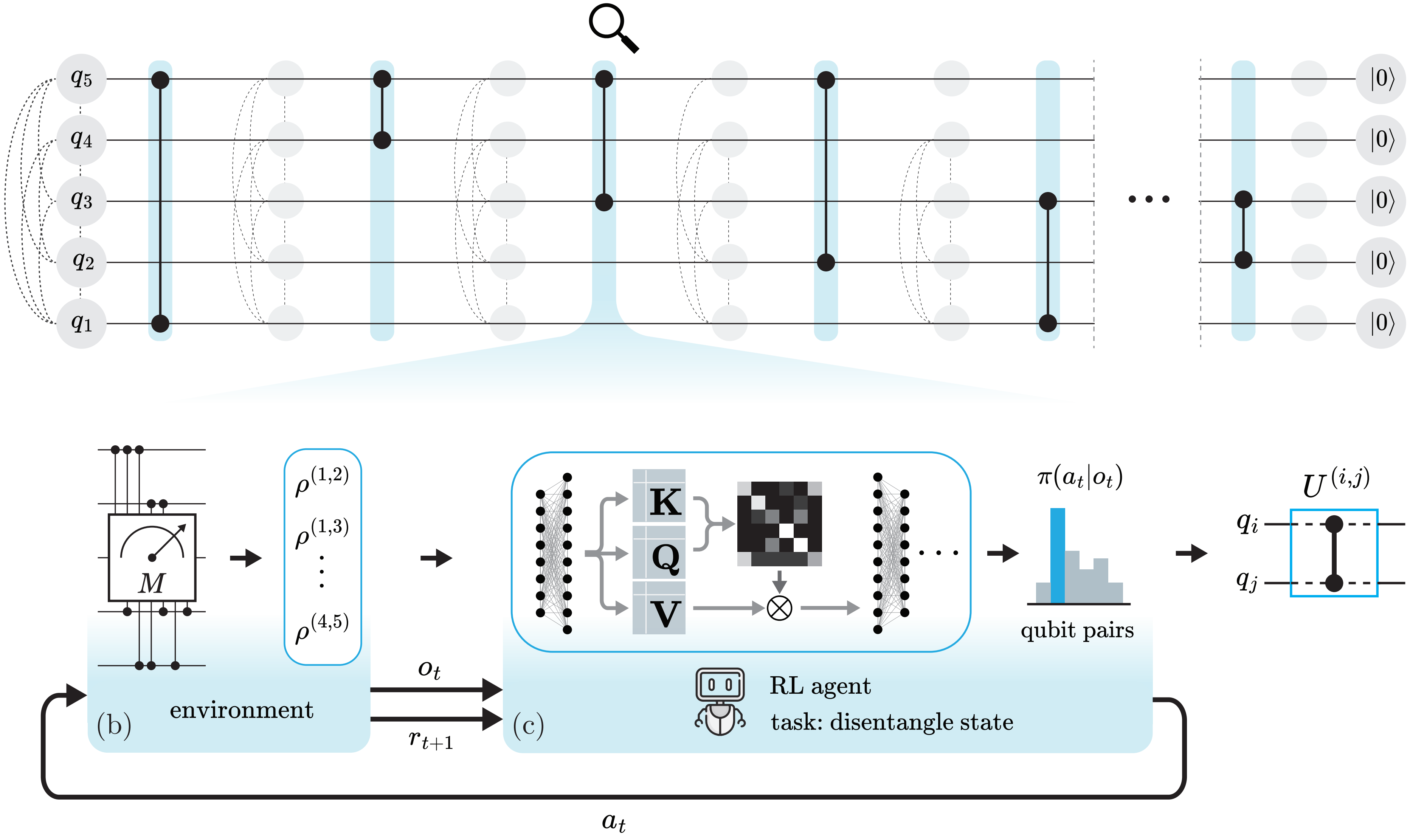
Deep Reinforcement Learning
for
Quantum Technology



Funded by the European Union

European Research Council
Established by the European Commission

Deutsche Forschungsgemeinschaft

Marín Bukov

MPI-PKS

RL related talks in ML4QT2

2nd Workshop "Machine Learning for Quantum Technology" - SCHEDULE

	WEDNESDAY, NOV. 6	**THURSDAY, NOV. 7**	**FRIDAY, NOV. 8**
9:00 - 9:30	**Registration + opening remarks**	**Registration**	**Registration**
9:30 - 10:05	Eliška Greplová *Autonomous Quantum Control in the age of AI*	Monika Aidelsburger *Quantum many-body systems under the microscope*	Vedran Dunjko *Provable exponential quantum advantages in learning from classical data*
10:05 - 10:40	Anton Frisk Kockum *Quantum state and process tomography with machine learning and gradient descent*	Simon Trebst *Decoding many-body teleportation*	Hans Briegel *Towards explainable AI in quantum science*
10:40 - 10:55	Martin Gärttner *Machine learning assisted quantum simulator readout*	Yue Ban *Neural-network-assisted parameter estimation for quantum detection*	Chenfeng Cao *Unveiling quantum phase transitions from traps in variational quantum algorithms*
10:55 - 11:20	**Coffee break**		
11:20 - 11:55	Christopher Eichler *Realizing a reinforcement learning agent for real-time quantum feedback*	Giuseppe Carleo *Neural quantum states for many-body electronic structure and dynamics*	Johannes Bausch *Machine Learning for Fault-Tolerant Quantum Computation*
11:55 - 12:30	Annabelle Bohrdt *Trying to solve quantum many-body problems with neural networks*	Markus Schmitt *(Neural) network representations of many-body wave functions*	Evert van Nieuwenburg *RL and RL for quantum systems*
12:30 - 12:45	Maximilian Prüfer *Physics-inspired machine learning models and optimal control for quantum experiments*	Dario Poletti *Paths towards time evolution with larger neural-network quantum states*	Matias Bilkis *Automatic re-calibration of quantum devices by RL*
12:45 - 13:00	Petr Zapletal *Error-tolerant quantum convolutional neural networks for symmetry-protected topological phases*	Gorka Muñoz-Gil *Representation learning reaches the lab: let machines act!*	Clara Wanjura *Quantum Equilibrium Propagation for efficient training of quantum systems based on Onsager reciprocity*
13:00 - 14:30	**Lunch break**		
14:30 - 15:05	Marín Bukov *Reinforcement learning transmon-qubit entangling gates*	Roger Melko *Language Models for Quantum Simulation*	Jonas Schuff *Autonomous tuning of spin qubits*
15:05 - 15:40	Volodymyr Sivak *Calibration of decoders for quantum error correction using multi-agent reinforcement learning*	Markus Heyl *Solving 2D quantum matter with neural quantum states*	Christof Weitenberg *Machine learning and ultracold quantum gases*
15:40 - 15:55	Maciej Koch-Janusz *Analyzing and constructing efficient data encoding quantum circuits*	Cristian Bonato *Learning the dynamics of Markovian open quantum systems from experimental data*	**Closing remarks**
15:55 - 16:20	**Coffee break**		
16:20 - 16:35	Bijita Sarma *Fast Hardware-efficient Quantum Gate Design using Optimal Control with Reinforcement Learning Ansatz*	Akash Kundu *Program synthesis-driven quantum architecture search for optimal quantum circuit design in variational quantum algorithms*	
16:35 - 17:10	Mats Granath *Graph neural network based decoders for quantum error correcting codes*	Mario Krenn *Towards an Artificial Muse for new ideas in Science*	
17:10 - 18:00	**Poster flash talks (1′ each) + poster setup**	**Poster flash talks (1′ each) + poster setup**	
From 18:00	**Poster session A (including dinner)**	**Poster session B (including dinner)**	

LEGEND

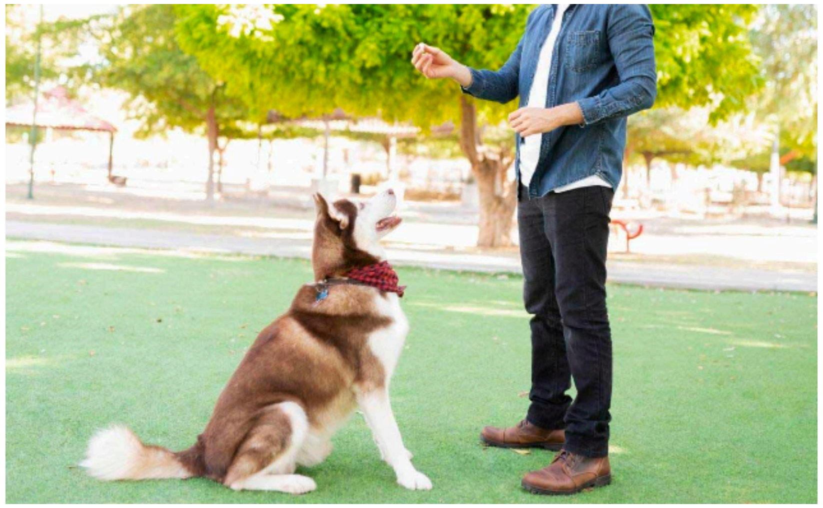
🟩 Invited talk (30′+5′ Q&A)

🟦 Contributed talk (12′+3′ Q&A)

Outline

Part 1

- Reinforcement learning (RL) in quantum physics
    - RL as a branch of machine learning
- Applications of RL
    - hallmark applications of RL
    - applications in quantum technologies
- RL framework in a nutshell
    - environment, states, actions, rewards
    - RL algorithms



Outline

Part 2

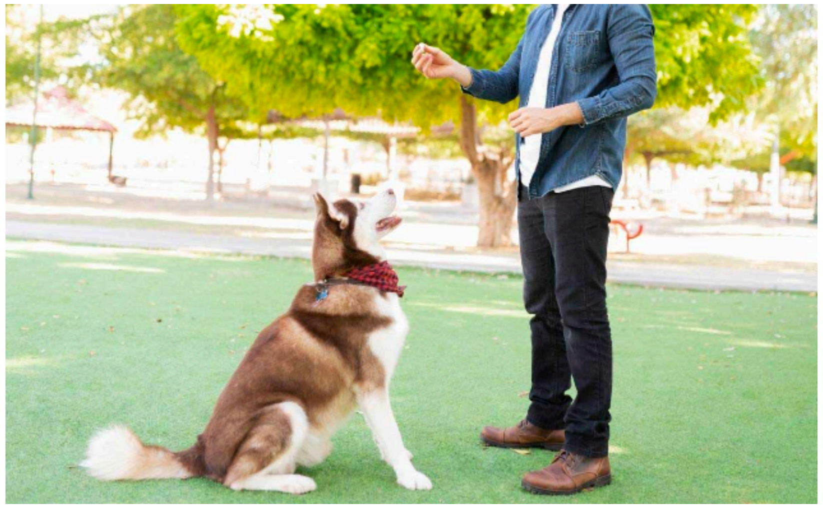
- RL for qubit state preparation
  - effect of noise (measurement shot noise, coherent, incoherent noise)

- experimentally friendly RL framework
  - partially observable environments
  - environment, states, actions, rewards

Outline

Part 1

- Reinforcement learning (RL) in quantum physics
  - RL as a branch of machine learning



- Applications of RL
  - hallmark applications of RL
  - applications in quantum technologies
- RL framework in a nutshell
  - environment, states, actions, rewards
  - RL algorithms

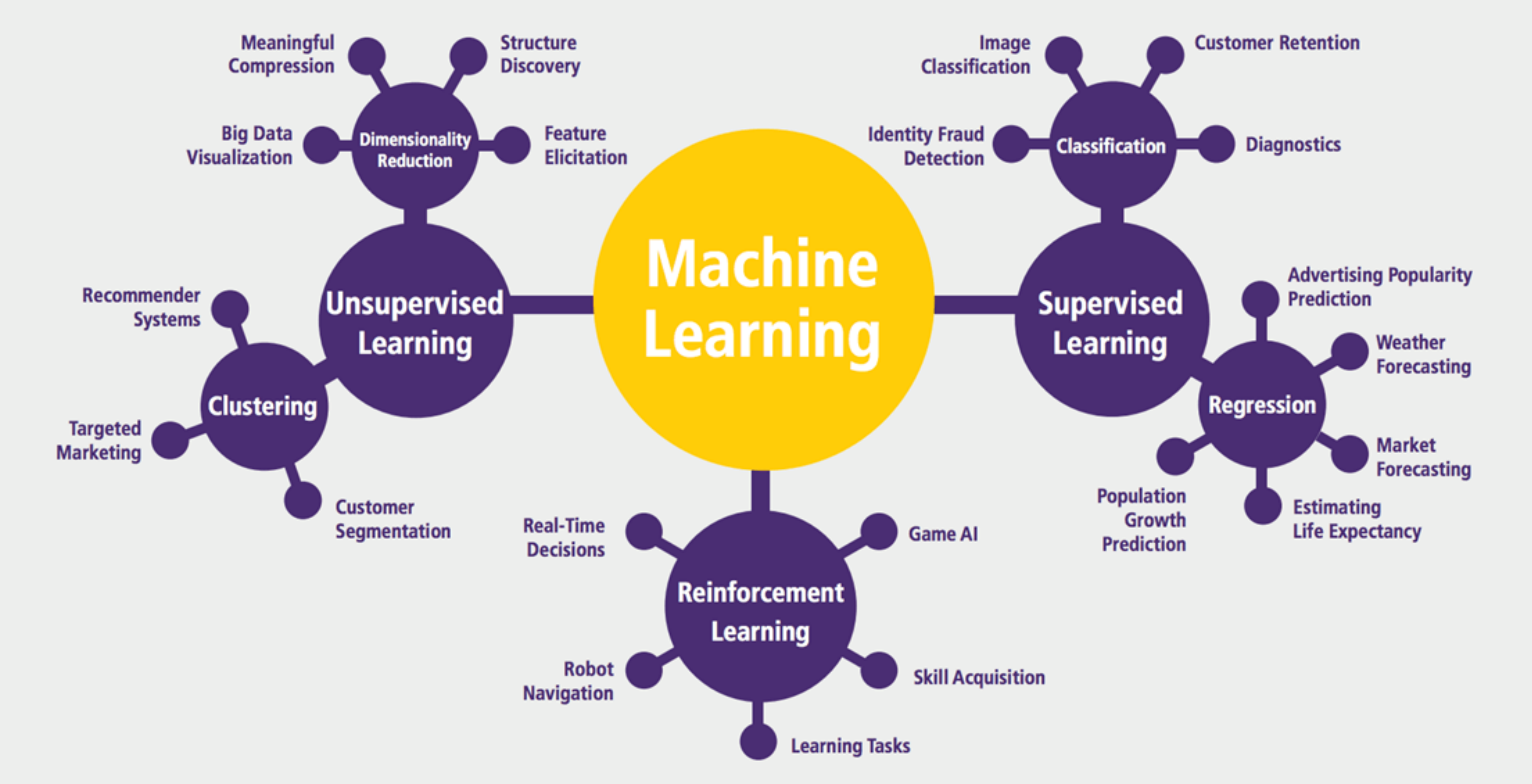


image: Priya Pareek

Supervised Learning

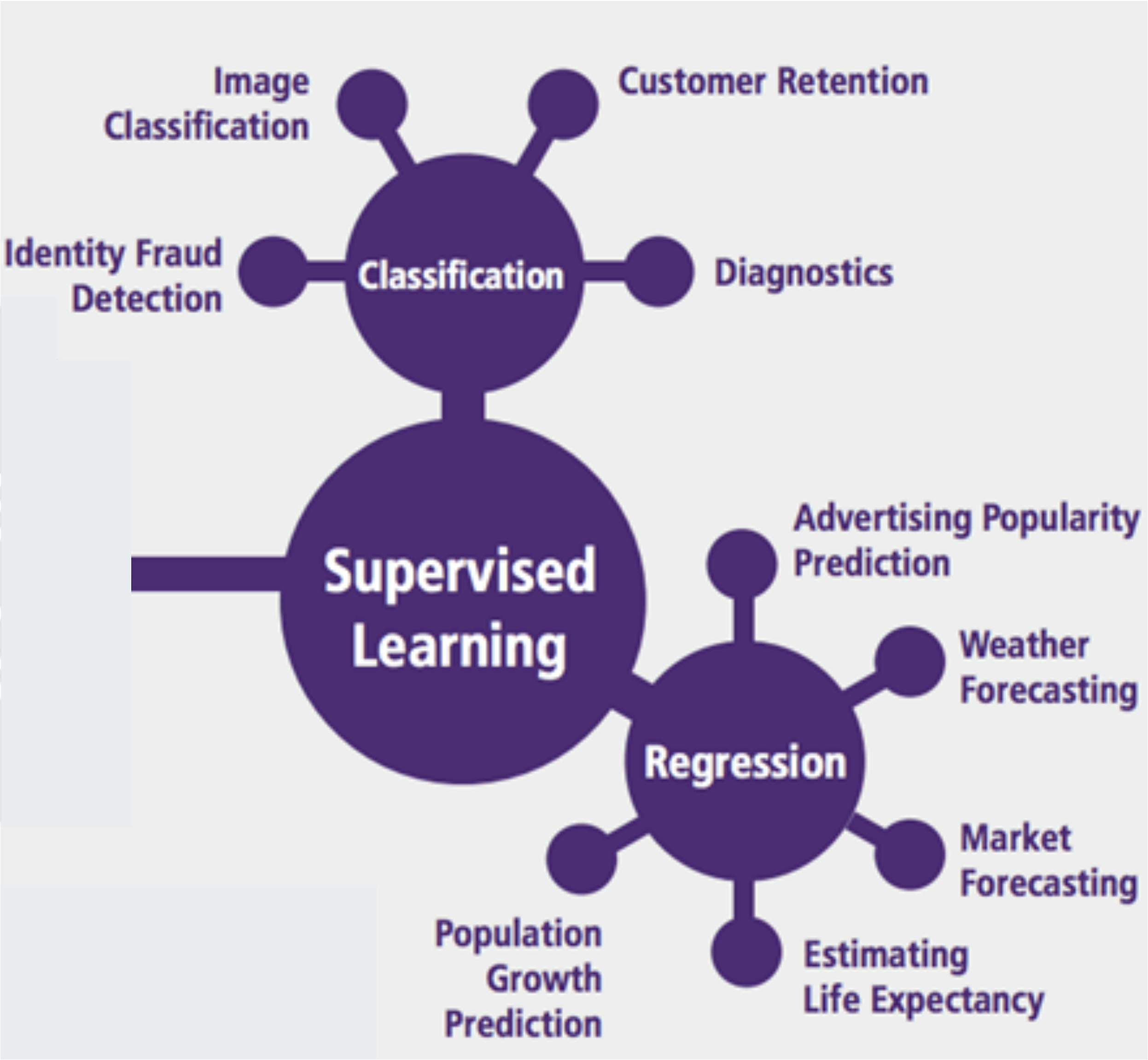
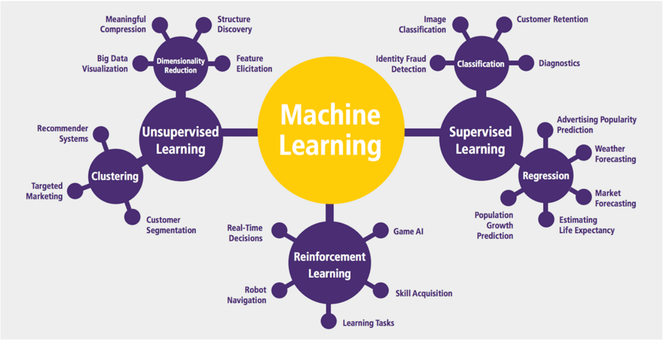
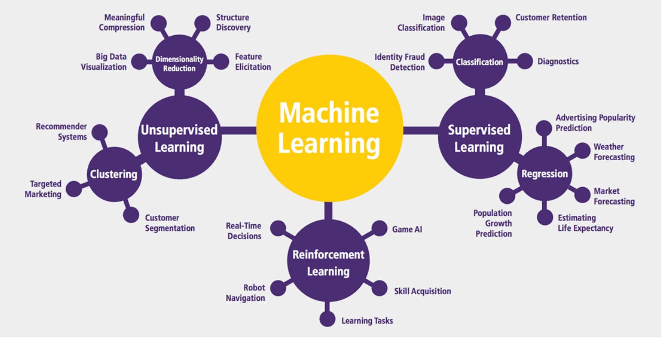


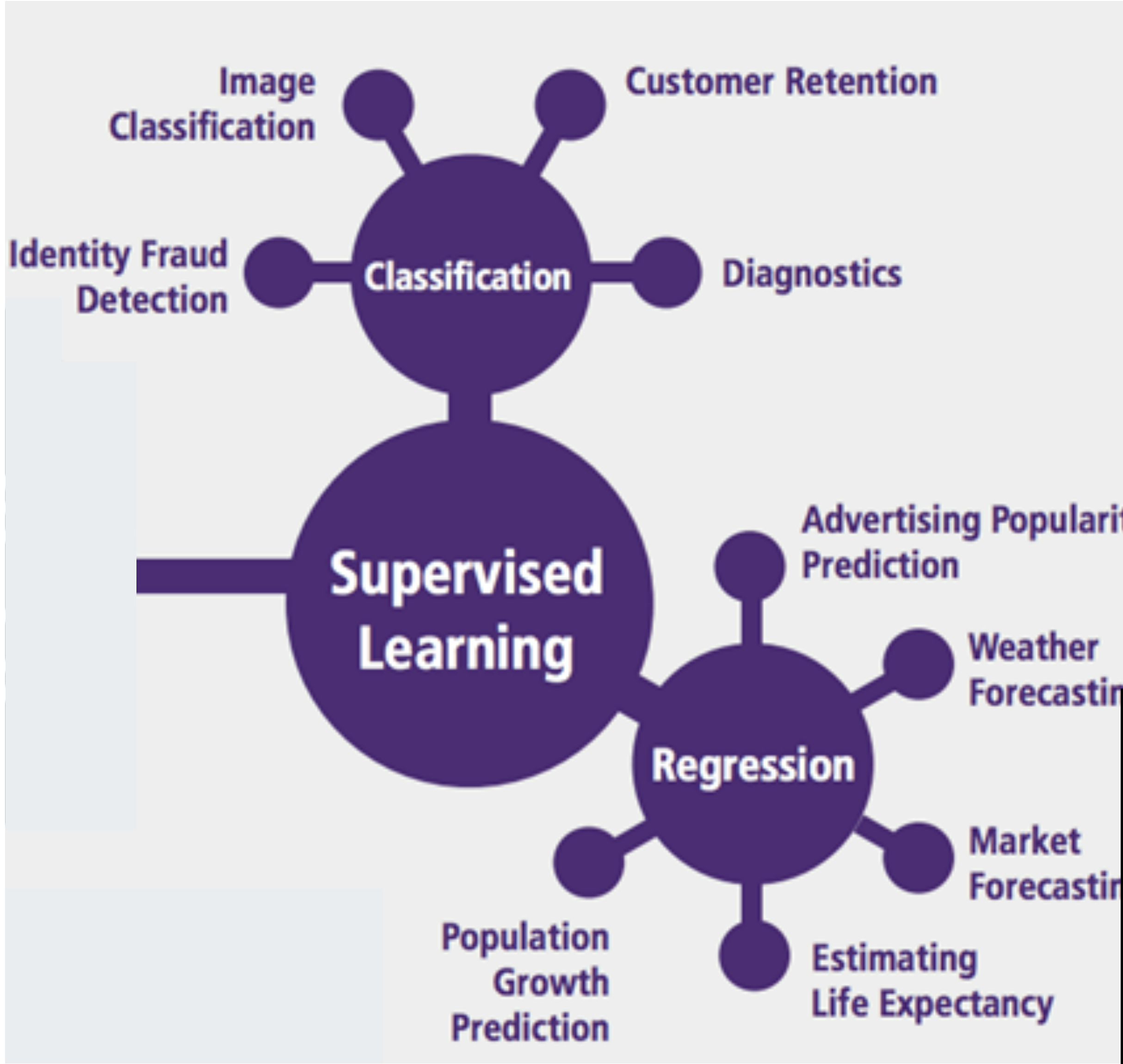
image: Priya Pareek



- learning from examples (labeled data)

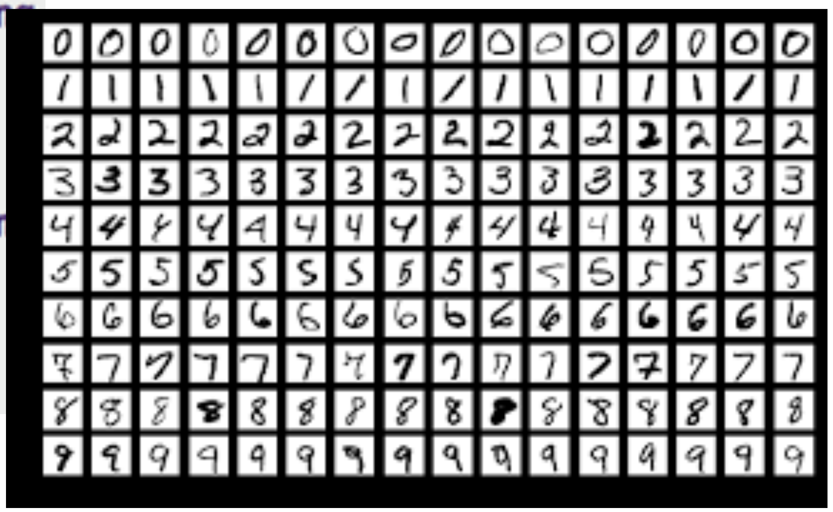
Supervised Learning

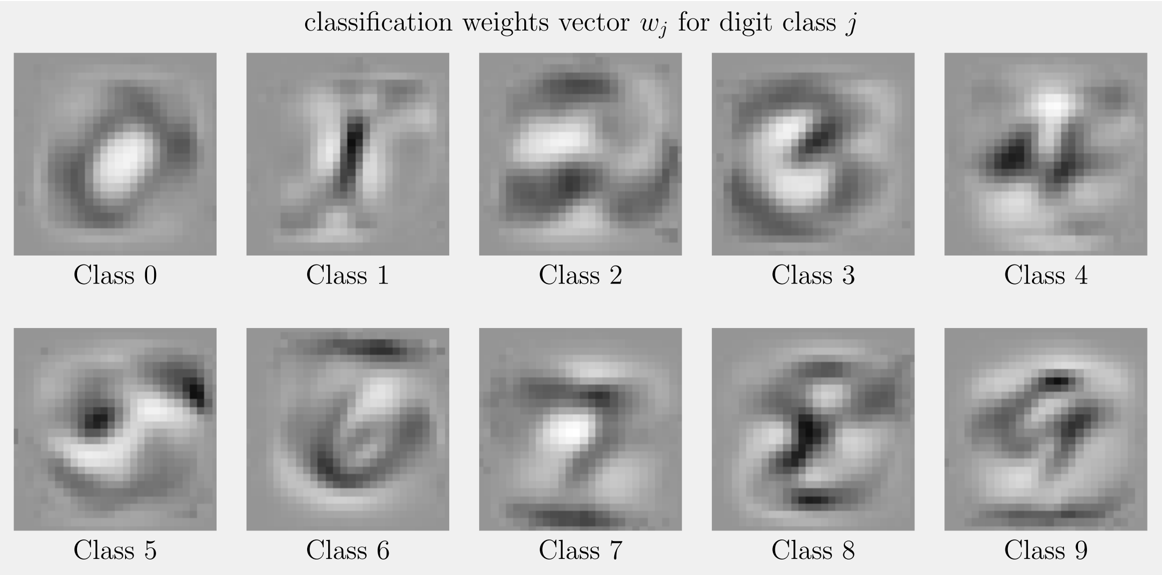




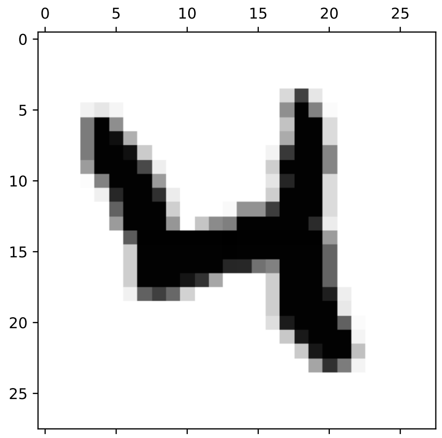
- learning from examples (labeled data)

MNIST



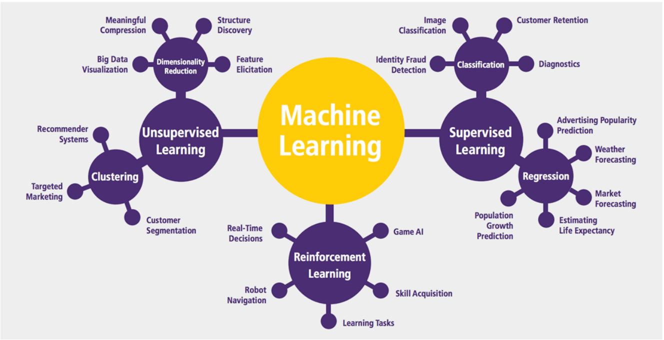


classification weights vector w_j for digit class j



Mehta, MB,…, Phys. Reports ’19

Supervised Learning



- learning from examples (labeled data)
  - MNIST
  - recognizing hand-written digits

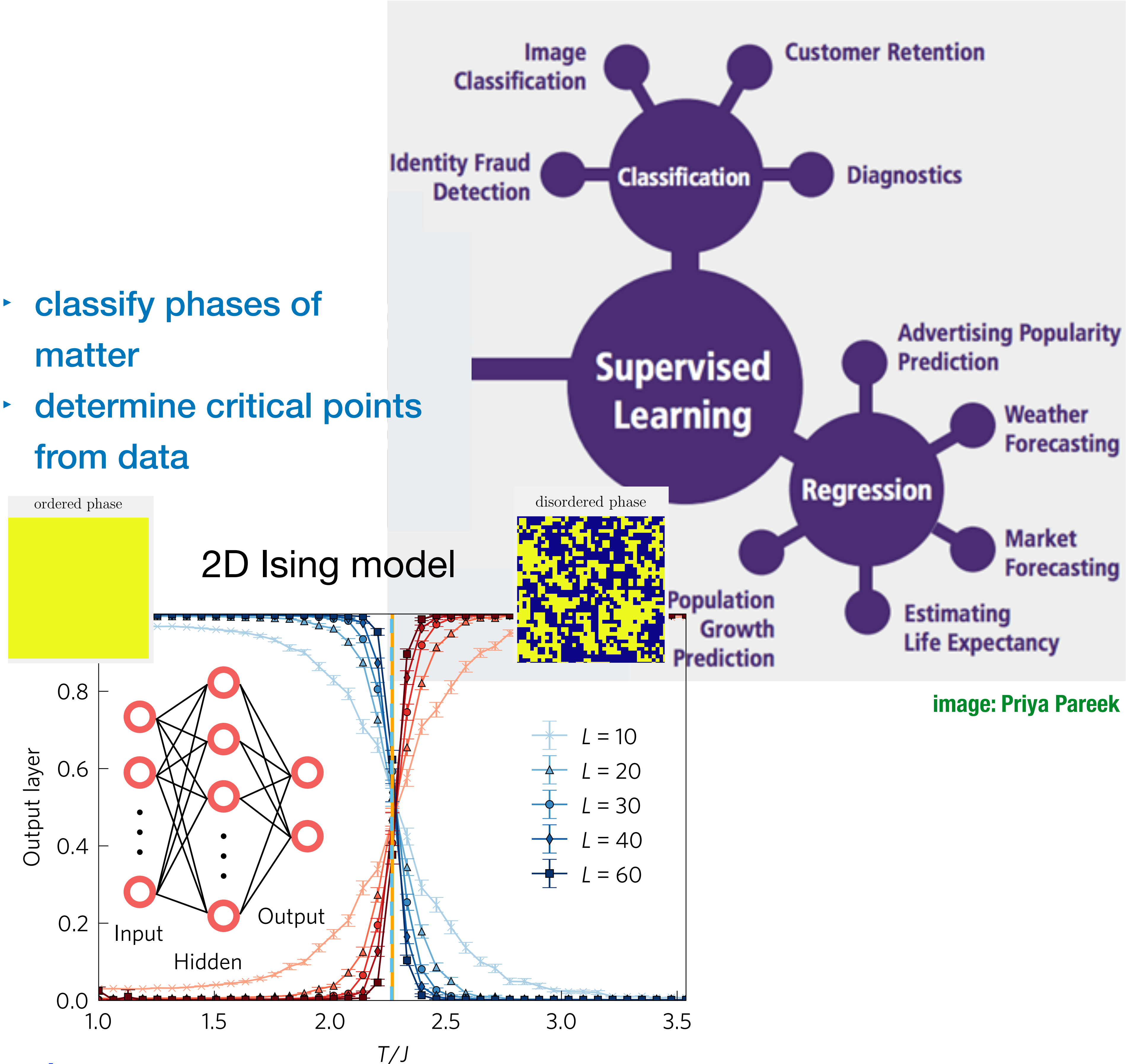


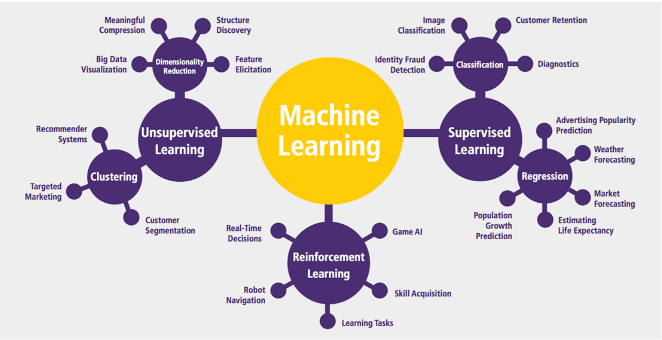
image: Priya Pareek

- classify phases of matter
- determine critical points from data

2D Ising model

Carrasquilla & Melko, Nat. Phys. 2017, v. Nieuwenburg et al, Nat Phys. 2017 + many more

Unsupervised Learning



- learning the distribution that generated data
    - compose music
    - draw paintings
    - write text (Chat GPT)

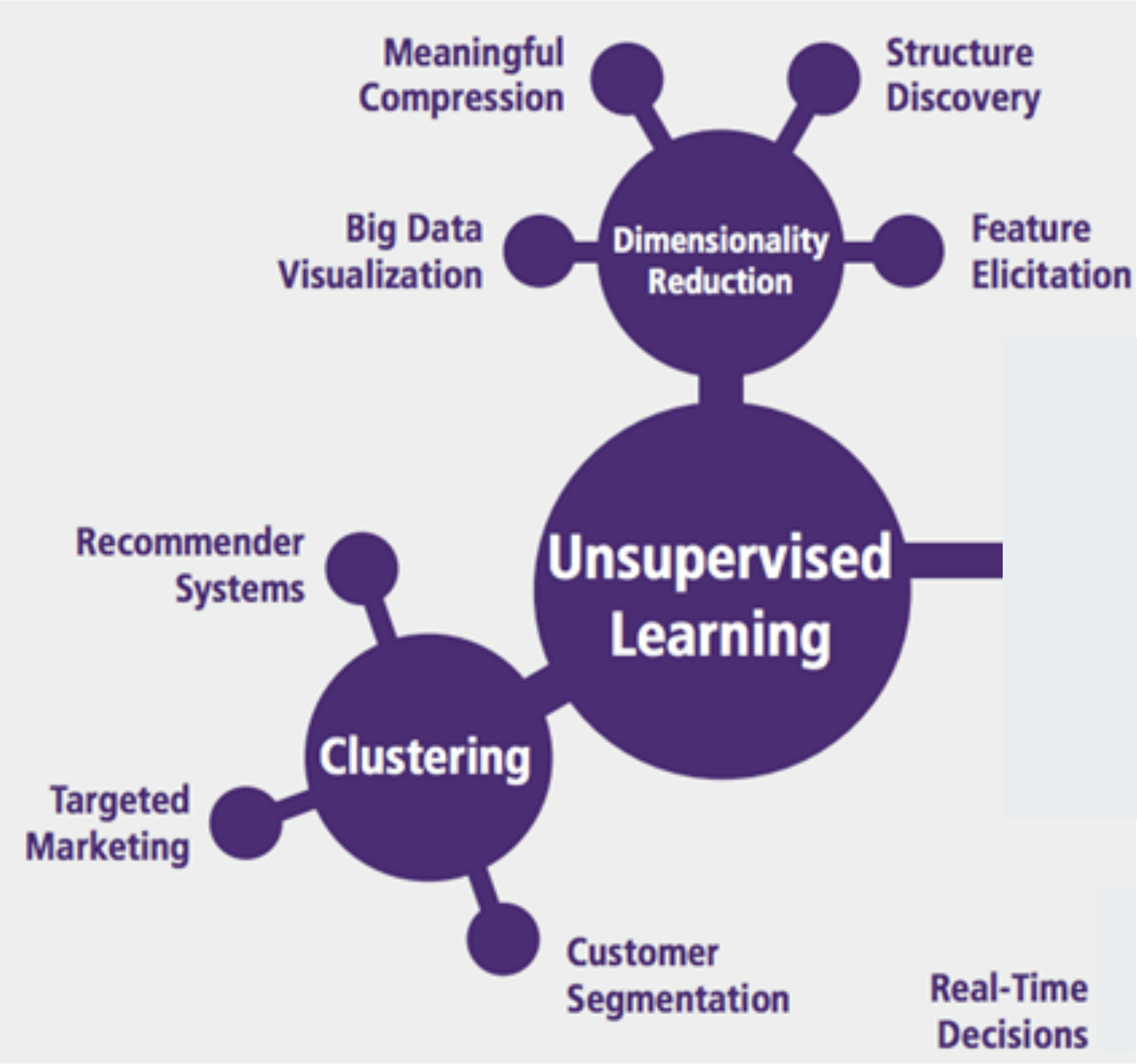
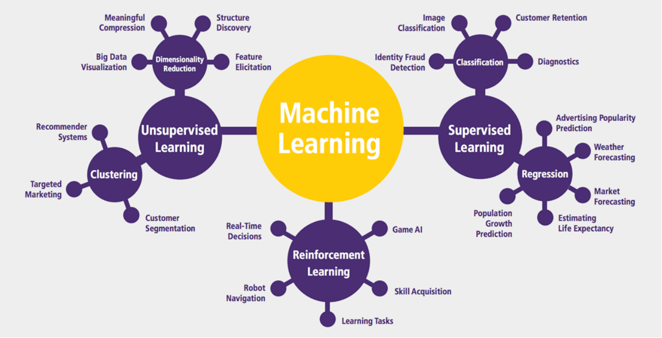


image: Priya Pareek

Unsupervised Learning



- learning the distribution that generated data
  - compose music
  - draw paintings
  - write text (Chat GPT)

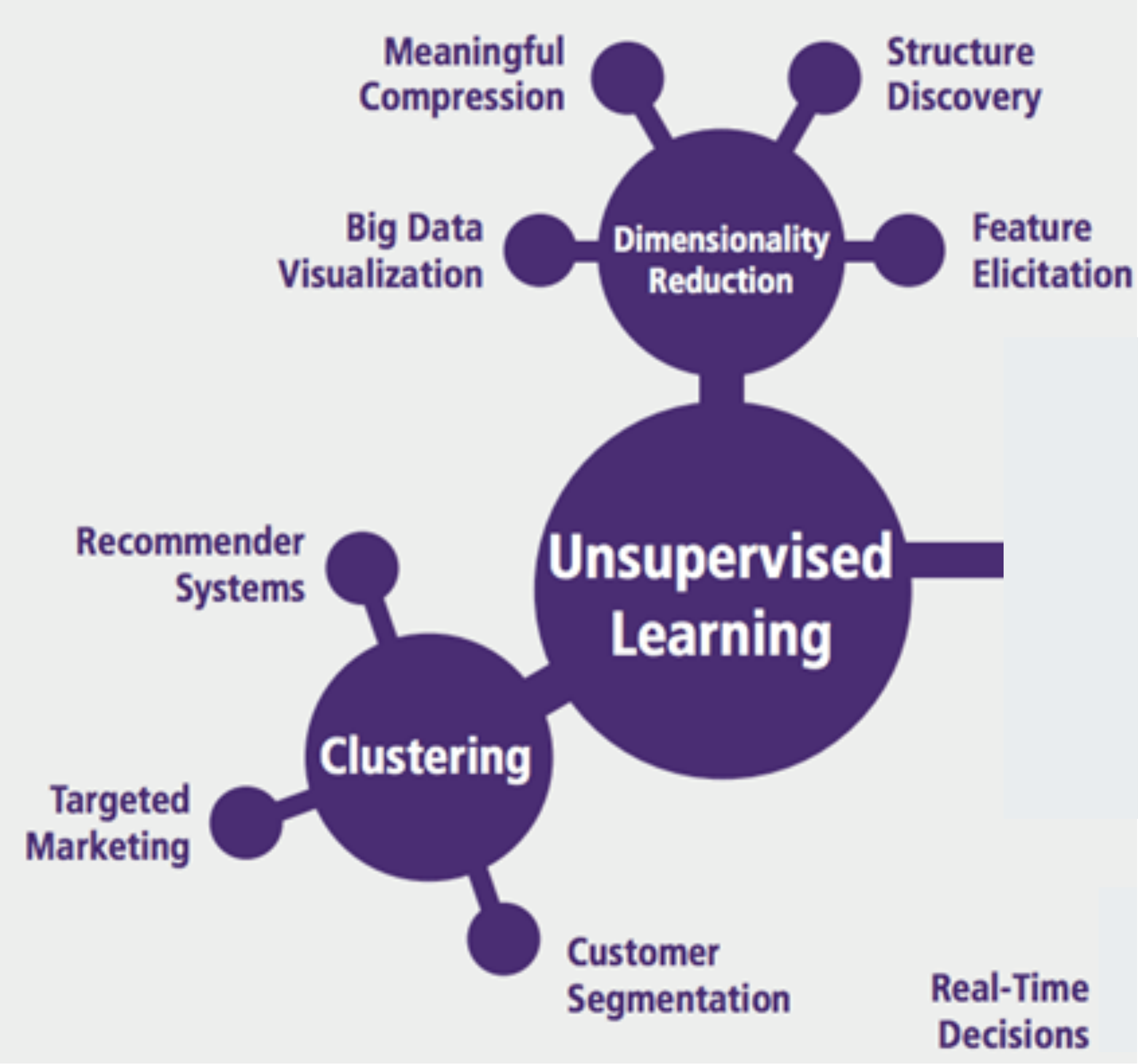
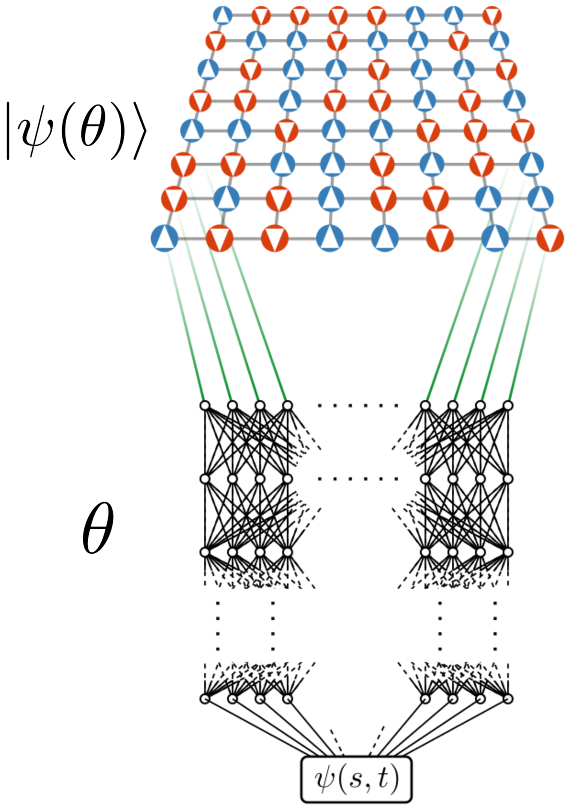


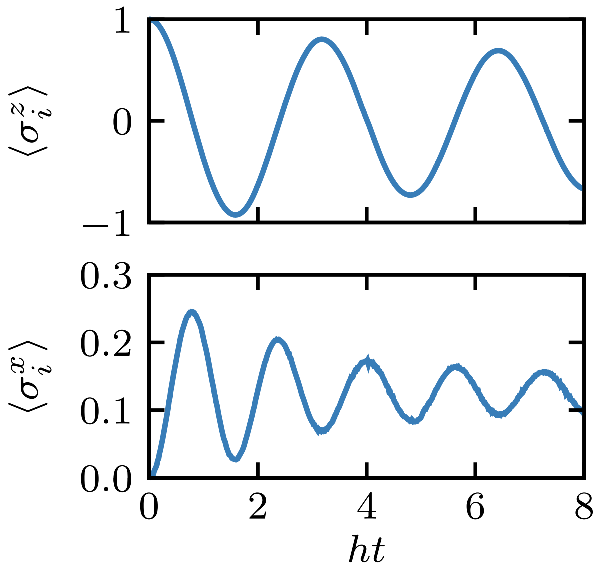
image: Priya Pareek

Neural Quantum States

$$E(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle$$

$$\partial_\theta E(\theta) \overset{!}{=} 0$$



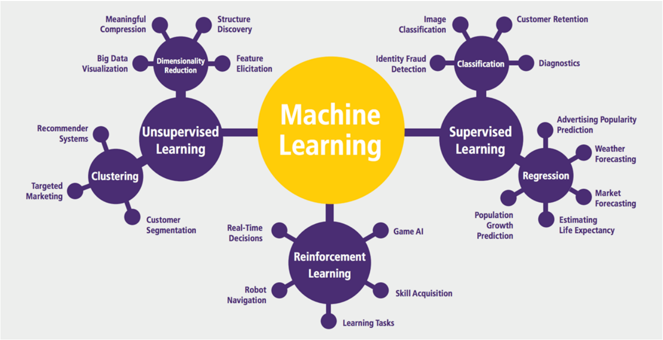


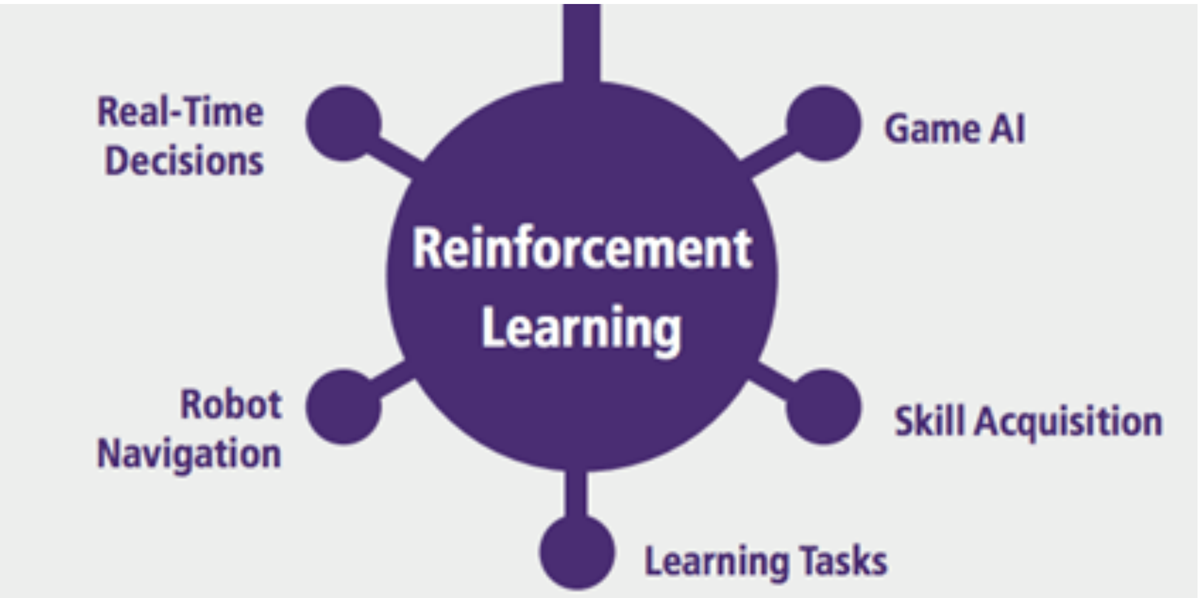
Carleo and Troyer, Science 2017

Schmitt and Heyl, PRL 2020

+ many more

Reinforcement Learning





- learning from experience

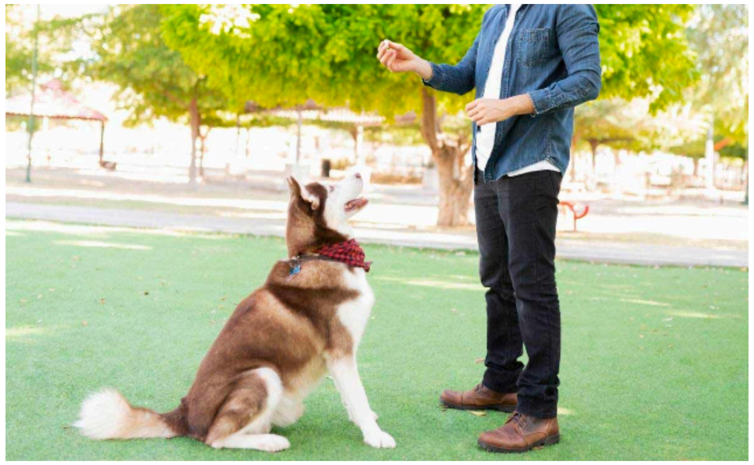


image: Canine Journal

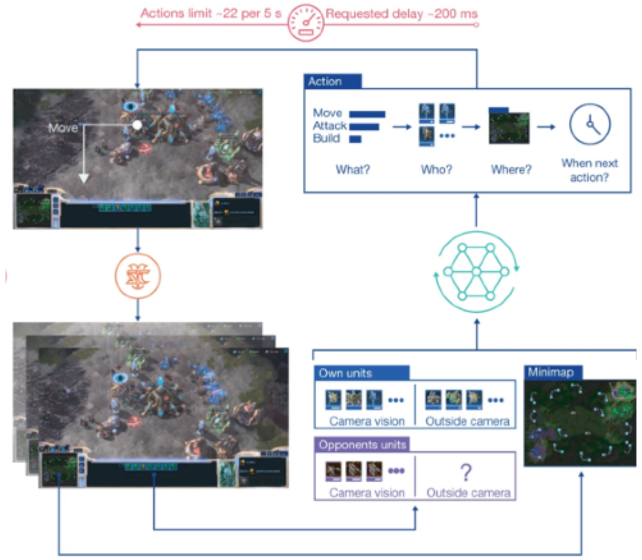
RL *entails* interactive dynamics

What is reinforcement learning used for?

Mastering the game of Go with deep neural networks and tree search

Silver, et. al, Nature 529 484–489 (2016)

Mastering video games (StarCraft II)



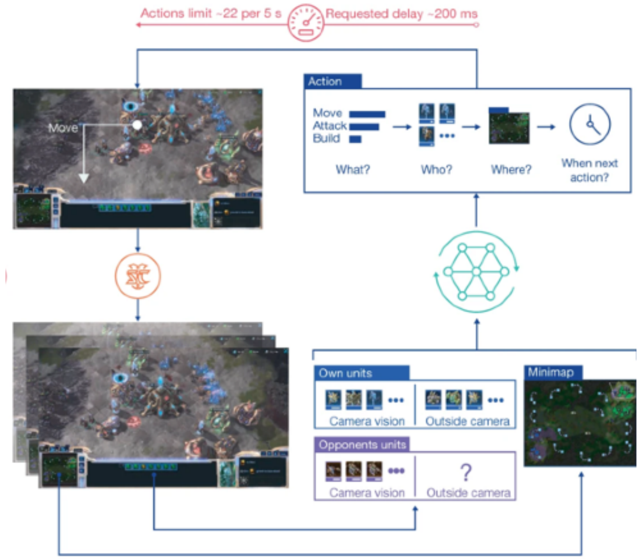
Vinyals, et. al, Nature 350 (2019)

What is reinforcement learning used for?

Mastering the game of Go with deep neural networks and tree search

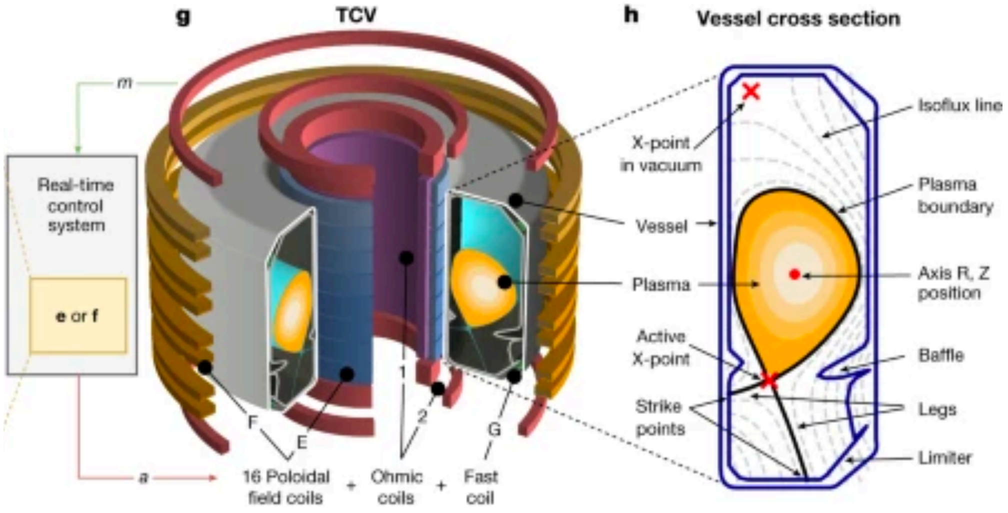
Silver, et. al, Nature 529 484–489 (2016)

Mastering video games (StarCraft II)



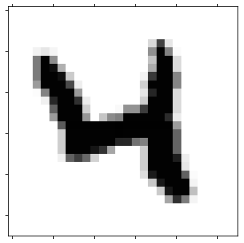
Vinyals, et. al, Nature 350 (2019)

Magnetic control of tokamak plasmas thru deep RL

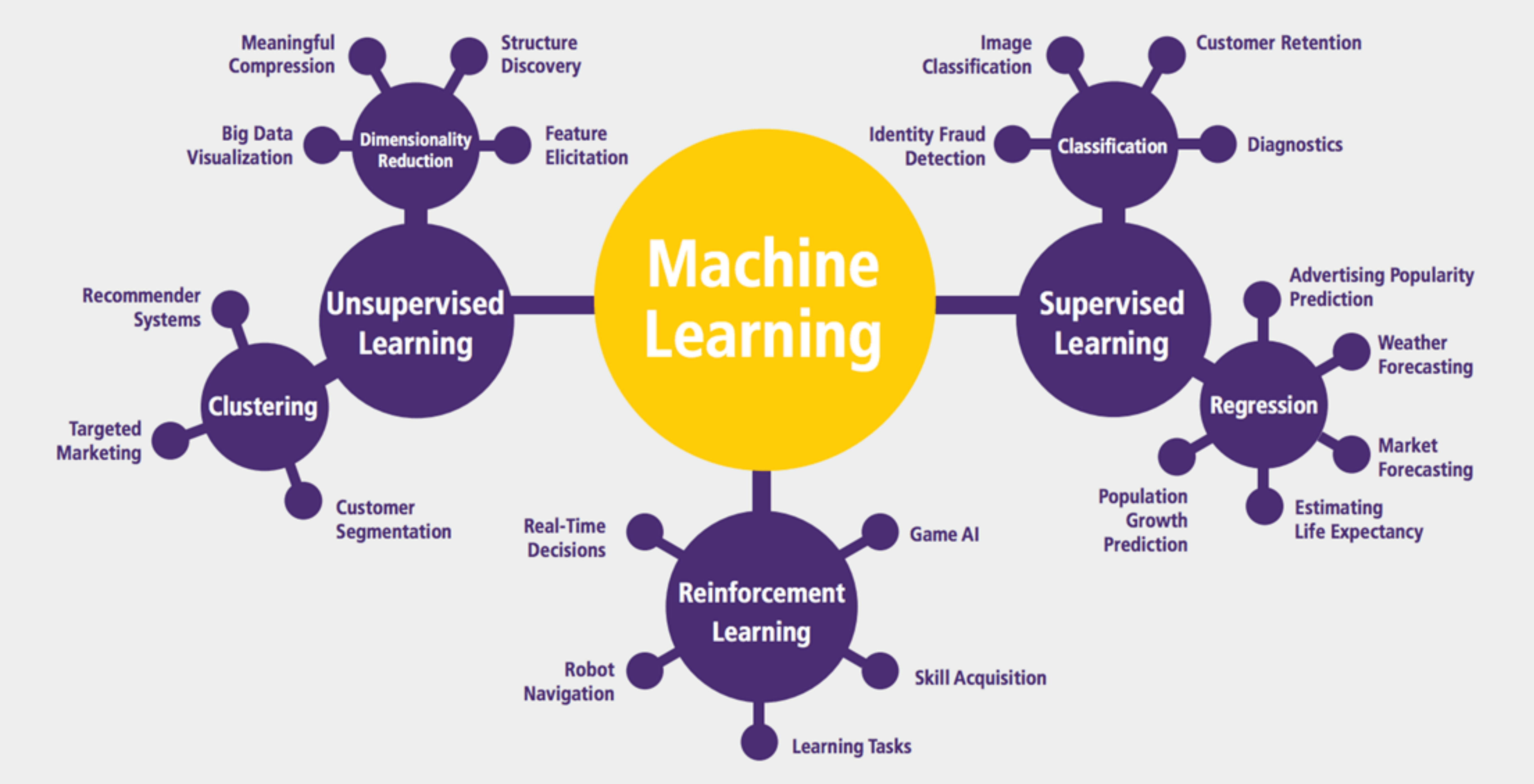


Degrave, et. al, Nature 602 414–419 (2022)

- learning distribution that generates data



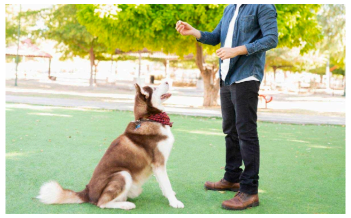
- learning from examples (labeled data)



Machine Learning

- **Unsupervised Learning**
  - **Dimensionality Reduction**
    - Meaningful Compression
    - Structure Discovery
    - Big Data Visualization
    - Feature Elicitation
  - **Clustering**
    - Recommender Systems
    - Targeted Marketing
    - Customer Segmentation
- **Supervised Learning**
  - **Classification**
    - Image Classification
    - Customer Retention
    - Identity Fraud Detection
    - Diagnostics
  - **Regression**
    - Advertising Popularity Prediction
    - Weather Forecasting
    - Market Forecasting
    - Population Growth Prediction
    - Estimating Life Expectancy
- **Reinforcement Learning**
  - Real-Time Decisions
  - Game AI
  - Robot Navigation
  - Skill Acquisition
  - Learning Tasks

image: Priya Pareek

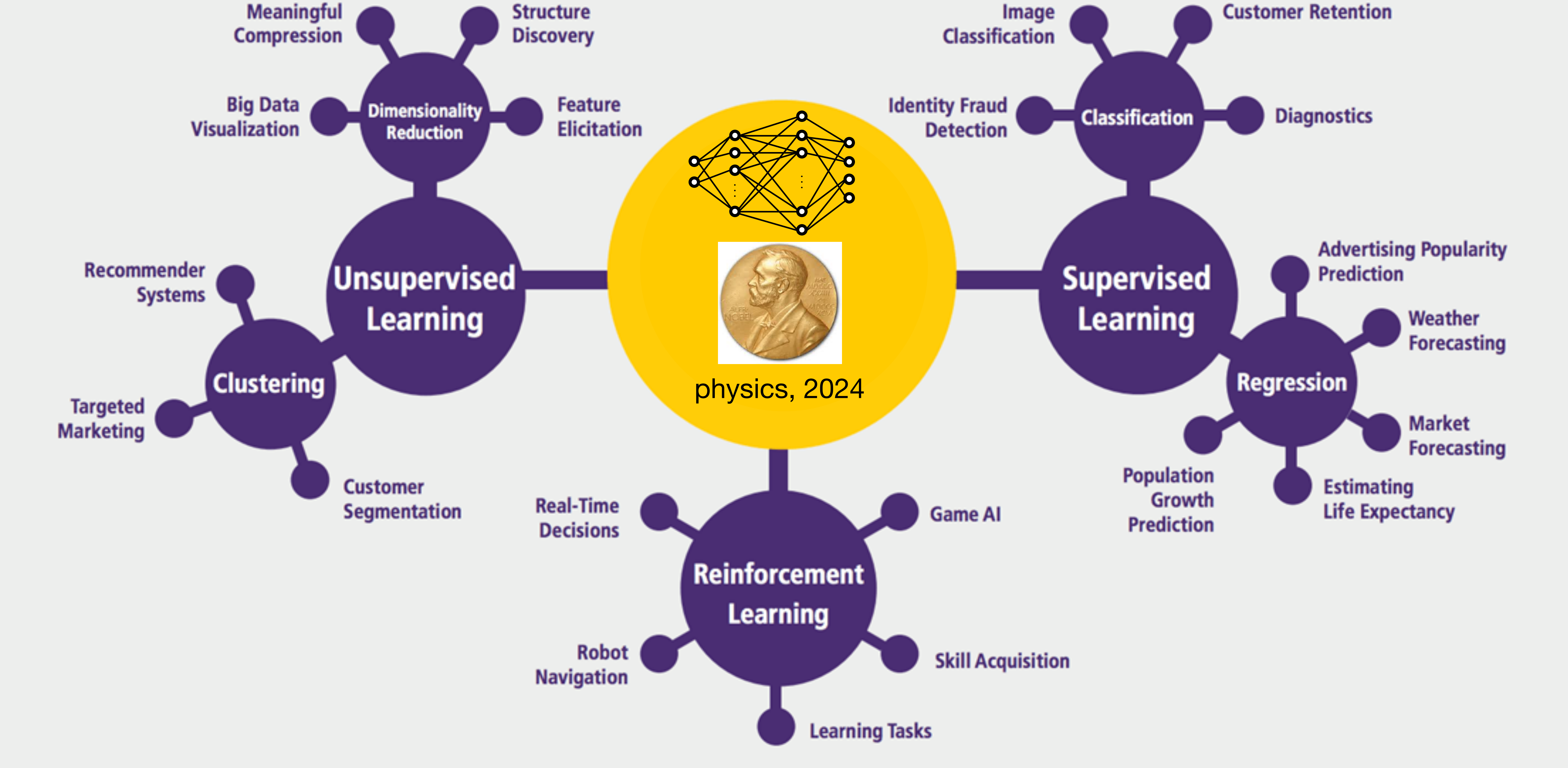


- **learning from experience**

- learning distribution that generates data

chemistry, 2024

- learning from examples (labeled data)



physics, 2024

image: Priya Pareek

- **learning from experience**

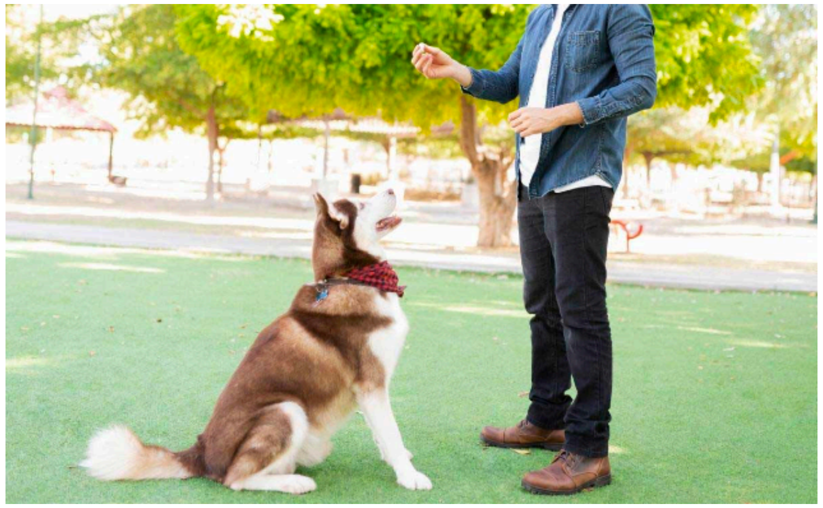
Outline

Part 1

✓ Reinforcement learning (RL) in quantum physics

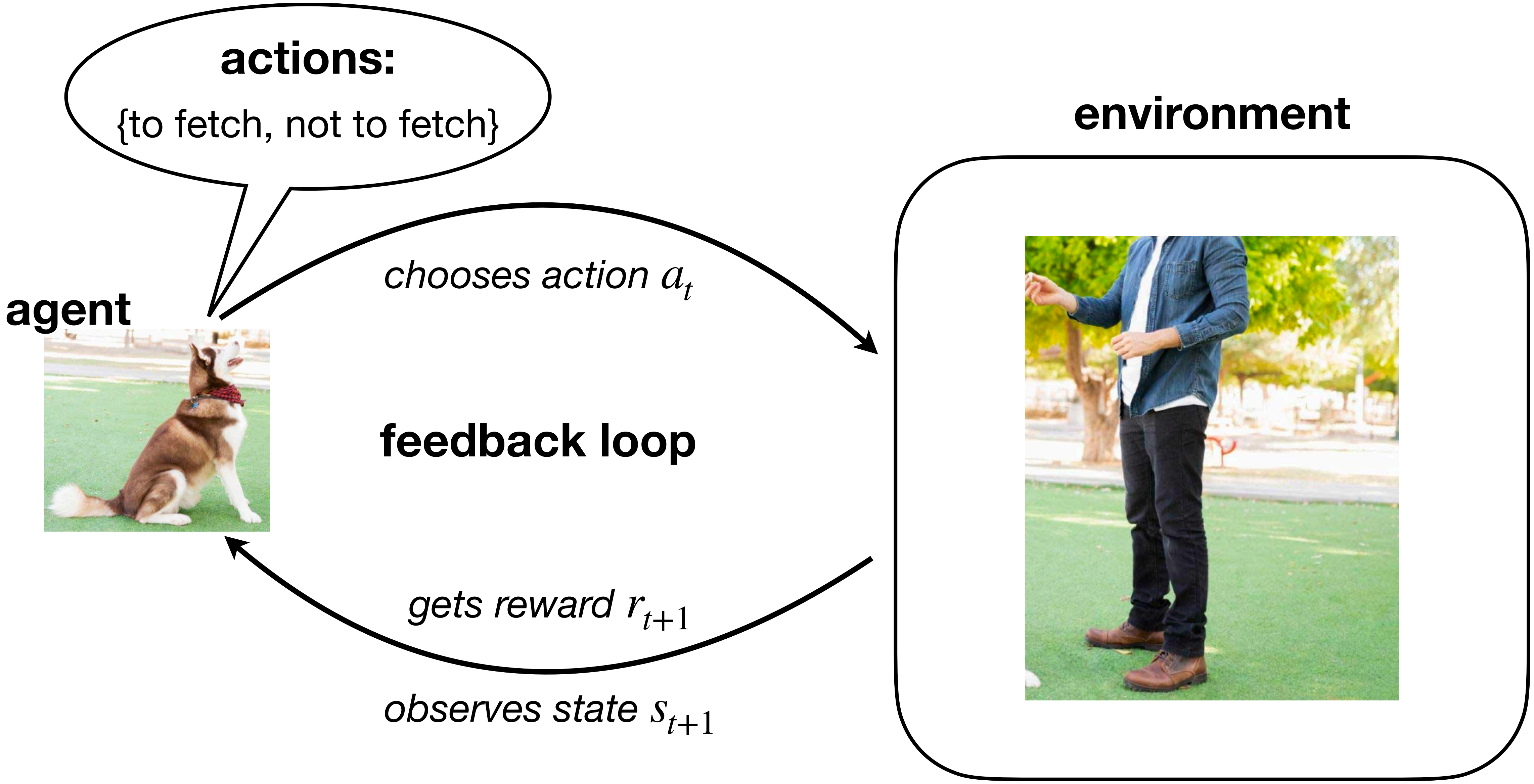
  - RL as a branch of machine learning

- Applications of RL
  - hallmark applications of RL
  - applications in quantum technologies



- RL framework in a nutshell
  - environment, states, actions, rewards
  - RL algorithms

Reinforcement learning (RL) in a nutshell



actions:

{to fetch, not to fetch}

agent

environment

chooses action a_t

feedback loop

gets reward r_{t+1}

observes state s_{t+1}

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

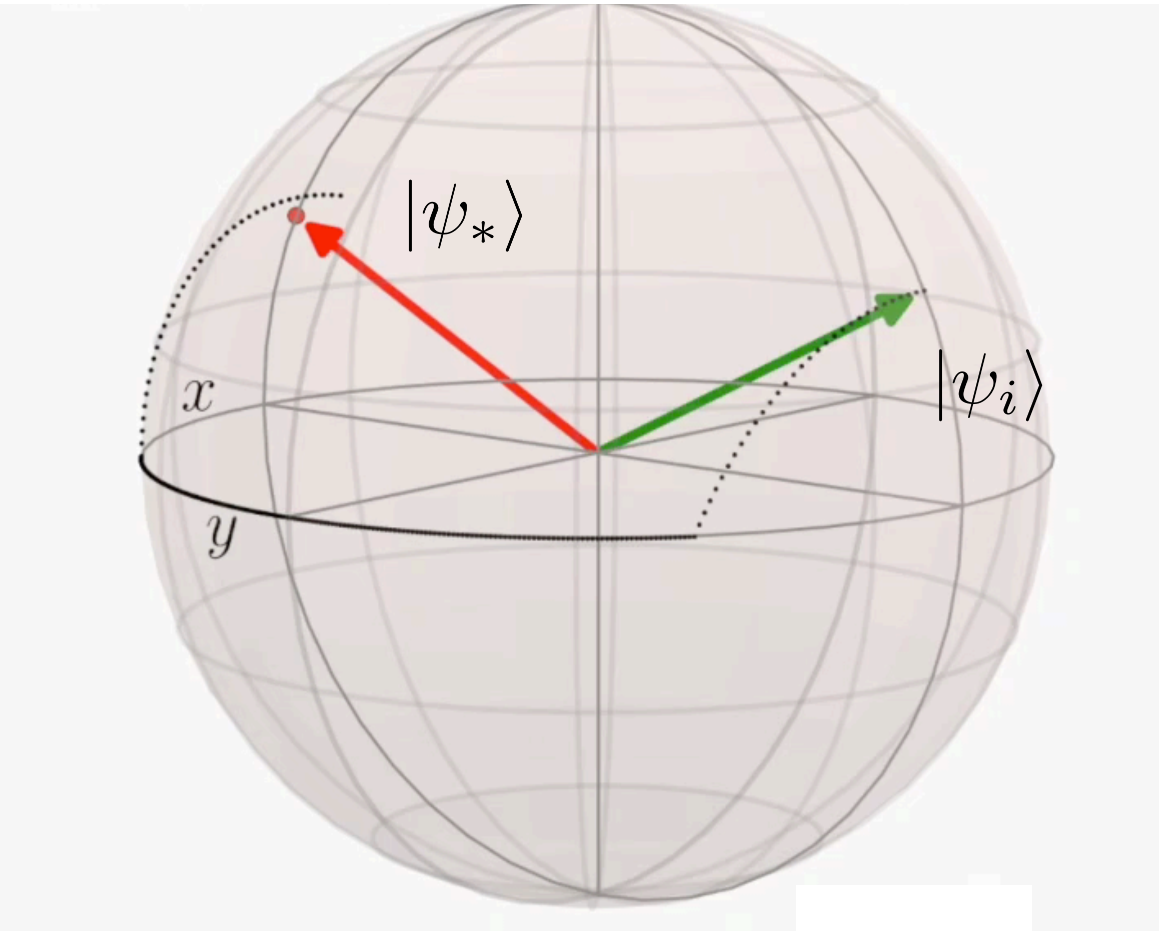
Applications of RL in Quantum Physics

- quantum control

$$H(t) = -\frac{1}{2}\left(Z + h_x(t)X\right)$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Bloch sphere



state preparation = fidelity optimization

$$F_h(t) = |\langle\psi_*|\psi(t)\rangle|^2$$

$$|\psi(t)\rangle = \mathcal{T}\mathrm{e}^{-i\int_0^t \mathrm{d}t' H(t')}|\psi_i\rangle$$

$$h_x(t) = ?$$

MB et al, PRX 8 031086 (2018)

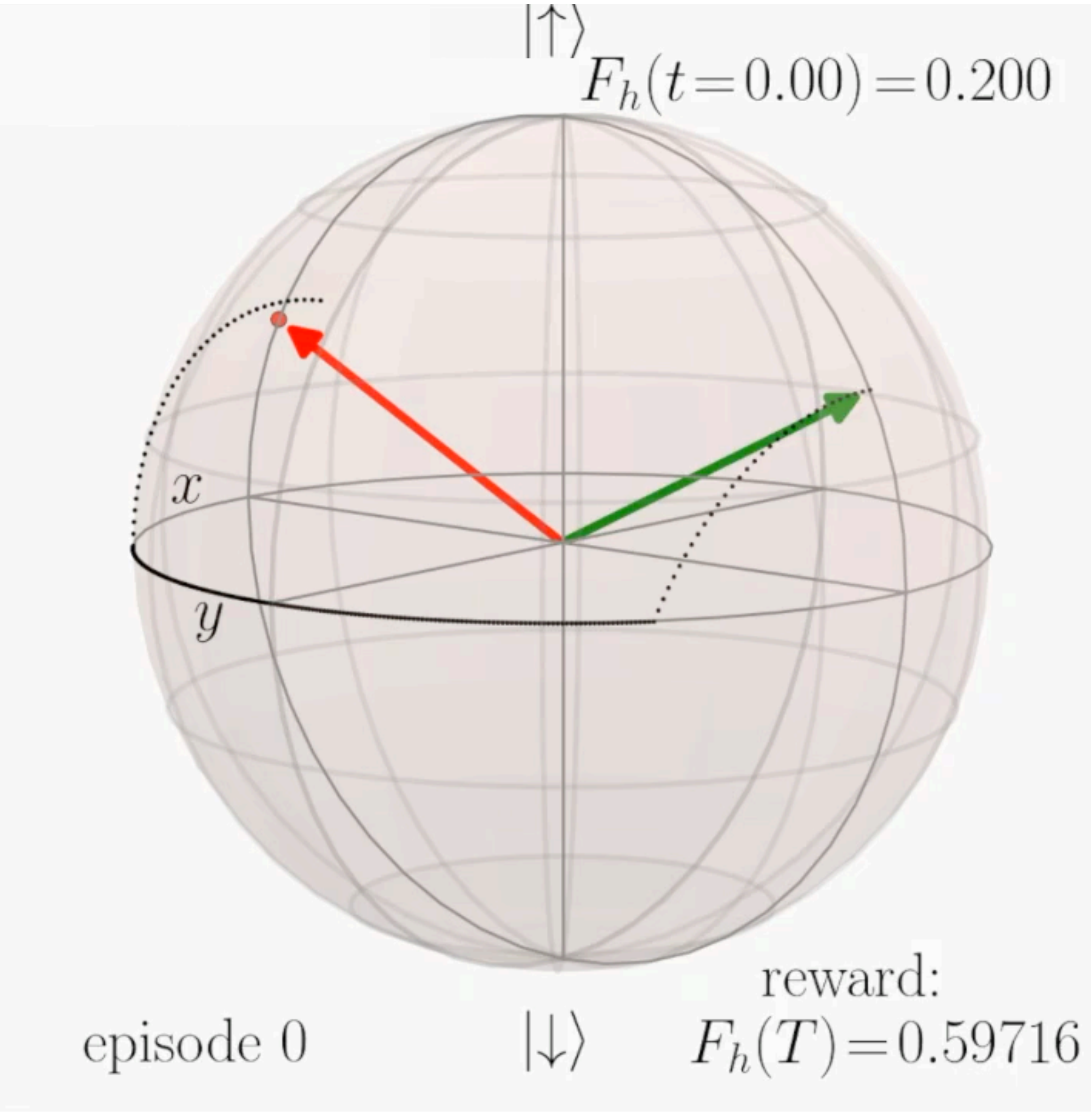
Applications of RL in Quantum Physics

- quantum control

$$H(t) = -\frac{1}{2}\left(Z + h_x(t)X\right)$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Bloch sphere



state preparation = fidelity optimization

$$F_h(t) = |\langle\psi_*|\psi(t)\rangle|^2$$

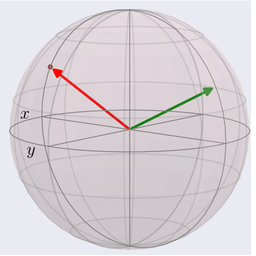
$$|\psi(t)\rangle = \mathcal{T}\mathrm{e}^{-i\int_0^t \mathrm{d}t' H(t')}|\psi_i\rangle$$

$$h_x(t) = ?$$

MB et al, PRX 8 031086 (2018)

Applications of RL in Quantum Physics

- quantum control

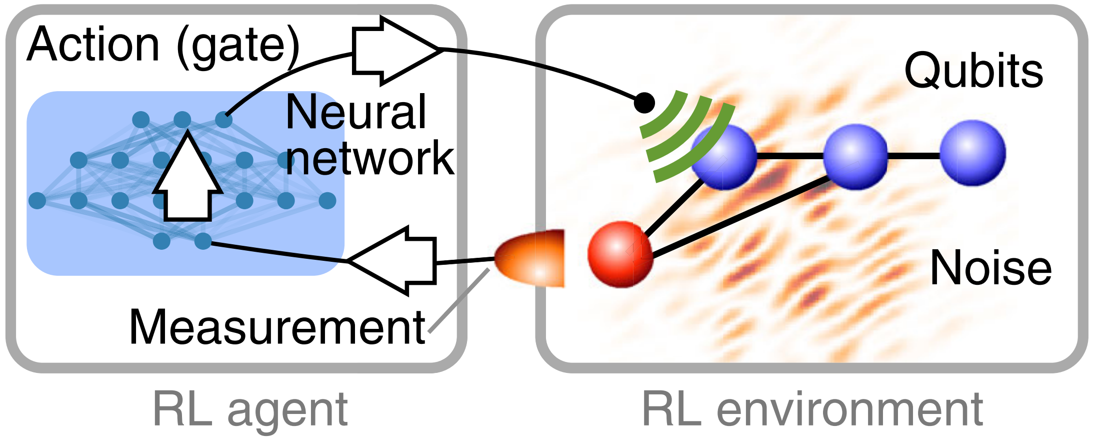


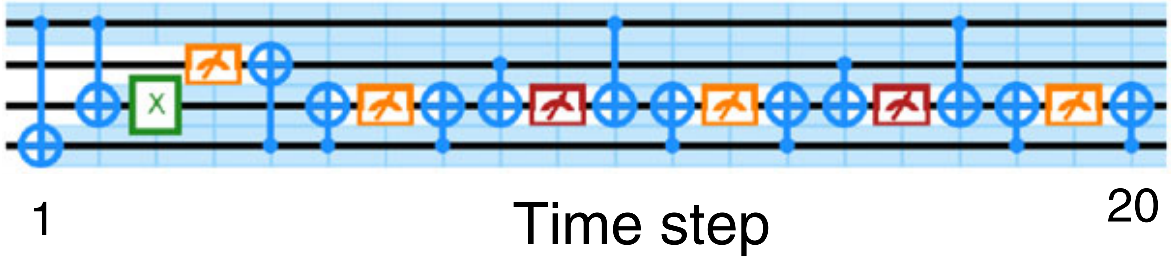
MB et al, PRX 8 031086 (2018)
Niu et al, npj 5 33 (2019)
Sivak et al, PRX 12, 011059 (2022)
Gispen et al, MSML (2021)

Reuer, Nat Comm 14 7138 (2023)
Yao et al, PRX 11 (3), 031070 (2021)
Porotti, Comm Phys 2 (2019)
Dalgaard et al, npj 6 6 (2020)

+ many more

- quantum error correction
  - **task:** find error correcting code that protects qubits from decoherence



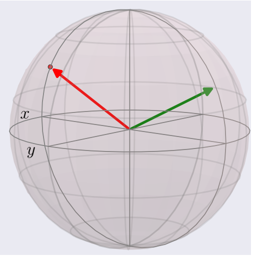


Fössel et al, PRX 8 031086 (2018)

Olle et al, arXiv:2311.04750

Applications of RL in Quantum Physics

- quantum control

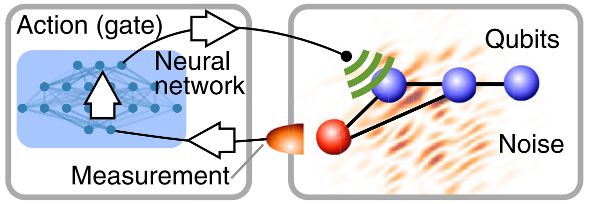


MB et al, PRX 8 031086 (2018)
Niu et al, npj 5 33 (2019)
Sivak et al, PRX 12, 011059 (2022)
Gispen et al, MSML (2021)

Reuer, Nat Comm 14 7138 (2023)
Yao et al, PRX 11 (3), 031070 (2021)
Porotti, Comm Phys 2 (2019)
Dalgaard et al, npj 6 6 (2020)

+ many more

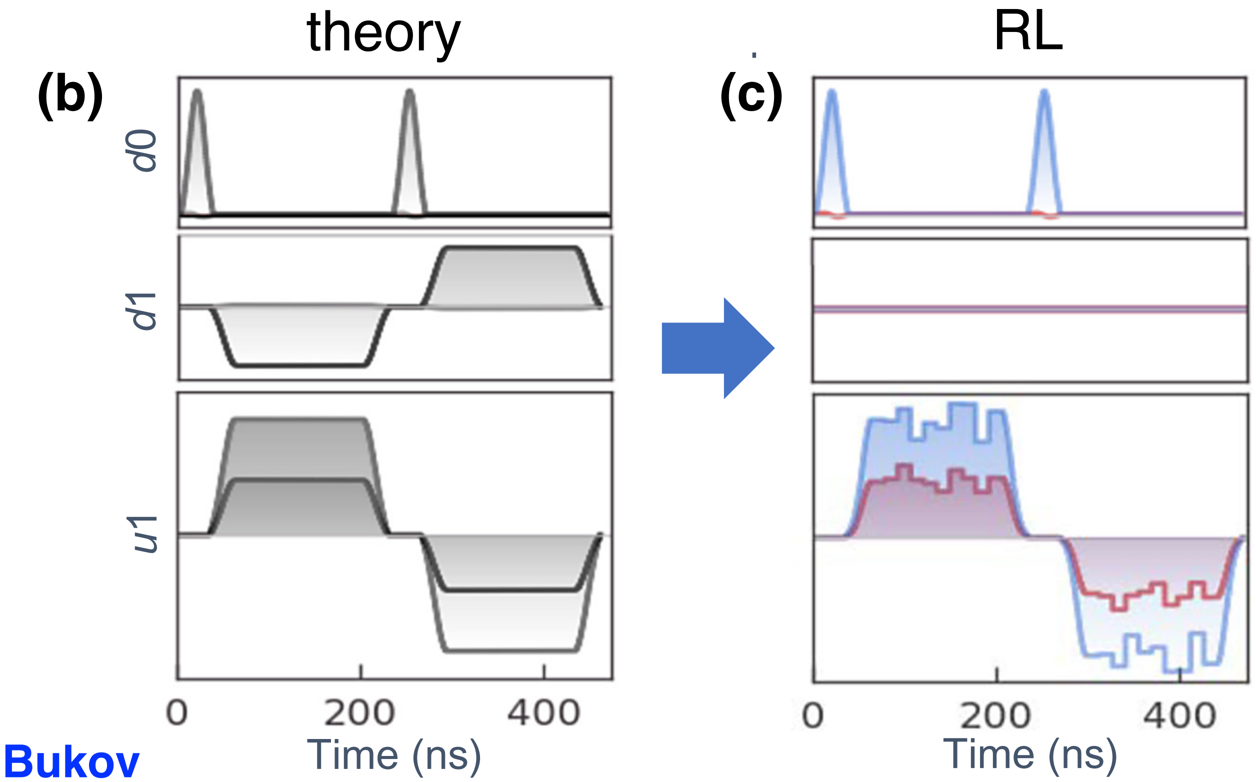
- quantum error correction



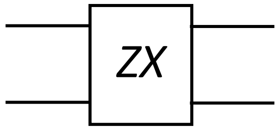
Fössel et al, PRX 8 031086 (2018)
Andreasson et al, Quantum 3 183 (2019)
Sweke et al, ML Sci Tech 2 025005 (2020)
Sivak et al, Nature 616 50-55 (2023)
Olle et al, arXiv:2311.04750

+ many more

- quantum gate design
    - **task:** find high-fidelity pulses that emulate gates on a quantum computer



Baum et al, PRX Quantum 2, 040324 (2021)



cross resonance gate

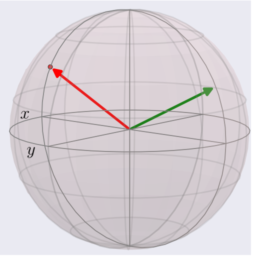
— IBM $(1.41 \pm 0.06) \times 10^{-2}$

— DRL $(6.50 \pm 0.61) \times 10^{-3}$

infidelity / error rate

Applications of RL in Quantum Physics

- quantum control

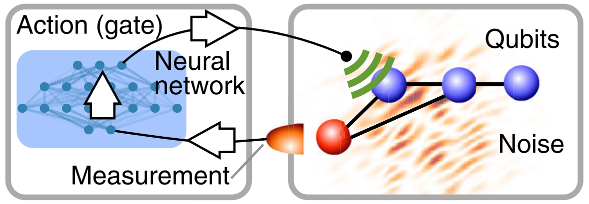


MB et al, PRX 8 031086 (2018)
Niu et al, npj 5 33 (2019)
Sivak et al, PRX 12, 011059 (2022)
Gispen et al, MSML (2021)

Reuer, Nat Comm 14 7138 (2023)
Yao et al, PRX 11 (3), 031070 (2021)
Porotti, Comm Phys 2 (2019)
Dalgaard et al, npj 6 6 (2020)

+ many more

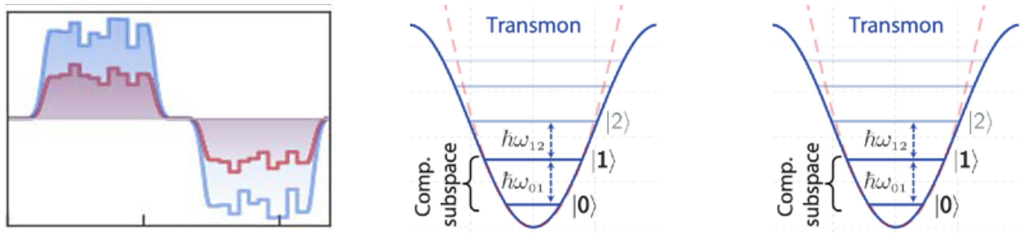
- quantum error correction



Fössel et al, PRX 8 031086 (2018)
Andreasson et al, Quantum 3 183 (2019)
Sweke et al, ML Sci Tech 2 025005 (2020)
Sivak et al, Nature 616 50-55 (2023)
Olle et al, arXiv:2311.04750

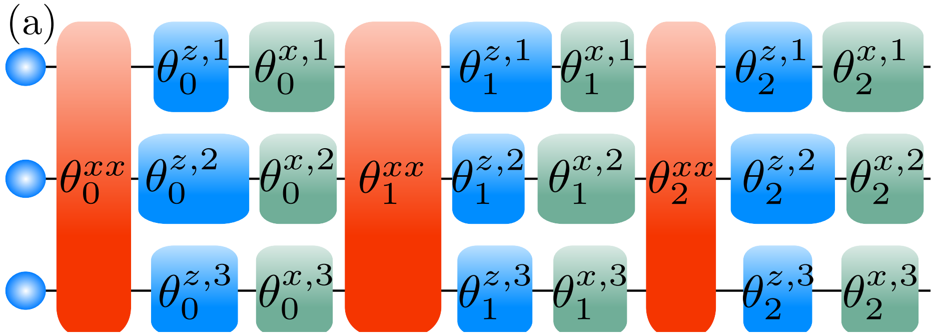
+ many more

- quantum gate design

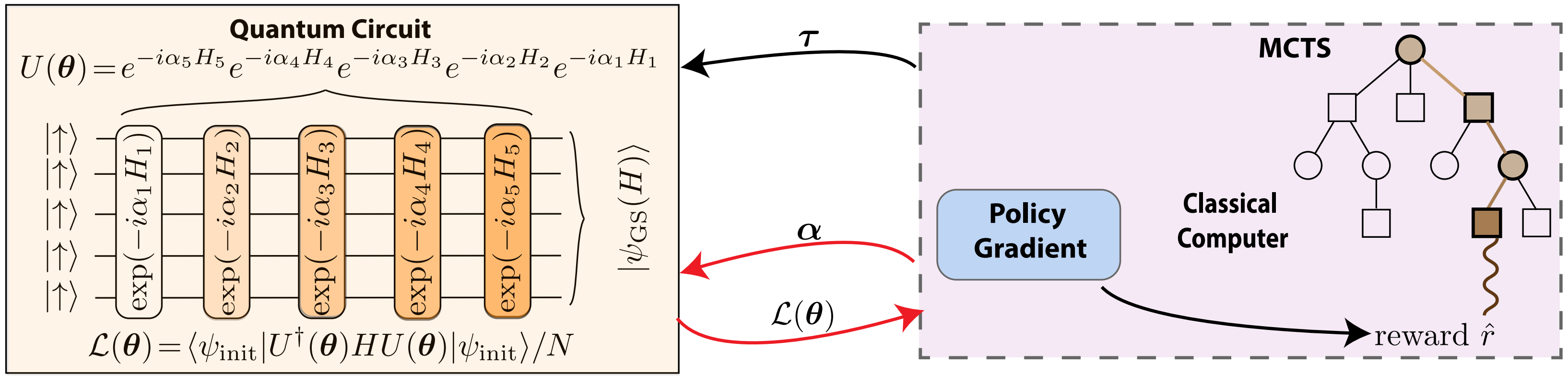


Baum et al, PRX Quantum 2, 040324 (2021)
Nguyen et al, ML Sci & Tech, 5, 025066 (2024)

- quantum circuit design and synthesis



Bolens et al, PRL 2021

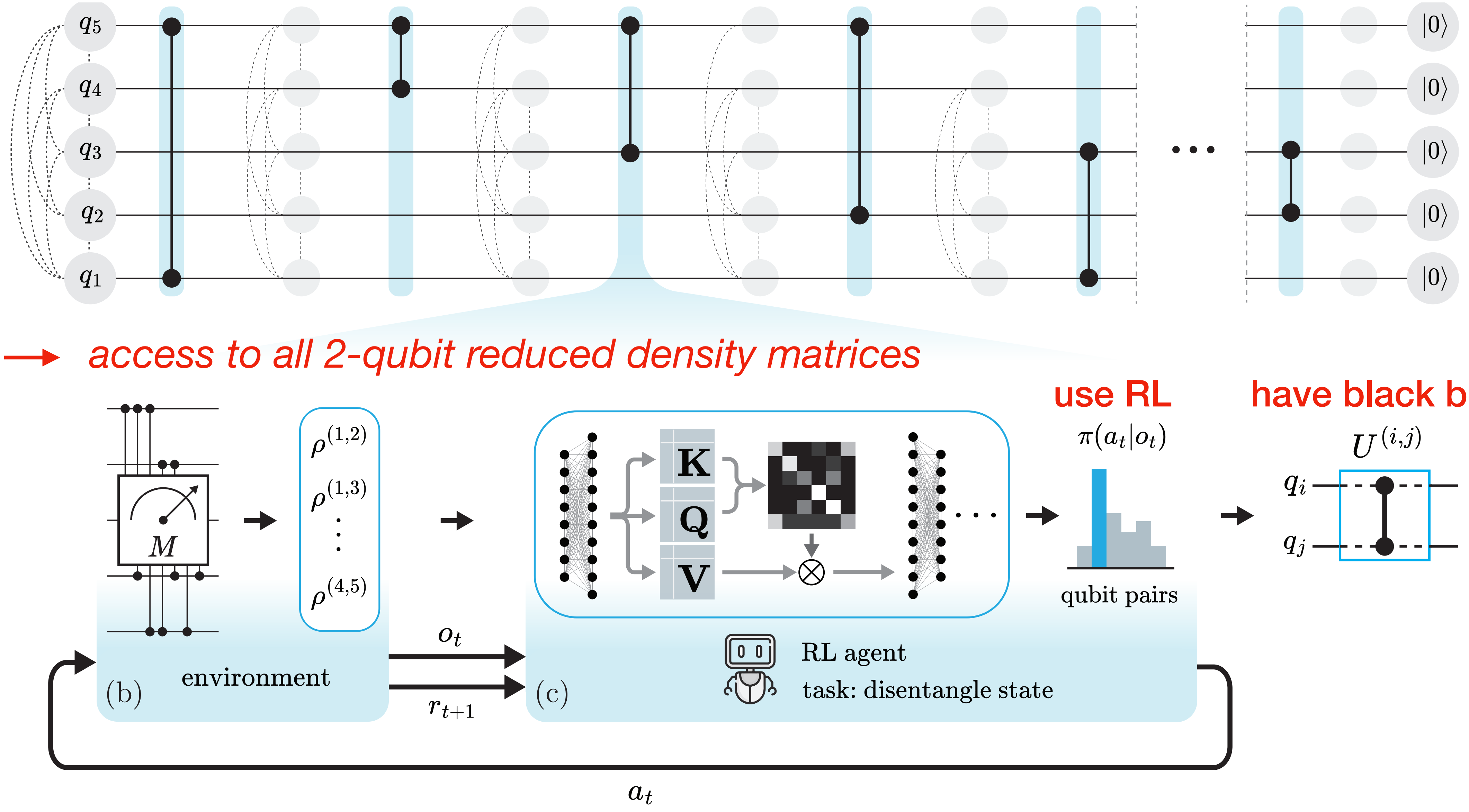


Yao et al, MSML 2019, 2021, 2022

+ many more

Applications of RL in Quantum Physics

- quantum control
- quantum gate design
- quantum error correction
- quantum circuit design and synthesis
  - construct disentangling circuits



→ *access to all 2-qubit reduced density matrices*

use RL

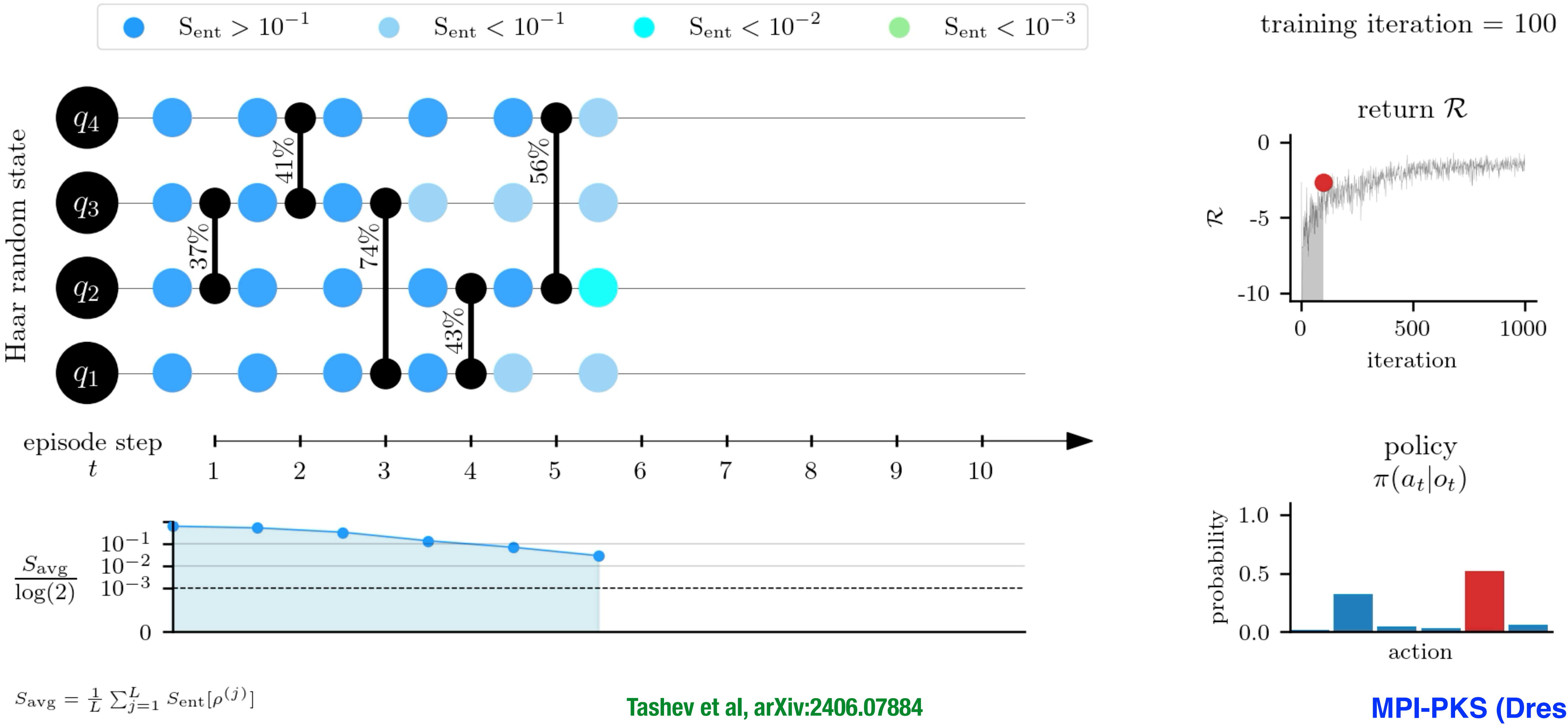
have black box

Tashev et al, arXiv:2406.07884

Applications of RL in Quantum Physics

- quantum control
- quantum gate design
- quantum error correction
- quantum circuit design and synthesis
  - construct disentangling circuits

then it starts minimizing the overall number of gates applied..



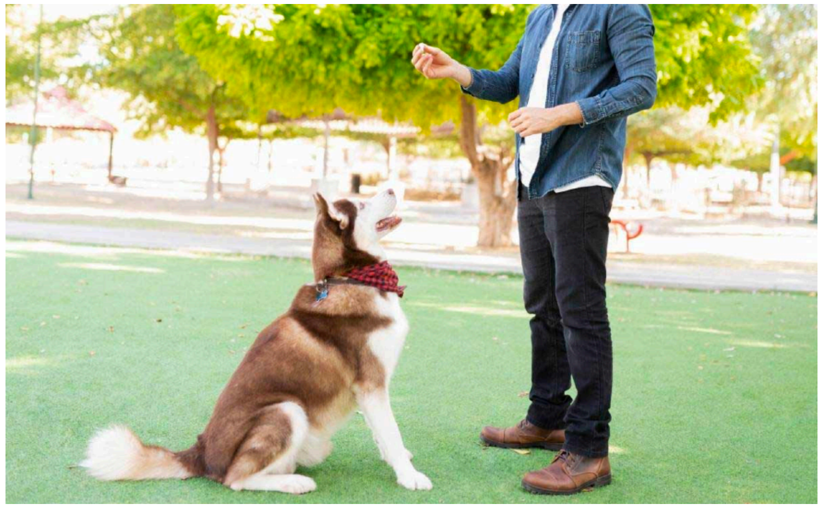
Tashev et al, arXiv:2406.07884

MPI-PKS (Dresden)

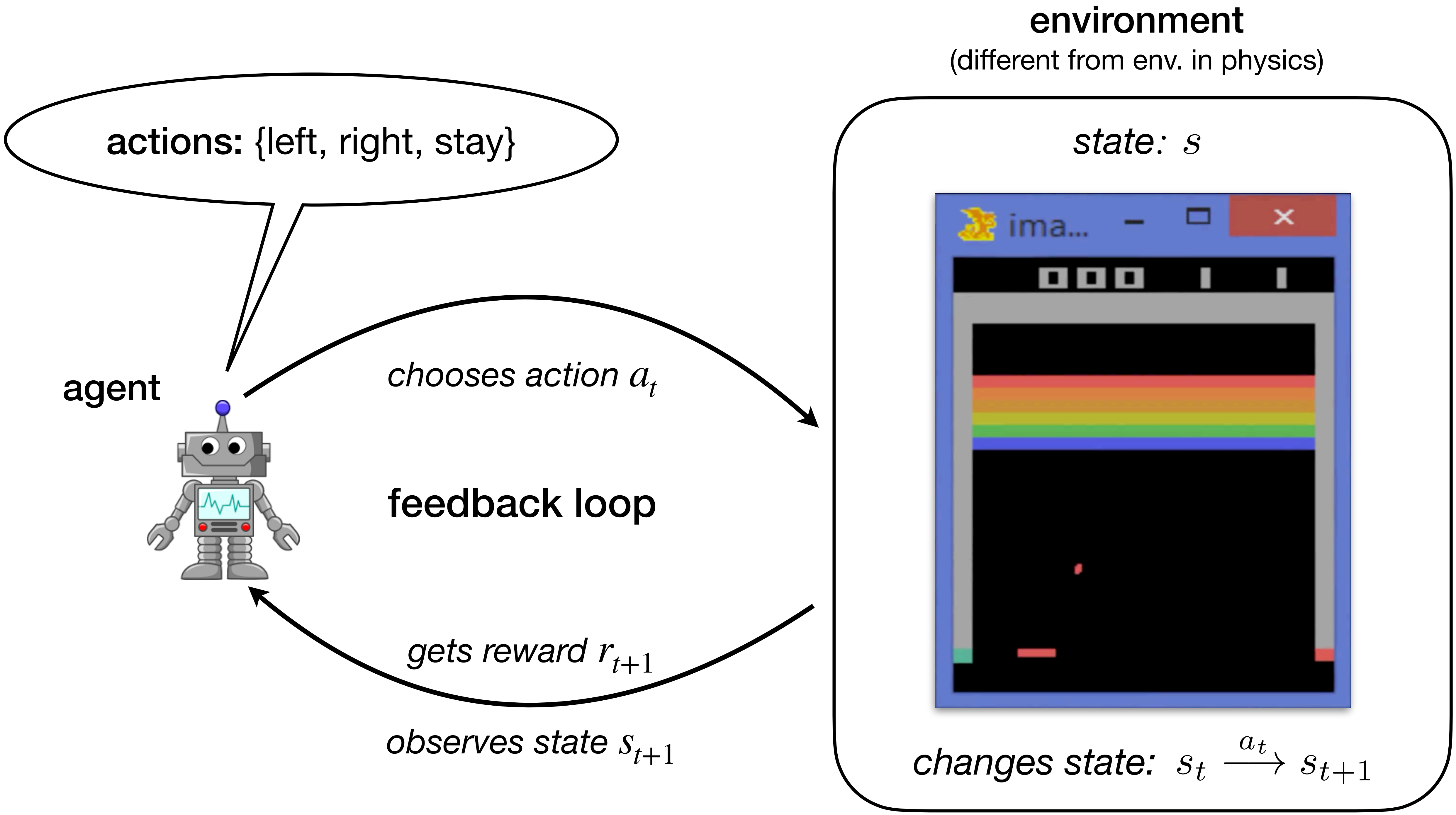
Outline

Part 1

- ✓ Reinforcement learning (RL) in quantum physics
  - RL as a branch of machine learning
- ✓ Applications of RL
  - hallmark applications of RL
  - applications in quantum technologies
- RL framework in a nutshell
  - environment, states, actions, rewards
  - RL algorithms



Reinforcement Learning (RL) formalism



environment

(different from env. in physics)

actions: {left, right, stay}

state: s

agent

chooses action a_t

feedback loop

gets reward r_{t+1}

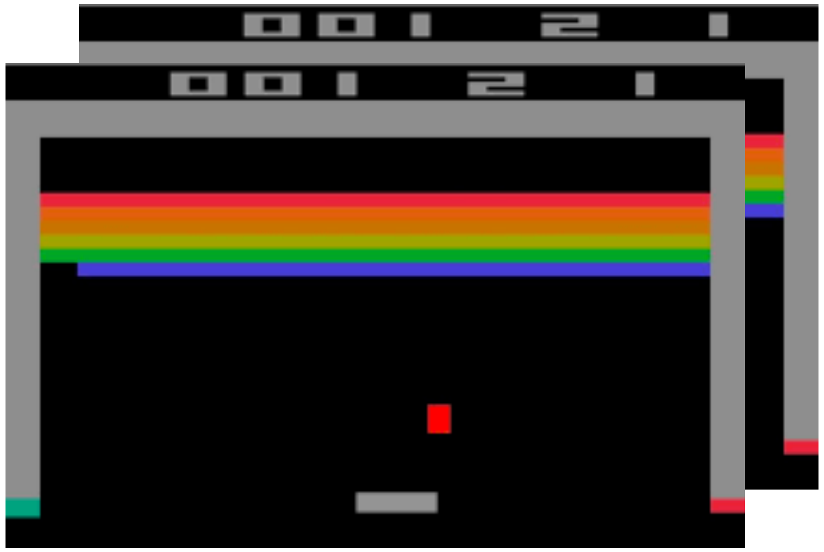
observes state s_{t+1}

changes state: $s_t \xrightarrow{a_t} s_{t+1}$

Mnih et al., Nature 518 (2015) [Google DeepMind]

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

RL in a Nutshell

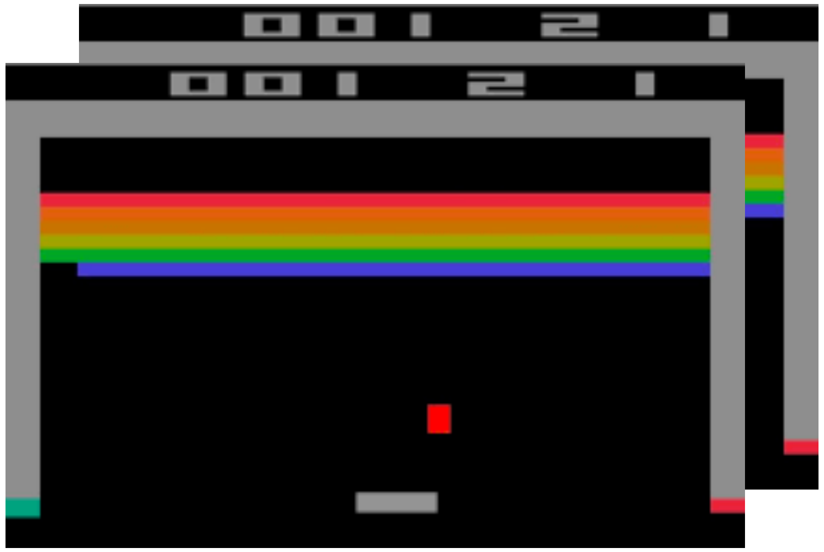


- RL formalism
    - action space: $\mathscr{A} = \{\text{left, stay, right}\}$
    - state space: $\mathscr{S}$ pixelized image of the screen
    - reward function: $r =$ score

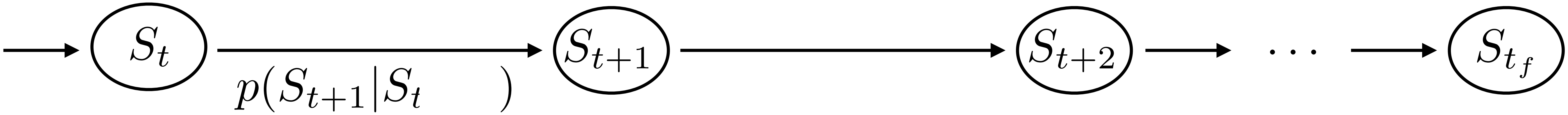
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

RL in a Nutshell

- RL formalism
  - action space: $\mathscr{A} = \{\text{left, stay, right}\}$
  - state space: $\mathscr{S}$ pixelized image of the screen
  - reward function: $r =$ score



- RL episode is a Markov decision process



  - transition probability: $p(S_{t+1}|S_t, \quad)$

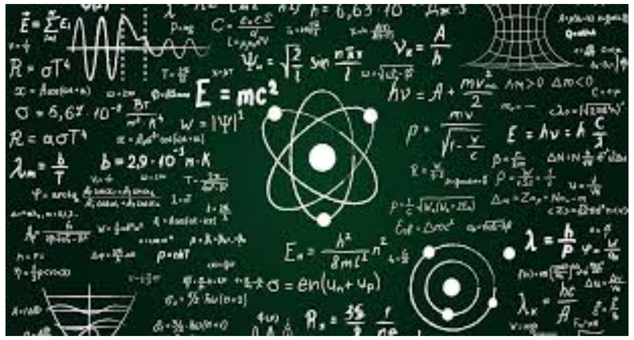
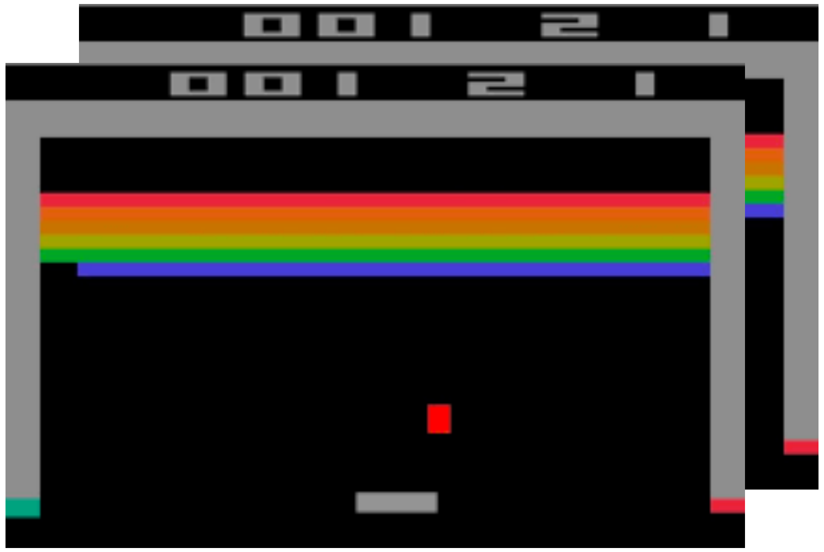


image: Kardashev Scale Wiki

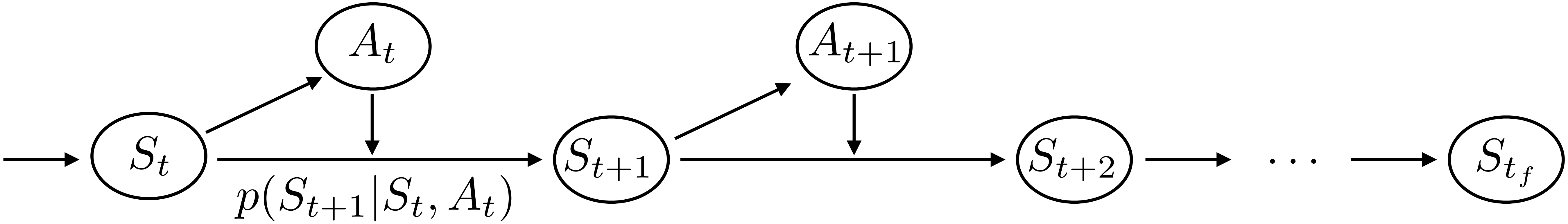
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

RL in a Nutshell



- RL formalism
  - action space: $\mathscr{A} = \{\text{left, stay, right}\}$
  - state space: $\mathscr{S}$ pixelized image of the screen
  - reward function: $r =$ score

- RL episode is a Markov **decision** process



  - transition probability: $p(S_{t+1}\,|\,S_t, A_t)$

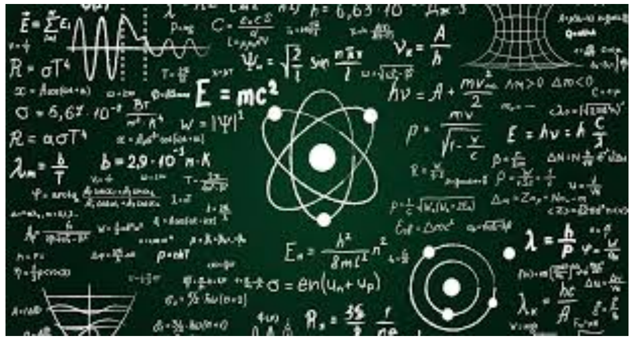
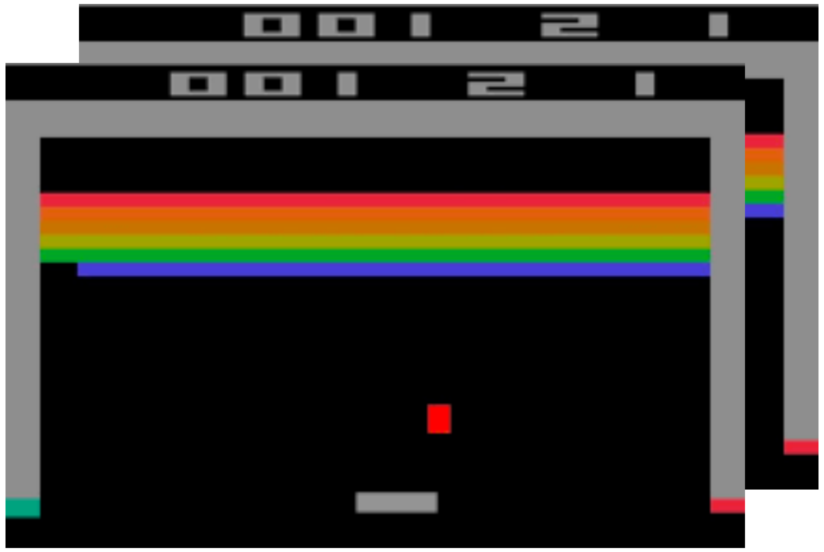


image: Kardashev Scale Wiki

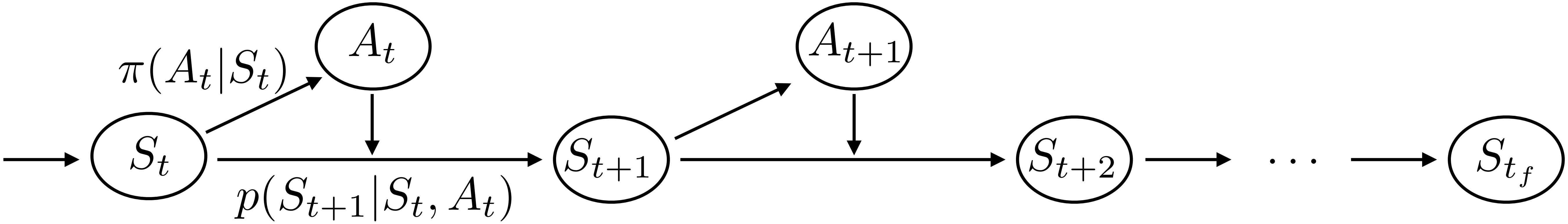
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

RL in a Nutshell

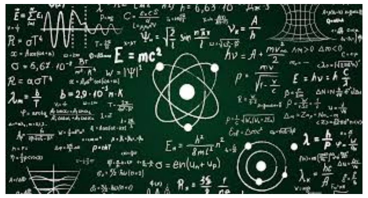
- RL formalism
  - action space: $\mathscr{A} = \{\text{left, stay, right}\}$
  - state space: $\mathscr{S}$ pixelized image of the screen
  - reward function: $r =$ score



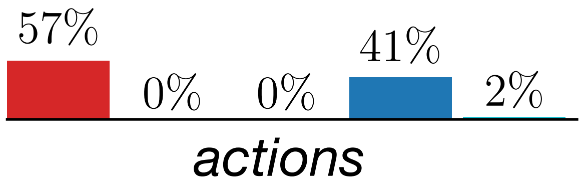
- RL episode is a Markov **decision** process



  - transition probability: $p(S_{t+1}\,|\,S_t, A_t)$

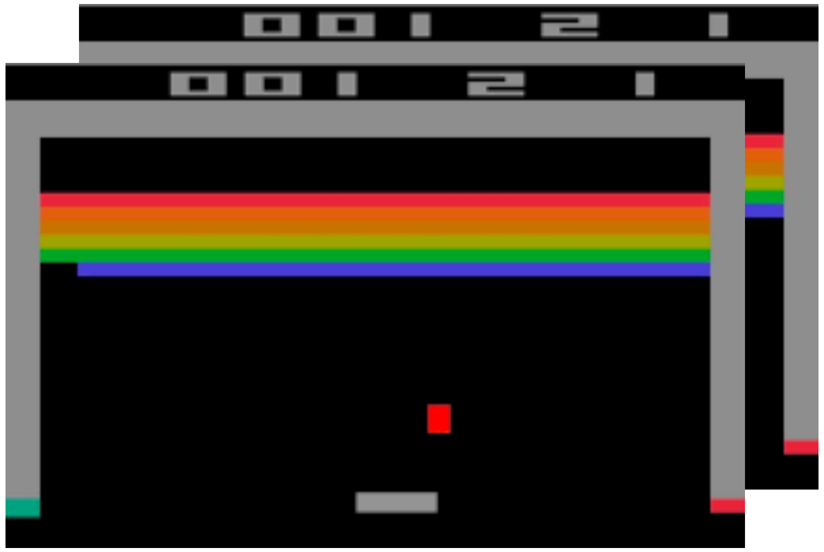


  - policy: $\pi(A_t, S_t)$ — probability to take action A_t in the state S_t



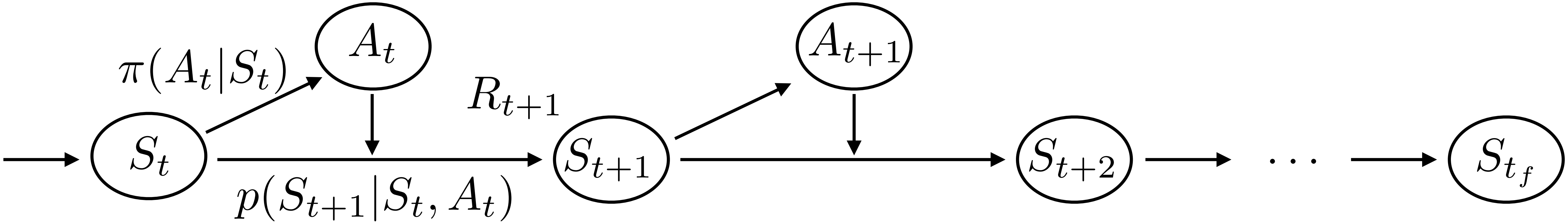
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

RL in a Nutshell



- RL formalism
    - action space: $\mathscr{A} = \{\text{left, stay, right}\}$
    - state space: $\mathscr{S}$ pixelized image of the screen
    - reward function: $r =$ score

- RL episode is a Markov **decision** process



- RL **objective**: *find policy* which maximizes the total *expected return*

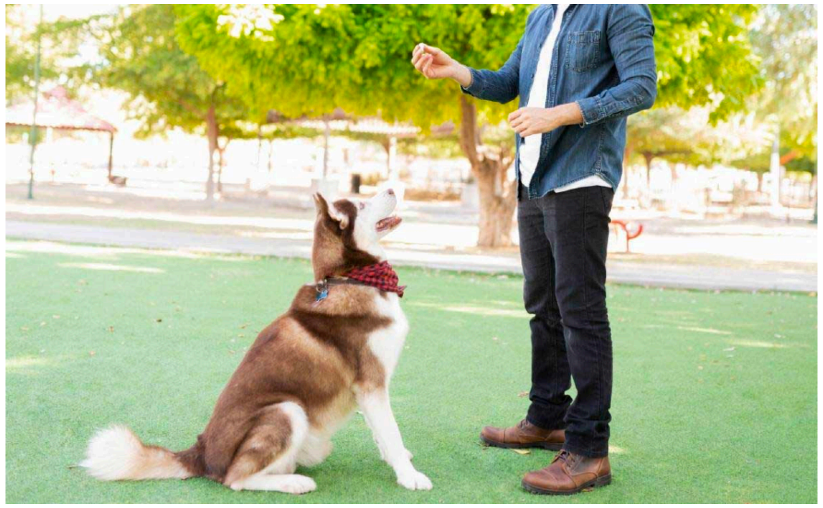
$$J = \mathbb{E}_{a\sim\pi(a|s)}[R_{t+1} + \cdots + R_{t_f}|S_0 = s]$$

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Outline

Part 1

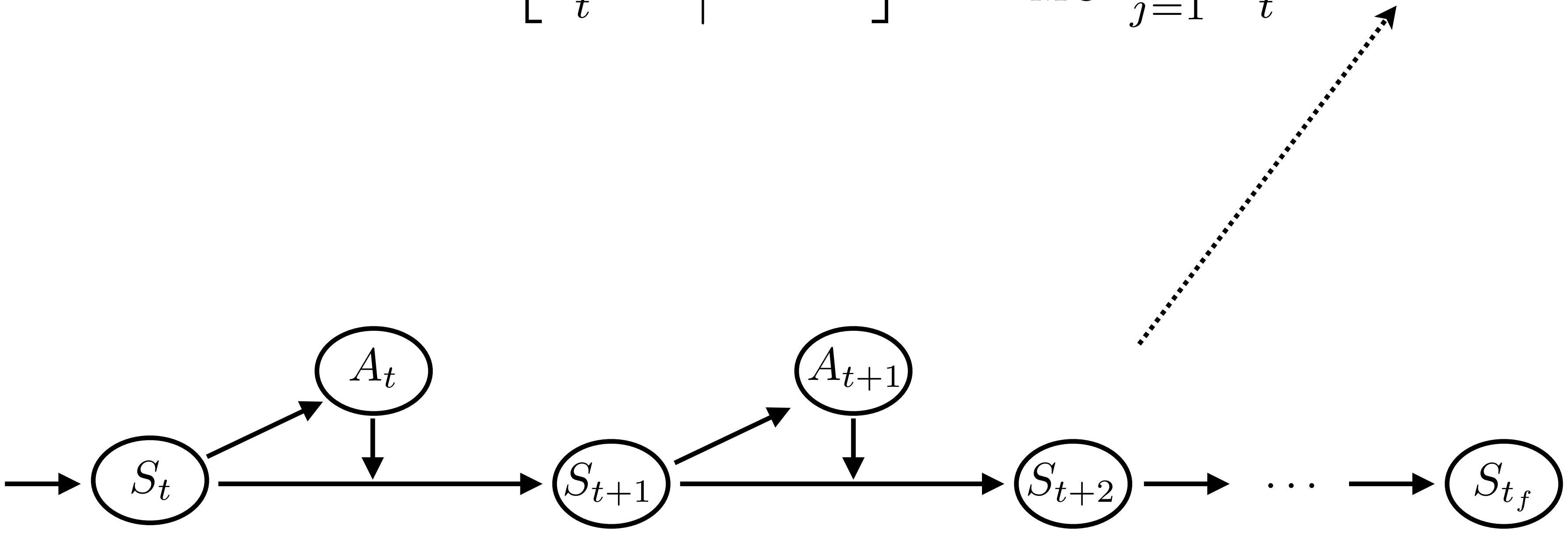
- ✓ Reinforcement learning (RL) in quantum physics
  - RL as a branch of machine learning
- ✓ Applications of RL
  - hallmark applications of RL
  - applications in quantum technologies
- RL framework in a nutshell
  - environment, states, actions, rewards
  - RL algorithms



Policy Gradient

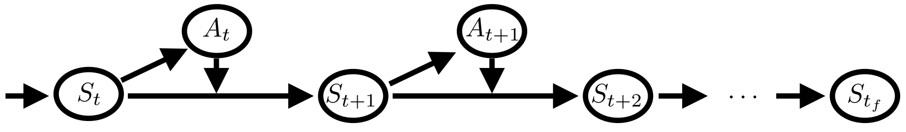
- *in practice:* **evaluate** RL objective by sampling trajectories

$$J = \mathbb{E}_{a\sim\pi(a|s)}\left[\sum_t R_t \middle| S_0 = s\right] \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\sum_t R_t(\tau_j)$$



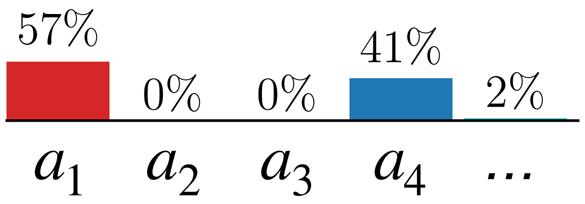
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Policy Gradient



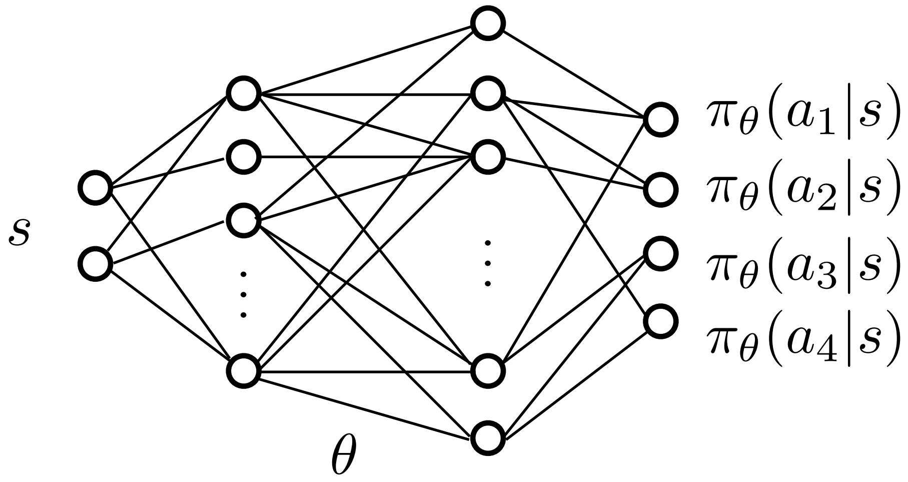
- *in practice:* **evaluate** RL objective by sampling trajectories

$$J = \mathbb{E}_{a\sim\pi(a|s)}\left[\sum_t R_t \middle| S_0 = s\right] \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\sum_t R_t(\tau_j)$$



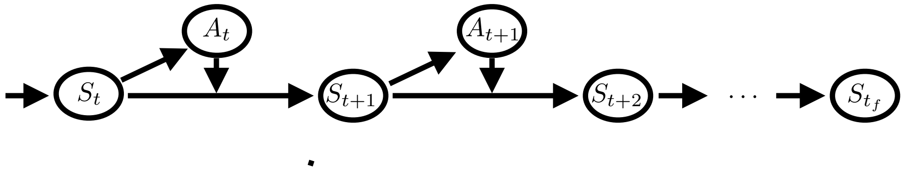
- **improve** policy

⟶ parametrize policy $\pi(a|s) \approx \pi_\theta(a|s)$



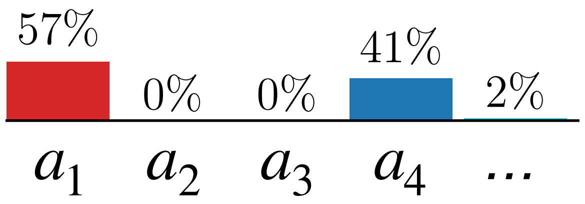
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Policy Gradient

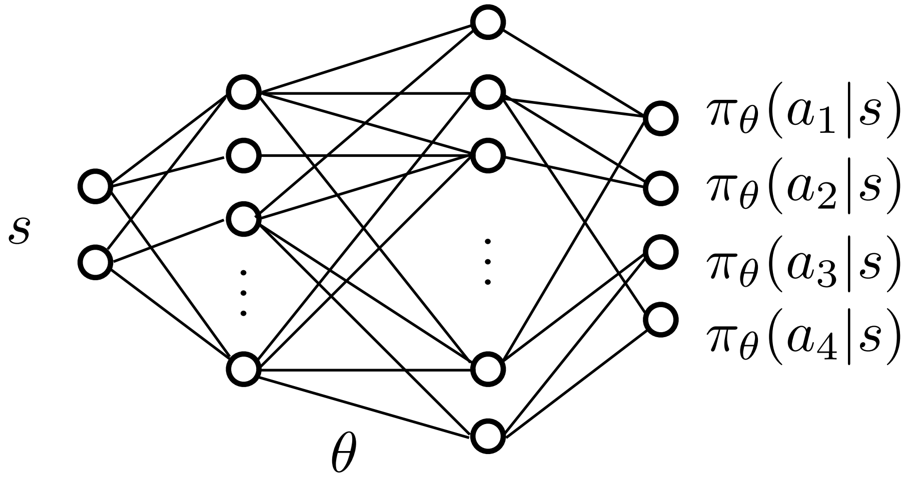


- *in practice:* **evaluate** RL objective by sampling trajectories

$$J = \mathbb{E}_{a\sim\pi(a|s)}\left[\sum_t R_t \middle| S_0 = s\right] \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\sum_t R_t(\tau_j)$$



- **improve** policy



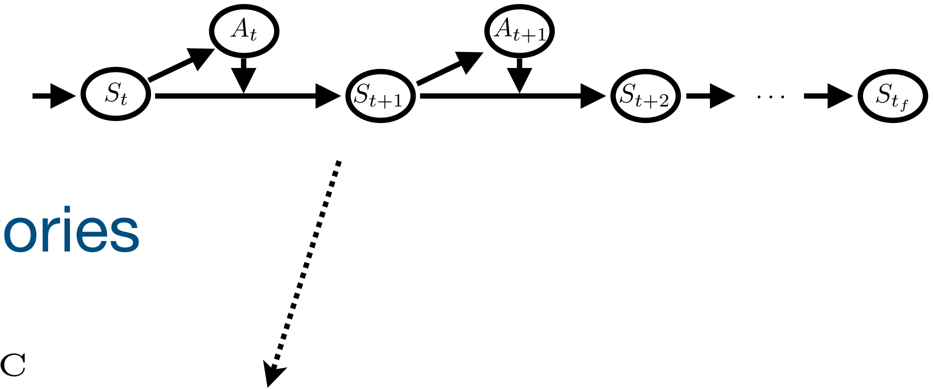
⟶ parametrize policy $\pi(a|s) \approx \pi_\theta(a|s)$

⟶ compute gradients

$$\nabla_\theta J(\theta) = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log\pi_\theta \sum_t R_t \middle| S_0 = s\right] \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}} \nabla_\theta \log\pi_\theta(\tau_j)\sum_t R_t(\tau_j)$$

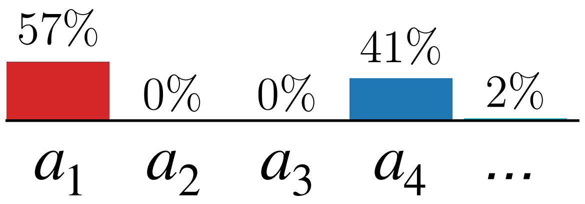
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Policy Gradient

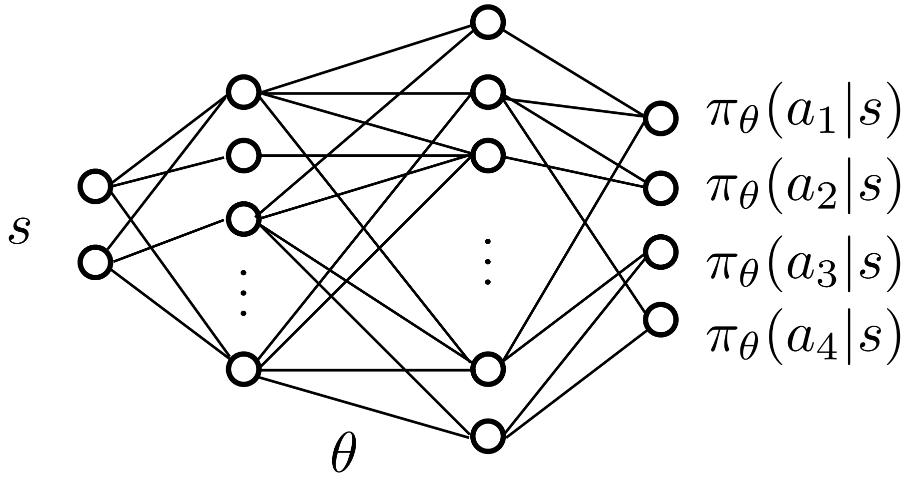


- *in practice:* **evaluate** RL objective by sampling trajectories

$$J = \mathbb{E}_{a\sim\pi(a|s)}\left[\sum_t R_t \middle| S_0 = s\right] \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\sum_t R_t(\tau_j)$$



- **improve** policy



⟶ parametrize policy $\pi(a|s) \approx \pi_\theta(a|s)$

⟶ compute gradients

$$\nabla_\theta J(\theta) = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log\pi_\theta \sum_t R_t \middle| S_0 = s\right] \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}} \nabla_\theta \log\pi_\theta(\tau_j)\sum_t R_t(\tau_j)$$

⟶ update parameters θ to maximize return

$$\theta_{\mathrm{new}} = \theta_{\mathrm{old}} + \alpha\nabla_\theta J(\theta)$$

gradient ascent

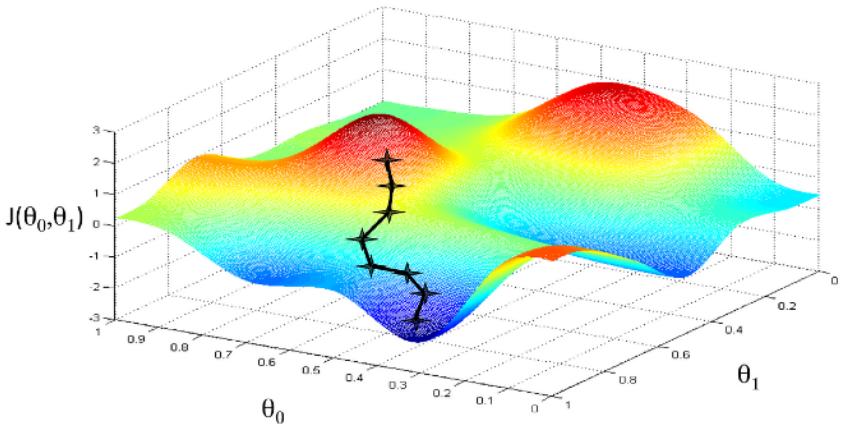
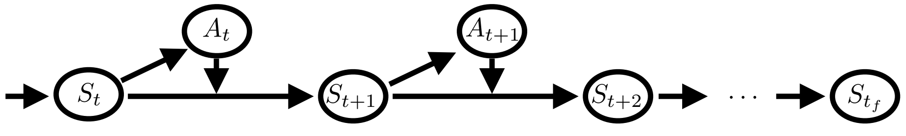


image: medium.com

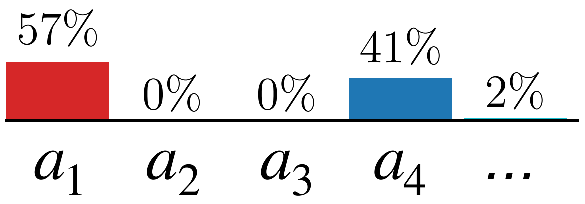
Marín Bukov Sutton and Barto, Reinforcement Learning: an Introduction, MIT press **MPI-PKS (Dresden)**

Policy Gradient

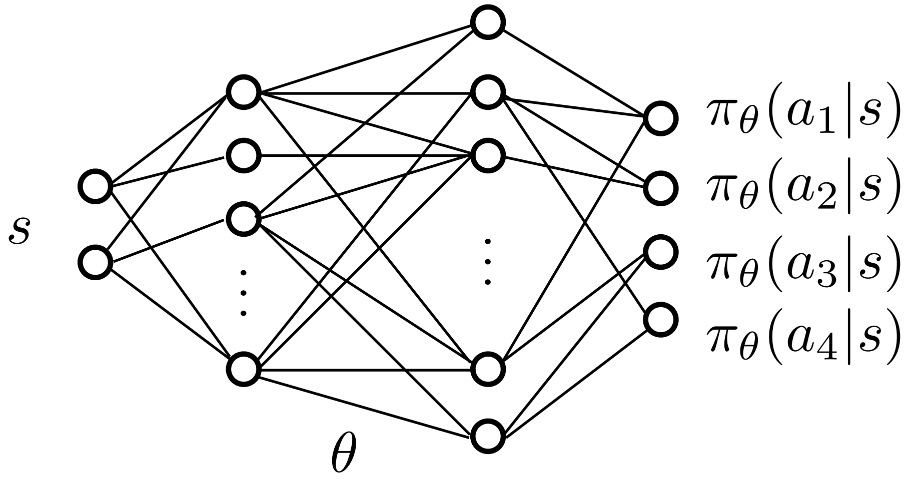


- *in practice:* **evaluate** RL objective by sampling trajectories

$$J = \mathbb{E}_{a\sim\pi(a|s)}\left[\sum_t R_t \middle| S_0 = s\right] \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\sum_t R_t(\tau_j)$$



- **improve** policy



⟶ parametrize policy $\pi(a|s) \approx \pi_\theta(a|s)$

⟶ compute gradients

$$\nabla_\theta J(\theta) = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta \sum_t R_t \middle| S_0 = s\right] \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}} \nabla_\theta \log \pi_\theta(\tau_j)\sum_t R_t(\tau_j)$$

⟶ update parameters θ to maximize return

$$\theta_{\mathrm{new}} = \theta_{\mathrm{old}} + \alpha\nabla_\theta J(\theta)$$

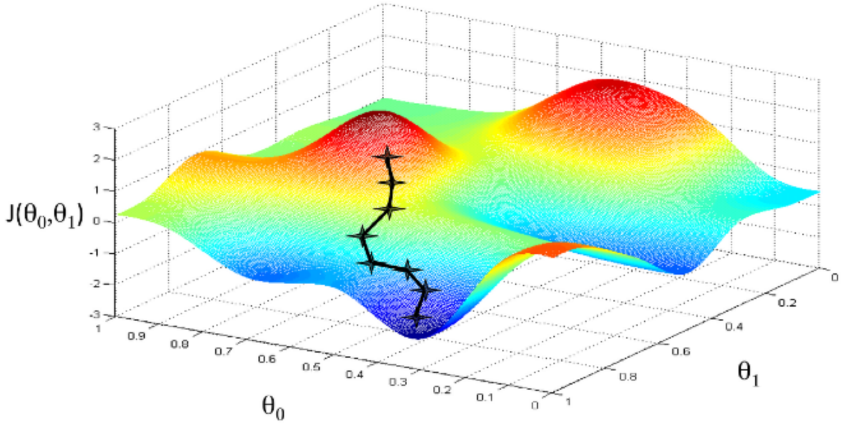


image: medium.com

- pseudo-loss function $\tilde{J}(\theta)$ $\nabla_\theta \tilde{J}(\theta) = \nabla_\theta J_{\mathrm{MC}}(\theta)$

$$\tilde{J}(\theta) = \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}} \log \pi_\theta(\tau_j)\sum_t R_t(\tau_j)$$

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

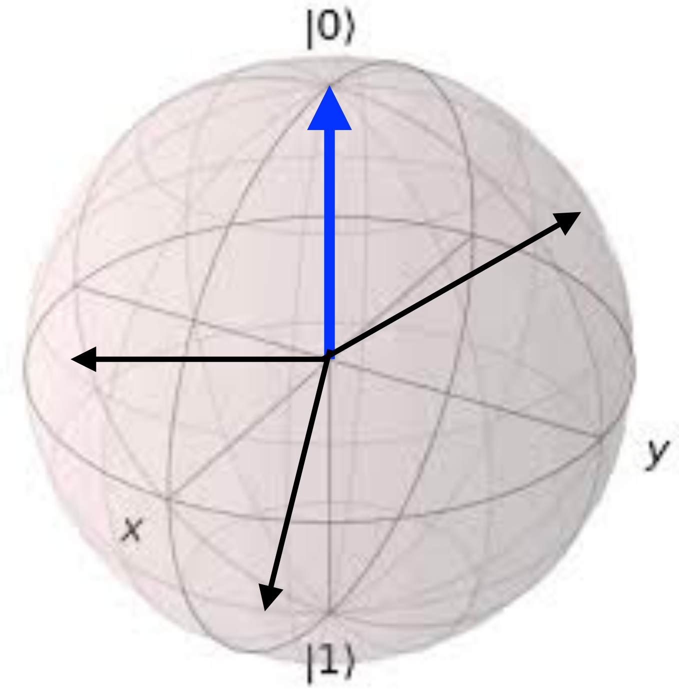
Example: RL for Quantum State Initialization

→ **quantum control:** two-level system

- **task:** prepare $|0\rangle$ using infinitesimal rotations

$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

Bloch sphere



RL for Quantum State Initialization

→ **quantum control:** two-level system

- **task:** prepare $|0\rangle$ using infinitesimal rotations

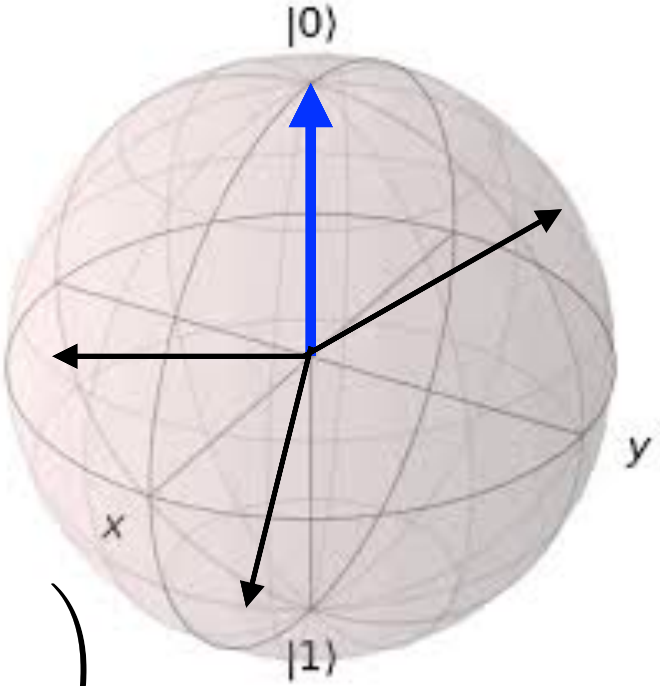
$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

- define **RL states**:

parametrization of state of system: $$|\psi\rangle = \begin{pmatrix} \cos\frac{\vartheta}{2} \\ \mathrm{e}^{i\varphi}\sin\frac{\vartheta}{2} \end{pmatrix}$$

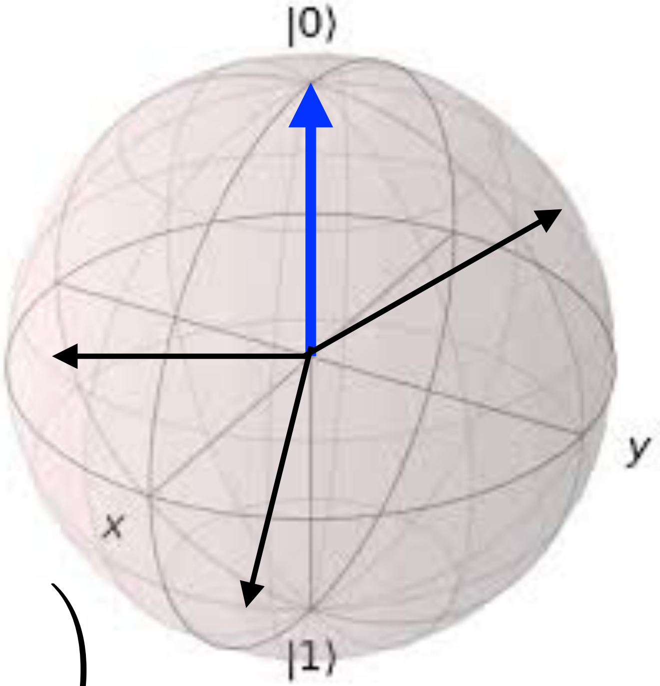
RL state space: $$\mathcal{S} = \{(\vartheta,\varphi)|\vartheta\in[0,\pi],\ \varphi\in[0,2\pi)\}$$

Bloch sphere



RL for Quantum State Initialization

Bloch sphere



→ **quantum control:** two-level system

- **task:** prepare $|0\rangle$ using infinitesimal rotations

$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

- define **RL states**:

parametrization of state of system: $|\psi\rangle = \begin{pmatrix} \cos\frac{\vartheta}{2} \\ \mathrm{e}^{i\varphi}\sin\frac{\vartheta}{2} \end{pmatrix}$

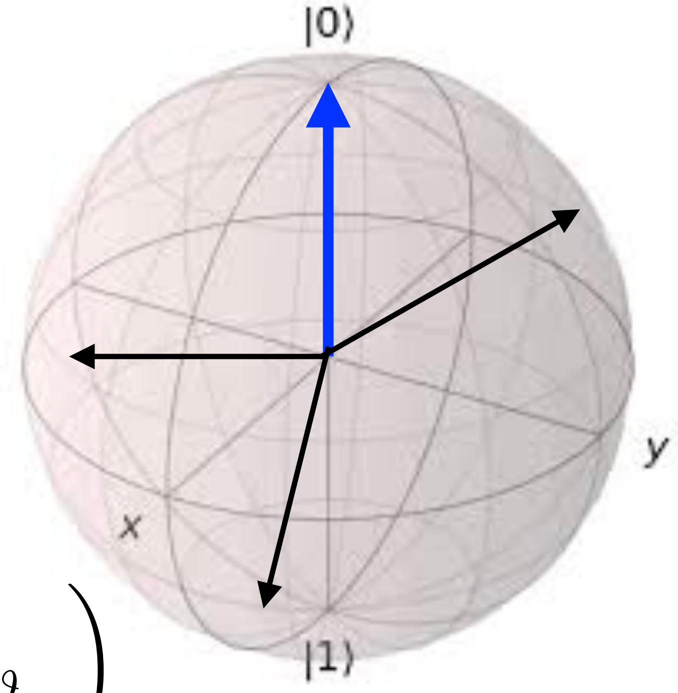
RL state space: $\mathcal{S} = \{(\vartheta,\varphi)|\vartheta\in[0,\pi],\ \varphi\in[0,2\pi)\}$

- define **RL actions**: $\mathcal{A} = \{I, U_\alpha|\alpha = x,y,z\}$ $\qquad |\psi'\rangle = U_\alpha|\psi\rangle \qquad s\mapsto s'$

RL for Quantum State Initialization

→ **quantum control:** two-level system

Bloch sphere



- **task:** prepare $|0\rangle$ using infinitesimal rotations

$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

- define **RL states**:

  parametrization of state of system: $|\psi\rangle = \begin{pmatrix} \cos\frac{\vartheta}{2} \\ \mathrm{e}^{i\varphi}\sin\frac{\vartheta}{2} \end{pmatrix}$

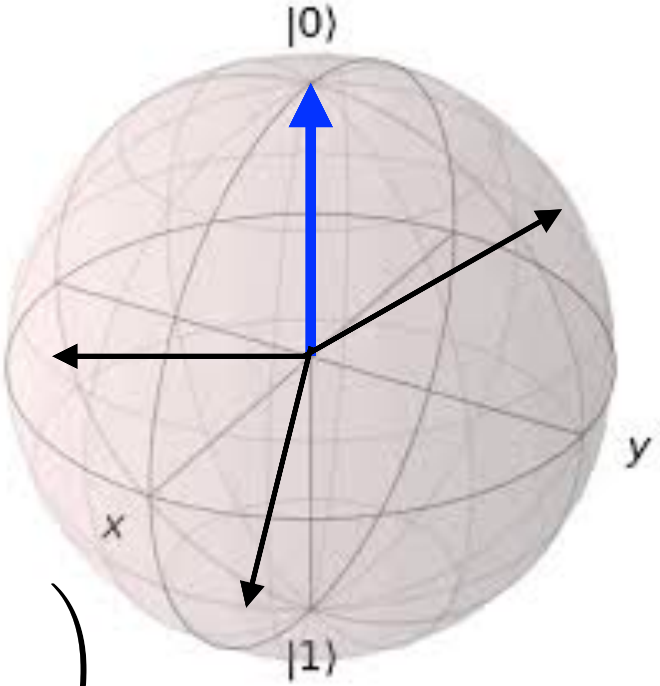
  RL state space: $\mathcal{S} = \{(\vartheta,\varphi)|\vartheta\in[0,\pi],\ \varphi\in[0,2\pi)\}$

- define **RL actions**: $\mathcal{A} = \{I, U_\alpha|\alpha = x, y, z\}$ $\qquad |\psi'\rangle = U_\alpha|\psi\rangle \qquad s \mapsto s'$

- define reward: $r_t = |\langle 0|\psi_t\rangle|^2$

RL for Quantum State Initialization

Bloch sphere



→ **quantum control:** two-level system

- **task:** prepare $|0\rangle$ using infinitesimal rotations

$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

- define **RL states**:

  parametrization of state of system: $|\psi\rangle = \begin{pmatrix} \cos\frac{\vartheta}{2} \\ \mathrm{e}^{i\varphi}\sin\frac{\vartheta}{2} \end{pmatrix}$

  RL state space: $\mathcal{S} = \{(\vartheta,\varphi)|\vartheta\in[0,\pi],\ \varphi\in[0,2\pi)\}$

- define **RL actions**: $\mathcal{A} = \{I, U_\alpha|\alpha = x,y,z\}$ $\quad|\psi'\rangle = U_\alpha|\psi\rangle\quad$ $s\mapsto s'$

- define reward: $r_t = |\langle 0|\psi_t\rangle|^2$

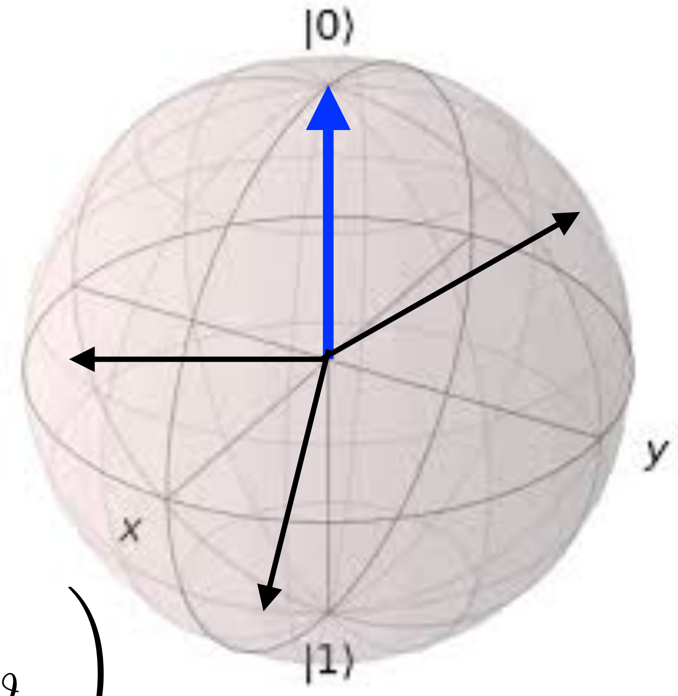
- choose **RL algorithm:** policy gradient

  simulate trial trajectories τ_j

$$\nabla_\theta J(\theta) \approx \frac{1}{N_\mathrm{MC}}\sum_{j=1}^{N_\mathrm{MC}}\nabla_\theta\log\pi_\theta(\tau_j)\sum_t R_t(\tau_j)$$

RL for Quantum State Initialization

Bloch sphere



→ **quantum control:** two-level system

- **task:** prepare $|0\rangle$ using infinitesimal rotations

$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

- define **RL states**:

parametrization of state of system: $|\psi\rangle = \begin{pmatrix} \cos\frac{\vartheta}{2} \\ \mathrm{e}^{i\varphi}\sin\frac{\vartheta}{2} \end{pmatrix}$

RL state space: $\mathcal{S} = \{(\vartheta,\varphi)|\vartheta\in[0,\pi],\ \varphi\in[0,2\pi)\}$

- define **RL actions**: $\mathcal{A} = \{I, U_\alpha|\alpha = x,y,z\}$ $\qquad |\psi'\rangle = U_\alpha|\psi\rangle \qquad s\mapsto s'$

- define reward: $r_t = |\langle 0|\psi_t\rangle|^2$

problem: *continuous* state space has infinitely many configurations

RL with Function Approximation

→ **problem:** state space has exponentially/continuously many configurations $|\mathcal{A}|^{N_T}$

- can we estimate values of not yet encountered states?

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

RL with Function Approximation

→ **problem:** state space has exponentially/continuously many configurations $|\mathcal{A}|^{N_T}$

- can we estimate values of not yet encountered states?

→ YES, via inter- & extrapolation: parametrize the Q-function/policy

$$\pi(a|s) \to \pi_\theta(a|s)$$

variational parameters θ

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

RL with Function Approximation

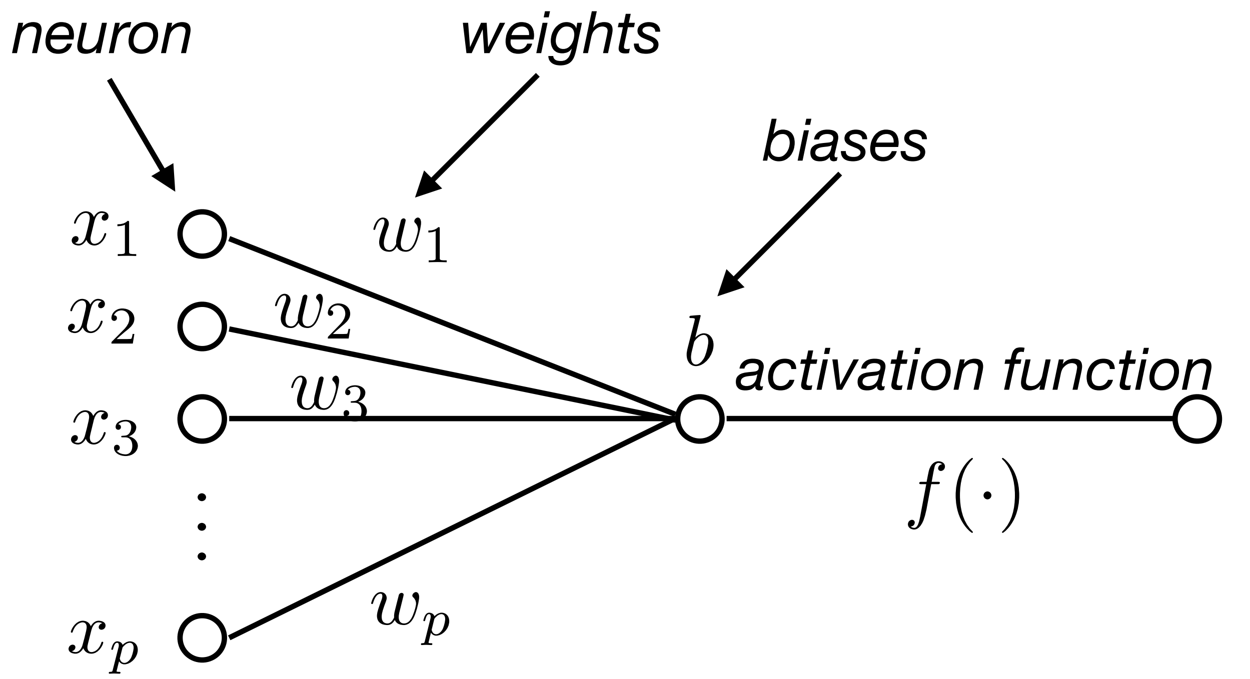
→ **problem:** state space has exponentially/continuously many configurations $|\mathcal{A}|^{N_T}$

- can we estimate values of not yet encountered states?

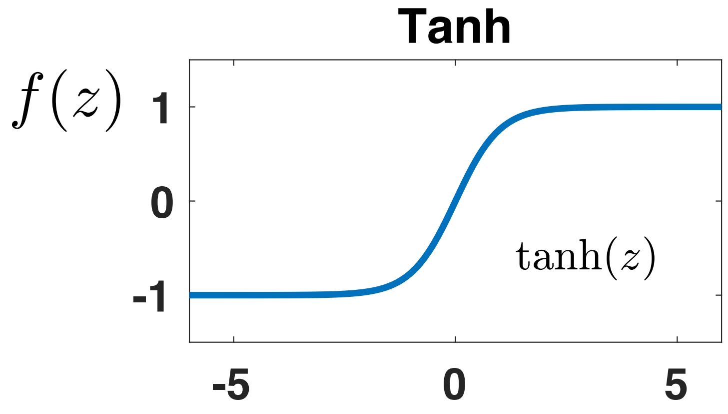
→ YES, via inter- & extrapolation: parametrize the Q-function/policy

$$\pi(a|s) \to \pi_\theta(a|s)$$

- typical approach: use deep neural network **(Deep RL)**



$$y = f(W \cdot x + b)$$



$$W = (w_1, w_2, \dots, w_p) \qquad \theta = \{W, b\}$$

RL with Function Approximation

→ **problem:** state space has exponentially/continuously many configurations $|\mathcal{A}|^{N_T}$

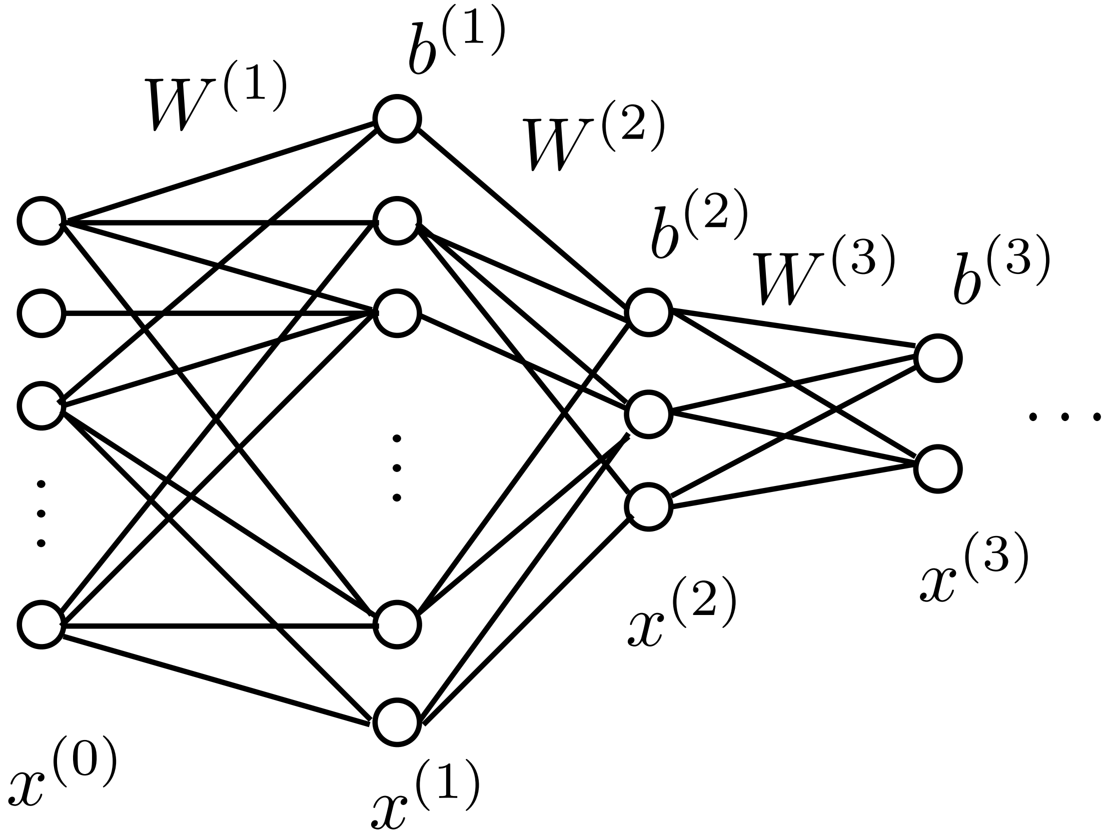
- can we estimate values of not yet encountered states?

→ YES, via inter- & extrapolation: parametrize the Q-function/policy

$$\pi(a|s) \to \pi_\theta(a|s)$$

$$\theta = \{W, b\}$$

- typical approach: use deep neural network **(Deep RL)**



$x^{(0)}$ input layer

$$x_i^{(1)} = f^{(1)}\left(W_{ij}^{(1)} x_j^{(0)} + b_i^{(1)}\right)$$

$$x_i^{(2)} = f^{(2)}\left(W_{ij}^{(2)} x_j^{(1)} + b_i^{(2)}\right)$$

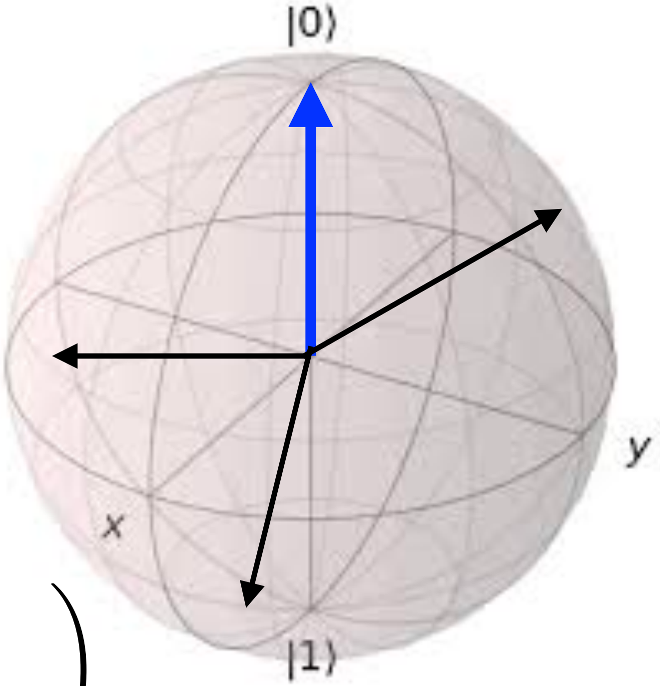
$b_i^{(l)}$: bias vector of layer l

$W_{ij}^{(l)}$: weight matrix of layer l

$f^{(l)}$: activation function of layer l

RL for Quantum State Initialization

Bloch sphere



→ **quantum control:** two-level system

- **task:** prepare $|0\rangle$ using infinitesimal rotations

$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

- define **RL states**:

parametrization of state of system: $|\psi\rangle = \begin{pmatrix} \cos\frac{\vartheta}{2} \\ \mathrm{e}^{i\varphi}\sin\frac{\vartheta}{2} \end{pmatrix}$

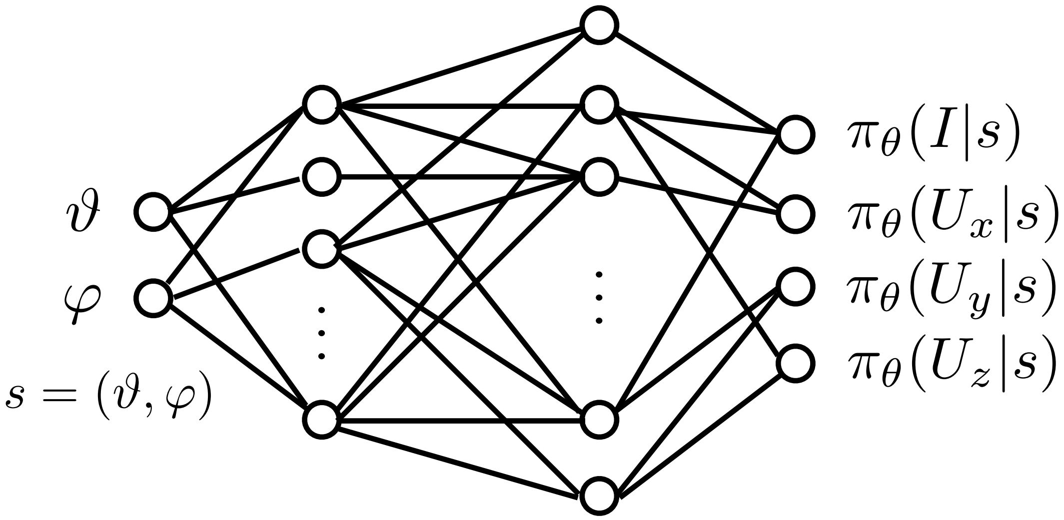
RL state space: $\mathcal{S} = \{(\vartheta, \varphi) | \vartheta \in [0, \pi],\ \varphi \in [0, 2\pi)\}$

- define **RL actions**: $\mathcal{A} = \{I, U_\alpha | \alpha = x, y, z\}$

- define reward: $r_t = |\langle 0|\psi_t\rangle|^2$

- choose **RL algorithm:** policy gradient

$$\pi_\theta(a|s)$$



RL for Quantum State Initialization

→ **quantum control:** two-level system

- **task:** prepare $|0\rangle$ using infinitesimal rotations

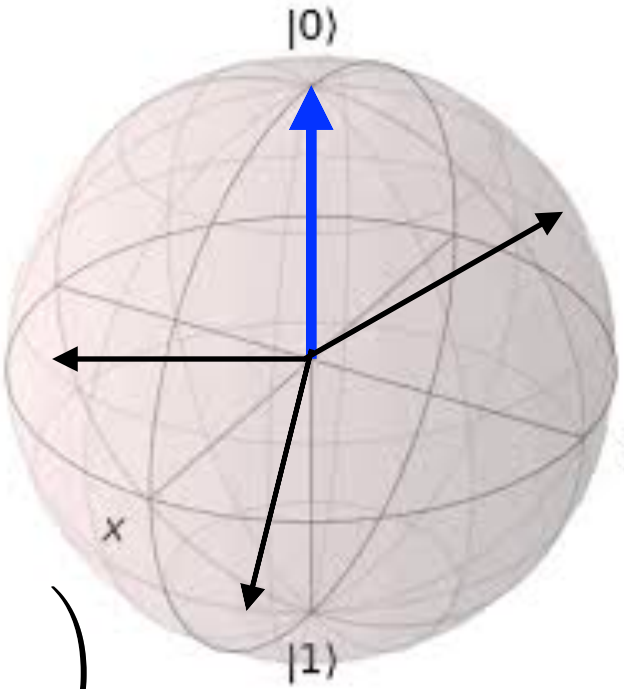
$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

- define **RL states**:

parametrization of state of system: $|\psi\rangle = \begin{pmatrix} \cos\frac{\vartheta}{2} \\ \mathrm{e}^{i\varphi}\sin\frac{\vartheta}{2} \end{pmatrix}$

RL state space: $\mathcal{S} = \{(\vartheta,\varphi)|\vartheta \in [0,\pi],\ \varphi \in [0,2\pi)\}$

Bloch sphere



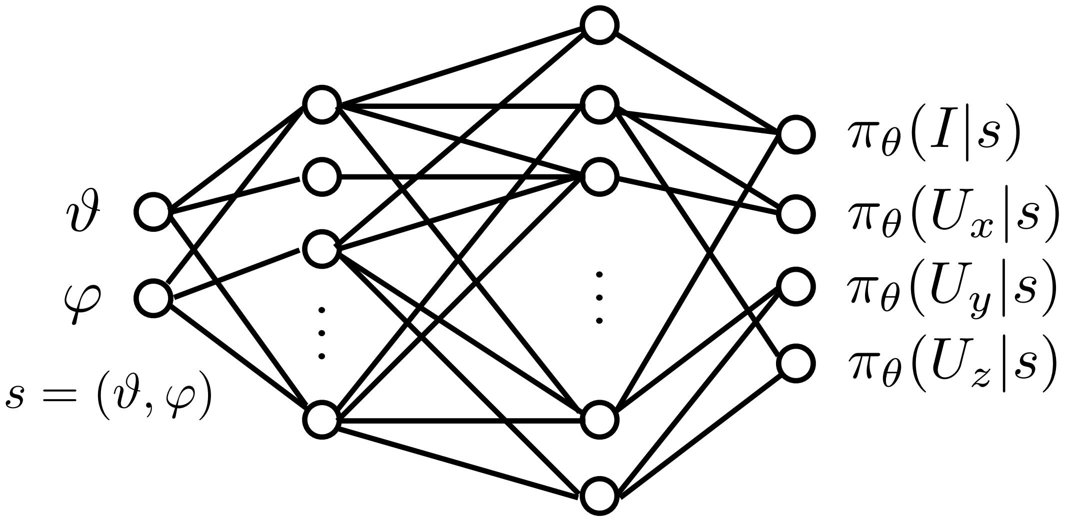
- define **RL actions**: $\mathcal{A} = \{I, U_\alpha|\alpha = x,y,z\}$

- define reward: $r_t = |\langle 0|\psi_t\rangle|^2$

- choose **RL algorithm:** policy gradient

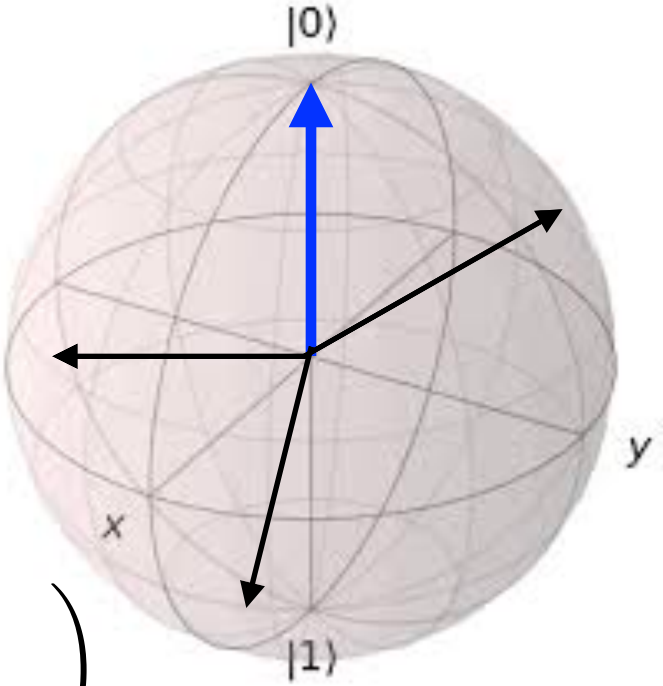
simulate trial trajectories τ_j

$$\nabla_\theta J(\theta) \approx \frac{1}{N_\mathrm{MC}}\sum_{j=1}^{N_\mathrm{MC}} \nabla_\theta \log\pi_\theta(\tau_j)\sum_t R_t(\tau_j)$$



RL for Quantum State Initialization

Bloch sphere



→ **quantum control:** two-level system

- **task:** prepare $|0\rangle$ using infinitesimal rotations

$$U_\alpha = \exp\left(-i\frac{\delta t}{2}\sigma^\alpha\right) \qquad \alpha = x, y, z$$

- define **RL states**:

  parametrization of state of system: $|\psi\rangle = \begin{pmatrix} \cos\frac{\vartheta}{2} \\ \mathrm{e}^{i\varphi}\sin\frac{\vartheta}{2} \end{pmatrix}$

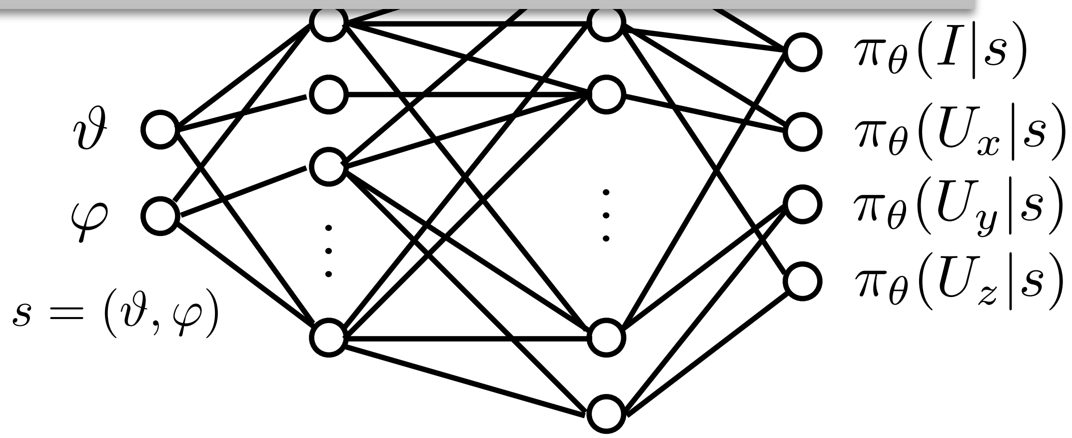
Check out Jupyter notebook for how this works in practice!

https://github.com/mgbukov/RL_quantum

- choose **RL algorithm:** policy gradient

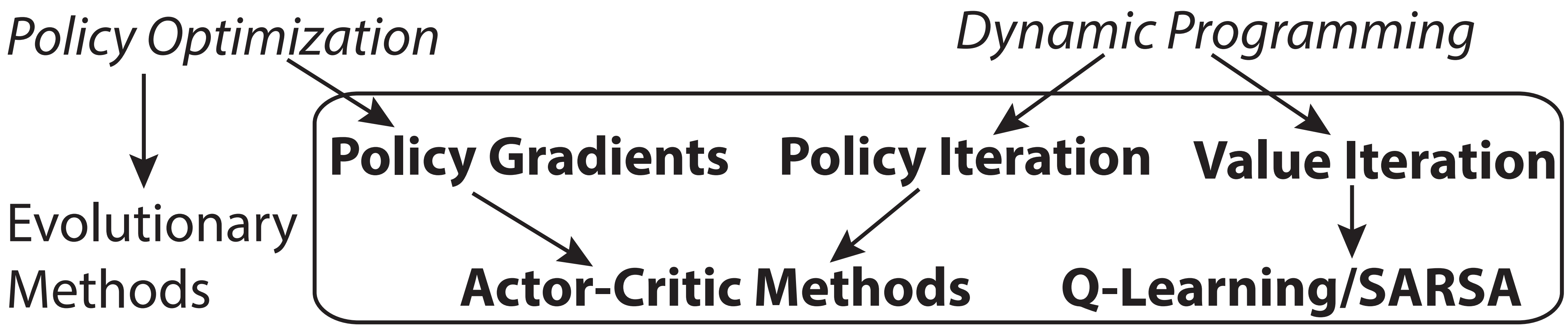
  simulate trial trajectories τ_j

$$\nabla_\theta J(\theta) \approx \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}} \nabla_\theta \log \pi_\theta(\tau_j) \sum_t R_t(\tau_j)$$



What other RL Algorithms are there?

→ overview of RL algorithms



Policy Optimization

Evolutionary Methods

Dynamic Programming

Policy Gradients

Policy Iteration

Value Iteration

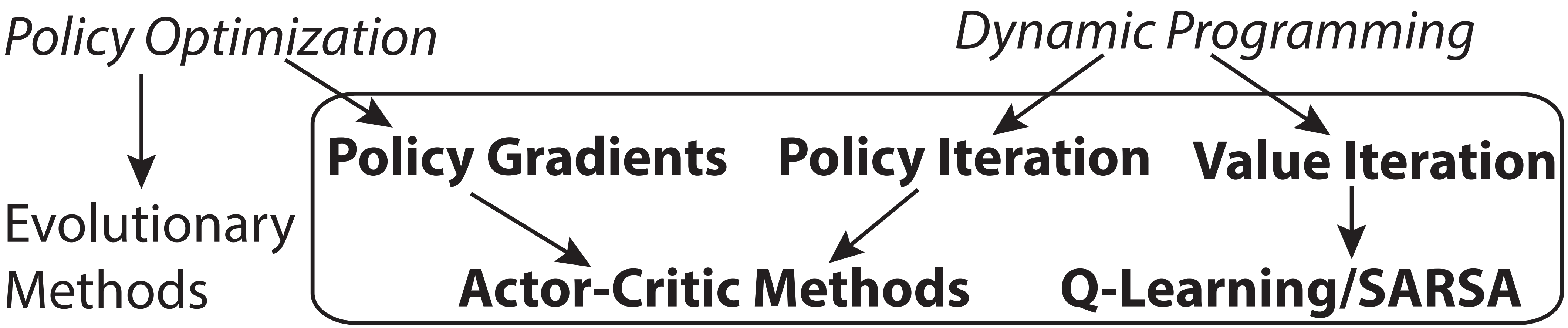
Actor-Critic Methods

Q-Learning/SARSA

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

What other RL Algorithms are there?

→ overview of RL algorithms



Policy Optimization

Dynamic Programming

Evolutionary Methods

Policy Gradients

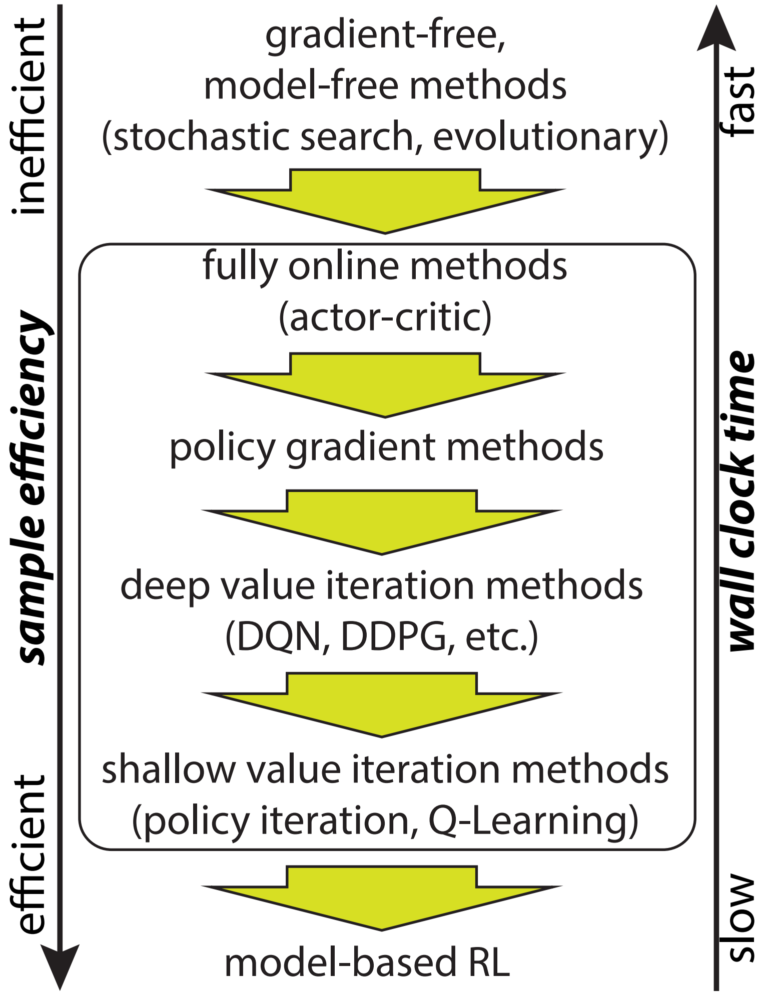
Policy Iteration

Value Iteration

Actor-Critic Methods

Q-Learning/SARSA

→ which algorithm to use?



gradient-free, model-free methods (stochastic search, evolutionary)

fully online methods (actor-critic)

policy gradient methods

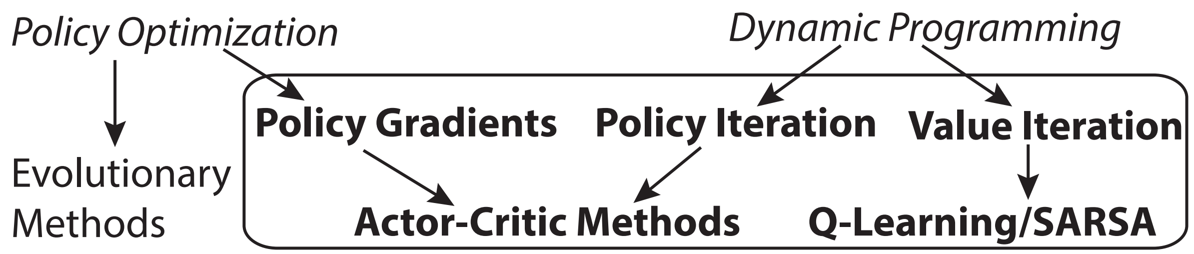
deep value iteration methods (DQN, DDPG, etc.)

shallow value iteration methods (policy iteration, Q-Learning)

model-based RL

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Value function methods



Policy Optimization

Dynamic Programming

Evolutionary Methods

Policy Gradients

Policy Iteration

Value Iteration

Actor-Critic Methods

Q-Learning/SARSA

→ Value Iteration methods

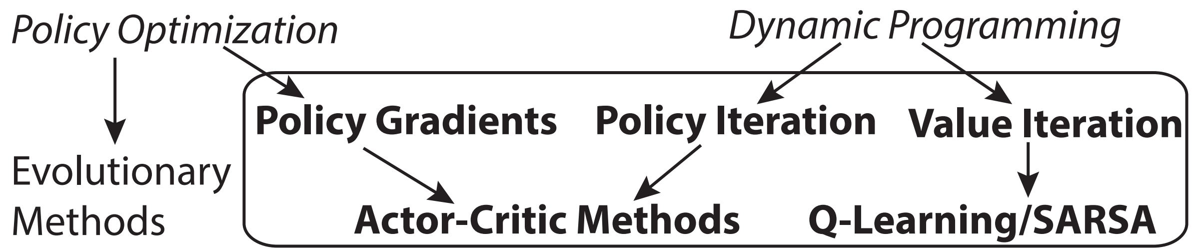
- value function: ***expected*** total return under the policy $\pi(a|s)$ from state s

$$v_\pi(s) = \mathbb{E}_{a\sim\pi(a|s)}[G_t|S_t = s] \qquad\qquad G_t = R_{t+1} + G_{t+1}$$

problem: cannot reconstruct the policy

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Value function methods


Policy Optimization

Evolutionary Methods

Dynamic Programming

Policy Gradients

Policy Iteration

Value Iteration

Actor-Critic Methods

Q-Learning/SARSA

→ Value Iteration methods

- value function: ***expected*** total return under the policy $\pi(a|s)$ from state s

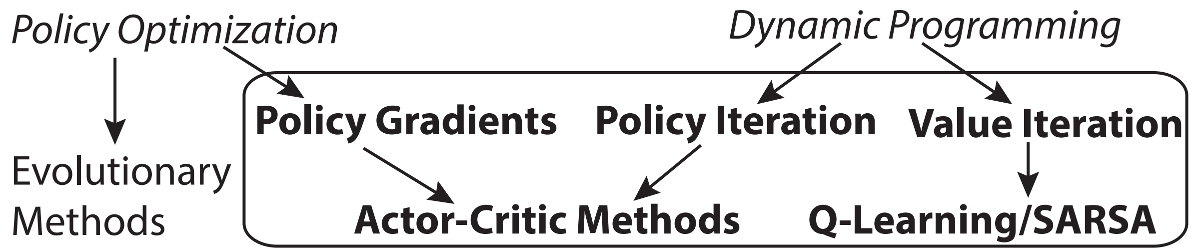
$$v_\pi(s) = \mathbb{E}_{a\sim\pi(a|s)}[G_t|S_t = s] \qquad\qquad G_t = R_{t+1} + G_{t+1}$$

- action-value (or Q-) function: ***expected*** total return under the policy $\pi(a|s)$ starting from state s and taking action a:

$$Q_\pi(s,a) = \mathbb{E}_{a\sim\pi(a|s)}[G_t|S_t = s, A_t = a]$$

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Value function methods


Policy Optimization

Dynamic Programming

Evolutionary Methods

Policy Gradients

Policy Iteration

Value Iteration

Actor-Critic Methods

Q-Learning/SARSA

→ Value Iteration methods

- value function: ***expected*** total return under the policy $\pi(a|s)$ from state s

$$v_\pi(s) = \mathbb{E}_{a\sim\pi(a|s)}[G_t|S_t = s] \qquad G_t = R_{t+1} + G_{t+1}$$

- action-value (or Q-) function: ***expected*** total return under the policy $\pi(a|s)$ starting from state s and taking action a:

$$Q_\pi(s,a) = \mathbb{E}_{a\sim\pi(a|s)}[G_t|S_t = s, A_t = a]$$

→ optimal action-value function: $Q_*(s,a) = \max_\pi Q_\pi(s,a)$

$$\pi_*(a|s) = \mathrm{argmax}_a Q_*(s,a)$$

Bellman's equation: $Q_*(s,a) = \sum_{s'} p(s'|s,a)\left[r(s,s',a) + \max_{a'} Q_*(s',a')\right]$

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Meaning of Q-function

→ assign a value to each state

- first step deterministic, then follow policy

$$Q(s,a) = \mathbb{E}_{a\sim\pi}[R_{t+1} + \cdots + R_{t_f} | S_0 = s, A_0 = a]$$

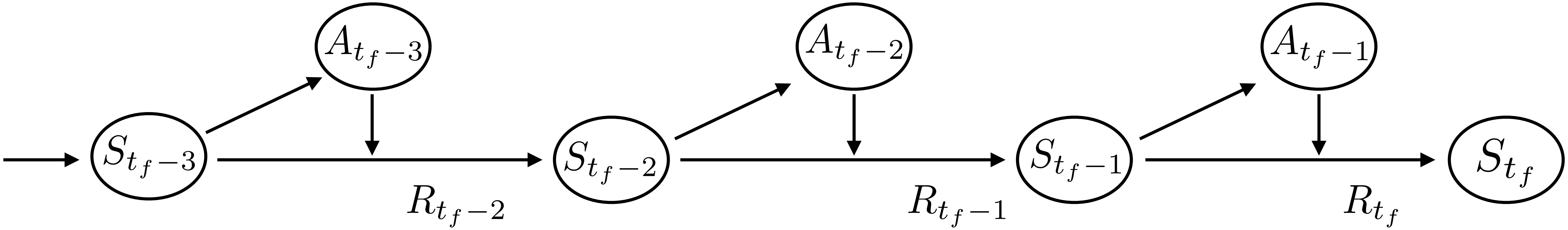
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Meaning of Q-function

→ assign a value to each state

- first step deterministic, then follow policy

$$Q(s,a) = \mathbb{E}_{a\sim\pi}[R_{t+1} + \cdots + R_{t_f} | S_0 = s, A_0 = a]$$



Marín Bukov

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

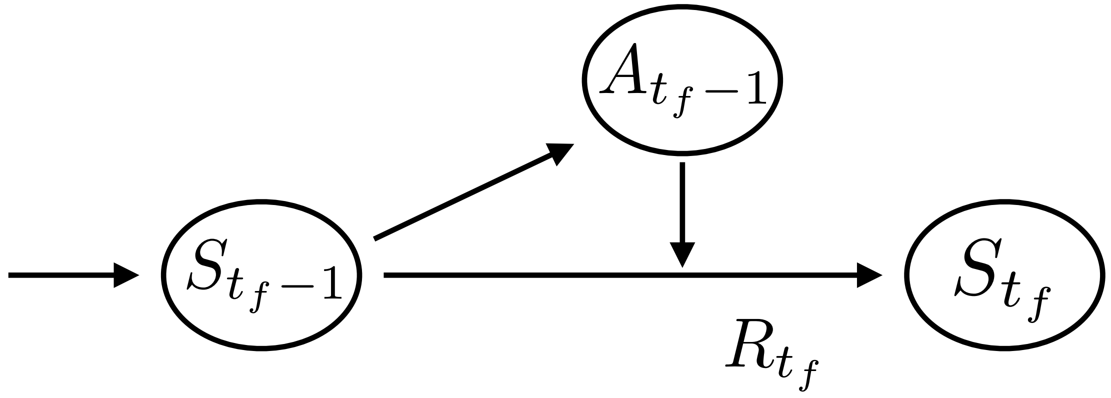
MPI-PKS

Meaning of Q-function

→ assign a value to each state

- first step deterministic, then follow policy

$$Q(s,a) = \mathbb{E}_{a\sim\pi}[R_{t+1} + \cdots + R_{t_f} | S_0 = s, A_0 = a]$$



$$Q(S_{t_f-1}, A_{t_f-1}) = R_{t_f}$$

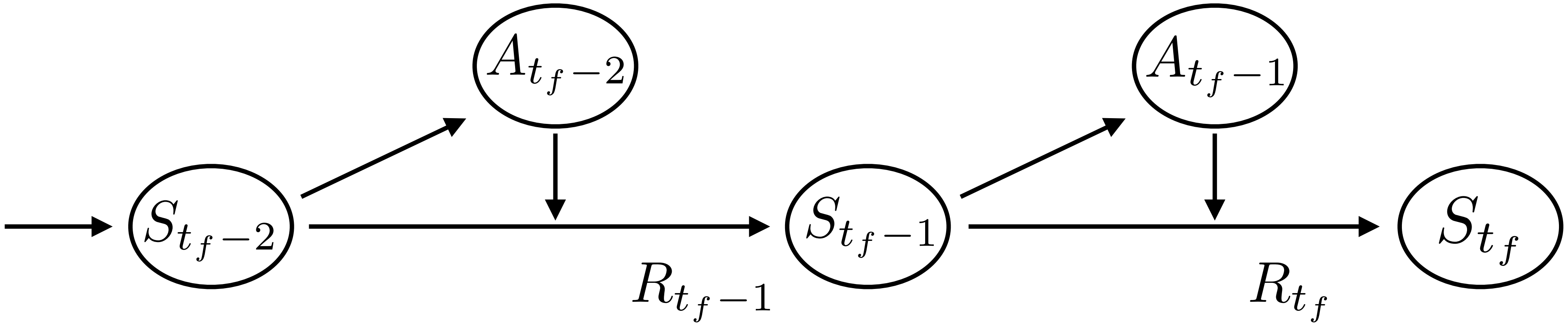
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Meaning of Q-function

→ assign a value to each state

- first step deterministic, then follow policy

$$Q(s,a) = \mathbb{E}_{a\sim\pi}[R_{t+1} + \cdots + R_{t_f} | S_0 = s, A_0 = a]$$



$$Q(S_{t_f-1}, A_{t_f-1}) = R_{t_f}$$

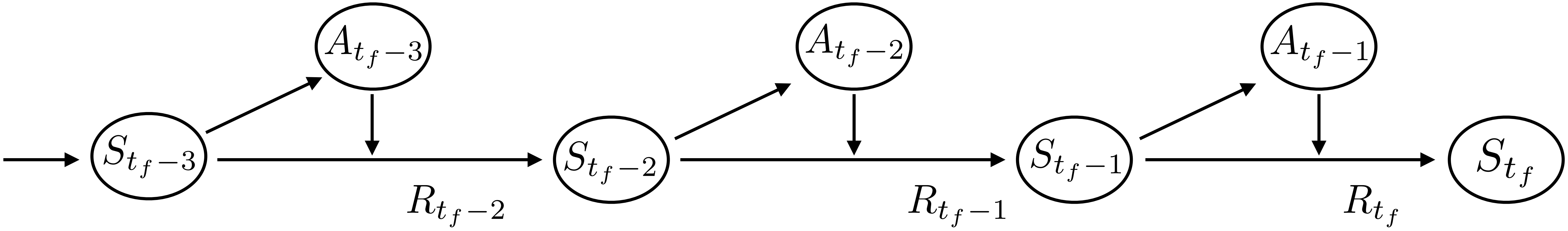
$$Q(S_{t_f-2}, A_{t_f-2}) = R_{t_f-1} + R_{t_f}$$

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Meaning of Q-function

- assign a value to each state
    - first step deterministic, then follow policy

$$Q(s,a) = \mathbb{E}_{a\sim\pi}[R_{t+1} + \cdots + R_{t_f} | S_0 = s, A_0 = a]$$



$$Q(S_{t_f-1}, A_{t_f-1}) = R_{t_f}$$

$$Q(S_{t_f-2}, A_{t_f-2}) = R_{t_f-1} + R_{t_f}$$

$$Q(S_{t_f-3}, A_{t_f-3}) = R_{t_f-2} + R_{t_f-1} + R_{t_f}$$

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Q-Learning

→ assign a value to each state $Q(s,a) = \mathbb{E}_{a\sim\pi}[R_{t+1} + \cdots + R_{t_f} | S_0 = s, A_0 = a]$

- first step deterministic, then follow policy

$Q(s,a)$	A_1	A_2	A_3
S_1	1.5	0.5	2.1
S_2	1.3	4.4	0.2
S_3	0.9	0.9	3.3

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Q-Learning

→ assign a value to each state $Q(s,a) = \mathbb{E}_{a\sim\pi}[R_{t+1} + \cdots + R_{t_f} | S_0 = s, A_0 = a]$

- first step deterministic, then follow policy

$Q(s,a)$	A_1	A_2	A_3
S_1	1.5	0.5	2.1
S_2	1.3	4.4	0.2
S_3	0.9	0.9	3.3

$$A = \mathrm{argmax}_a Q(S,a)$$

$$\pi(S_1) = A_3$$

$$\pi(S_2) = A_2$$

$$\pi(S_3) = A_3$$

- take action which maximizes the Q-value at each step

Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

Q-Learning

→ assign a value to each state $Q(s,a) = \mathbb{E}_{a\sim\pi}[R_{t+1} + \cdots + R_{t_f} | S_0 = s, A_0 = a]$

- first step deterministic, then follow policy

$Q(s,a)$	A_1	A_2	A_3
S_1	1.5	0.5	2.1
S_2	1.3	4.4	0.2
S_3	0.9	0.9	3.3

$$A = \mathrm{argmax}_a Q(S,a)$$

$$\pi(S_1) = A_3$$

$$\pi(S_2) = A_2$$

$$\pi(S_3) = A_3$$

- take action which maximizes the Q-value at each step

→ iterate following two steps until convergence

- error according to definition $\delta_t = Q(S_t, A_t) + R_{t+1} - \max_a Q(S_{t+1}, a)$
- update current function $Q_{\text{new}} = Q_{\text{old}} + \alpha\delta_t$

$$\alpha \in [0,1)$$

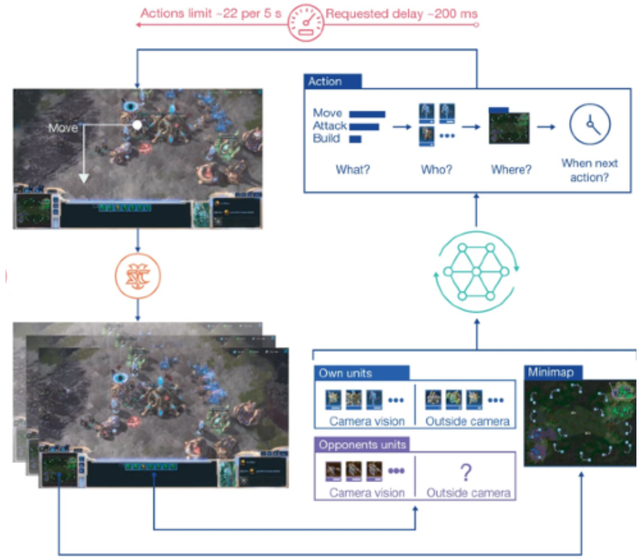
Sutton and Barto, Reinforcement Learning: an Introduction, MIT press

What is reinforcement learning used for?

Mastering the game of Go with deep neural networks and tree search

Silver, et. al, Nature 529 484–489 (2016)

Mastering video games (StarCraft II)

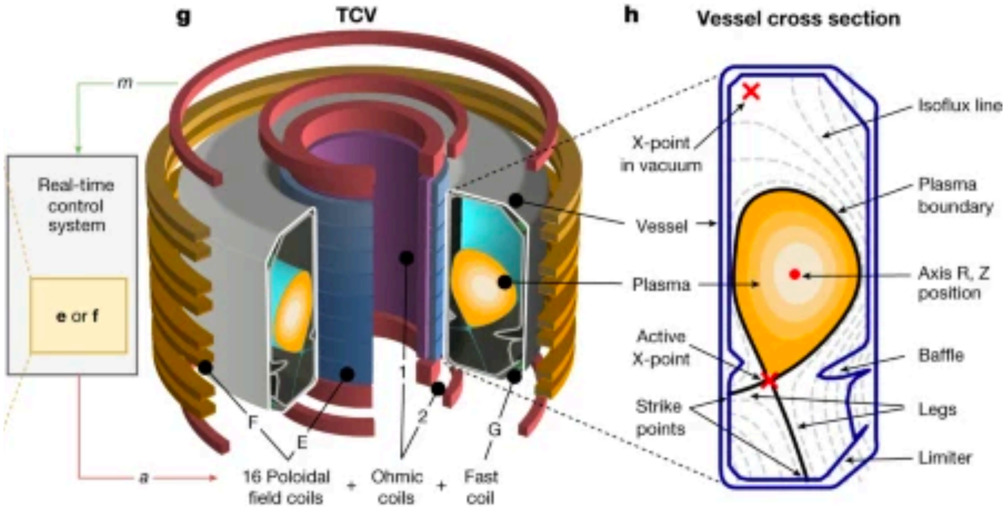


Vinyals, et. al, Nature 350 (2019)

Atari games

Mnih et al., Nature 518 (2015) [Google DeepMind]

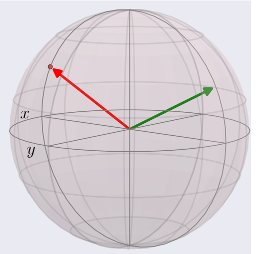
Magnetic control of tokamak plasmas thru deep RL



Degrave, et. al, Nature 602 414–419 (2022)

Applications of RL in Quantum Physics

- quantum control

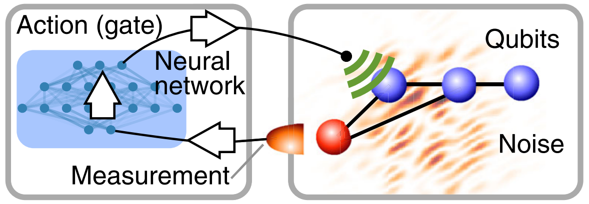


MB et al, PRX 8 031086 (2018)
Niu et al, npj 5 33 (2019)
Sivak et al, PRX 12, 011059 (2022)
Gispen et al, MSML (2021)

Reuer, Nat Comm 14 7138 (2023)
Yao et al, PRX 11 (3), 031070 (2021)
Porotti, Comm Phys 2 (2019)
Dalgaard et al, npj 6 6 (2020)

+ many more

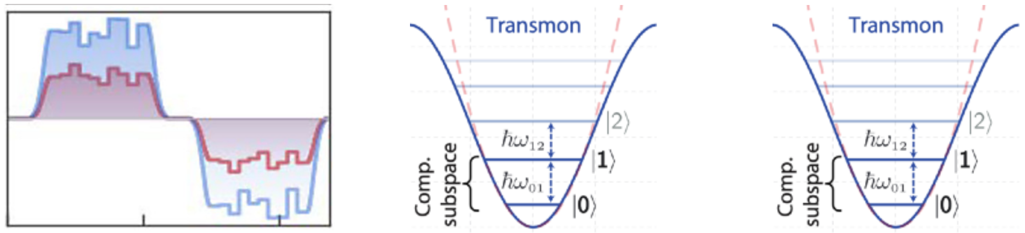
- quantum error correction



Fössel et al, PRX 8 031086 (2018)
Andreasson et al, Quantum 3 183 (2019)
Sweke et al, ML Sci Tech 2 025005 (2020)
Sivak et al, Nature 616 50-55 (2023)
Olle et al, arXiv:2311.04750

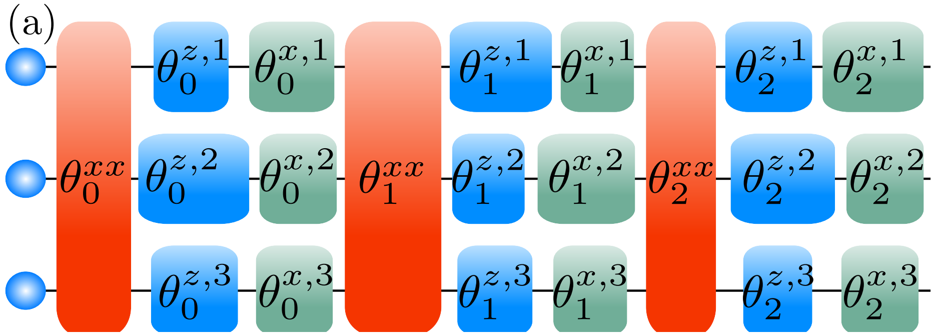
+ many more

- quantum gate design

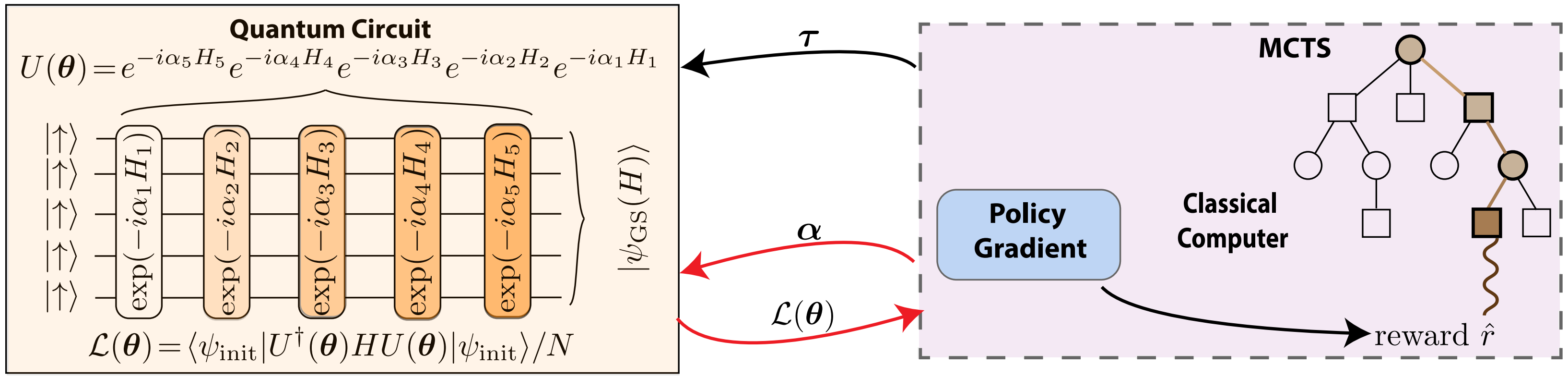


Baum et al, PRX Quantum 2, 040324 (2021)
Nguyen et al, ML Sci & Tech, 5, 025066 (2024)

- quantum circuit design and synthesis



Bolens et al, PRL 2021



Yao et al, MSML 2019, 2021, 2022

+ many more

What advantages does RL offer?

- *model-free:* requires no pre-knowledge of the controlled physical system
    - unknown sources of noise (e.g., quantum computing)
    - not fully known Hamiltonian (solid state materials, superconducting qubits, etc.)
- *adaptive:* transfers acquired knowledge that might allow us to identify connections between unrelated phenomena
- *interactive:* designed for feedback control
    - quantum feedback control
- *autonomous:* provides novel insights into automating complex manipulation protocols
    - experiments

Outline

Part 2

- RL for qubit state preparation
  - effect of noise (measurement shot noise, coherent, incoherent noise)

- experimentally friendly RL framework
  - partially observable environments
  - environment, states, actions, rewards

Outline

Part 2

- RL for qubit state preparation
    - effect of noise (measurement shot noise, coherent, incoherent noise)

- experimentally friendly RL framework
    - partially observable environments
    - environment, states, actions, rewards

RL frameworks for quantum systems

- **rewards obtained from (projective) measurements**
  - can be given only once, or else need to restart episode

- **quantum states cannot be observed**
  - extracting data from quantum system changes its state

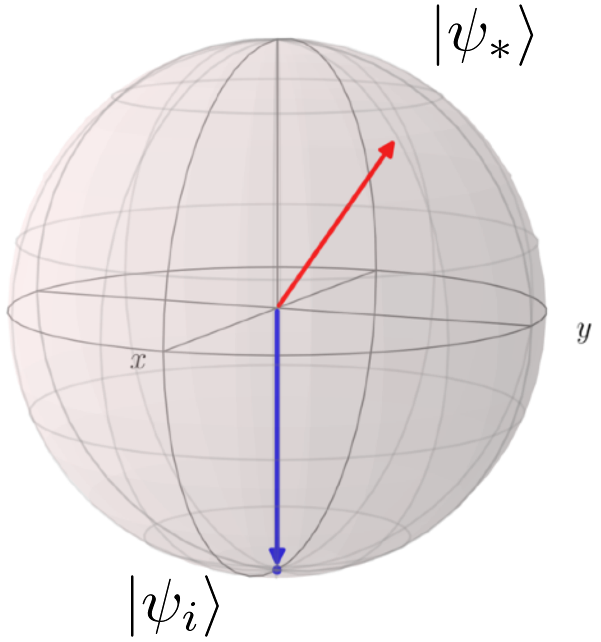
- **quantum data is binary (qubits)**
  - minimal amount of information, difficult to learn

- **quantum data is probabilistic**
  - Heisenberg uncertainty

RL for qubit control

⟶ **goal:** prepare state of a singe qubit

- initial state: $|\psi_i\rangle = |1\rangle$
- target state: $|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$



RL for qubit control

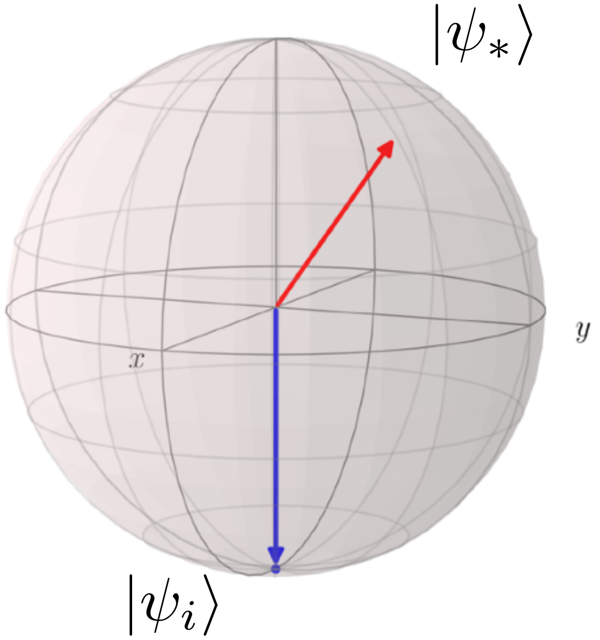
⟶ **goal:** prepare state of a singe qubit

- initial state: $|\psi_i\rangle = |1\rangle$
- target state: $|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$

⟶ use quantum gates:

$$U_{\text{ctrl}}(\alpha,\beta,\gamma) = e^{-i\gamma\sigma^z/2}e^{-i\beta\sigma^y/2}e^{-i\alpha\sigma^x/2}$$

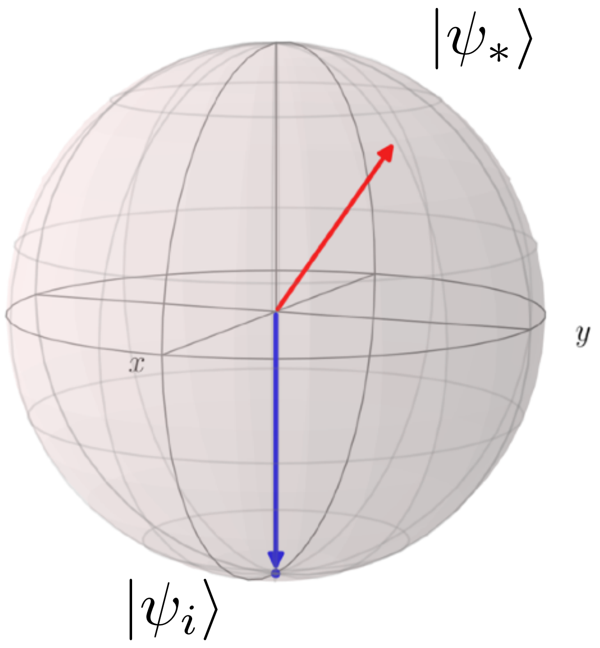
- need to determine optimal angles α, β, γ
- implements arbitrary rotations



RL for qubit control

⟶ **goal:** prepare state of a singe qubit

- initial state: $|\psi_i\rangle = |1\rangle$
- target state: $|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$



⟶ use quantum gates:

$$U_{\text{ctrl}}(\alpha,\beta,\gamma) = e^{-i\gamma\sigma^z/2}e^{-i\beta\sigma^y/2}e^{-i\alpha\sigma^x/2}$$

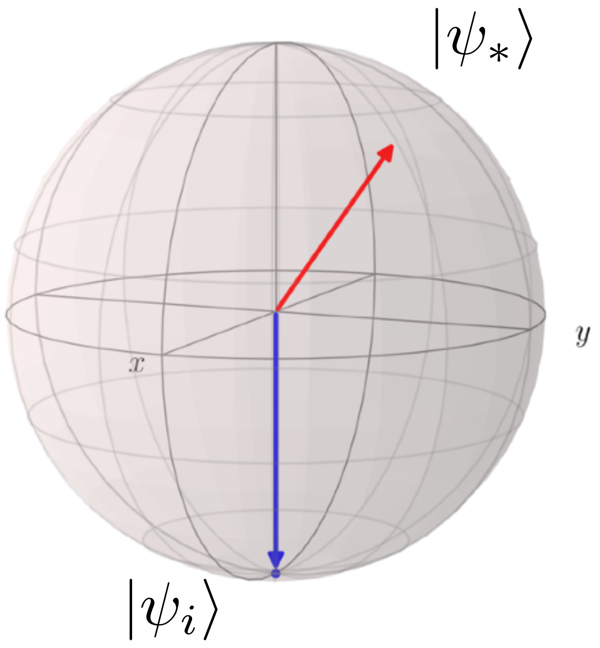
- need to determine optimal angles α, β, γ
- implements arbitrary rotations

⟶ solution: maximize the fidelity $F(\alpha,\beta,\gamma) = |\langle\psi_*|U_{\text{ctrl}}(\alpha,\beta,\gamma)|\psi_i\rangle|^2$

RL for *noisy* qubit control

⟶ **goal:** prepare state of a singe qubit

- initial state: $|\psi_i\rangle = |1\rangle$
- target state: $|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$



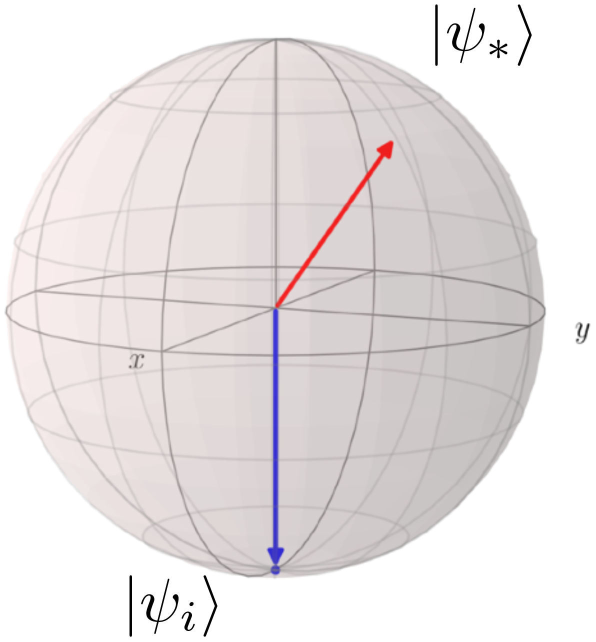
issues:

⟶ how do we compute $F = |\langle\psi_*|U_{\rm ctrl}|\psi_i\rangle|^2$?

RL for *noisy* qubit control

⟶ **goal:** prepare state of a singe qubit

- initial state: $|\psi_i\rangle = |1\rangle$
- target state: $|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$



issues:

⟶ how do we compute $F = |\langle\psi_*|U_{\rm ctrl}|\psi_i\rangle|^2$?

- need to measure state in target direction $|\psi_*\rangle$

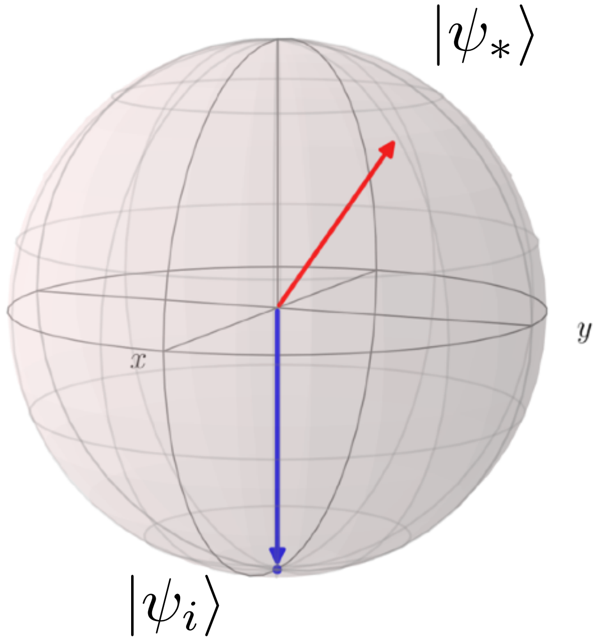
find operator whose eigenstate is $|\psi_*\rangle$: $\sigma_* = \hat{n}_* \cdot \vec{\sigma}$ $\hat{n}_* = \hat{n}_*(\theta_*, \varphi_*)$

eigenvectors: $\sigma_*|\psi_*\rangle = +|\psi_*\rangle, \quad \sigma_*|\psi_*^\perp\rangle = -|\psi_*^\perp\rangle$

RL for *noisy* qubit control

⟶ **goal:** prepare state of a singe qubit

- initial state: $|\psi_i\rangle = |1\rangle$
- target state: $|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$



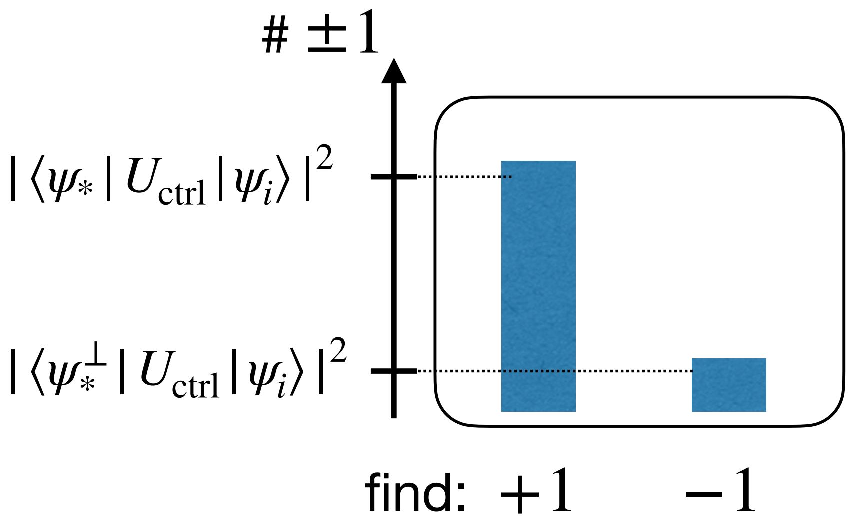
issues:

⟶ how do we compute $F = |\langle\psi_*|U_{\text{ctrl}}|\psi_i\rangle|^2$?

- need to measure state in target direction $|\psi_*\rangle$

find operator whose eigenstate is $|\psi_*\rangle$: $\sigma_* = \hat{n}_* \cdot \vec{\sigma}$ $\hat{n}_* = \hat{n}_*(\theta_*, \varphi_*)$

eigenvectors: $\sigma_*|\psi_*\rangle = +|\psi_*\rangle, \quad \sigma_*|\psi_*^\perp\rangle = -|\psi_*^\perp\rangle$

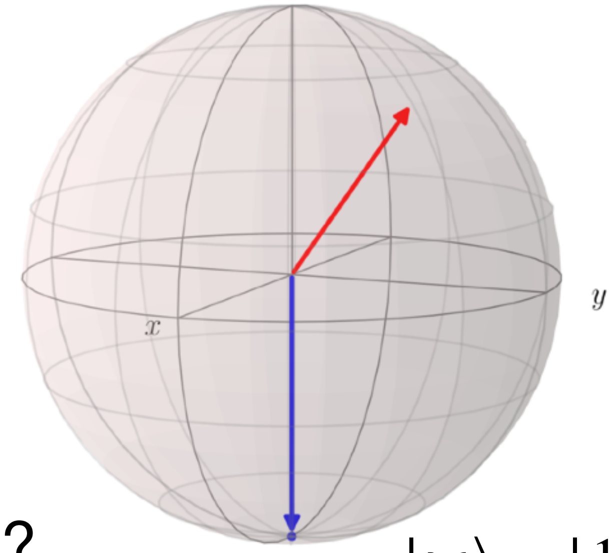


measurement shot noise!

discrete measurement outcome

RL for *noisy* qubit control

$|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$



$|\psi_i\rangle = |1\rangle$

⟶ **goal:** prepare state of a singe qubit

issues:

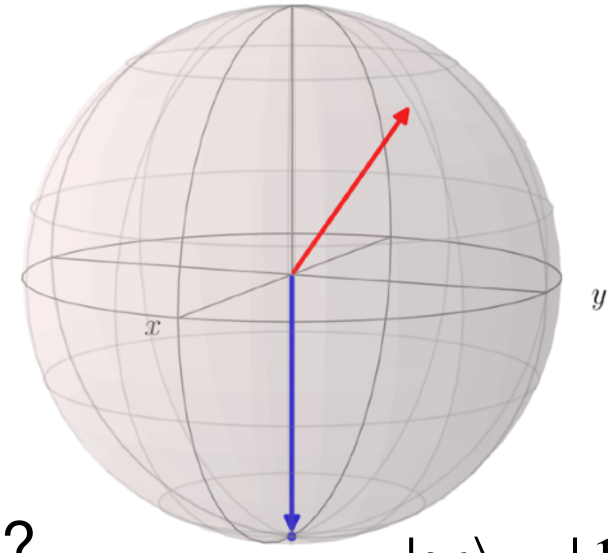
⟶ how do we compute $F = |\langle\psi_*|U_{\text{ctrl}}|\psi_i\rangle|^2$?

- measurement shot noise

⟶ how do we measure $\sigma_* = \hat{n}_* \cdot \vec{\sigma}$ on a quantum device?

RL for *noisy* qubit control

$$|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$$



$$|\psi_i\rangle = |1\rangle$$

→ **goal:** prepare state of a singe qubit

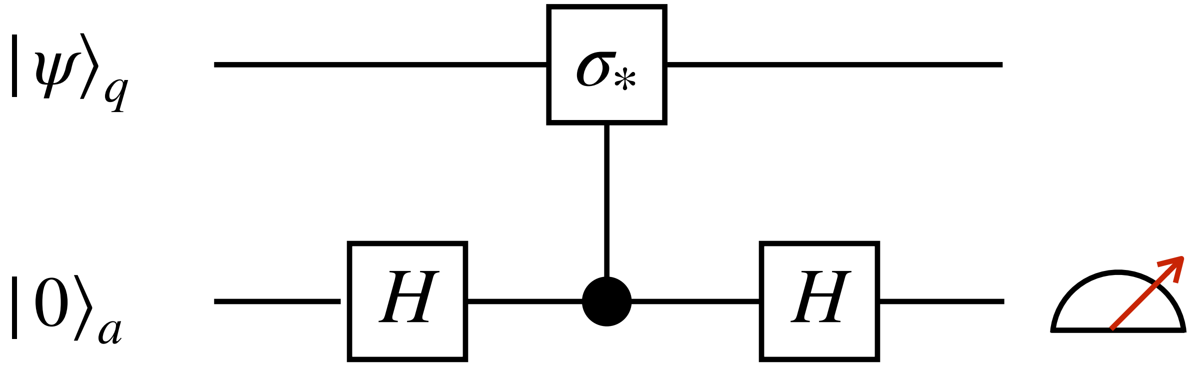
issues:

→ how do we compute $F = |\langle\psi_*|U_{\text{ctrl}}|\psi_i\rangle|^2$?

- measurement shot noise

→ how do we measure $\sigma_* = \hat{n}_* \cdot \vec{\sigma}$ on a quantum device?

- need ancilla $|\psi\rangle = |\psi\rangle_q|0\rangle_a$



measure

in z-basis

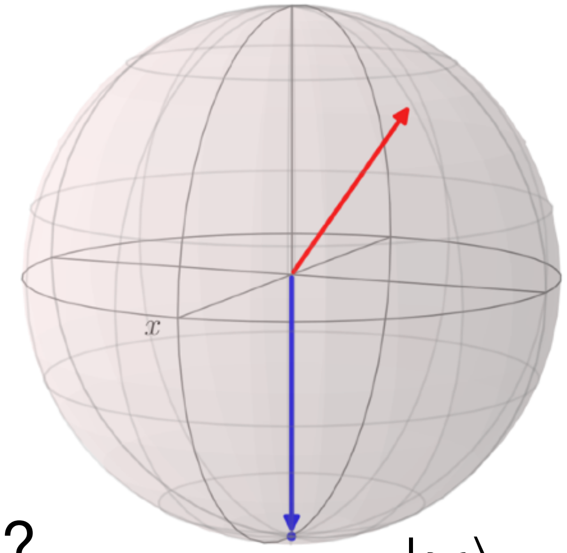
Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

controlled-σ_* gate: $C\sigma_* = \begin{cases} \sigma_{*q} & \text{if ancilla is in } |1\rangle_a \\ 1 & \text{if ancilla is in } |0\rangle_a \end{cases}$

RL for *noisy* qubit control

⟶ **goal:** prepare state of a singe qubit

$$|\psi_*\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|1\rangle$$



$|\psi_i\rangle = |1\rangle$

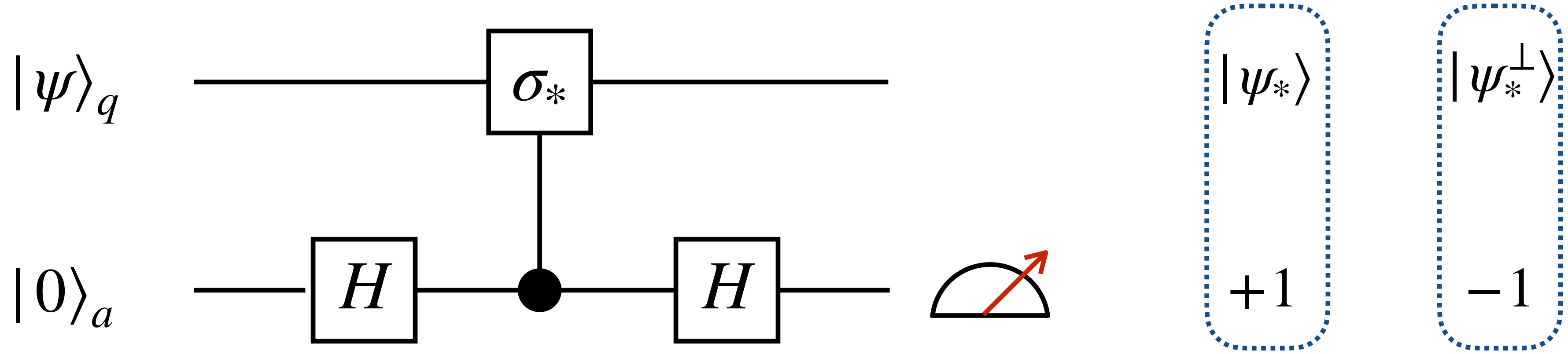
issues:

⟶ how do we compute $F = |\langle\psi_*|U_{\mathrm{ctrl}}|\psi_i\rangle|^2$?

- measurement shot noise

⟶ how do we measure $\sigma_* = \hat{n}_* \cdot \vec{\sigma}$ on a quantum device?

- need ancilla $|\psi\rangle = |\psi\rangle_q |0\rangle_a$

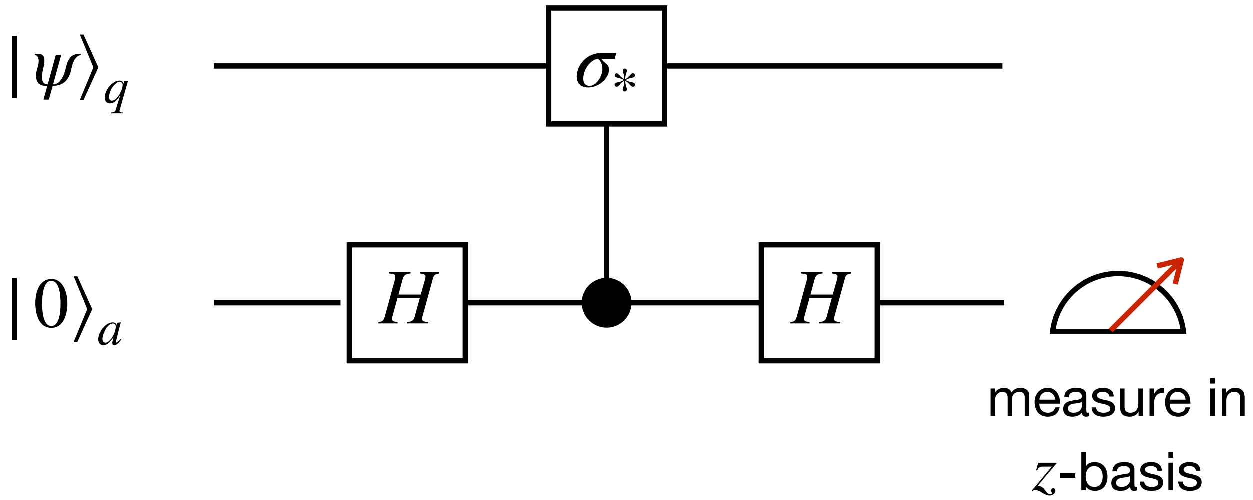


measure
in z-basis

Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

controlled-σ_* gate: $C\sigma_* = \begin{cases} \sigma_{*q} & \text{if ancilla is in } |1\rangle_a \\ 1 & \text{if ancilla is in } |0\rangle_a \end{cases}$

Measuring arbitrary operator



$\vert\psi_*\rangle$	$\vert\psi_*^\perp\rangle$
$+1$	-1

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$C\sigma_* = \begin{cases} \sigma_{*q} & \text{if ancilla is in } |1\rangle_a \\ 1 & \text{if ancilla is in } |0\rangle_a \end{cases}$$

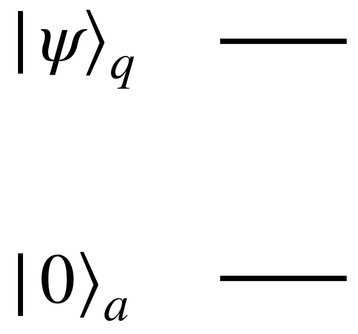
Measuring arbitrary operator

$\longrightarrow$ $|\psi\rangle_q = a\,|\psi_*\rangle_q + b\,|\psi_*^\perp\rangle_q$

$$\sigma_*\,|\psi_*\rangle = +\,|\psi_*\rangle, \quad \sigma_*\,|\psi_*^\perp\rangle = -\,|\psi_*^\perp\rangle$$

$$\sigma^z\,|0\rangle = +\,|0\rangle, \qquad \sigma^z\,|1\rangle = -\,|1\rangle$$

Measuring arbitrary operator



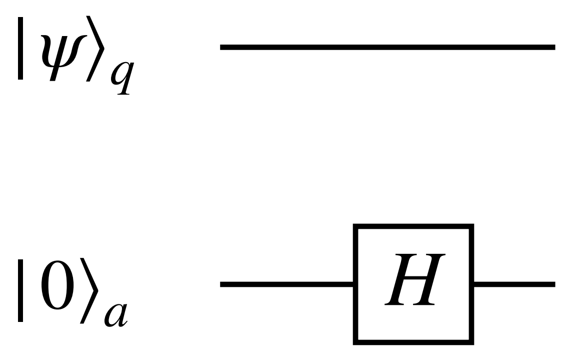
→ $|\psi\rangle_q = a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q$

→ $|\psi\rangle_q|0\rangle_a = a|\psi_*\rangle_q|0\rangle_a + b|\psi_*^\perp\rangle_q|0\rangle_a$

$\sigma_*|\psi_*\rangle = +|\psi_*\rangle, \quad \sigma_*|\psi_*^\perp\rangle = -|\psi_*^\perp\rangle$

$\sigma^z|0\rangle = +|0\rangle, \qquad \sigma^z|1\rangle = -|1\rangle$

Measuring arbitrary operator



$$H = \frac{1}{\sqrt{2}} \begin{array}{cc} & \begin{array}{cc} |0\rangle & |1\rangle \end{array} \\ & \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{array}{c} |0\rangle \\ |1\rangle \end{array} \end{array}$$

$\longrightarrow$ $$|\psi\rangle_q = a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q$$

$$\sigma_*|\psi_*\rangle = +|\psi_*\rangle, \quad \sigma_*|\psi_*^\perp\rangle = -|\psi_*^\perp\rangle$$

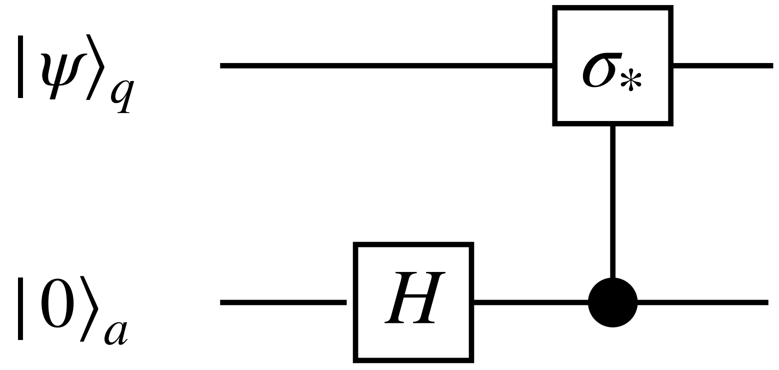
$$\sigma^z|0\rangle = +|0\rangle, \qquad \sigma^z|1\rangle = -|1\rangle$$

$\longrightarrow$ $$|\psi\rangle_q|0\rangle_a = a|\psi_*\rangle_q|0\rangle_a + b|\psi_*^\perp\rangle_q|0\rangle_a$$

$\longrightarrow$ $$H_a|\psi\rangle_q|0\rangle_a = \frac{a}{\sqrt{2}}|\psi_*\rangle_q(|0\rangle_a + |1\rangle_a) + \frac{b}{\sqrt{2}}|\psi_*^\perp\rangle_q(|0\rangle_a + |1\rangle_a)$$

$$= \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|0\rangle_a + \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|1\rangle_a$$

Measuring arbitrary operator



$$H = \frac{1}{\sqrt{2}} \begin{array}{cc} & \begin{array}{cc} |0\rangle & |1\rangle \end{array} \\ & \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{array}{c} |0\rangle \\ |1\rangle \end{array} \end{array}$$

$$C\sigma_* = \begin{cases} \sigma_{*q} & \text{if ancilla is in } |1\rangle_a \\ 1 & \text{if ancilla is in } |0\rangle_a \end{cases}$$

$$\sigma_* |\psi_*\rangle = +|\psi_*\rangle, \quad \sigma_* |\psi_*^\perp\rangle = -|\psi_*^\perp\rangle$$

$$\sigma^z |0\rangle = +|0\rangle, \qquad \sigma^z |1\rangle = -|1\rangle$$

$$\longrightarrow \qquad |\psi\rangle_q = a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q$$

$$\longrightarrow \qquad |\psi\rangle_q|0\rangle_a = a|\psi_*\rangle_q|0\rangle_a + b|\psi_*^\perp\rangle_q|0\rangle_a$$

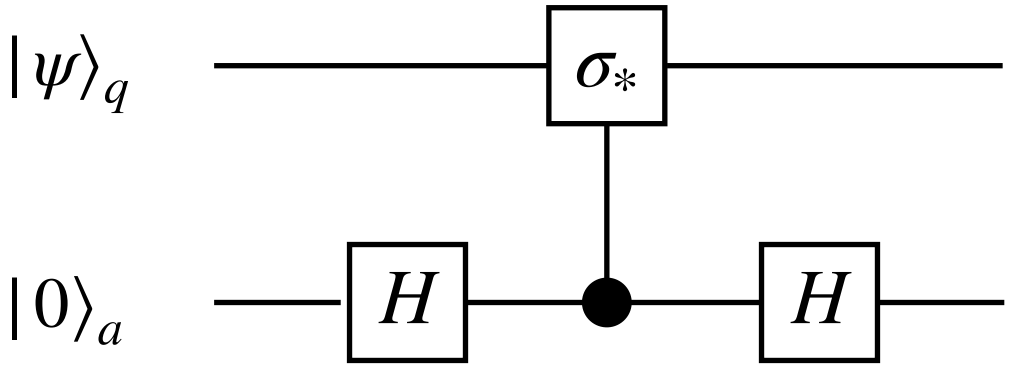
$$\longrightarrow \qquad H_a|\psi\rangle_q|0\rangle_a = \frac{a}{\sqrt{2}}|\psi_*\rangle_q(|0\rangle_a + |1\rangle_a) + \frac{b}{\sqrt{2}}|\psi_*^\perp\rangle_q(|0\rangle_a + |1\rangle_a)$$

$$= \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|0\rangle_a + \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|1\rangle_a$$

$$\longrightarrow \qquad C\sigma_* H_a|\psi\rangle_q|0\rangle_a = \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|0\rangle_a + \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q - b|\psi_*^\perp\rangle_q)|1\rangle_a$$

$$= \frac{a}{\sqrt{2}}|\psi_*\rangle_q(|0\rangle_a + |1\rangle_a) + \frac{b}{\sqrt{2}}|\psi_*^\perp\rangle_q(|0\rangle_a - |1\rangle_a)$$

Measuring arbitrary operator



$$H = \frac{1}{\sqrt{2}} \begin{array}{cc} |0\rangle & |1\rangle \\ \end{array} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{array}{c} |0\rangle \\ |1\rangle \end{array}$$

$$C\sigma_* = \begin{cases} \sigma_{*q} & \text{if ancilla is in } |1\rangle_a \\ 1 & \text{if ancilla is in } |0\rangle_a \end{cases}$$

$$\sigma_*|\psi_*\rangle = +|\psi_*\rangle, \quad \sigma_*|\psi_*^\perp\rangle = -|\psi_*^\perp\rangle$$

$$\sigma^z|0\rangle = +|0\rangle, \qquad \sigma^z|1\rangle = -|1\rangle$$

⟶ $$|\psi\rangle_q = a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q$$

⟶ $$|\psi\rangle_q|0\rangle_a = a|\psi_*\rangle_q|0\rangle_a + b|\psi_*^\perp\rangle_q|0\rangle_a$$

⟶ $$H_a|\psi\rangle_q|0\rangle_a = \frac{a}{\sqrt{2}}|\psi_*\rangle_q(|0\rangle_a + |1\rangle_a) + \frac{b}{\sqrt{2}}|\psi_*^\perp\rangle_q(|0\rangle_a + |1\rangle_a)$$

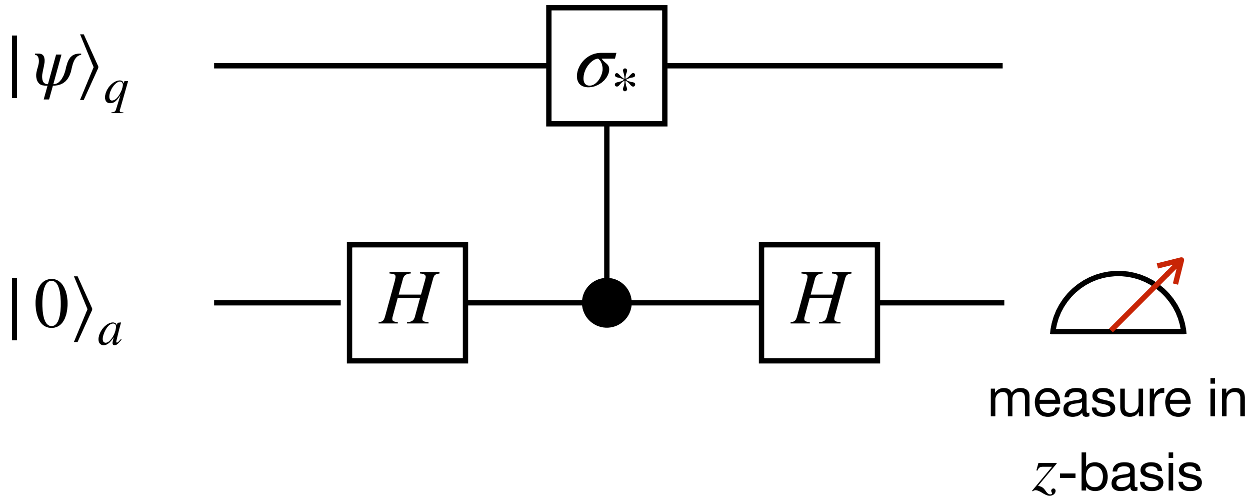
$$= \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|0\rangle_a + \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|1\rangle_a$$

⟶ $$C\sigma_*H_a|\psi\rangle_q|0\rangle_a = \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|0\rangle_a + \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q - b|\psi_*^\perp\rangle_q)|1\rangle_a$$

$$= \frac{a}{\sqrt{2}}|\psi_*\rangle_q(|0\rangle_a + |1\rangle_a) + \frac{b}{\sqrt{2}}|\psi_*^\perp\rangle_q(|0\rangle_a - |1\rangle_a)$$

⟶ $$H_aC\sigma_*H_a|\psi\rangle_q|0\rangle_a = a|\psi_*\rangle_q|0\rangle_a + b|\psi_*^\perp\rangle_q|1\rangle_a$$

Measuring arbitrary operator



$|\psi_*\rangle$ $+1$

$|\psi_*^\perp\rangle$ -1

$$H = \frac{1}{\sqrt{2}} \begin{array}{c|c} & |0\rangle \quad |1\rangle \\ \hline \begin{matrix} |0\rangle \\ |1\rangle \end{matrix} & \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{array}$$

$$C\sigma_* = \begin{cases} \sigma_{*q} & \text{if ancilla is in } |1\rangle_a \\ 1 & \text{if ancilla is in } |0\rangle_a \end{cases}$$

$$\longrightarrow \quad |\psi\rangle_q = a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q$$

$$\sigma_*|\psi_*\rangle = +|\psi_*\rangle, \quad \sigma_*|\psi_*^\perp\rangle = -|\psi_*^\perp\rangle$$

$$\sigma^z|0\rangle = +|0\rangle, \quad \sigma^z|1\rangle = -|1\rangle$$

$$\longrightarrow \quad |\psi\rangle_q|0\rangle_a = a|\psi_*\rangle_q|0\rangle_a + b|\psi_*^\perp\rangle_q|0\rangle_a$$

$$\longrightarrow \quad H_a|\psi\rangle_q|0\rangle_a = \frac{a}{\sqrt{2}}|\psi_*\rangle_q(|0\rangle_a + |1\rangle_a) + \frac{b}{\sqrt{2}}|\psi_*^\perp\rangle_q(|0\rangle_a + |1\rangle_a)$$

$$= \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|0\rangle_a + \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|1\rangle_a$$

$$\longrightarrow \quad C\sigma_* H_a|\psi\rangle_q|0\rangle_a = \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q + b|\psi_*^\perp\rangle_q)|0\rangle_a + \frac{1}{\sqrt{2}}(a|\psi_*\rangle_q - b|\psi_*^\perp\rangle_q)|1\rangle_a$$

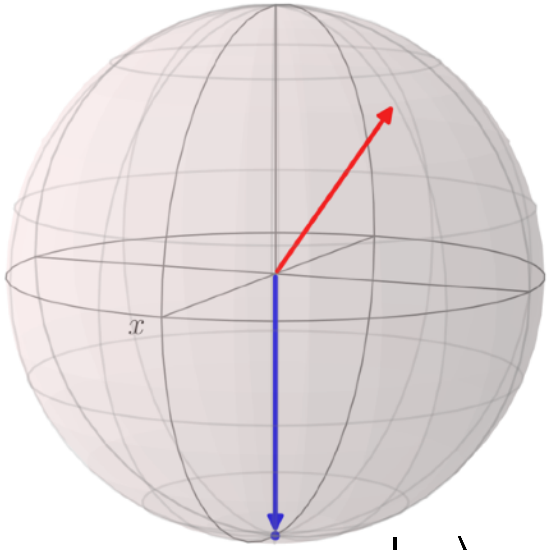
$$= \frac{a}{\sqrt{2}}|\psi_*\rangle_q(|0\rangle_a + |1\rangle_a) + \frac{b}{\sqrt{2}}|\psi_*^\perp\rangle_q(|0\rangle_a - |1\rangle_a)$$

$$\longrightarrow \quad H_a C\sigma_* H_a|\psi\rangle_q|0\rangle_a = a|\psi_*\rangle_q|0\rangle_a + b|\psi_*^\perp\rangle_q|1\rangle_a$$

RL for ***noisy*** qubit control

$|\psi_*\rangle = \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|0\rangle + \cos\frac{\pi}{8}|1\rangle$

⟶ **goal:** prepare state of a singe qubit



$|\psi_i\rangle = |0\rangle$

issues:

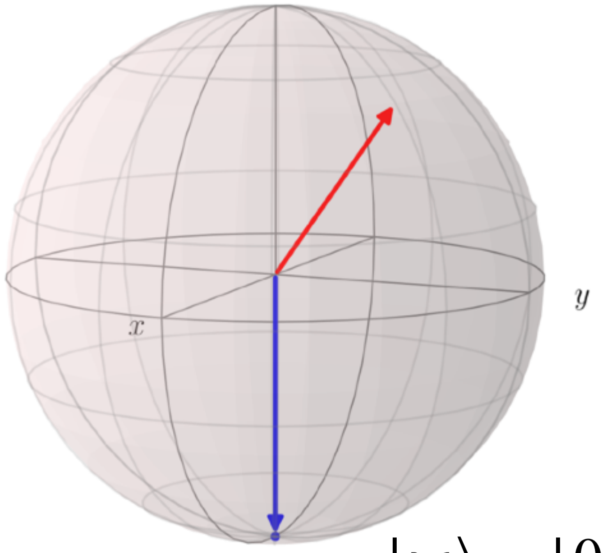
1. how do we compute $F = |\langle\psi_*|U_{\rm ctrl}|\psi_i\rangle|^2$?

2. qubit + ancilla apparatus can be noisy

   - coherent noise

RL for ***noisy*** qubit control

$|\psi_*\rangle = \sin\frac{\pi}{8} e^{i\frac{\pi}{3}}|0\rangle + \cos\frac{\pi}{8}|1\rangle$



$|\psi_i\rangle = |0\rangle$

⟶ **goal:** prepare state of a singe qubit

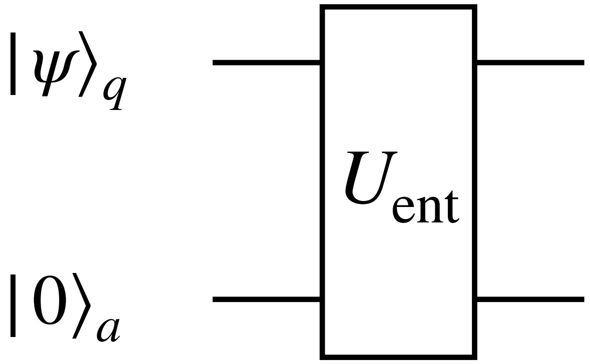
issues:

1. how do we compute $F = |\langle\psi_*|U_{\text{ctrl}}|\psi_i\rangle|^2$?

2. qubit + ancilla apparatus can be noisy

   - coherent noise

$|\psi\rangle = |\psi\rangle_q|0\rangle_a$



$U_{\text{ent}} = e^{-ia\sigma_a^x\sigma_q^x}\, e^{-ib\sigma_a^y\sigma_q^y}\, e^{-ic\sigma_a^z\sigma_q^z}$

noise strength set by a, b, c

qubit and ancilla can get entangled

RL for *noisy* qubit control

→ **goal:** prepare state of a singe qubit

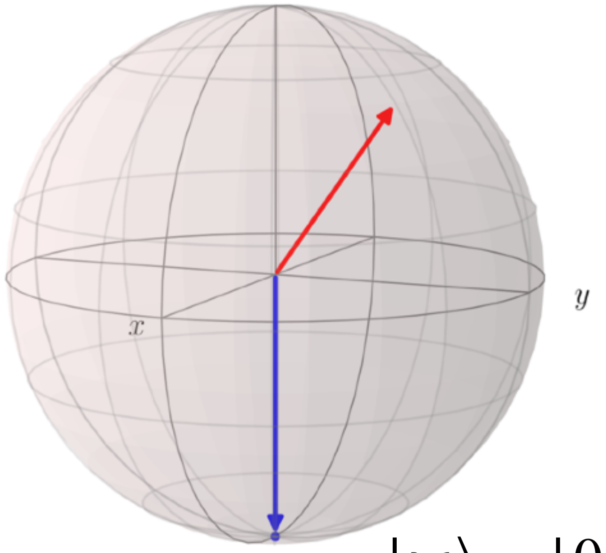
issues:

1. how do we compute $F = |\langle\psi_*|U_{\text{ctrl}}|\psi_i\rangle|^2$?

2. qubit + ancilla apparatus can be noisy

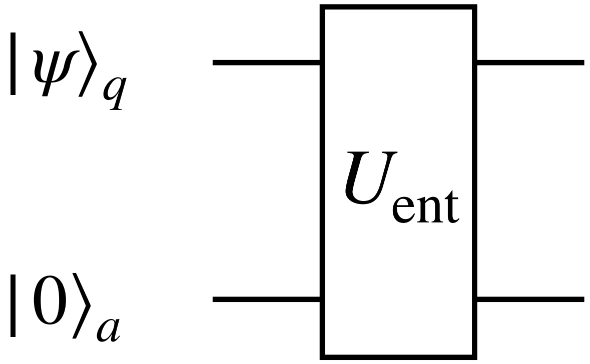
   - coherent noise

$$|\psi_*\rangle = \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|0\rangle + \cos\frac{\pi}{8}|1\rangle$$



$$|\psi_i\rangle = |0\rangle$$

$|\psi\rangle = |\psi\rangle_q|0\rangle_a$



$$U_{\text{ent}} = e^{-ia\sigma_a^x\sigma_q^x}\,e^{-ib\sigma_a^y\sigma_q^y}\,e^{-ic\sigma_a^z\sigma_q^z}$$

noise strength set by a, b, c

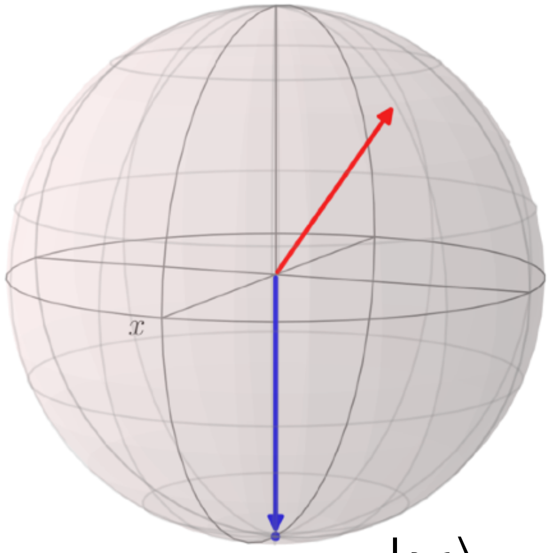
qubit and ancilla can get entangled

how do we get rid of this entanglement? measure ancilla in z-basis to reset it!

RL for *noisy* qubit control

⟶ **goal:** prepare state of a singe qubit

$|\psi_*\rangle = \sin\frac{\pi}{8} e^{i\frac{\pi}{3}} |0\rangle + \cos\frac{\pi}{8} |1\rangle$



$|\psi_i\rangle = |0\rangle$

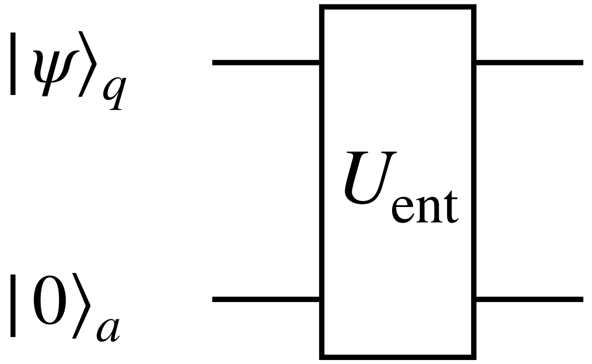
issues:

1. how do we compute $F = |\langle\psi_*|U_{\rm ctrl}|\psi_i\rangle|^2$?

2. qubit + ancilla apparatus can be noisy

   - coherent noise

$|\psi\rangle = |\psi\rangle_q |0\rangle_a$



$U_{\rm ent} = e^{-ia\sigma^x_a\sigma^x_q}\, e^{-ib\sigma^y_a\sigma^y_q}\, e^{-ic\sigma^z_a\sigma^z_q}$

noise strength set by a, b, c

qubit and ancilla can get entangled

how do we get rid of this entanglement? measure ancilla in z-basis to reset it!

$$|\psi\rangle \longrightarrow \begin{cases} \frac{P^z_a|\psi\rangle}{\sqrt{p}}, & \text{with prob. } p = |\langle\psi|\, P^z_a\, |\psi\rangle|^2 \quad \& \text{ measurement outcome } -1 \\ \frac{(1-P^z_a)|\psi\rangle}{\sqrt{1-p}}, & \text{with prob. } 1-p \quad \& \text{ measurement outcome } +1 \end{cases}$$

$$P^z_a = 1_q \otimes \frac{1}{2}(1-\sigma^z)_a$$

RL for *noisy* qubit control

⟶ **goal:** prepare state of a singe qubit

issues:

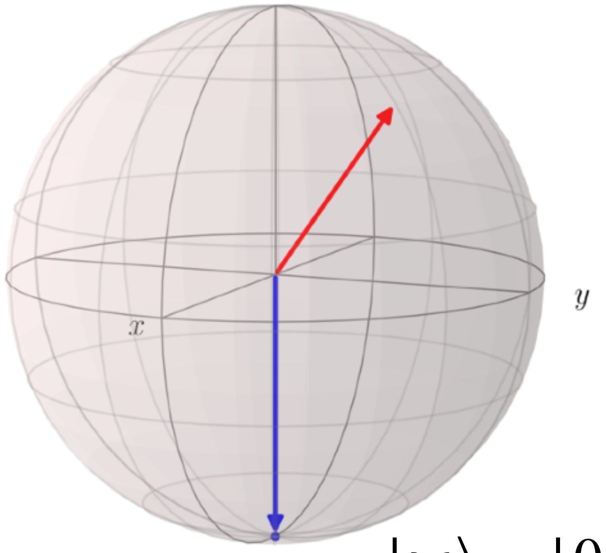
1. how do we compute $F = |\langle \psi_* | U_{\text{ctrl}} | \psi_i \rangle|^2$?

2. qubit + ancilla apparatus can be noisy

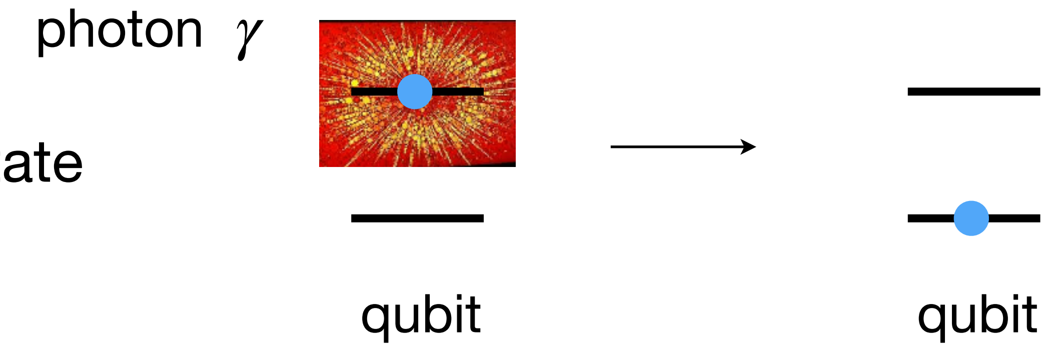
   - coherent noise

   - incoherent noise: spontaneous decay of qubit state

$$|\psi_*\rangle = \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|0\rangle + \cos\frac{\pi}{8}|1\rangle$$



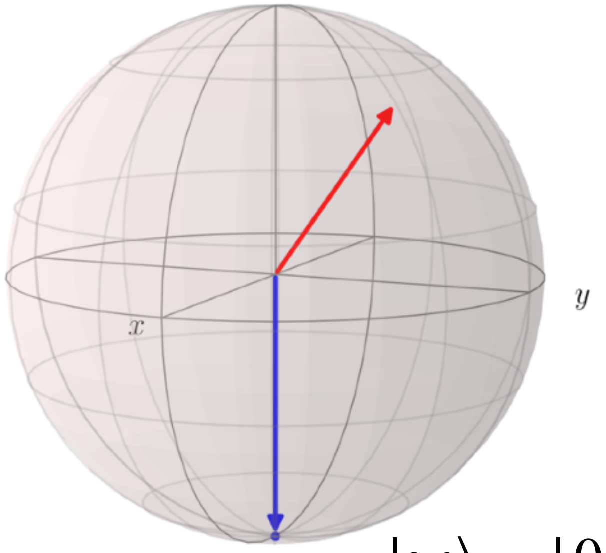
$$|\psi_i\rangle = |0\rangle$$



RL for ***noisy*** qubit control

$$|\psi_*\rangle = \sin\frac{\pi}{8} e^{i\frac{\pi}{3}}|0\rangle + \cos\frac{\pi}{8}|1\rangle$$

→ **goal:** prepare state of a singe qubit



$$|\psi_i\rangle = |0\rangle$$

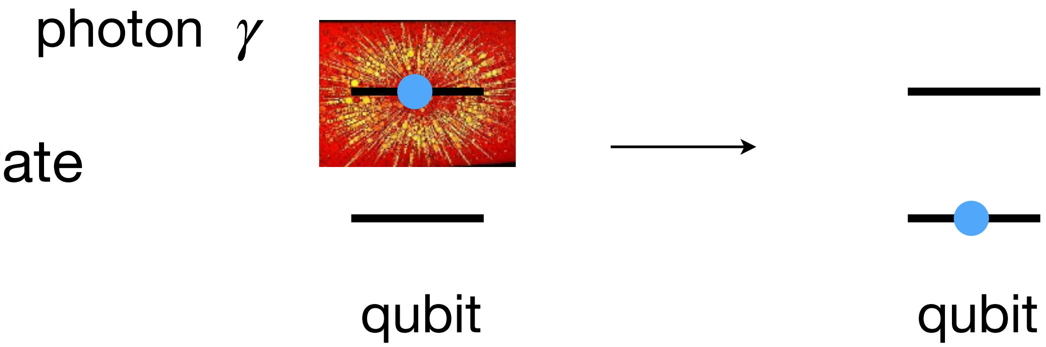
issues:

1. how do we compute $F = |\langle\psi_*|U_{\rm ctrl}|\psi_i\rangle|^2$?

2. qubit + ancilla apparatus can be noisy

- coherent noise

- incoherent noise: spontaneous decay of qubit state



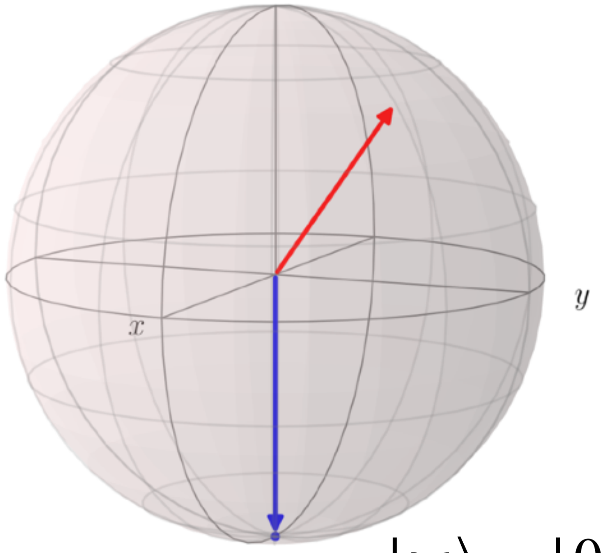
$$|\psi\rangle_q \longrightarrow \begin{cases} \frac{P^z_q|\psi\rangle}{\sqrt{|\langle\psi|P_z|\psi\rangle_q}}, & \text{with prob. } p_{\rm emit} \\ |\psi\rangle_q\,, & \text{with prob. } 1-p_{\rm emit} \end{cases}$$

$$P^z_q = \frac{1}{2}(1-\sigma^z)_q$$

RL for noisy qubit control

→ **goal:** prepare state of a singe qubit

$|\psi_*\rangle = \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|0\rangle + \cos\frac{\pi}{8}|1\rangle$

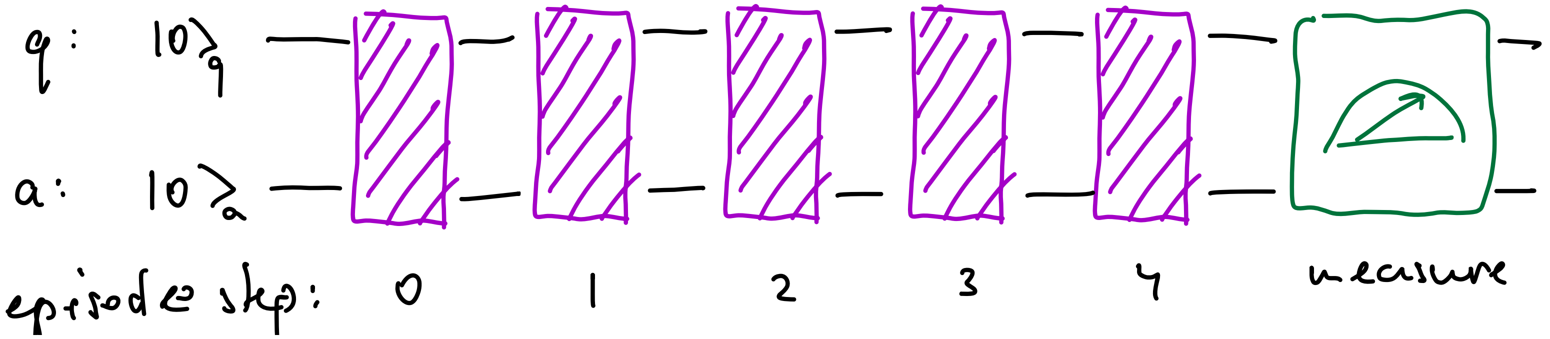


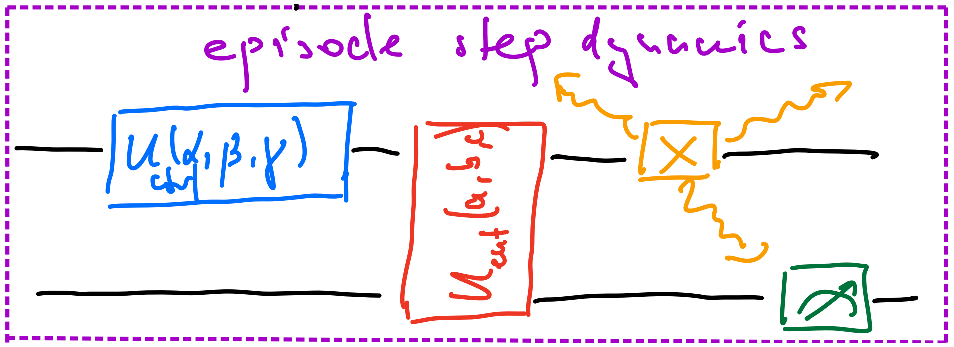
$|\psi_i\rangle = |0\rangle$

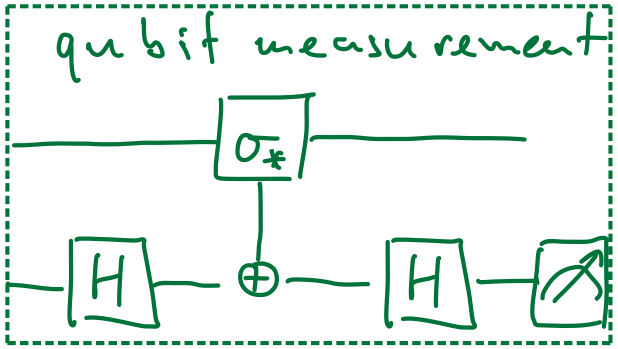
issues:

1. how do we compute $F = |\langle\psi_*|U_{\text{ctrl}}|\psi_i\rangle|^2$?

2. qubit + ancilla apparatus can be noisy



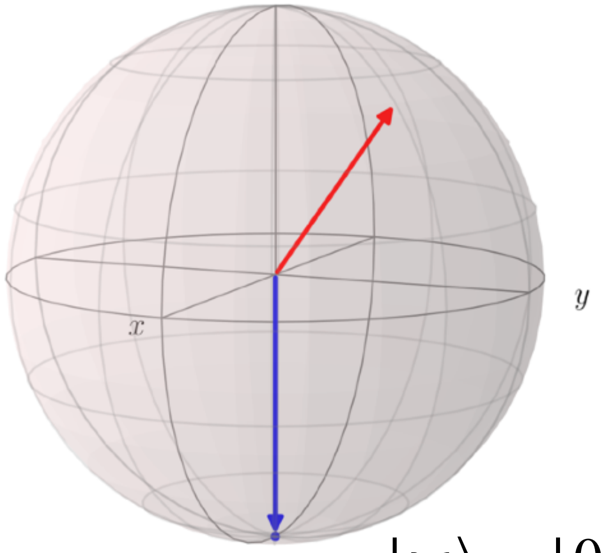




RL for noisy qubit control

→ **goal:** prepare state of a singe qubit

$$|\psi_*\rangle = \sin\frac{\pi}{8}e^{i\frac{\pi}{3}}|0\rangle + \cos\frac{\pi}{8}|1\rangle$$

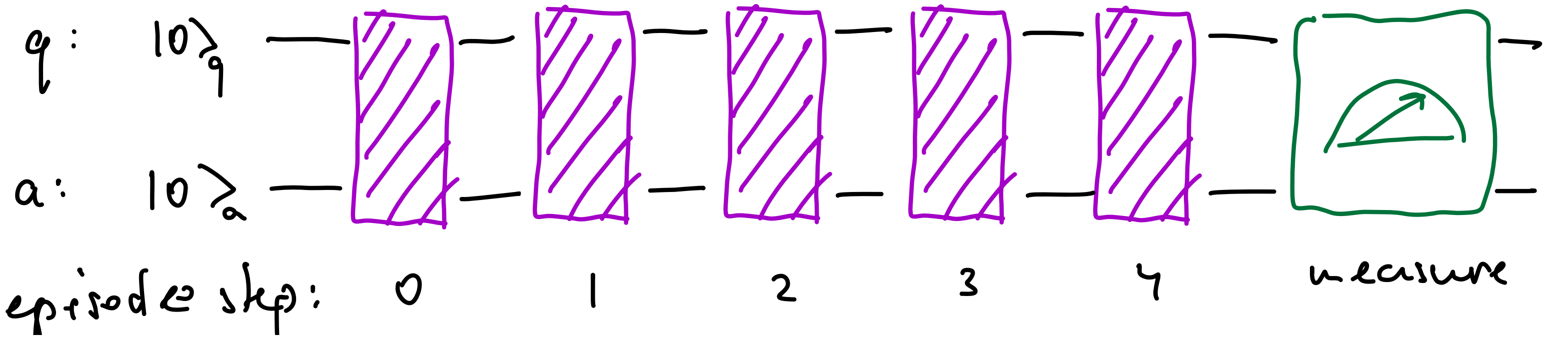


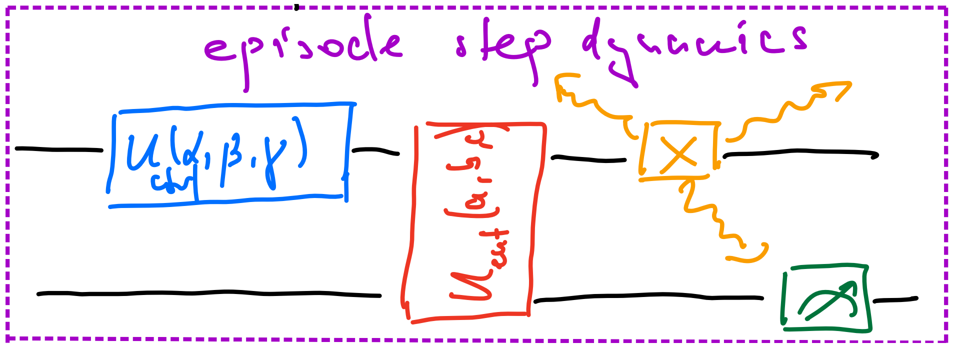
$|\psi_i\rangle = |0\rangle$

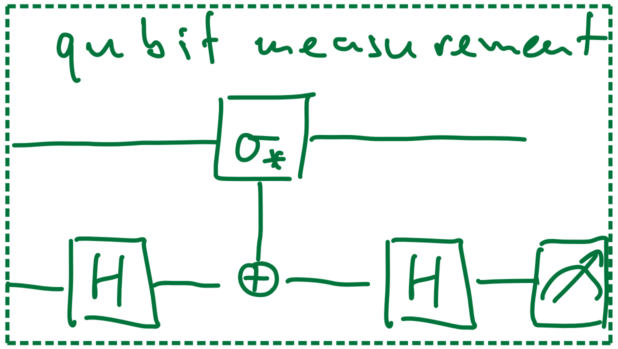
issues:

1. how do we compute $F = |\langle\psi_*|U_{\text{ctrl}}|\psi_i\rangle|^2$?

2. qubit + ancilla apparatus can be noisy







3. accessible data: is all binary!!

- ancilla measurement
- single photon detection
- qubit measurement

Outline

Part 2

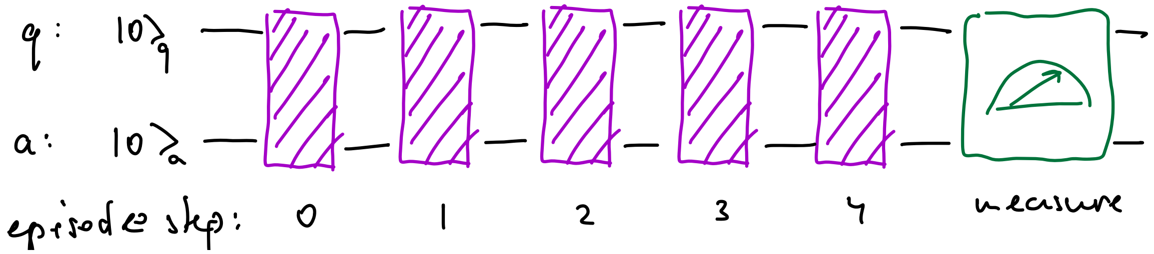
- RL for qubit state preparation
  - effect of noise (measurement shot noise, coherent, incoherent noise)

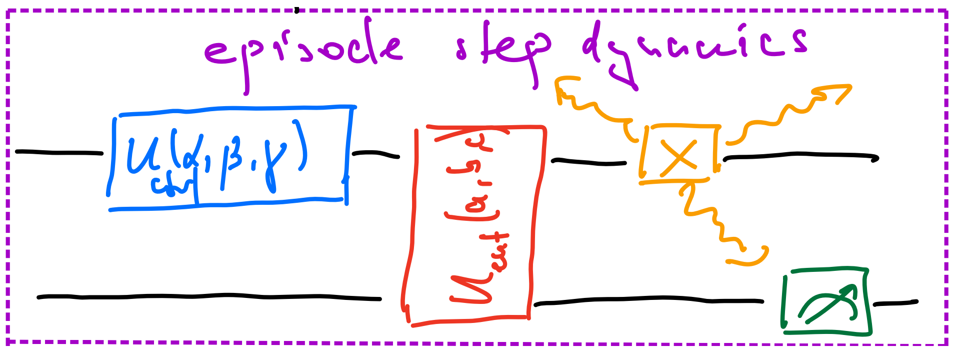
- experimentally friendly RL framework
  - partially observable environments
  - environment, states, actions, rewards

RL framework

→ **rewards**

- qubit measurement output: ± 1 (binary)





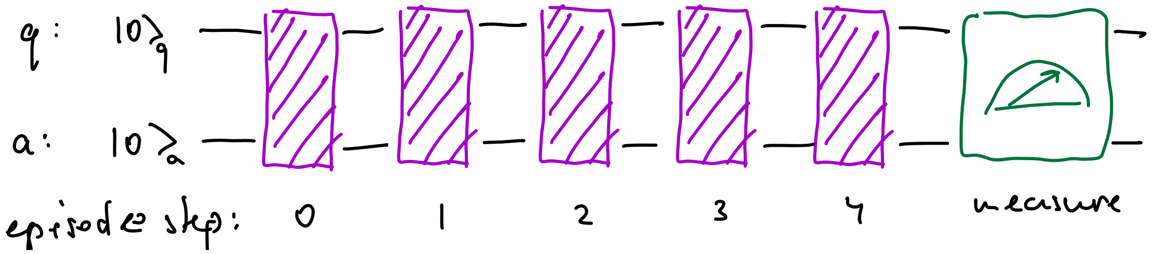
RL framework

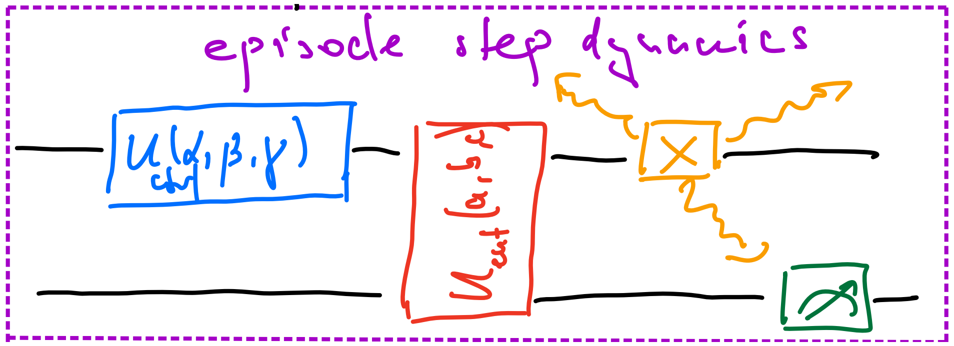
→ **rewards**

- qubit measurement output: ± 1 (binary)

→ **actions**

- angles of control unitary $U(\alpha, \beta, \gamma)$: continuous (!)

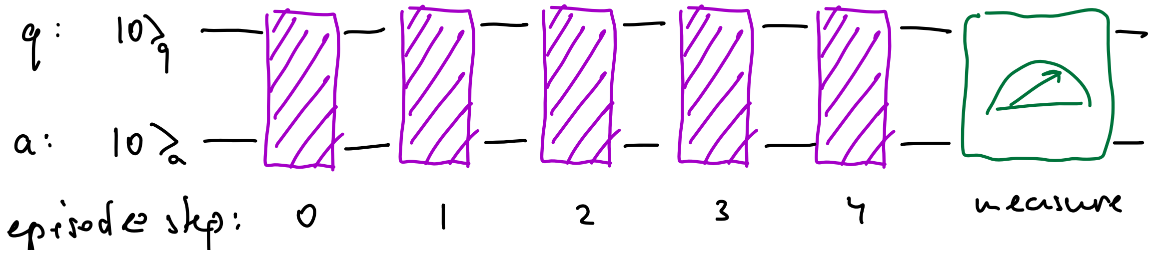


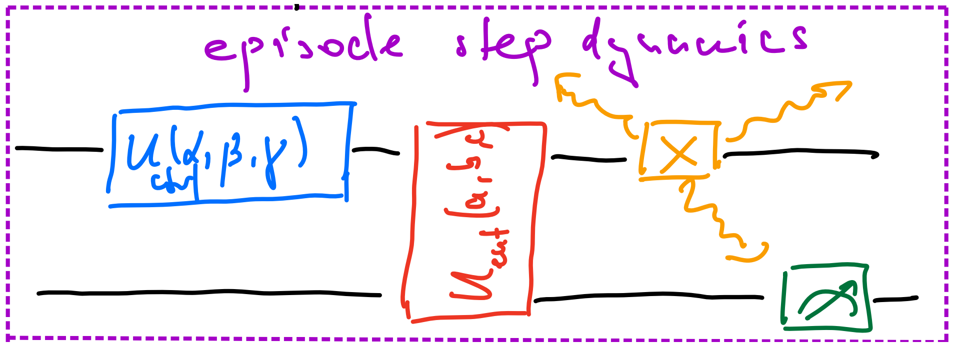


RL framework

→ **rewards**

- qubit measurement output: ± 1 (binary)





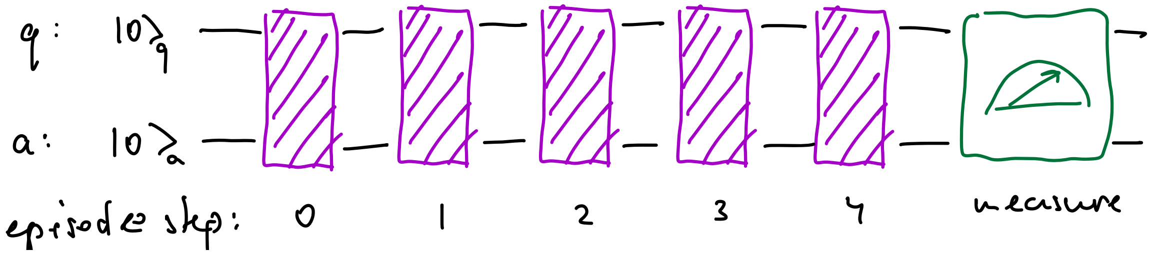
→ **actions**

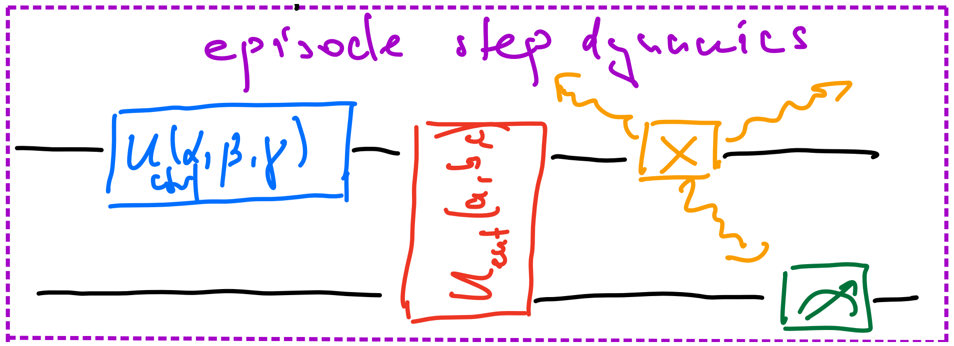
- angles of control unitary $U(\alpha, \beta, \gamma)$: continuous (!)

→ **states**

quantum states cannot be measured/observed!!!

RL framework





→ **rewards**

- qubit measurement output: ± 1 (binary)

→ **actions**

- angles of control unitary $U(\alpha, \beta, \gamma)$: continuous (!)

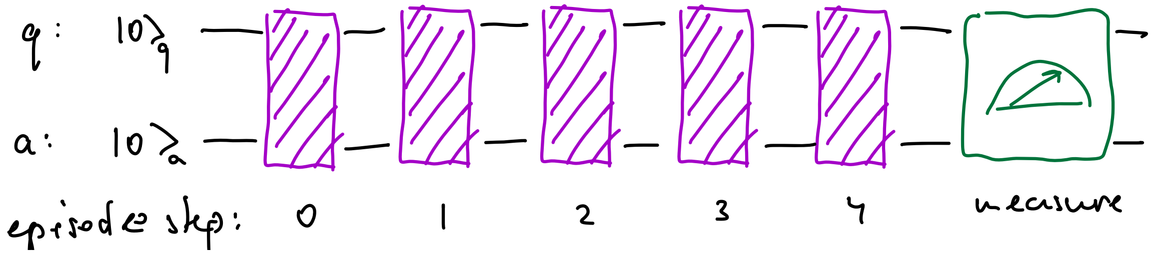
→ states $\longrightarrow$ **observations**

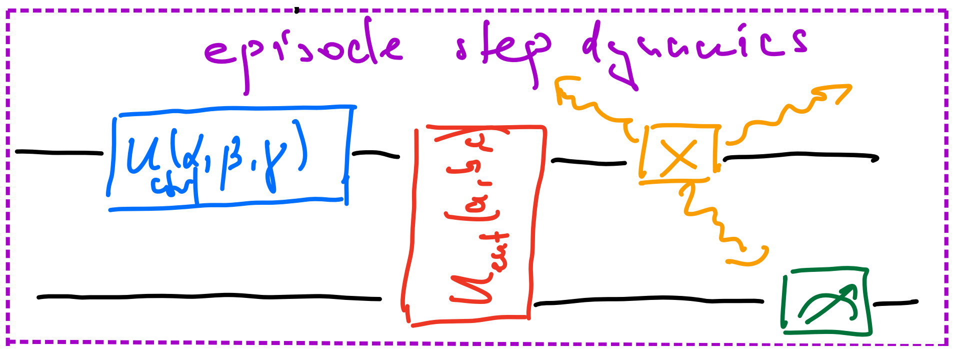
- wall clock time: one-hot representation of the step number

$$\begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix}$$

quantum states cannot be measured/observed!!!

RL framework





→ **rewards**

- qubit measurement output: ± 1 (binary)

→ **actions**

- angles of control unitary $U(\alpha, \beta, \gamma)$: continuous (!)

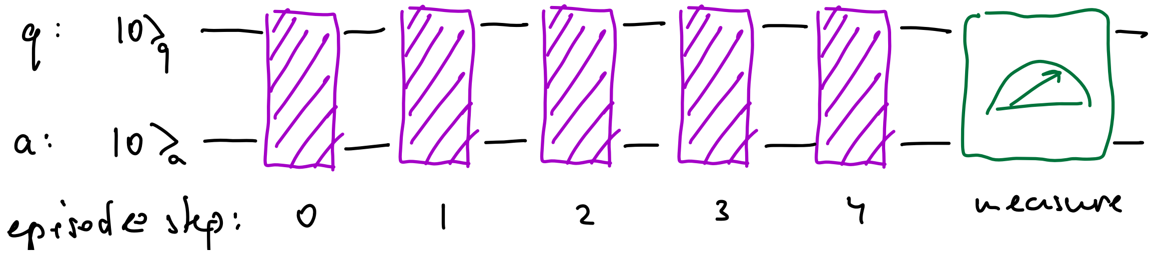
→ states ⟶ **observations**

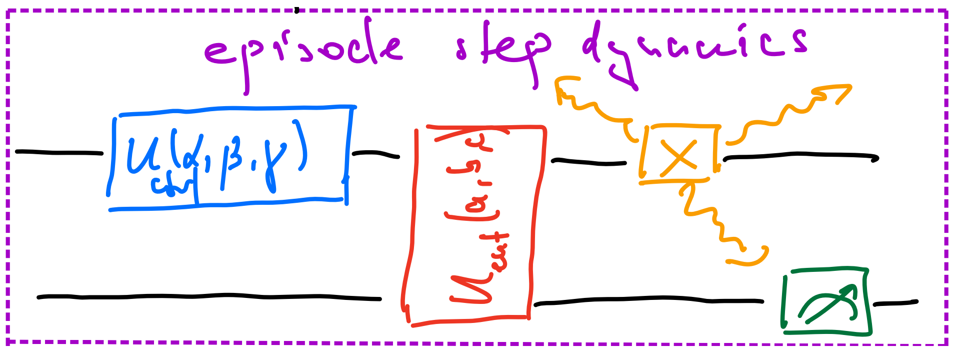
- wall clock time: one-hot representation of the step number
- ancilla measurement output: ± 1 (binary)

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$$

quantum states cannot be measured/observed!!!

RL framework





→ **rewards**

- qubit measurement output: ± 1 (binary)

→ **actions**

- angles of control unitary $U(\alpha,\beta,\gamma)$: continuous (!)

→ states ⟶ **observations**

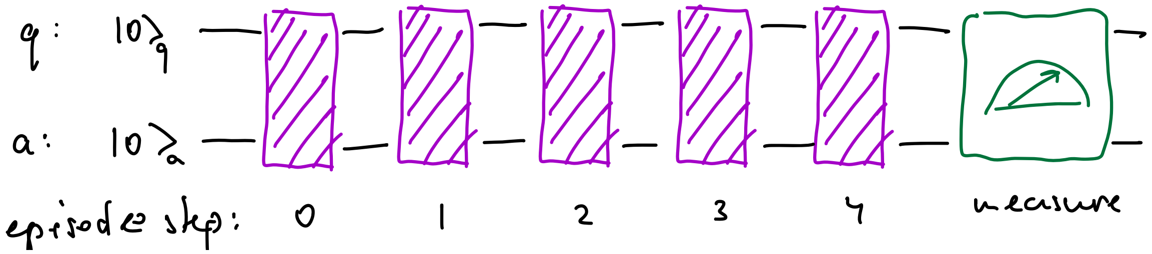
- wall clock time: one-hot representation of the step number
- ancilla measurement output: ± 1 (binary)
- spontaneously emitted photon detection: ± 1 (binary)

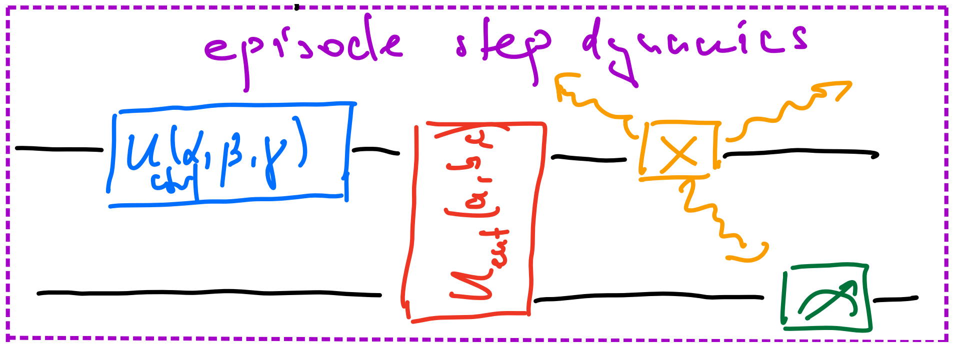
-1 = 'qubit found in GS'
(photon detected)

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

quantum states cannot be measured/observed!!!

RL framework





⟶ **rewards**

- qubit measurement output: ± 1 (binary)

⟶ **actions**

- angles of control unitary $U(\alpha, \beta, \gamma)$: continuous (!)

⟶ states ⟶ **observations**

- wall clock time: one-hot representation of the step number
- ancilla measurement output: ± 1 (binary)
- spontaneously emitted photon detection: ± 1 (binary)

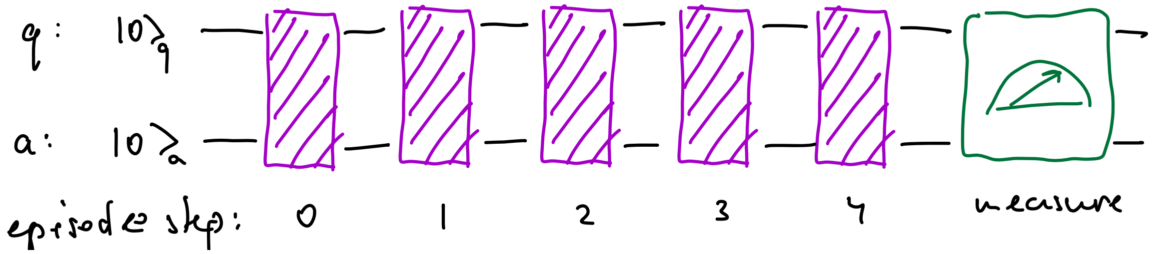
-1 = ‘qubit found in GS’
(photon detected)

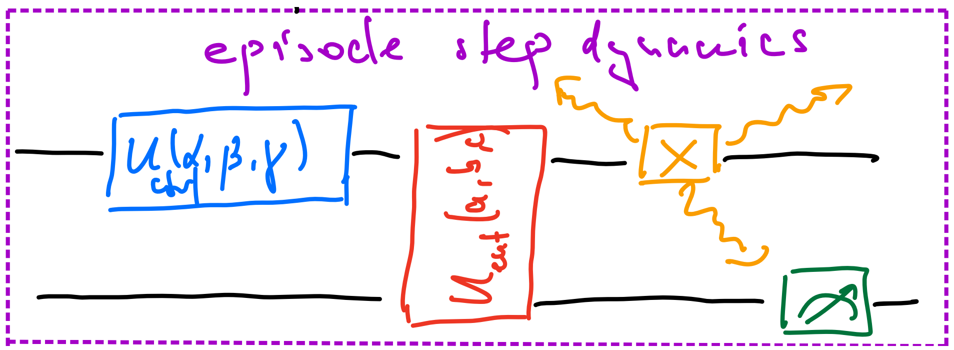
$$\begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix} \quad \begin{pmatrix}1\\1\\1\\-1\\1\end{pmatrix} \quad \begin{pmatrix}1\\1\\-1\\1\\1\end{pmatrix} \quad \longrightarrow \quad \begin{pmatrix}0 & 1 & 1\\1 & 1 & 1\\0 & 1 & -1\\0 & -1 & 1\\0 & 1 & 1\end{pmatrix} = o_t$$

RL observation

quantum states cannot be measured/observed!!!

RL framework





→ **rewards**

- qubit measurement output: ± 1 (binary)

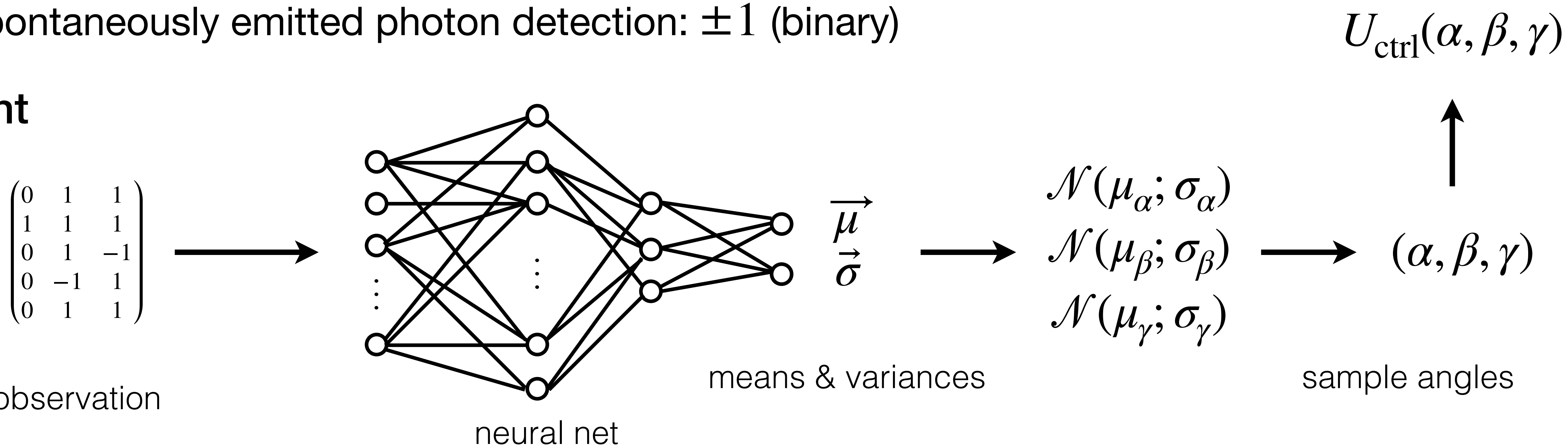
→ **actions**

- angles of control unitary $U(\alpha, \beta, \gamma)$: continuous (!)

→ states ⟶ **observations**

- wall clock time: one-hot representation of the step number
- ancilla measurement output: ± 1 (binary)
- spontaneously emitted photon detection: ± 1 (binary)

→ **agent**



$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \longrightarrow \text{neural net} \quad \vec{\mu}, \vec{\sigma} \longrightarrow \begin{matrix} \mathcal{N}(\mu_\alpha; \sigma_\alpha) \\ \mathcal{N}(\mu_\beta; \sigma_\beta) \\ \mathcal{N}(\mu_\gamma; \sigma_\gamma) \end{matrix} \longrightarrow (\alpha, \beta, \gamma) \longrightarrow U_{\text{ctrl}}(\alpha, \beta, \gamma)$$

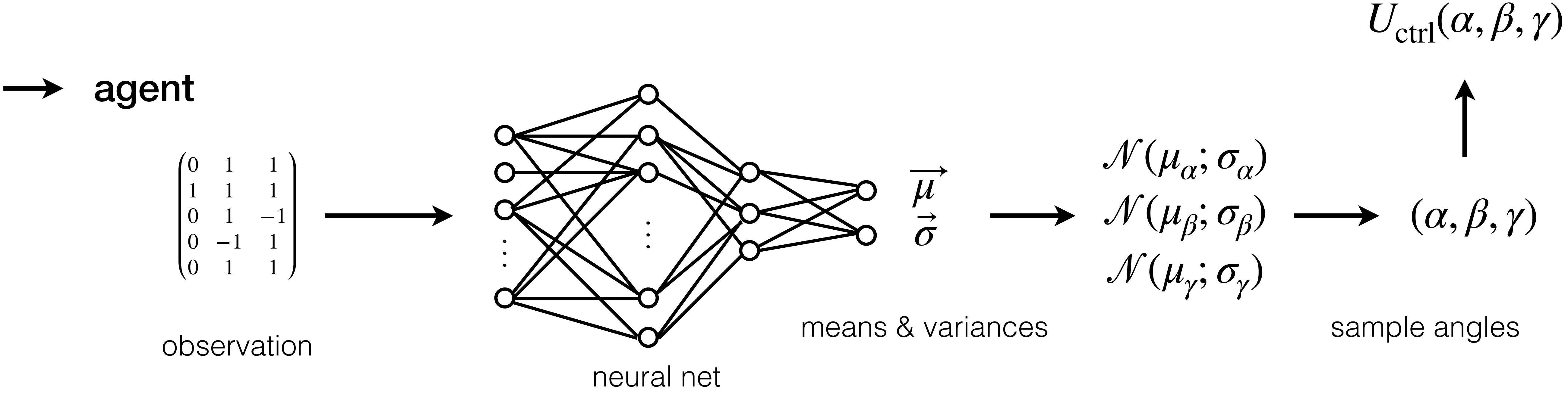
observation

neural net

means & variances

sample angles

Training curves



→ agent

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

observation

neural net

$\vec{\mu}$ $\vec{\sigma}$

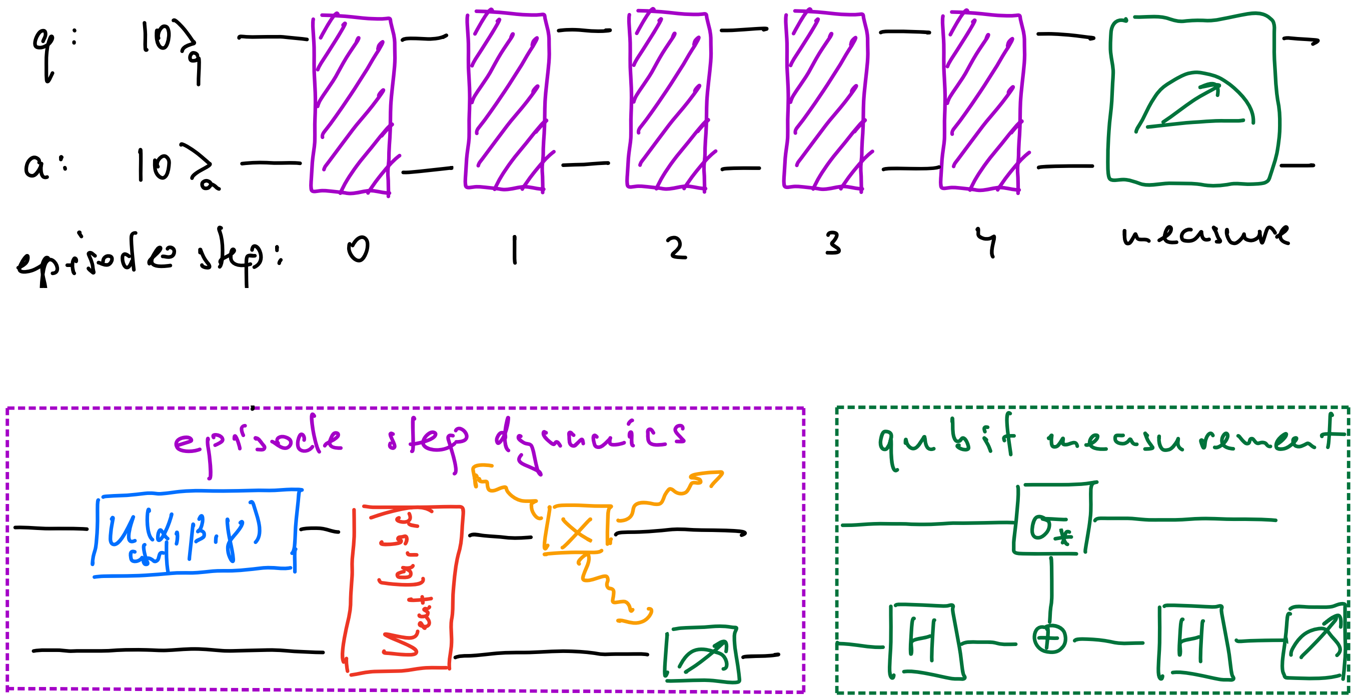
means & variances

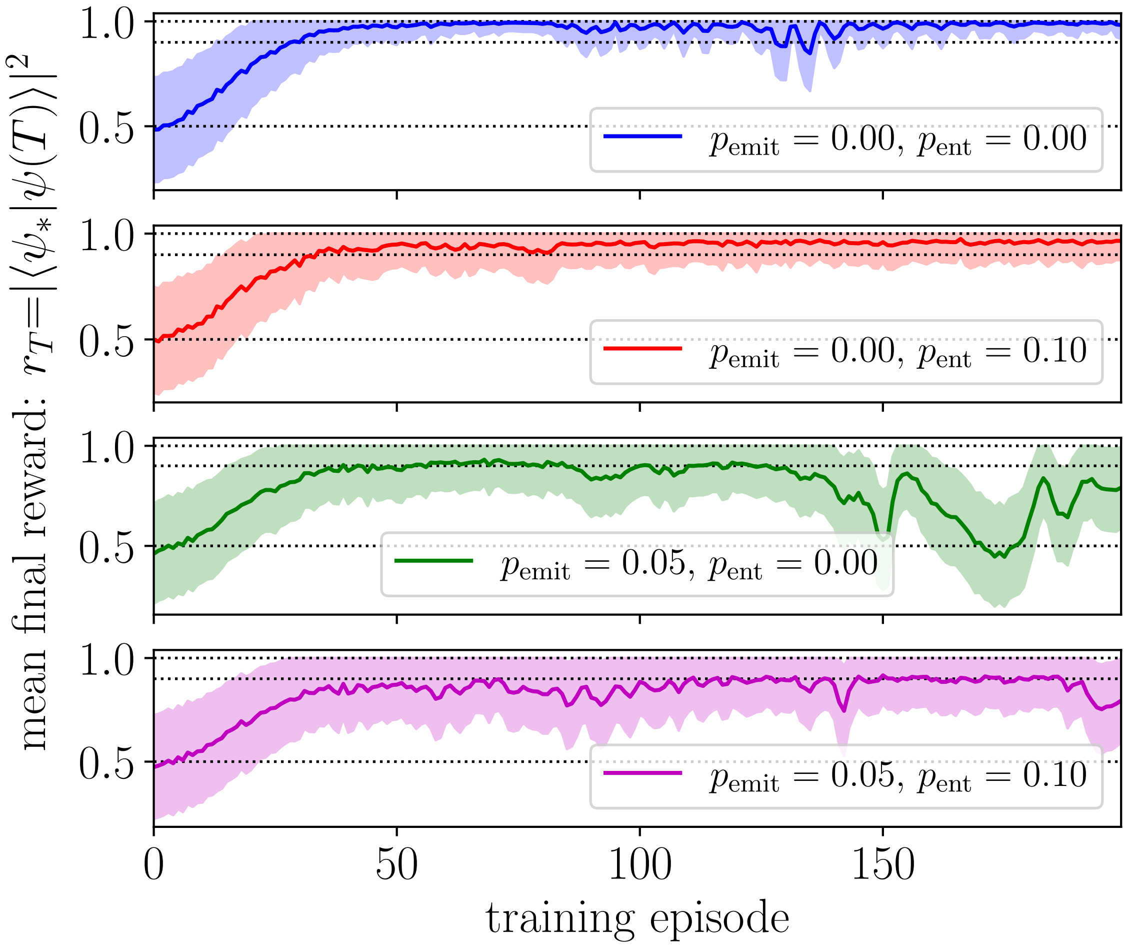
$\mathcal{N}(\mu_\alpha; \sigma_\alpha)$ $\mathcal{N}(\mu_\beta; \sigma_\beta)$ $\mathcal{N}(\mu_\gamma; \sigma_\gamma)$

(α, β, γ)

sample angles

$U_{\mathrm{ctrl}}(\alpha, \beta, \gamma)$



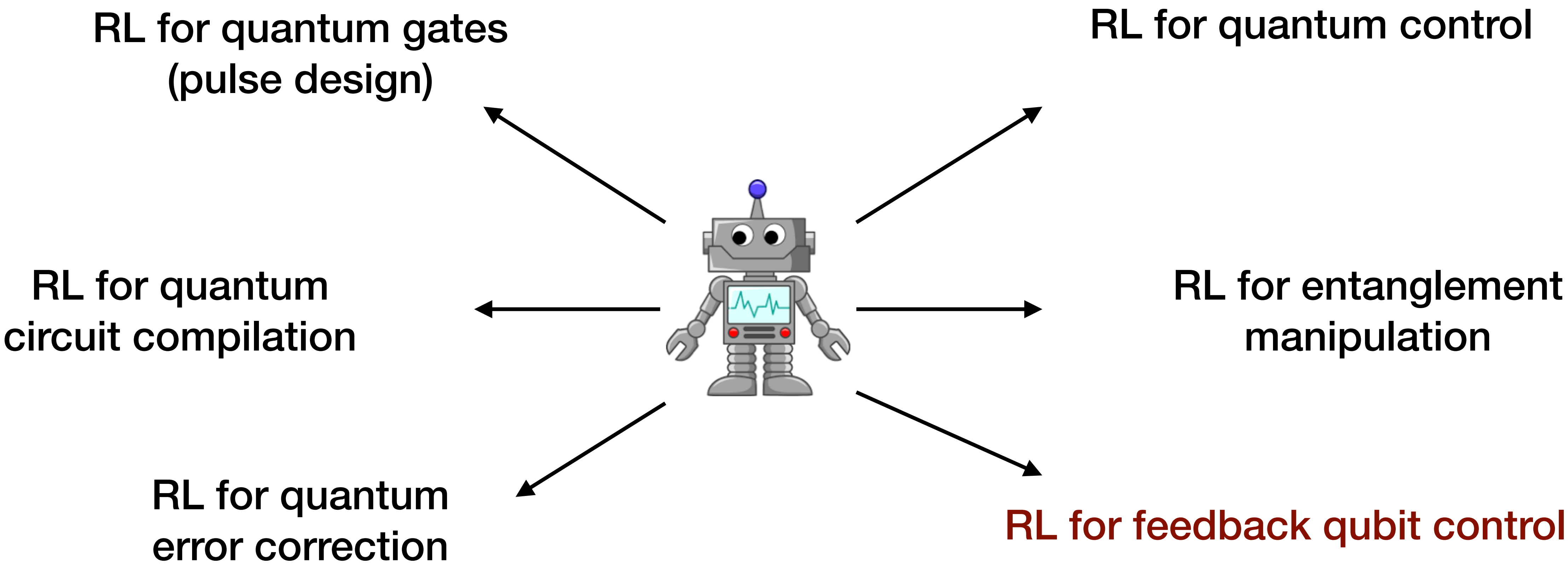


Hands-on policy gradient

Check out Jupyter notebook for how this works in practice!

https://github.com/mgbukov/RL_quantum

RL in Quantum Physics: an Overview



RL for quantum gates (pulse design)

RL for quantum control

RL for quantum circuit compilation

RL for entanglement manipulation

RL for quantum error correction

RL for feedback qubit control

Check out Jupyter notebook for how this works in practice!

https://github.com/mgbukov/RL_quantum

Thanks for your attention!

Useful Literature

M. Nielsen, *Neural Networks and Deep Learning* (online book)

Sutton and Barto, *Reinforcement Learning: an Introduction*, MIT press

S. Levine, *You Tube*, UC Berkeley (videos of lecture course)

M. Bukov, *lecture course,* Sofia University

http://quantum-dynamics.phys.uni-sofia.bg/teaching/WiSe_2020_RL_class/

Check out Jupyter notebook for how this works in practice!

https://github.com/mgbukov/RL_quantum