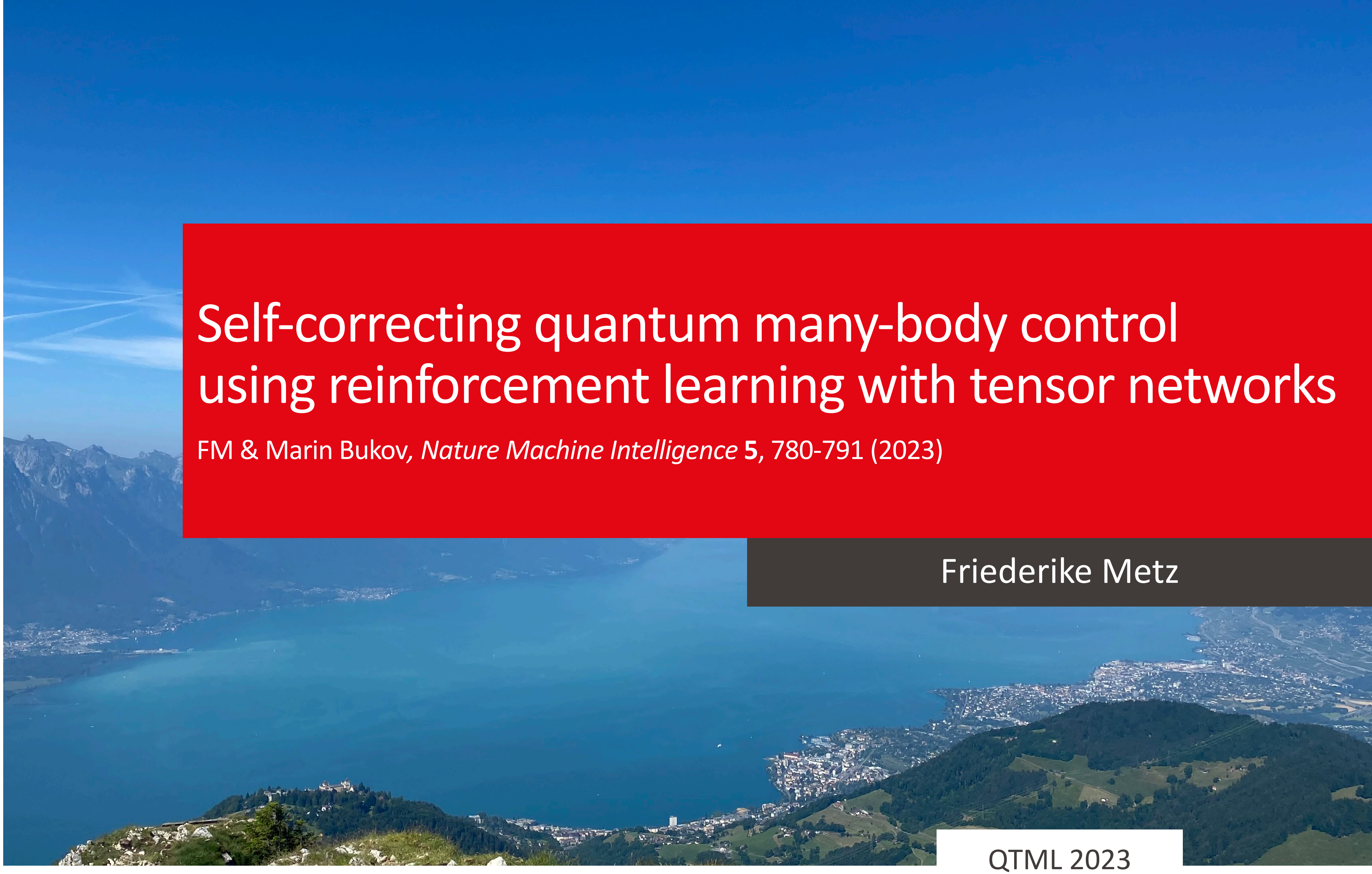


Self-correcting quantum many-body control using reinforcement learning with tensor networks

FM & Marin Bukov, *Nature Machine Intelligence* **5**, 780-791 (2023)

Friederike Metz

EPFL

OIST

mpipks

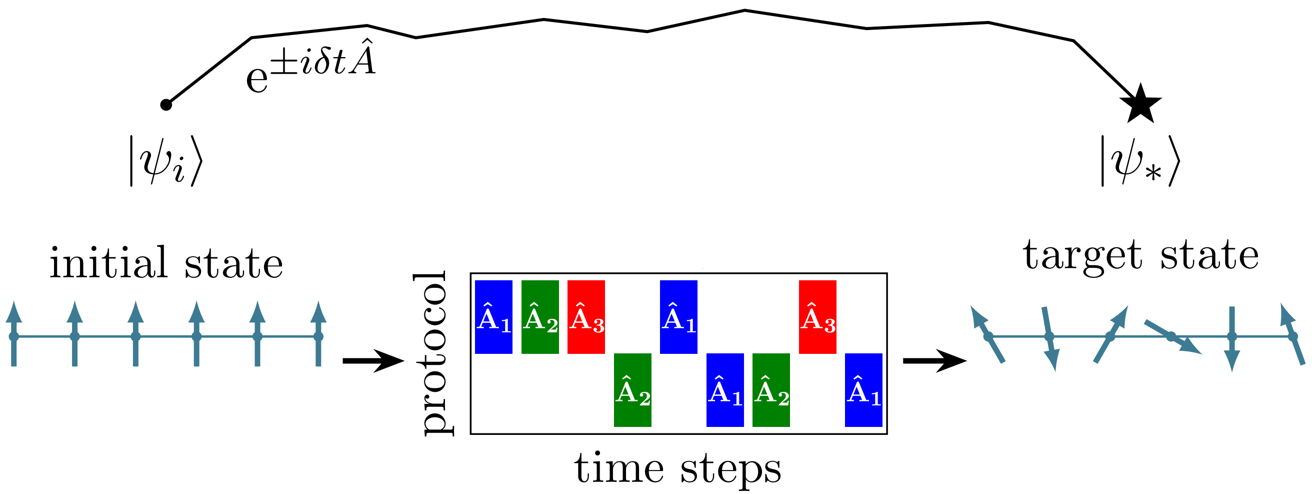
SOFIA UNIVERSITY
ST. KLIMENT OHRIDSKI

QTML 2023

Quantum many-body control

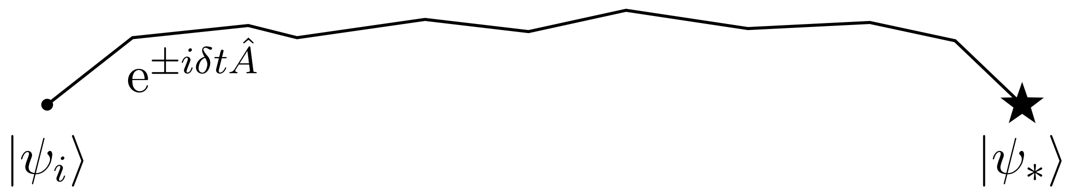
Essential for most quantum technologies (computing, simulation, metrology)

e.g. state preparation:

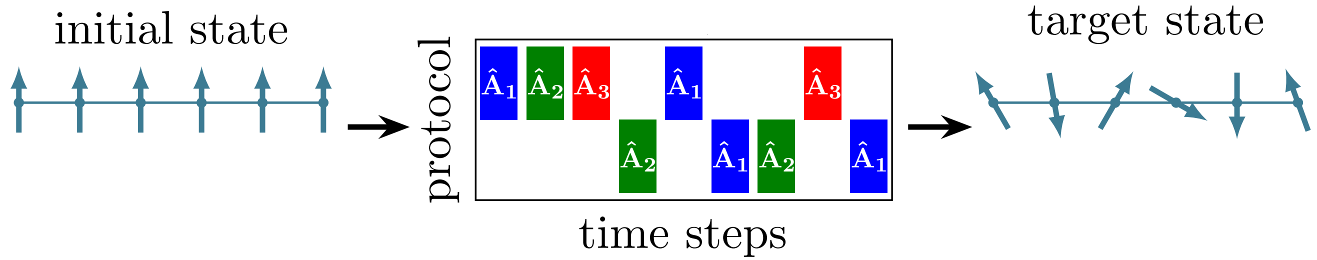


Quantum many-body control

Essential for most quantum technologies (computing, simulation, metrology)



e.g. state preparation:

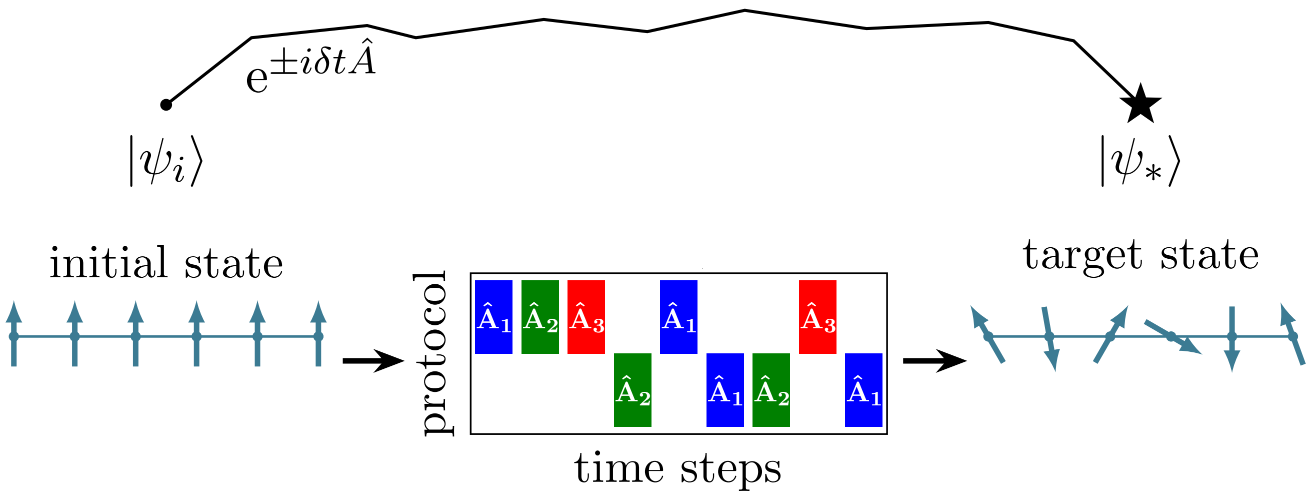


Problem: Hilbert space dimension grows exponentially with system size

Quantum many-body control

Essential for most quantum technologies (computing, simulation, metrology)

e.g. state preparation:



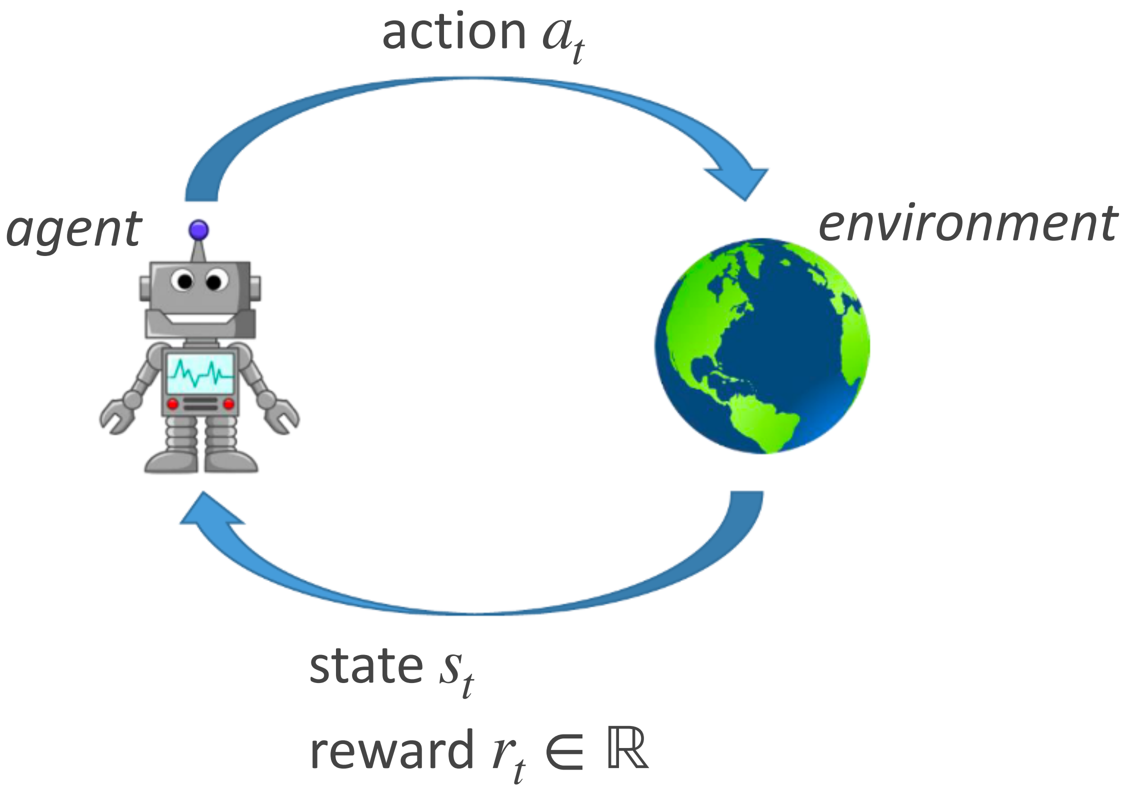
Problem: Hilbert space dimension grows exponentially with system size

→ Our control framework: Deep reinforcement learning with matrix product states

Representing the quantum many-body state

Trainable machine learning architecture for the RL agent

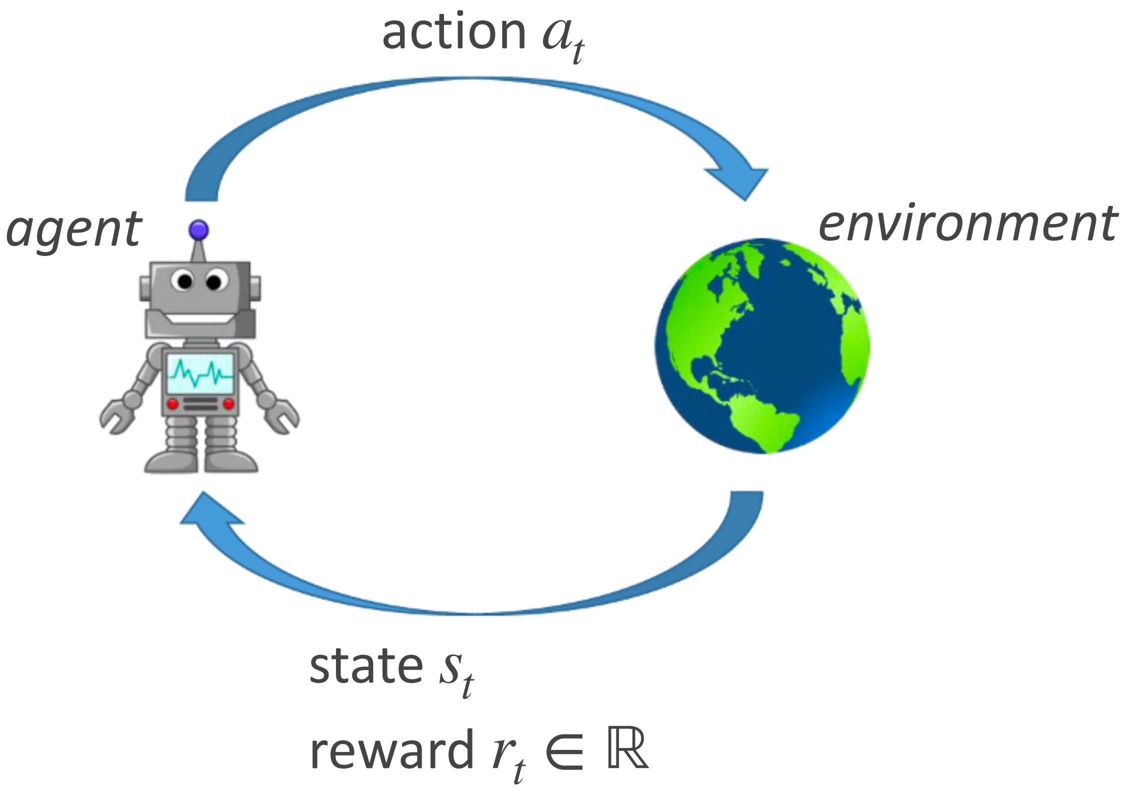
Reinforcement learning (RL)



Goal: Maximize expected cumulative reward by acting optimally in each state.

R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction. MIT press

Q-learning

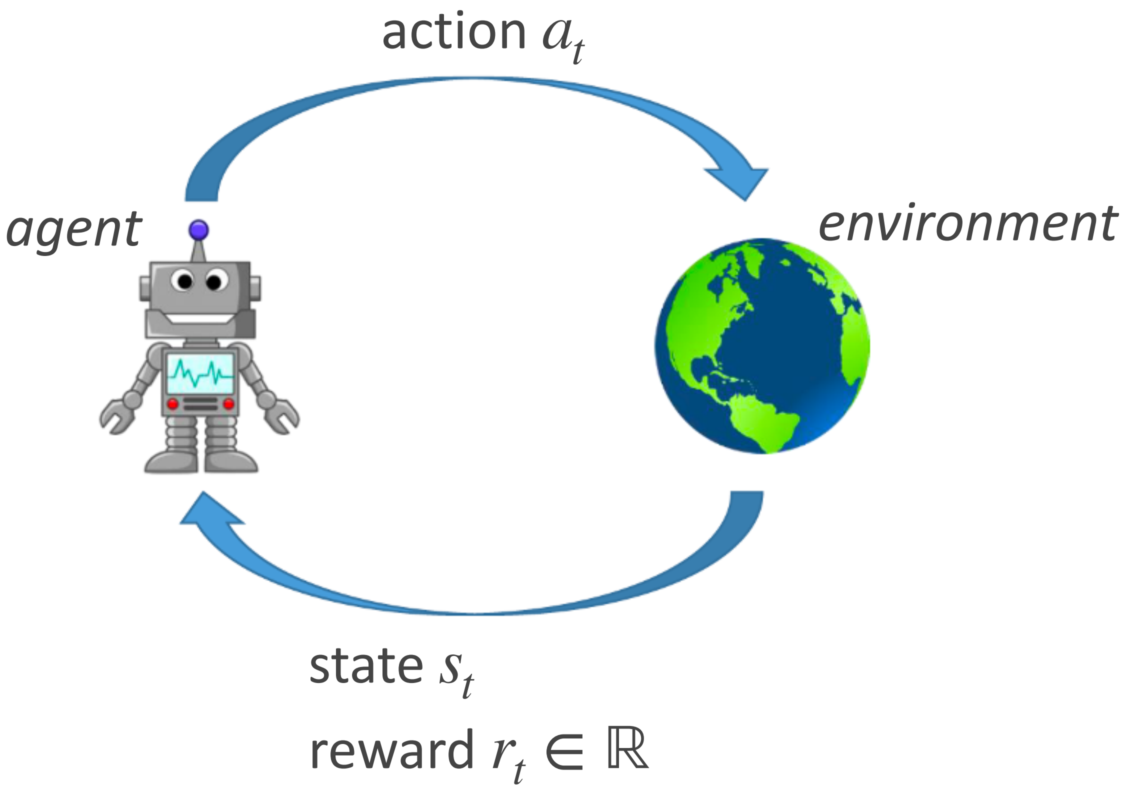


Learn optimal Q values $Q^*(s, a)$:

Maximum expected sum of future rewards if you start in state s and take action a

Mnih et al., Nature 518, 529–533 (2015)
R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction. MIT press

Q-learning



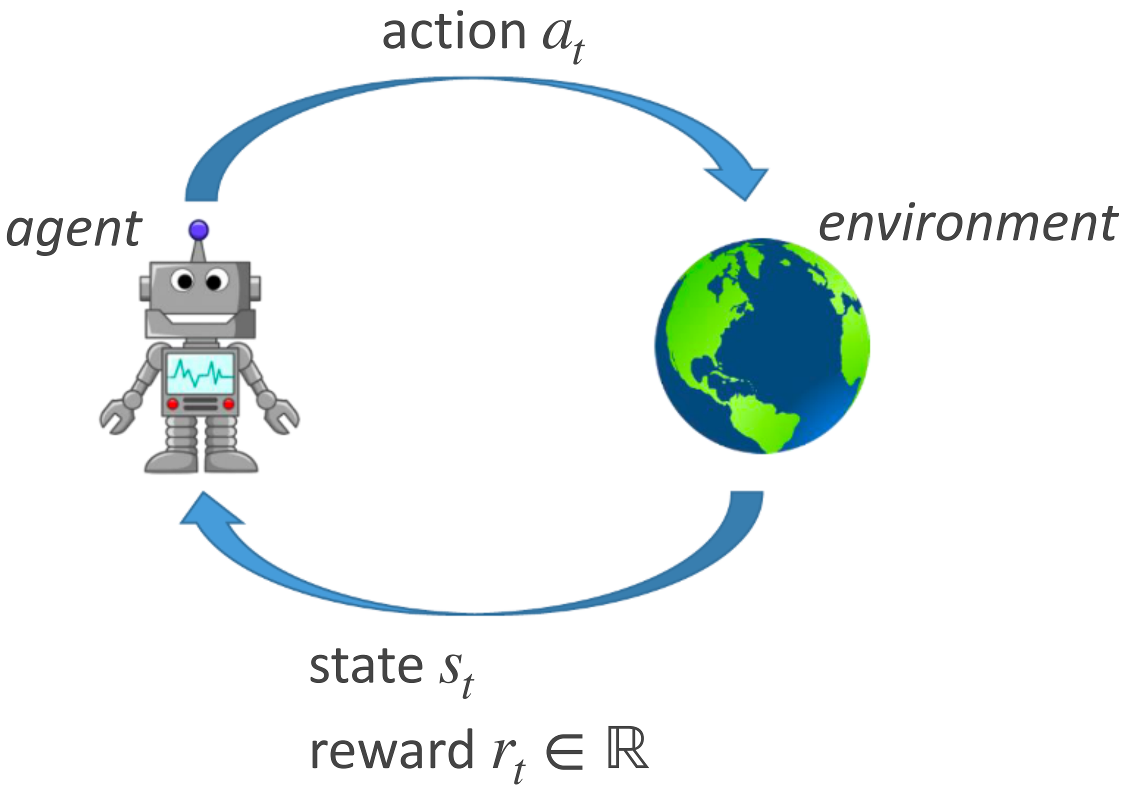
Learn optimal Q values $Q^*(s,a)$:

Maximum expected sum of future rewards if you start in state s and take action a

$\rightarrow$ optimal action: $a_t = \arg\max_a Q^*(s_t, a)$

Mnih et al., Nature 518, 529–533 (2015)
R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction. MIT press

Q-learning

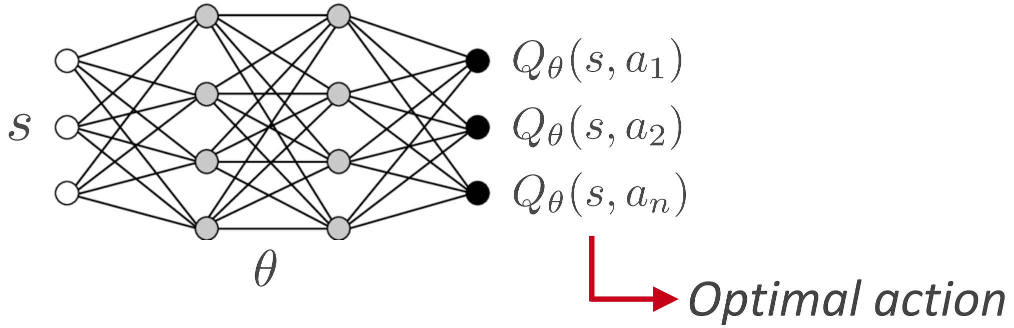


Learn optimal Q values $Q^*(s, a)$:

Maximum expected sum of future rewards if you start in state s and take action a

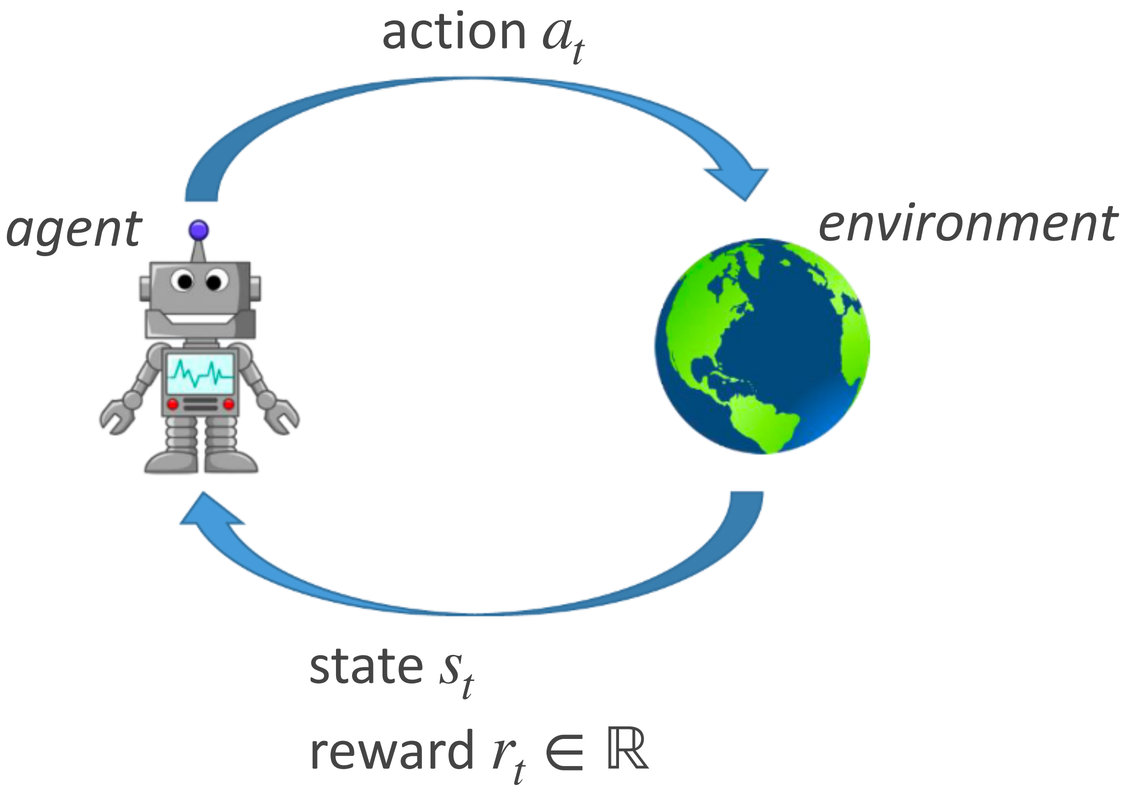
→ optimal action: $a_t = \arg\max_a Q^*(s_t, a)$

DQN (Deep Q-Network):



Mnih et al., Nature 518, 529–533 (2015)
R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction. MIT press

Q-learning

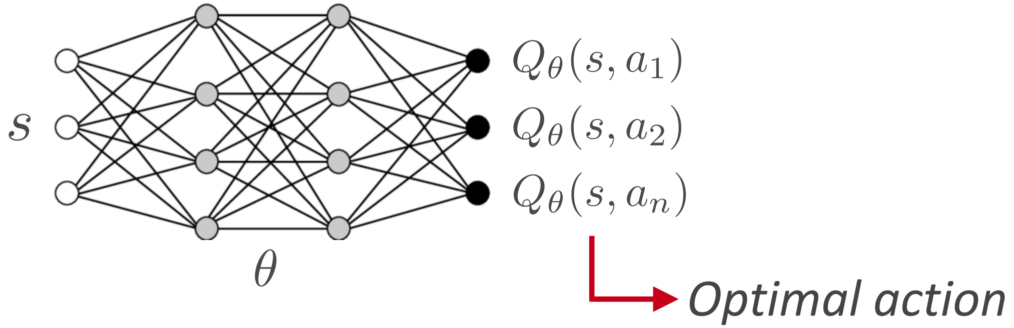


Learn optimal Q values $Q^*(s,a)$:

Maximum expected sum of future rewards if you start in state s and take action a

→ optimal action: $a_t = \arg\max_a Q^*(s_t, a)$

DQN (Deep Q-Network):

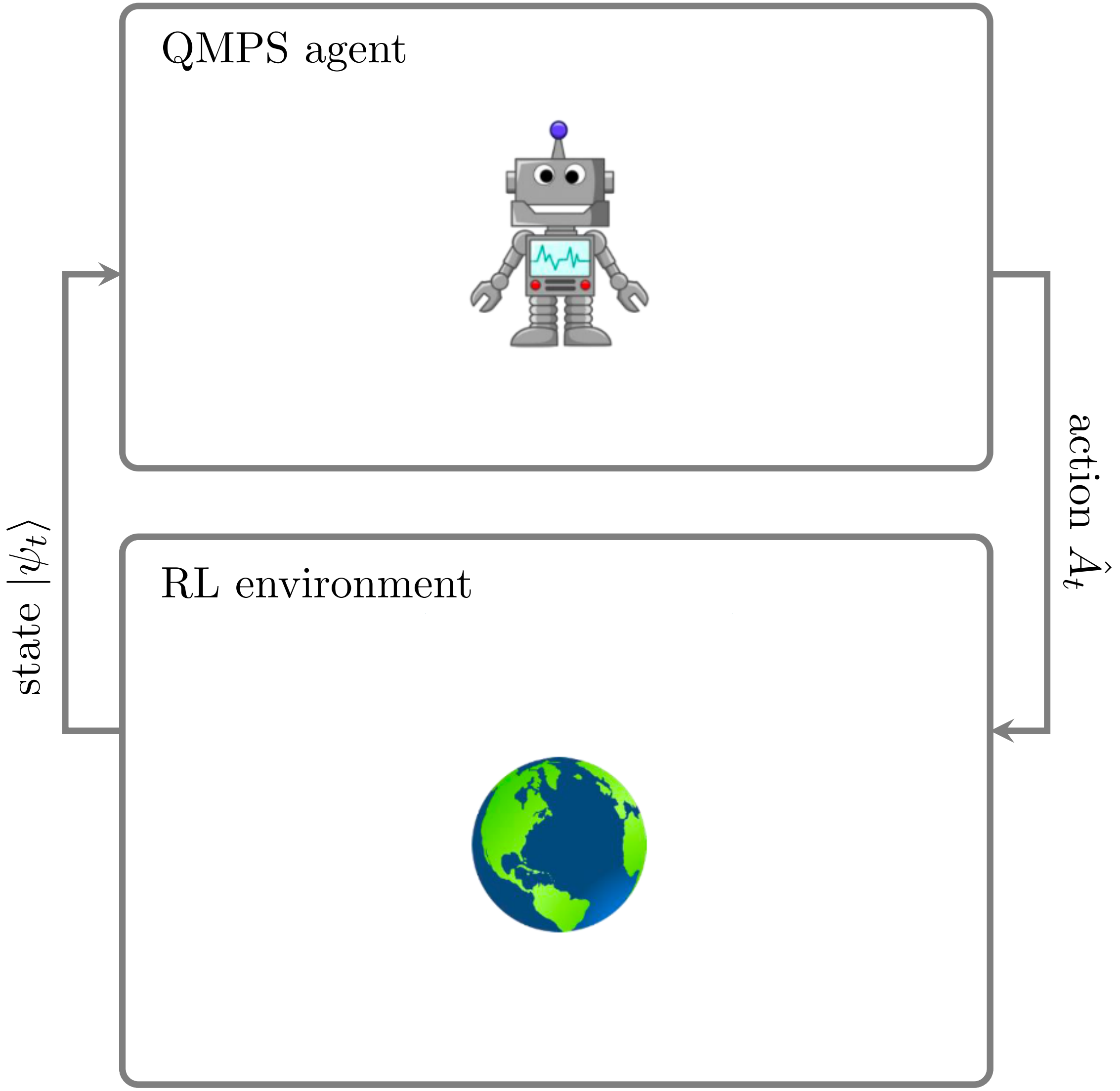


Optimize by minimizing:

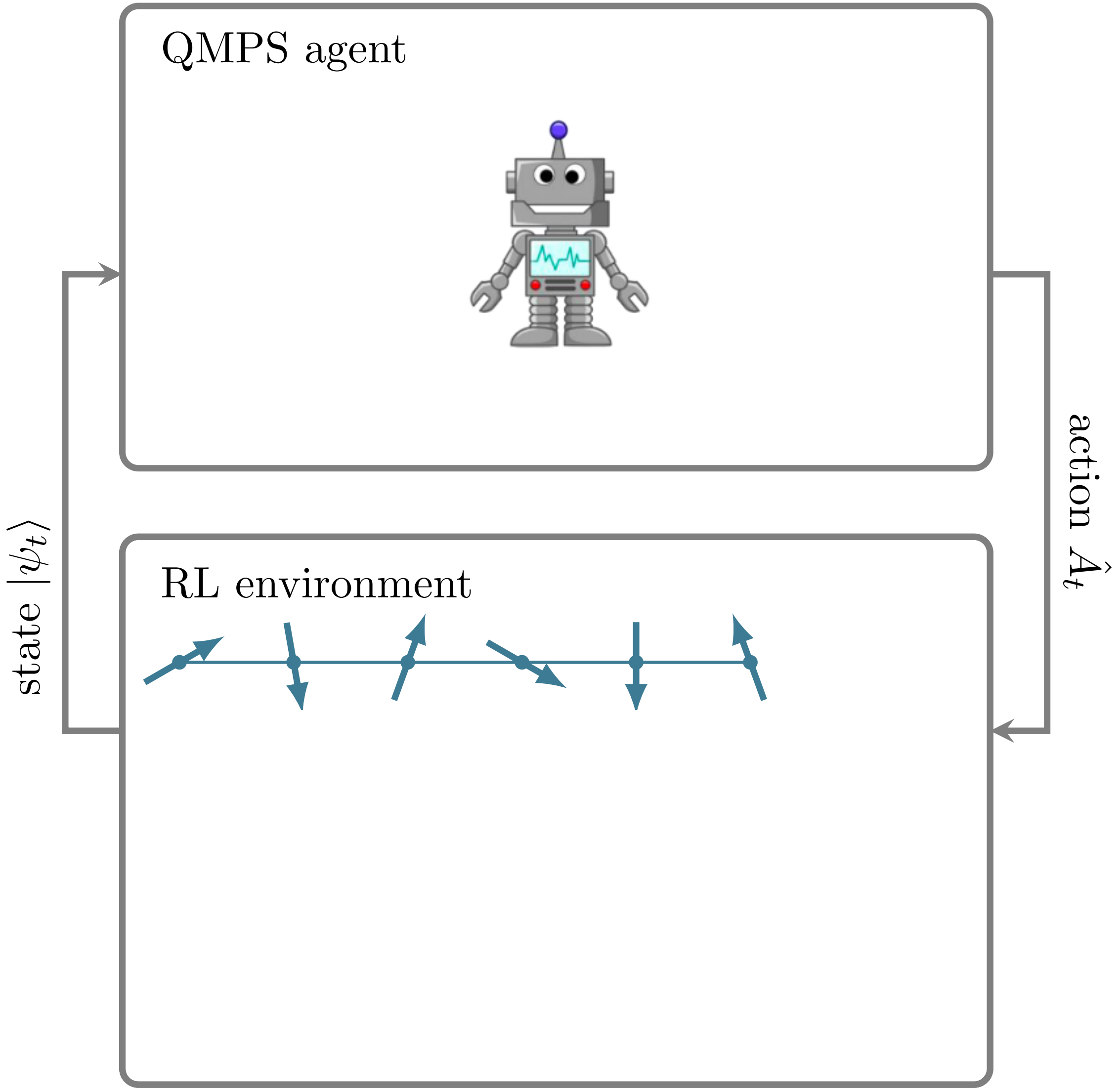
$$L(\theta) = \mathbb{E}\left[\left(Q_\theta(s,a) - \left[r(s,a) + \max_{a'} Q_{\bar{\theta}}(s', a')\right]\right)^2\right]$$

Mnih et al., Nature 518, 529–533 (2015)
R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction. MIT press

Matrix product state ansatz for Q-learning (QMPS)

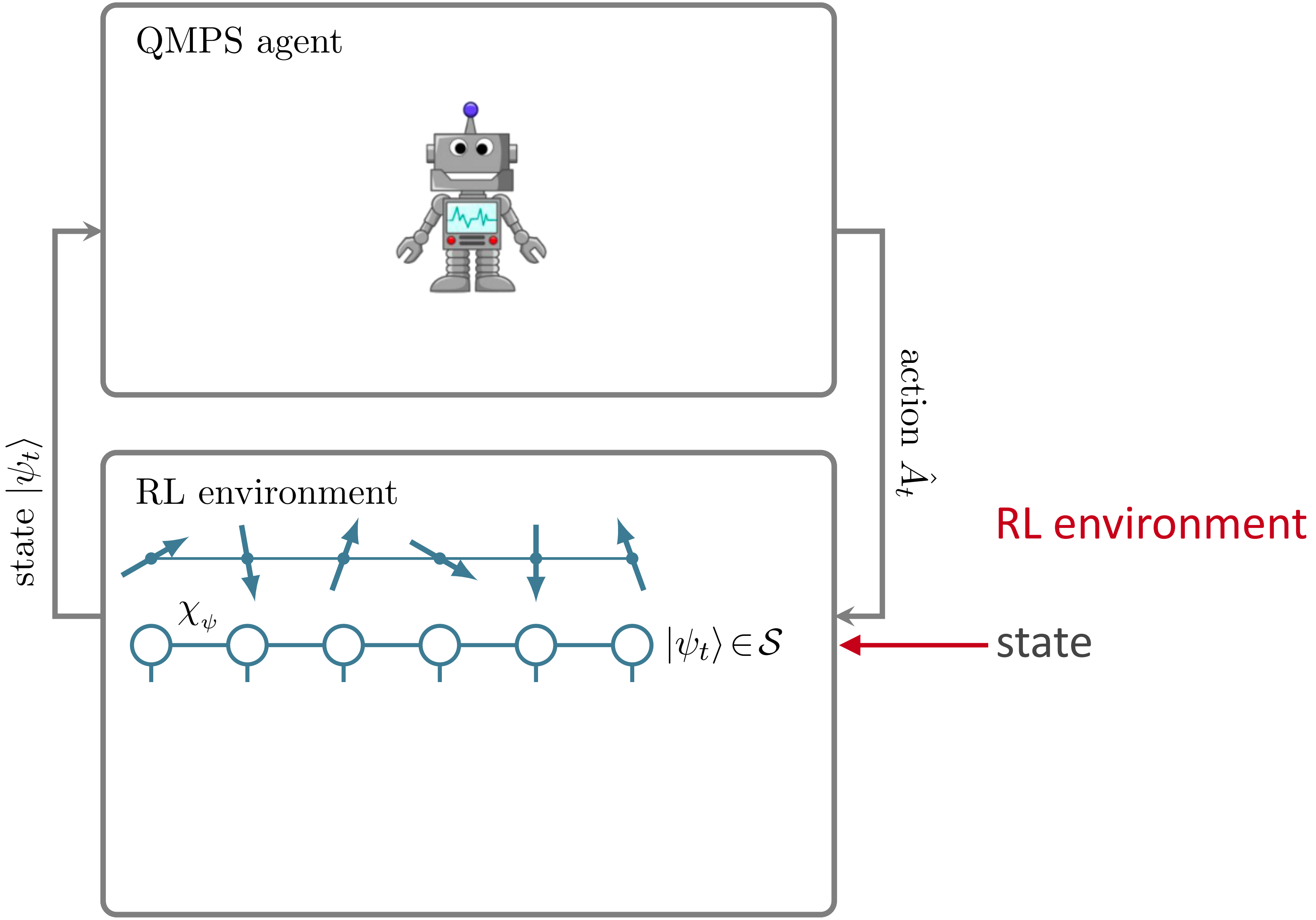


Matrix product state ansatz for Q-learning (QMPS)

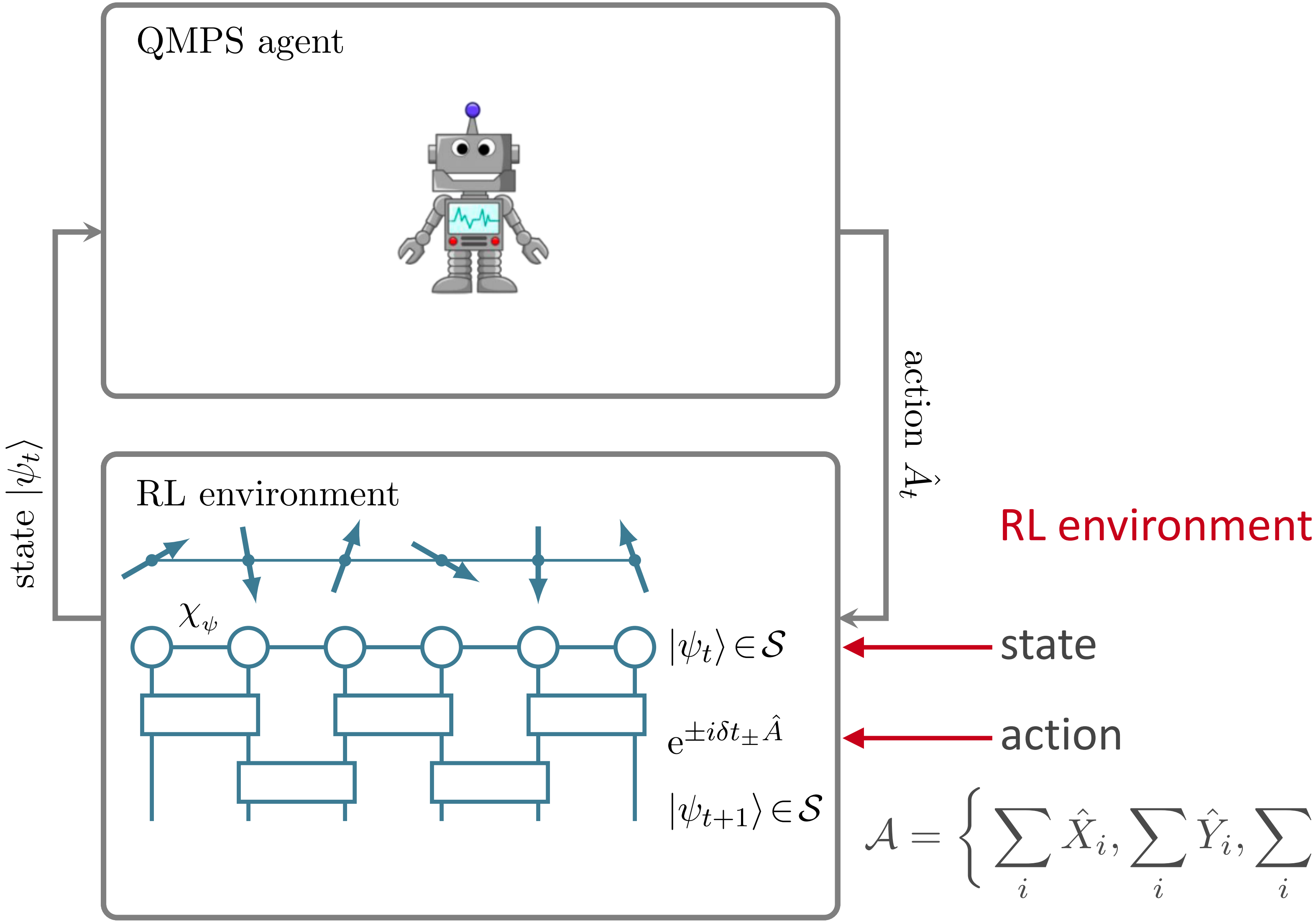


RL environment

Matrix product state ansatz for Q-learning (QMPS)

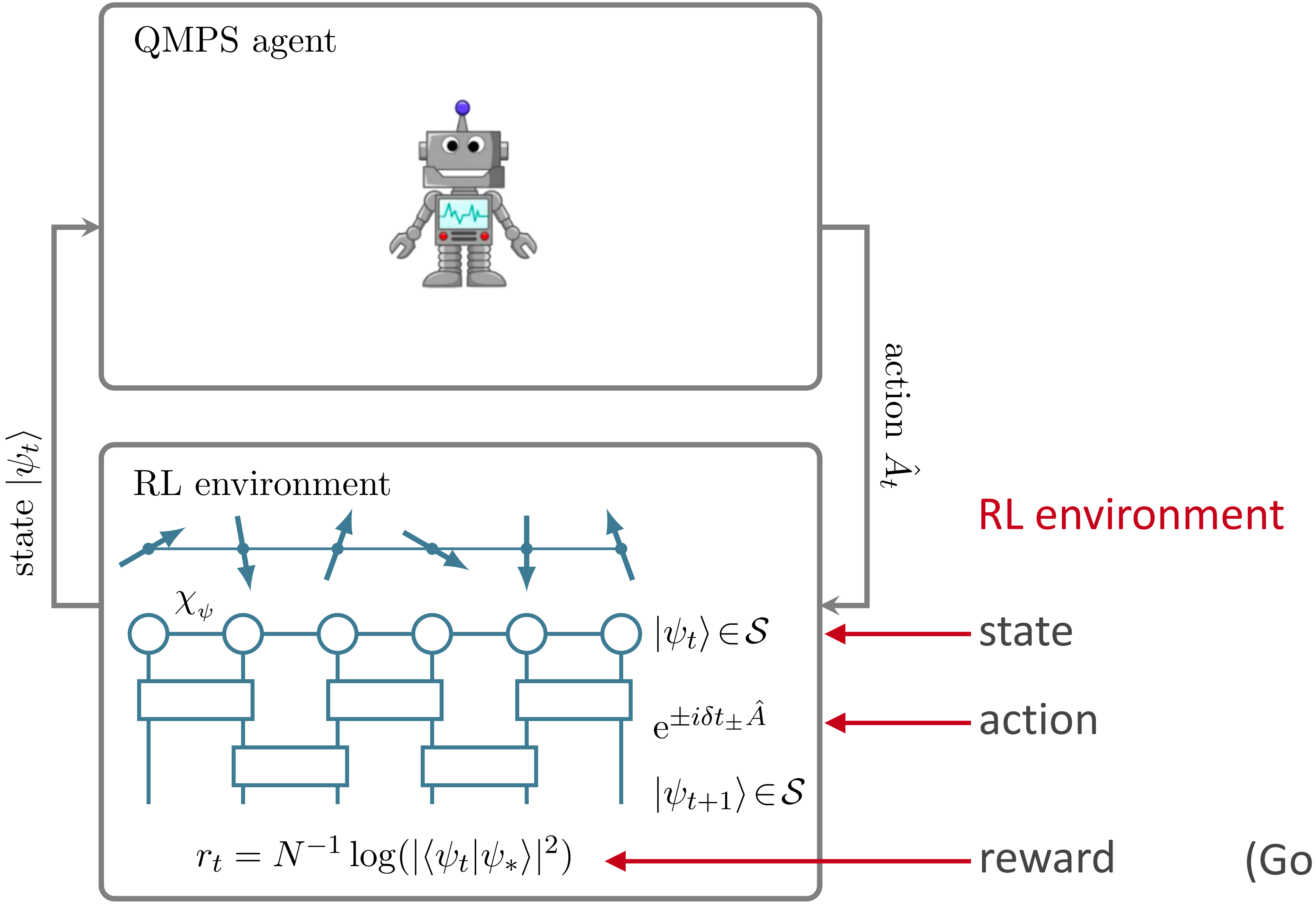


Matrix product state ansatz for Q-learning (QMPS)



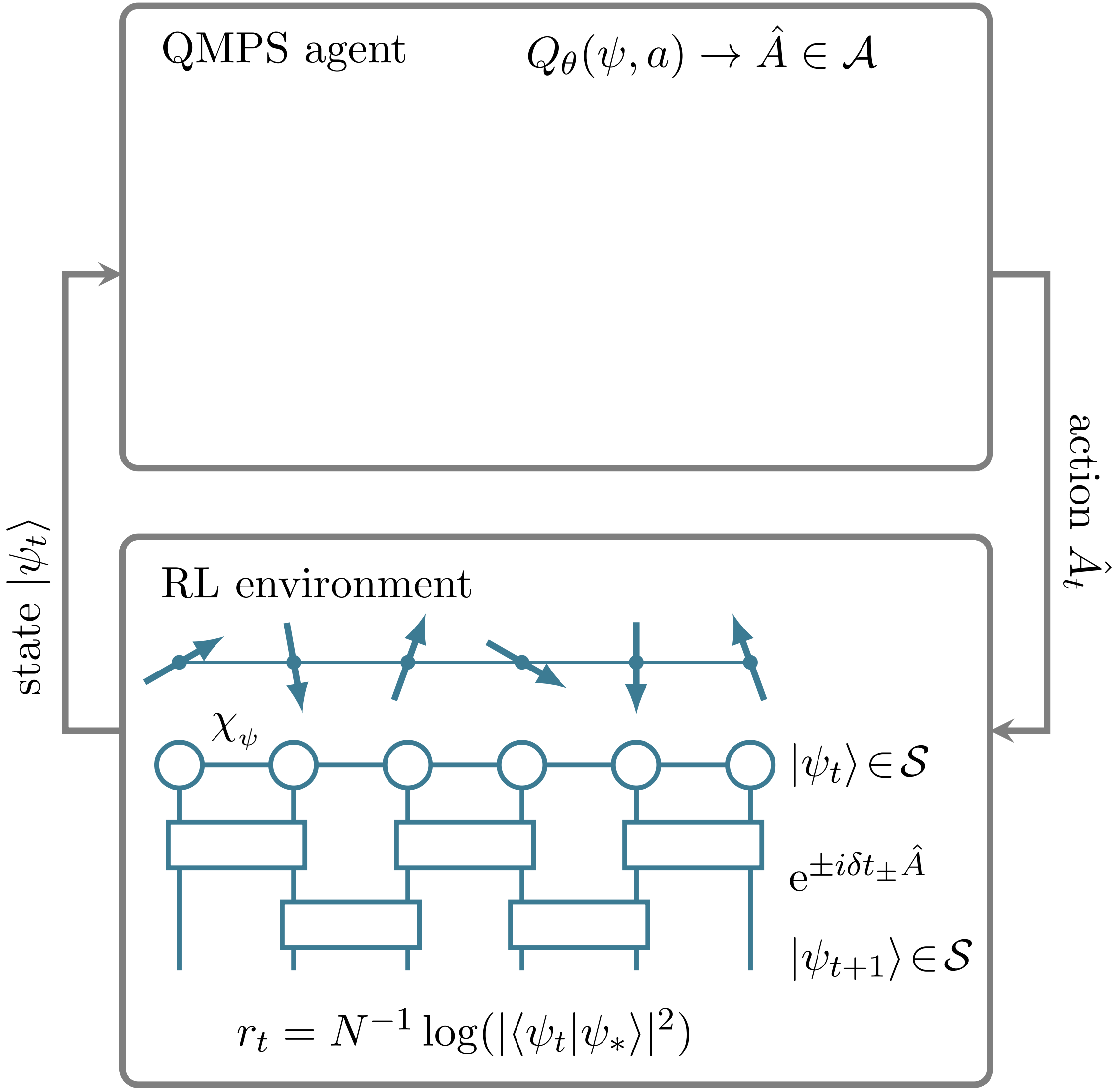
$$\mathcal{A} = \left\{ \sum_i \hat{X}_i, \sum_i \hat{Y}_i, \sum_i \hat{Z}_i, \sum_i \hat{X}_i\hat{X}_{i+1}, \sum_i \hat{Y}_i\hat{Y}_{i+1}, \sum_i \hat{Z}_i\hat{Z}_{i+1} \right\}$$

Matrix product state ansatz for Q-learning (QMPS)



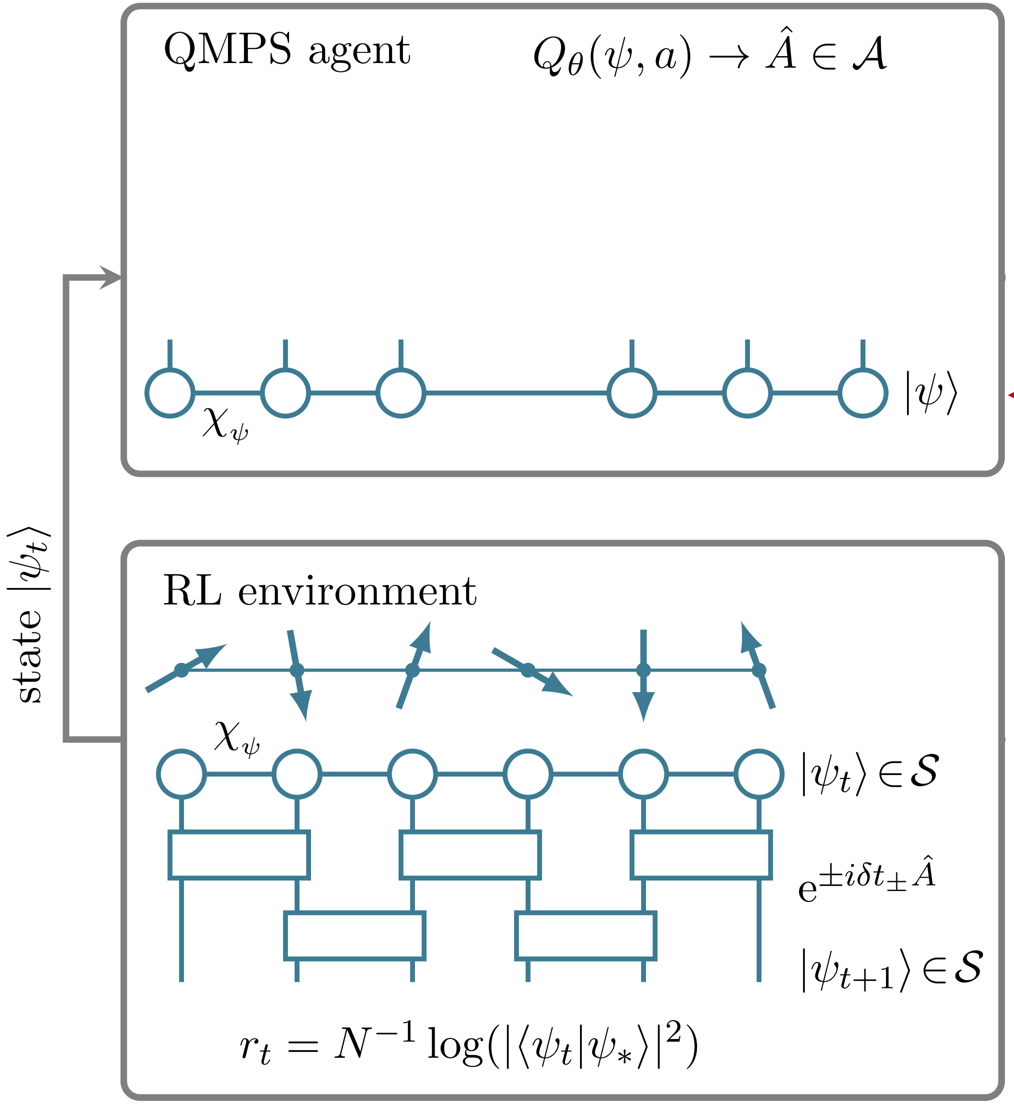
(Goal: Prepare target state $|\psi_*\rangle$)

Matrix product state ansatz for Q-learning (QMPS)



QMPS agent

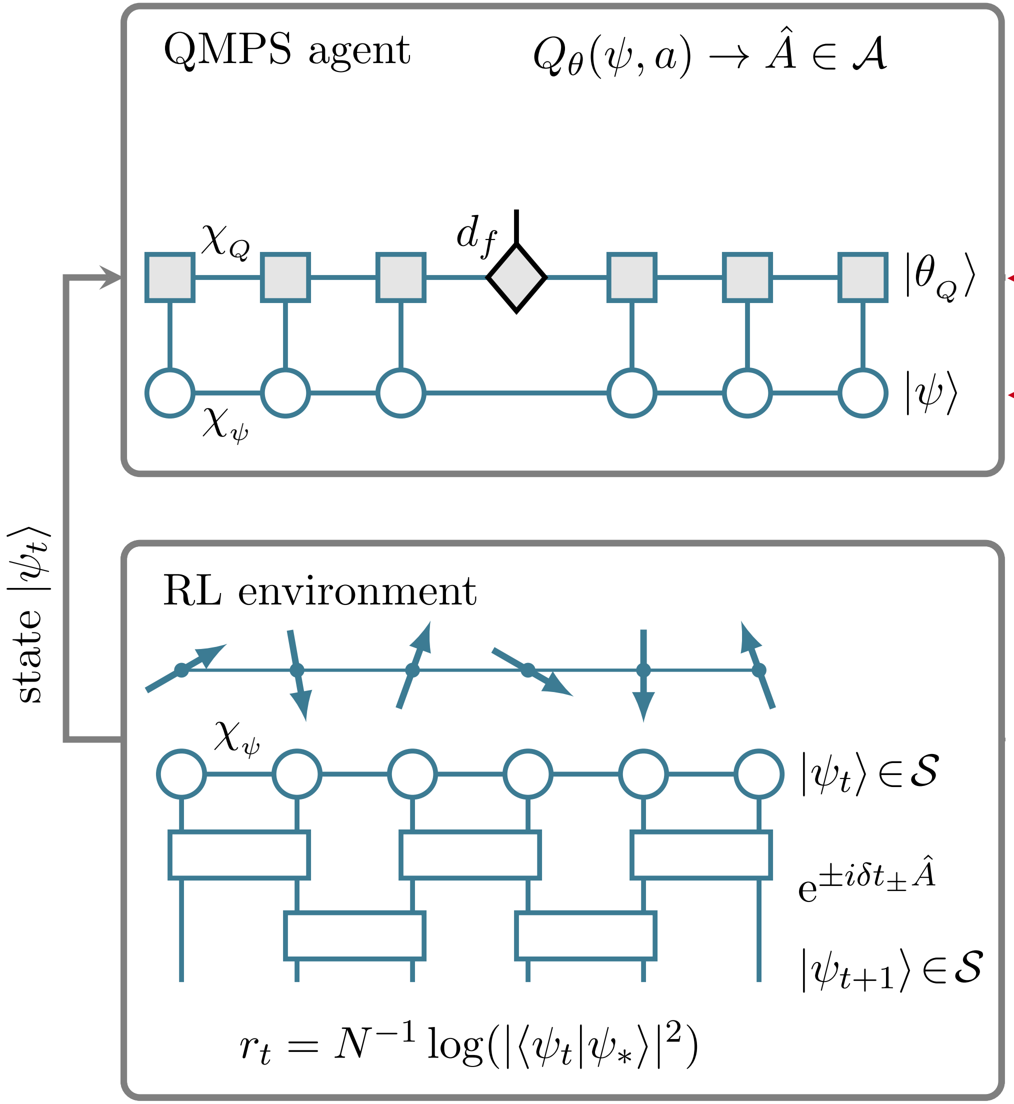
Matrix product state ansatz for Q-learning (QMPS)



QMPS agent

Quantum state input (MPS)

Matrix product state ansatz for Q-learning (QMPS)

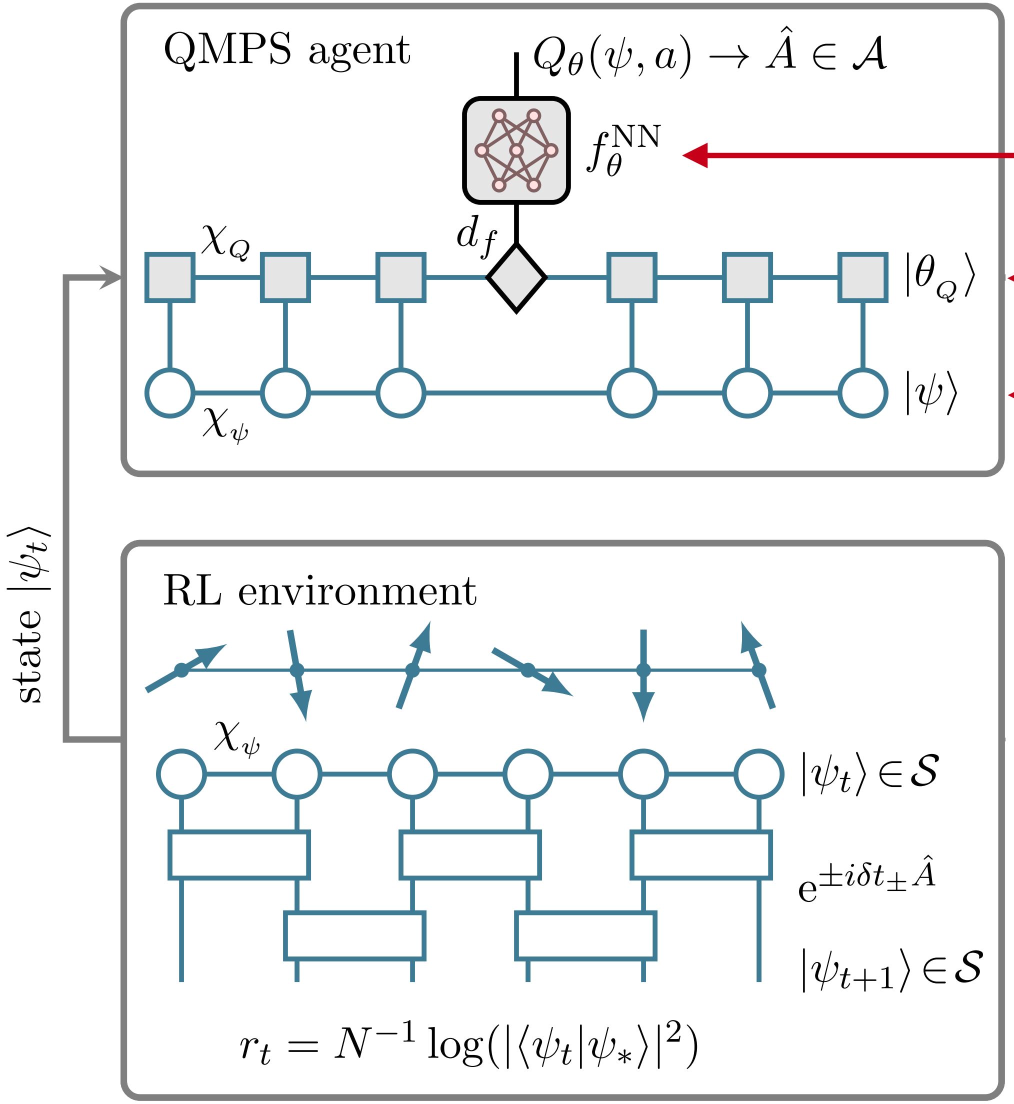


QMPS agent

Trainable parameters (MPS)

Quantum state input (MPS)

Matrix product state ansatz for Q-learning (QMPS)



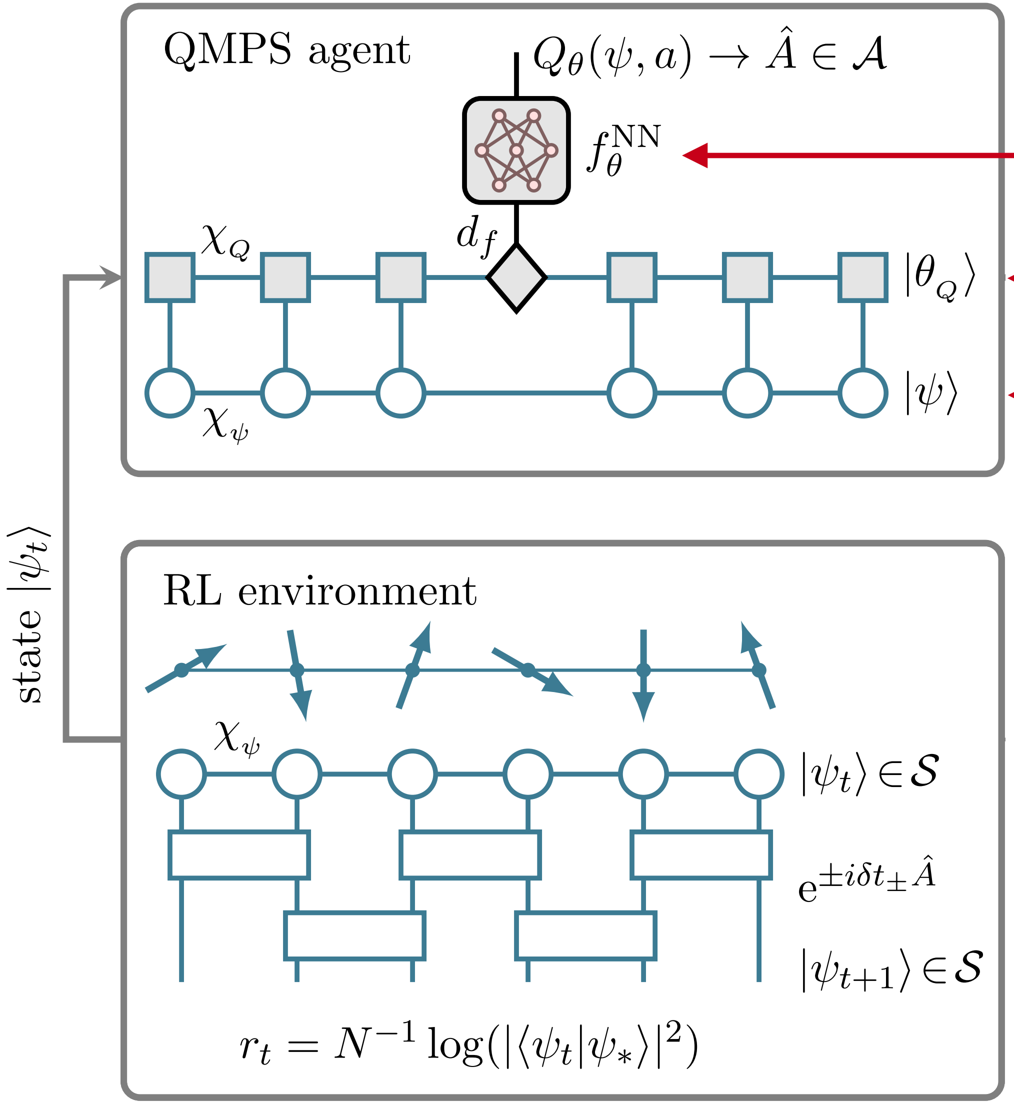
QMPS agent

Neural network (NN)

Trainable parameters (MPS)

Quantum state input (MPS)

Matrix product state ansatz for Q-learning (QMPS)



QMPS agent

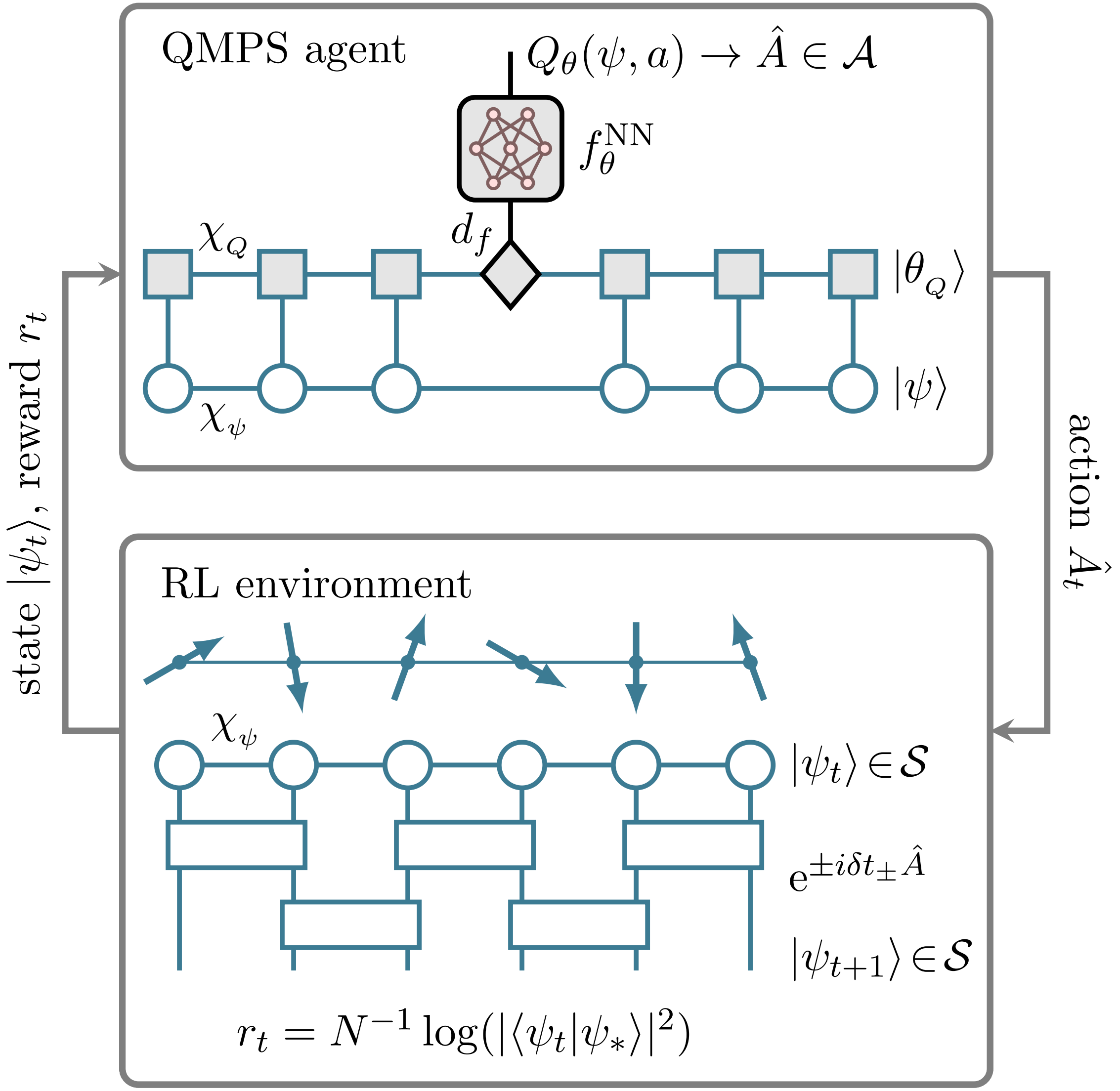
Neural network (NN)

Trainable parameters (MPS)

Quantum state input (MPS)

$\rightarrow$ Resources scale only linearly in system size N

Matrix product state ansatz for Q-learning (QMPS)



QMPS framework

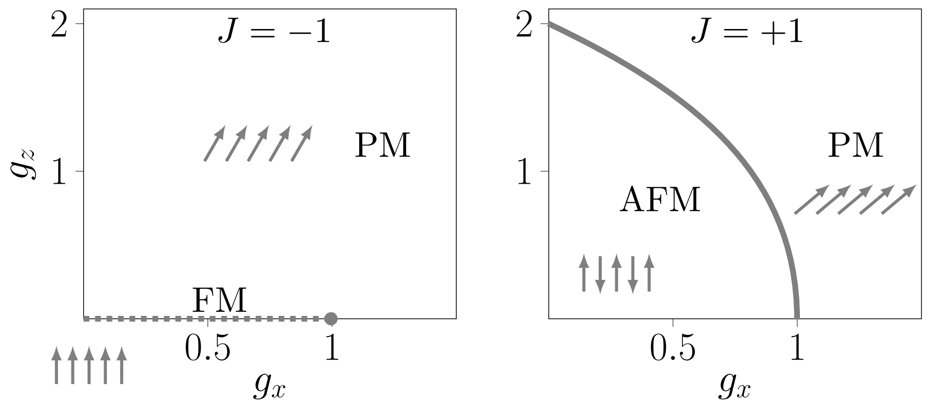
Start each episode from a (random) initial state

Terminate episode when fidelity threshold is reached, e.g. $F = 0.99$

State preparation

Mixed-field Ising:

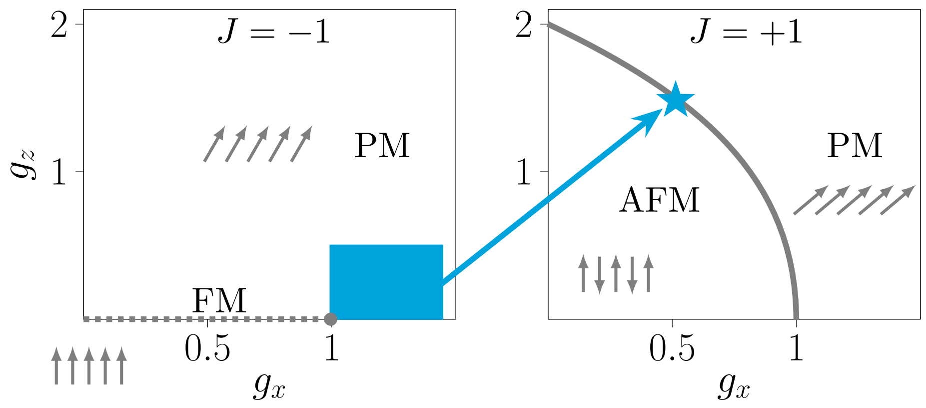
$$\hat{H} = J \sum_{i=1}^{N-1} \hat{Z}_i \hat{Z}_{i+1} - g_x \sum_{i=1}^{N} \hat{X}_i - g_z \sum_{i=1}^{N} \hat{Z}_i$$



State preparation

Mixed-field Ising:

$$\hat{H} = J \sum_{i=1}^{N-1} \hat{Z}_i \hat{Z}_{i+1} - g_x \sum_{i=1}^{N} \hat{X}_i - g_z \sum_{i=1}^{N} \hat{Z}_i$$

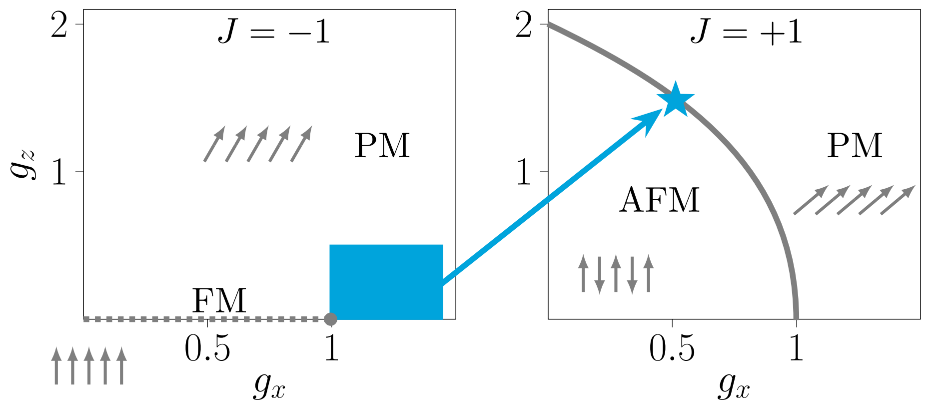


State preparation

Mixed-field Ising:

$$\hat{H} = J \sum_{i=1}^{N-1} \hat{Z}_i \hat{Z}_{i+1} - g_x \sum_{i=1}^{N} \hat{X}_i - g_z \sum_{i=1}^{N} \hat{Z}_i$$

$\rightarrow N = 16$ spins

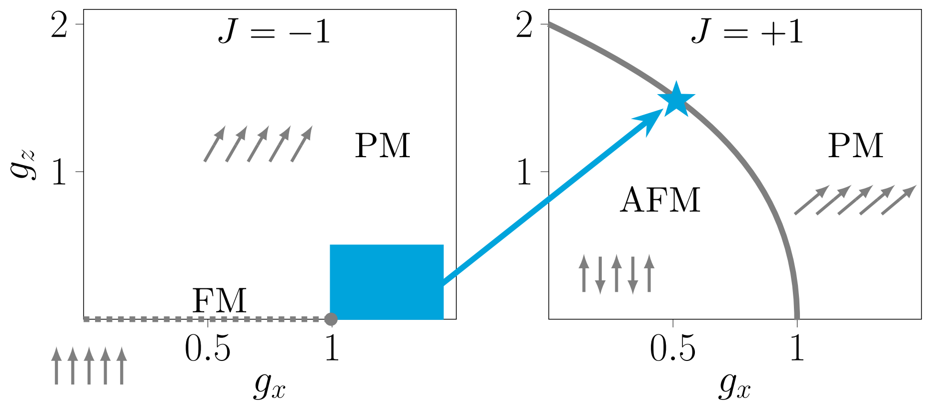


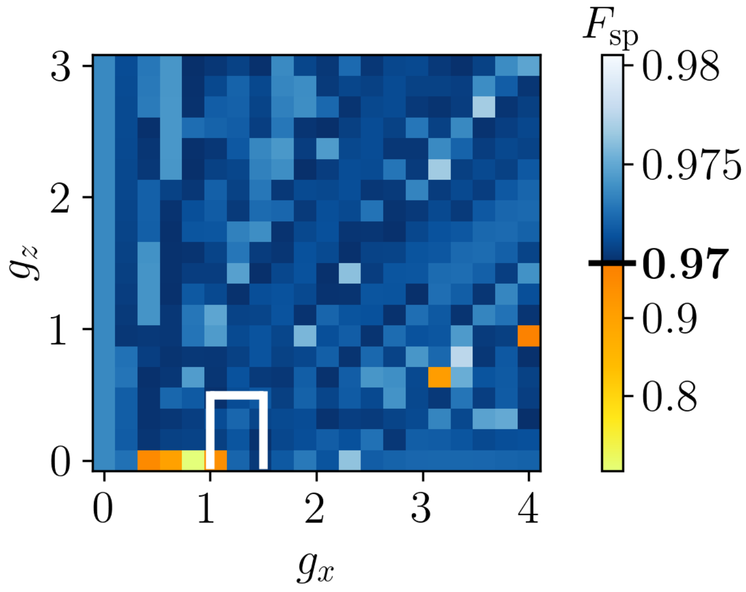
State preparation

Mixed-field Ising:

$$\hat{H} = J\sum_{i=1}^{N-1}\hat{Z}_i\hat{Z}_{i+1} - g_x\sum_{i=1}^{N}\hat{X}_i - g_z\sum_{i=1}^{N}\hat{Z}_i$$

$\rightarrow N = 16$ spins



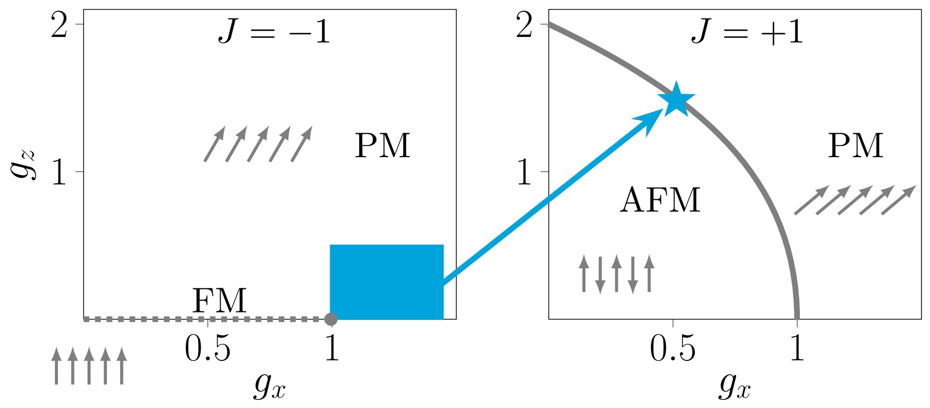


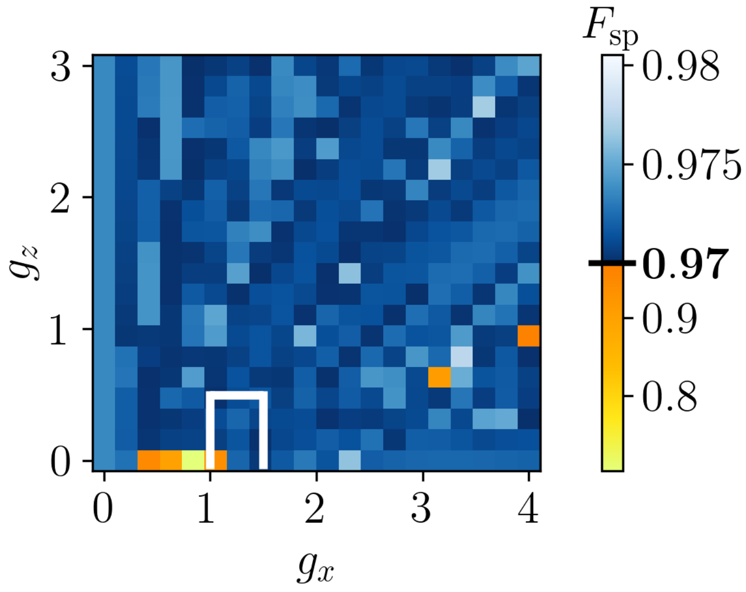
State preparation

Mixed-field Ising:

$$\hat{H} = J\sum_{i=1}^{N-1}\hat{Z}_i\hat{Z}_{i+1} - g_x\sum_{i=1}^{N}\hat{X}_i - g_z\sum_{i=1}^{N}\hat{Z}_i$$

$\rightarrow N = 16$ spins



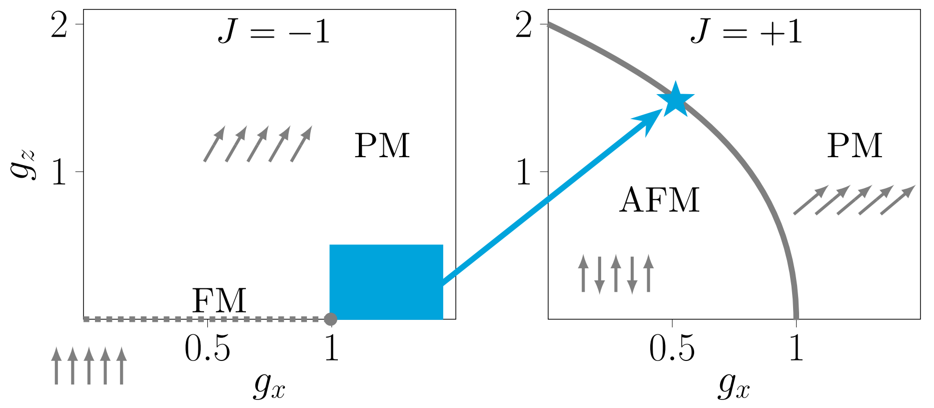


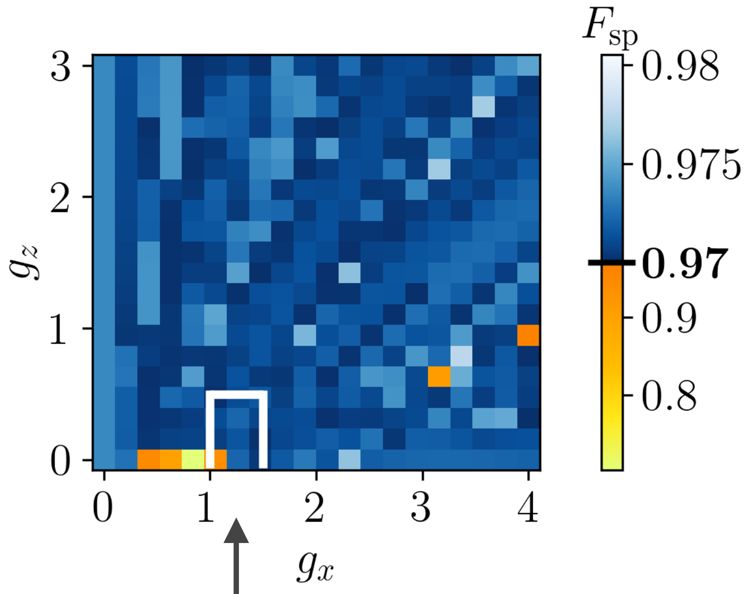
State preparation

Mixed-field Ising:

$$\hat{H} = J \sum_{i=1}^{N-1} \hat{Z}_i \hat{Z}_{i+1} - g_x \sum_{i=1}^{N} \hat{X}_i - g_z \sum_{i=1}^{N} \hat{Z}_i$$

$\rightarrow N = 16$ spins





Fidelity threshold

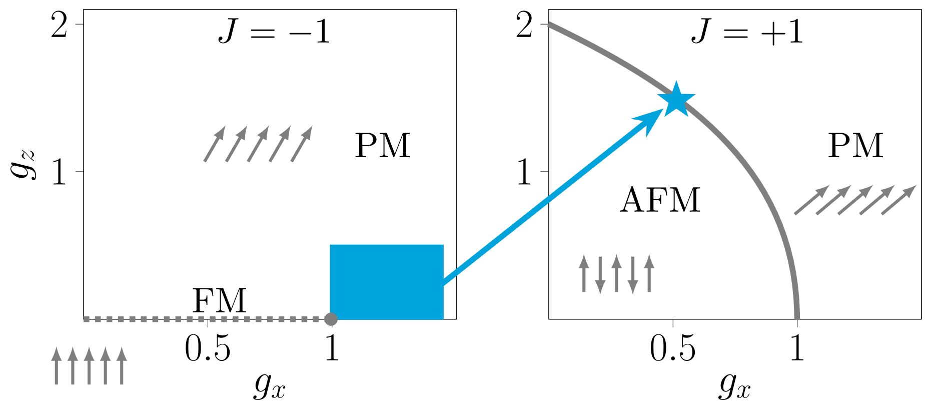
Training

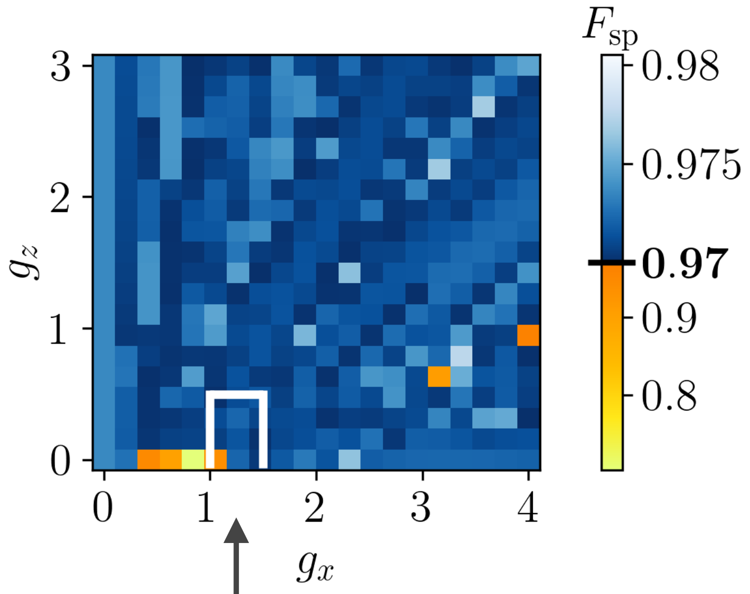
State preparation

Mixed-field Ising:

$$\hat{H} = J \sum_{i=1}^{N-1} \hat{Z}_i \hat{Z}_{i+1} - g_x \sum_{i=1}^{N} \hat{X}_i - g_z \sum_{i=1}^{N} \hat{Z}_i$$

$\rightarrow N = 16$ spins





Fidelity threshold

Training

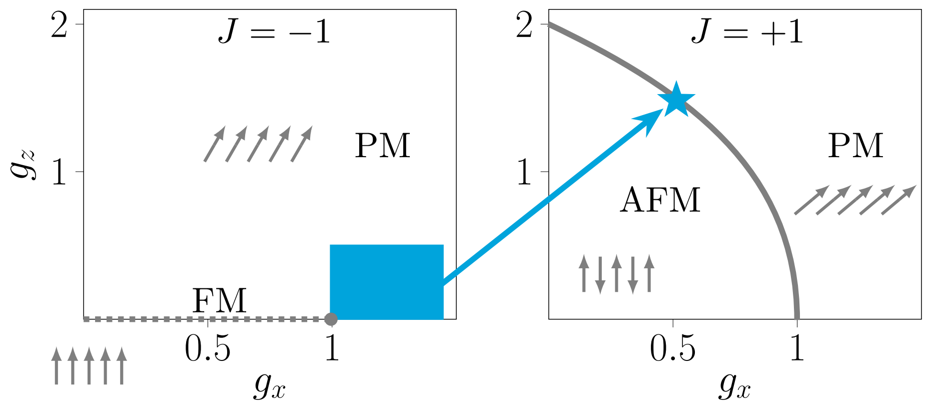
QMPS agent can extrapolate optimal protocols well beyond training region

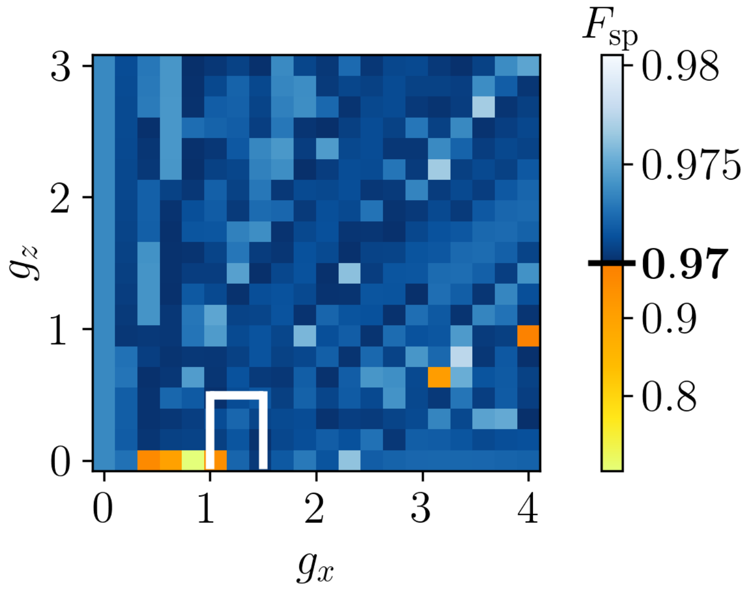
State preparation

Mixed-field Ising:

$$\hat{H} = J \sum_{i=1}^{N-1} \hat{Z}_i \hat{Z}_{i+1} - g_x \sum_{i=1}^{N} \hat{X}_i - g_z \sum_{i=1}^{N} \hat{Z}_i$$

$\rightarrow N = 16$ spins





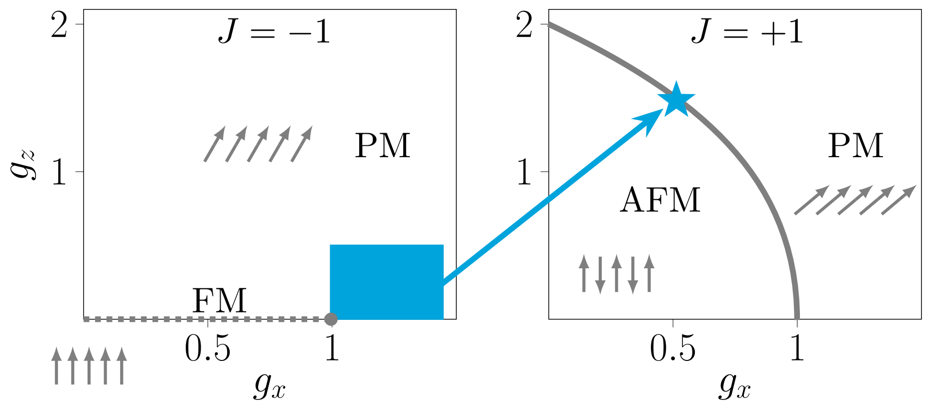
Noise: At each step, white Gaussian noise with std σ is added to step duration δt $\quad (\mathrm{e}^{\pm i \delta t_{\pm} \hat{A}})$

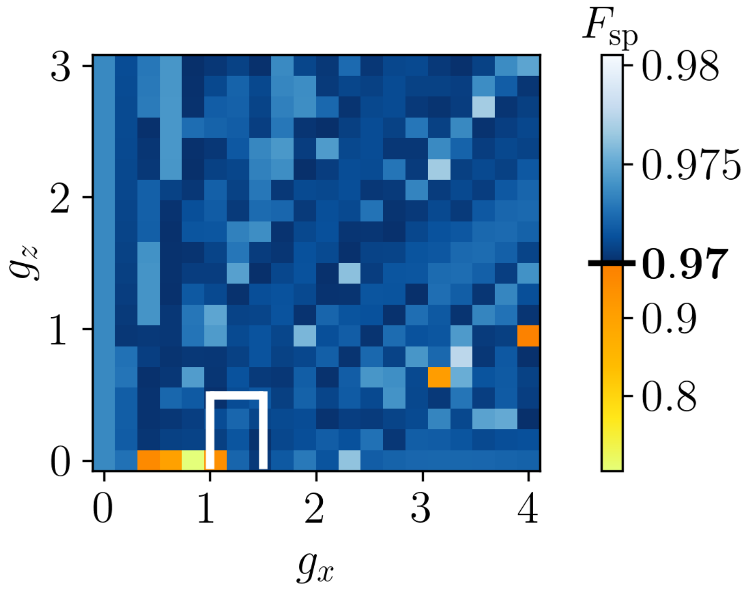
State preparation

Mixed-field Ising:

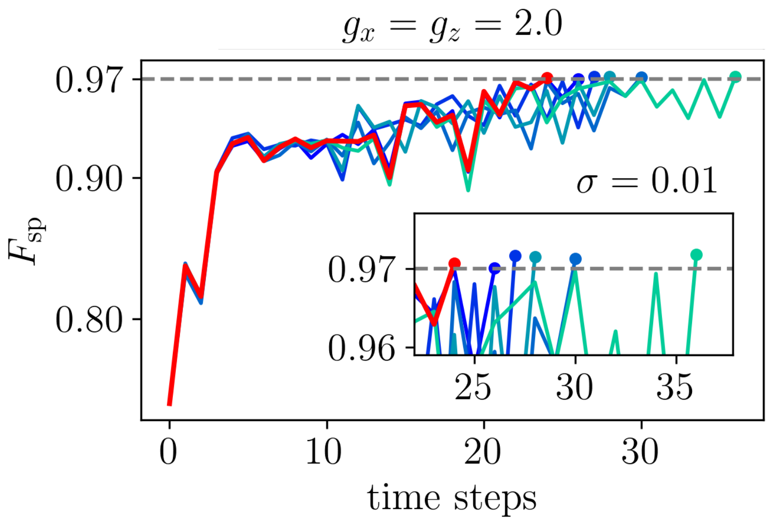
$$\hat{H} = J\sum_{i=1}^{N-1}\hat{Z}_i\hat{Z}_{i+1} - g_x\sum_{i=1}^{N}\hat{X}_i - g_z\sum_{i=1}^{N}\hat{Z}_i$$

$\rightarrow N = 16$ spins





Noise: At each step, white Gaussian noise with std σ is added to step duration δt $(\mathrm{e}^{\pm i\delta t_{\pm}\hat{A}})$

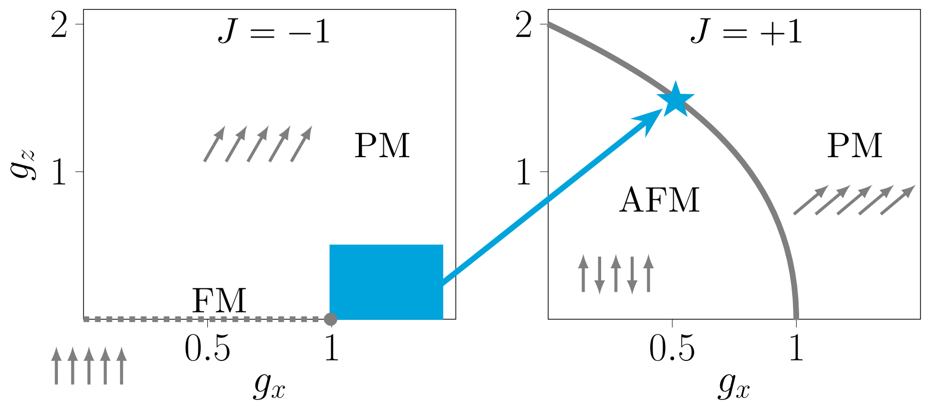


State preparation

Mixed-field Ising:

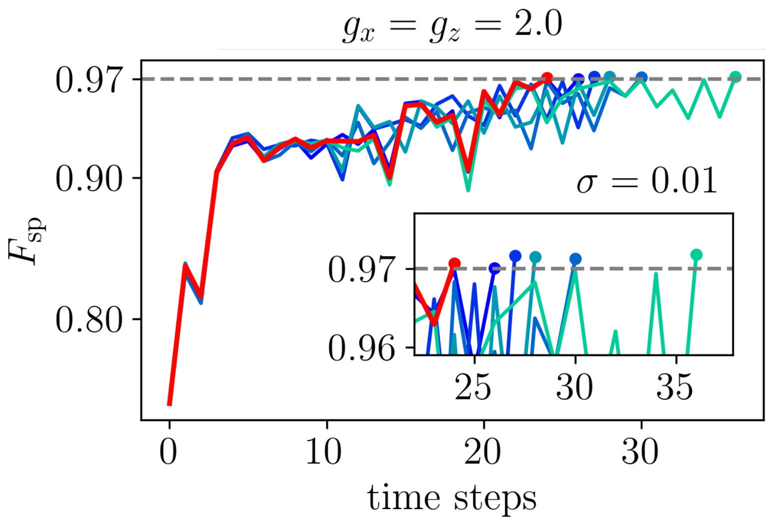
$$\hat{H} = J\sum_{i=1}^{N-1}\hat{Z}_i\hat{Z}_{i+1} - g_x\sum_{i=1}^{N}\hat{X}_i - g_z\sum_{i=1}^{N}\hat{Z}_i$$

$\rightarrow N = 16$ spins



Noise: At each step, white Gaussian noise with std σ is added to step duration δt $(\mathrm{e}^{\pm i\delta t_\pm \hat{A}})$

QMPS agent can self-correct protocols on-the-fly



Outlook

- Universal state preparation / many-body state preparation

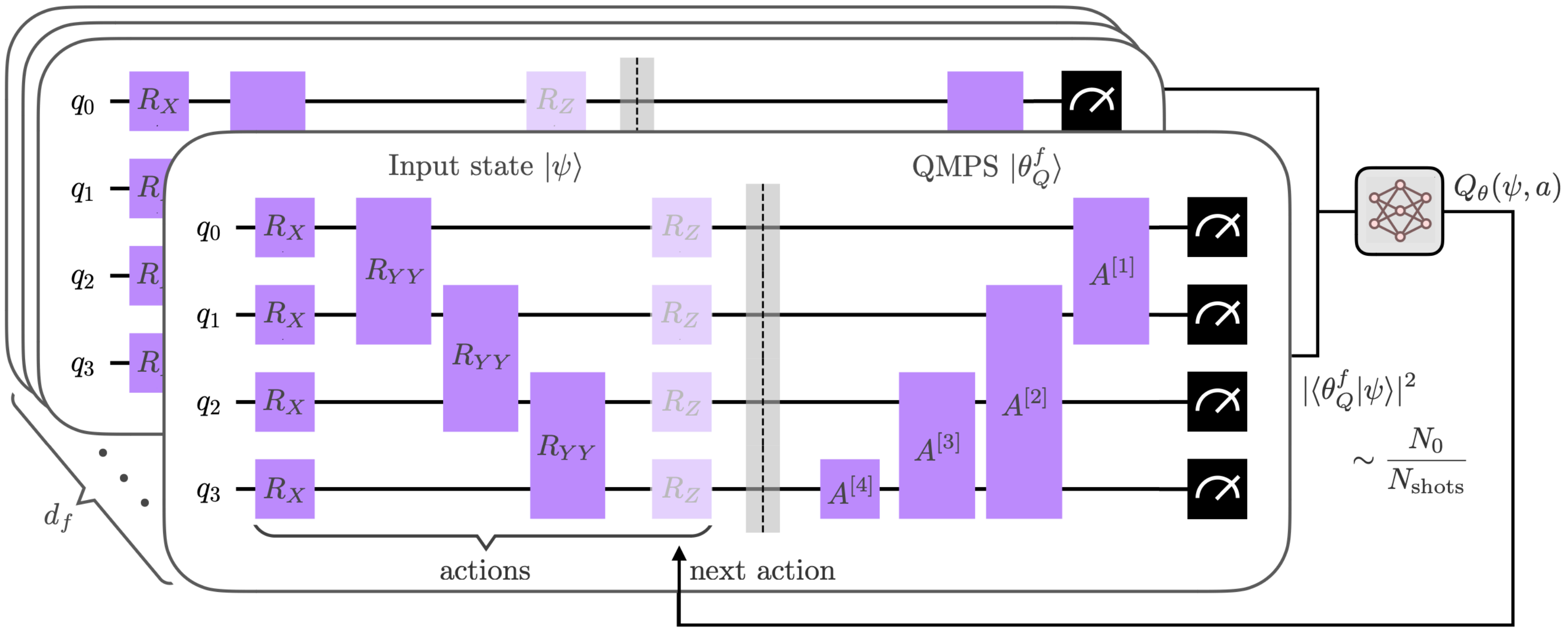
Nature Machine Intelligence **5**, 780-791 (2023)

Outlook

- Universal state preparation / many-body state preparation

- **QMPS circuit framework**

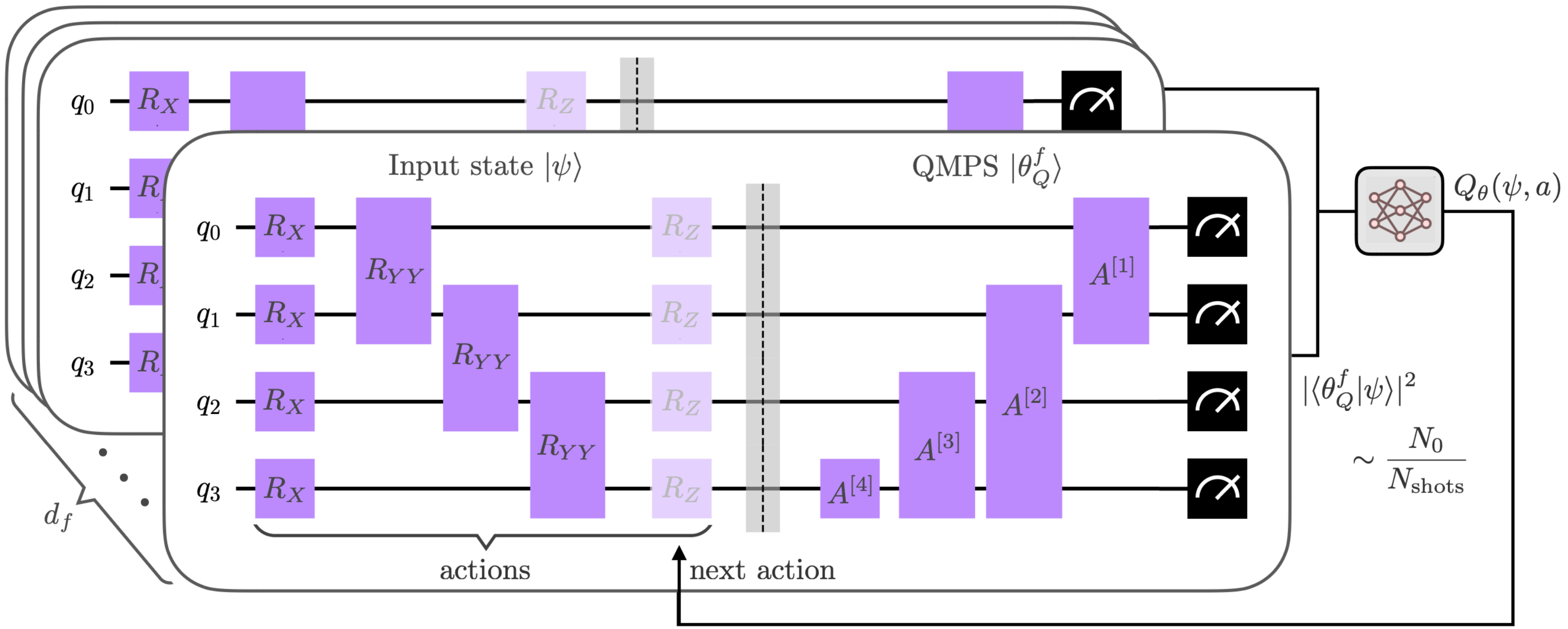
→ Hybrid quantum-classical ansatz:



Nature Machine Intelligence **5**, 780-791 (2023)

Outlook

- Universal state preparation / many-body state preparation



- **QMPS circuit framework**

→ Hybrid quantum-classical ansatz:

Marin Bukov

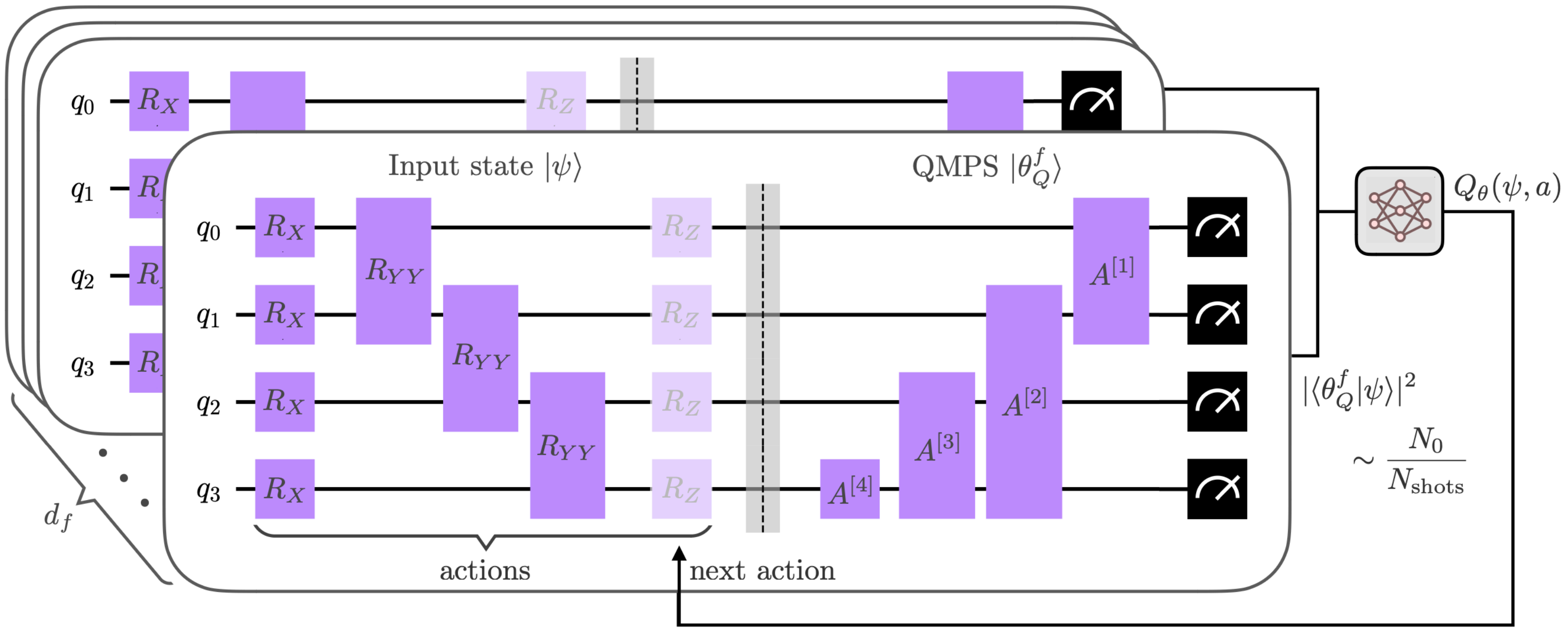
Nature Machine Intelligence **5**, 780-791 (2023)

Outlook

- Universal state preparation / many-body state preparation

- **QMPS circuit framework**

→ Hybrid quantum-classical ansatz:



Marin Bukov

Thank you!

Nature Machine Intelligence **5**, 780-791 (2023)