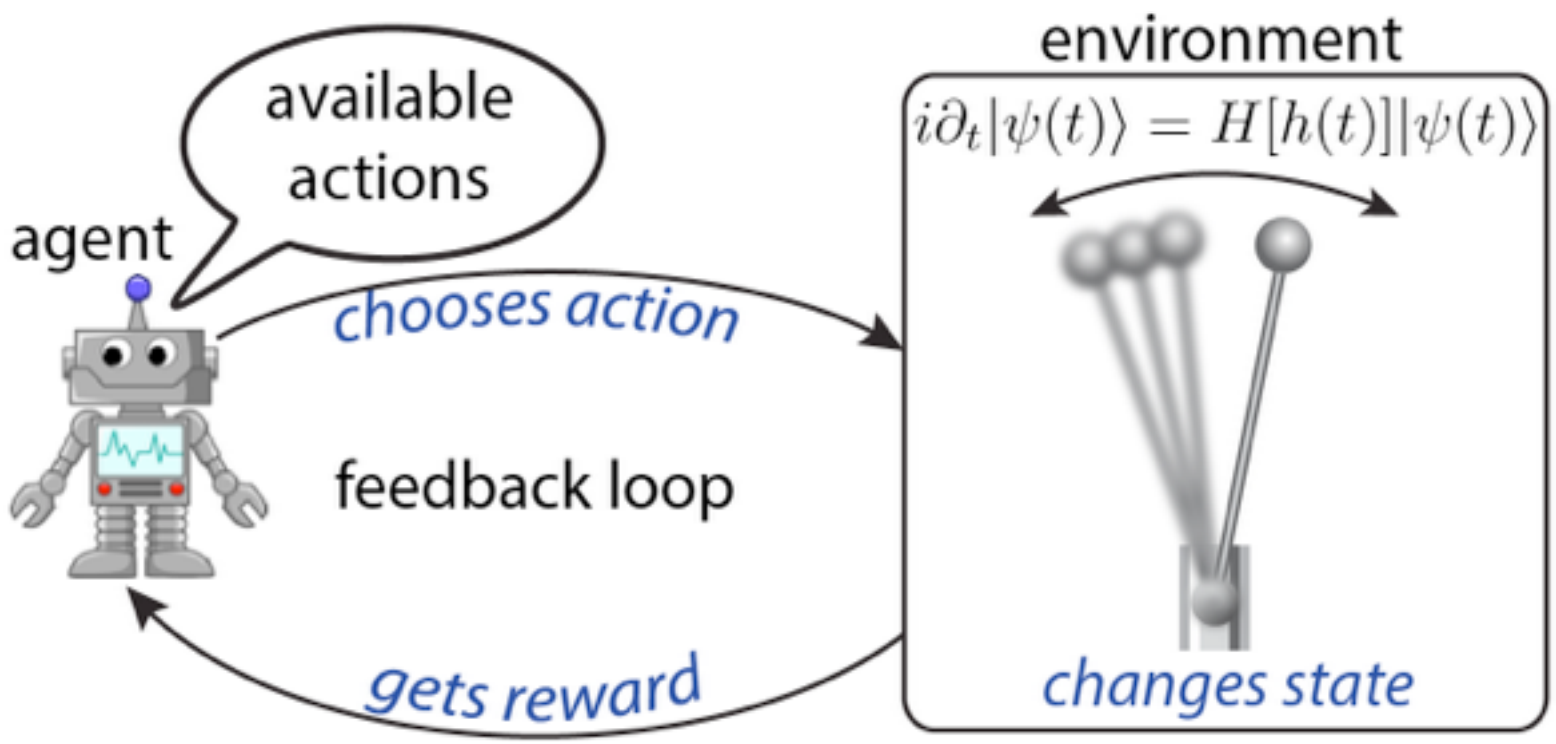
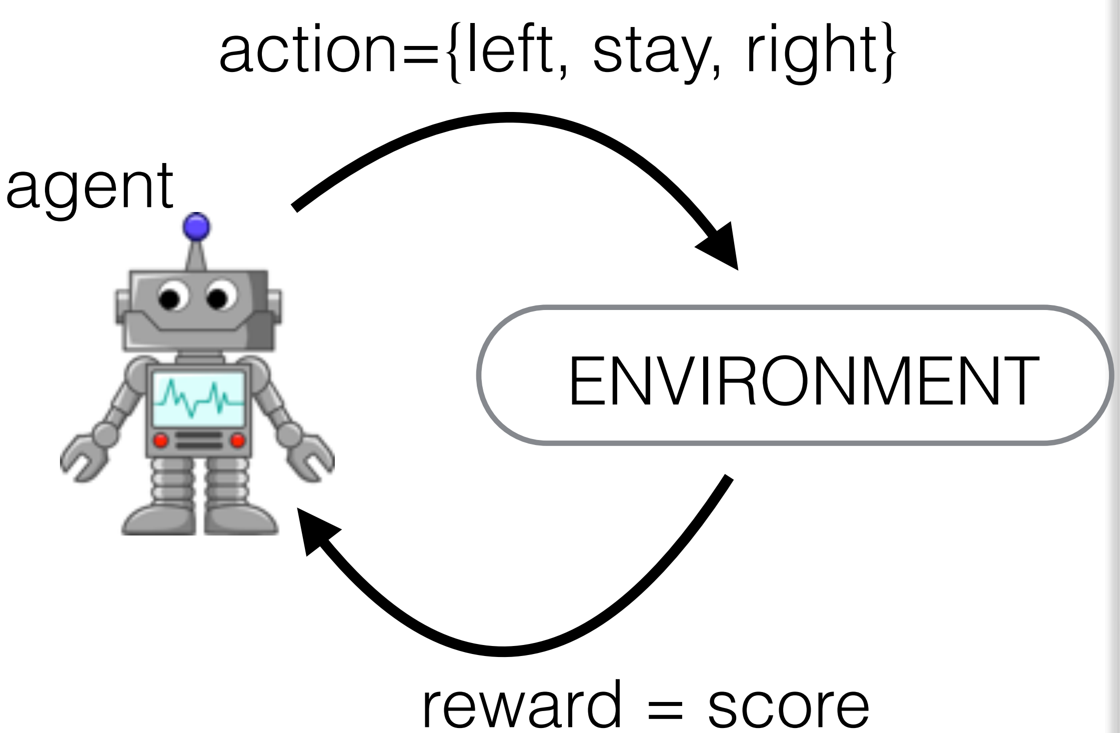
Reinforcement Learning: Introduction and Applications to Nonequilibrium Dynamics



Big Questions: Which problems can we study with ML that we can't do otherwise?
Can ML lead to the discovery of new physics?
What's ML's most appropriate physics application as a toolbox?

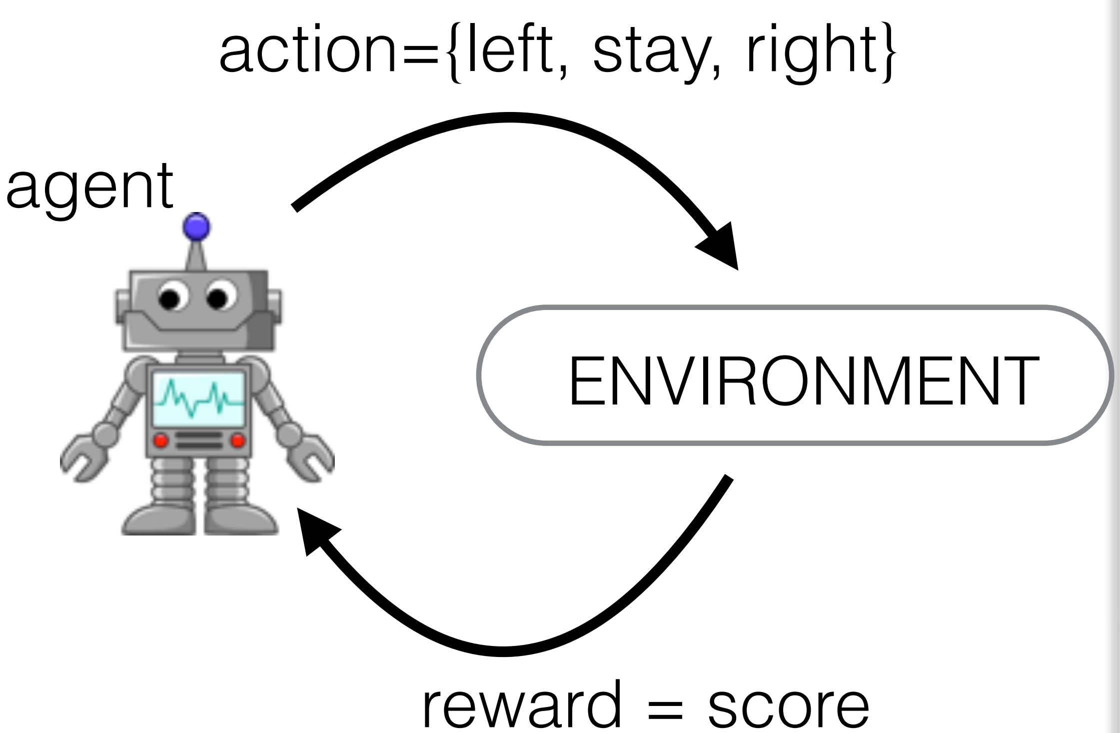
Marin Bukov

What is Reinforcement Learning?



Silver et al., Nature 529 (2016) [Google DeepMind]

What is Reinforcement Learning?



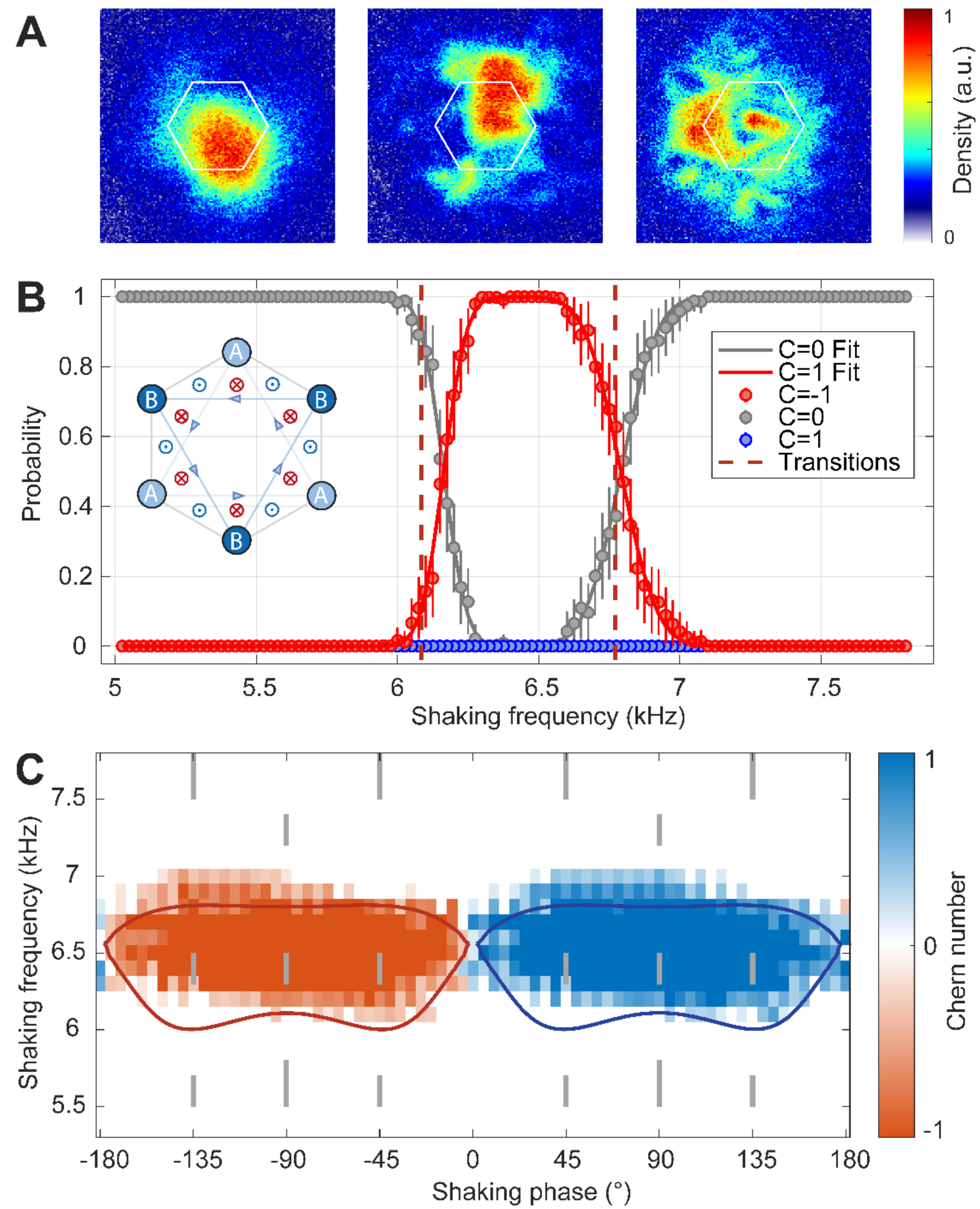
Silver et al., Nature 529 (2016) [Google DeepMind]

Machine Learning

→ Supervised Learning

- labelled data $\{(x,y)\}$
- find approx. model for the mapping $x \mapsto y$

learning topological phases in experiment



train set: 10,436 raw ToF images
test set: 15,963 images
validation set: 3,992 images

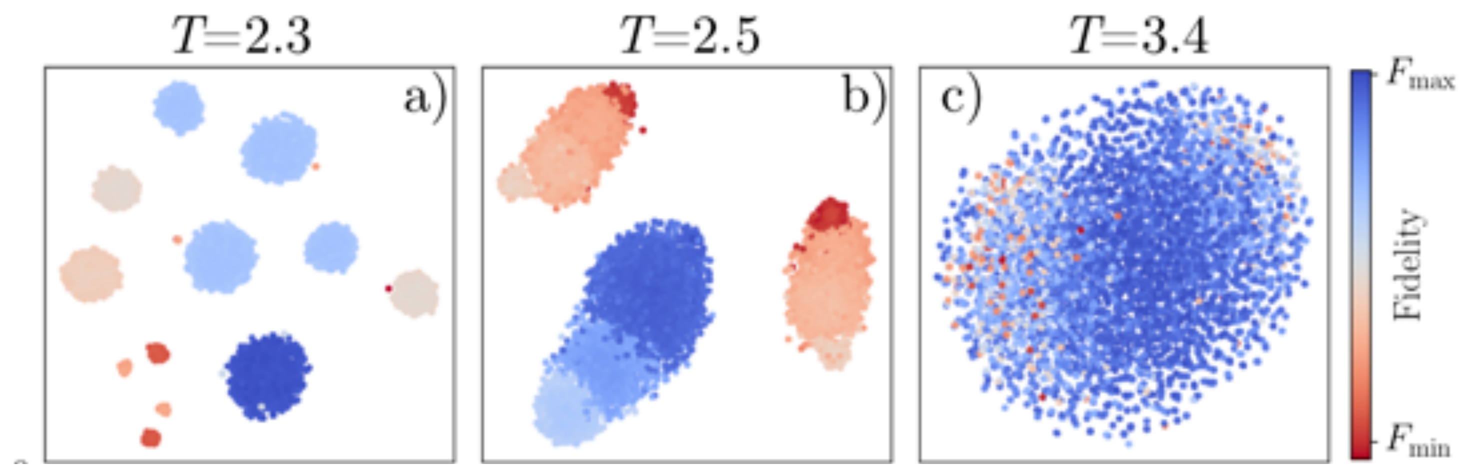
Berkeley UNIVERSITY OF CALIFORNIA

Machine Learning

→ Supervised Learning

- labelled data
- find approx. model which generalizes beyond fitting

visualize glassy (control) transitions



→ **Unsupervised Learning**

A. Day, M.B., et al, arXiv:1803:10856

- **un**labelled data $\{x\}$
- find approx. probability distr. which generates the data

Marin Bukov

Machine Learning

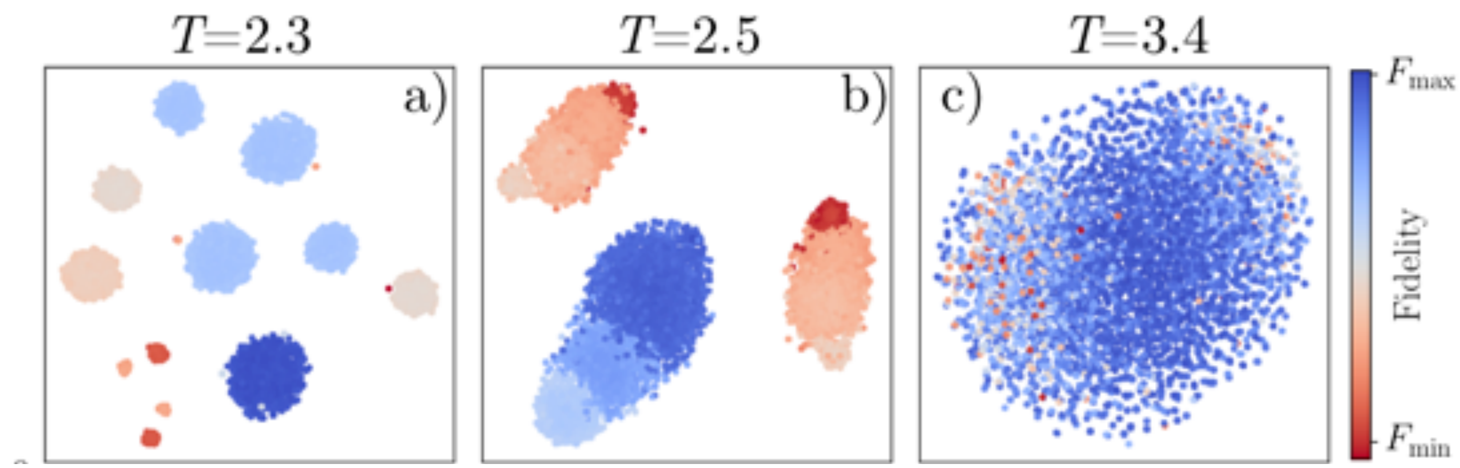
→ Supervised Learning

- labelled data
- find approx. model which generalizes beyond fitting

→ **Unsupervised Learning**

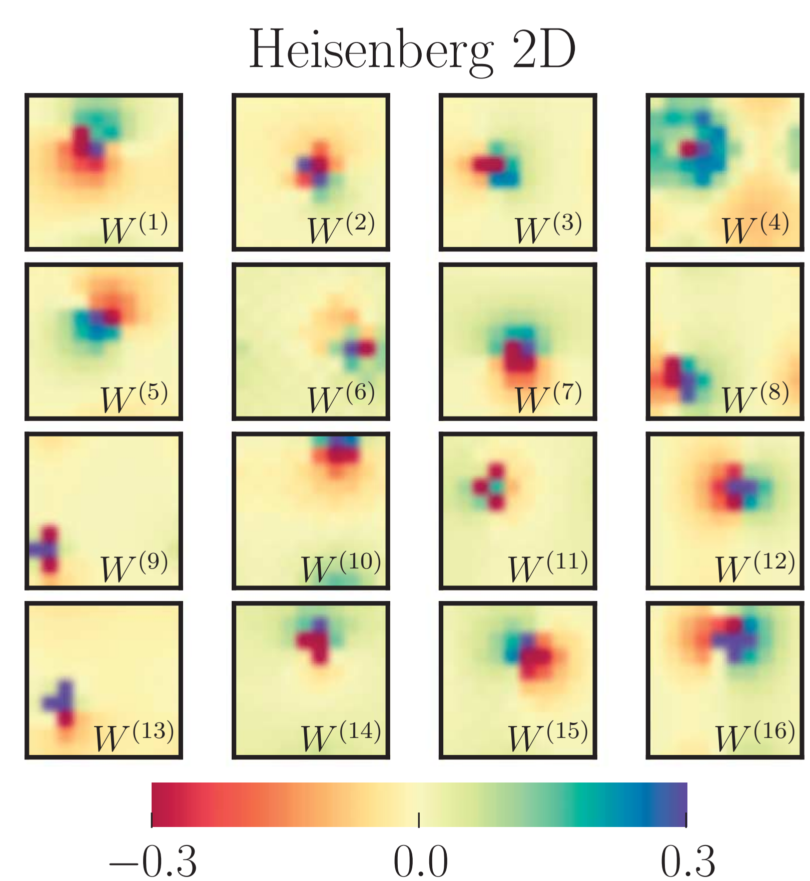
- **un**labelled data $\{x\}$
- find approx. probability distr. which generates the data

visualize glassy (control) transitions



A. Day, M.B., et al, arXiv:1803:10856

variational quantum states



Carleo and Troyer, Science (2017)

Berkeley
UNIVERSITY OF CALIFORNIA

Machine Learning

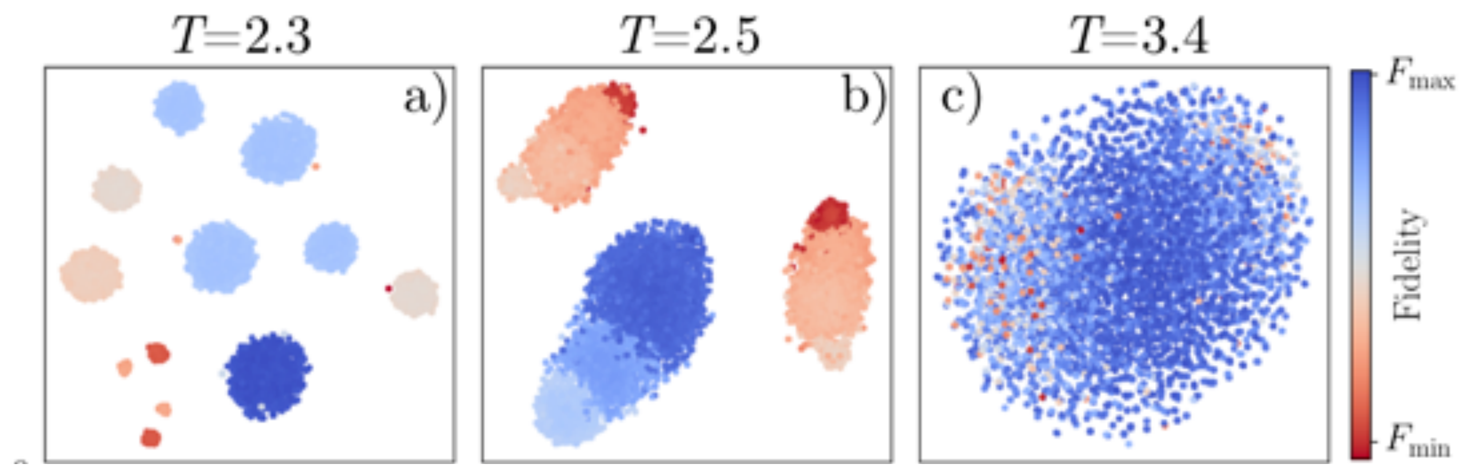
→ Supervised Learning

- labelled data
- find approx. model which generalizes beyond fitting

→ **Unsupervised Learning**

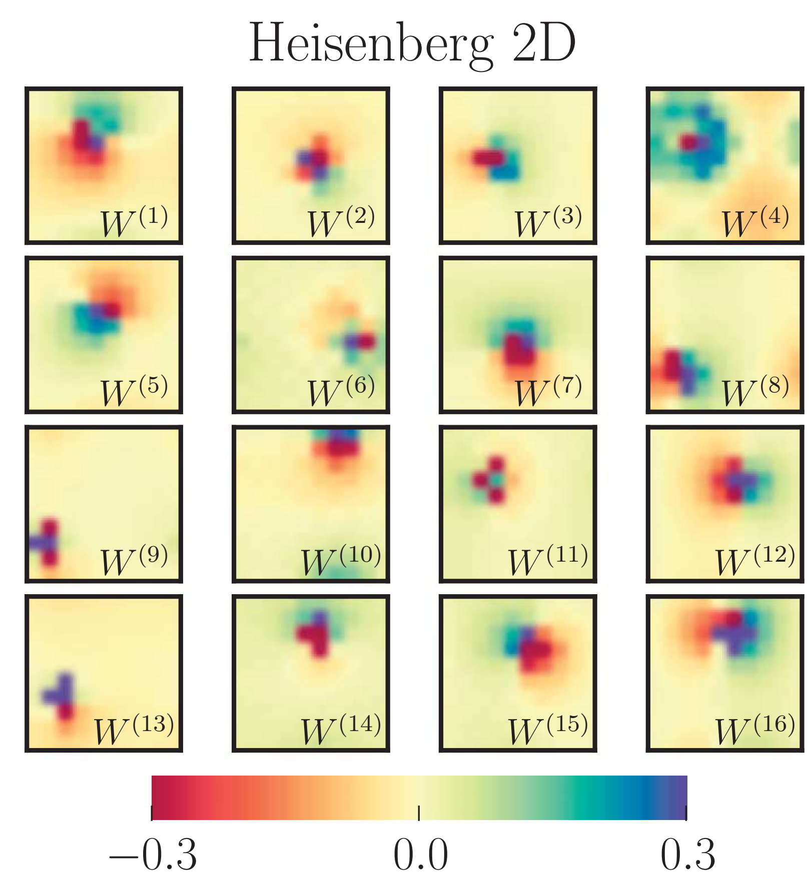
- **un**labelled data $\{x\}$
- find approx. probability distr. which generates the data

visualize glassy (control) transitions



A. Day, M.B., et al, arXiv:1803:10856

variational quantum states



Carleo and Troyer, Science (2017)

reviews: ML in physics

Dunjko and Briegel: *ML & AI in the Quantum Domain, Rep Prog Phys 81 074001 (2018)*

P. Mehta, M.B., et al: *High Bias, Low-Variance Intro to ML for Physicists,* arXiv: 1803:08823 (2018)

Marin Bukov

Machine Learning

Berkeley UNIVERSITY OF CALIFORNIA

→ Supervised Learning

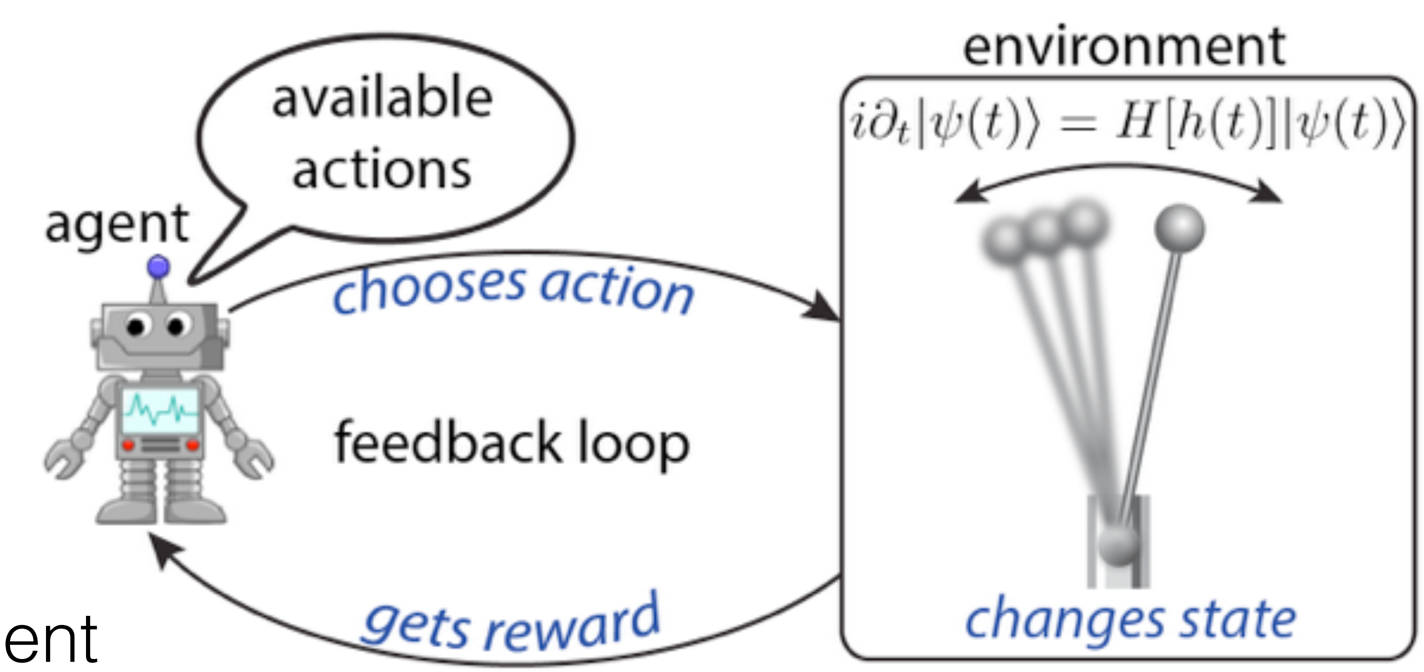
- labelled data
- find approx. model which generalizes beyond fitting

→ Unsupervised Learning

- unlabelled data
- find approx. probability distr. which generates the data

→ **Reinforcement Learning**

- agent learns strategy by interactions with its environment
- probability which generates the learning data changes with time due to interaction with the environment



Marin Bukov

Examples of RL Applications

outside physics

video games

Mnih et al, Nature (2015)

board games

Silver et al, Nature (2016)

locomotion

Lillicrap et al, arXiv:1509:02971

Berkeley
UNIVERSITY OF CALIFORNIA

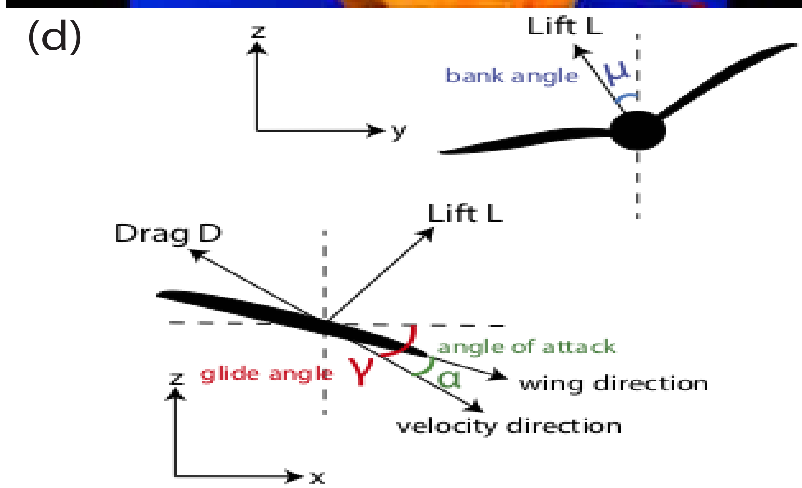
Examples of RL Applications

outside physics

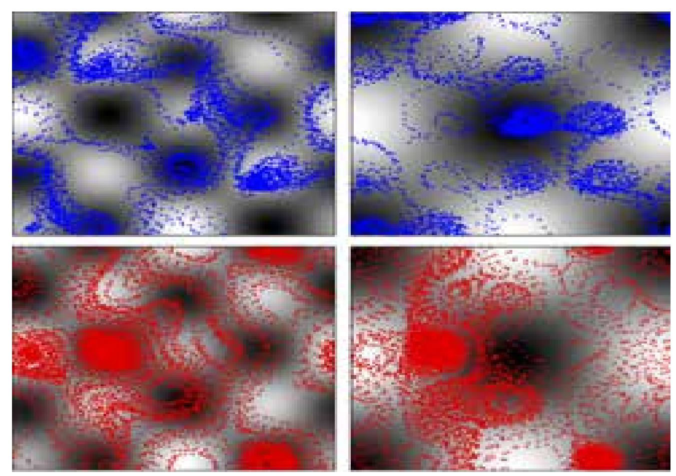
video games — **Mnih et al, Nature (2015)**

board games — **Silver et al, Nature (2016)**

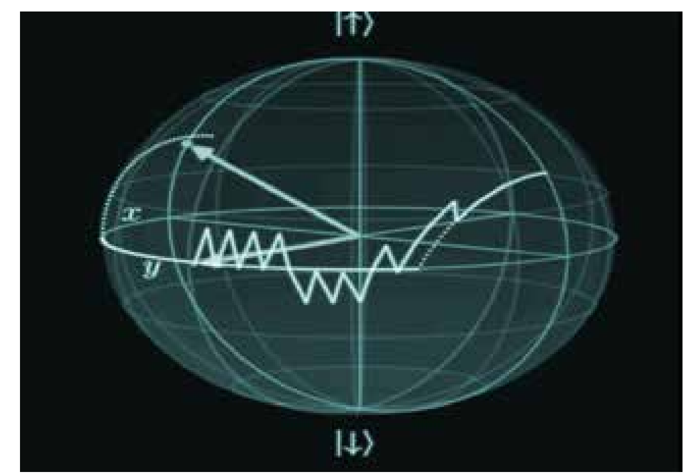
locomotion — **Lillicrap et al, arXiv:1509:02971**



(e)



(f)



Reddy et al, PNAS 113 4877 (2016)

Colabrese et al, PRL 118 15004 (2018)

M.B. et al, PRX 8 0311086 (2018)

Fossil et al, PRX 8 031084 (2018)

August et al, arXiv:1802.04063

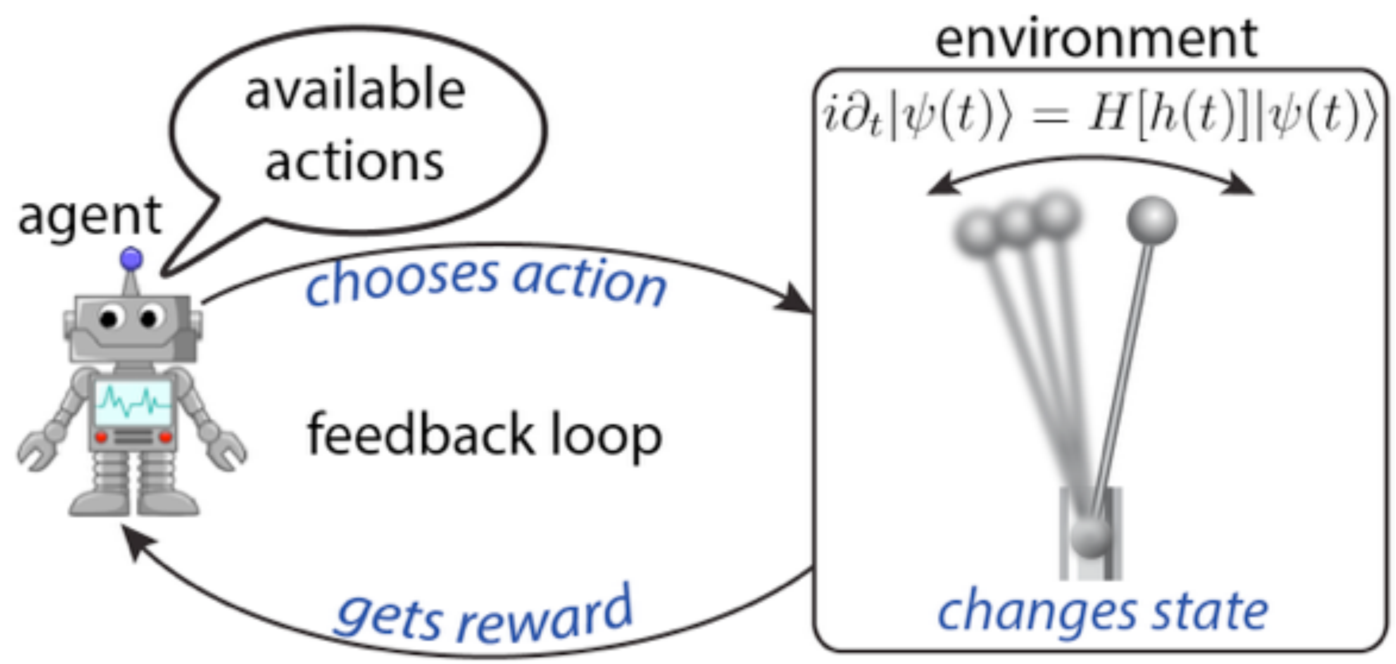
in physics

and more: design of molecular properties, quantum optics experiment design, etc.

The RL Formalism

→ RL formalism

- state space $\mathcal{S}$
- action space $\mathcal{A}$
- reward space $\mathcal{R}$



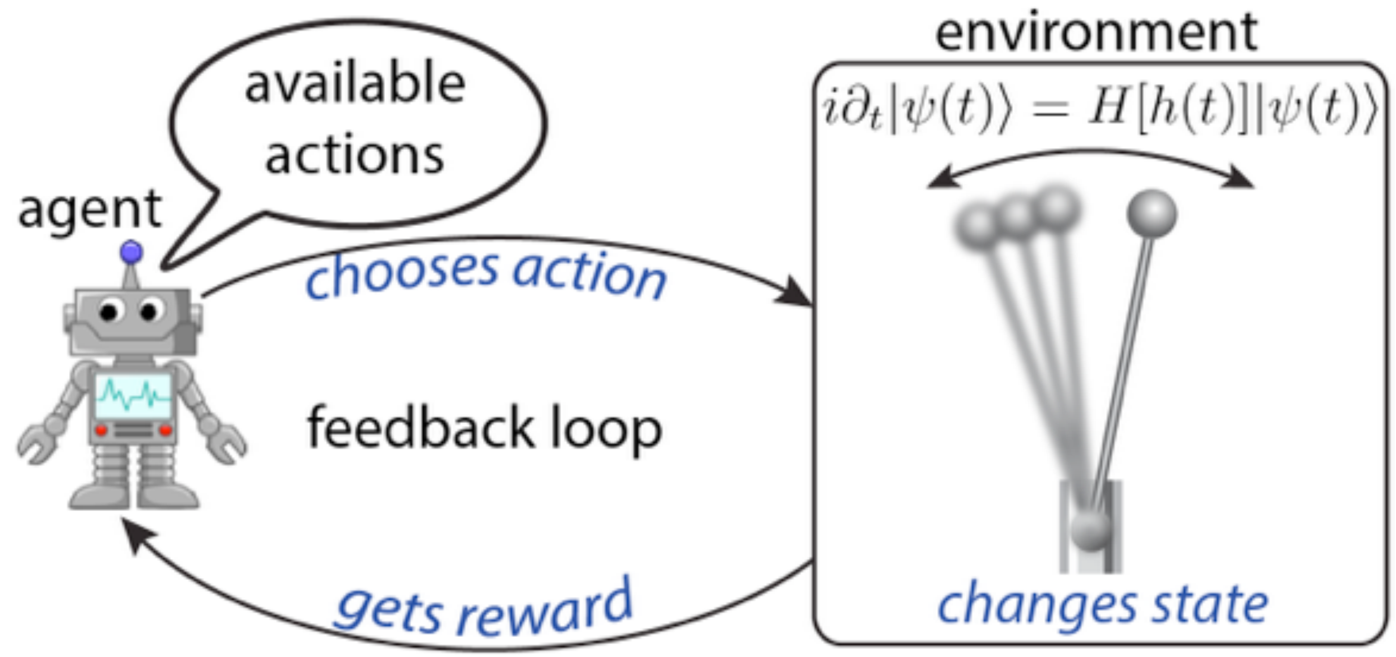
Marin Bukov

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

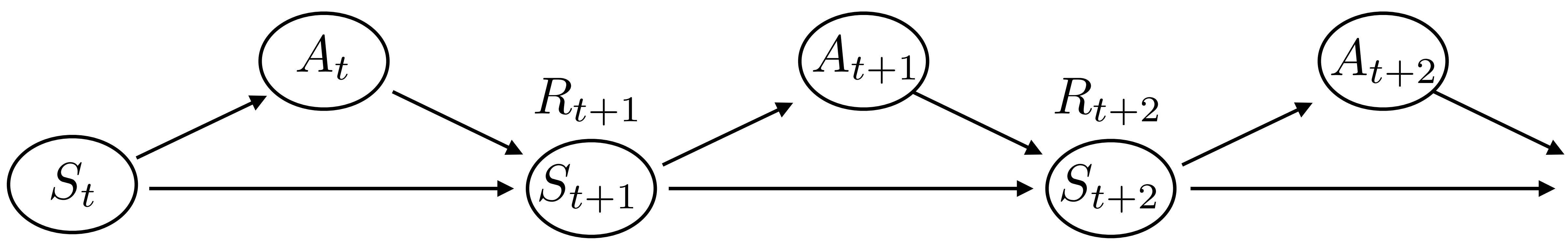
The RL Formalism

→ RL formalism

- state space $\mathcal{S}$
- action space $\mathcal{A}$
- reward space $\mathcal{R}$



→ RL as Markov decision process



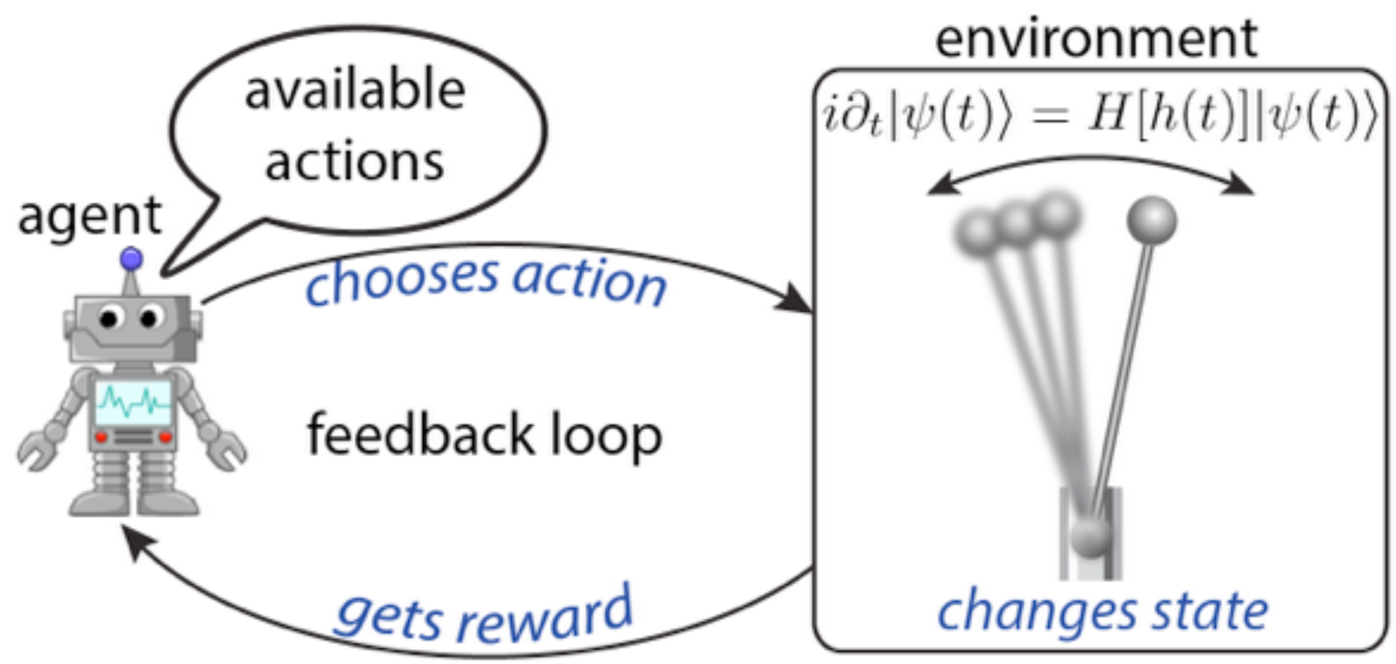
Marin Bukov

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

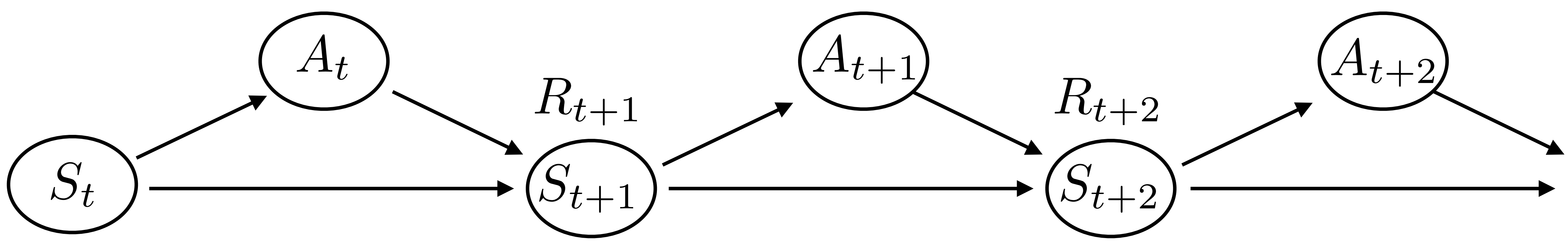
The RL Formalism

→ RL formalism

- state space $\mathcal{S}$
- action space $\mathcal{A}$
- reward space $\mathcal{R}$



→ RL as Markov decision process

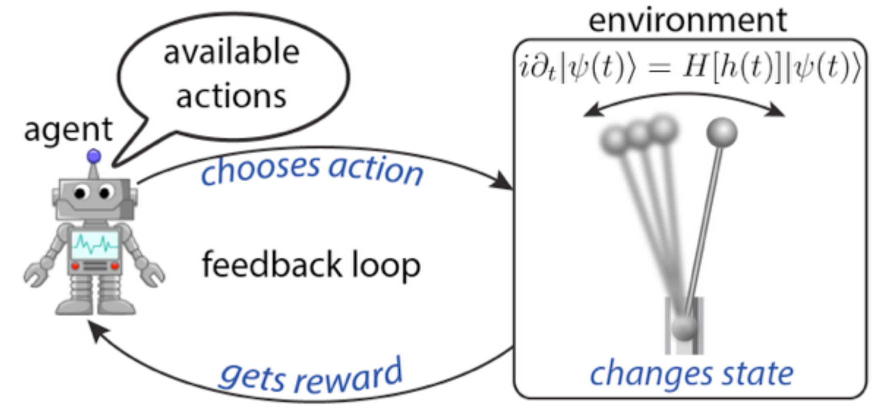


→ episodic learning

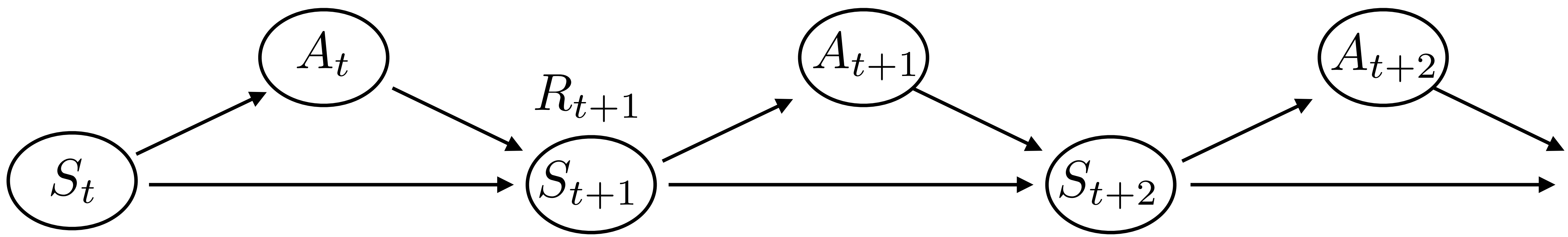
Marin Bukov

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

The RL Problem

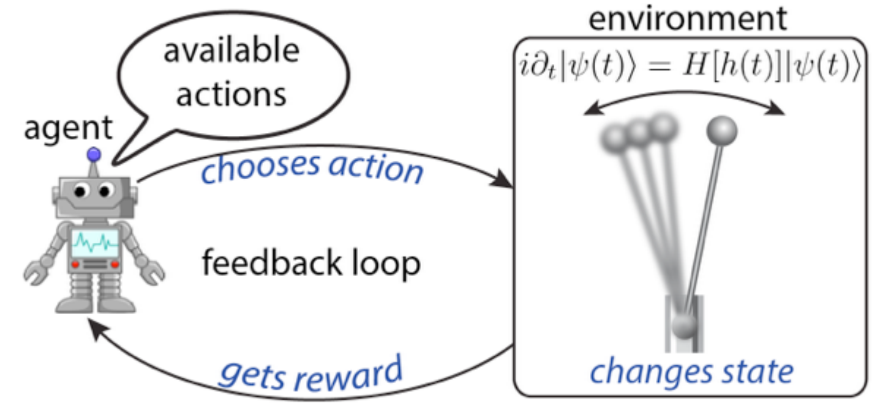


→ RL as Markov decision process

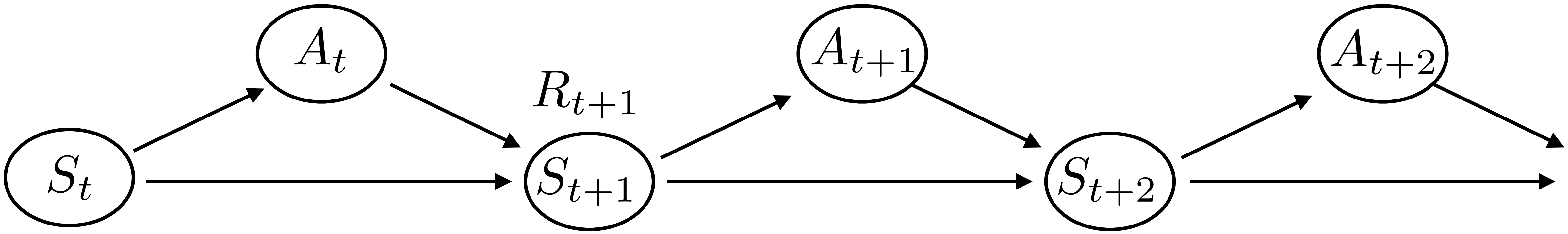


Sutton and Barto, *Reinforcement Learning: an Introduction*, MIT press

The RL Problem



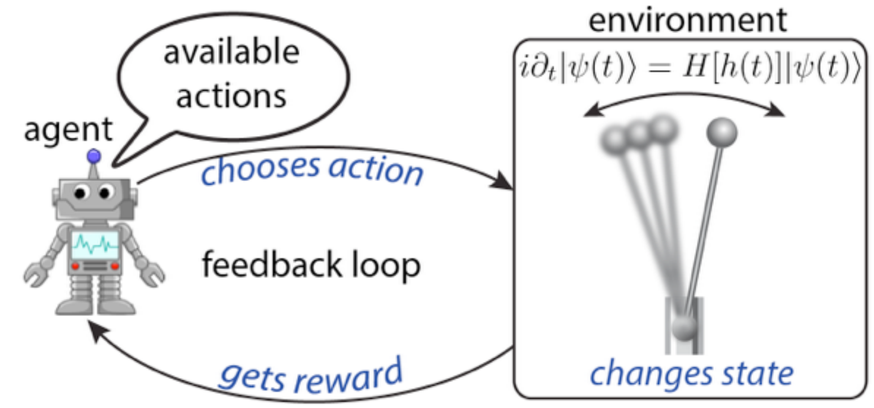
- RL as Markov decision process



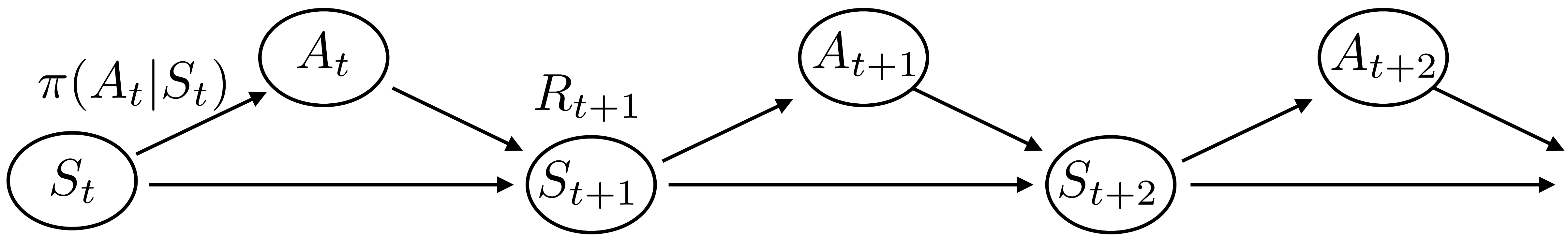
- RL algos have modular structure:

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

The RL Problem



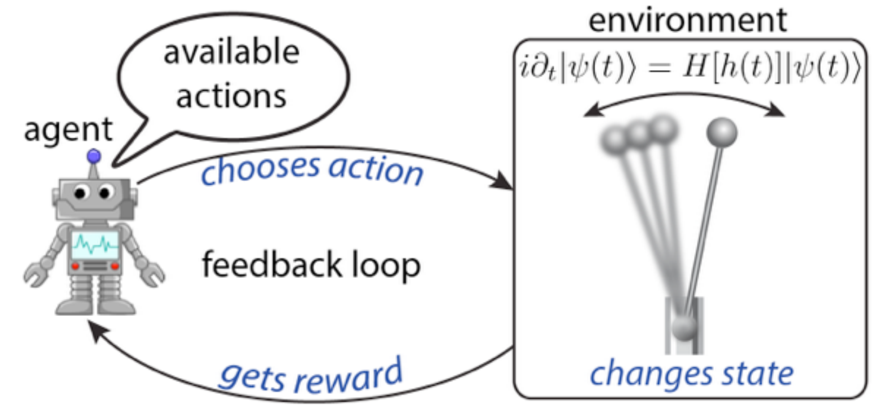
- RL as Markov decision process



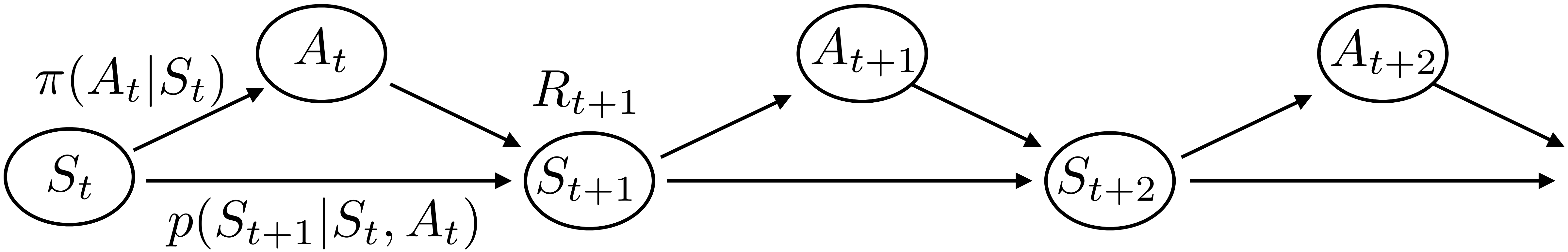
- RL algos have modular structure:
  - **RL agent**: decision-making algorithm to learn & improve the ***policy***

    $\pi(a|s)$ probability distribution $\mathcal{A} \times \mathcal{S} \to [0,1]$

The RL Problem



→ RL as Markov decision process



→ RL algos have modular structure:

- **RL agent**: decision-making algorithm to learn & improve the ***policy***

  $\pi(a|s)$ probability distribution $\mathcal{A} \times \mathcal{S} \to [0,1]$

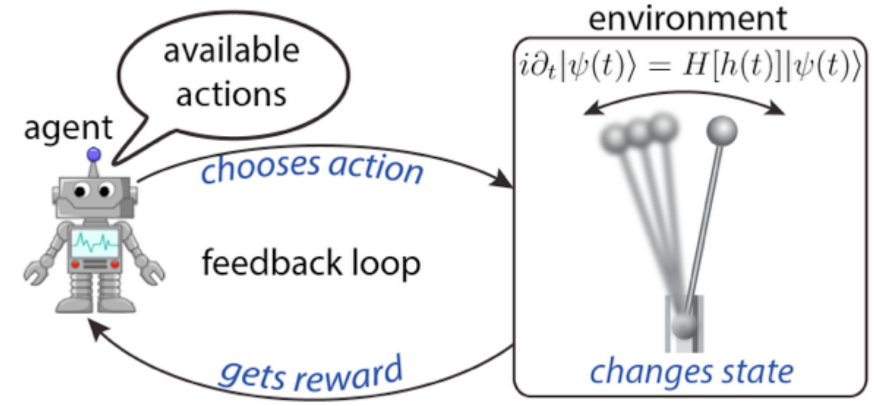
- **environment**: contains fixed state-action ***transition probabilities***

  $p(s'|s,a)$ probability distribution $\mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0,1]$

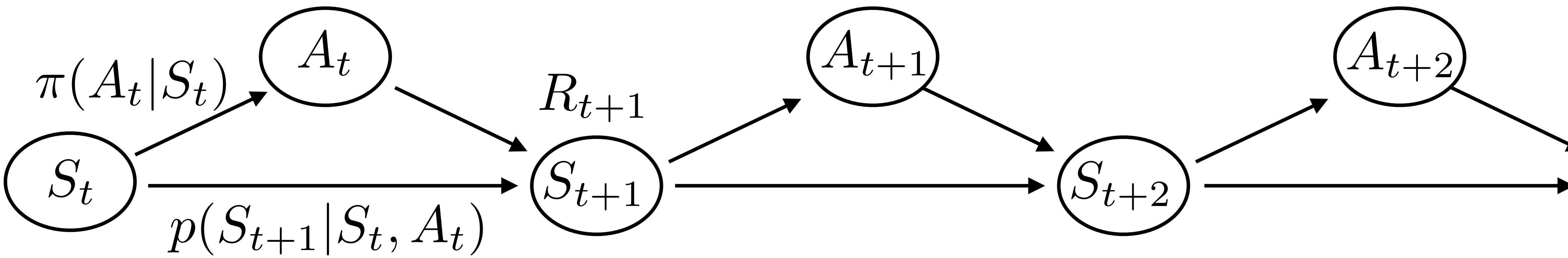
Sutton and Barto, *Reinforcement Learning: an Introduction*, MIT press

Berkeley
UNIVERSITY OF CALIFORNIA

The RL Problem



→ RL as Markov decision process



→ RL algos have modular structure:

- **RL agent**: decision-making algorithm to learn & improve the ***policy***

  $\pi(a|s)$ probability distribution $\mathcal{A}\times\mathcal{S}\to[0,1]$

- **environment**: contains fixed state-action ***transition probabilities***

  $p(s'|s,a)$ probability distribution $\mathcal{S}\times\mathcal{S}\times\mathcal{A}\to[0,1]$

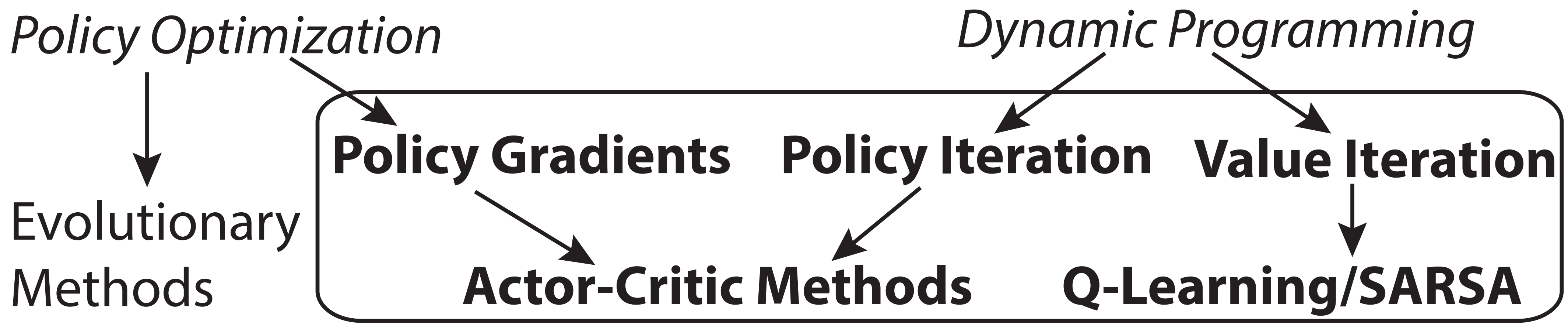
→ RL **objective**: find policy which maximizes the total *expected return* from step t onwards $G_t = R_{t+1} + \cdots + R_T$

$$R_{t+1} = \sum_{s'} p(s'|S_t, A_t)\, r(s', S_t, A_t)$$

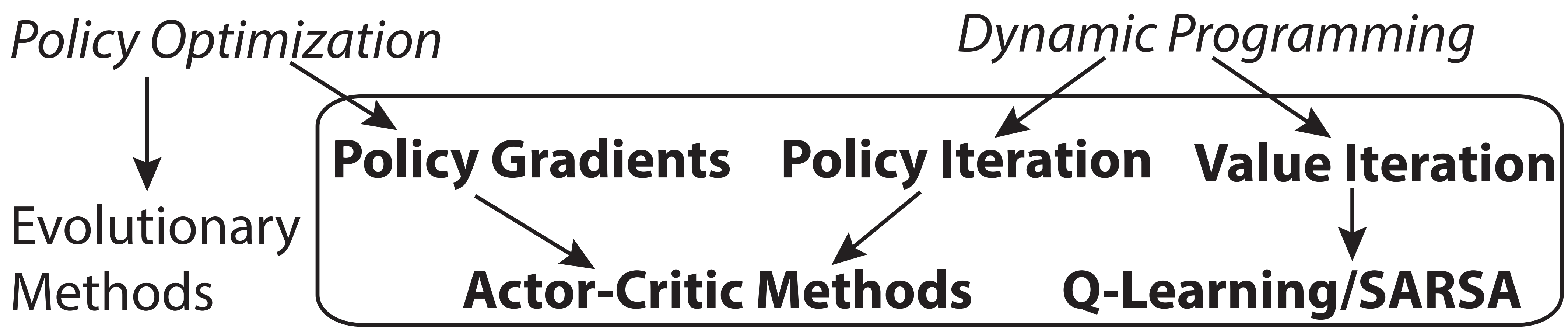
Marin Bukov

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

Overview of RL Algorithms



Overview of RL Algorithms



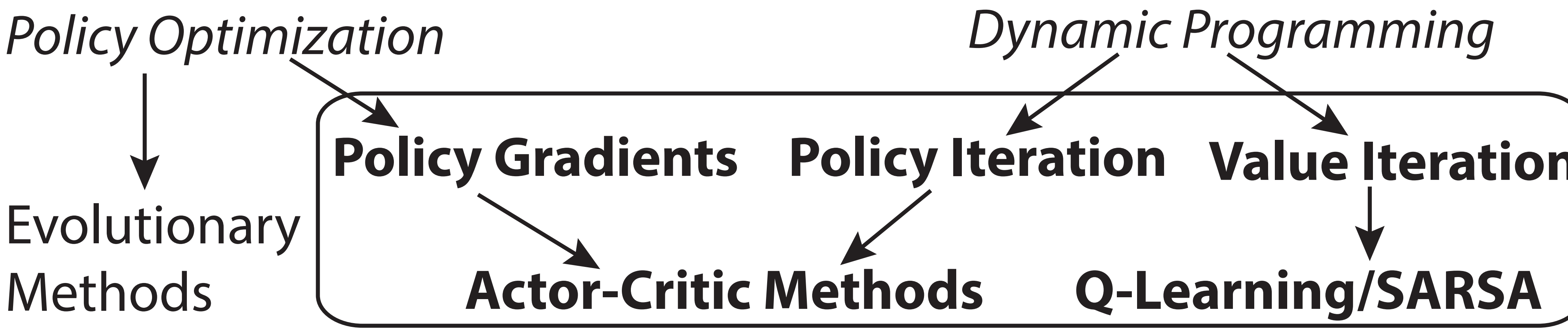
→ Value Iteration methods

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

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

Berkeley UNIVERSITY OF CALIFORNIA

Overview of RL Algorithms



→ Value Iteration methods

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

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

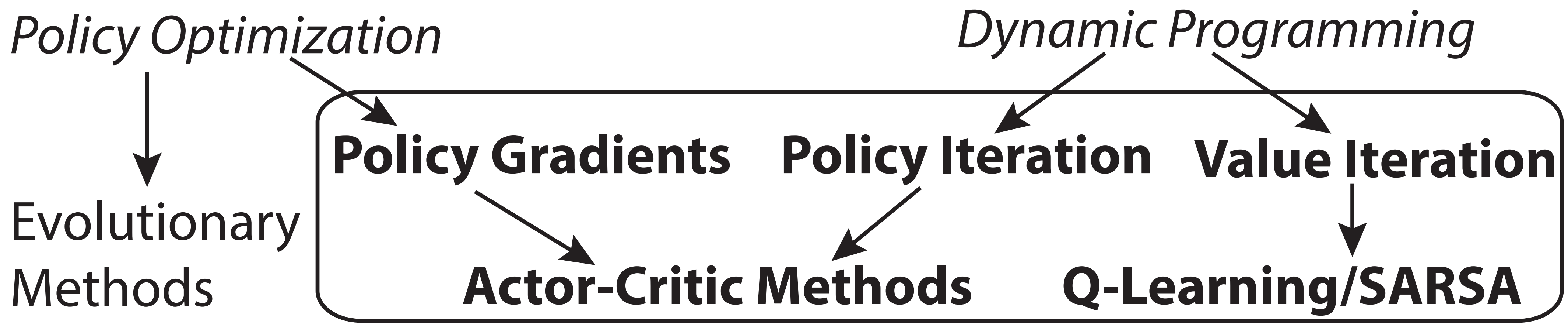
- action-value (or Q-) function: ***expected*** total return under the policy $\pi(a|s)$ starting from state s and taking action a:

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

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

Berkeley UNIVERSITY OF CALIFORNIA

Overview of RL Algorithms



→ Value Iteration methods

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

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

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

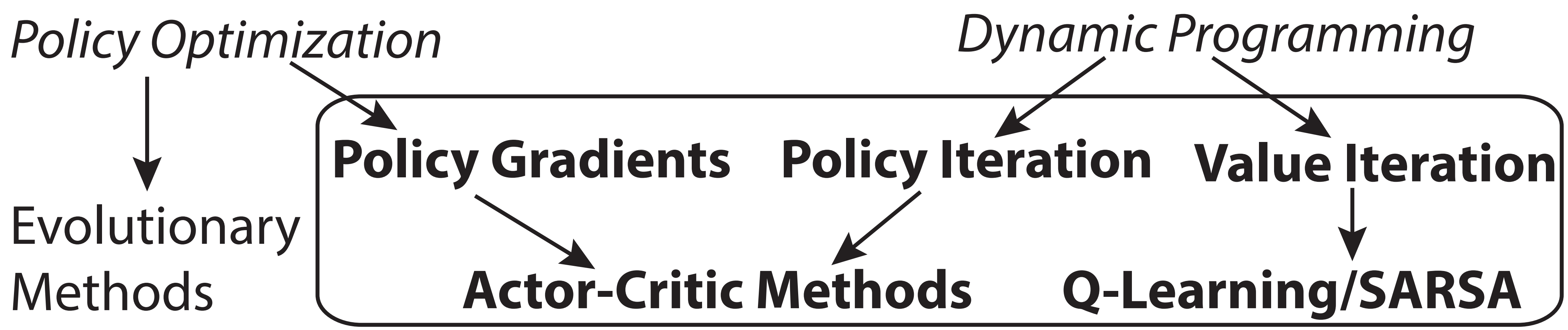
$$q_\pi(s,a) = \mathbb{E}_{a\sim\pi(a|s)}[G_t|S_t = s, A_t = a]$$

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

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

Berkeley
UNIVERSITY OF CALIFORNIA

Overview of RL Algorithms



→ Value Iteration methods

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

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

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

$$G_t = R_{t+1} + G_{t+1}$$

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

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

Marin Bukov

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

RL with Function Approximation

→ problem: state space has exponentially many configurations

- can we estimate values of not yet encountered states?

RL with Function Approximation

- problem: state space has exponentially many configurations
  - can we estimate values of not yet encountered states?
- YES, interpolate by parametrizing the Q-function/policy

$$q(s,a) \to q_\theta(s,a) \qquad \pi(a|s) \to \pi_\theta(a|s)$$

Berkeley UNIVERSITY OF CALIFORNIA

RL with Function Approximation

→ problem: state space has exponentially many configurations

- can we estimate values of not yet encountered states?

→ YES, interpolate by parametrizing the Q-function/policy

$$q(s,a) \to q_\theta(s,a) \qquad\qquad \pi(a|s) \to \pi_\theta(a|s)$$

- typical approach: use deep neural network **(Deep RL)**
- caveat: RL algorithms have convergence guarantees only for linear function approximators
- lots of empirical tricks to combine Deep Learning and RL

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

Berkeley UNIVERSITY OF CALIFORNIA

RL with Function Approximation

→ problem: state space has exponentially many configurations

- can we estimate values of not yet encountered states?

→ YES, interpolate by parametrizing the Q-function/policy

$$q(s,a) \to q_\theta(s,a) \qquad\qquad \pi(a|s) \to \pi_\theta(a|s)$$

- typical approach: use deep neural network **(Deep RL)**
- caveat: RL algorithms have convergence guarantees only for linear function approximators
- lots of empirical tricks to combine Deep Learning and RL

→ examples of Deep RL:

- Tesauro's Backgammon RL player (1992)
- DeepMind: Atari games, AlphaGo, etc.
- self-driving cars, autonomous drone/helicopter hovering, etc.

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

RL and Optimal Control (OC)

→ different sides of the same medal

- RL: appeared first in behavioral psychology: decision making
- OC: appeared in optimization problem solving: variational calculus

Berkeley UNIVERSITY OF CALIFORNIA

RL and Optimal Control (OC)

→ different sides of the same medal

- RL: appeared first in behavioral psychology: decision making
- OC: appeared in optimization problem solving: variational calculus

→ modern Control Theory: both RL and OC under same hood

→ currently in physics: preferred approach is OC

- for technical reasons: RL required large computational power, big data

RL and Optimal Control (OC)

→ different sides of the same medal

- RL: appeared first in behavioral psychology: decision making
- OC: appeared in optimization problem solving: variational calculus

→ modern Control Theory: both RL and OC under same hood

→ currently in physics: preferred approach is OC

- for technical reasons: RL required large computational power, big data

OC ← *closely related* → **RL**

OC	**RL**
based on: ***variational calculus***	***Markov decision processes***
• needs model for environment to express cost function in.	• no model of controlled system, adaptive, autonomous.
• best suited for deterministic environments.	• stochastic/uncertain environments.
• differentiable cost function C_h uses gradient descent.	• reward function can be discontinuous, noisy.
• advantage: if we can compute analytically derivative of C_h.	• figures out effective degrees of freedom without a model.

Berkeley
UNIVERSITY OF CALIFORNIA

Why RL in Nonequilibrium Dynamics?

→ **model-free:** find effective control degrees of freedom (dof)

- microscopic descriptions have extensively many dof
- cannot solve equations of motion
- use (deep) RL to find guiding principles away from equilibrium?
- RL can handle uncertain environments and learn policies in the presence of various (correlated) sources of noise?

Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

Why RL in Nonequilibrium Dynamics?

→ **model-free:** find effective control degrees of freedom (dof)

- microscopic descriptions have extensively many dof
- cannot solve equations of motion
- use (deep) RL to find guiding principles away from equilibrium?
- RL can handle uncertain environments and learn policies in the presence of various (correlated) sources of noise?

→ **adaptive:** train on one env., use in a different env.

- Q-function contains knowledge about the environment which can be used after training
- RL can reveal similarities between at first sight unrelated problems?

Berkeley UNIVERSITY OF CALIFORNIA

Why RL in Nonequilibrium Dynamics?

→ **model-free:** find effective control degrees of freedom (dof)

- microscopic descriptions have extensively many dof
- cannot solve equations of motion
- use (deep) RL to find guiding principles away from equilibrium?
- RL can handle uncertain environments and learn policies in the presence of various (correlated) sources of noise?

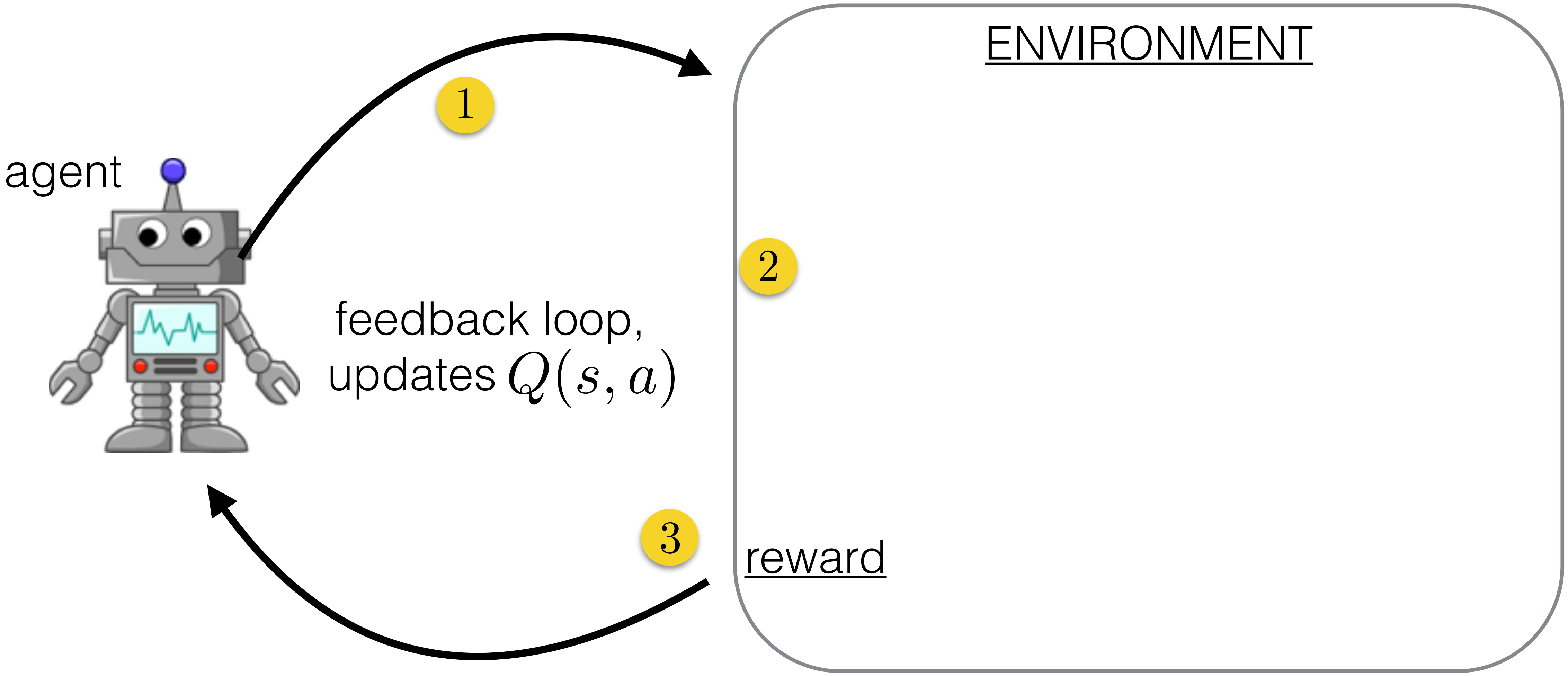
→ **adaptive:** train on one env., use in a different env.

- Q-function contains knowledge about the environment which can be used after training
- RL can reveal similarities between at first sight unrelated problems?

→ **autonomous:** does not require supervision

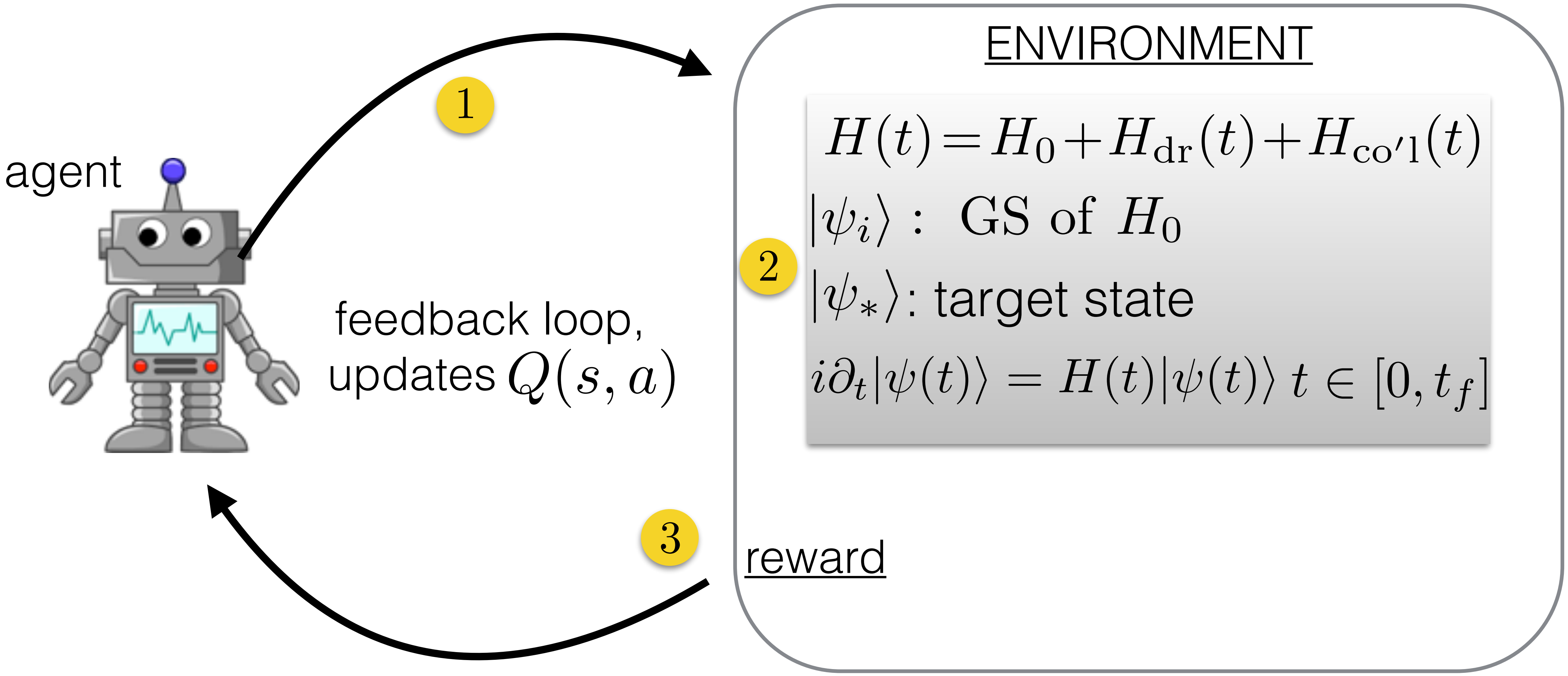
- RL can automate experimental setups?
- **on-line:** improve policy on-the-fly, i.e. before episode is over

RL Applied to Quantum State Preparation



t_f

RL Applied to Quantum State Preparation

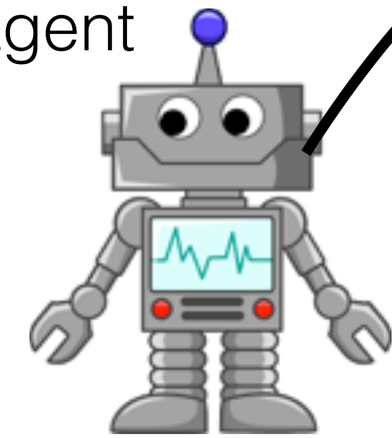


t_f

Berkeley
UNIVERSITY OF CALIFORNIA

RL Applied to Quantum State Preparation

agent



1

feedback loop, updates $Q(s,a)$

ENVIRONMENT

2

$$H(t) = H_0 + H_{\rm dr}(t) + H_{\rm co'l}(t)$$

$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t\in[0,t_f]$$

3

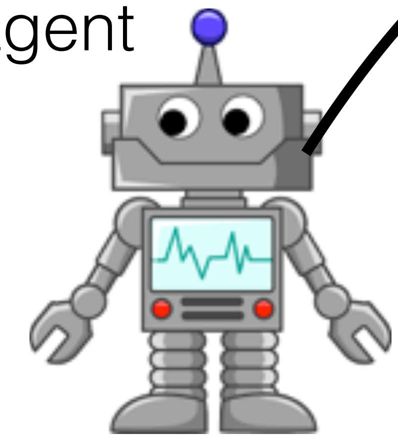
reward $r = \begin{cases} 0 & , 0 \le t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$

t_f

Marin Bukov

RL Applied to Quantum State Preparation

agent



1

feedback loop, updates $Q(s,a)$

ENVIRONMENT

$H(t) = H_0 + H_{\mathrm{dr}}(t) + H_{\mathrm{co'l}}(t)$

$|\psi_i\rangle$: GS of H_0

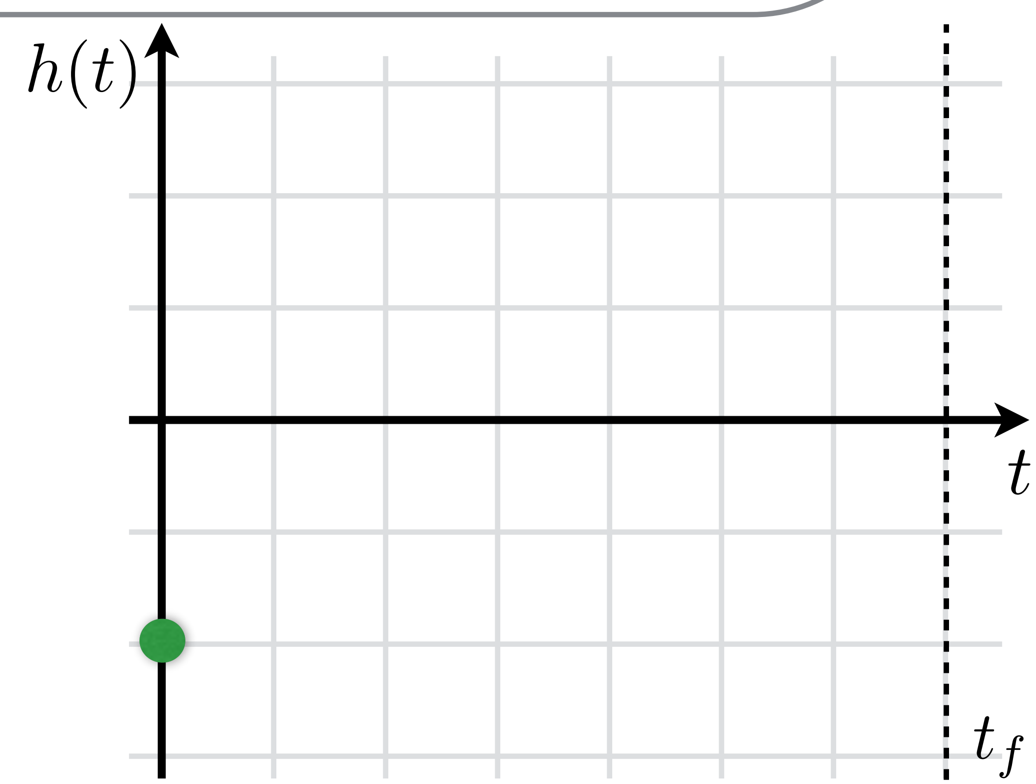
2

$|\psi_*\rangle$: target state

$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t \in [0, t_f]$

3

reward $r = \begin{cases} 0 & , 0 \leq t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2, & t = t_f \end{cases}$

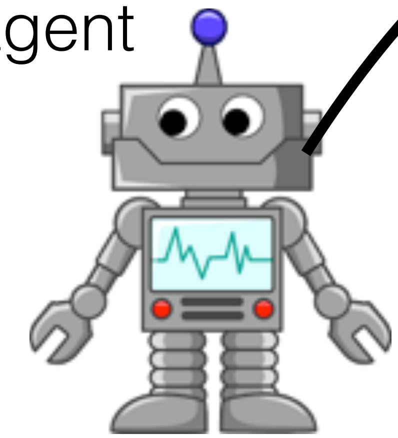


Marin Bukov

Berkeley UNIVERSITY OF CALIFORNIA

RL Applied to Quantum State Preparation

agent



feedback loop, updates $Q(s,a)$

ENVIRONMENT

$$H(t) = H_0 + H_{\mathrm{dr}}(t) + H_{\mathrm{co'l}}(t)$$

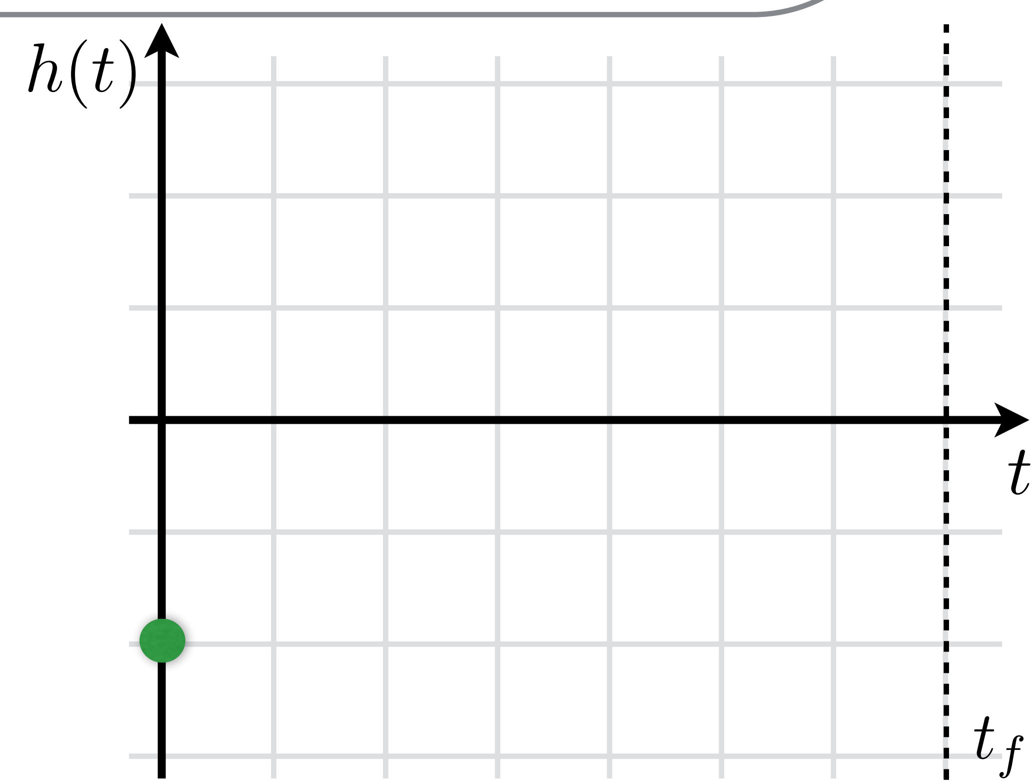
$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \quad t\in[0,t_f]$$

reward $r = \begin{cases} 0 & , 0 \le t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$

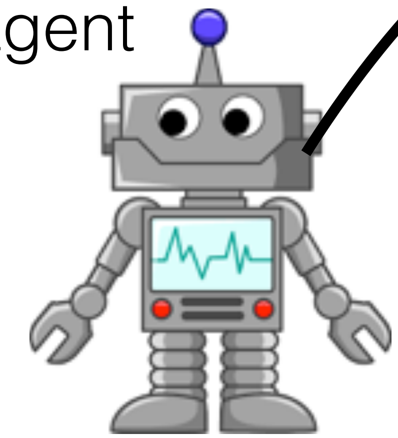
1 start from state $s_0 = [h(0)] = [-4]$



Marin Bukov

RL Applied to Quantum State Preparation

agent



feedback loop,
updates $Q(s,a)$

1

ENVIRONMENT

2

$$H(t)=H_0+H_{\mathrm{dr}}(t)+H_{\mathrm{co'l}}(t)$$

$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t\in[0,t_f]$$

3

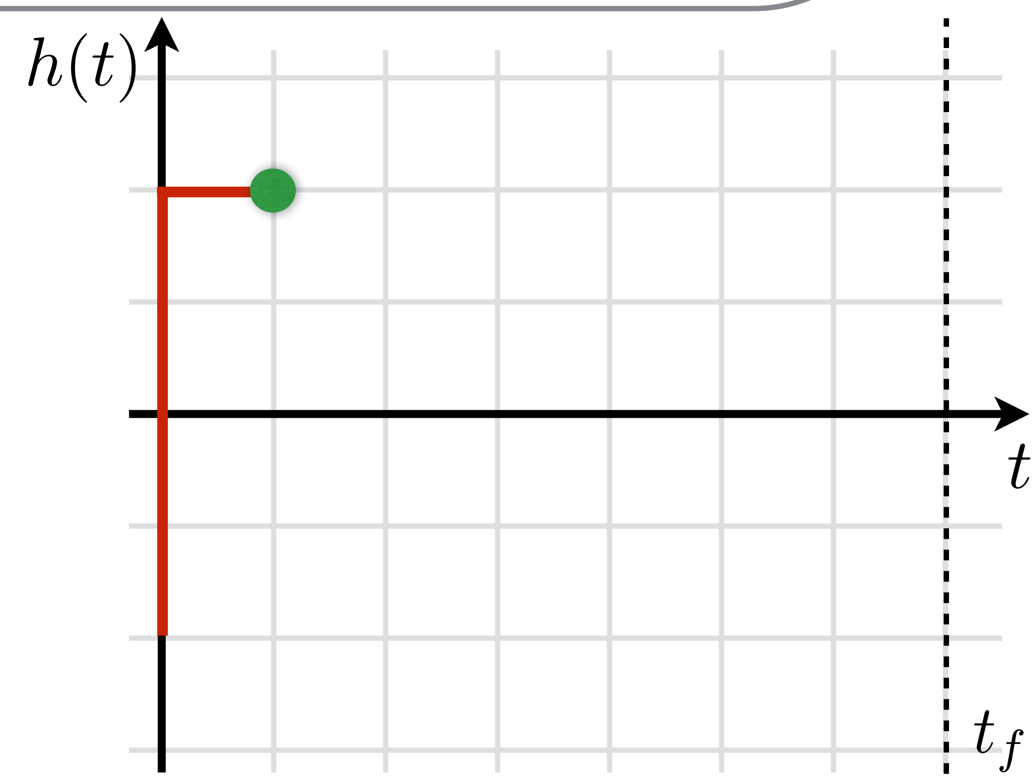
reward $r = \begin{cases} 0 & , 0\leq t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t=t_f \end{cases}$

1

start from state $s_0=[h(0)]=[-4]$

take action $a_0:\delta h=+4$

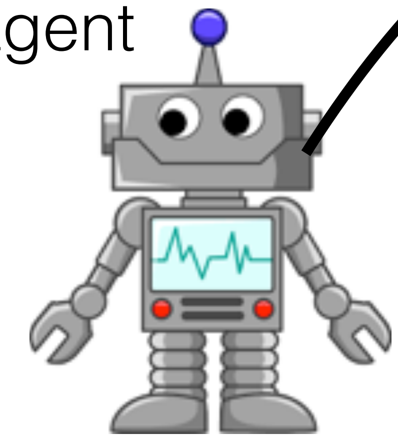
go to state $s_1=[h(0),h(\delta t)]=[-4,+4]$



Marin Bukov

RL Applied to Quantum State Preparation

agent



feedback loop, updates $Q(s,a)$

1

ENVIRONMENT

$$H(t) = H_0 + H_{\mathrm{dr}}(t) + H_{\mathrm{co'l}}(t)$$

$|\psi_i\rangle$: GS of H_0

2

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t \in [0, t_f]$$

3

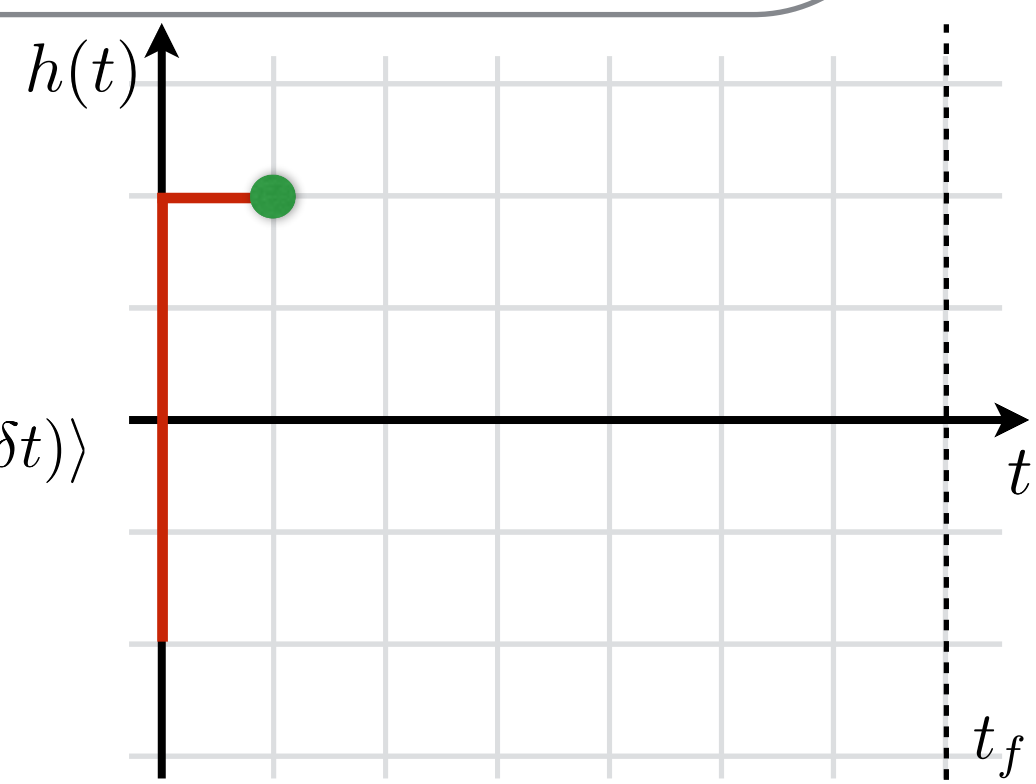
reward $r = \begin{cases} 0 & , 0 \leq t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$

1 start from state $s_0 = [h(0)] = [-4]$

take action $a_0 : \delta h = +4$

go to state $s_1 = [h(0), h(\delta t)] = [-4, +4]$

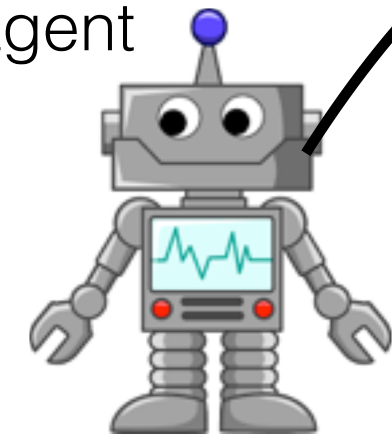
2 solve Schrödinger Eq. and obtain the QM state $|\psi(\delta t)\rangle$



Marin Bukov

RL Applied to Quantum State Preparation

agent



feedback loop, updates $Q(s, a)$

1

ENVIRONMENT

$$H(t) = H_0 + H_{\mathrm{dr}}(t) + H_{\mathrm{co'l}}(t)$$

2

$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t \in [0, t_f]$$

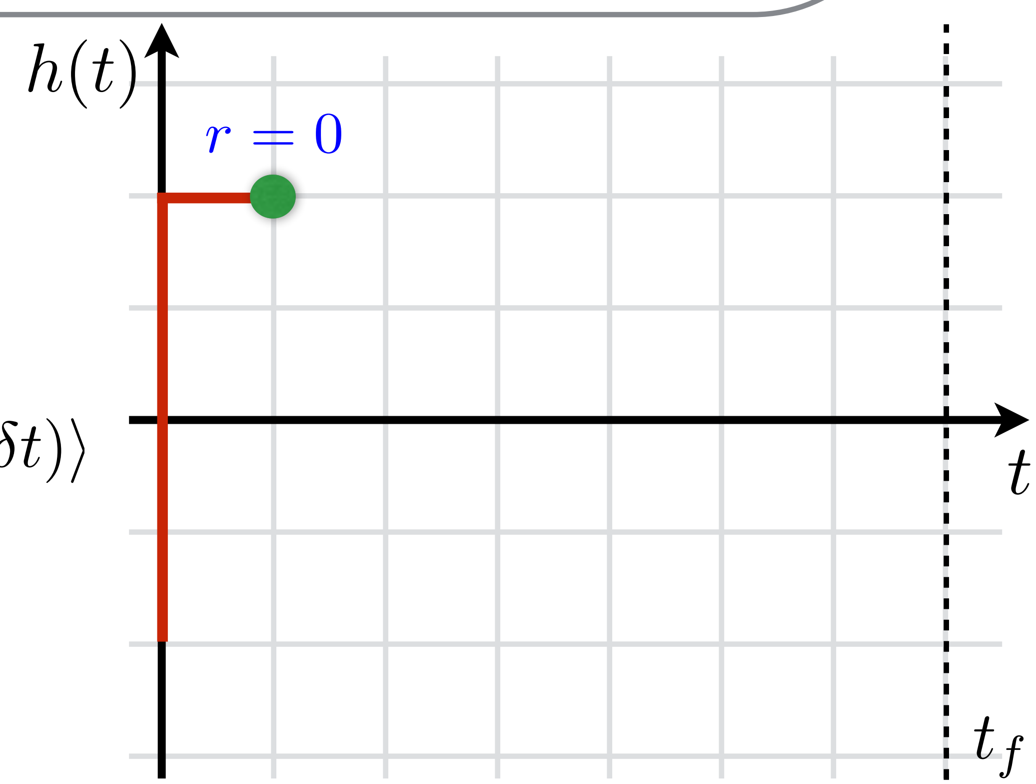
3

reward $r = \begin{cases} 0 & , 0 \leq t < t_f \\ |\langle\psi_*|\psi(t = t_f)\rangle|^2 & , t = t_f \end{cases}$

1 start from state $s_0 = [h(0)] = [-4]$
take action $a_0 : \delta h = +4$
go to state $s_1 = [h(0), h(\delta t)] = [-4, +4]$

2 solve Schrödinger Eq. and obtain the QM state $|\psi(\delta t)\rangle$

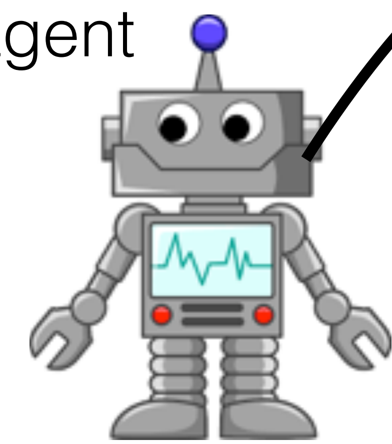
3 calculate reward r
and use it to update $Q(s, a)$
which in turn is used to choose subsequent actions



Marin Bukov

RL Applied to Quantum State Preparation

agent



(1)

feedback loop,
updates $Q(s,a)$

(3)

ENVIRONMENT

$$H(t) = H_0 + H_{dr}(t) + H_{co'l}(t)$$

(2)

$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

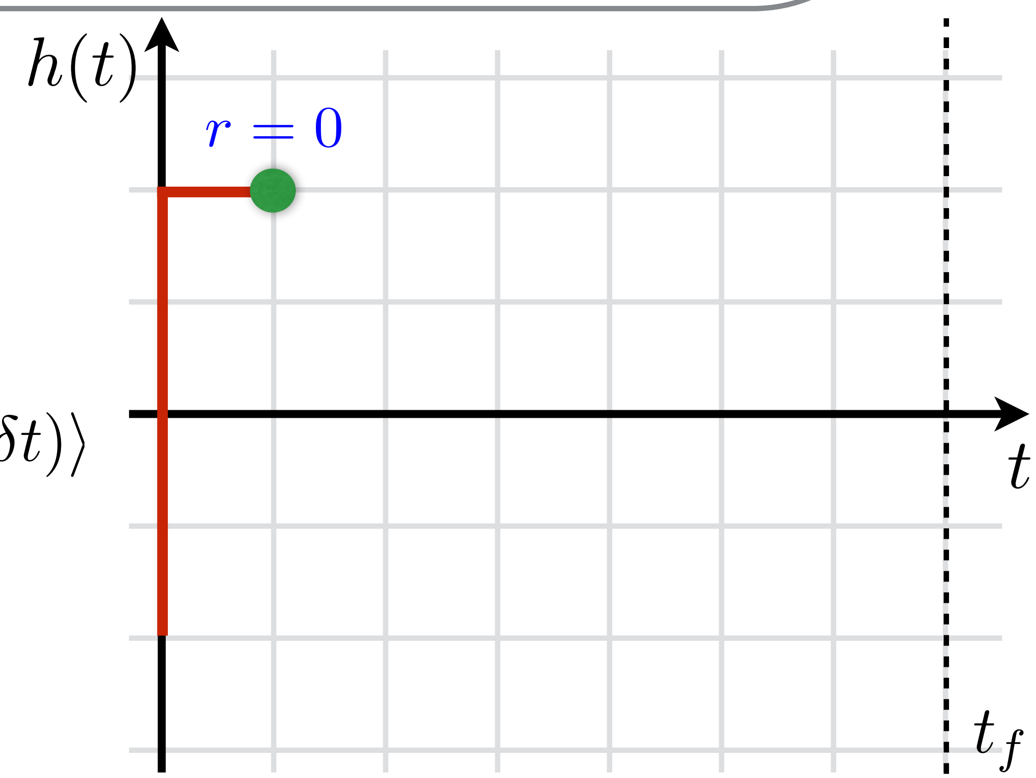
$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t \in [0, t_f]$

reward $r = \begin{cases} 0 & , 0 \leq t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$

(1) start from state $s_0 = [h(0)] = [-4]$
take action $a_0 : \delta h = +4$
go to state $s_1 = [h(0), h(\delta t)] = [-4, +4]$

(2) solve Schrödinger Eq. and obtain the QM state $|\psi(\delta t)\rangle$

(3) calculate reward r
and use it to update $Q(s,a)$
which in turn is used to choose subsequent actions

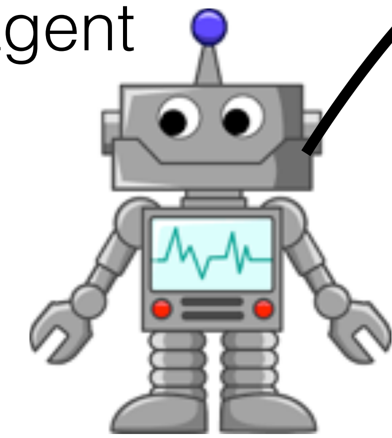


Marin Bukov

Berkeley UNIVERSITY OF CALIFORNIA

RL Applied to Quantum State Preparation

agent



feedback loop, updates $Q(s, a)$

1 → **ENVIRONMENT**

2

$$H(t) = H_0 + H_{\mathrm{dr}}(t) + H_{\mathrm{co'l}}(t)$$

$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

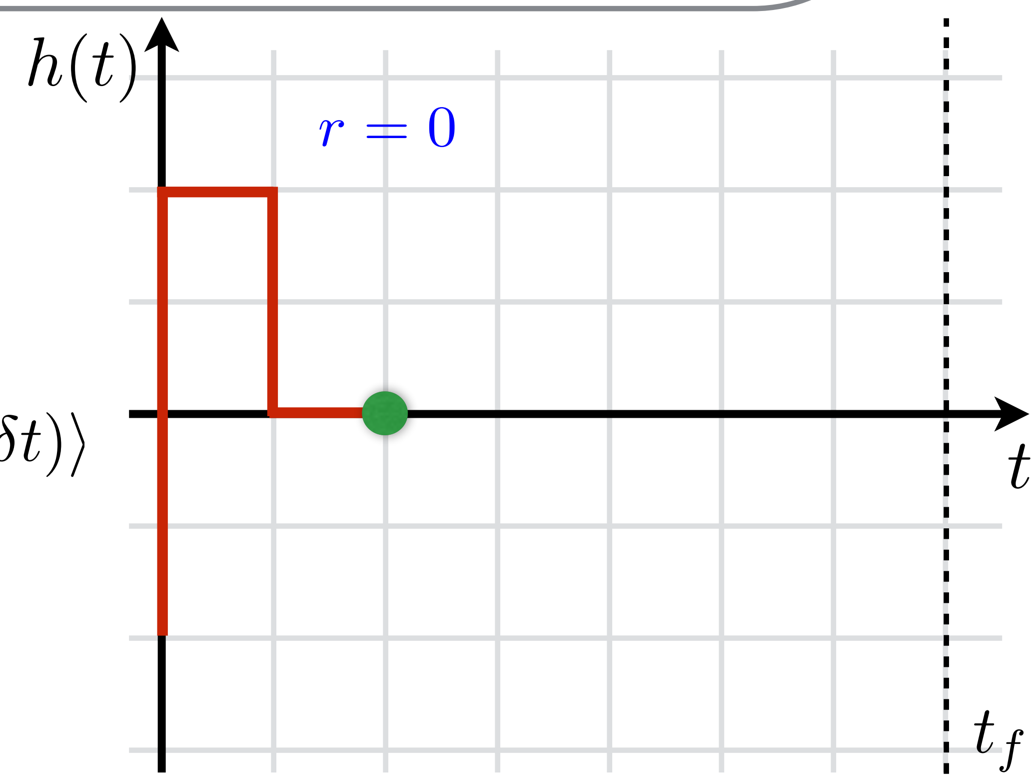
$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t \in [0, t_f]$$

3 ← reward $r = \begin{cases} 0 & , 0 \leq t < t_f \\ |\langle\psi_*|\psi(t = t_f)\rangle|^2 & , t = t_f \end{cases}$

1. start from state $s_0 = [h(0)] = [-4]$
   take action $a_0 : \delta h = +4$
   go to state $s_1 = [h(0), h(\delta t)] = [-4, +4]$

2. solve Schrödinger Eq. and obtain the QM state $|\psi(\delta t)\rangle$

3. calculate reward r
   and use it to update $Q(s, a)$
   which in turn is used to choose subsequent actions

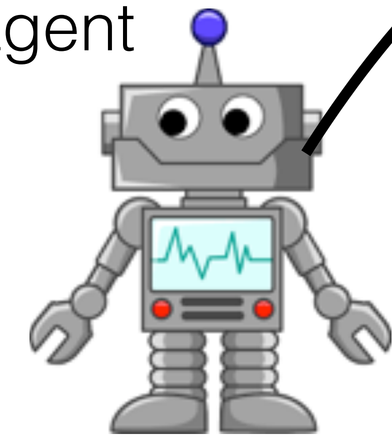


Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

RL Applied to Quantum State Preparation

agent



1

feedback loop,
updates $Q(s,a)$

3

ENVIRONMENT

2

$$H(t) = H_0 + H_{\mathrm{dr}}(t) + H_{\mathrm{co'l}}(t)$$

$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

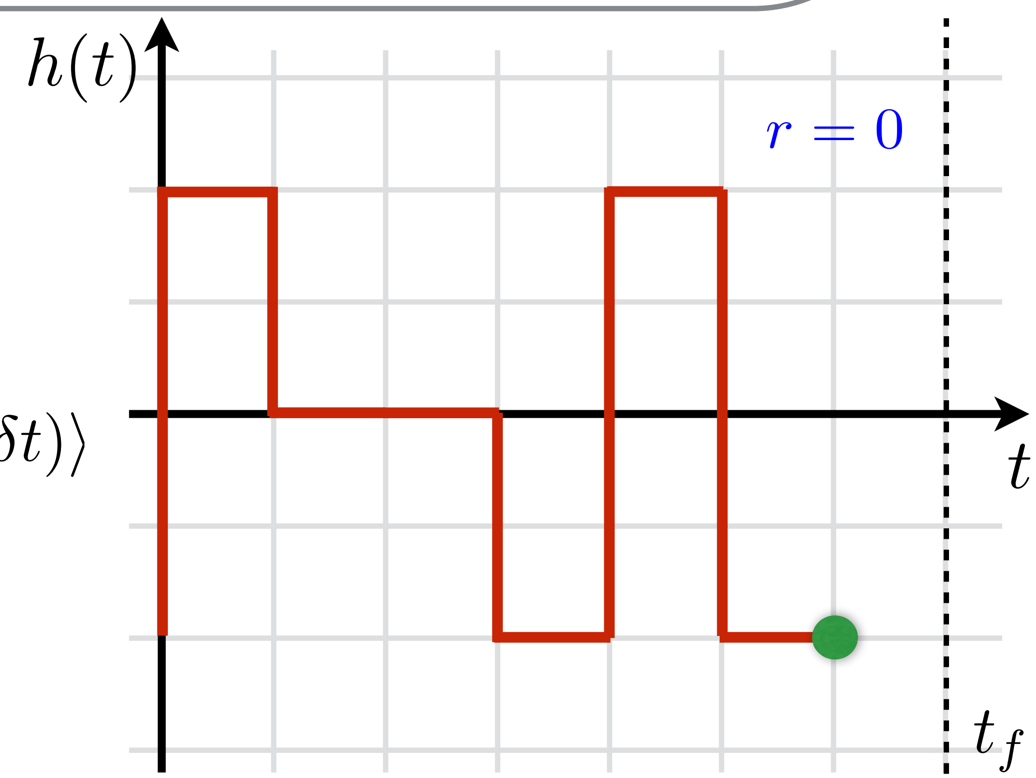
$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t \in [0, t_f]$$

reward $r = \begin{cases} 0 & , 0 \le t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$

1 start from state $s_0 = [h(0)] = [-4]$
take action $a_0 : \delta h = +4$
go to state $s_1 = [h(0), h(\delta t)] = [-4, +4]$

2 solve Schrödinger Eq. and obtain the QM state $|\psi(\delta t)\rangle$

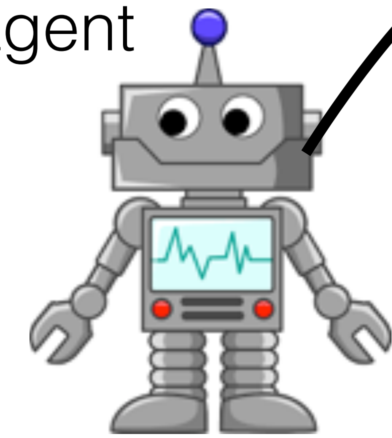
3 calculate reward r
and use it to update $Q(s,a)$
which in turn is used to choose subsequent actions



Marin Bukov

RL Applied to Quantum State Preparation

agent



1

feedback loop, updates $Q(s,a)$

ENVIRONMENT

2

$$H(t) = H_0 + H_{\mathrm{dr}}(t) + H_{\mathrm{co'l}}(t)$$

$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t\in[0,t_f]$$

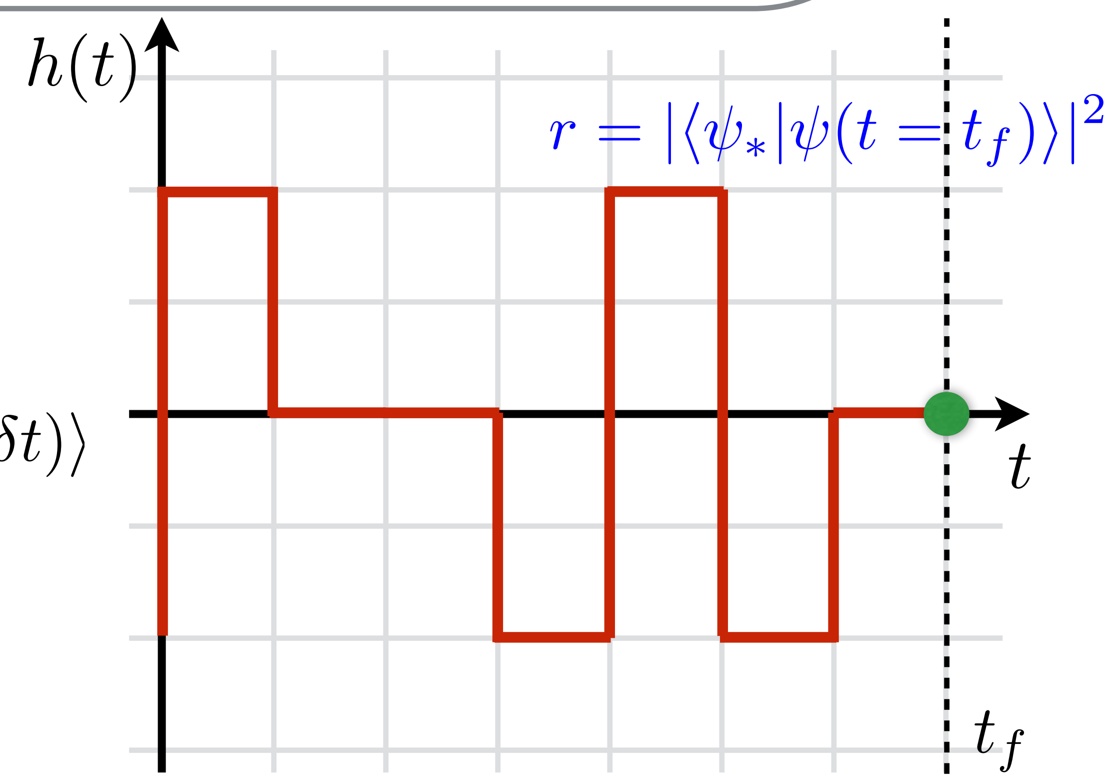
3

reward $r = \begin{cases} 0 & , 0 \le t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$

1. start from state $s_0 = [h(0)] = [-4]$
   take action $a_0 : \delta h = +4$
   go to state $s_1 = [h(0), h(\delta t)] = [-4, +4]$

2. solve Schrödinger Eq. and obtain the QM state $|\psi(\delta t)\rangle$

3. calculate reward r
   and use it to update $Q(s,a)$
   which in turn is used to choose subsequent actions



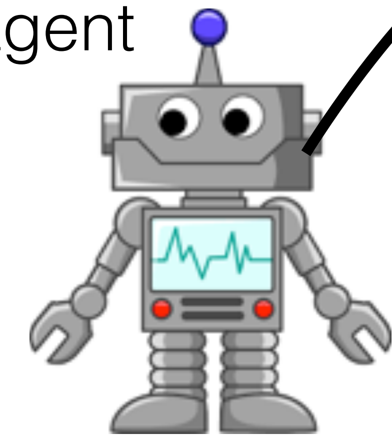
episode completed

Marin Bukov

Berkeley UNIVERSITY OF CALIFORNIA

RL Applied to Quantum State Preparation

agent



1

feedback loop, updates $Q(s,a)$

ENVIRONMENT

2

$$H(t) = H_0 + H_{\text{dr}}(t) + H_{\text{co'l}}(t)$$

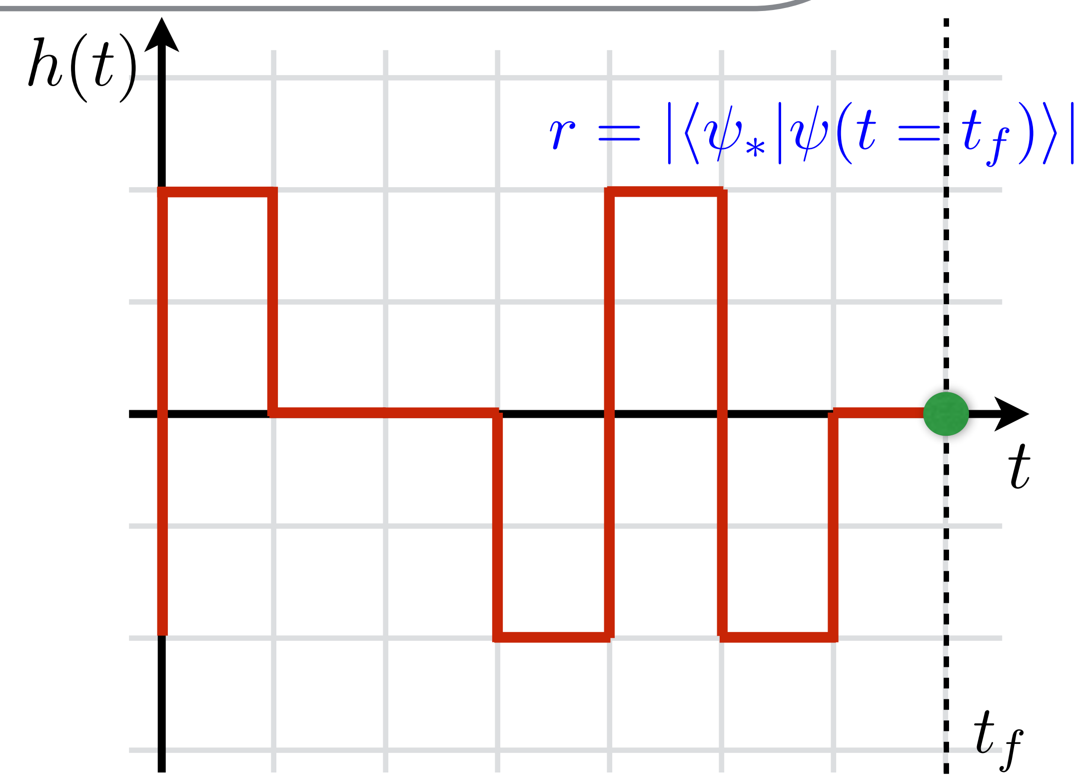
$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \quad t \in [0, t_f]$$

3

reward $r = \begin{cases} 0 & , 0 \leq t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$



problems:

→ state space *exponentially* big

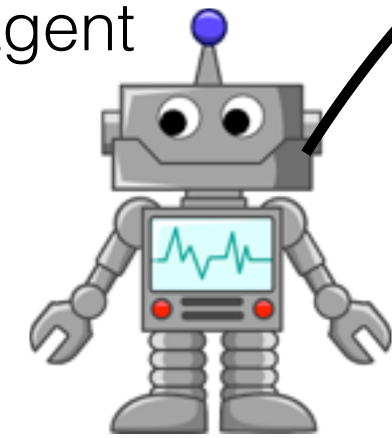
→ how do we choose actions?

Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

RL Applied to Quantum State Preparation

agent



1

feedback loop, updates $Q(s,a)$

ENVIRONMENT

2

$$H(t) = H_0 + H_{\rm dr}(t) + H_{\rm co'l}(t)$$

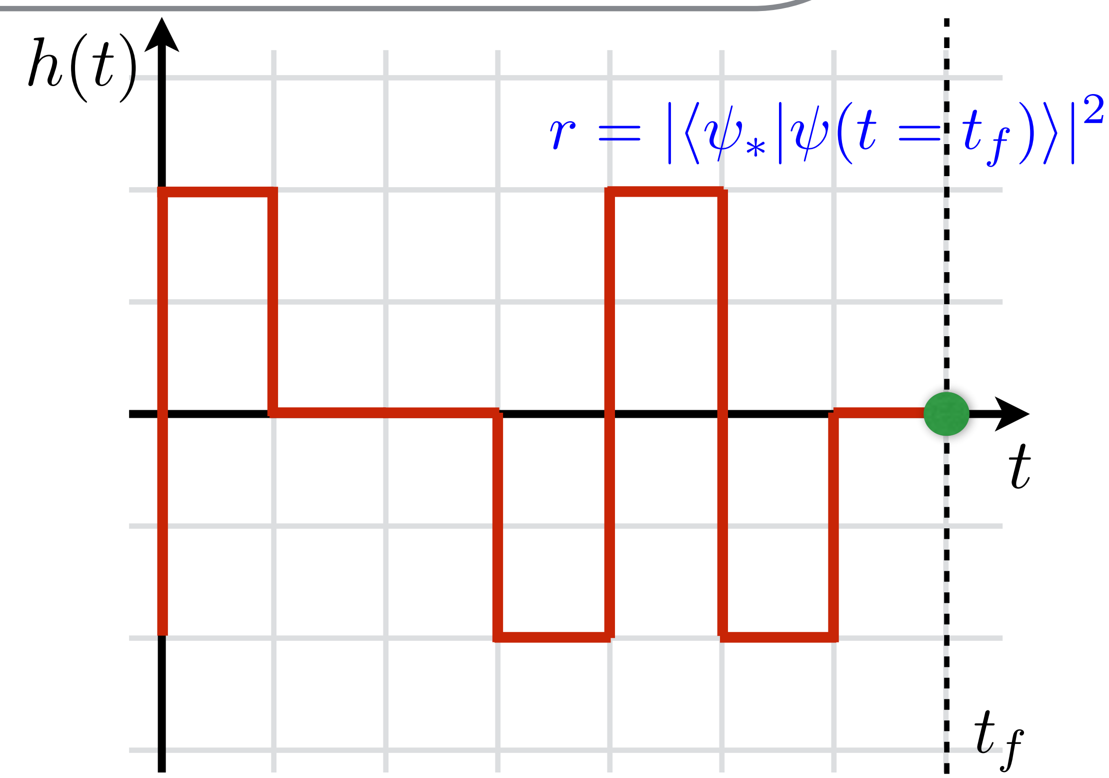
$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \quad t\in[0,t_f]$$

3

reward $r = \begin{cases} 0 & , 0\le t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$



problems:

→ state space *exponentially* big

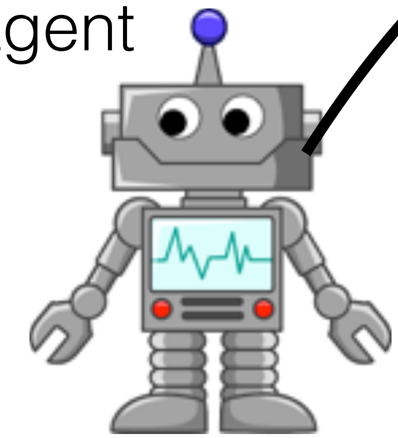
RL ~ biased MC sampling

→ how do we choose actions?

Marin Bukov

RL Applied to Quantum State Preparation

agent



1

feedback loop, updates $Q(s,a)$

ENVIRONMENT

2

$$H(t) = H_0 + H_{\mathrm{dr}}(t) + H_{\mathrm{co'l}}(t)$$

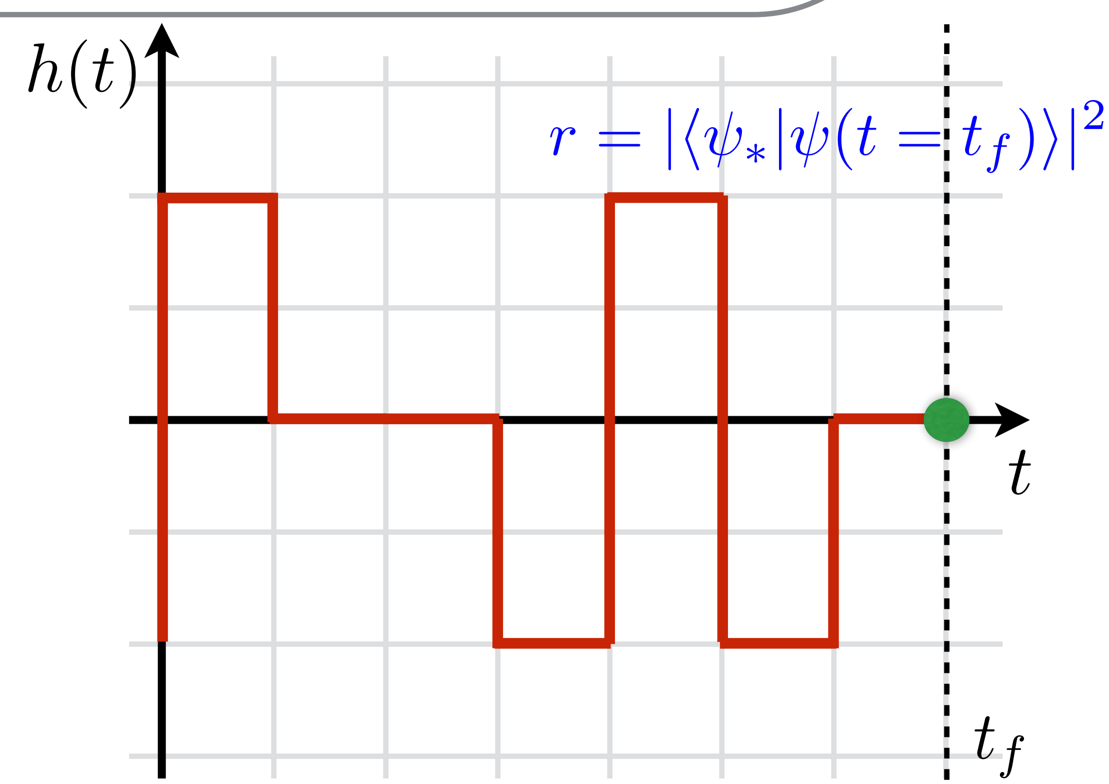
$|\psi_i\rangle$: GS of H_0

$|\psi_*\rangle$: target state

$$i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle \; t\in[0,t_f]$$

3

reward $r = \begin{cases} 0 & , 0 \le t < t_f \\ |\langle\psi_*|\psi(t=t_f)\rangle|^2 & , t = t_f \end{cases}$



problems:

→ state space *exponentially* big

RL ~ biased MC sampling

→ how do we choose actions?

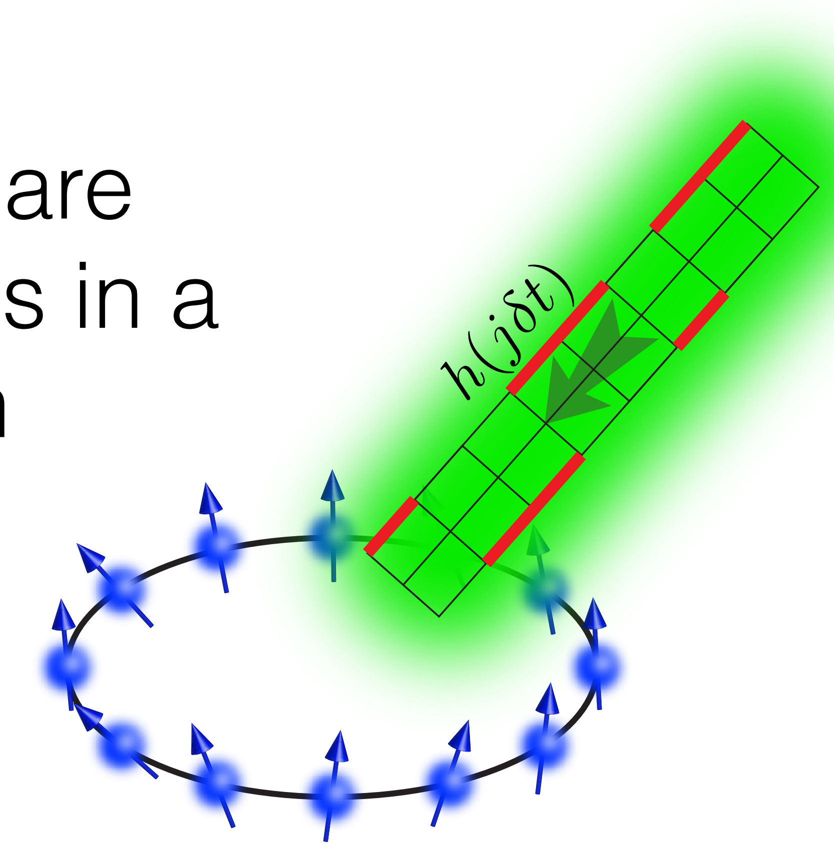
exploration exploitation dilemma

Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

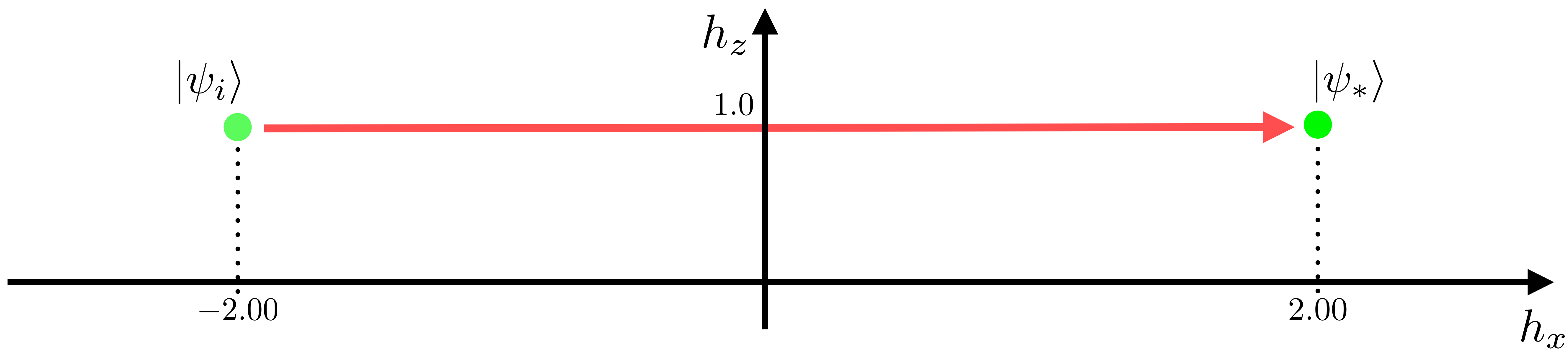
Example 1:

use RL to autonomously prepare paramagnetic many-body states in a **nonintegrable spin chain**



$$H(t) = -\sum_{j=1}^{L} S^z_{j+1}S^z_j + h_z S^z_j + h_x(t) S^x$$

→ initial $|\psi_i\rangle$ and target $|\psi_*\rangle$ states are (paramag) GS at:

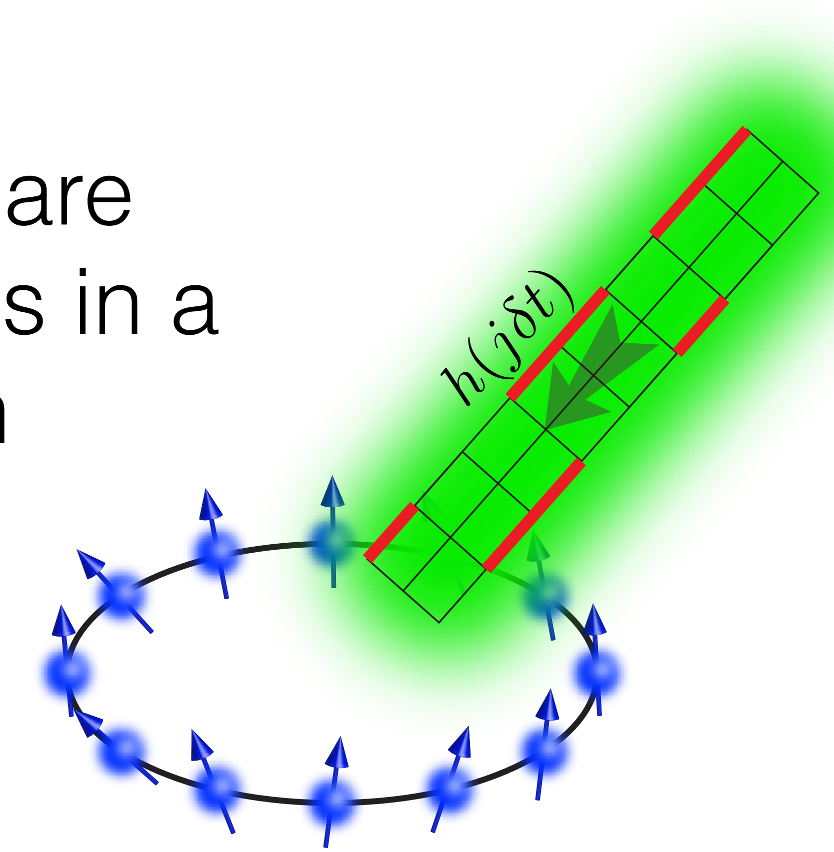


Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

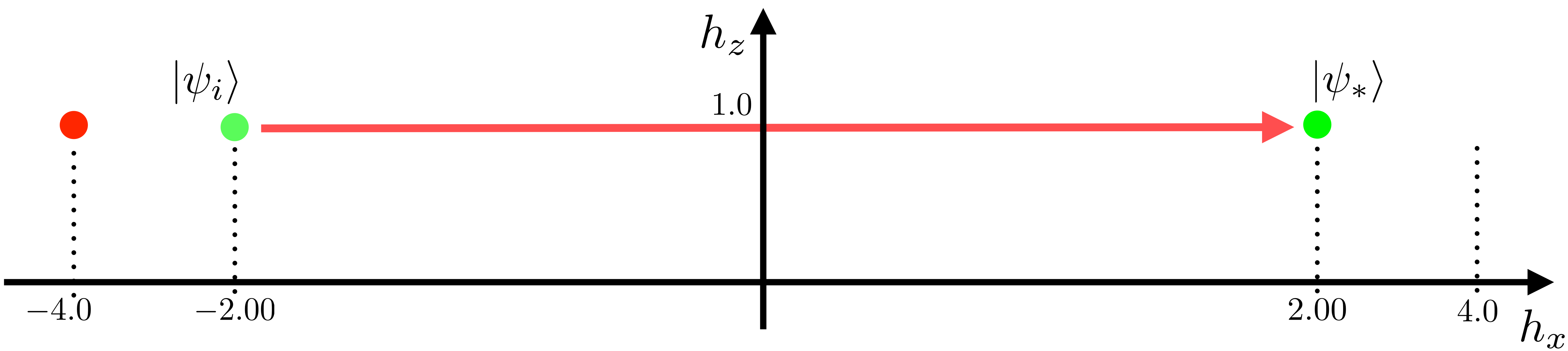
Example 1:

use RL to autonomously prepare paramagnetic many-body states in a **nonintegrable spin chain**



$$H(t) = -\sum_{j=1}^{L} S^z_{j+1}S^z_j + h_z S^z_j + h_x(t) S^x$$

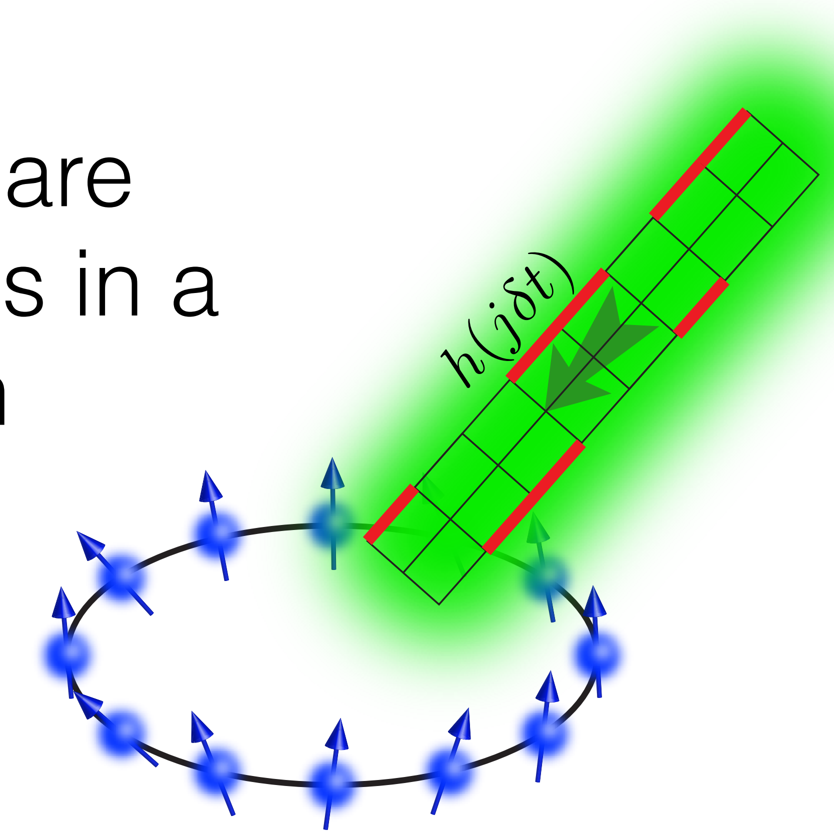
→ initial $|\psi_i\rangle$ and target $|\psi_*\rangle$ states are (paramag) GS at:



Berkeley
UNIVERSITY OF CALIFORNIA

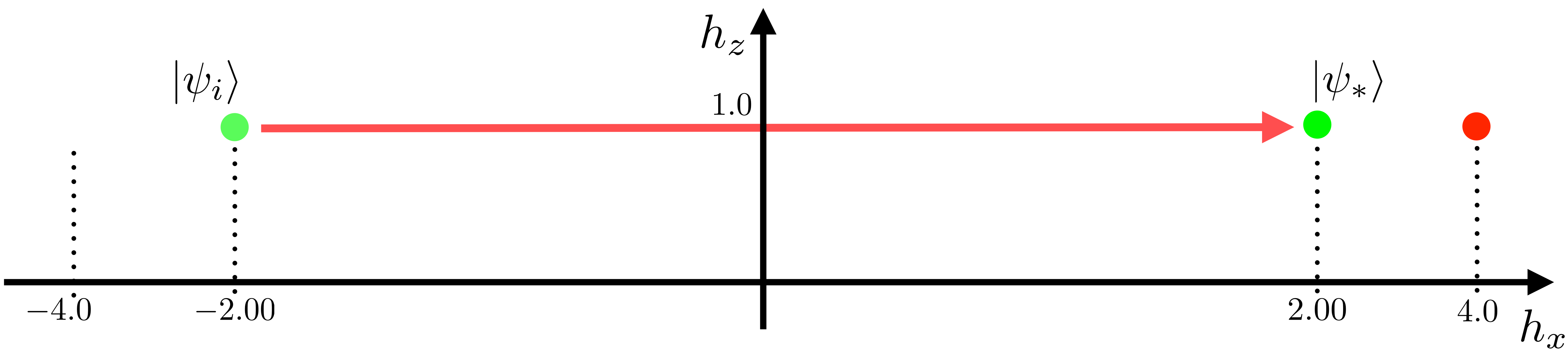
Example 1:

use RL to autonomously prepare paramagnetic many-body states in a **nonintegrable spin chain**



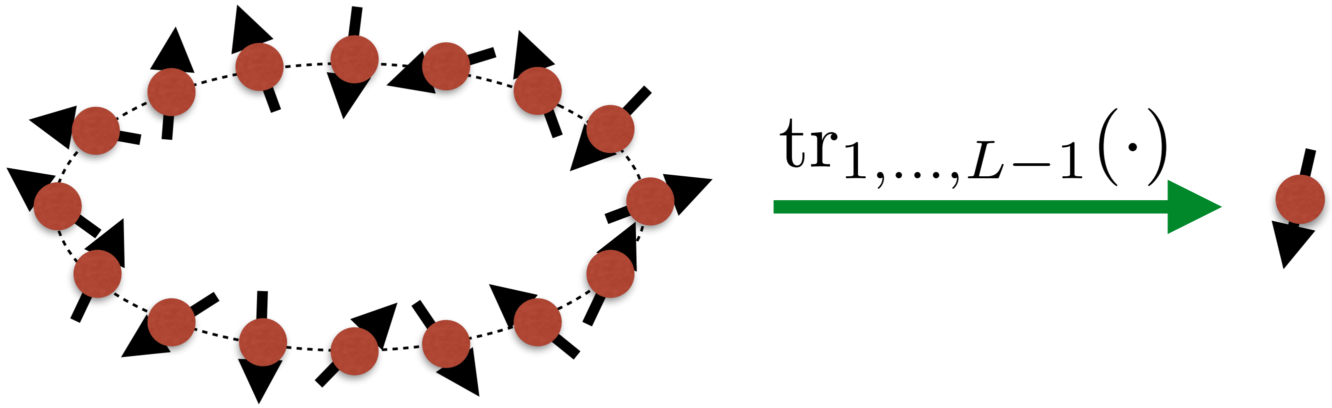
$$H(t) = -\sum_{j=1}^{L} S^z_{j+1} S^z_j + h_z S^z_j + h_x(t) S^x$$

→ initial $|\psi_i\rangle$ and target $|\psi_*\rangle$ states are (paramag) GS at:

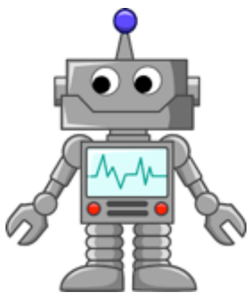


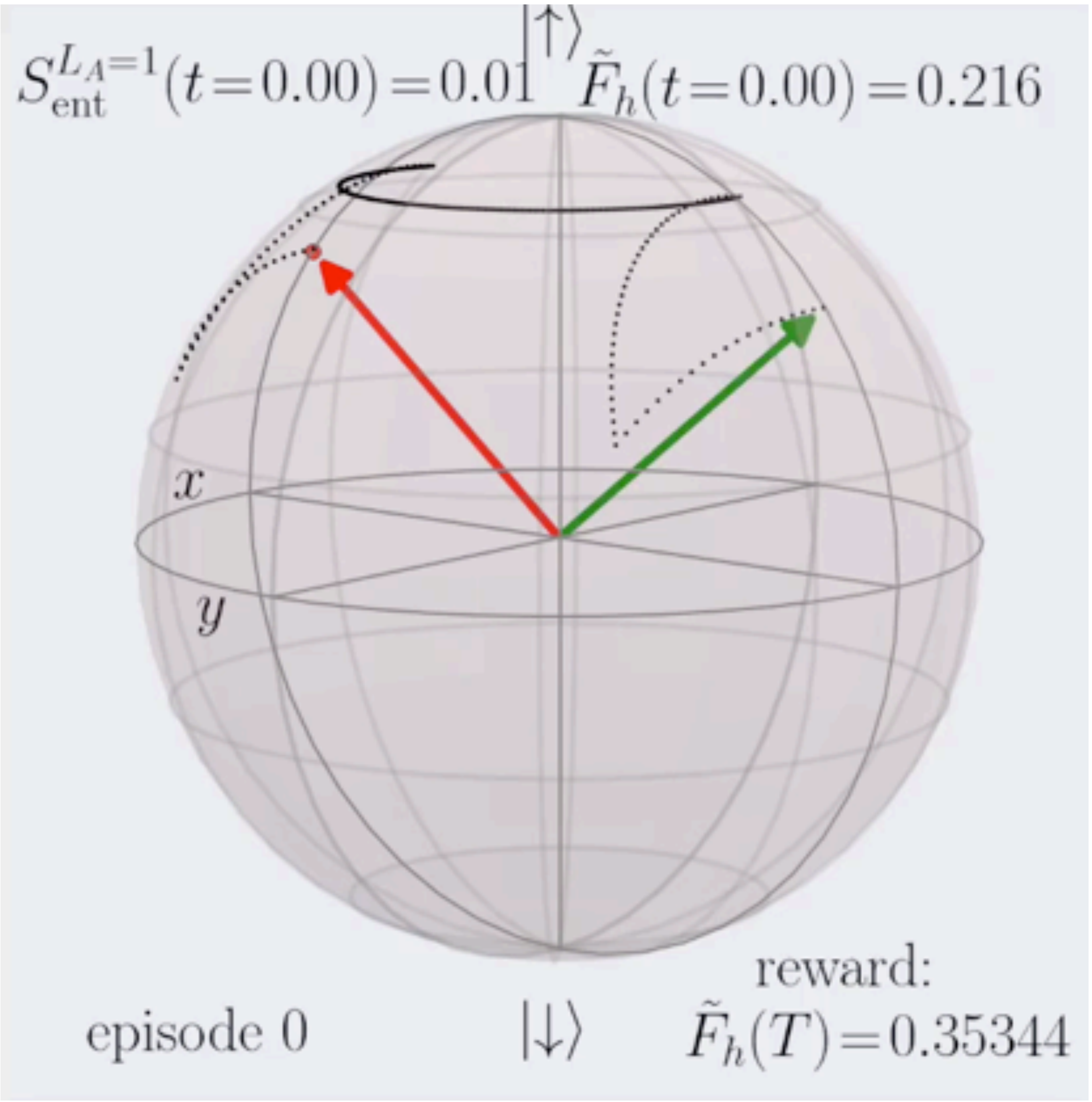
Marin Bukov

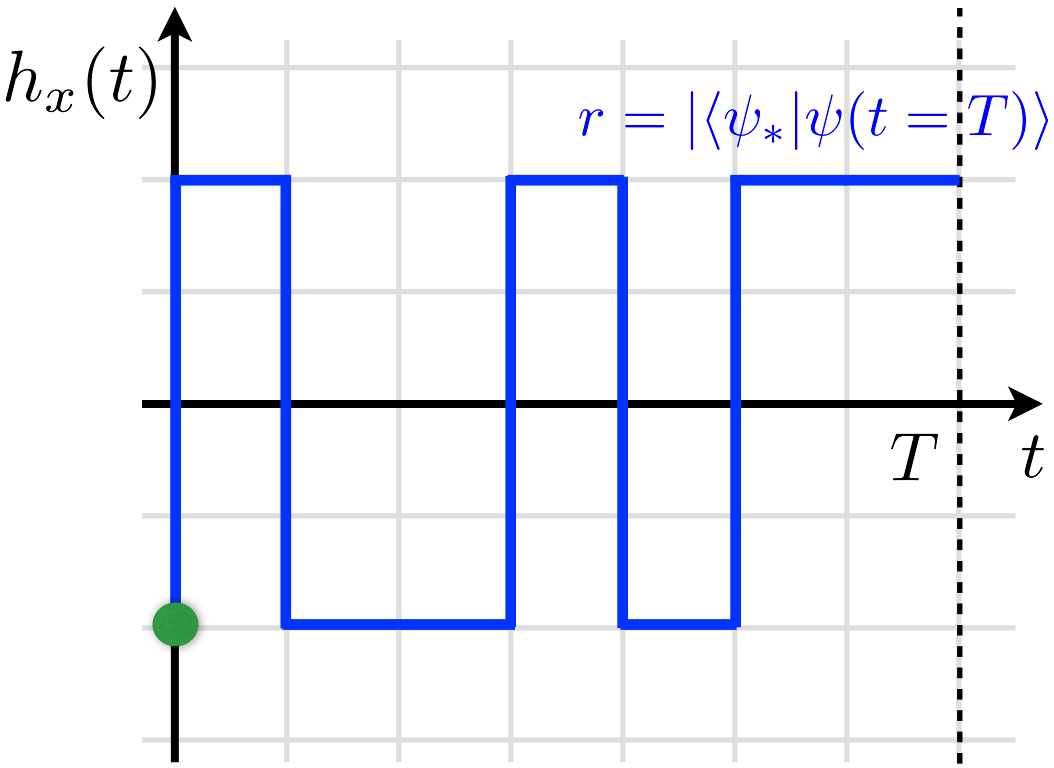
Learning a Many-Body Protocol



$h_x \in \{\pm 4\}$ bang-bang protocols





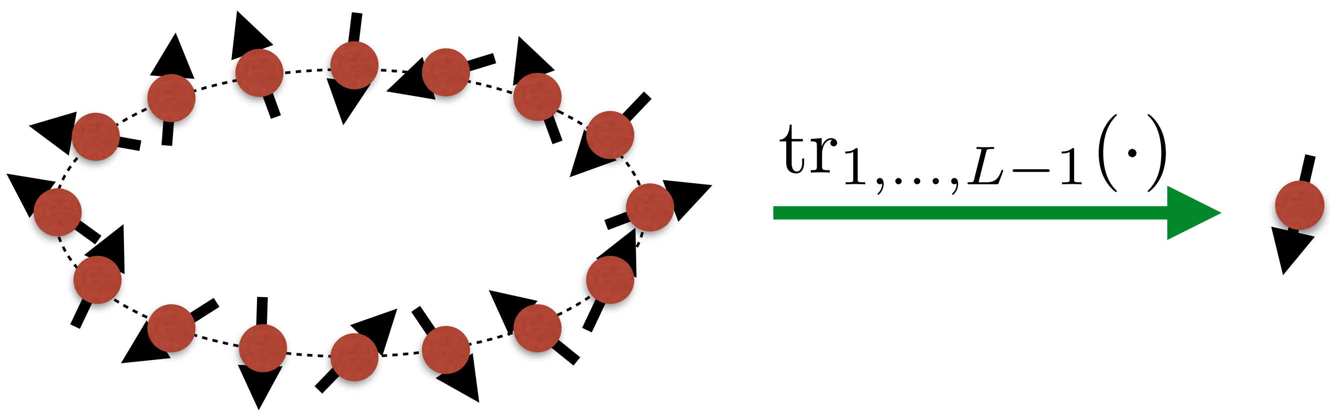


episode completed

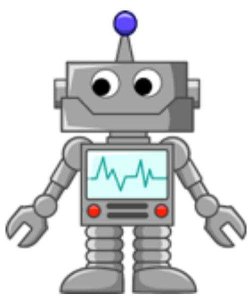
$$H = \sum -S^z_{j+1}S^z_j - h_z S^z_j - h_x(t) S^x_j$$

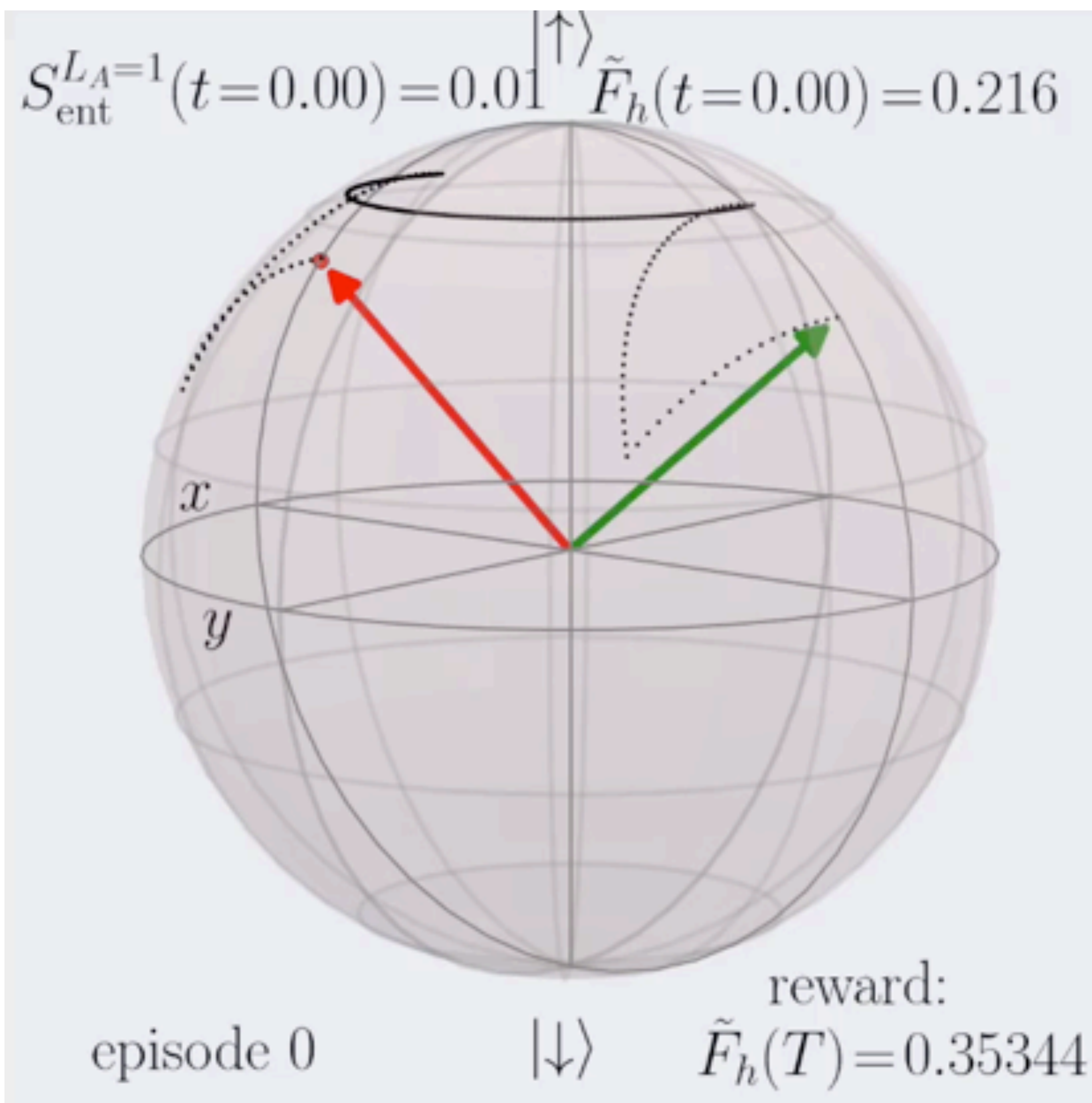
arXiv: *1705.00565* (2017)

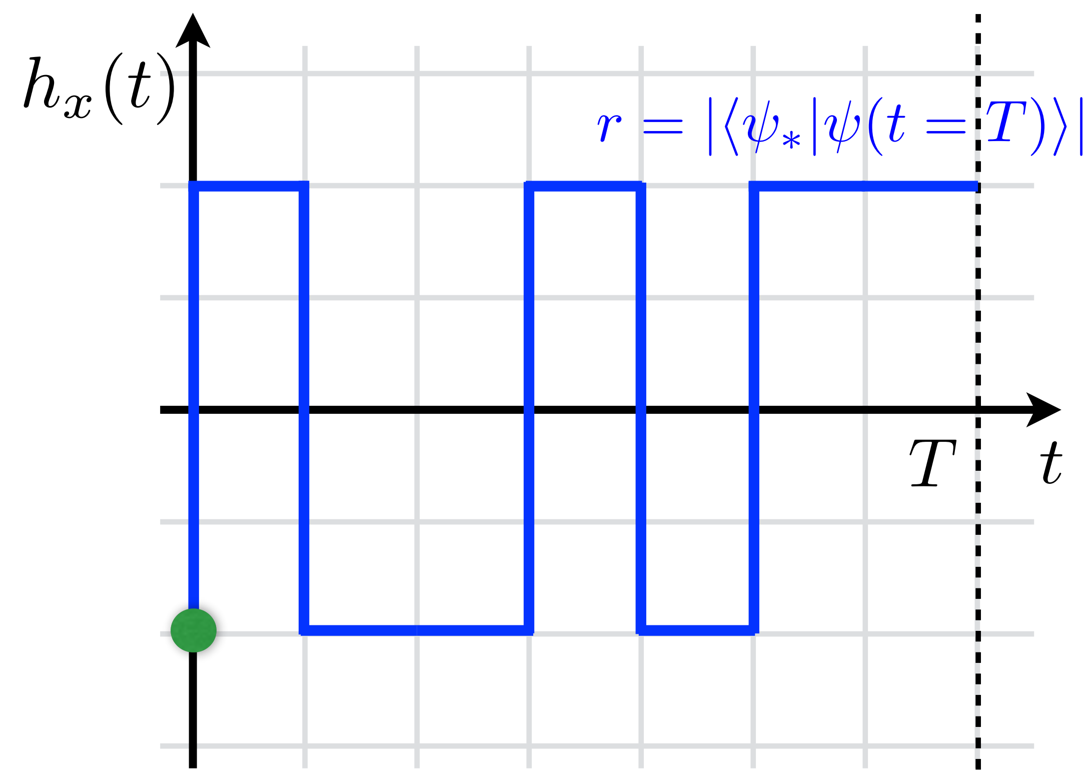
Learning a Many-Body Protocol



$h_x \in \{\pm 4\}$ bang-bang protocols







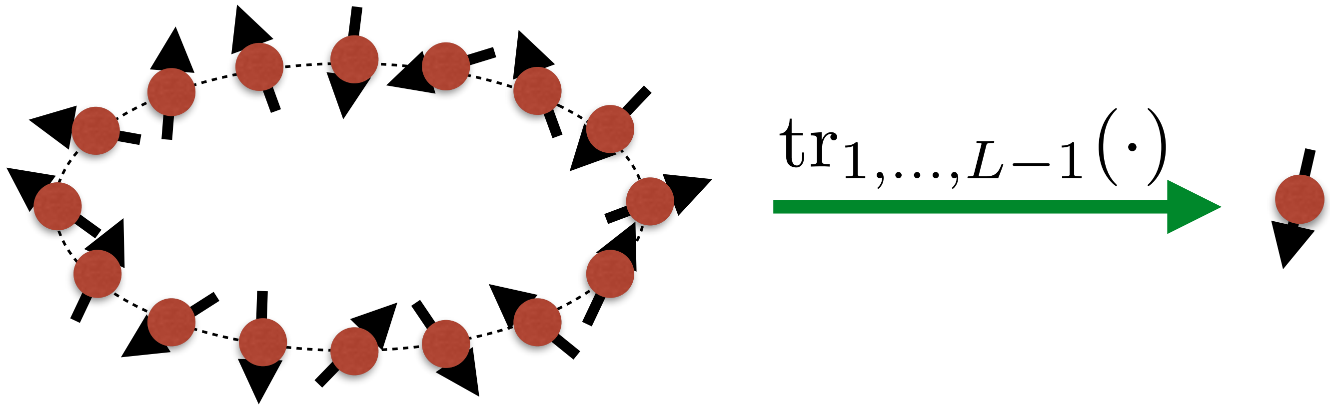
episode completed

$$H = \sum -S^z_{j+1}S^z_j - h_z S^z_j - h_x(t) S^x_j$$

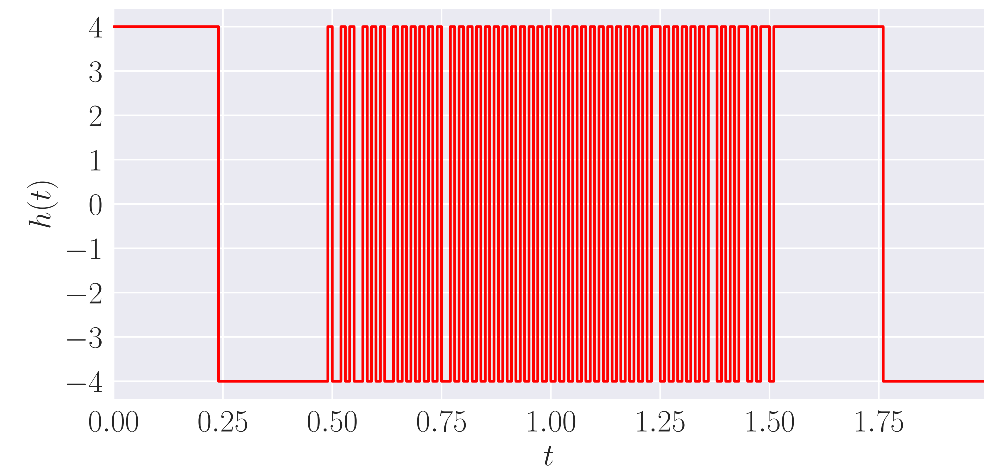
arXiv: *1705.00565* (2017)

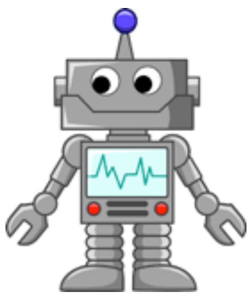
Learning a Many-Body Protocol

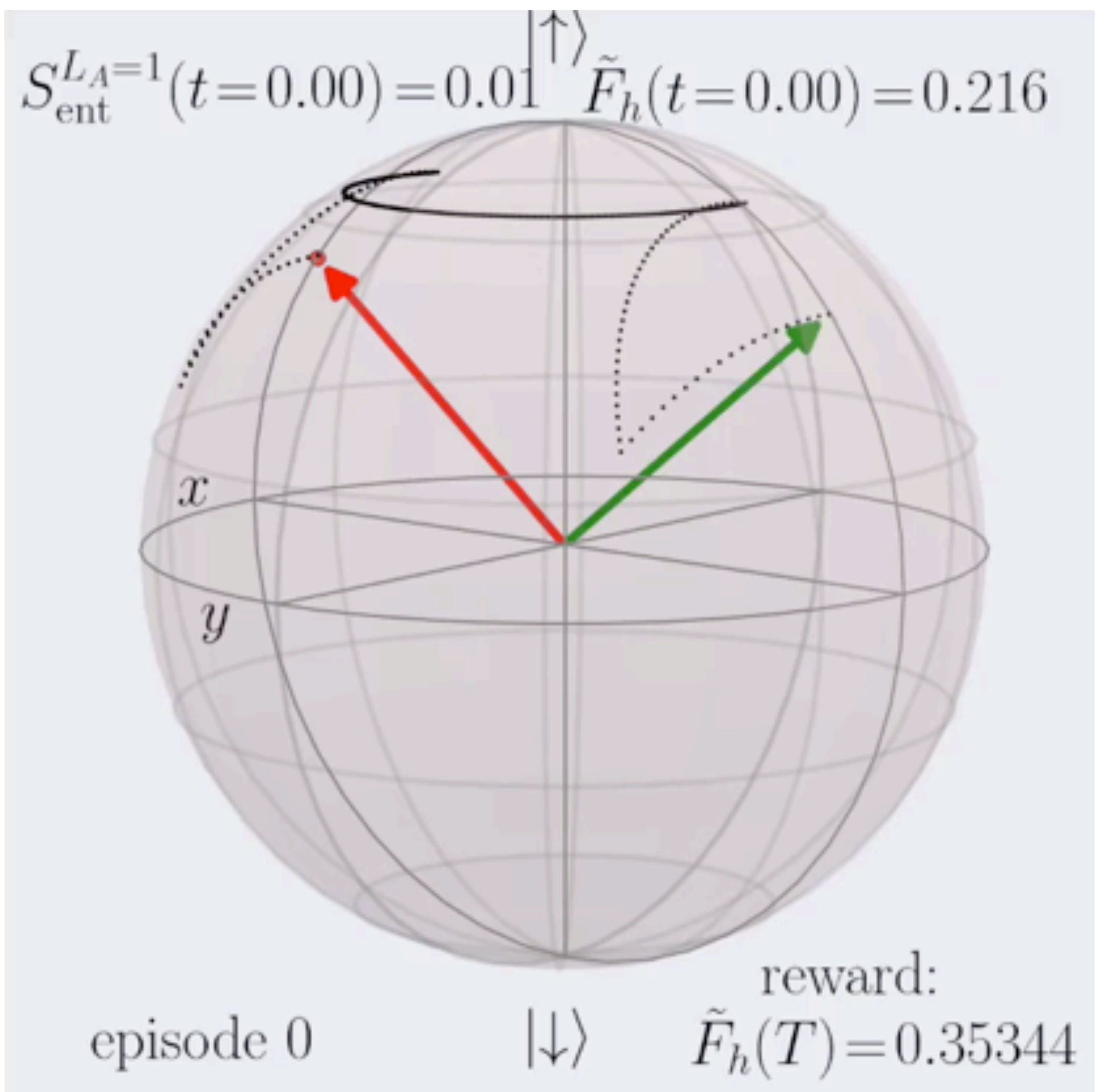
Berkeley UNIVERSITY OF CALIFORNIA

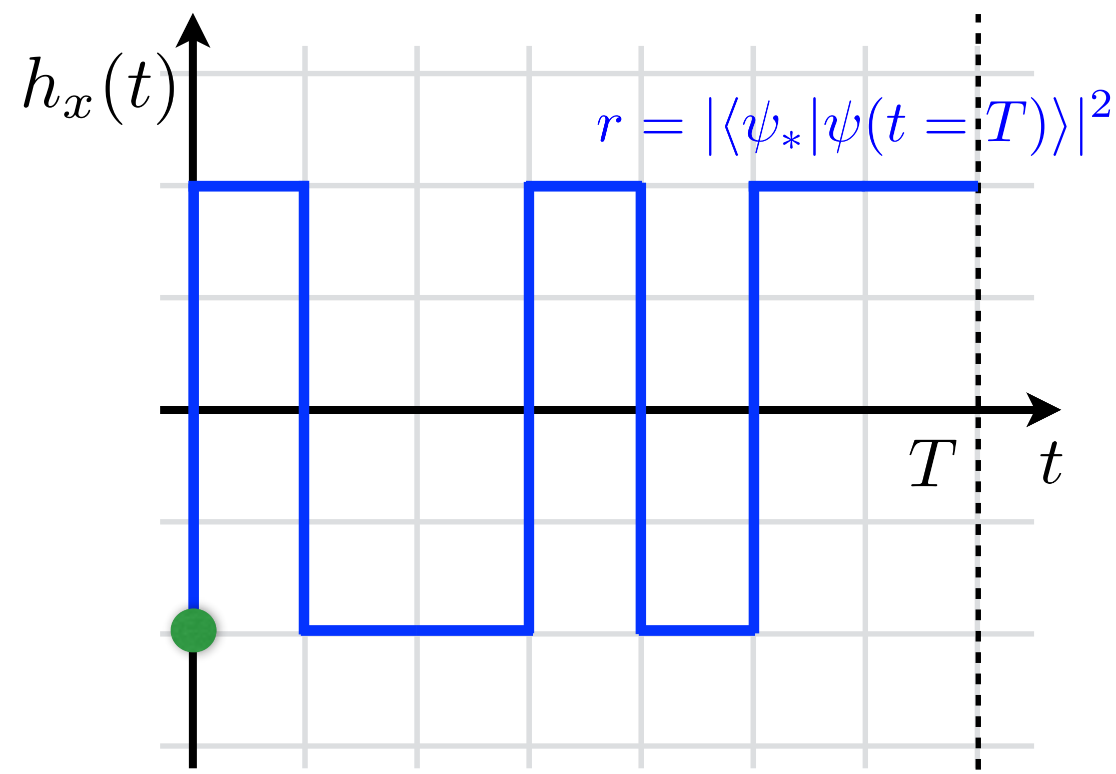


$h_x \in \{\pm 4\}$ bang-bang protocols









episode completed

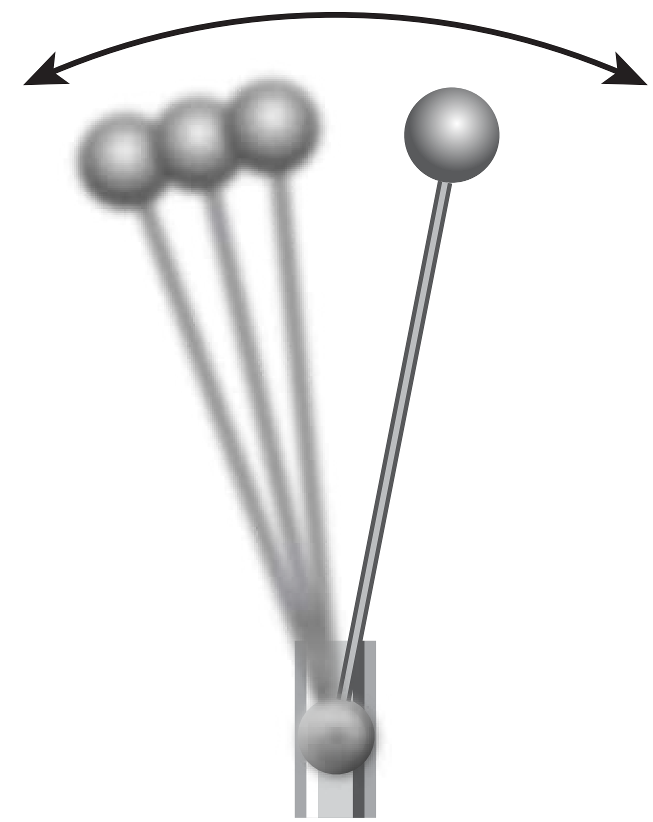
$$H = \sum -S^z_{j+1}S^z_j - h_z S^z_j - h_x(t)S^x_j$$

arXiv: *1705.00565* (2017)

Example 2:

use RL to autonomously prepare
Floquet engineered states in a ***simulation of an "experiment"***

Challenges:



→ **no** direct **access** to quantum state:
play game w/o looking at screen

→ **probabilistic** quantum measurements

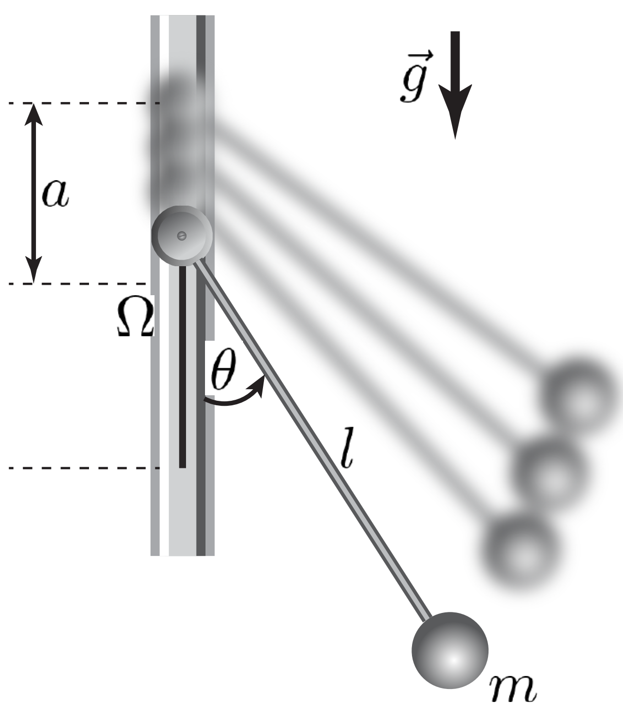
→ **uncertainty** in preparing initial state

→ occasional **failure** of control apparatus

→ additionally: all other problems of how to actually prepare the state if the above were absent and no analytics is known

The Kapitza pendulum

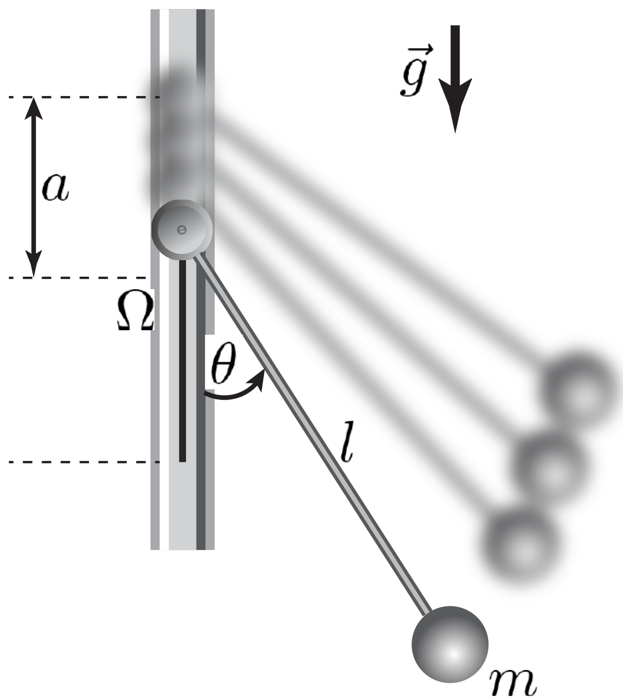
- Kapitza, 1951
- paradigmatic example of Floquet engineering



bluedwarf1127: YouTube (2009)

The Kapitza pendulum

- Kapitza, 1951
- paradigmatic example of Floquet engineering



bluedwarf1127: YouTube (2009)

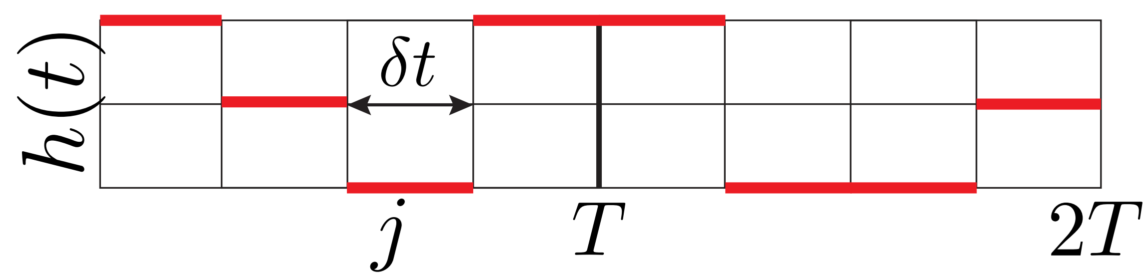
Floquet Engineering Control Problem

→ find optimal control field *on top of periodic drive*

$$H_{\rm rot}(t) = H_0 + H_{\rm drive}(t) + H_{\rm control}(t)$$

$$H_{\rm drive}(t) = -\frac{A}{2m}\mathrm{sign}(\cos\Omega t)[p_\theta, \sin\theta]_+ + \frac{A^2}{4}\left(1-\mathrm{sign}(\sin\Omega t)\right)\cos 2\theta$$

$$H_0 = \frac{p_\theta^2}{2m} - m\omega_0^2\cos\theta \qquad H_{\rm control}(t) = h(t)\sin\theta$$ horizontal kicks



Marin Bukov

Floquet Engineering Control Problem

→ find optimal control field *on top of periodic drive*

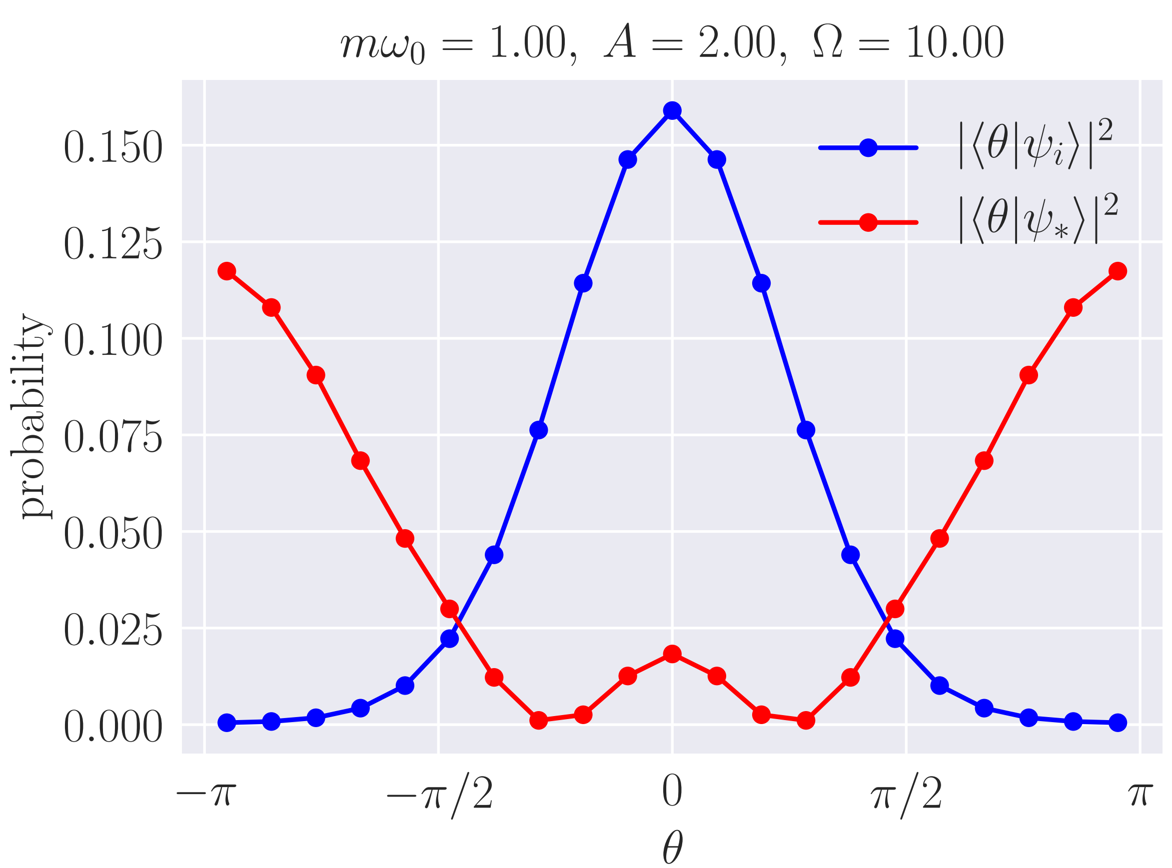
$$H_{\rm rot}(t) = H_0 + H_{\rm drive}(t) + H_{\rm control}(t)$$

$$H_{\rm drive}(t) = -\frac{A}{2m}\mathrm{sign}(\cos\Omega t)[p_\theta, \sin\theta]_+ + \frac{A^2}{4}\left(1-\mathrm{sign}(\sin\Omega t)\right)\cos 2\theta$$

$$H_0 = \frac{p_\theta^2}{2m} - m\omega_0^2\cos\theta \qquad H_{\rm control}(t) = h(t)\sin\theta$$ horizontal kicks

initial state: $|\psi_i\rangle$: GS of H_0

target state: $|\psi_*\rangle$ inverted position eigenstate of $H_F(\Omega)$



Marin Bukov

Floquet Engineering Control Problem

→ find optimal control field *on top of periodic drive*

$$H_{\rm rot}(t) = H_0 + H_{\rm drive}(t) + H_{\rm control}(t)$$

$$H_{\rm drive}(t) = -\frac{A}{2m}\mathrm{sign}(\cos\Omega t)[p_\theta, \sin\theta]_+ + \frac{A^2}{4}\left(1-\mathrm{sign}(\sin\Omega t)\right)\cos 2\theta$$

$$H_0 = \frac{p_\theta^2}{2m} - m\omega_0^2\cos\theta \qquad H_{\rm control}(t) = h(t)\sin\theta$$ horizontal kicks

initial state: $|\psi_i\rangle$: GS of H_0

target state: $|\psi_*\rangle$ inverted position eigenstate of $H_F(\Omega)$

GOAL: find bang-bang protocol $h(t) = h(j\delta t) \in \{-4, 0, +4\}$

such that $|\psi(t=0)\rangle = |\psi_i\rangle, |\psi(t=t_f)\rangle = |\psi_*\rangle$

Marin Bukov

Berkeley UNIVERSITY OF CALIFORNIA

Floquet Engineering Control Problem

→ find optimal control field *on top of periodic drive*

$$H_{\rm rot}(t) = H_0 + H_{\rm drive}(t) + H_{\rm control}(t)$$

$$H_{\rm drive}(t) = -\frac{A}{2m}\text{sign}(\cos\Omega t)[p_\theta, \sin\theta]_+ + \frac{A^2}{4}(1-\text{sign}(\sin\Omega t))\cos 2\theta$$

$$H_0 = \frac{p_\theta^2}{2m} - m\omega_0^2\cos\theta \qquad H_{\rm control}(t) = h(t)\sin\theta \quad \text{horizontal kicks}$$

initial state: $|\psi_i\rangle$: GS of H_0

target state: $|\psi_*\rangle$ inverted position eigenstate of $H_F(\Omega)$

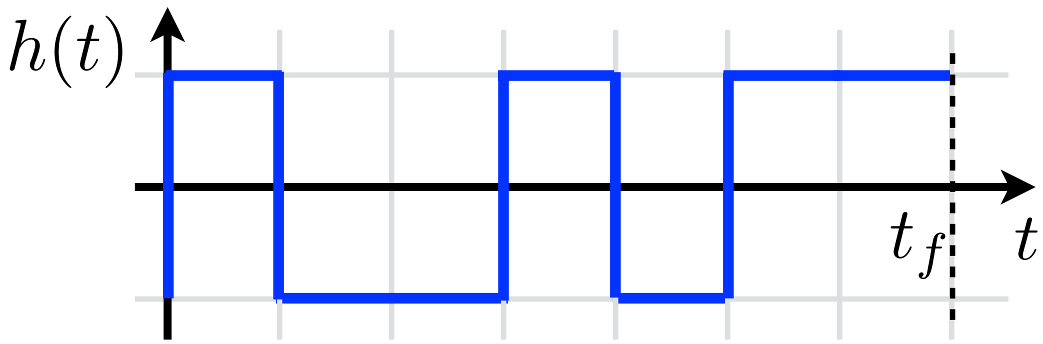
GOAL: find bang-bang protocol $h(t) = h(j\delta t) \in \{-4, 0, +4\}$

such that $|\psi(t=0)\rangle = |\psi_i\rangle, |\psi(t=t_f)\rangle = |\psi_*\rangle$

measure: quantum measurement of final state $|\psi(t=t_f)\rangle$ along target state $|\psi_*\rangle$

Marin Bukov

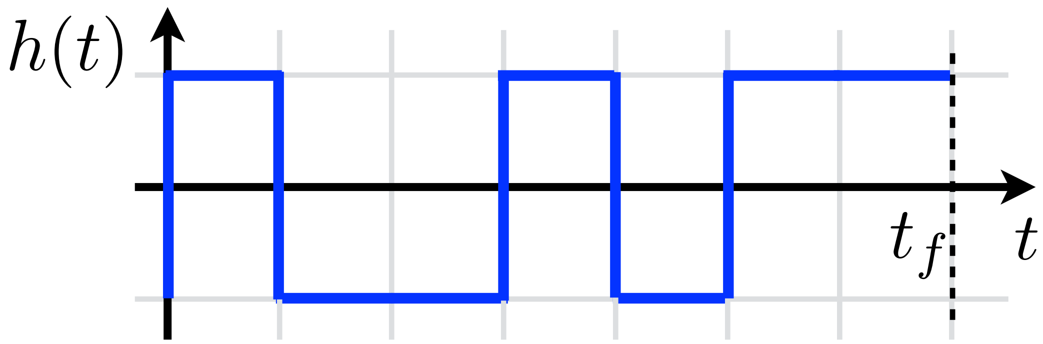
Let's give this game a try!



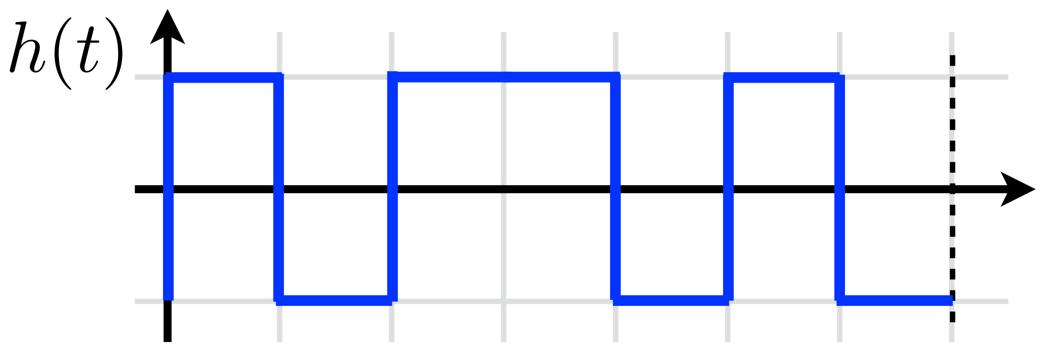
measurement: -1

Marin Bukov

Let's give this game a try!



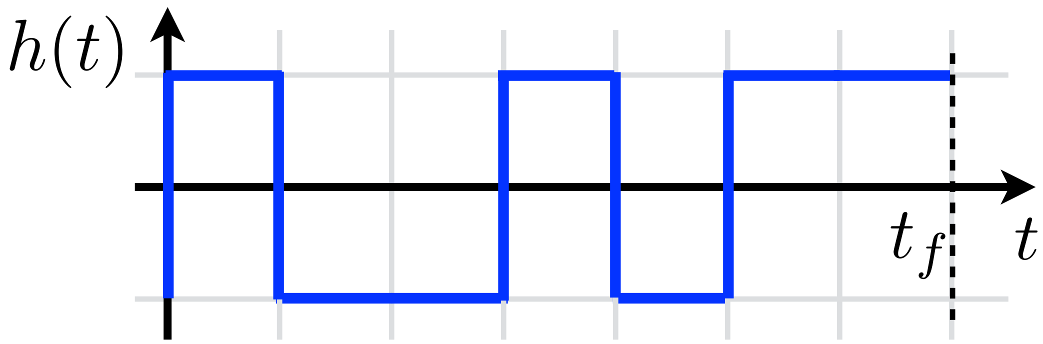
measurement: -1



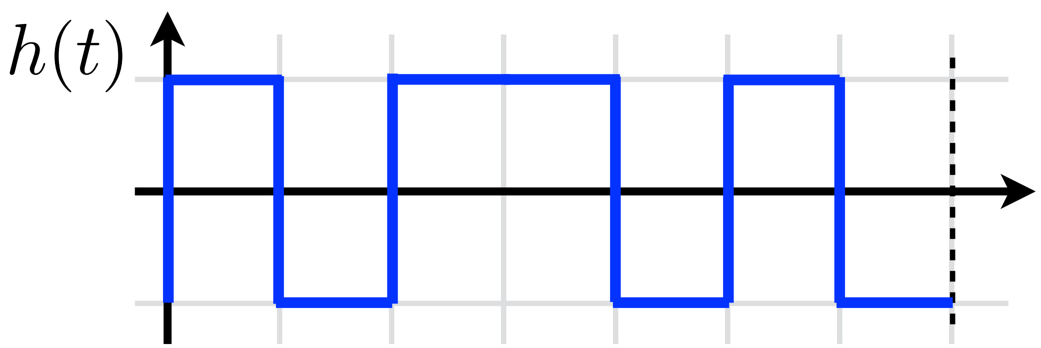
measurement: $+1$

(different final state: different probability to be in the target state)

Let’s give this game a try!



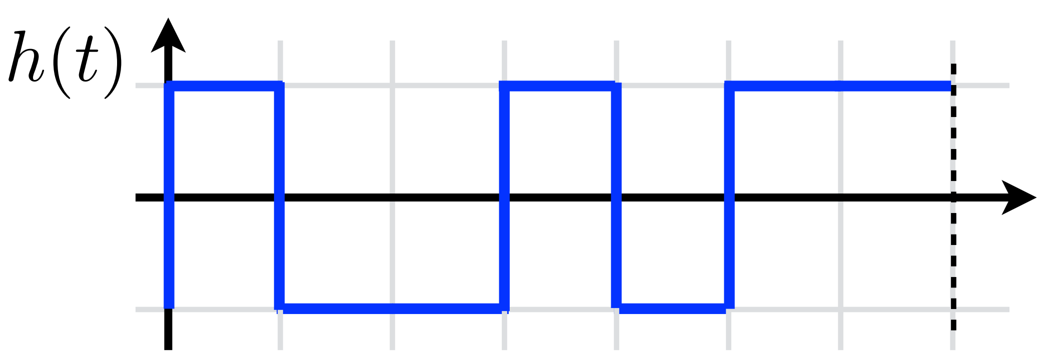
measurement: -1



measurement: $+1$

(different final state: different probability to be in the target state)

→ repeat protocol!

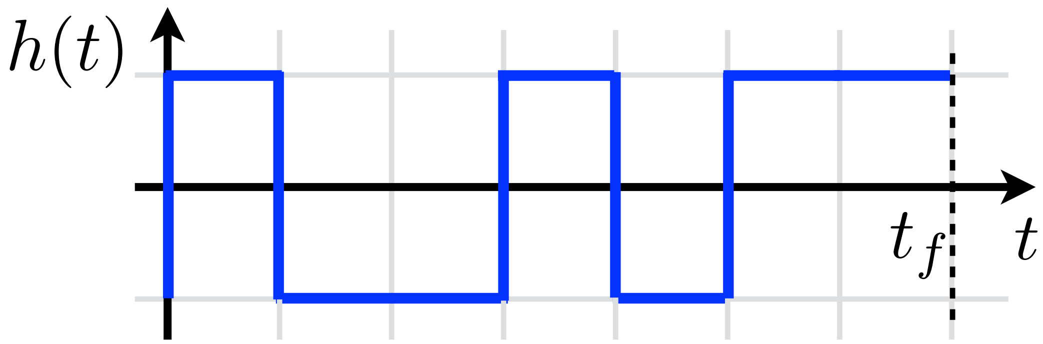


measurement: $+1$

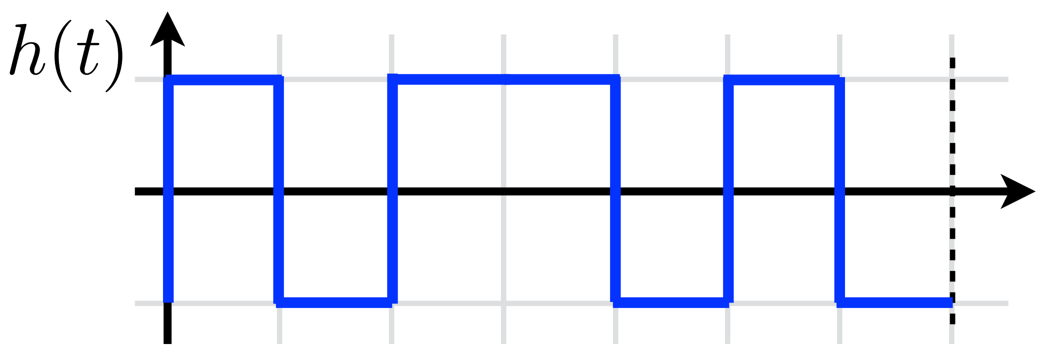
Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

Let's give this game a try!



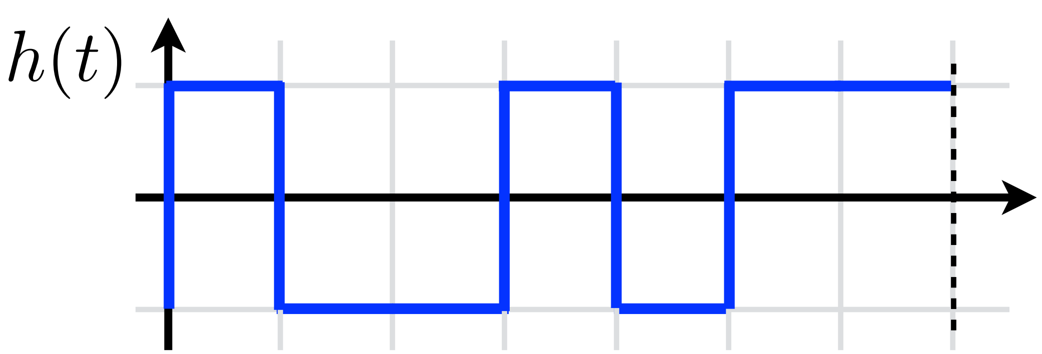
measurement: -1



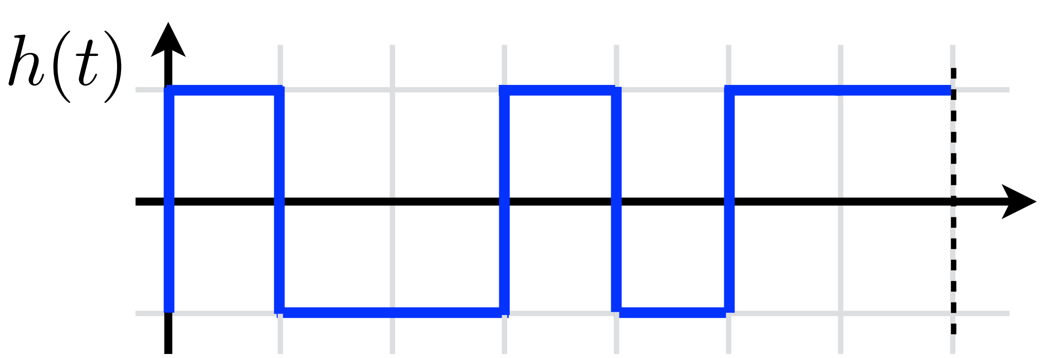
measurement: $+1$

(different final state: different probability to be in the target state)

→ repeat protocol!



measurement: $+1$

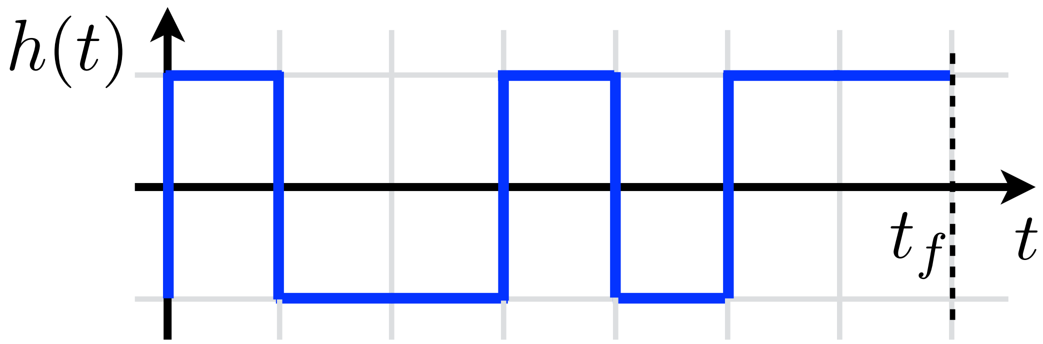


measurement: -1

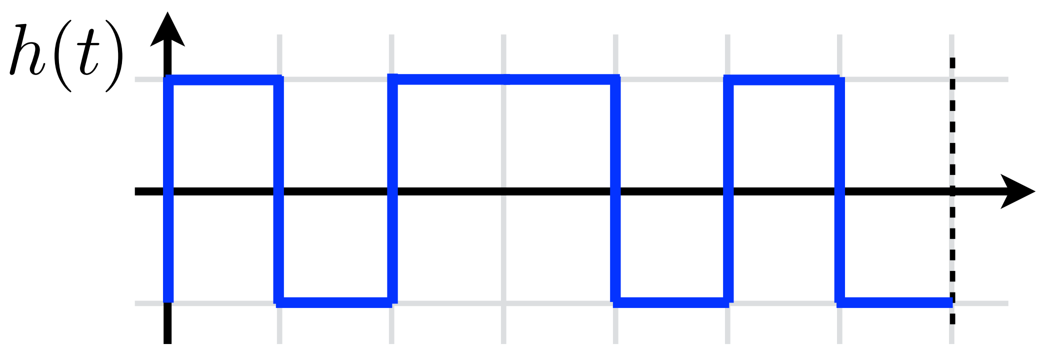
Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

Let's give this game a try!



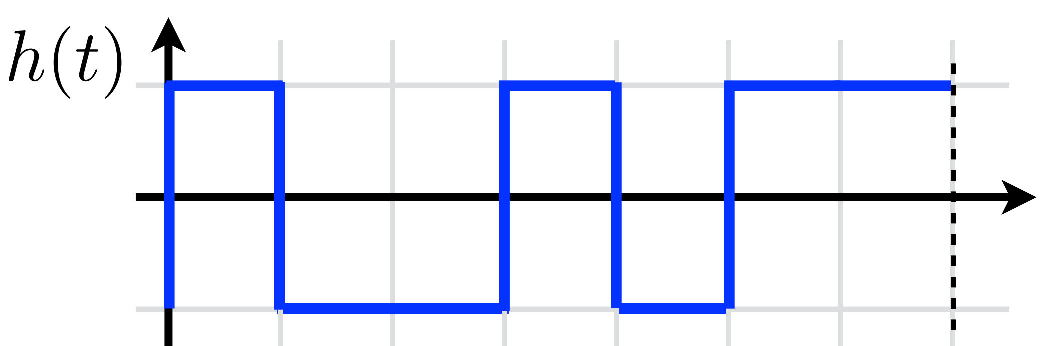
measurement: -1



measurement: $+1$

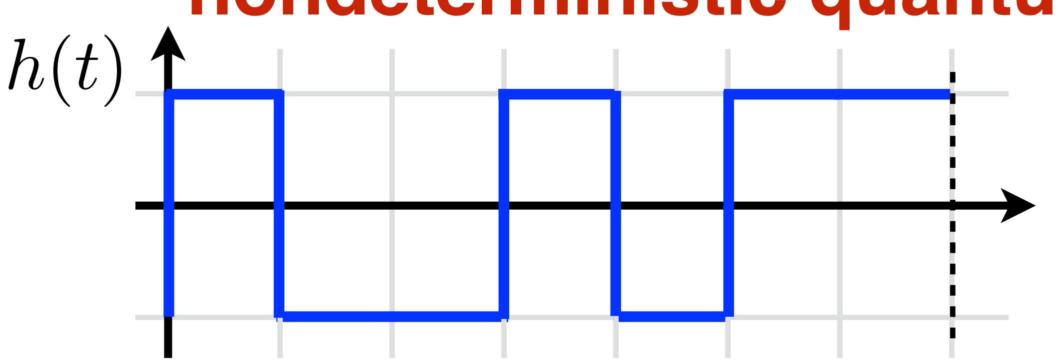
(different final state: different probability to be in the target state)

→ repeat protocol!



measurement: $+1$

nondeterministic quantum measurements create headache!

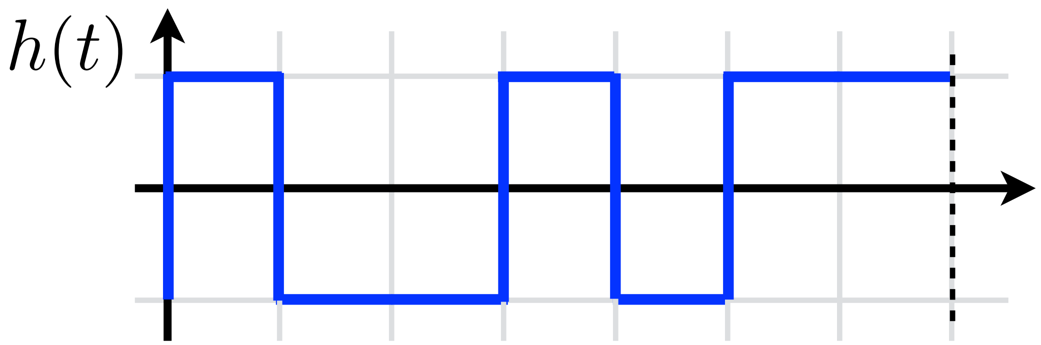


measurement: -1

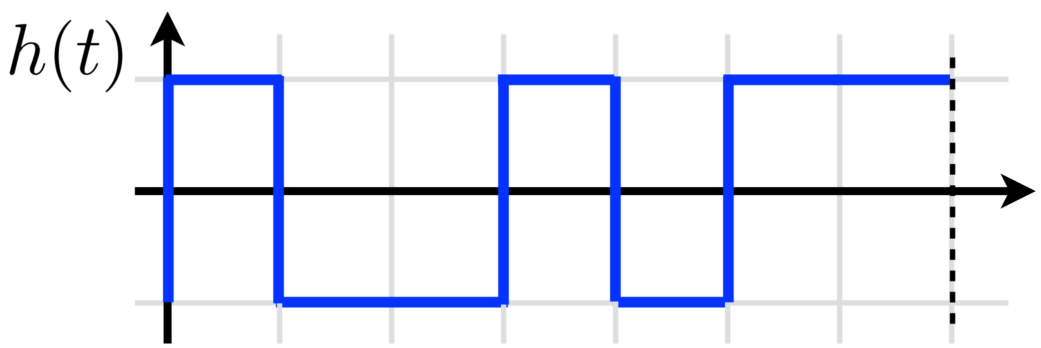
Marin Bukov

Let's get rid of this 'quantumness' for a sec

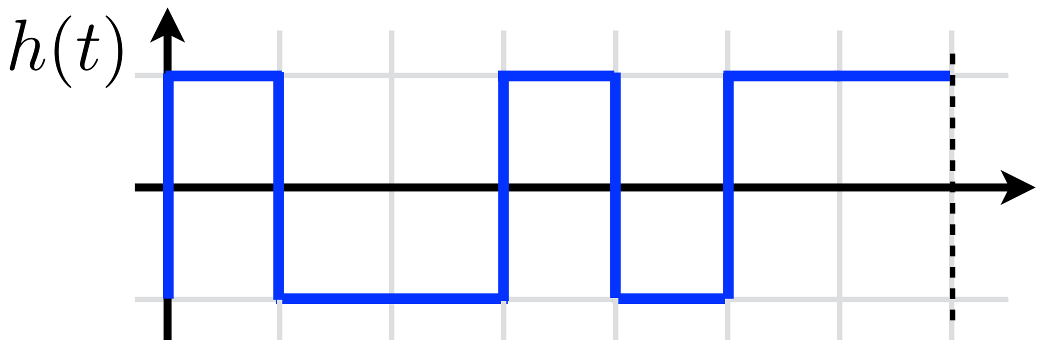
→ repeat protocol again!



measurement:
$F_h = |\langle\psi(T)|\psi_*\rangle|^2 = 0.632$



measurement: $F_h = 0.592$

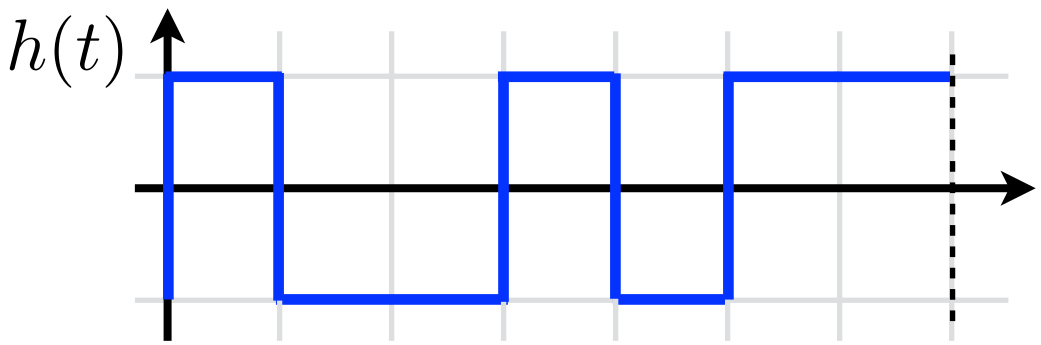


measurement: $F_h = 0.668$

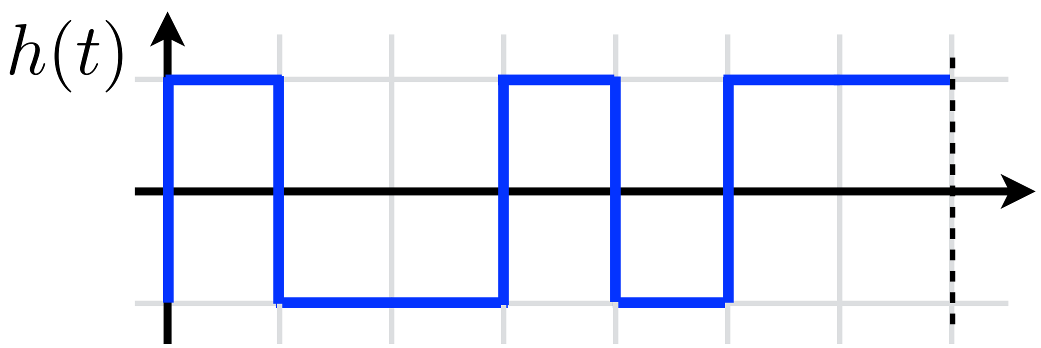
Marin Bukov

Let's get rid of this 'quantumness' for a sec

→ repeat protocol again!

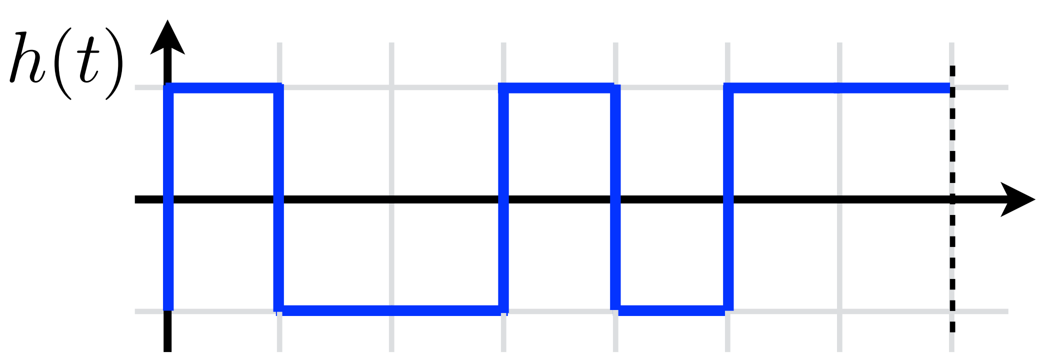


measurement:
$F_h = |\langle \psi(T) | \psi_* \rangle|^2 = 0.632$



measurement: $F_h = 0.592$

initial state could not be prepared perfectly: more headache!

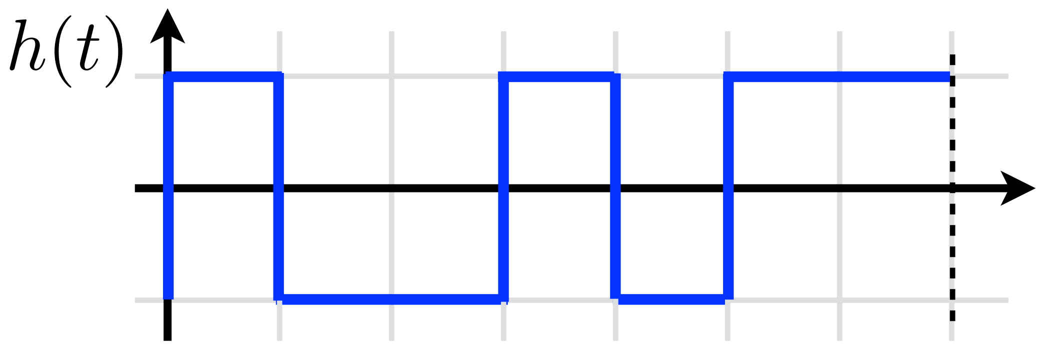


measurement: $F_h = 0.668$

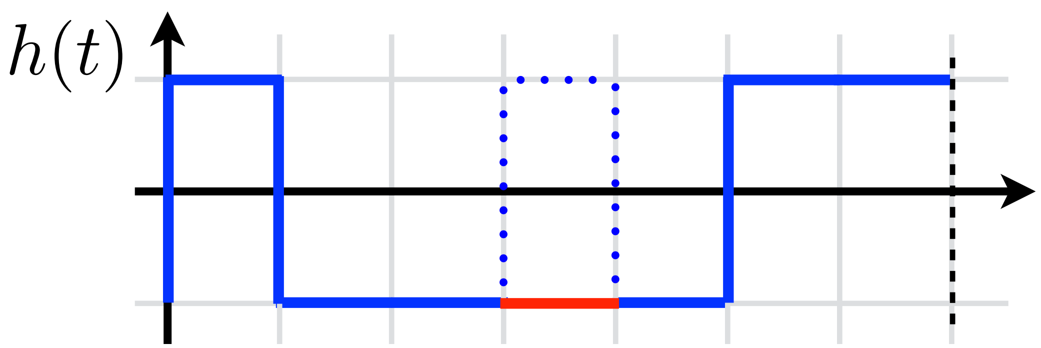
Marin Bukov

Let's maybe also fix the initial state

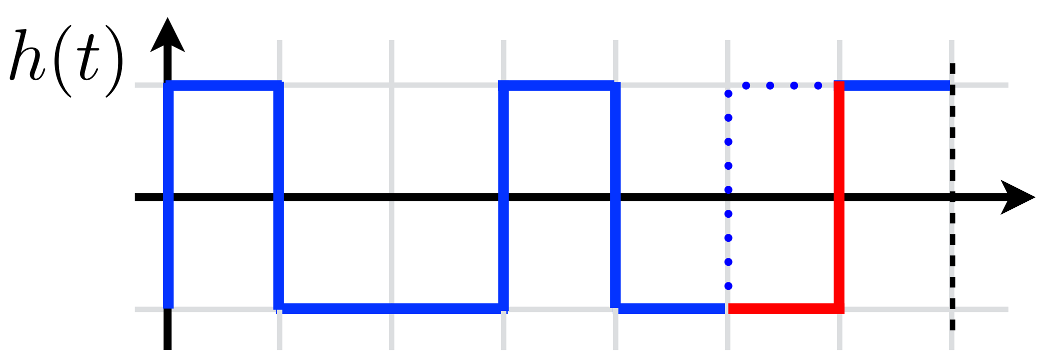
→ repeat protocol again!



measurement: $F_h = 0.627$



measurement: $F_h = 0.572$



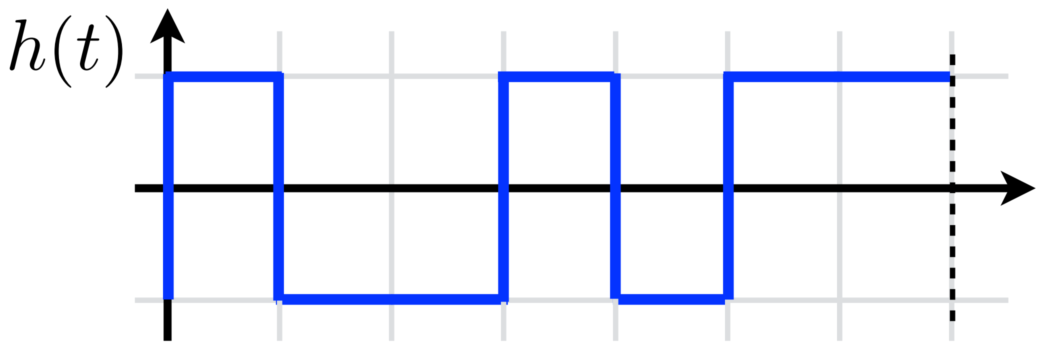
measurement: $F_h = 0.657$

Marin Bukov

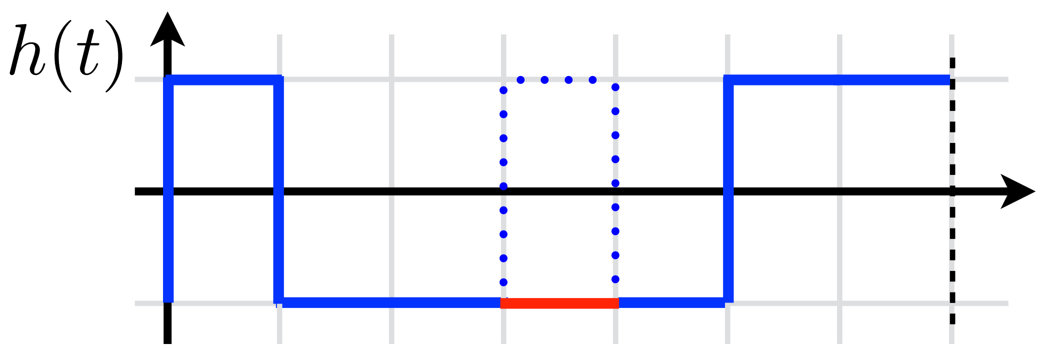
Berkeley
UNIVERSITY OF CALIFORNIA

Let's maybe also fix the initial state

→ repeat protocol again!

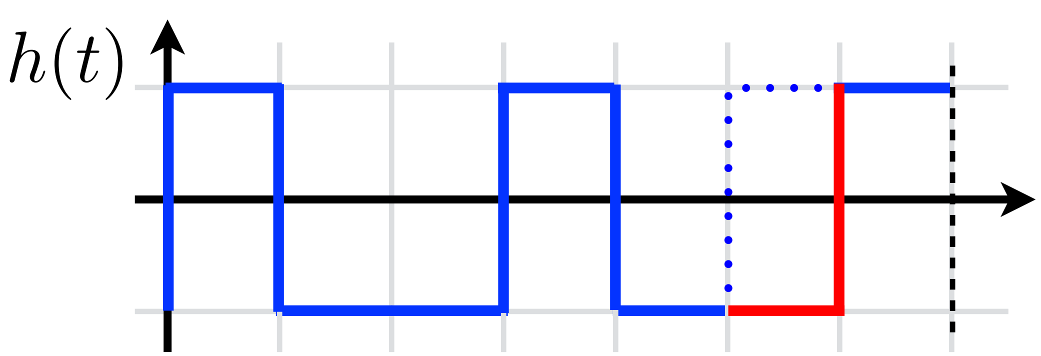


measurement: $F_h = 0.627$



measurement: $F_h = 0.572$

control apparatus failed: it can't be!

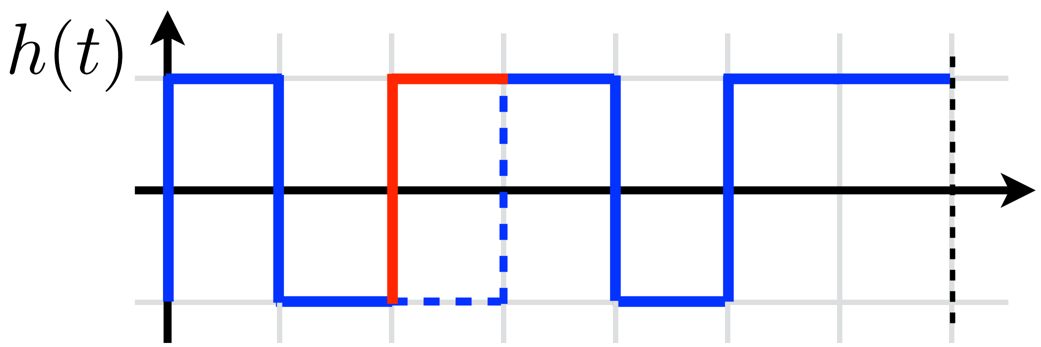


measurement: $F_h = 0.657$

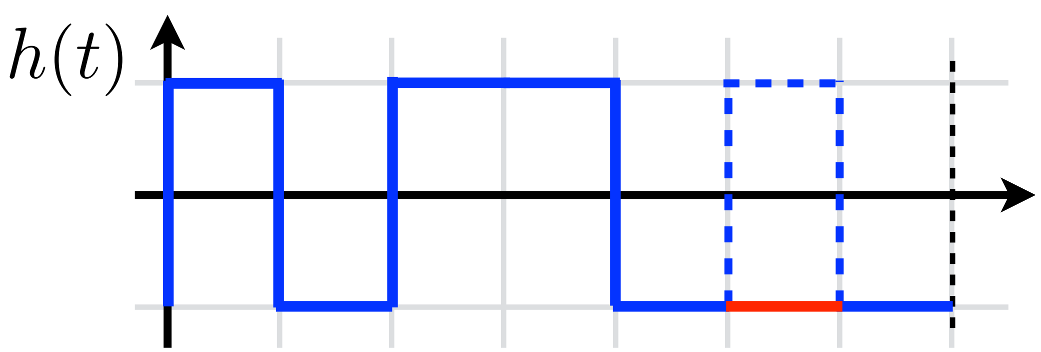
Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

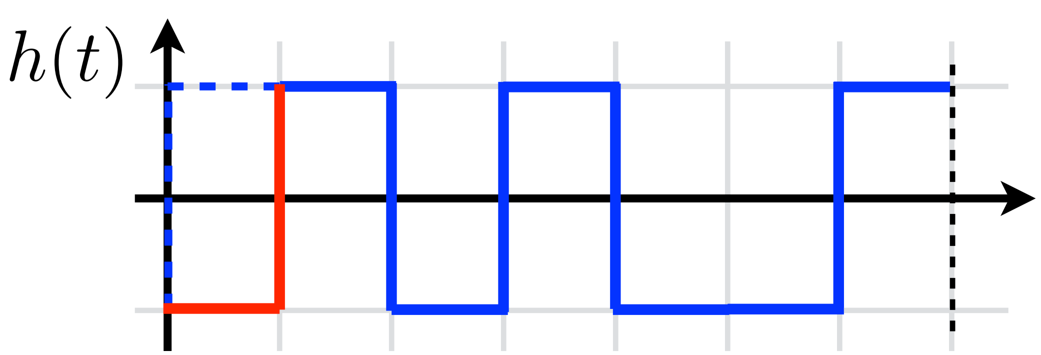
The Cruel Reality: all together (and probably much more!)



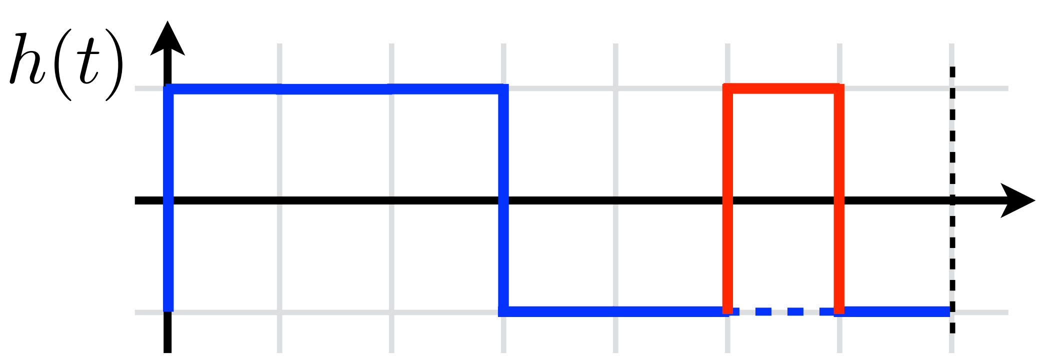
measurement: -1



measurement: $+1$



measurement: -1

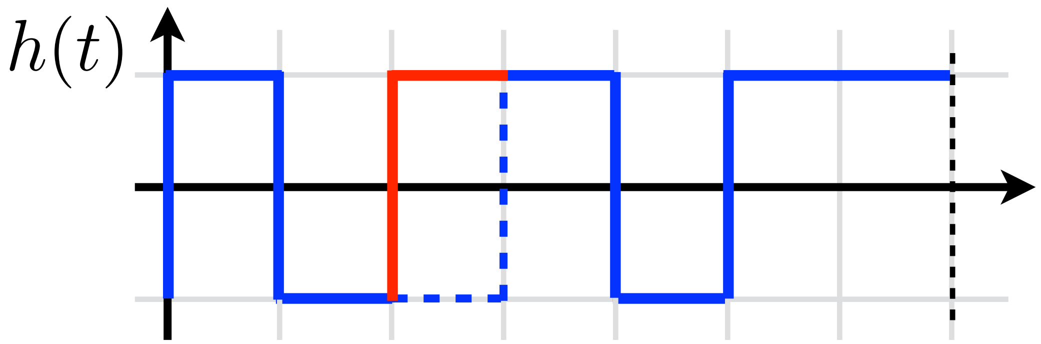


measurement: -1

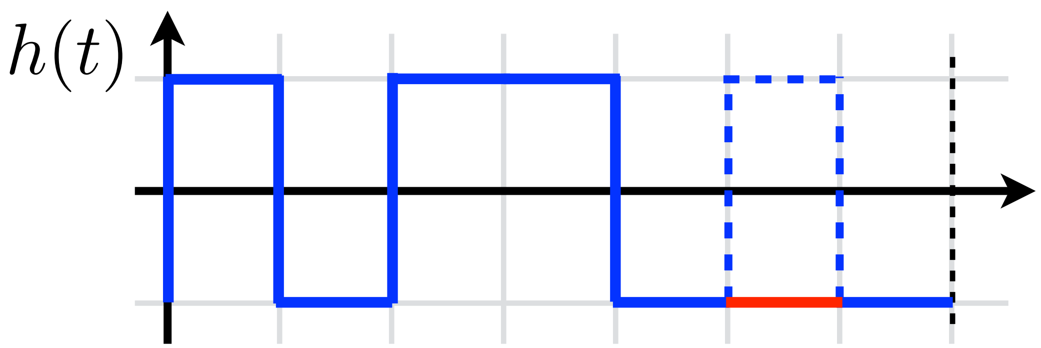
Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

The Cruel Reality: all together (and probably much more!)

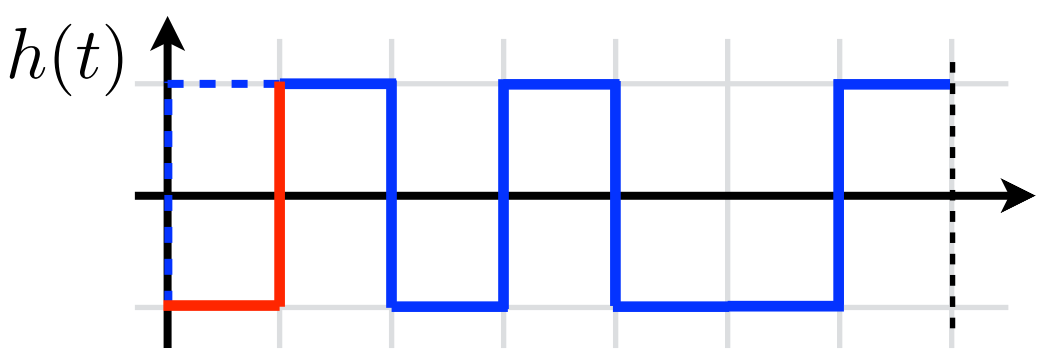


measurement: -1

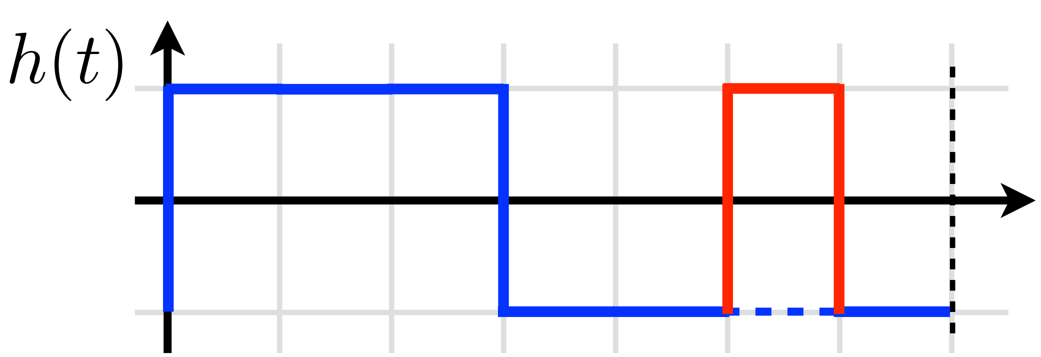


measurement: $+1$

extremely tedious task!



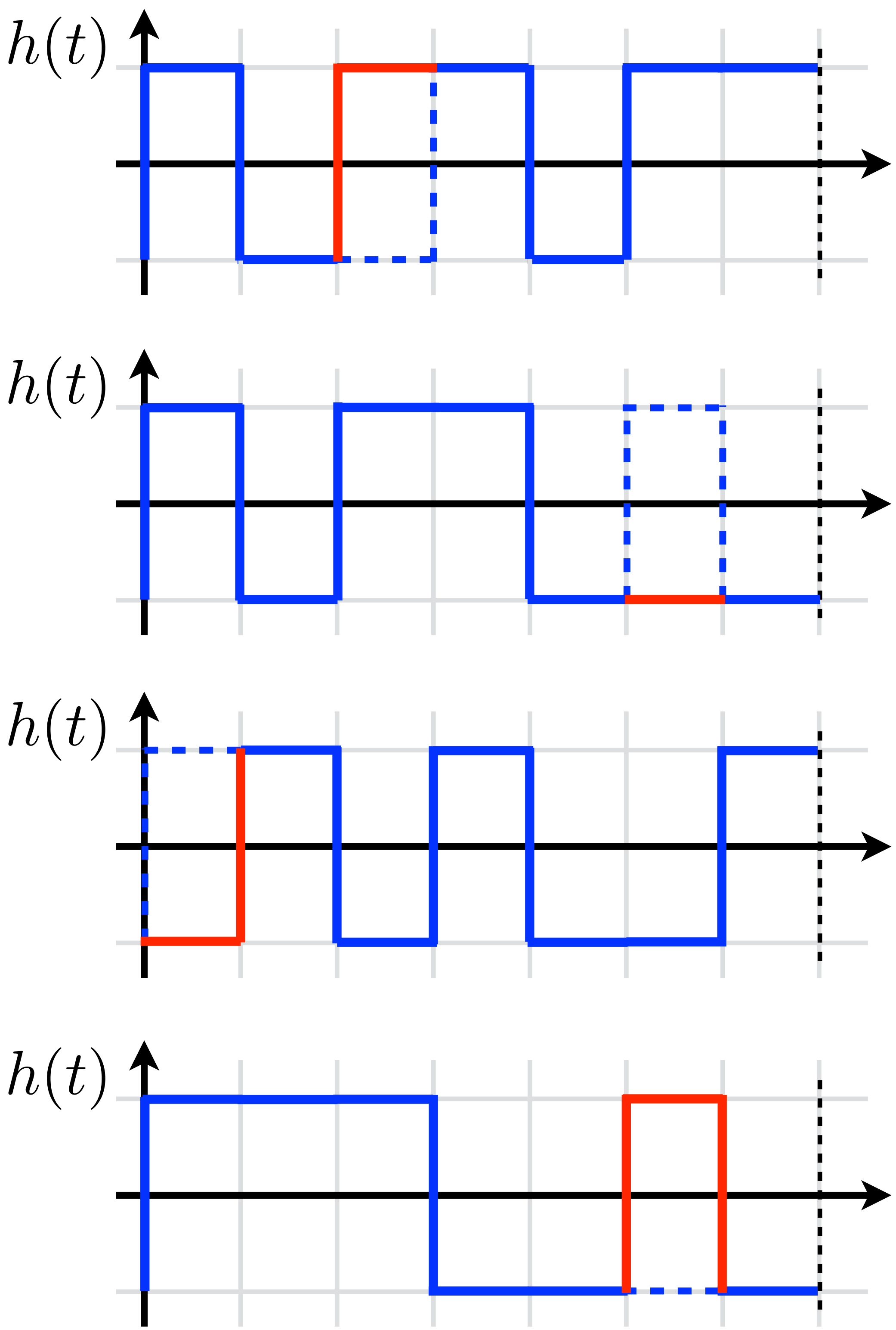
measurement: -1



measurement: -1

Berkeley
UNIVERSITY OF CALIFORNIA

The Cruel Reality: all together (and probably much more!)



measurement: -1

measurement: $+1$

extremely tedious task!

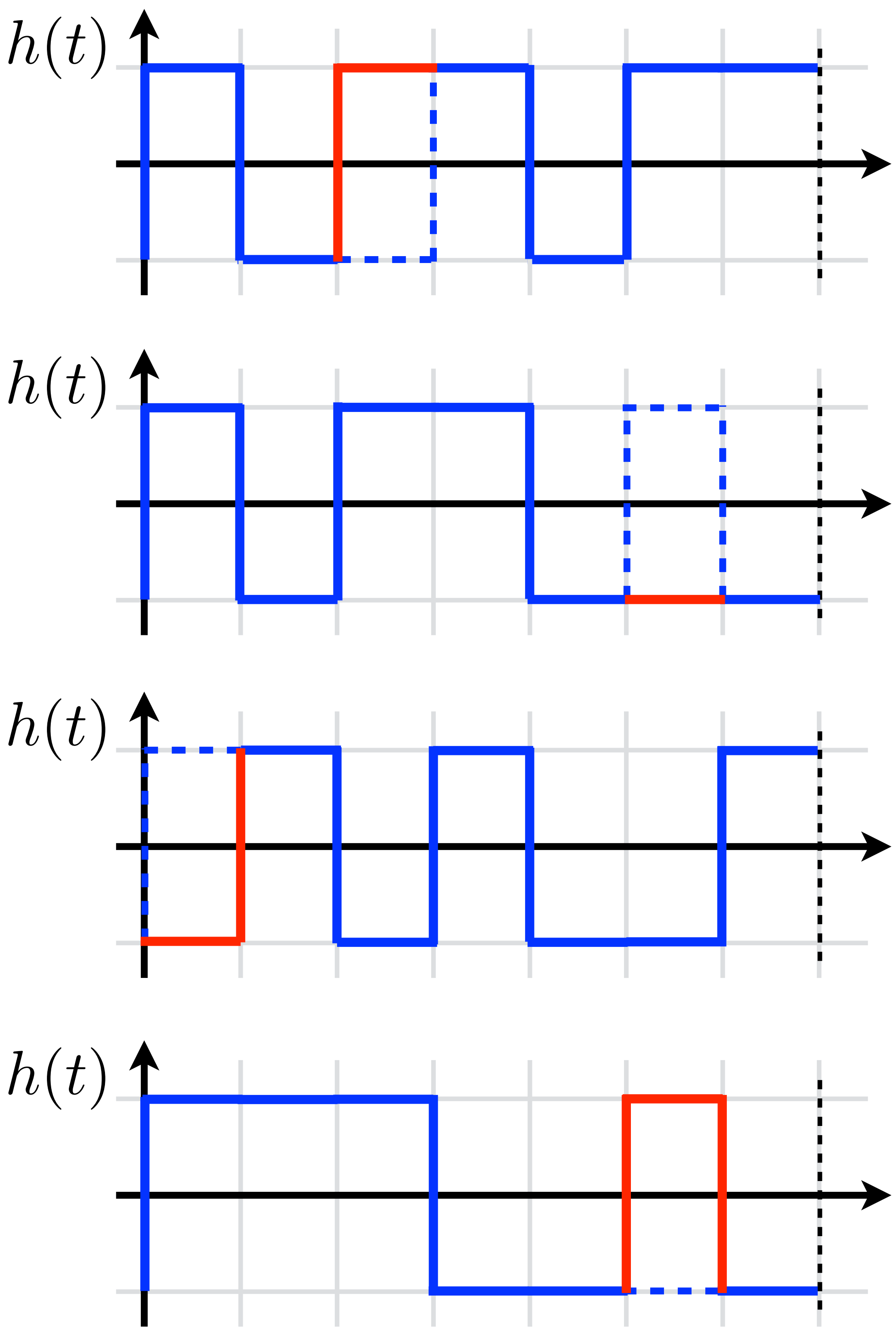
measurement: -1

how do we solve it efficiently?

measurement: -1

Marin Bukov

The Cruel Reality:
all together (and probably much more!)



measurement: -1

measurement: $+1$

extremely tedious task!

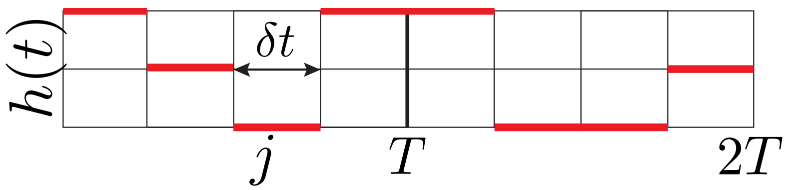
measurement: -1

how do we solve it efficiently?

measurement: -1

can we automate it?

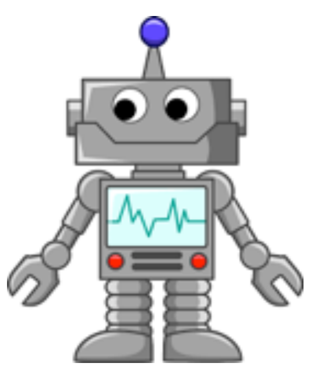
Reinforcement Learning to Prepare the Inverted Position Floquet State



4 driving cycles (periods), 32 steps (8 per period)

Kapitza pendulum

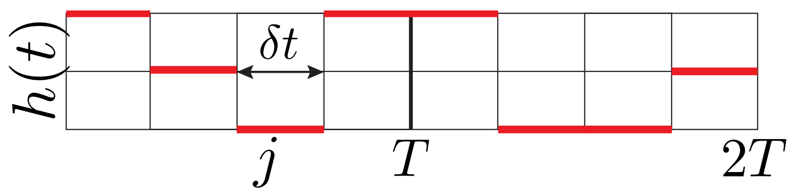
$t/T = 0.00,\ \theta(t) = 0.00\pi,\ p_\theta(t) = 0.00,\ r(t) = 0.00$



$h_{\max}/(m\omega_0) = 4.0$ $\Omega/(m\omega_0) = 10.0$ $A/(m\omega_0) = 2.0$

Marin Bukov

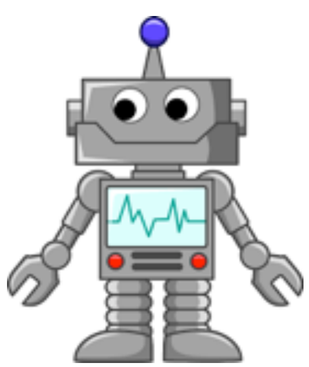
Reinforcement Learning to Prepare the Inverted Position Floquet State



4 driving cycles (periods), 32 steps (8 per period)

Kapitza pendulum

$t/T = 0.00,\ \theta(t) = 0.00\pi,\ p_\theta(t) = 0.00,\ r(t) = 0.00$

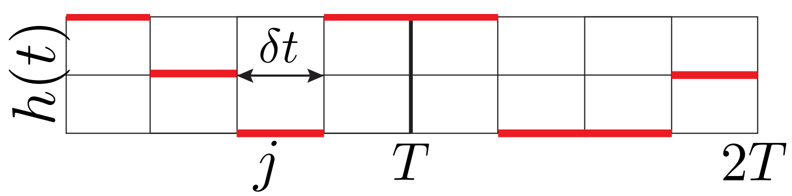


Marin Bukov

$h_{\max}/(m\omega_0) = 4.0$ $\Omega/(m\omega_0) = 10.0$ $A/(m\omega_0) = 2.0$

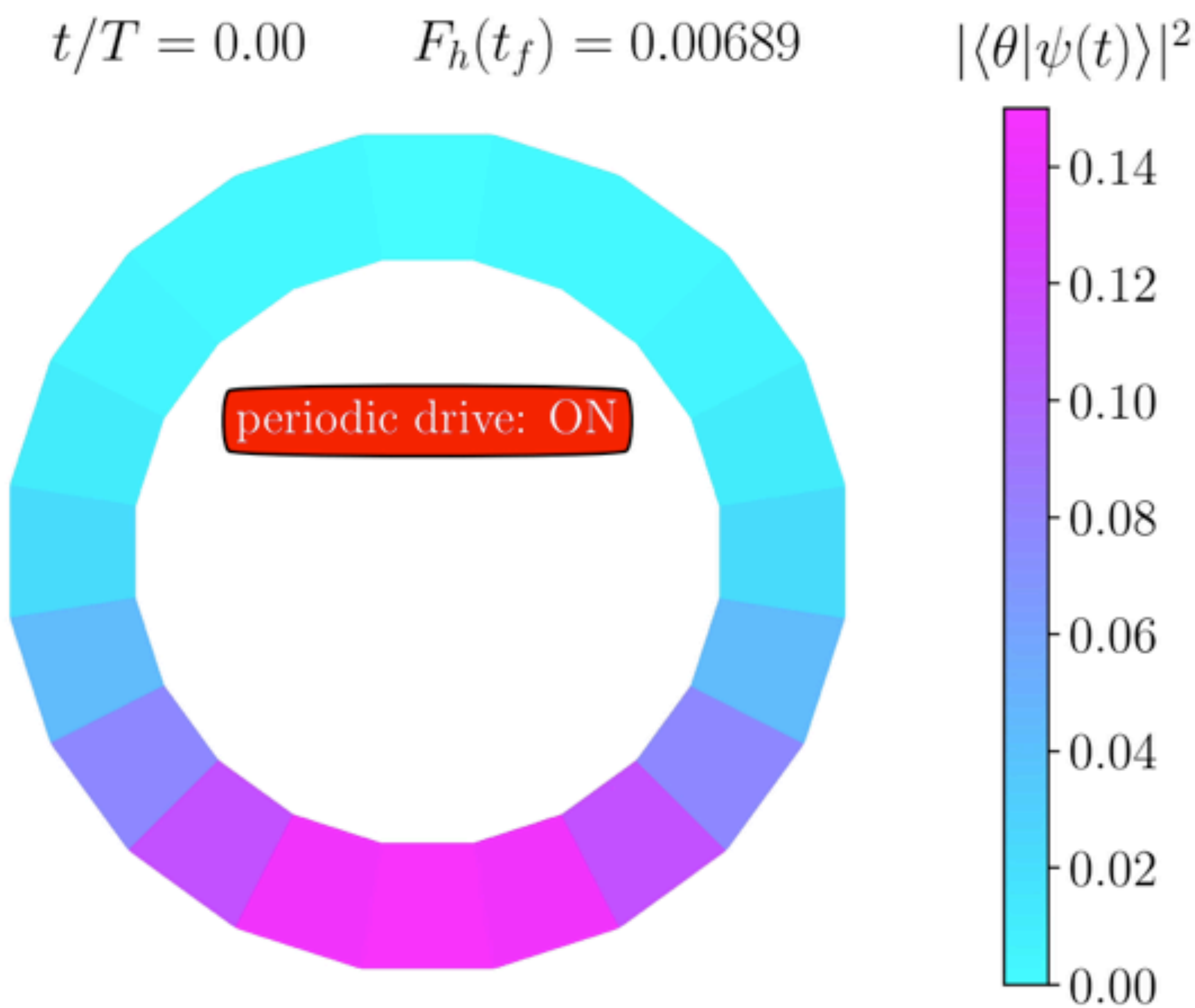
Berkeley UNIVERSITY OF CALIFORNIA

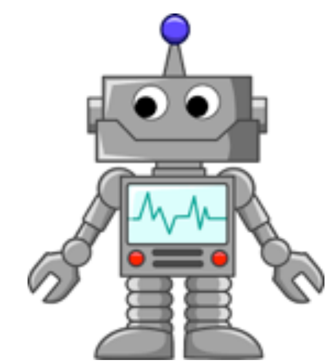
Reinforcement Learning to Prepare the Inverted Position Floquet State



15 driving cycles (periods), 120 steps (8 per period)

quantum Kapitza oscillator



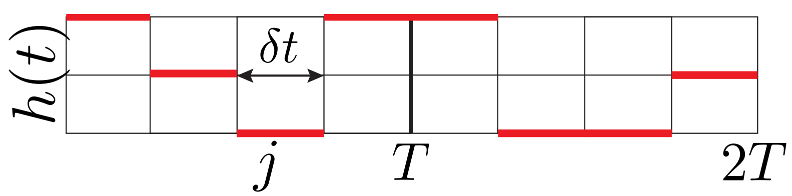


$h_{\max}/(m\omega_0) = 4.0$ $\Omega/(m\omega_0) = 10.0$ $A/(m\omega_0) = 2.0$

Marin Bukov

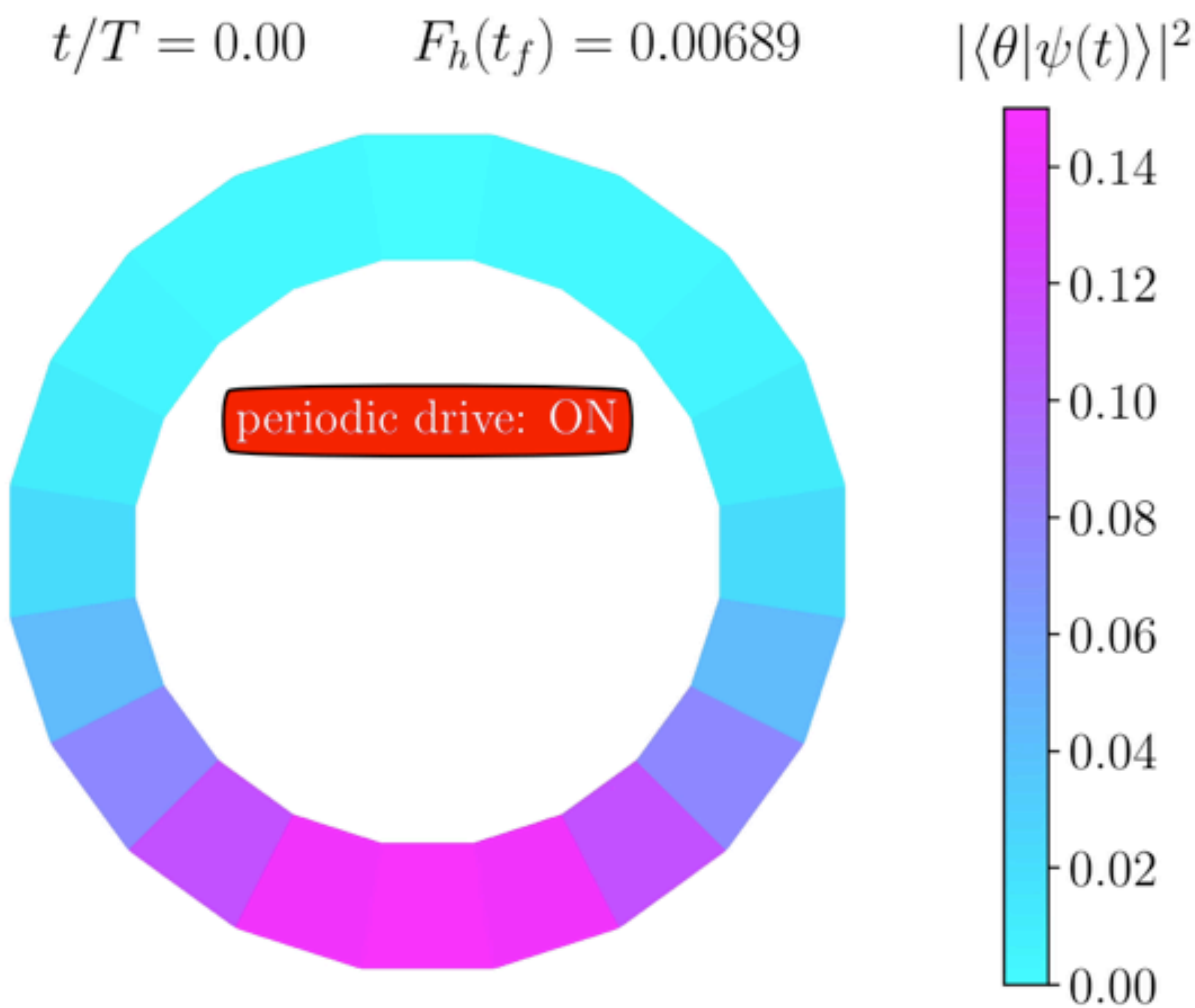
Berkeley UNIVERSITY OF CALIFORNIA

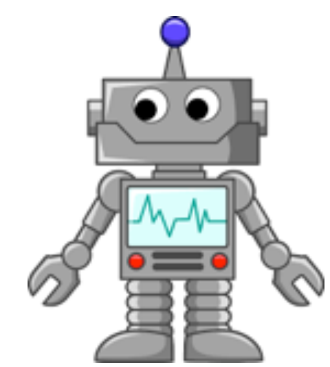
Reinforcement Learning to Prepare the Inverted Position Floquet State



15 driving cycles (periods), 120 steps (8 per period)

quantum Kapitza oscillator





$h_{\max}/(m\omega_0) = 4.0$ $\Omega/(m\omega_0) = 10.0$ $A/(m\omega_0) = 2.0$

Marin Bukov

Berkeley
UNIVERSITY OF CALIFORNIA

Outlook

web: *mgbukov.github.io*

→ Which problems can we study with RL that we can't do otherwise?

→ Can RL lead to the discovery of new physics?

→ What's RL's most appropriate physics application as a toolbox?

funding: GORDON AND BETTY MOORE FOUNDATION

ML review with Jupyter notebooks: arXiv: *1803.08823*

RL in non equilibrium dynamics : *PRX 8 0311086 (2018)*, arXiv: *1808.08910*

control phase transitions: *PRA 97 052114 (2018)*, arXiv: *1803.10856*

QuSpin: *http://weinbe58.github.io/QuSpin*

python package for ED & many-body dynamics (with P. Weinberg, BU)

Marin Bukov