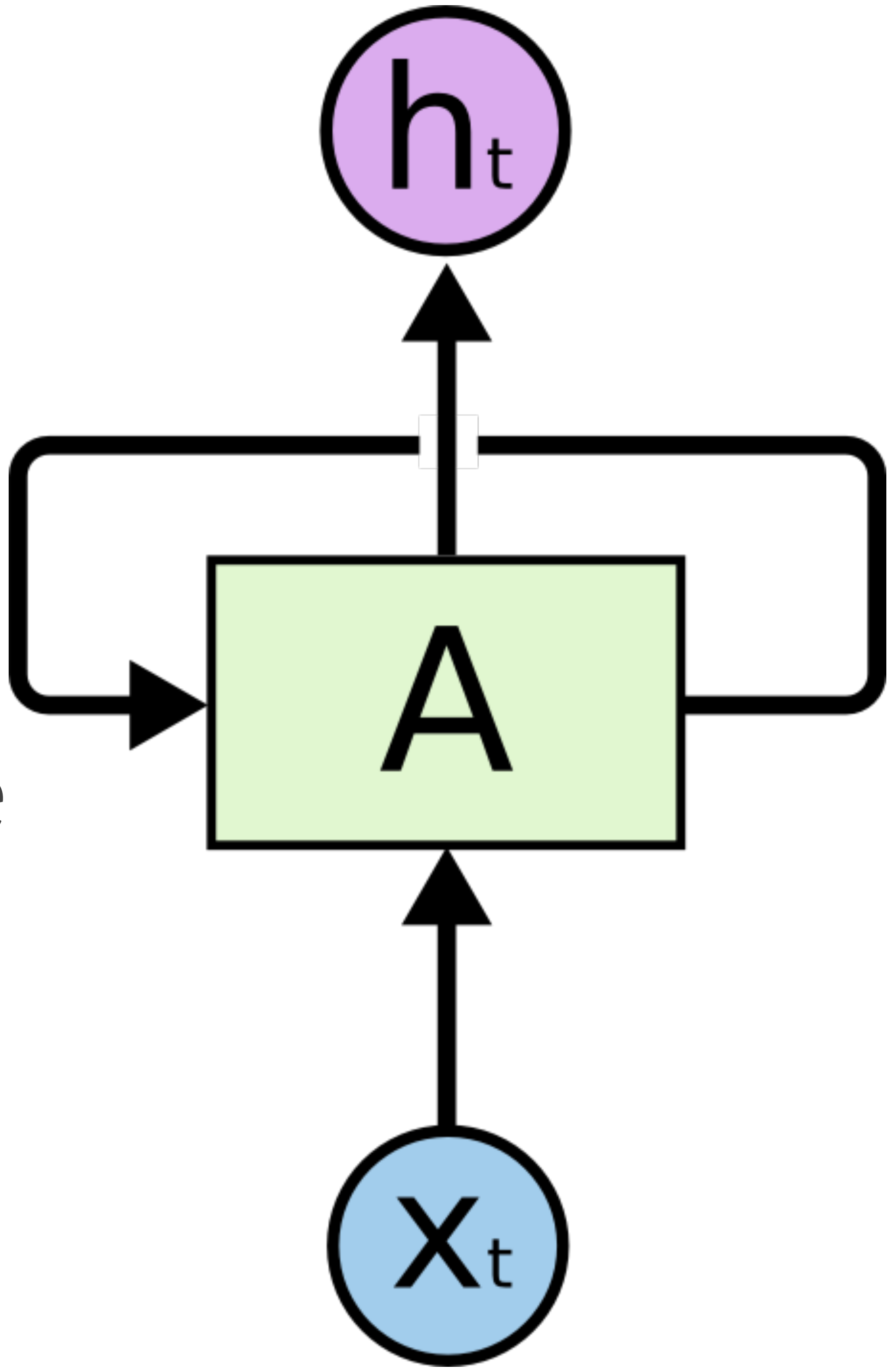
State aware network



Recurrent Network



- Applied in experiments
- Receives measurement results, most recent action
- Trained using supervised learning
- Training data: input and policy for each time step in each trajectory
- Memory



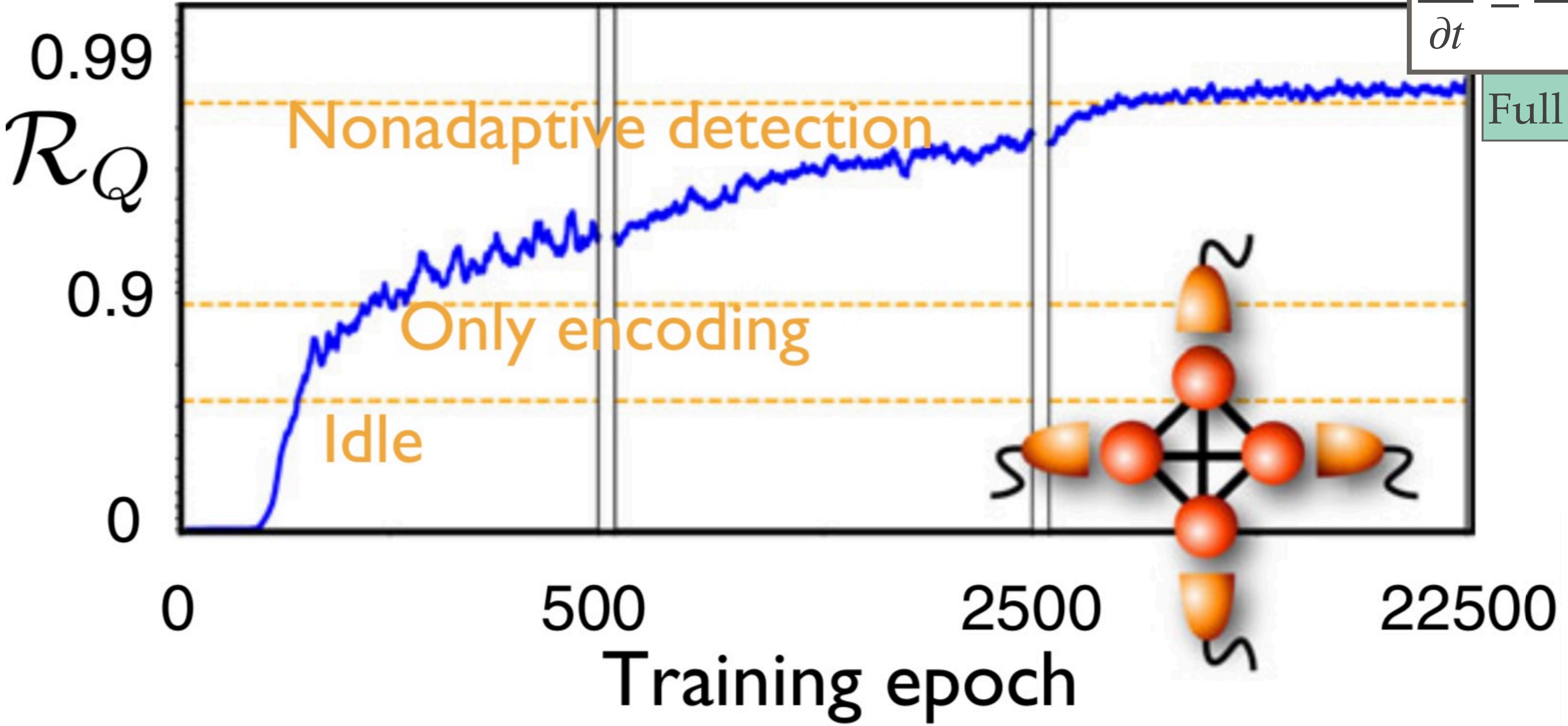
Results

Learns encoding and adaptive detection

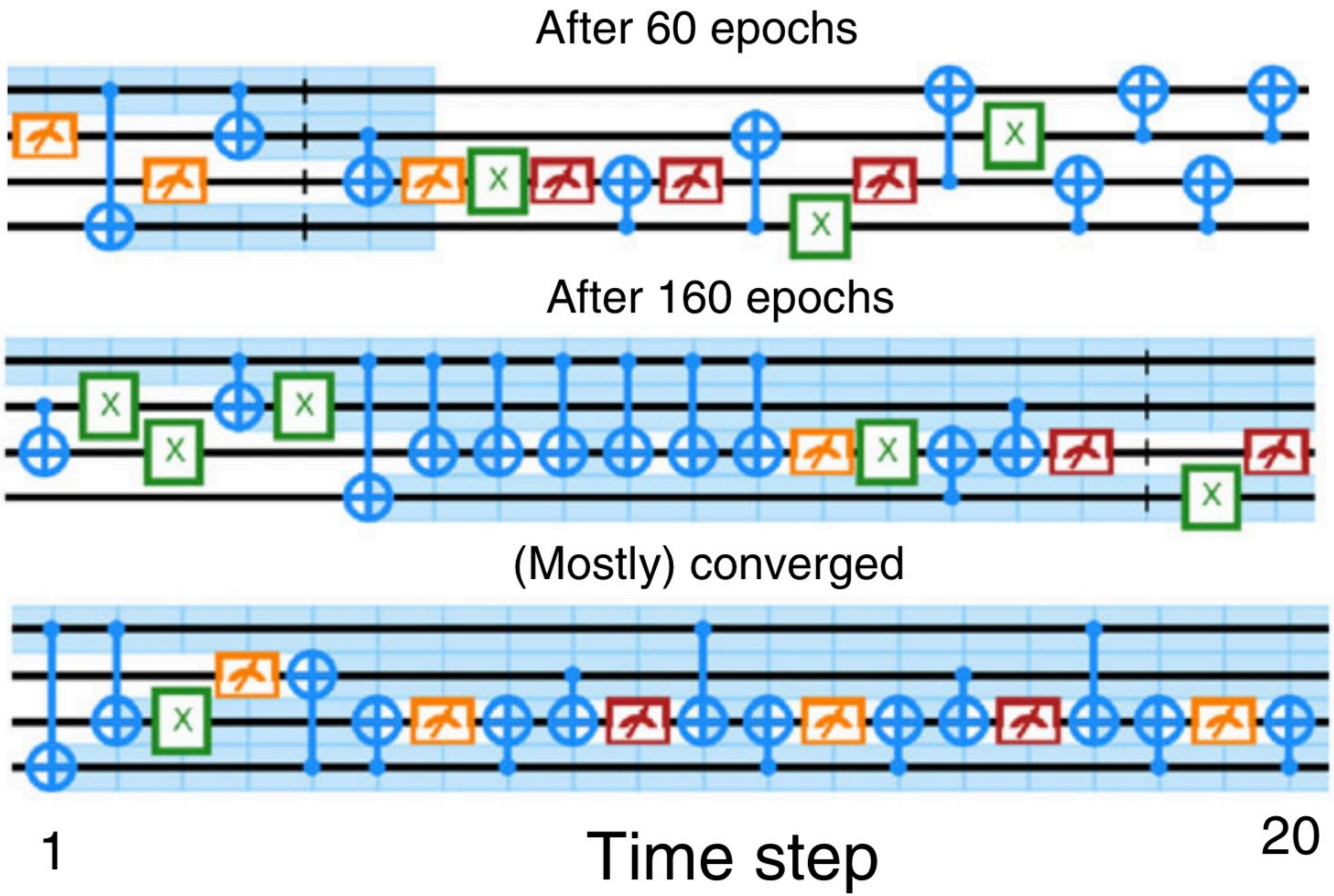
Error model:

$$\frac{\partial \hat{\rho}}{\partial t} = \frac{\sum_j (\hat{\sigma}_x \hat{\rho} \hat{\sigma}_x - \hat{\rho})}{T_{decay}}$$

Full connectivity!



Learns encoding and adaptive detection

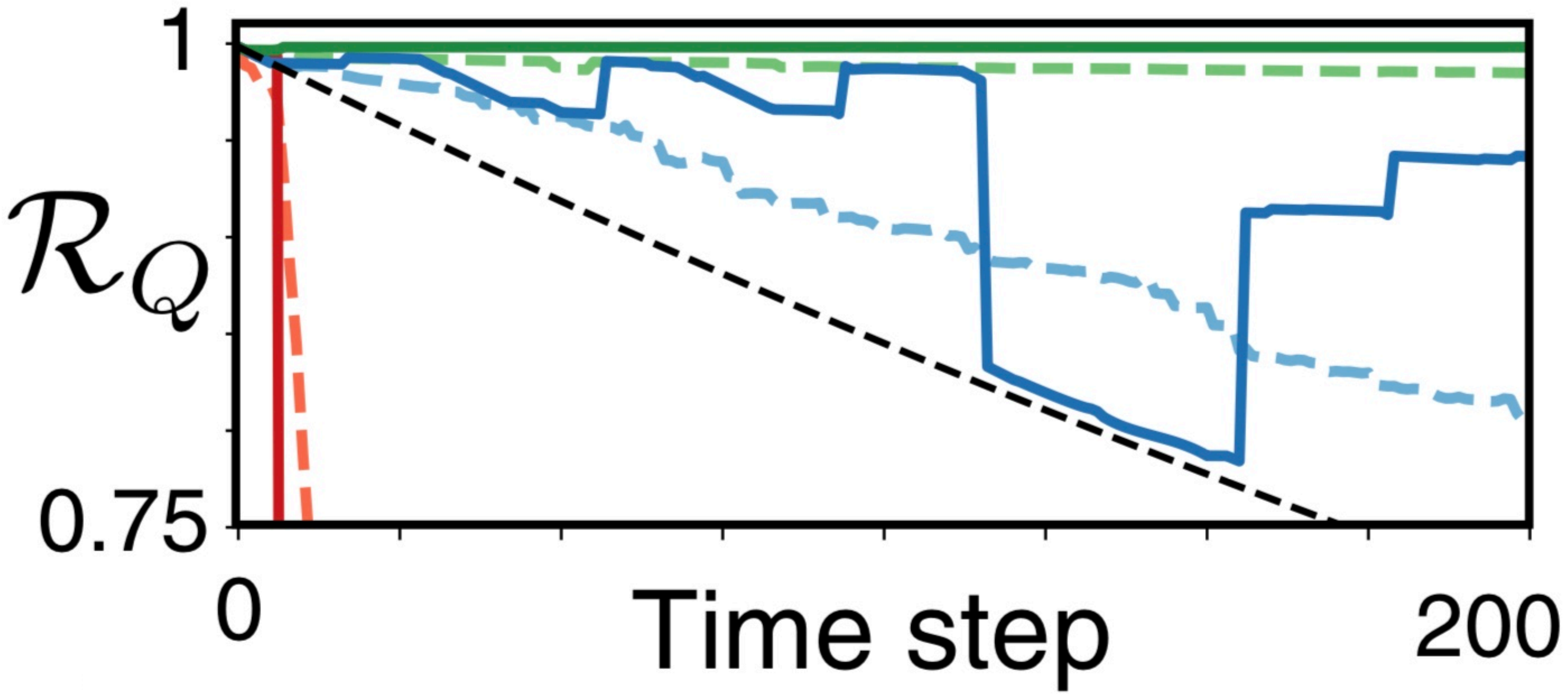


Error model:

$$\frac{\partial \hat{\rho}}{\partial t} = \frac{\sum_j (\hat{\sigma}_x \hat{\rho} \hat{\sigma}_x - \hat{\rho})}{T_{decay}}$$

Full connectivity!

Learns encoding and adaptive detection



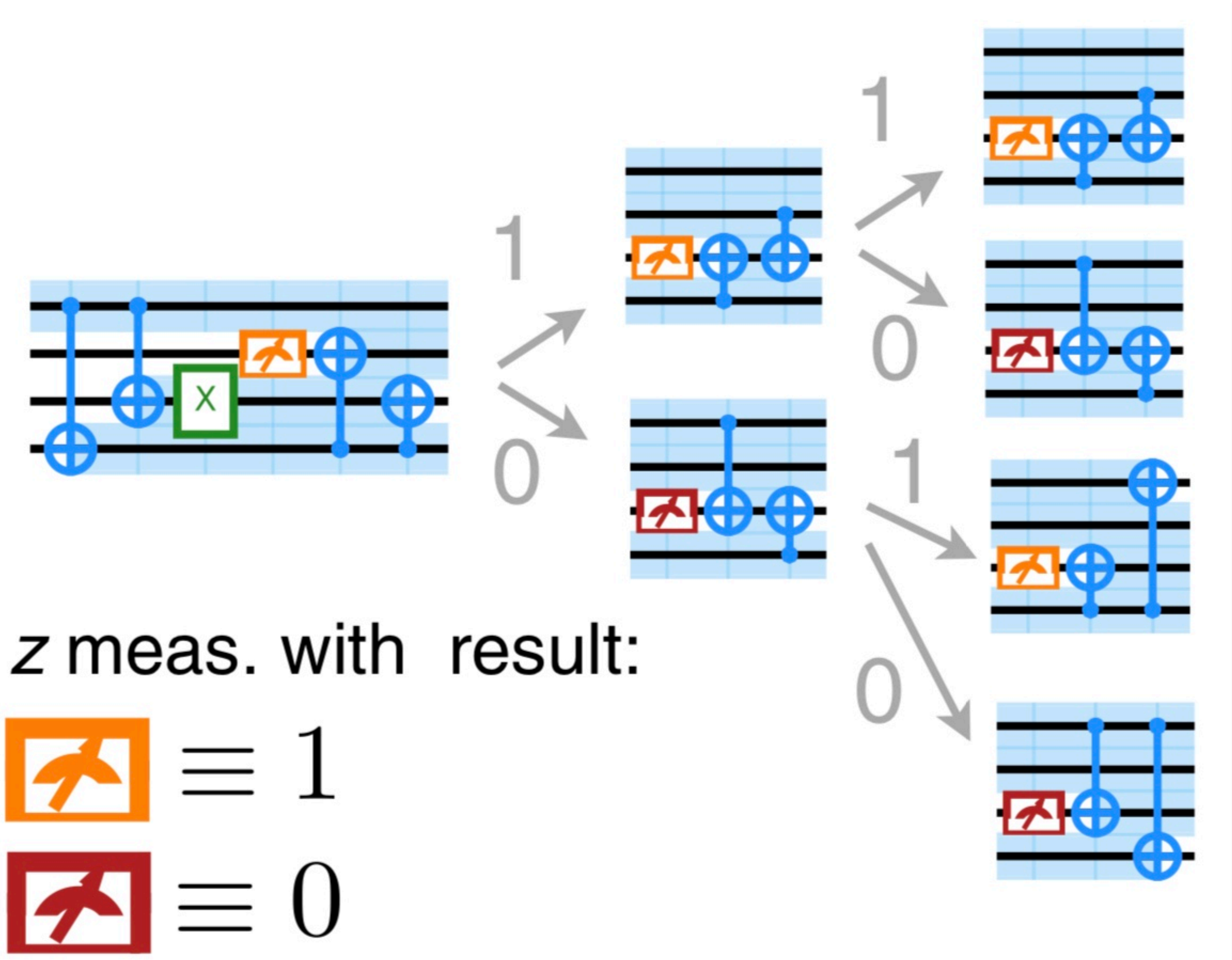
Error model:

$$\frac{\partial \hat{\rho}}{\partial t} = \frac{\sum_j (\hat{\sigma}_x \hat{\rho} \hat{\sigma}_x - \hat{\rho})}{T_{decay}}$$

Full connectivity!

Red: after 60 epochs
Blue: after 160 epochs
Green: Mostly converged
Dashed: averaged over many trajectories

Learns encoding and adaptive detection

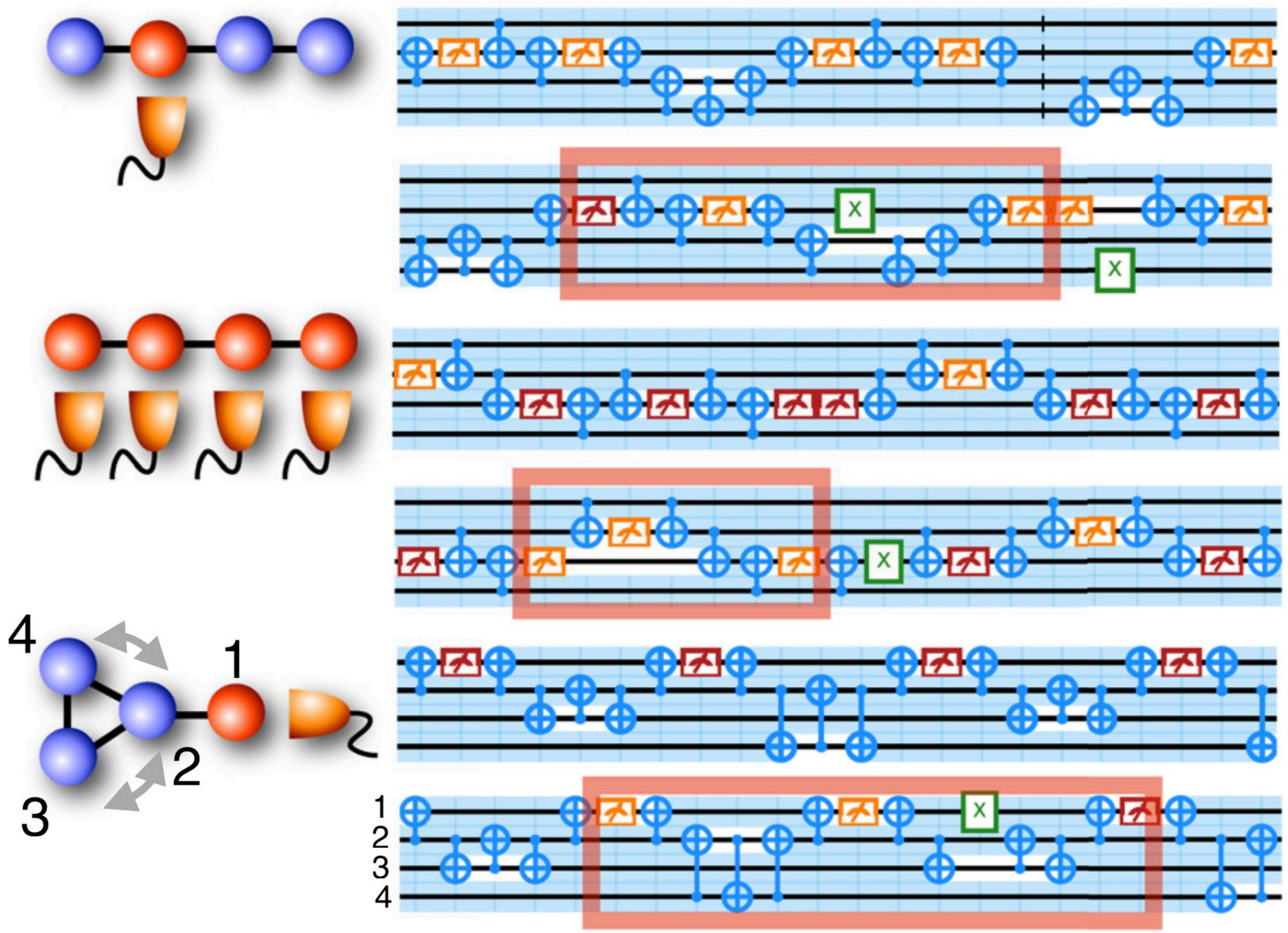


Error model:

$$\frac{\partial\hat{\rho}}{\partial t} = \frac{\sum_j (\hat{\sigma}_x \hat{\rho} \hat{\sigma}_x - \hat{\rho})}{T_{decay}}$$

Full connectivity!

Discovers feedback strategies based on available resources

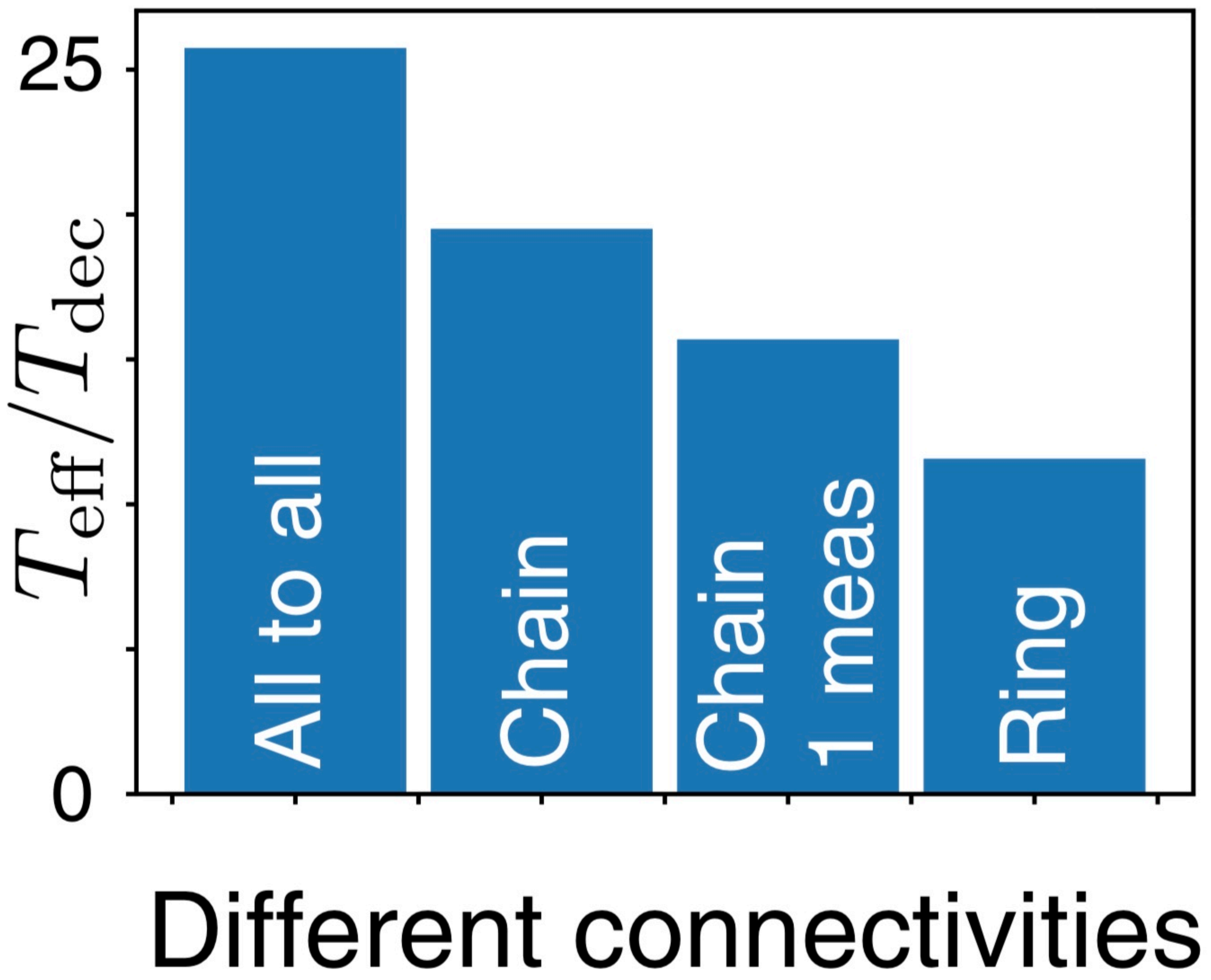


A) CNOT gates only available between nearest neighbours; single measurement location.

B) CNOT gates only available between nearest neighbours; every qubit can be measured

C) Ring like connectivity for CNOTs; measurement on first qubit

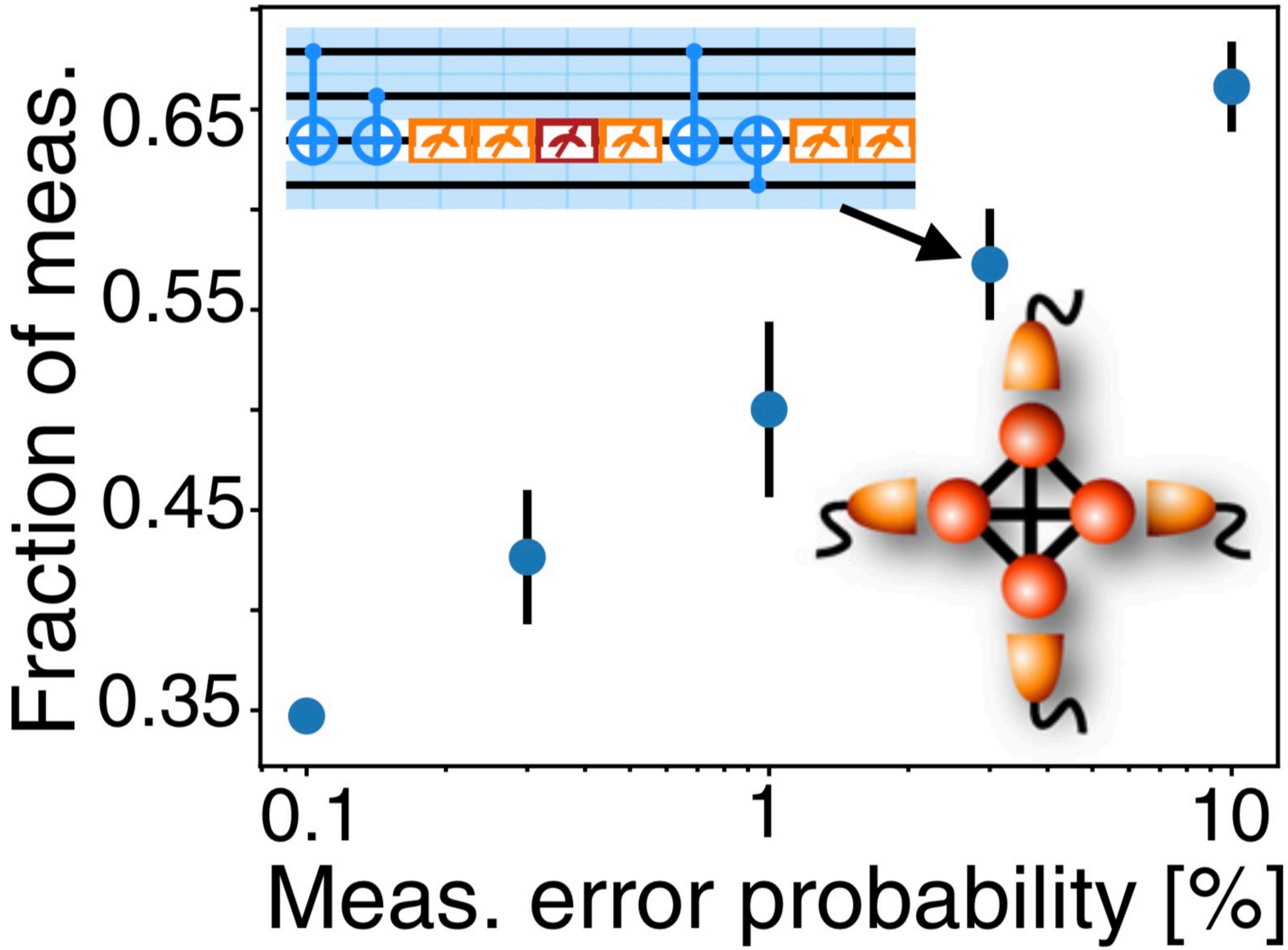
Discovers feedback strategies based on available resources



T_{dec} = 1200 is the single qubit decoherence time (in units of gate time that defines the time step)

T_{eff} extracted from decay of R_Q after 200 steps

Discovers feedback strategies based on available resources



Summary

- QEC from scratch
- Detection and recovery sequences for diverse settings
- Trained neural networks can be applied to experiments
- This approach can be applied to diverse noises/errors
- RL is a flexible and general tool which can be used for exploring problems requiring feedback based control in physics.

Thank You !

References

- Reinforcement Learning with neural networks for quantum feedback; Thomas Fösel, Petru Tighineanu, Talitha Weiss, and Florian Marquardt; Phys. Rev. X
- Reinforcement Learning, second edition: An Introduction (Adaptive Computation and Machine Learning series) (2. Aufl.); Sutton, R. S. & Barto, A. G. (2018). Bradford Books.
- Lectures on Reinforcement Learning ; David Silver

Figures

- https://oxfordre.com/physics/view/10.1093/acrefore
- https://en.wikipedia.org/wiki/Controlled_NOT_gate
- https://quantumcomputing.stackexchange.com/three-qubit-bit-flip-code
- https://qpiai.tech/quantum-error-correction/
- https://www.kdnuggets.com/5-things-reinforcement-learning.html
- https://www.analyticsvidhya.com/introduction-deep-q-learning-python/
- https://mpl.mpg.de/2019_Machine_Learning/
- https://logosconcarne.com/2021/03/15/qm-101-bloch-sphere/
- https://colah.github.io/posts/2015-08-Understanding-LSTMs/