University of Stuttgart
Institute for Computational Physics

SimTech

PN 8
-
General Meeting

Daniel Fink

Quantum Computing & Stochastic Processes

August 8th, 2022

Content

- Master Project
- Libraries & Tools
- People & Community
- My Goals

Master Project

Simulating Stochastic Processes

with

Variational Quantum Circuits

- February 14th, 2022

Advisors

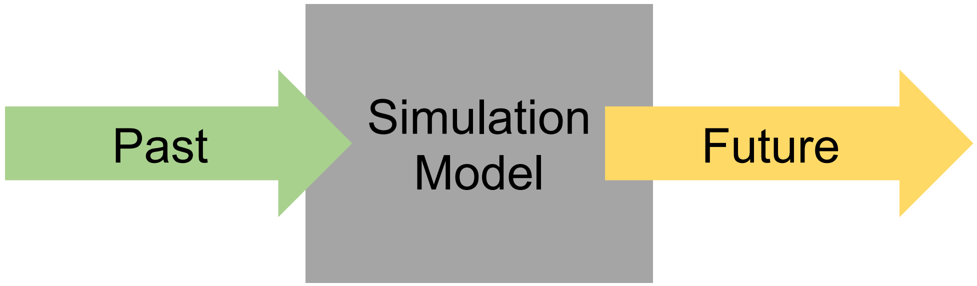
- University of Stuttgart
  - Prof. Dr. Christian Holm
- Free University of Berlin
  - Prof. Dr. Jens Eisert
  - Dr. Nora Tischler
  - Dr. Ryan Sweke
  - M.Sc. Paul Fährmann

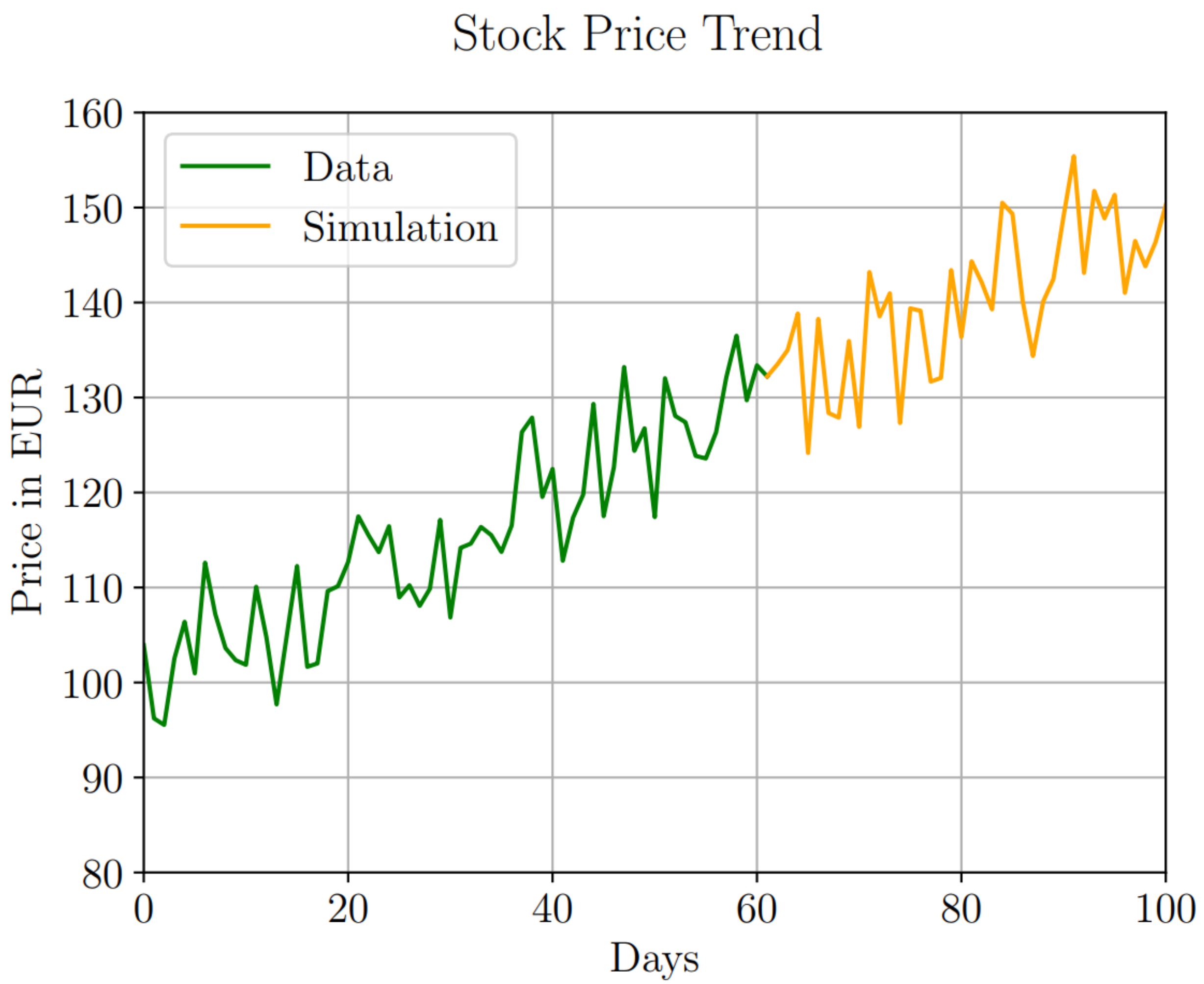
University of Stuttgart
Institute for Computational Physics

Freie Universität Berlin

Motivation

Assume data drawn
by a stochastic process





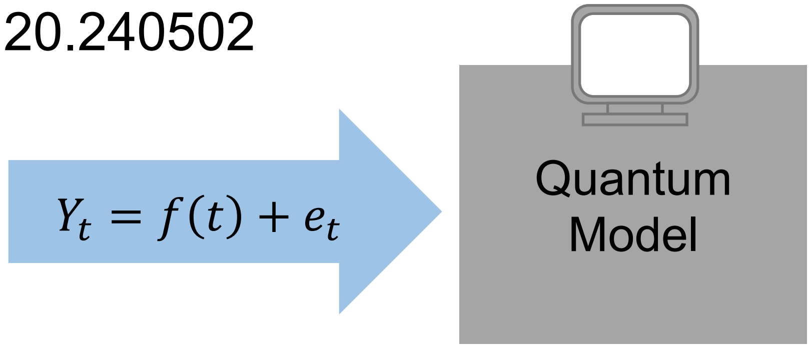
Motivation

Classical Models $\leq$ Quantum Models

How to get a quantum model?

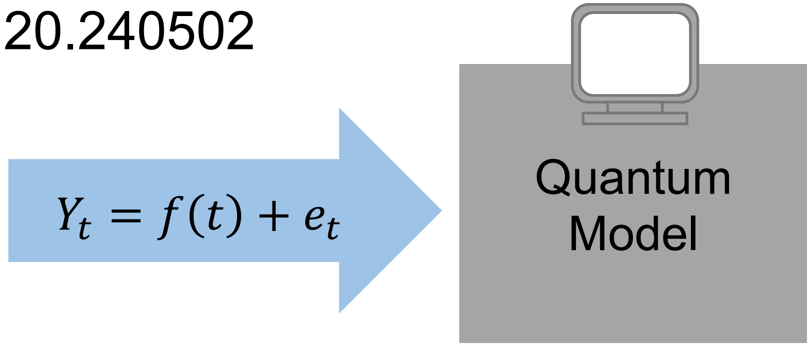
Motivation

- Classical description of the process → q–simulator
  - Binder et al., 10.1103/PhysRevLett.120.240502

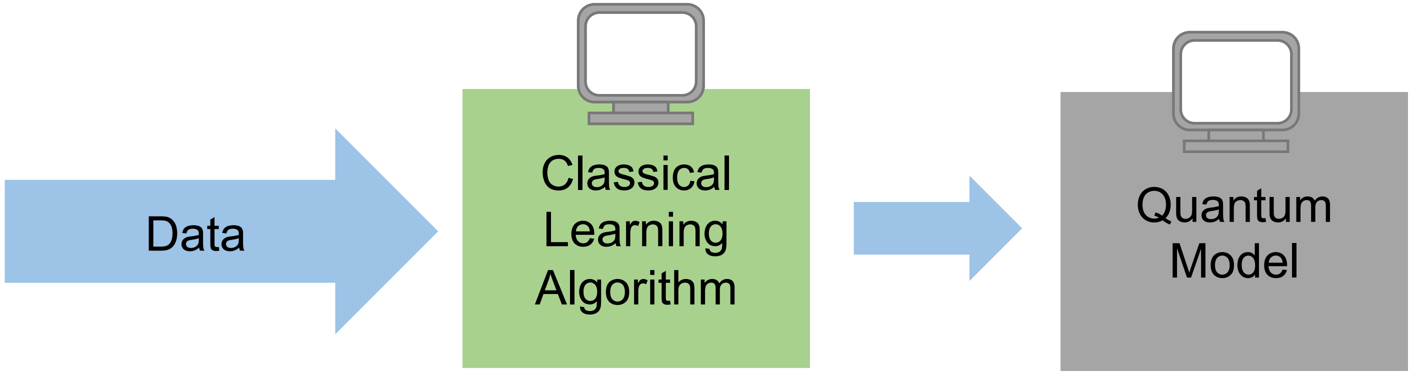


Motivation

- Classical description of the process → q–simulator
  - Binder et al., 10.1103/PhysRevLett.120.240502



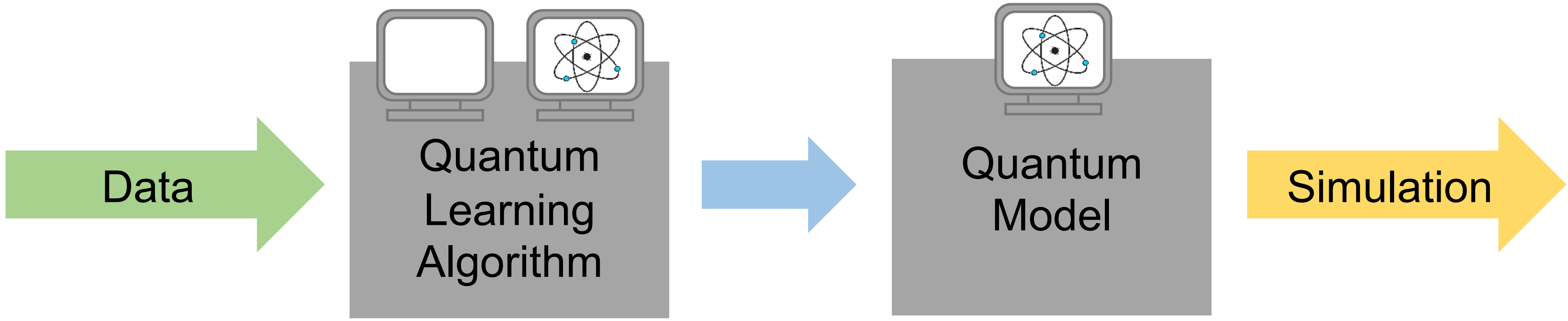
- Data from the process → classical discovery algorithm
  - Yang et al., arXiv:2105.14434



Motivation

Goal

Develop a quantum learning algorithm for simulation models, which uses only data as input.



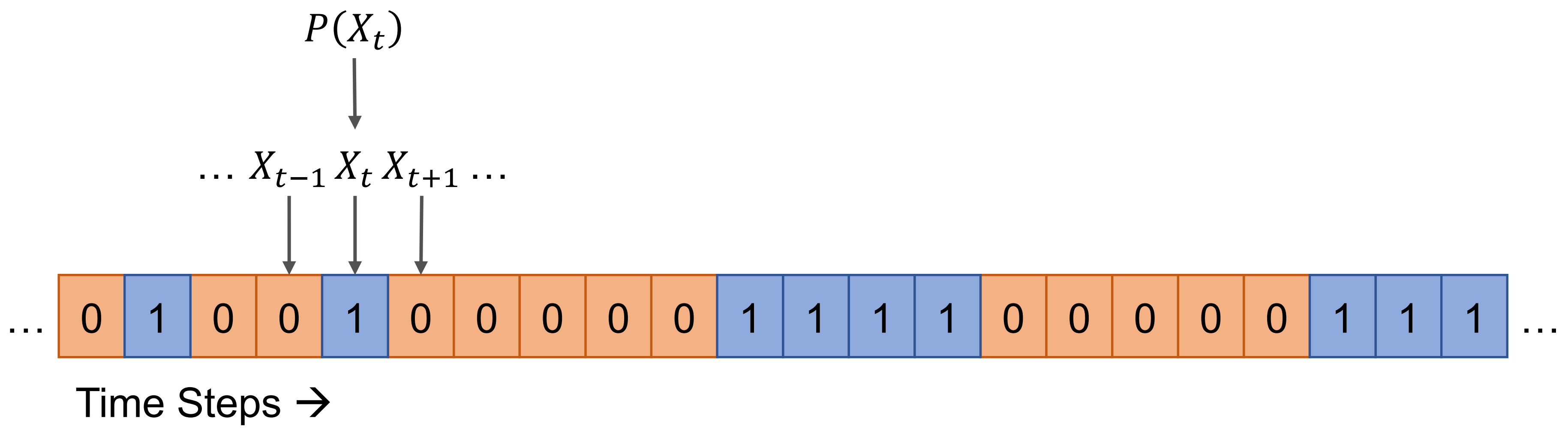
Stochastic Processes

Stochastic Processes

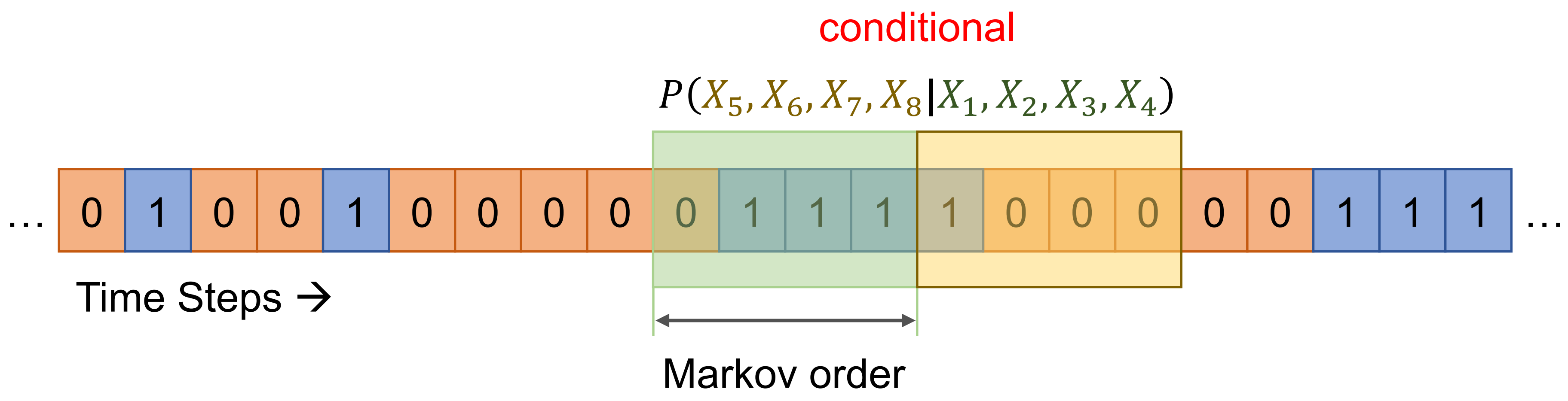
…	0	1	0	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	…

Time Steps →

Stochastic Processes



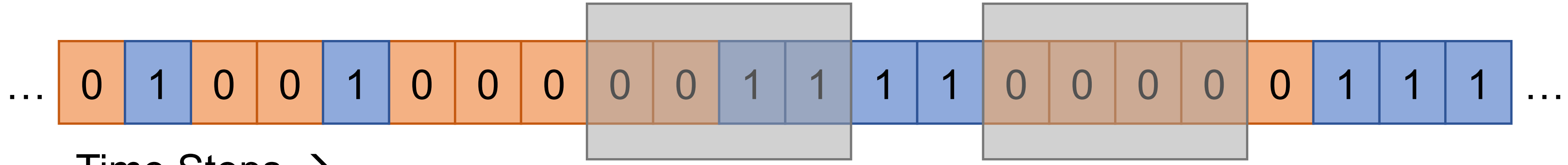
Stochastic Processes



Stochastic Processes

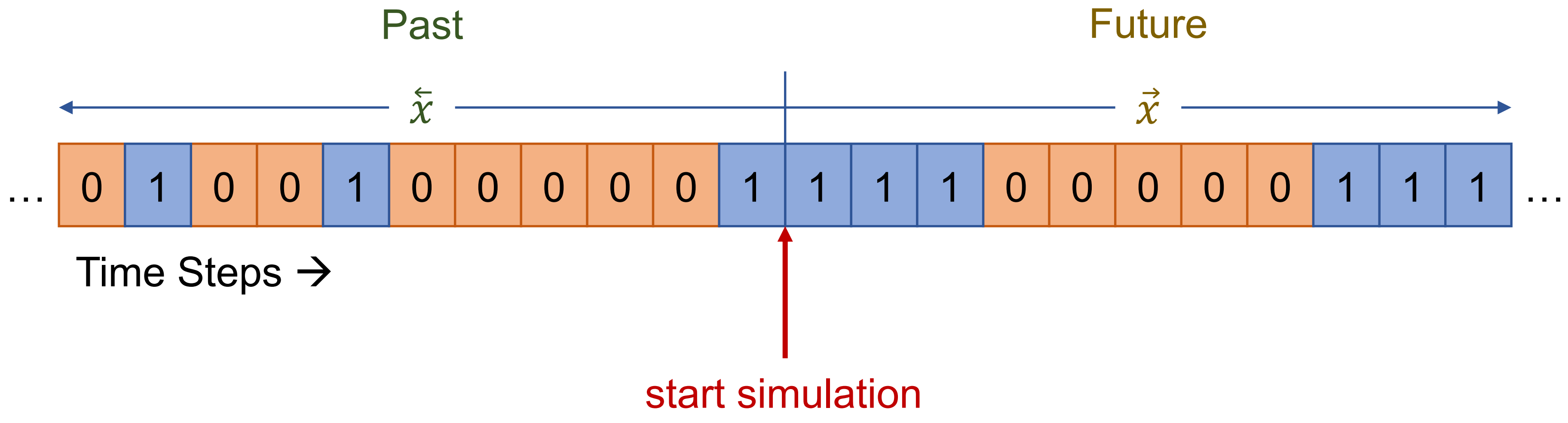
stationary

$$P(X_1, X_2, X_3, X_4) \quad = \quad P(X_5, X_6, X_7, X_8)$$

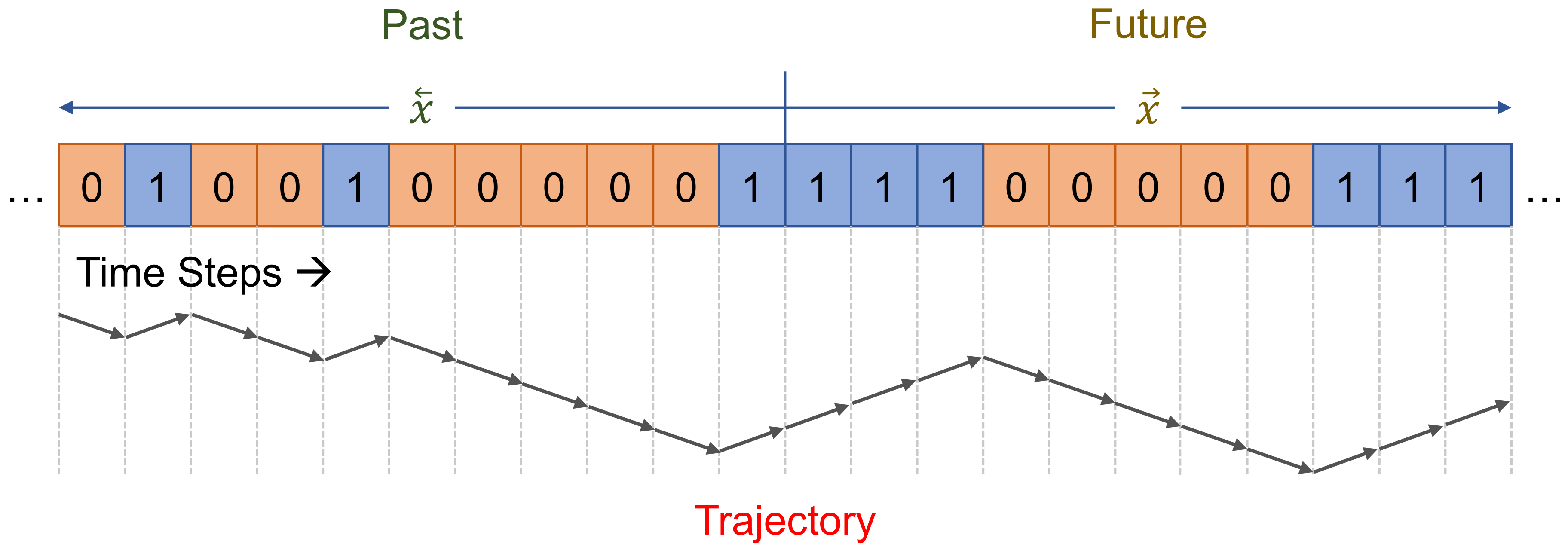


Time Steps →

Stochastic Processes

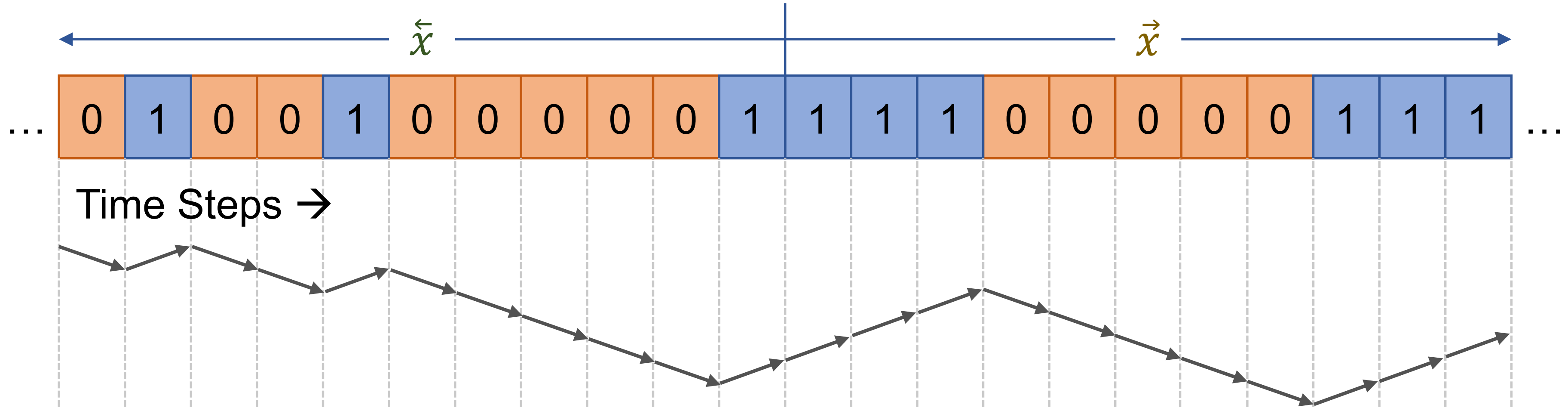


Stochastic Processes



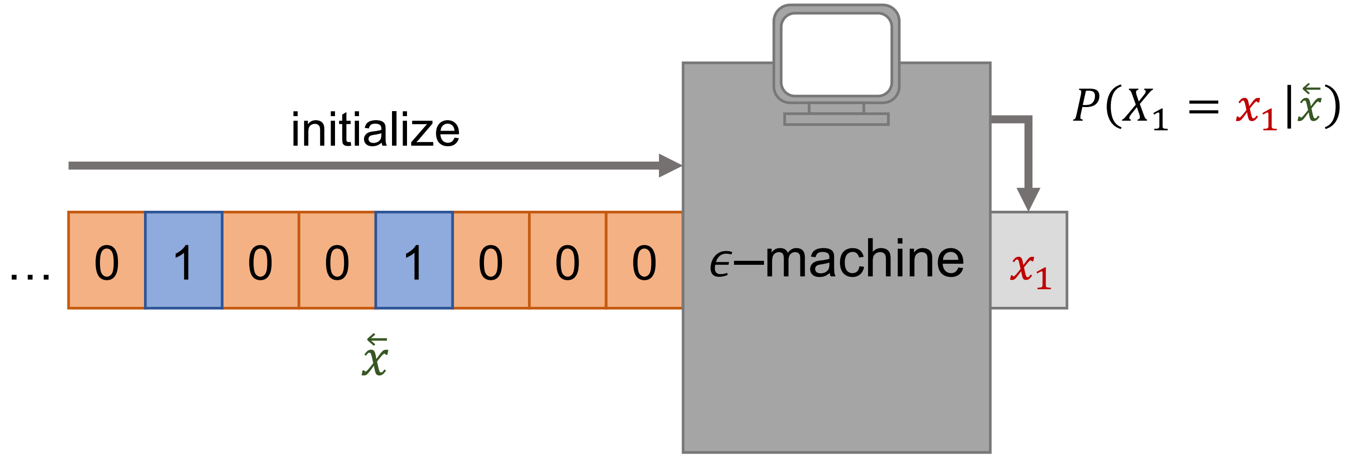
Stochastic Processes

- Simulating = sampling trajectories
- Trajectory is governed by $P(\vec{X}|\overleftarrow{X})$



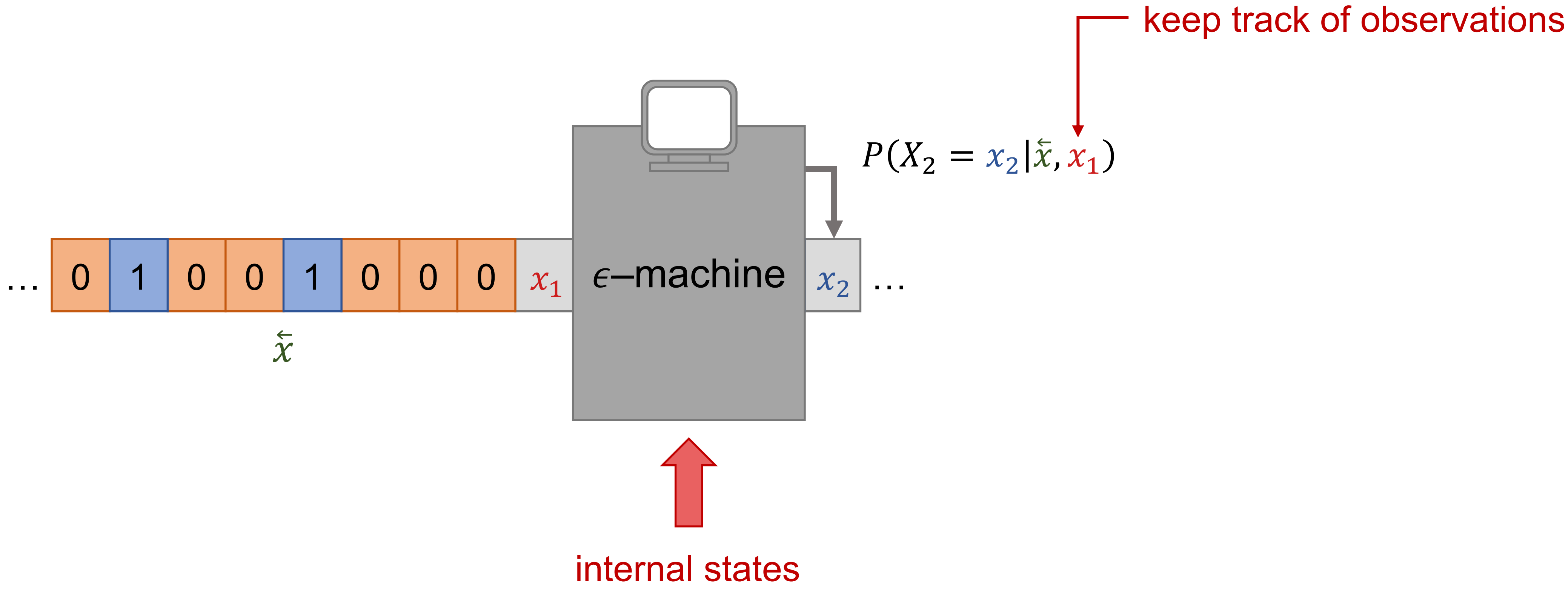
Trajectory

ϵ–machine

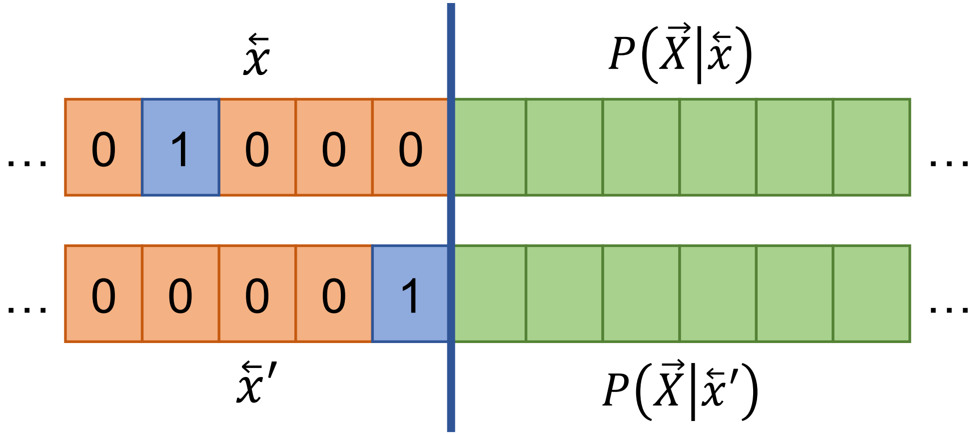


initialize
$P(X_1 = x_1 | \overleftarrow{x})$
… 0 1 0 0 1 0 0 0
ϵ–machine
x_1
$\overleftarrow{x}$

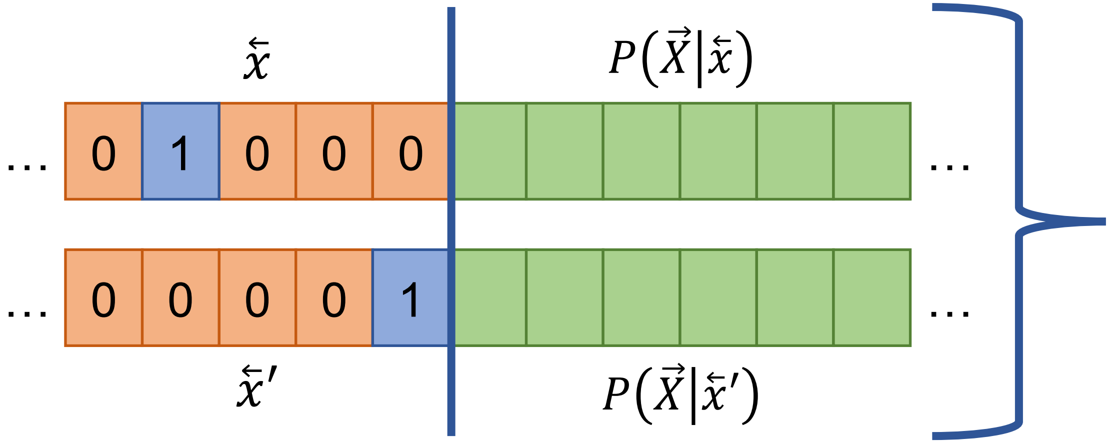
ϵ–machine



ϵ–machine

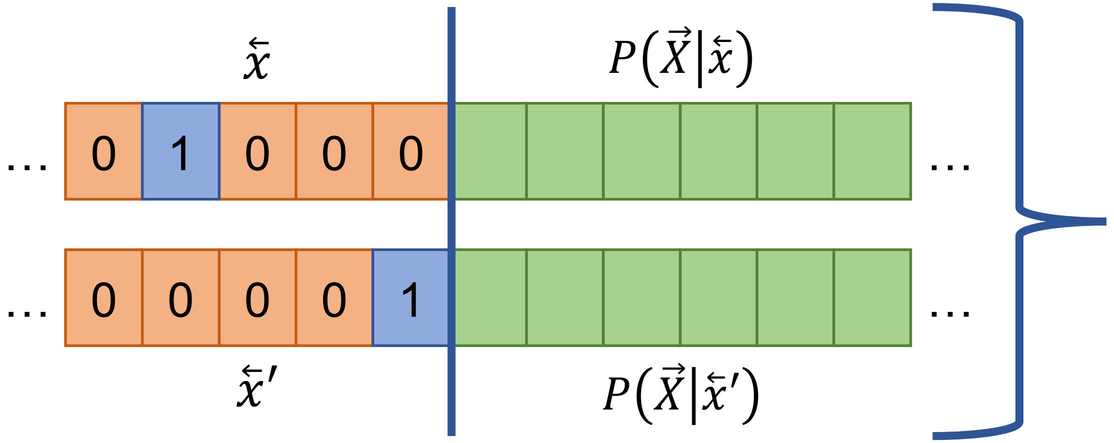


ϵ–machine



$$\overleftarrow{x} \sim \overleftarrow{x}' \iff P(\overrightarrow{X}|\overleftarrow{x}) = P(\overrightarrow{X}|\overleftarrow{x}')$$

ϵ–machine



$\overleftarrow{x} \sim \overleftarrow{x}' \iff P(\vec{X}|\overleftarrow{x}) = P(\vec{X}|\overleftarrow{x}')$

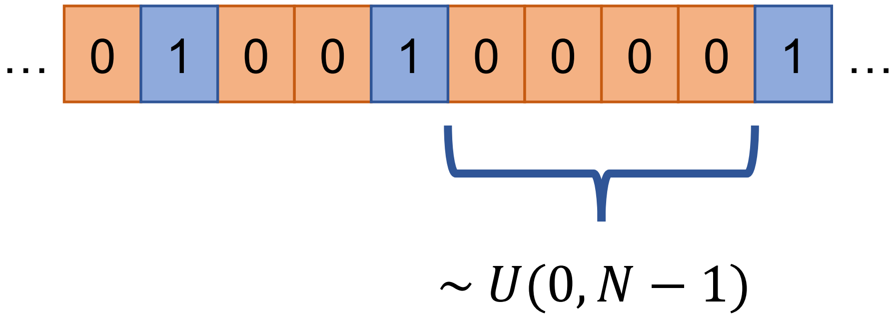
causal sates $\{S_i\}_{i=1}^{N}$ (minimal)

map $E(\overleftarrow{x}) = S_i$ (Initialization)

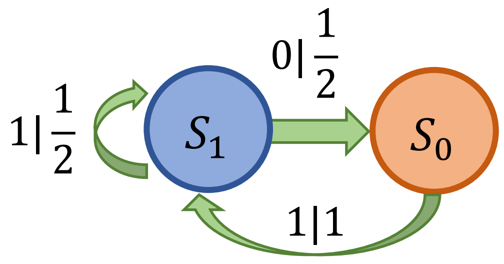
Classical Topological Complexity: $d_c = \log_2 N$

Example

Period-N Uniform Renewal Process



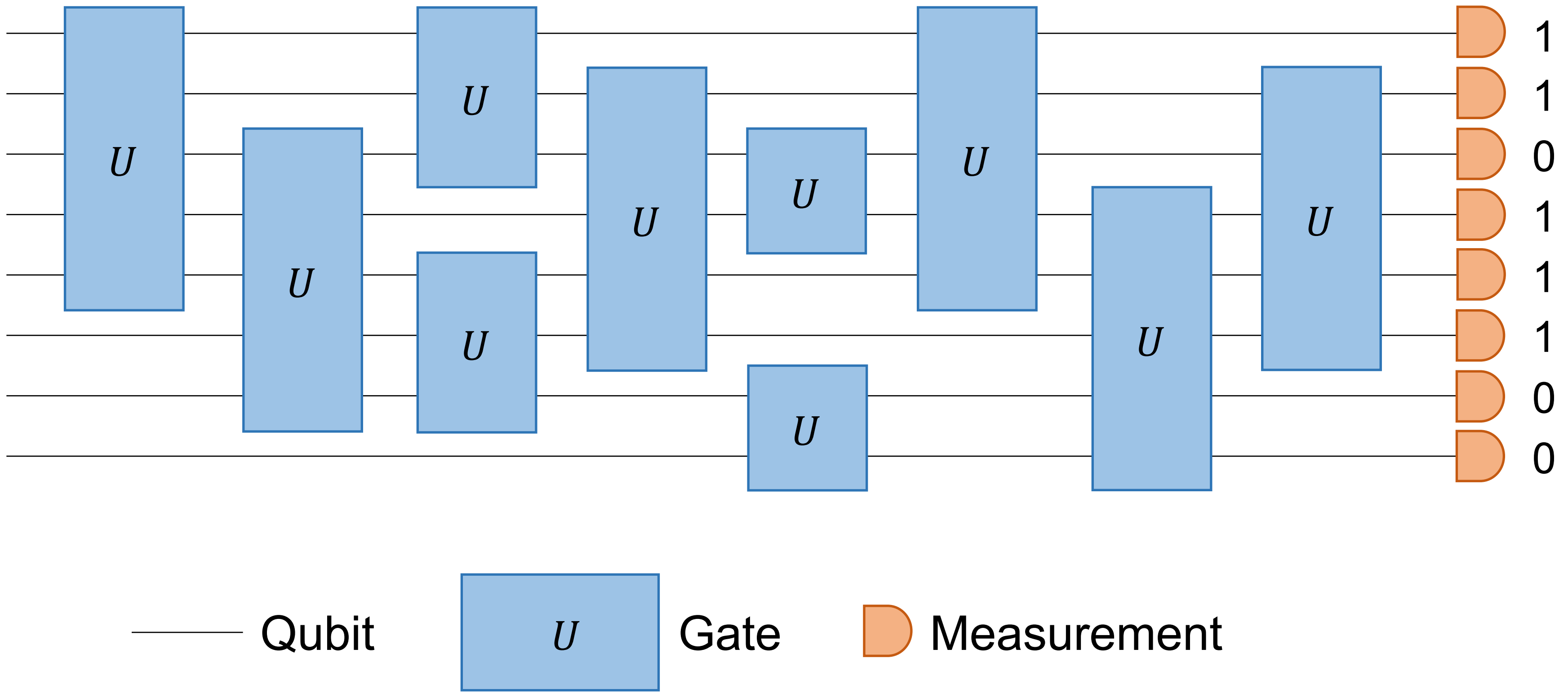
Period = 2



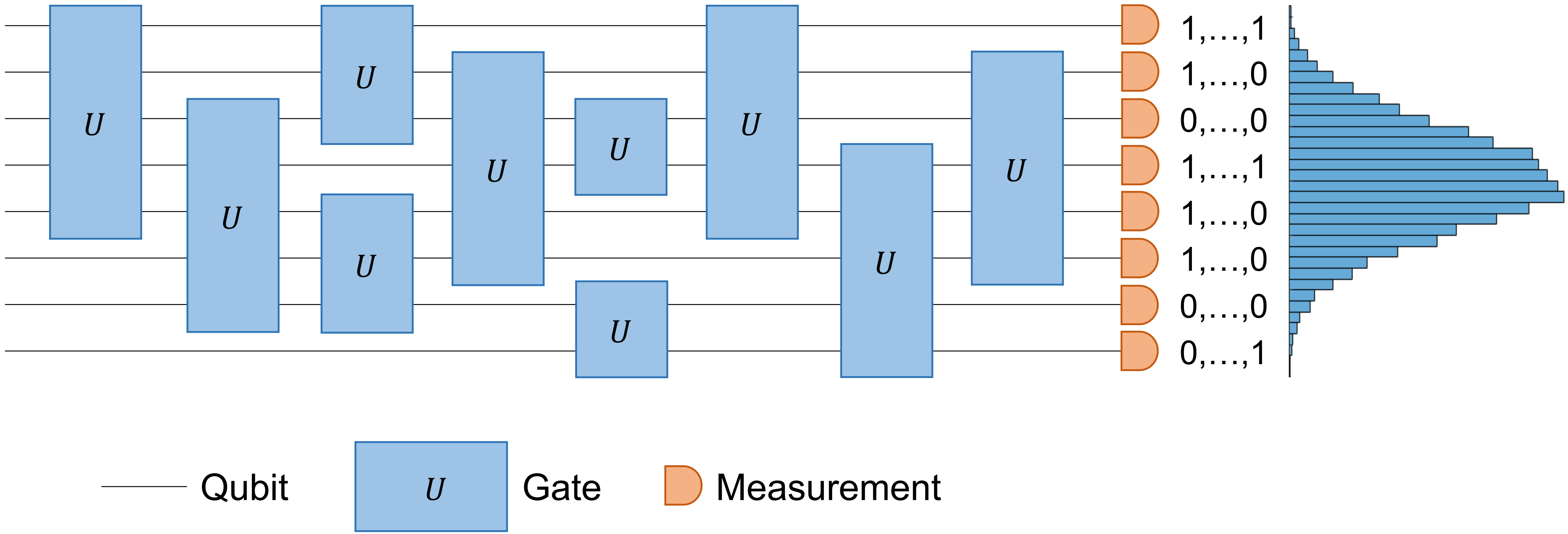
ϵ-machine

Quantum Circuits

Quantum Circuits

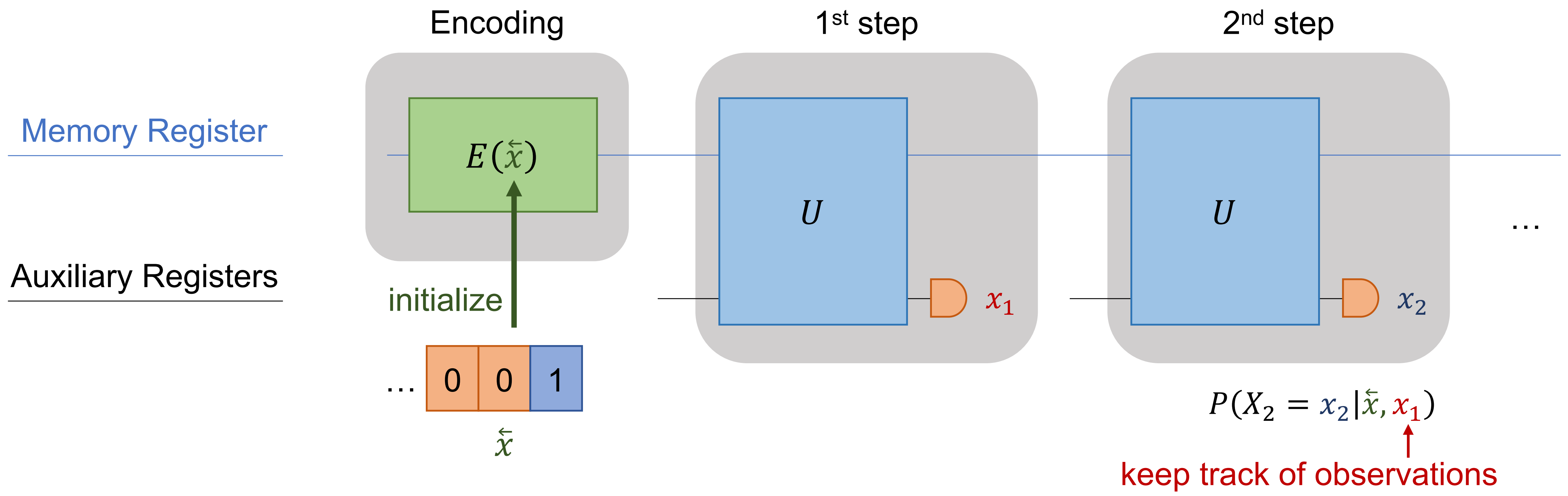


Quantum Circuits

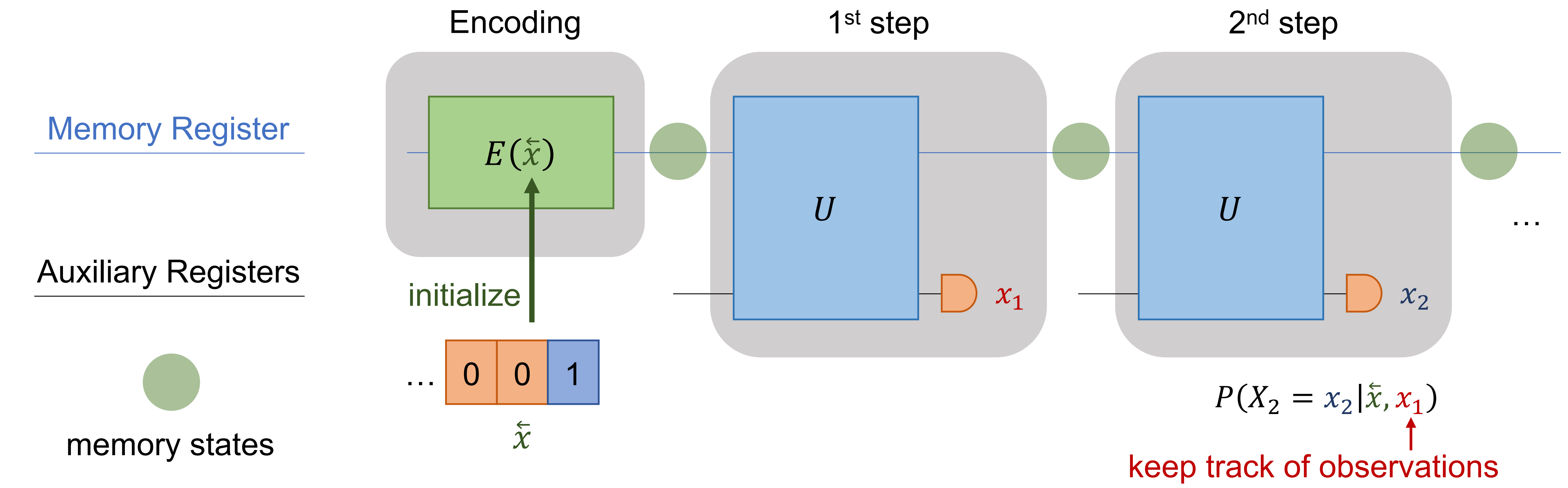


q–simulator

q–simulator

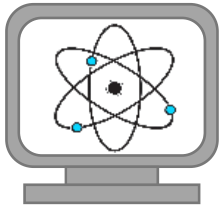

Encoding
1st step
2nd step
Memory Register
Auxiliary Registers
$E(\overleftarrow{x})$
initialize
… 0 0 1
$\overleftarrow{x}$
U
x_1
U
x_2
…
$P(X_2 = x_2 | \overleftarrow{x}, x_1)$
keep track of observations

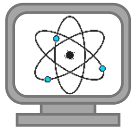
q–simulator



Quantum Topological Complexity: $d_q = \#\text{qubits}$

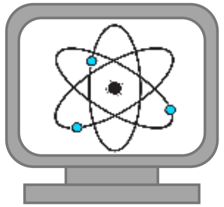
Advantage

In general:  $d_q \leq d_c$ 

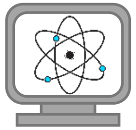
For some processes:  $d_q < d_c$ 

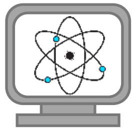
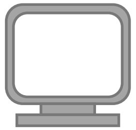
Thompson et al.,
10.1103/PhysRevX.8.031013

Advantage

In general:  $d_q \leq d_c$ 

Thompson et al., 10.1103/PhysRevX.8.031013

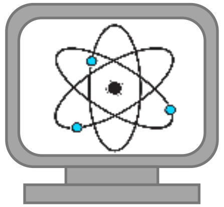
For some processes:  $d_q < d_c$ 

Approximate models:  $\hat{d}_q = \hat{d}_c$ 

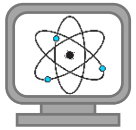
Q-Models can have better accuracy
Yang et al., arXiv:2105.14434

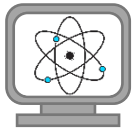
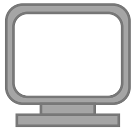
Advantage

SimTech

In general:  $d_q \leq d_c$ 

Thompson et al., 10.1103/PhysRevX.8.031013

For some processes:  $d_q < d_c$ 

Approximate models:  $\hat{d}_q = \hat{d}_c$ 

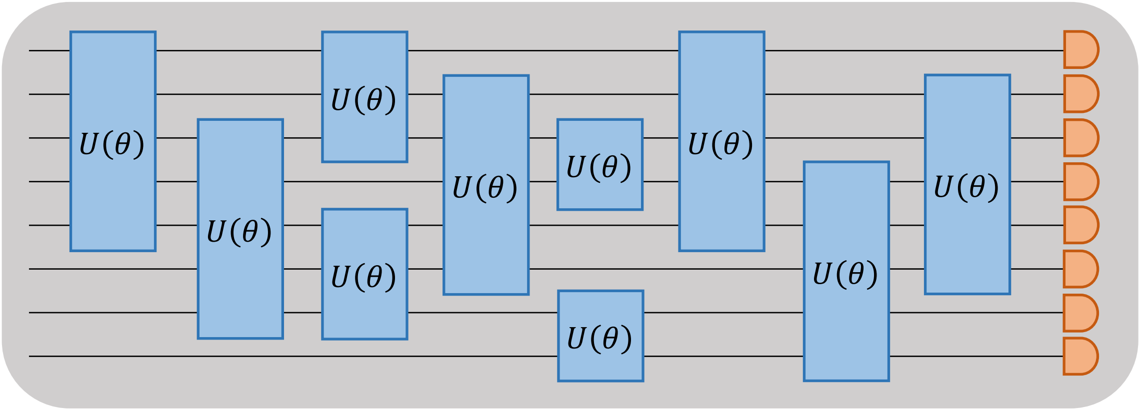
Q-Models can have better accuracy
Yang et al., arXiv:2105.14434

How to get a **quantum representation** of a quantum model?

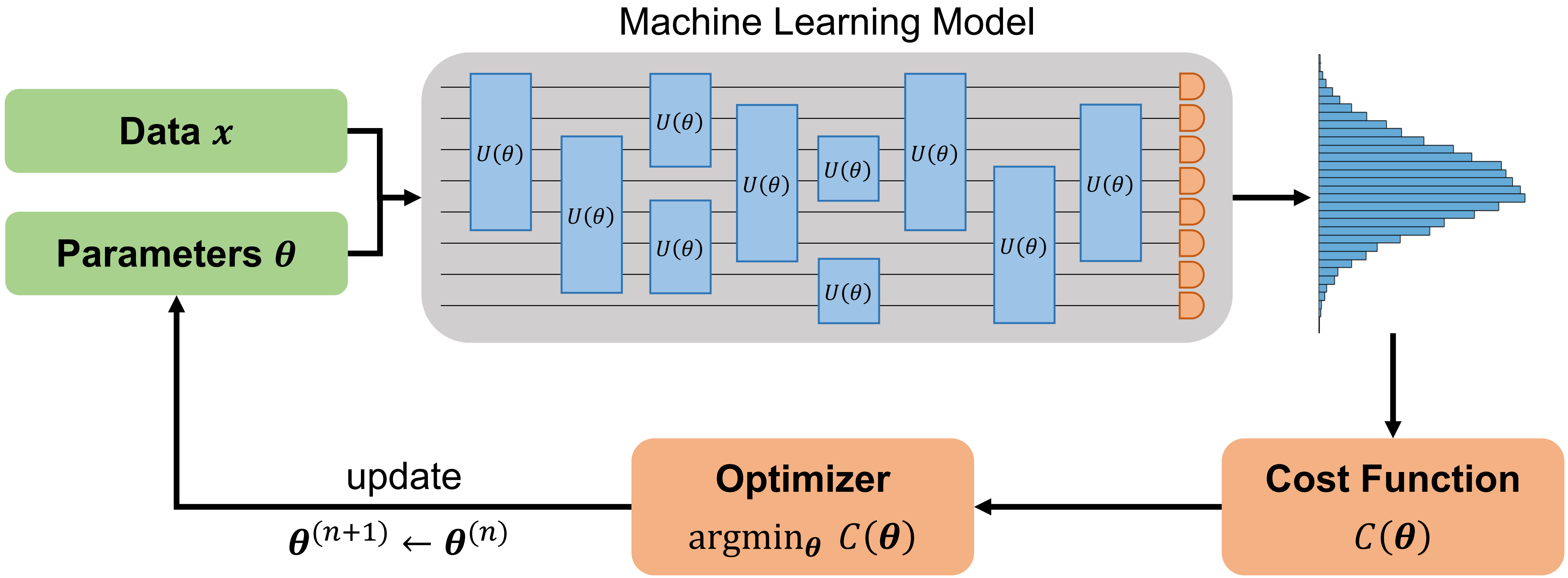
Variational Quantum Circuits

Variational Quantum Circuits

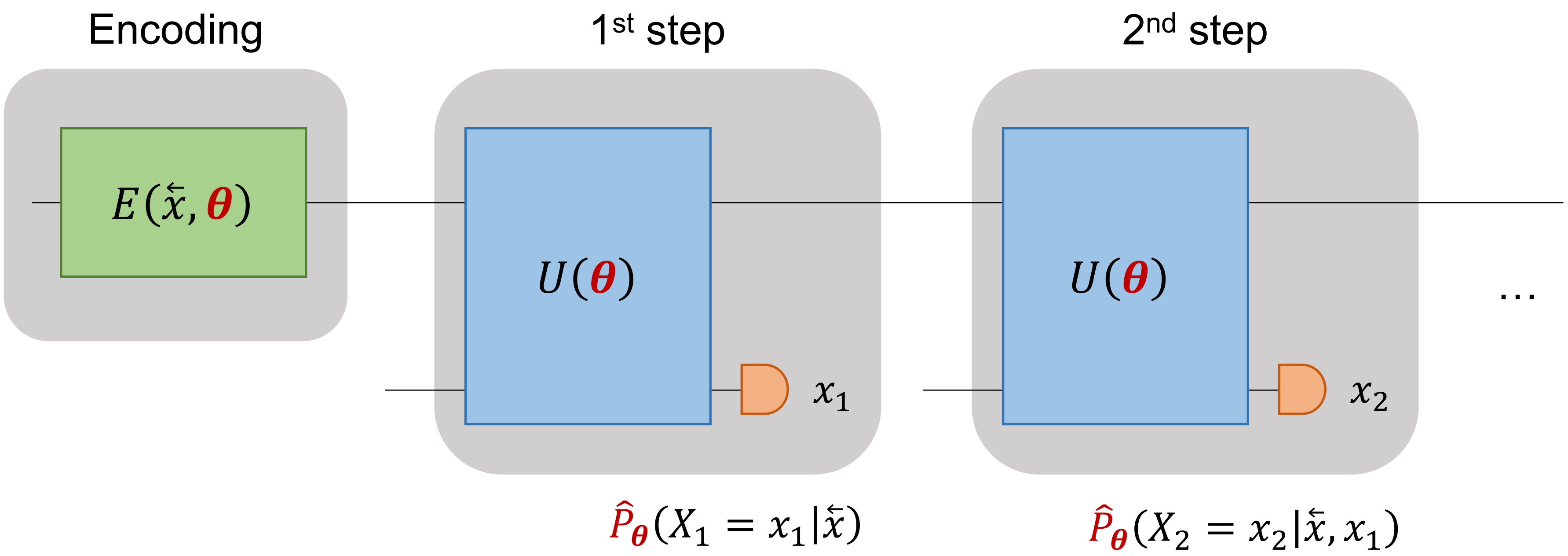
Machine Learning Model



Variational Quantum Algorithms



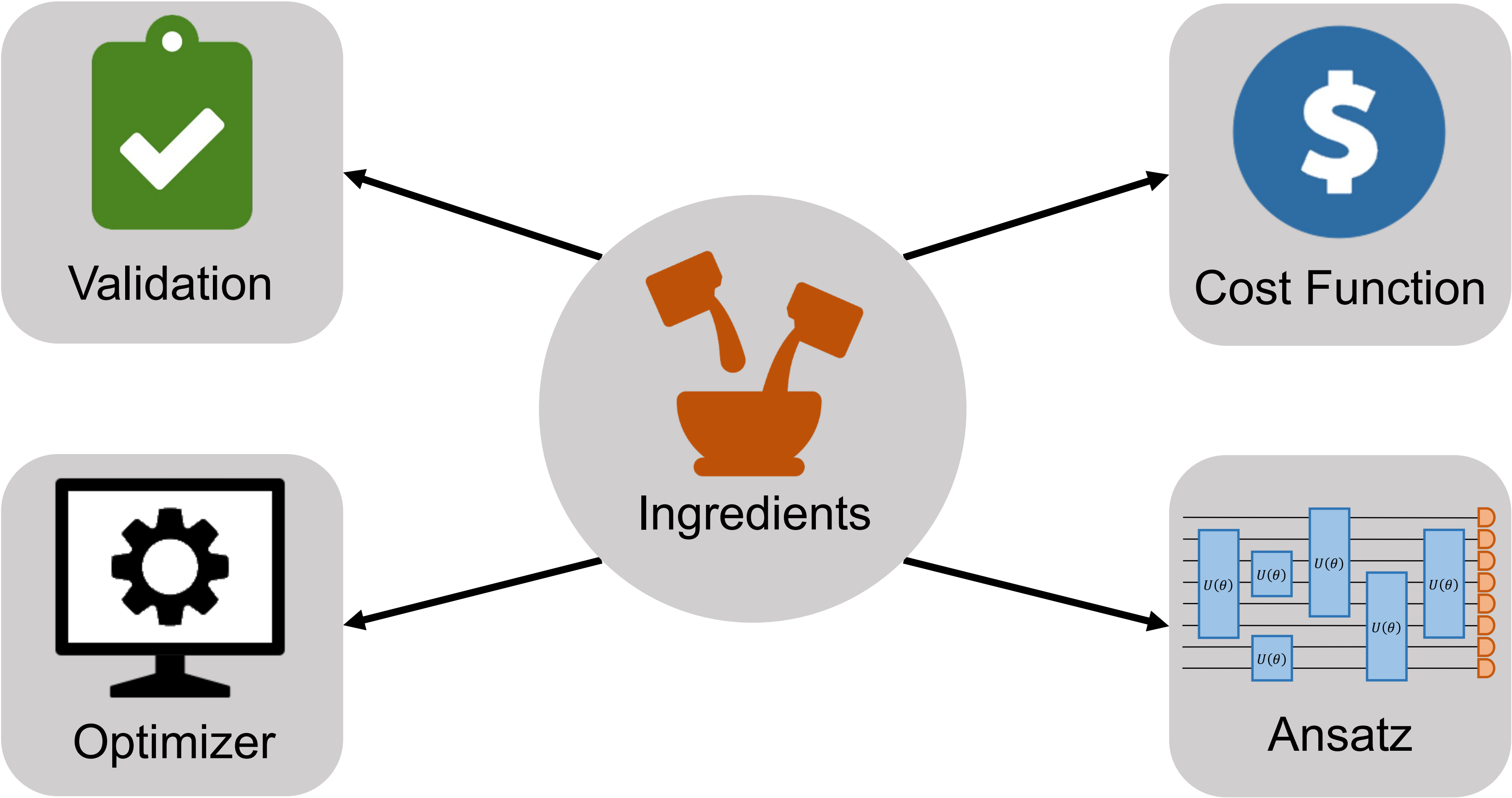
Idea



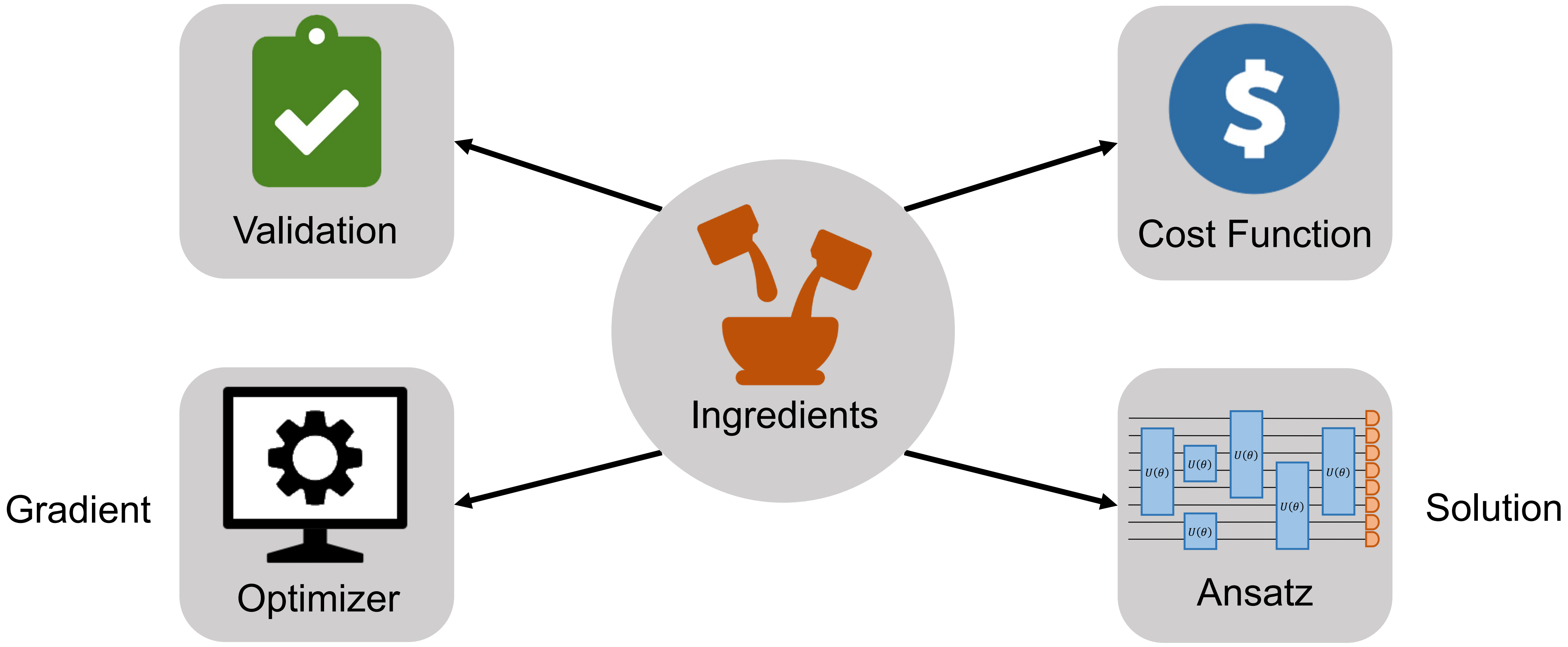
Approximate P → $|P - \hat{P}_{\boldsymbol{\theta}}| < \epsilon$

Quantum Learning Algorithm

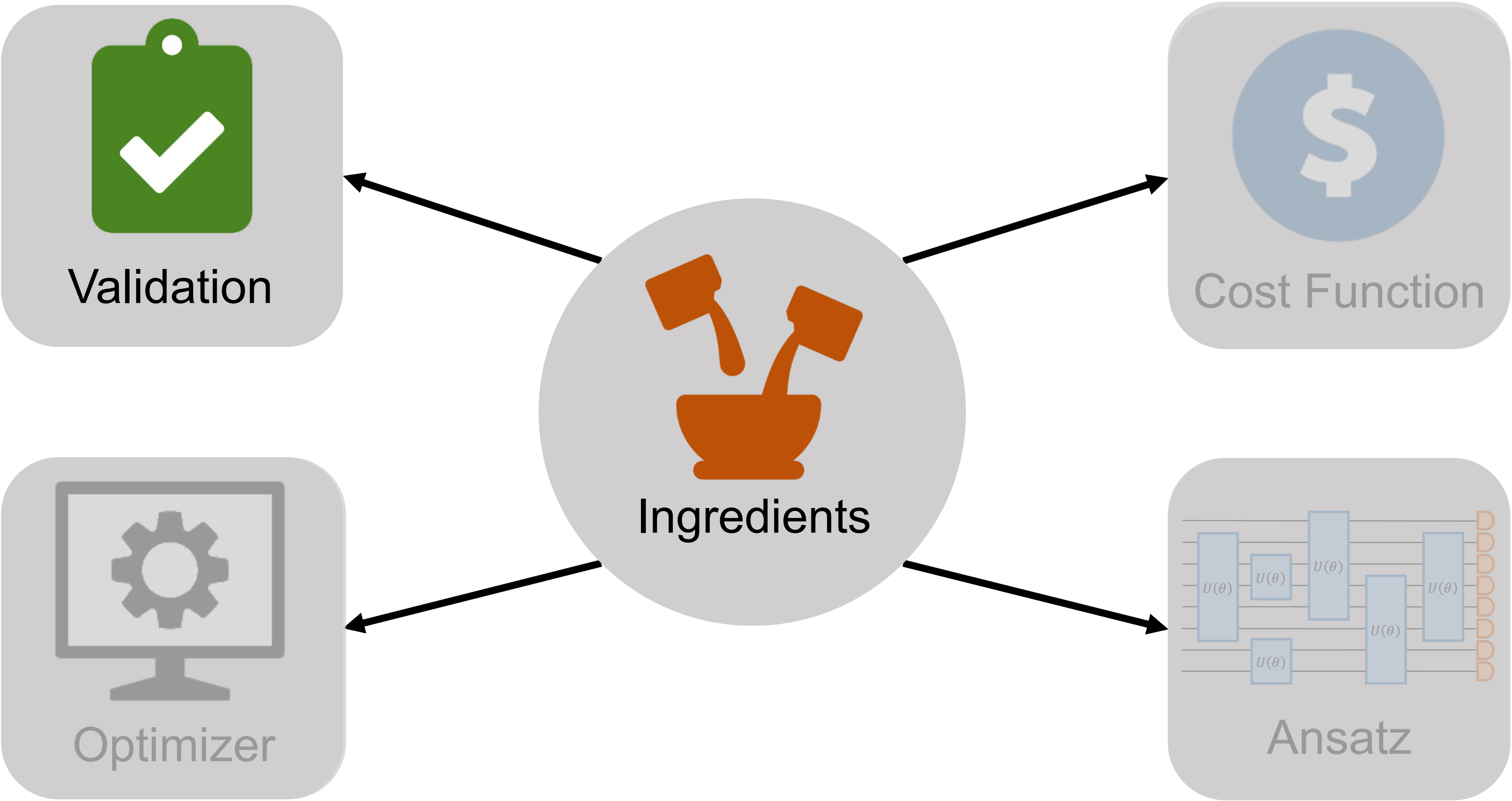
Quantum Learning Algorithm



Quantum Learning Algorithm



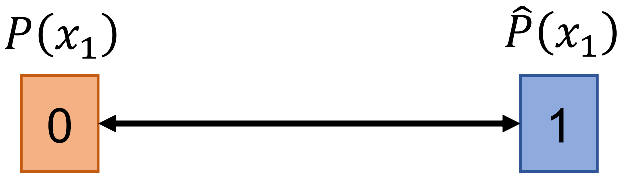
Quantum Learning Algorithm



Validation

Kullback-Leibler divergence:
(KL)

$$D_{KL}(P, \hat{P}) = \sum_x P(x) \log_2 \frac{P(x)}{\hat{P}(x)}$$



Validation

Kullback-Leibler divergence: (KL)

$$D_{KL}(P, \hat{P}) = \sum_x P(x) \log_2 \frac{P(x)}{\hat{P}(x)}$$

$P(x_1)$ 0 ⟷ 1 $\hat{P}(x_1)$

mean over time steps

average over pasts

Validation

Kullback-Leibler divergence:
(KL)

$$D_{KL}(P,\hat{P}) = \sum_x P(x)\log_2\frac{P(x)}{\hat{P}(x)}$$

$P(x_1)$ $\hat{P}(x_1)$

$0 \longleftrightarrow 1$

mean over time steps

average over pasts

Total Variation Distance:
(TV)

$$D_{TV}(P,\hat{P}) = \frac{1}{2}\sum_x \left|P(x) - \hat{P}(x)\right|$$

$P(x_1)$ $\hat{P}(x_1)$

$0 \longleftrightarrow 1$

Validation

Kullback-Leibler divergence:
(KL)

$$D_{KL}(P,\hat{P}) = \sum_x P(x)\log_2\frac{P(x)}{\hat{P}(x)}$$

$$\boxed{0} \; P(x_1) \longleftrightarrow \hat{P}(x_1) \; \boxed{1}$$

mean over time steps

average over pasts

Total Variation Distance:
(TV)

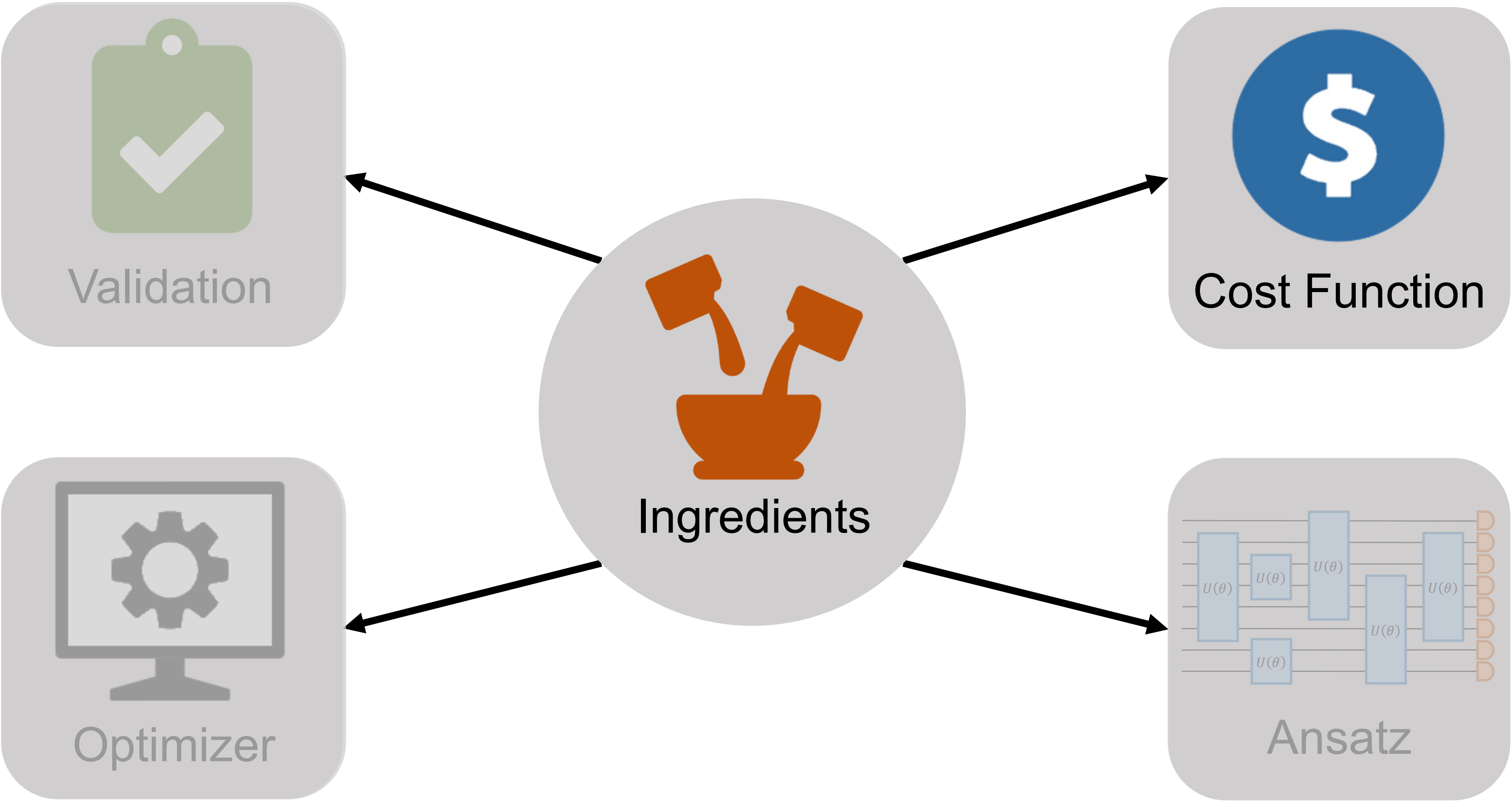
$$D_{TV}(P,\hat{P}) = \frac{1}{2}\sum_x \left|P(x) - \hat{P}(x)\right|$$

$$\boxed{0} \; P(x_1) \longleftrightarrow \hat{P}(x_1) \; \boxed{1}$$

sum up time steps

sum up pasts

Quantum Learning Algorithm



Cost Function

Ideally, use validation metric: $$D_{KL}(P,\hat{P}) = \sum_x P(x)\log_2 \frac{P(x)}{\hat{P}(x)}$$

Cost Function

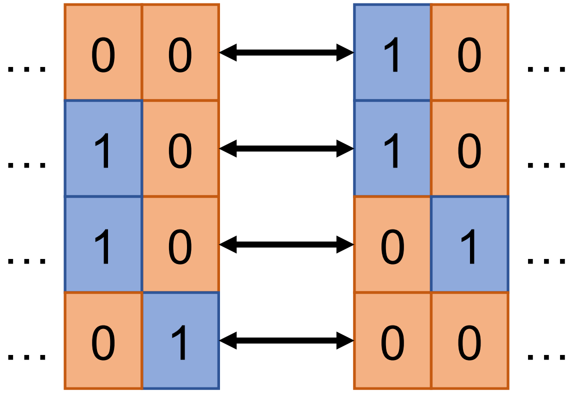
Ideally, use validation metric: $$D_{KL}(P,\hat{P}) = \sum_x P(x)\log_2 \frac{P(x)}{\hat{P}(x)}$$

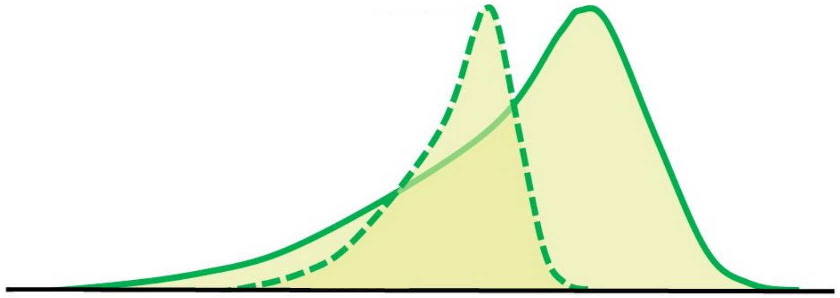
unknown

inefficient

Cost Function

Maximum Mean Discrepancy:
(MMD)

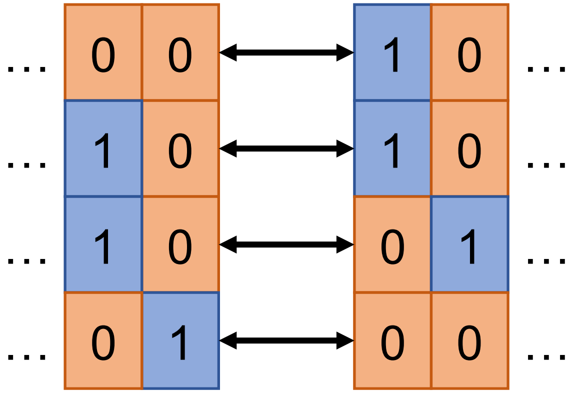


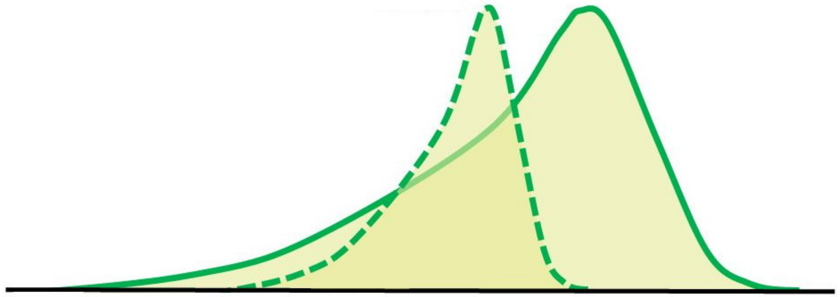


$$\mathrm{MMD}\left[P, \hat{P}\right] = 0 \quad \Leftrightarrow \quad P = \hat{P}$$

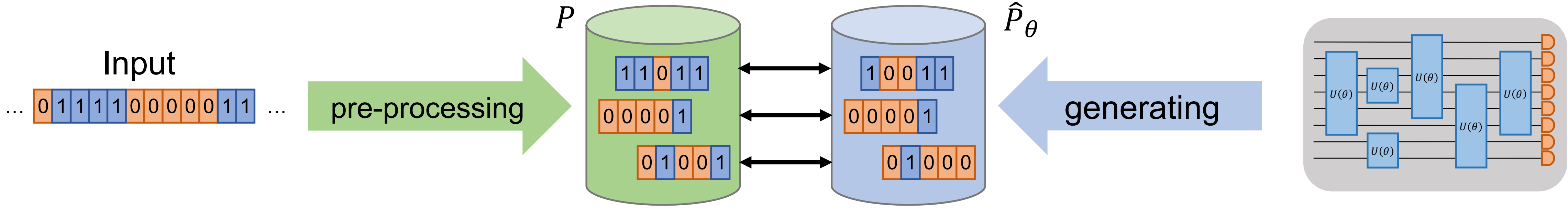
Cost Function

Maximum Mean Discrepancy:
(MMD)



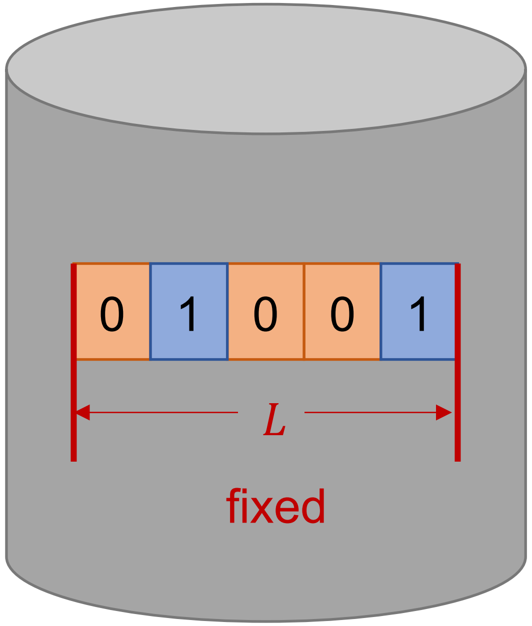


$$\mathrm{MMD}\left[P, \hat{P}\right] = 0 \quad \Leftrightarrow \quad P = \hat{P}$$



Cost Function

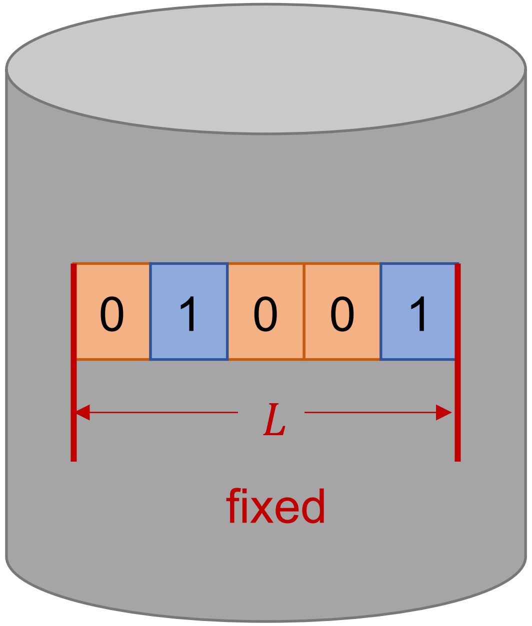
$$C(\boldsymbol{\theta}) = \sum_{\overleftarrow{x}} w_{\overleftarrow{x}} \cdot \mathrm{MMD}^2[\mathrm{P}, \hat{P}_{\boldsymbol{\theta}}|\overleftarrow{x}]$$



Cost Function

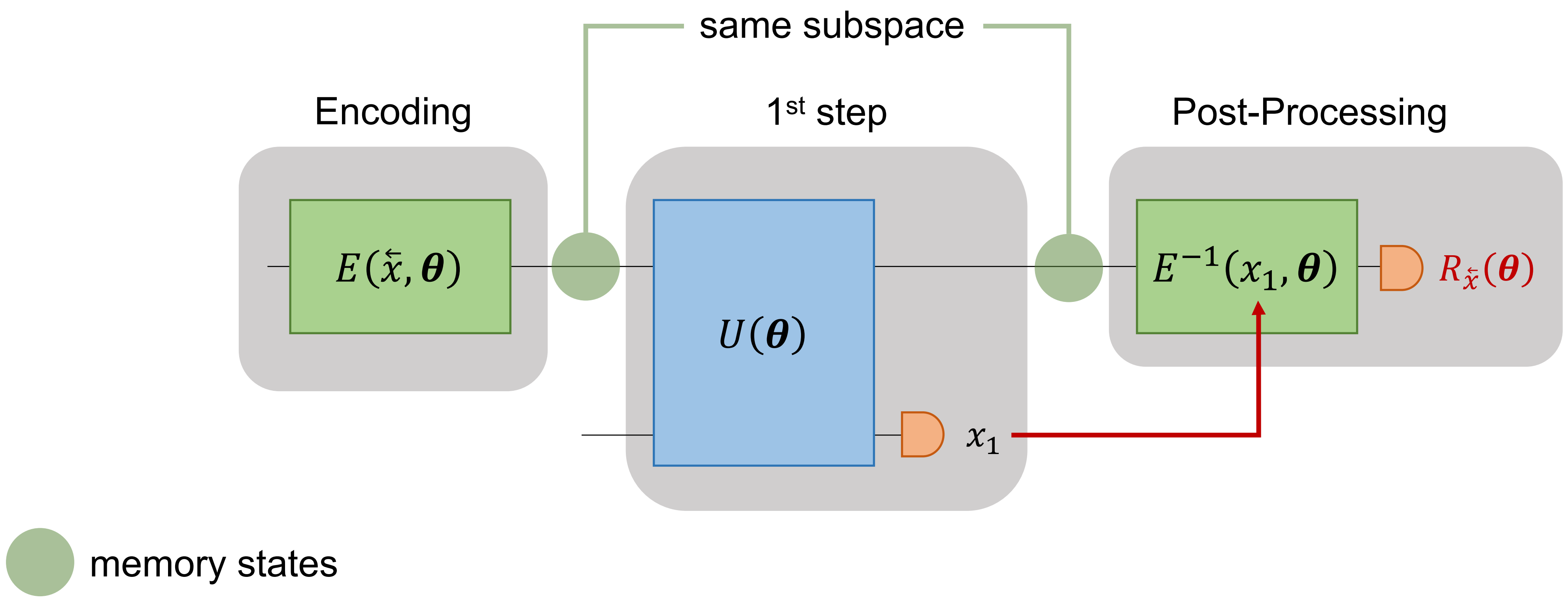
$$C(\boldsymbol{\theta}) = \sum_{\overleftarrow{x}} w_{\overleftarrow{x}} \cdot \mathrm{MMD}^2[\mathrm{P}, \hat{P}_{\boldsymbol{\theta}} | \overleftarrow{x}]$$

$$C(\boldsymbol{\theta}) = \sum_{\overleftarrow{x}} w_{\overleftarrow{x}} \cdot \mathrm{MMD}^2[\mathrm{P}, \hat{P}_{\boldsymbol{\theta}} | \overleftarrow{x}] + R_{\overleftarrow{x}}(\boldsymbol{\theta})$$



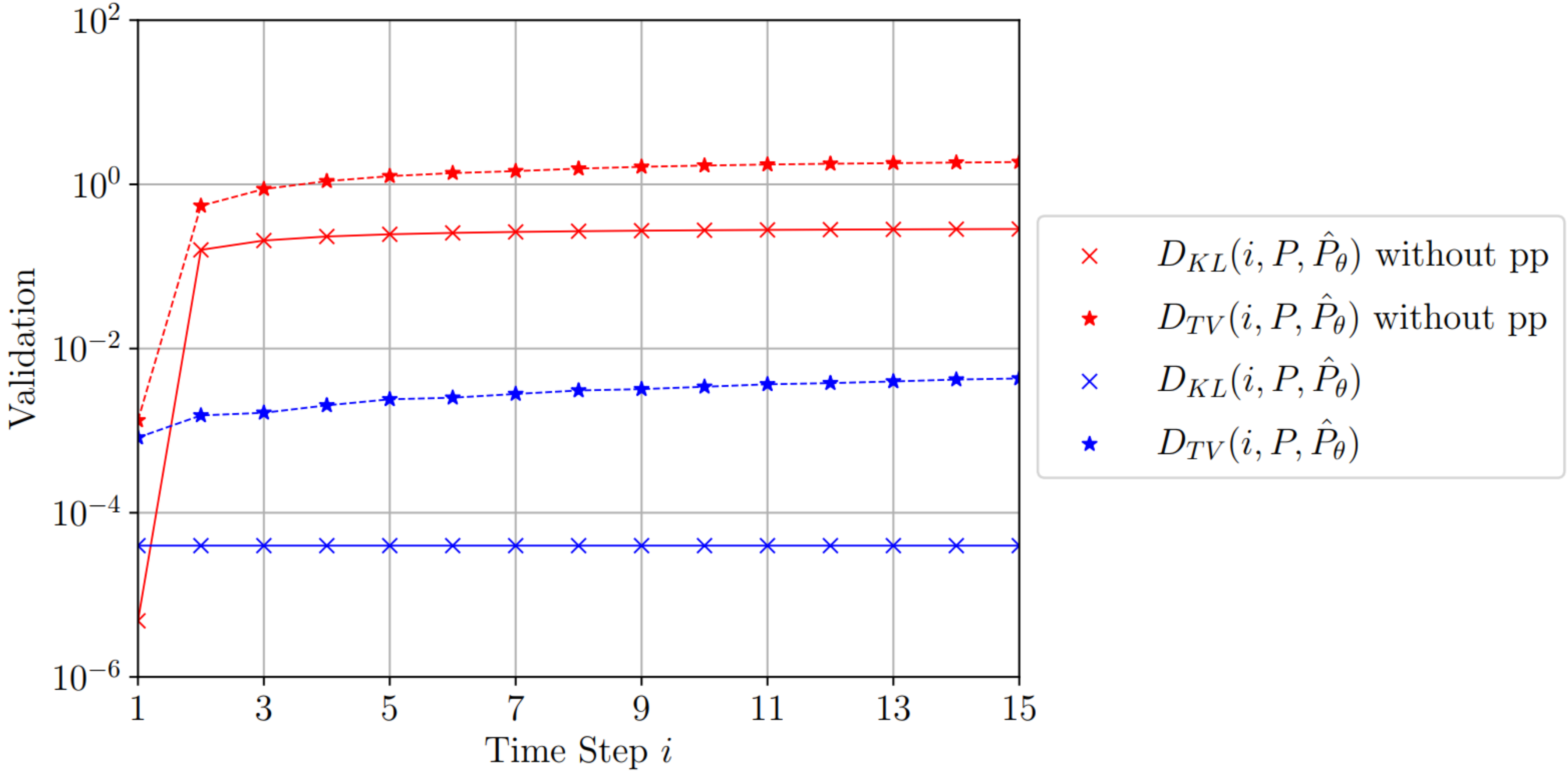
Regularization = penalizes models with a large set of memory states

Cost Function



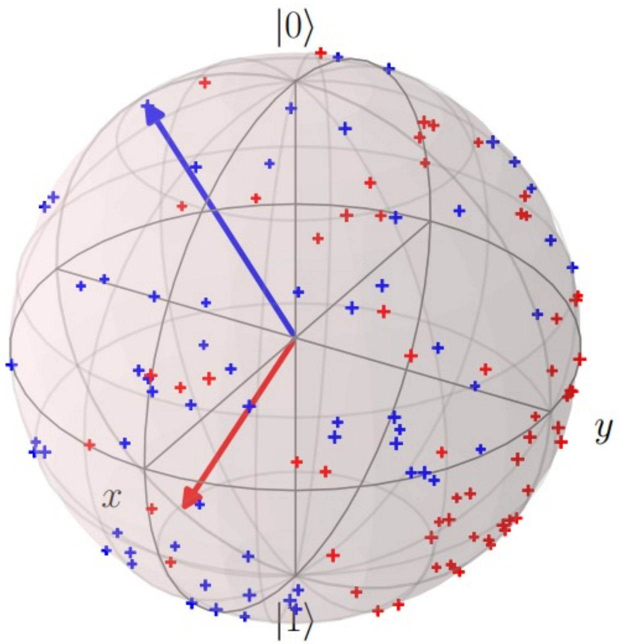
Results

Results


Validation
Time Step i
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
1
3
5
7
9
11
13
15
$D_{KL}(i, P, \hat{P}_\theta)$ without pp
$D_{TV}(i, P, \hat{P}_\theta)$ without pp
$D_{KL}(i, P, \hat{P}_\theta)$
$D_{TV}(i, P, \hat{P}_\theta)$

Results

random



$E(\overleftarrow{x}, \boldsymbol{\theta})$

initial state for $\overleftarrow{x} = 0$

initial state for $\overleftarrow{x} = 1$

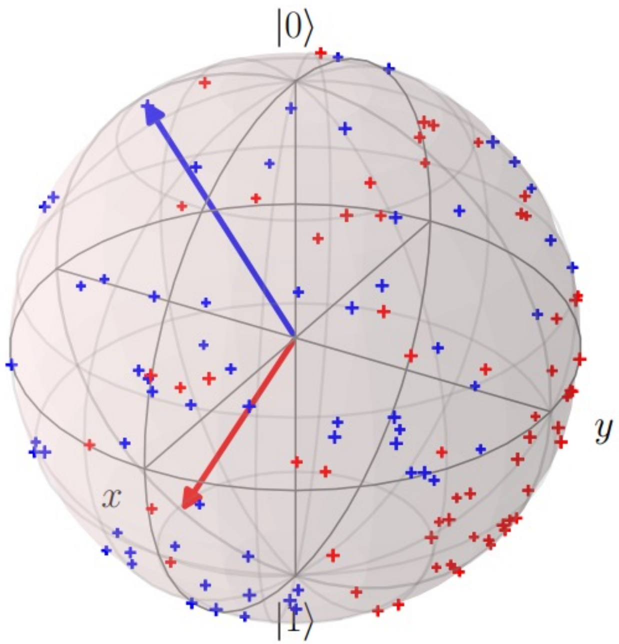
memory states for $x_i = 0$

memory states for $x_i = 1$

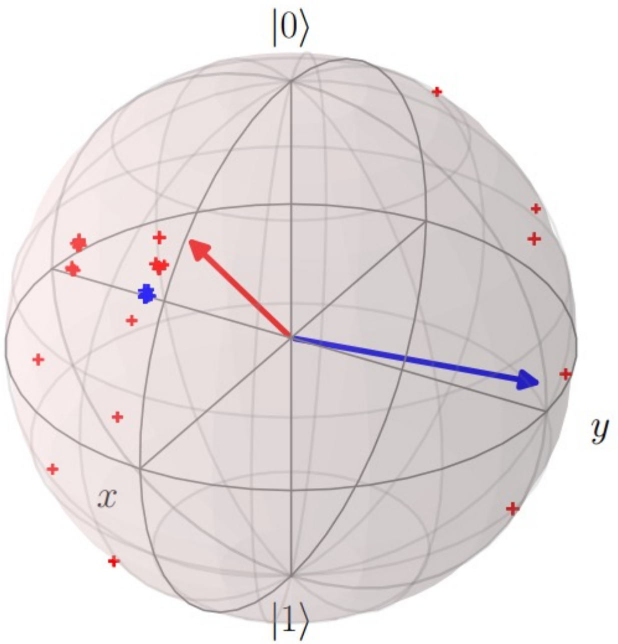
$U(\boldsymbol{\theta})$ | $U(\boldsymbol{\theta})$ | ...

Results

random



without pp



$E(\overleftarrow{x}, \boldsymbol{\theta})$

initial state for $\overleftarrow{x} = 0$

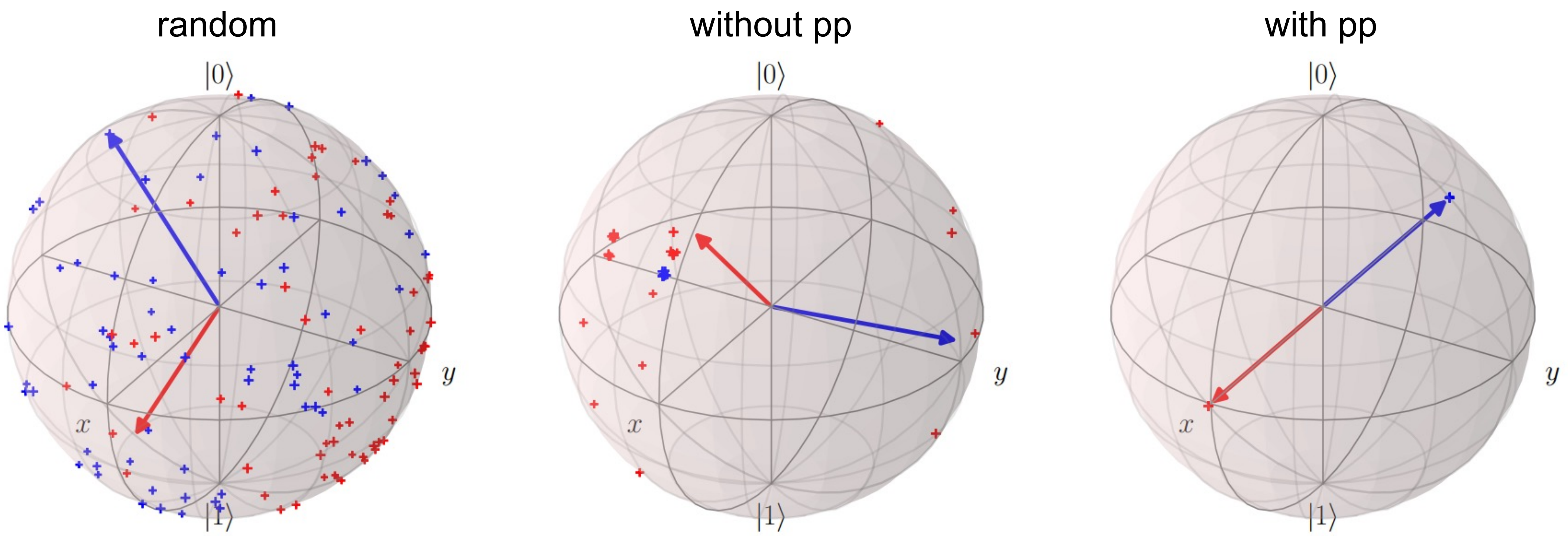
initial state for $\overleftarrow{x} = 1$

memory states for $x_i = 0$

memory states for $x_i = 1$

$U(\boldsymbol{\theta})$ | $U(\boldsymbol{\theta})$ | ...

Results



$E(\overleftarrow{x}, \boldsymbol{\theta})$

initial state for $\overleftarrow{x} = 0$

initial state for $\overleftarrow{x} = 1$

memory states for $x_i = 0$

memory states for $x_i = 1$

$U(\boldsymbol{\theta})$ | $U(\boldsymbol{\theta})$ | …

Conclusion
&
Outlook

Conclusion

- Developed a hybrid quantum learning algorithm for simulation models
- Learning algorithm is memory efficient
- MMD can decrease KL and TV
- Regularization → small set of memory states
- Learned models show constantly good simulation performance

Outlook

- arXiv:2105.14434: Lower bound onto the KL divergence of any classical model
- Apply the algorithm to more complicated processes

  → Showing "quantum advantage"

- Use only data, i.e., no analytical solutions

  → Create a classical version of the learning algorithm (quantum inspired)

  → Compare quantum approximate models with classical approximate models:

  Learning speed, Barren plateaus, simulation accuracy

Libraries & Tools

Vendors

- Google
- rigetti
- IONQ
- IBM
- D-Wave – The Quantum Computing Company™
- Honeywell
- qci
- AQT
- Microsoft
- XANADU

SDKs

Cirq

Amazon Braket aws

Q#

Qiskit

Ocean

Tools & Libraries

- Quirk: Prototyping + Visualization + Dynamic
- Link

- QuTIP: Easy numerical calculations + Visualization
- Link

- PennyLane: Quantum Machine Learning + almost cross-platform

→ Qiskit, Braket, Cirq, QDK, IonQ, …, Numpy, TensorFlow, PyTorch, JAX

People & Community

People & Community

Quantum Machine Learning

Seth Loyd (MIT)

Maria Schuld (Xanadu)

John Preskill (Caltech)

Jay M. Gambetta (IBM)

Amira Abbas (Google)

Jens Eisert (FUB)

…

My Goals

My Goals

"Present a holistic picture of whether, and if so, how, quantum devices offer a practical advantage for simulating stochastic processes in real-world scenarios."

My Goals

"Present a holistic picture of whether, and if so, how, quantum devices offer a practical advantage for simulating stochastic processes in real-world scenarios."

My Goals

SimTech

"Present a holistic picture of whether, and if so, how, quantum devices offer a practical advantage for simulating stochastic processes in real-world scenarios."

"Build a bridge between (quantum) simulation models for stochastic processes and the field of machine learning."

My Goals

"Present a holistic picture of whether, and if so, how, quantum devices offer a practical advantage for simulating stochastic processes in real-world scenarios."

"Build a bridge between (quantum) simulation models for stochastic processes and the field of machine learning."

→ PAC framework, generalization bounds, …

Thank you very much!

Let's discuss.