University of Stuttgart
Institute for Computational Physics

Freie Universität Berlin

SimTech

Simulating Stochastic Processes with Variational Quantum Circuits

Daniel Fink

Master Thesis

-

Presentation

February 14th, 2022

Advisors

- University of Stuttgart
  - Prof. Dr. Christian Holm

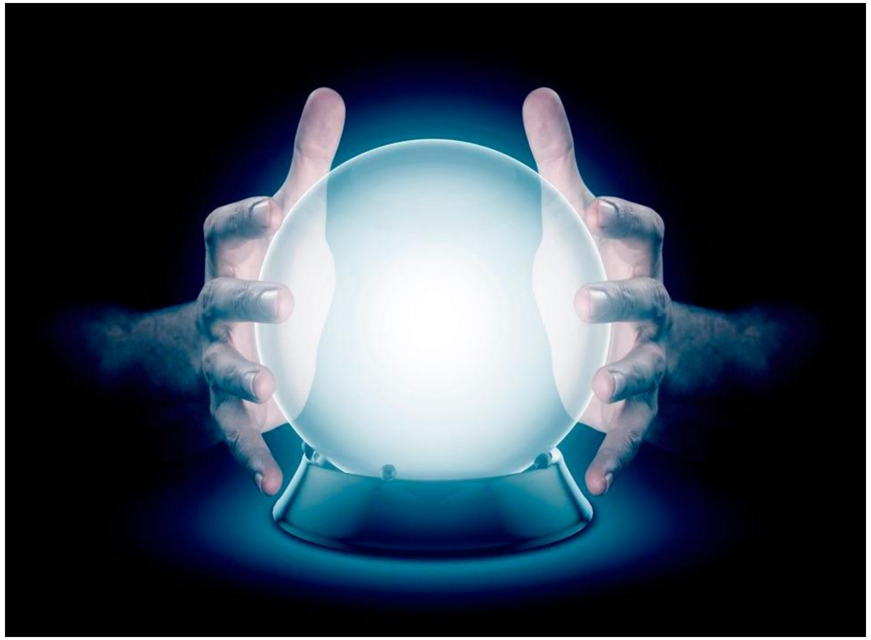
- Free University of Berlin
  - Prof. Dr. Jens Eisert
  - Dr. Nora Tischler
  - Dr. Ryan Sweke
  - M.Sc. Paul Fährmann

University of Stuttgart
Institute for Computational Physics

Freie Universität Berlin

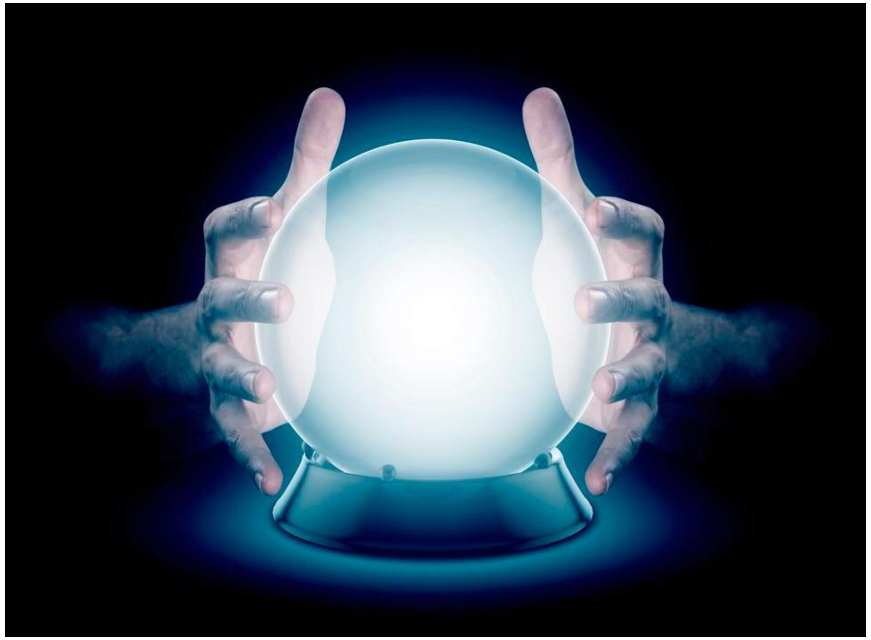
Motivation

Can we predict the future based on past observations?



Motivation

Can we predict the future based on past observations?



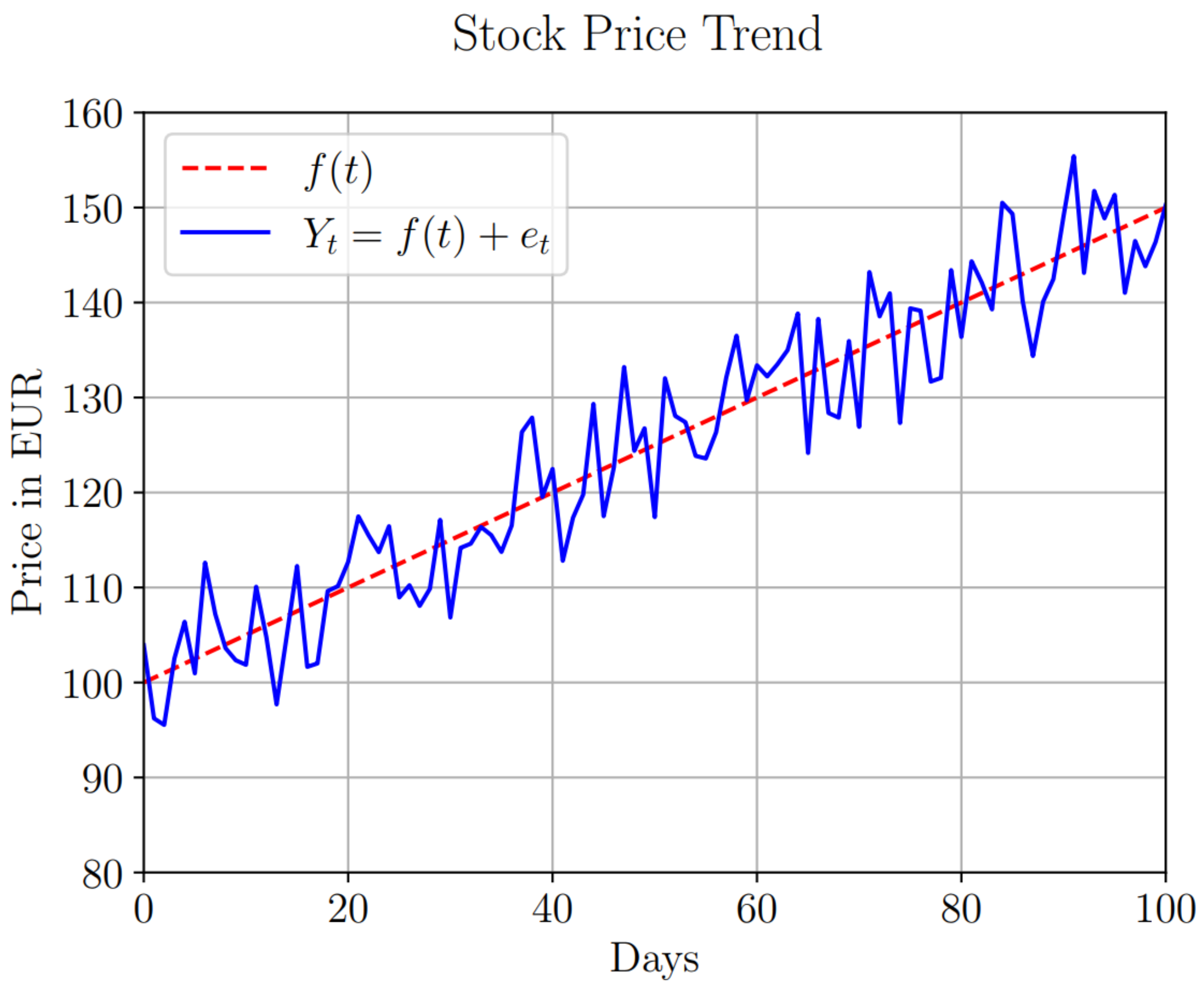
Simulations → show possible futures

Motivation

Assume linear trend $f(t)$

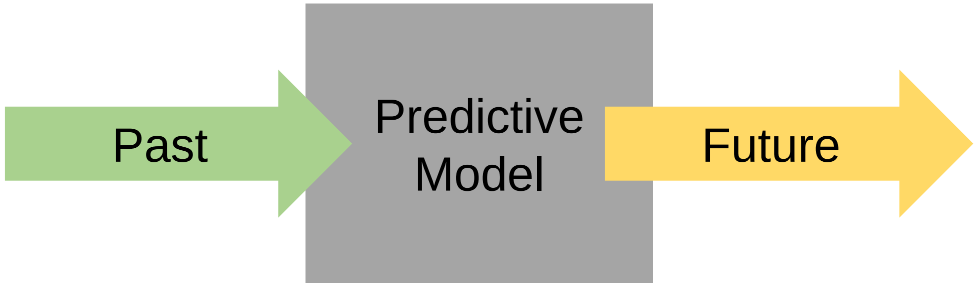
Add some noise e_t

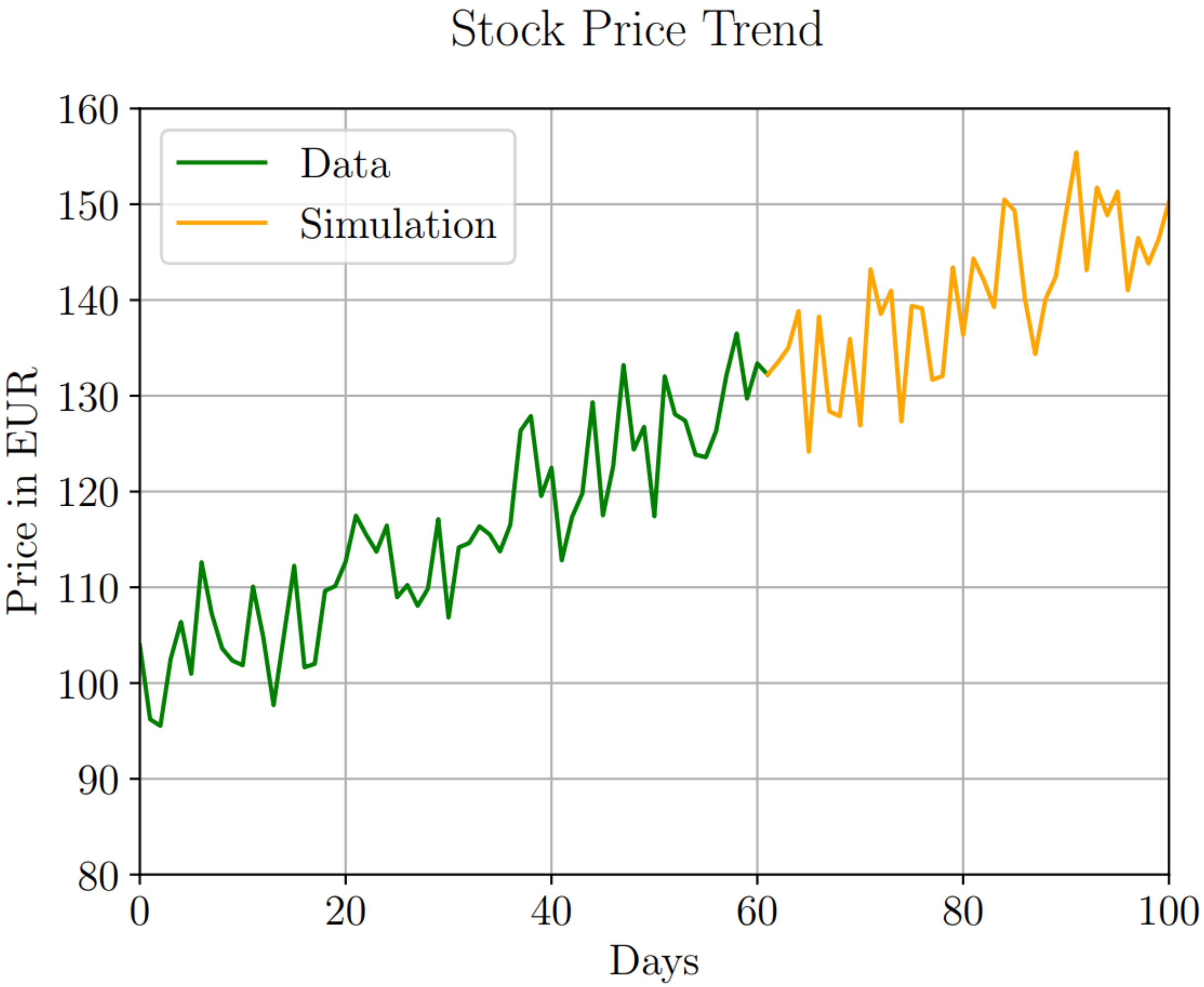
→ e_t is a stochastic process



Motivation

Assume data drawn
by a stochastic process





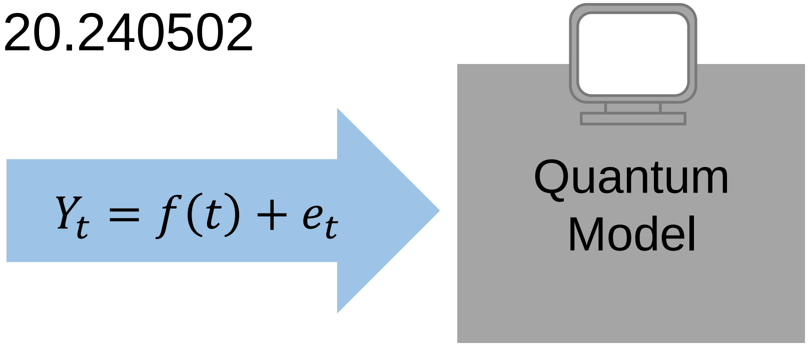
Motivation

Classical Models $\leq$ Quantum Models

How to get a quantum model?

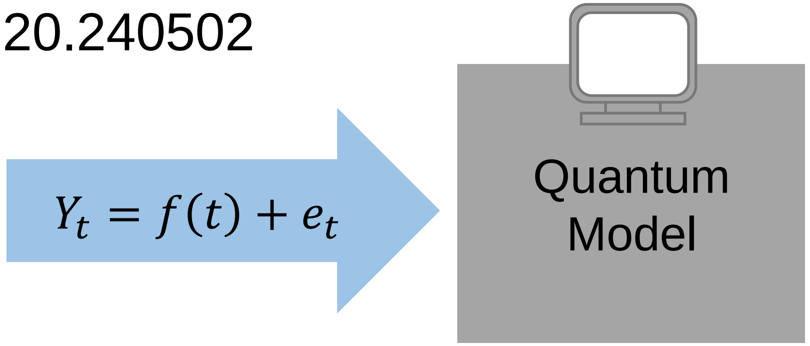
Motivation

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

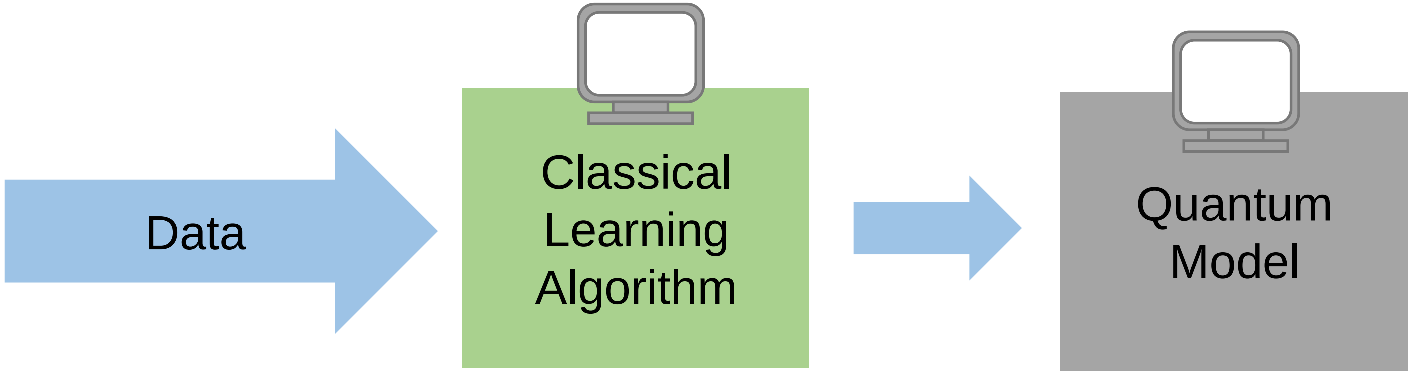


Motivation

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



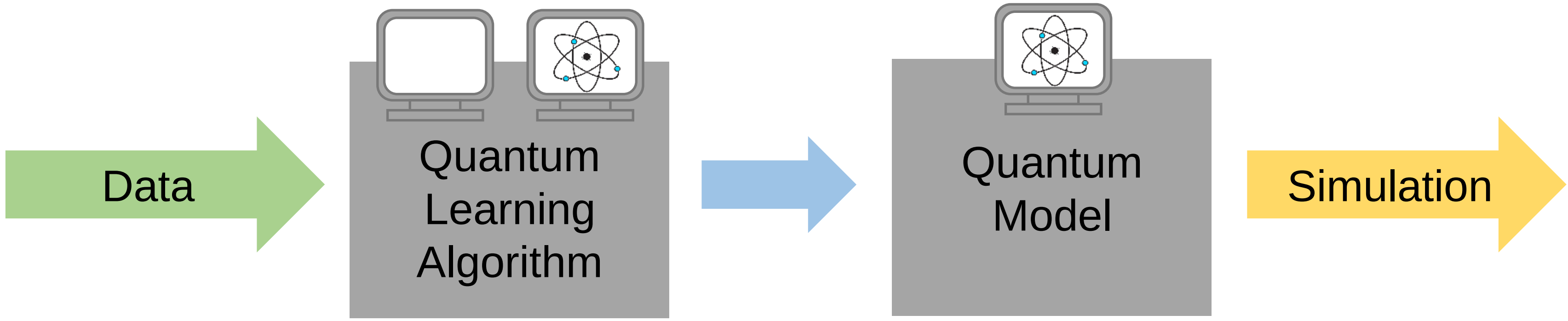
- Data from the process → classical discovery algorithm
  - Yang et al., arXiv:2105.14434



Motivation

Goal

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



Content

- Stochastic Processes
- ϵ–machine
- Quantum Circuits
- q–simulator
- Quantum Learning Algorithm
- Results
- Conclusion

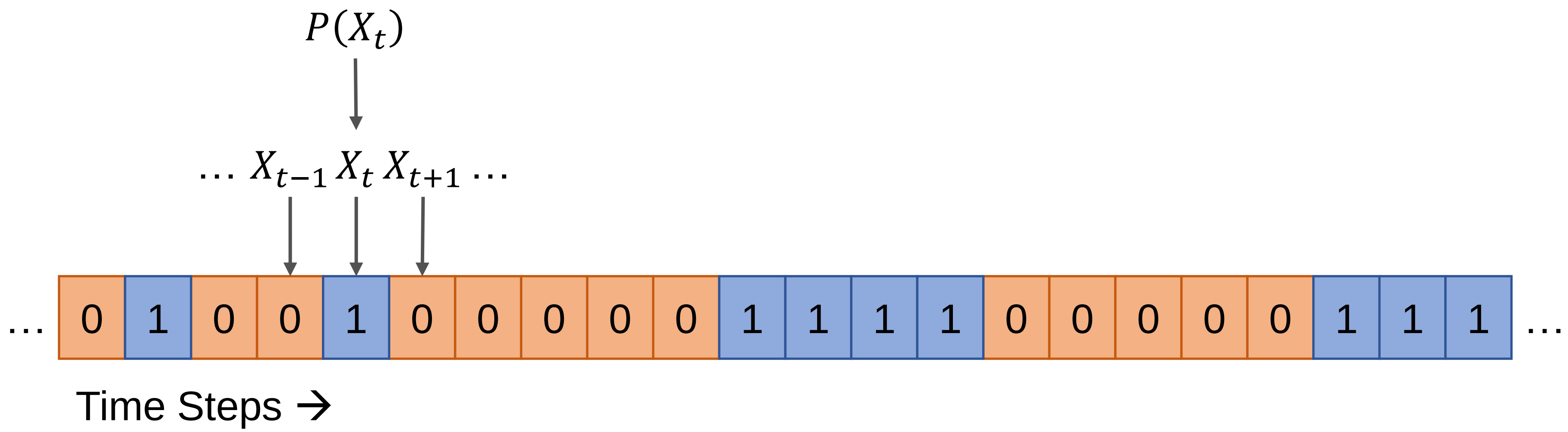
Stochastic Processes

Stochastic Processes

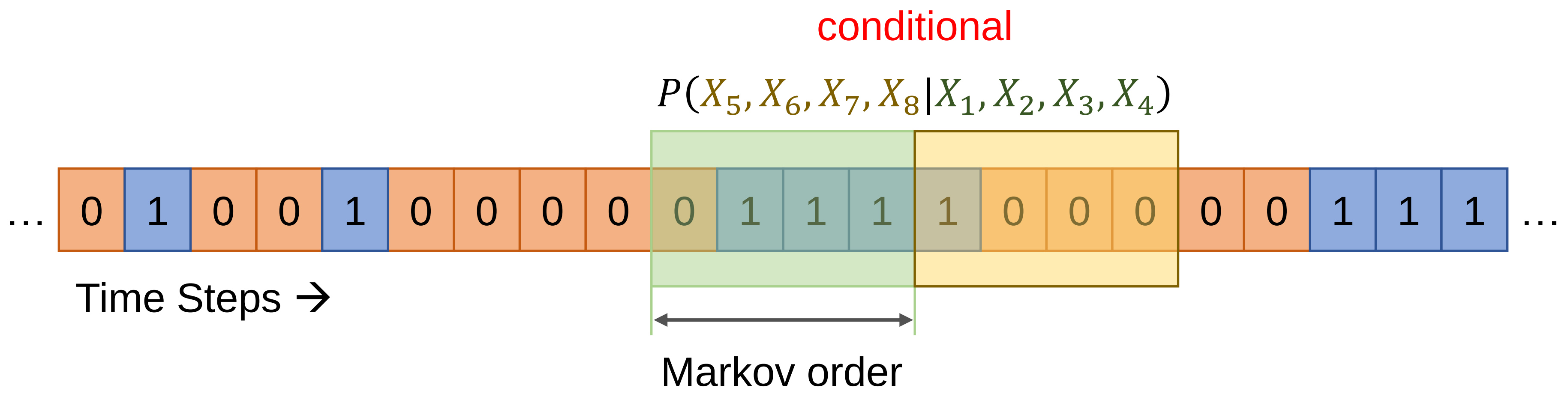
… | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | …

Time Steps →

Stochastic Processes

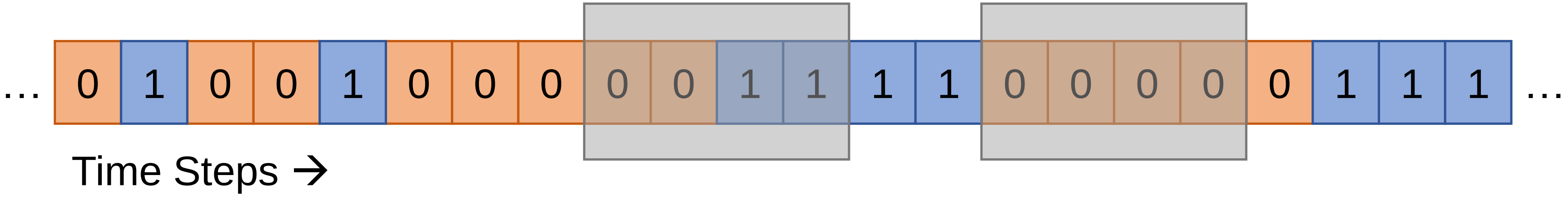


Stochastic Processes

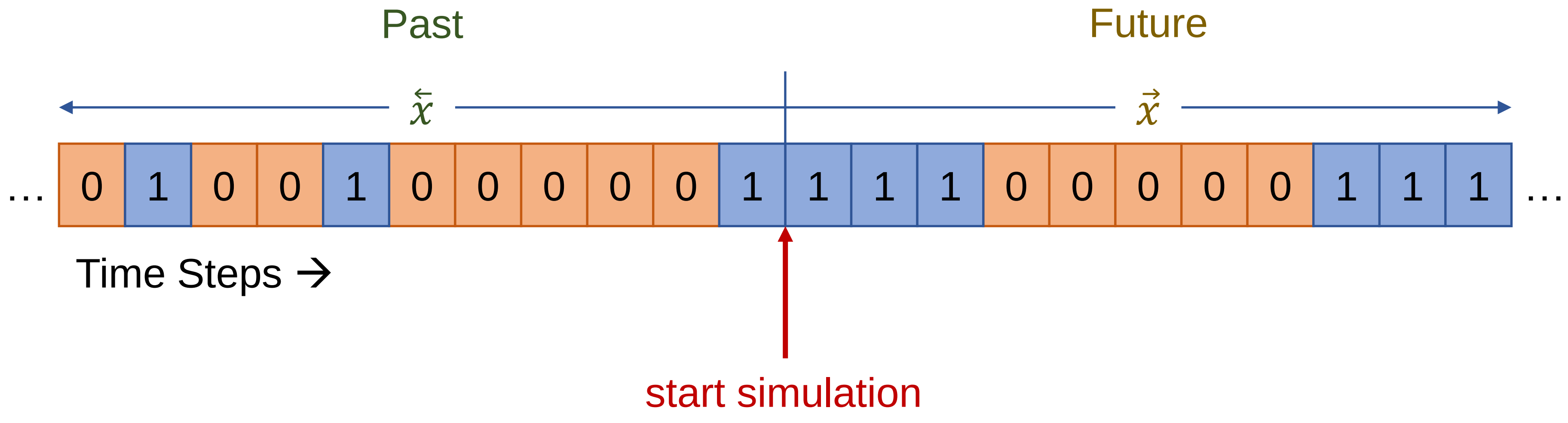


stationary

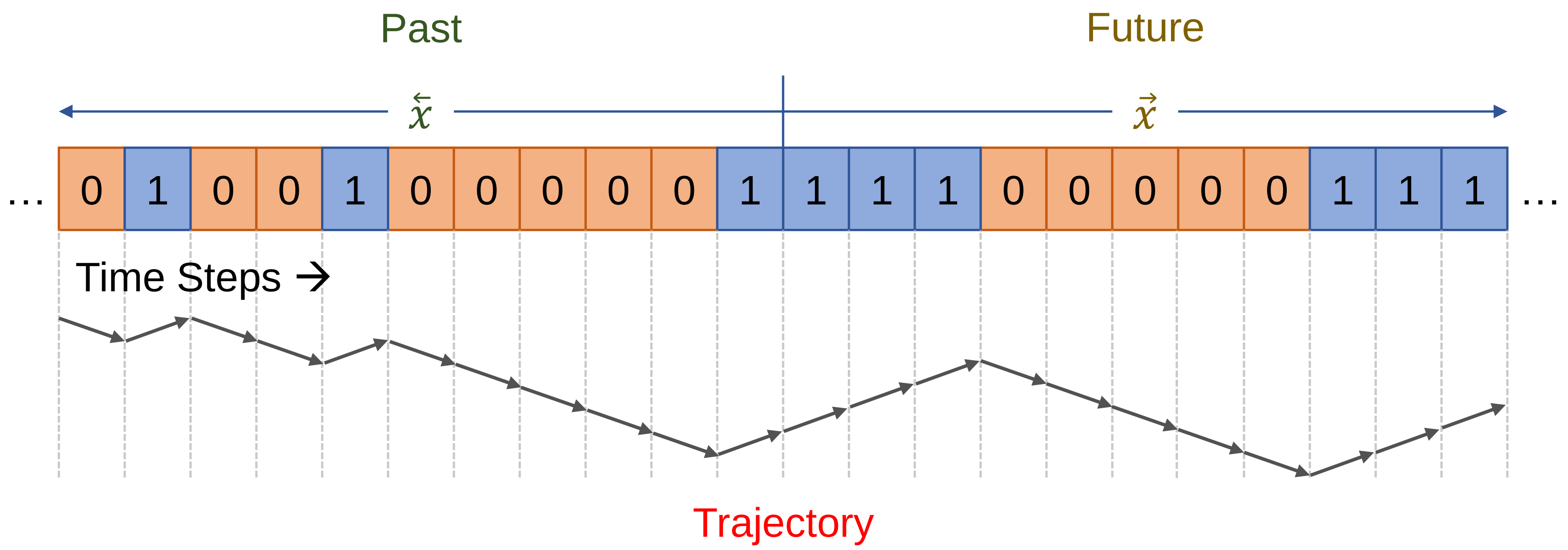
$$P(X_1, X_2, X_3, X_4) \quad = \quad P(X_5, X_6, X_7, X_8)$$



Stochastic Processes

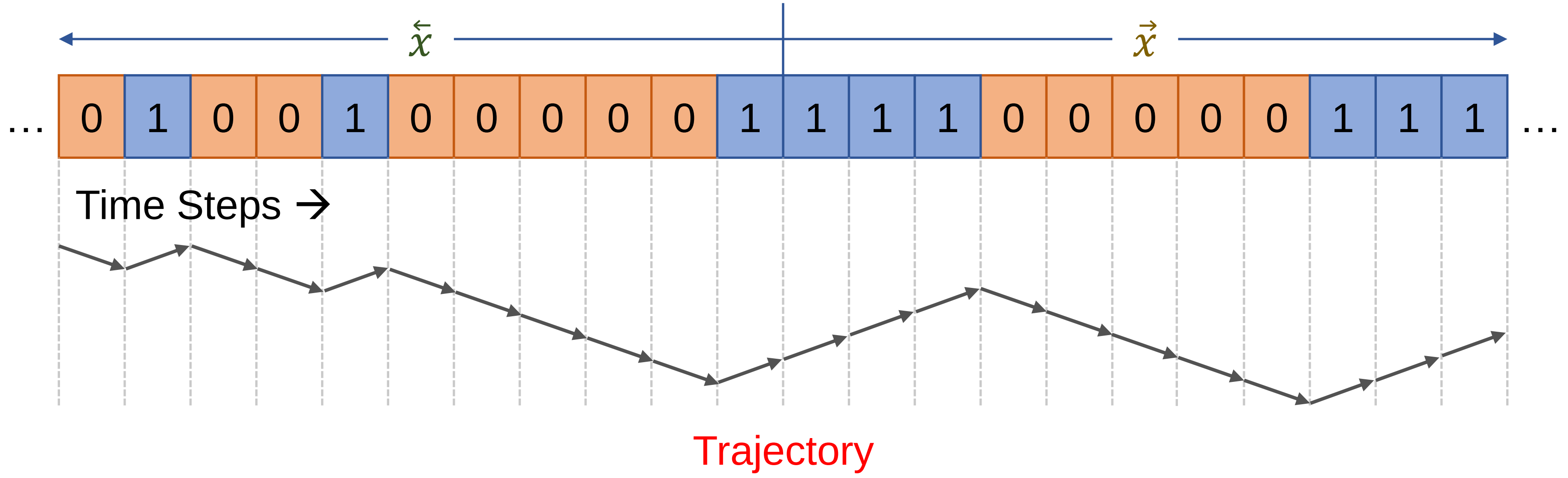


Stochastic Processes



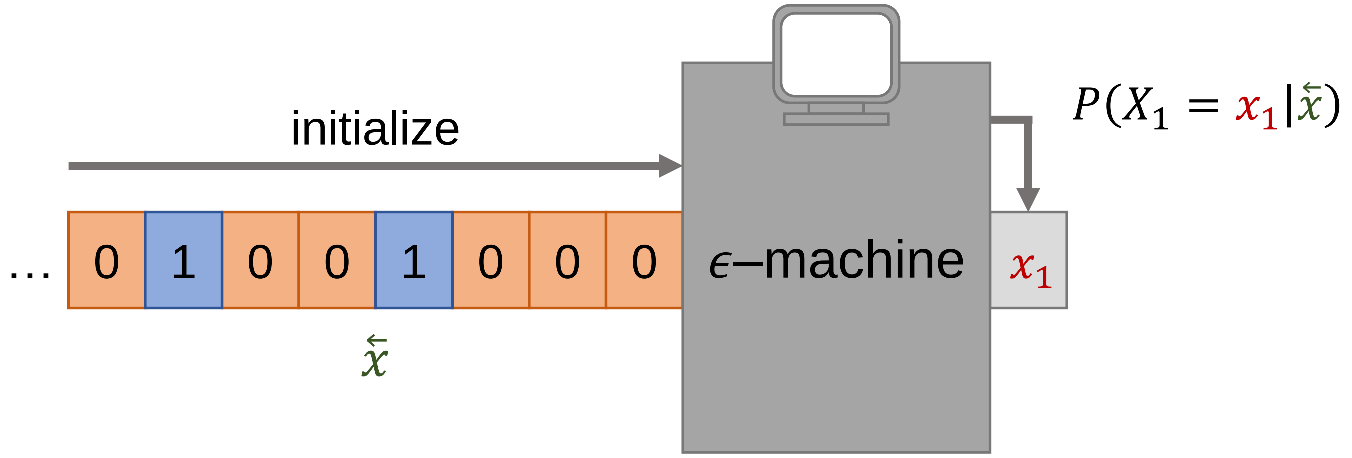
Stochastic Processes

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

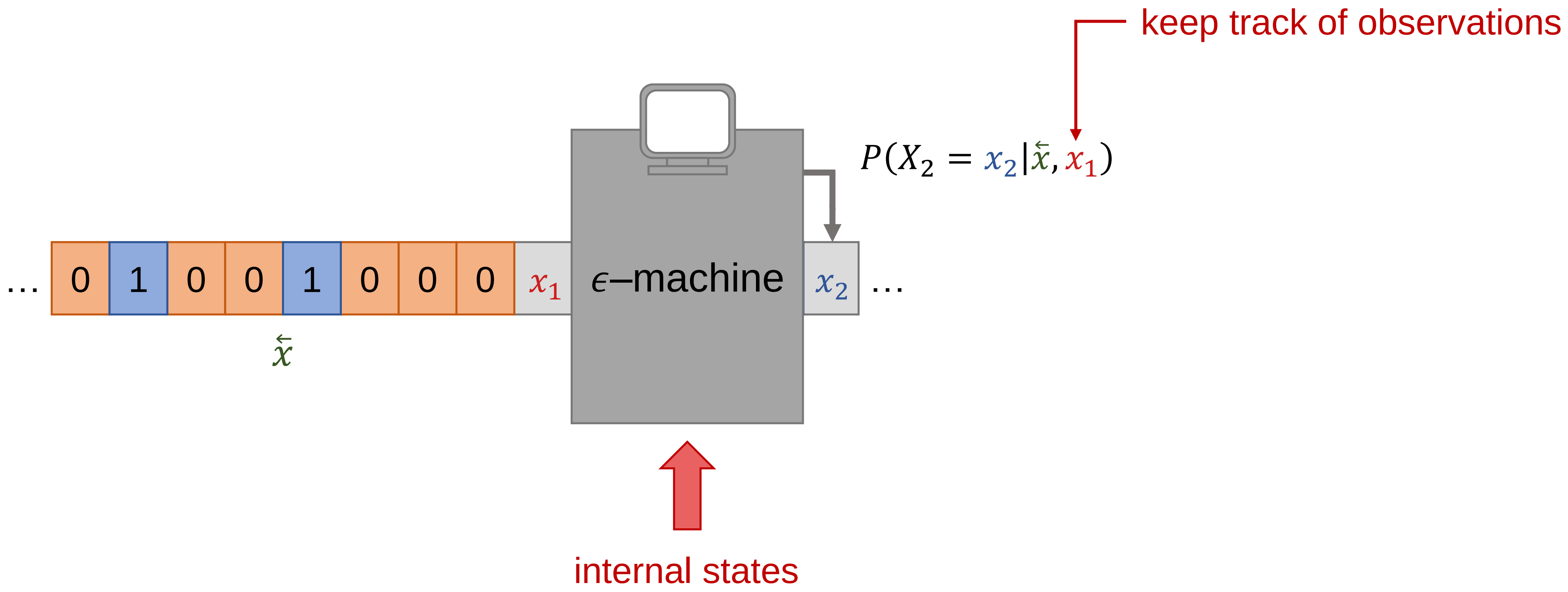


ϵ–machine

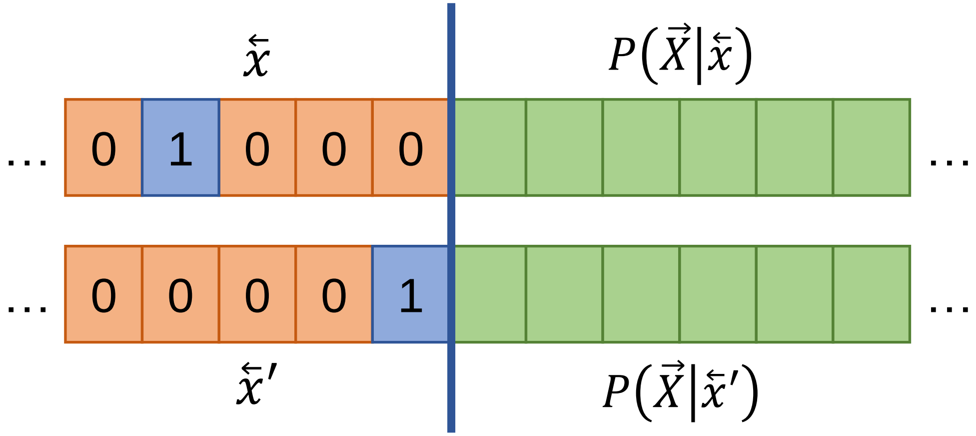
ϵ–machine



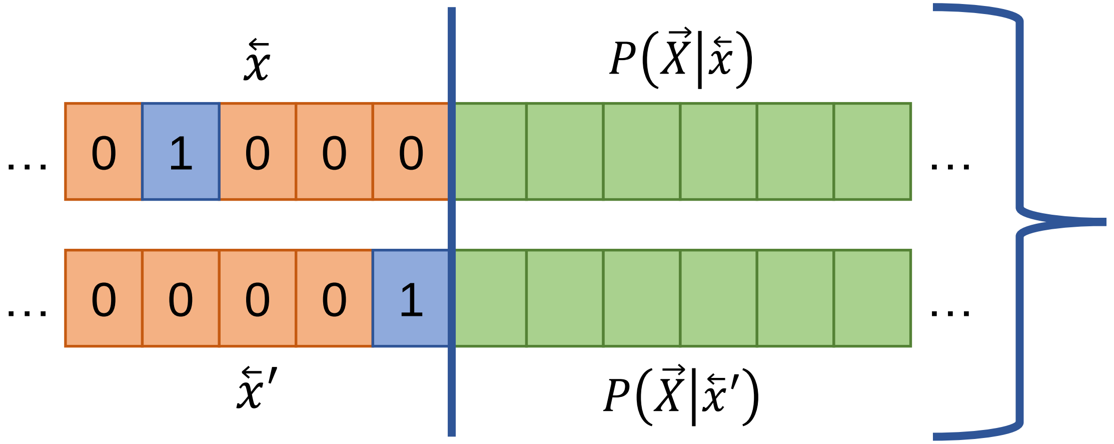
ϵ–machine



ϵ–machine

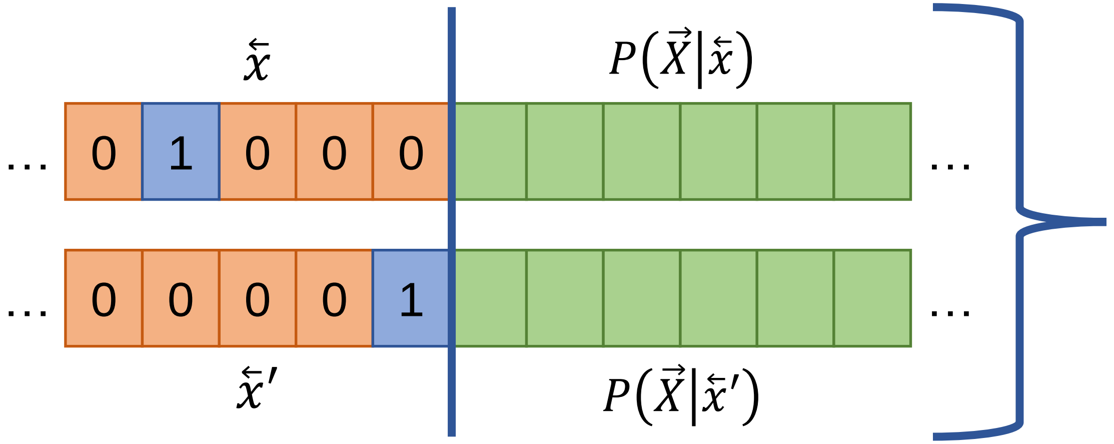


ϵ–machine



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

ϵ–machine



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

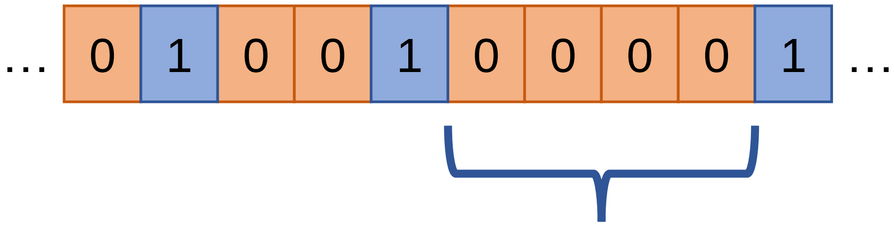
→ causal sates $\{S_i\}_{i=1}^{N}$ (minimal)

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

Classical Topological Complexity: $d_c = \log_2 N$

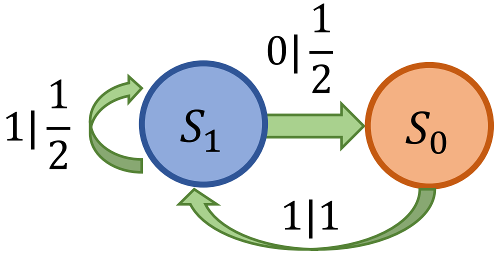
Example

Period-N Uniform Renewal Process



$\sim U(0, N-1)$

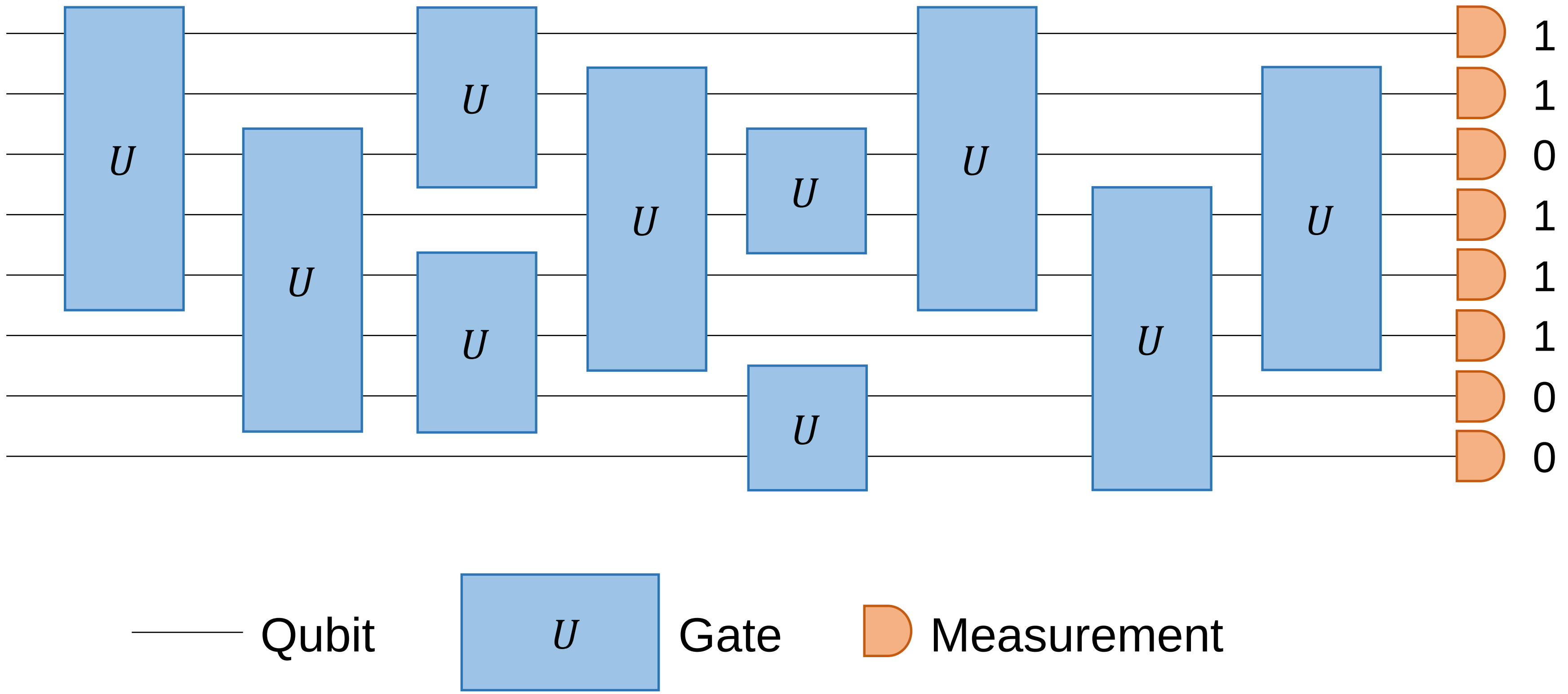
Period = 2



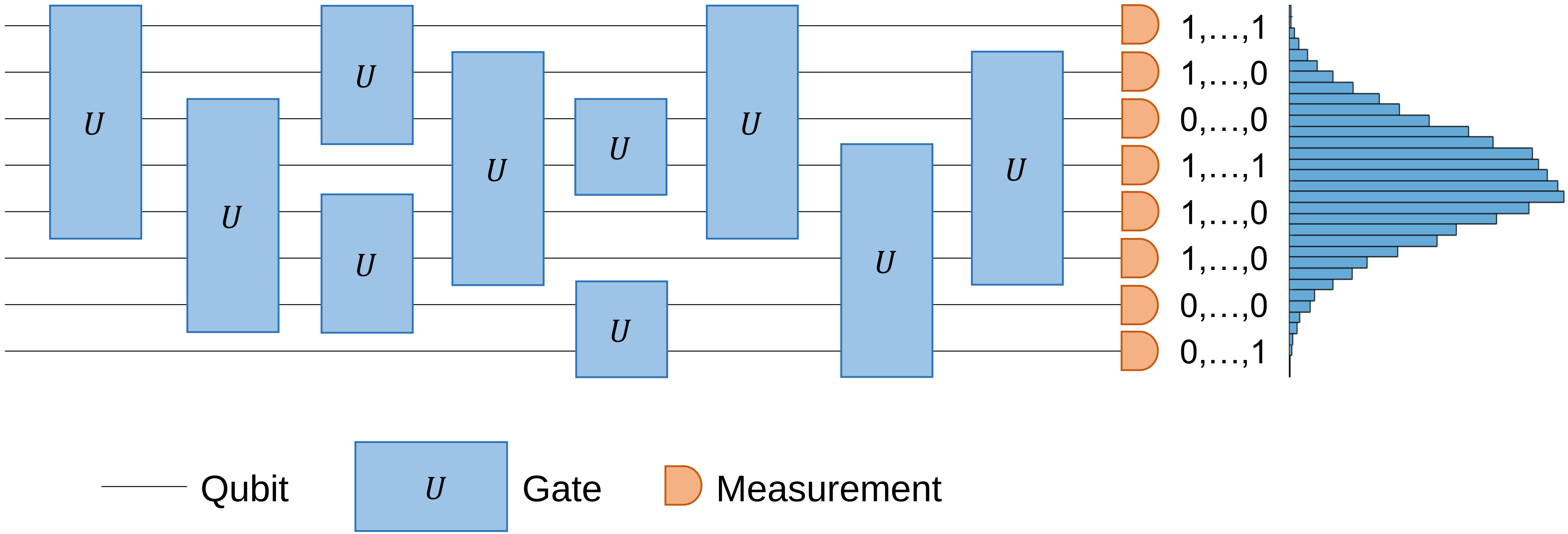
ϵ-machine

Quantum Circuits

Quantum Circuits

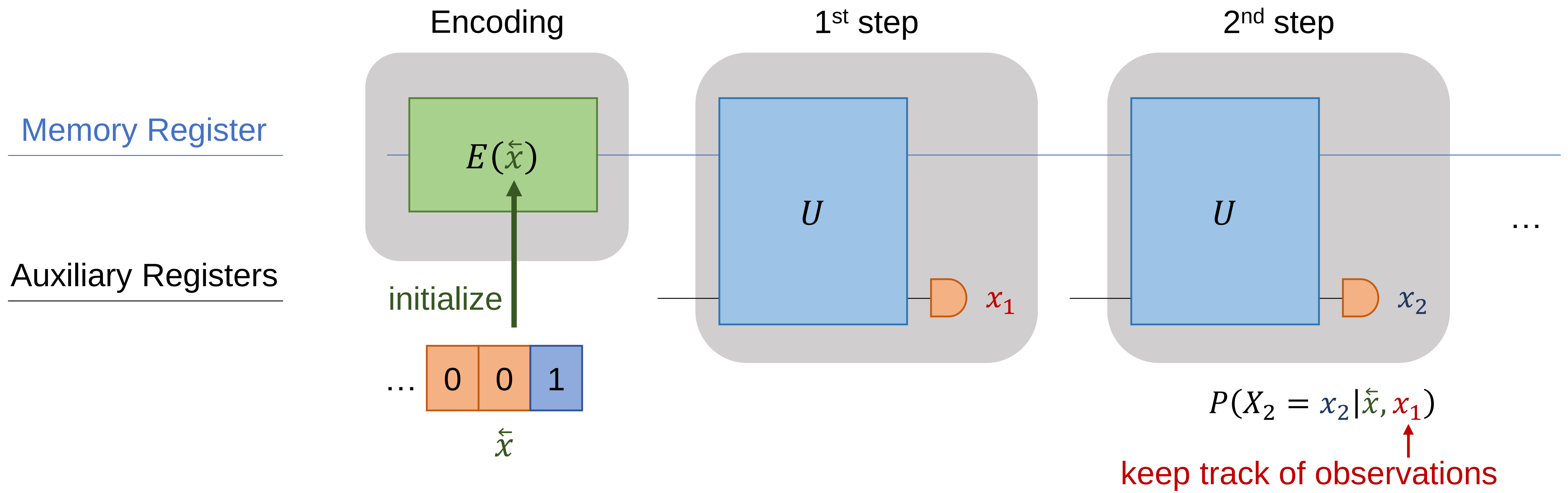


Quantum Circuits



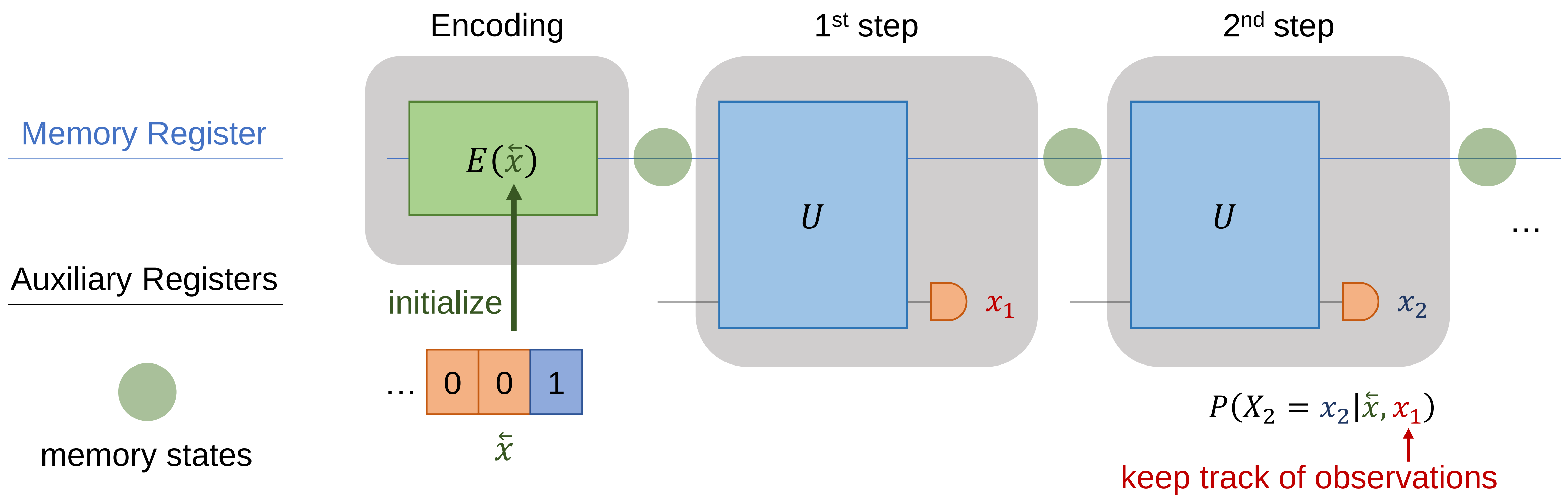
q–simulator

q–simulator



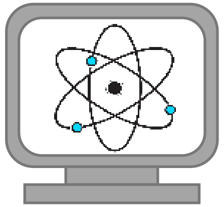
Encoding
1st step
2nd step
Memory Register
Auxiliary Registers
$E(\overleftarrow{x})$
initialize
U
U
x_1
x_2
…
… 0 0 1
$\overleftarrow{x}$
$P(X_2 = x_2 | \overleftarrow{x}, x_1)$
keep track of observations

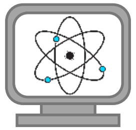
q–simulator



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

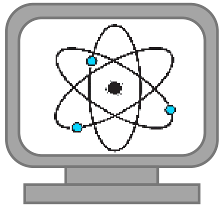
Advantage

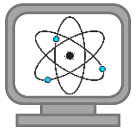
In general:  $d_q \leq d_c$ 

For some processes:  $d_q < d_c$ 

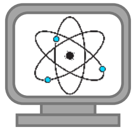
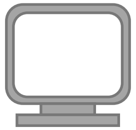
Thompson et al., 10.1103/PhysRevX.8.031013

Advantage

In general:  $d_q \leq d_c$ 

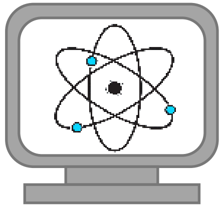
For some processes:  $d_q < d_c$ 

Thompson et al., 10.1103/PhysRevX.8.031013

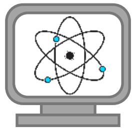
Approximate models:  $\hat{d}_q = \hat{d}_c$ 

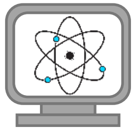
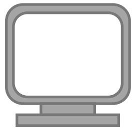
Q-Models can have better accuracy
Yang et al., arXiv:2105.14434

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

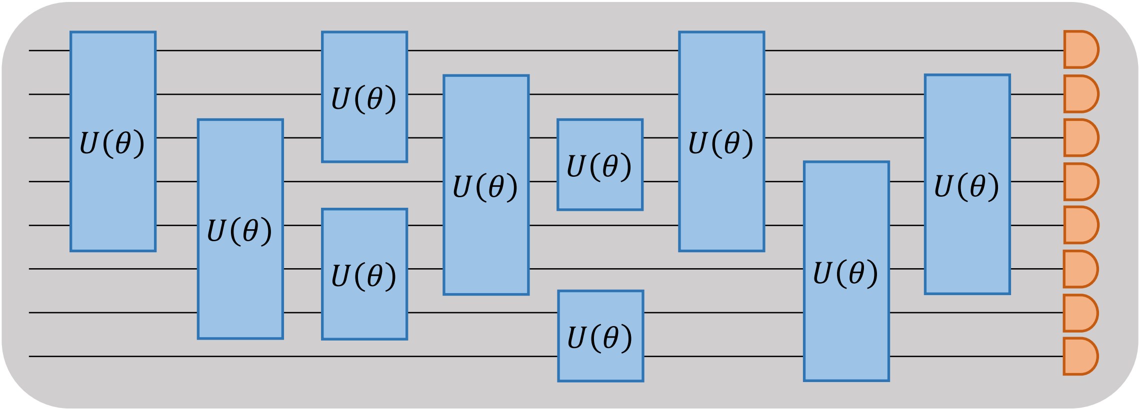
Q-Models can have better accuracy
Yang et al., arXiv:2105.14434

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

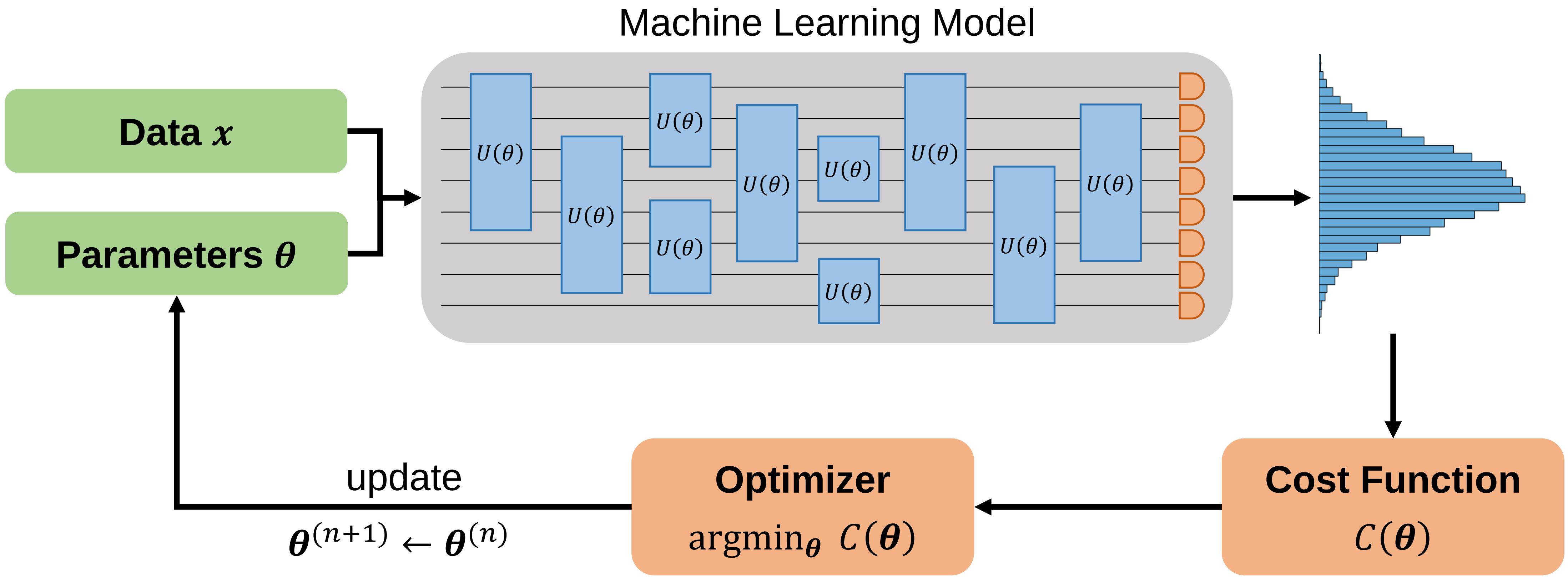
Variational Quantum Circuits

Variational Quantum Circuits

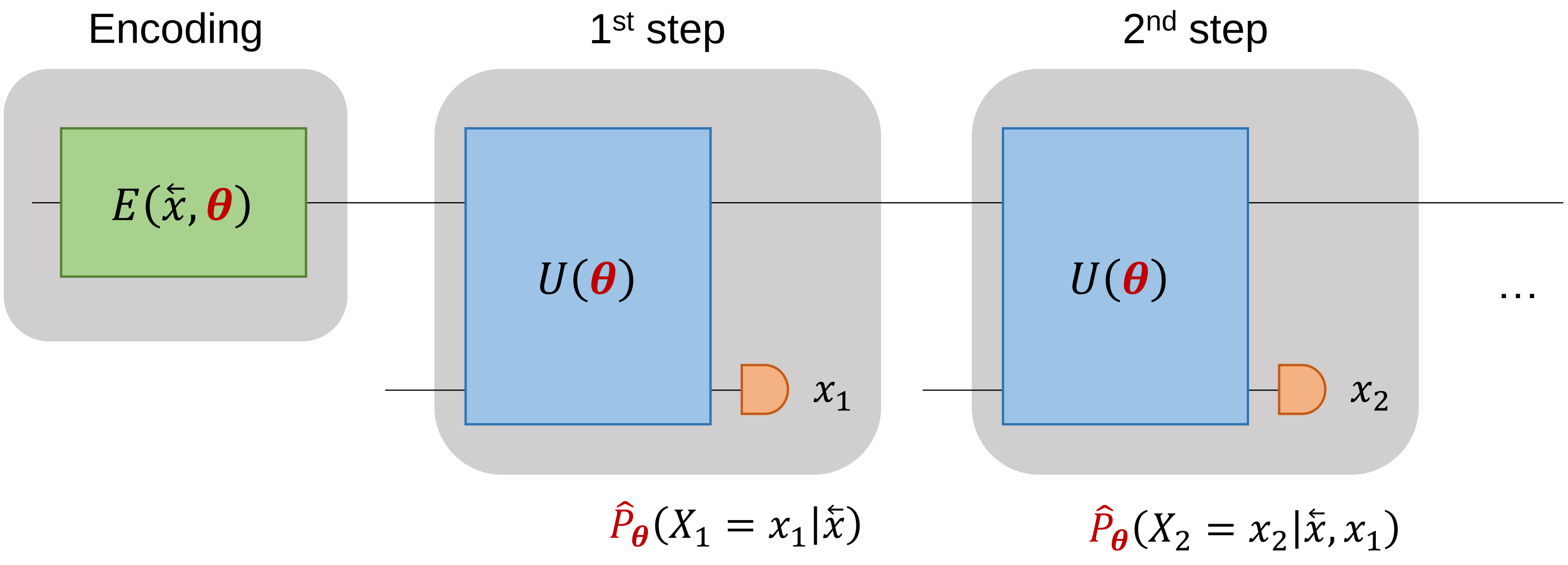
Machine Learning Model



Variational Quantum Algorithms



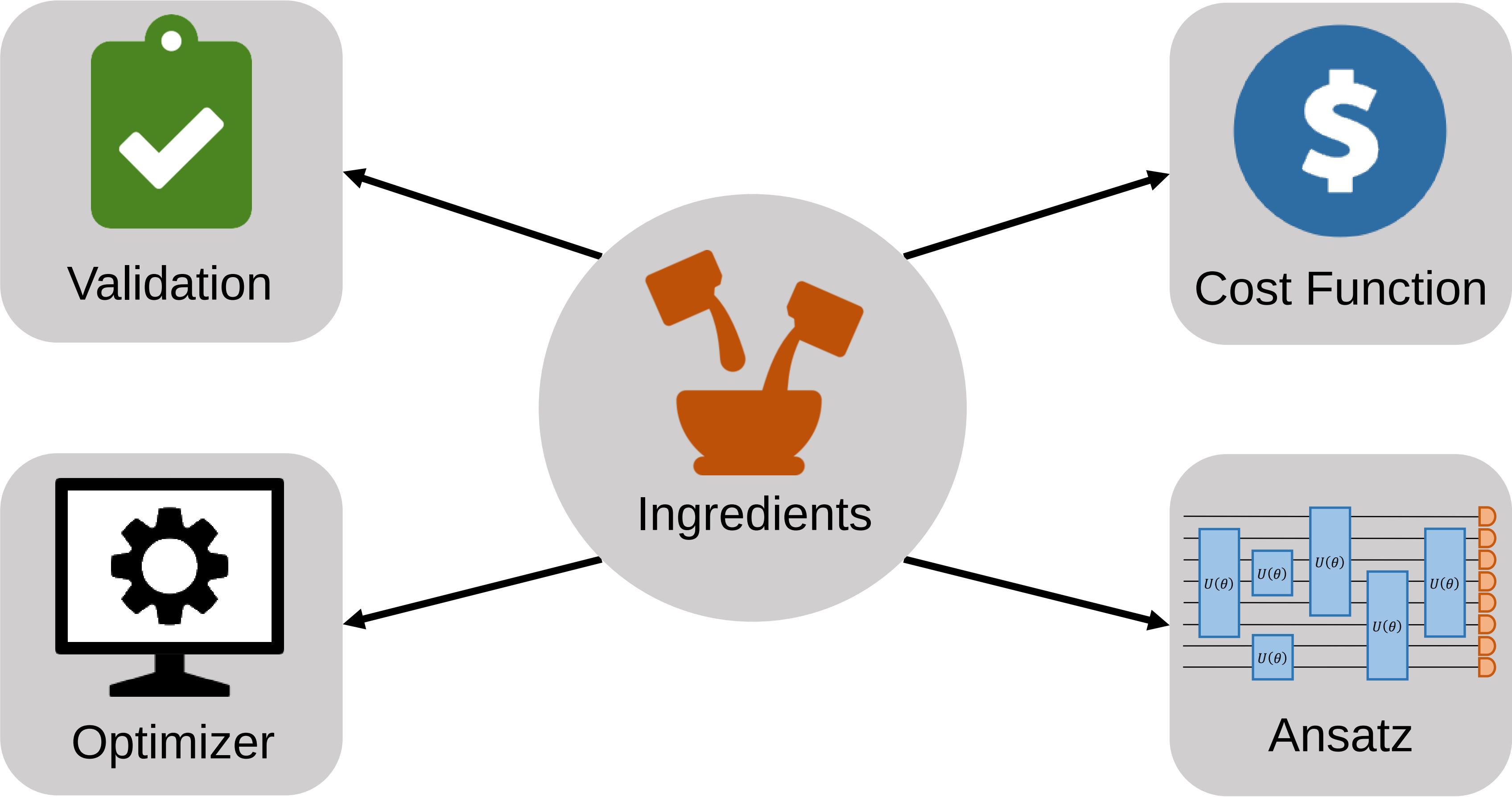
Idea



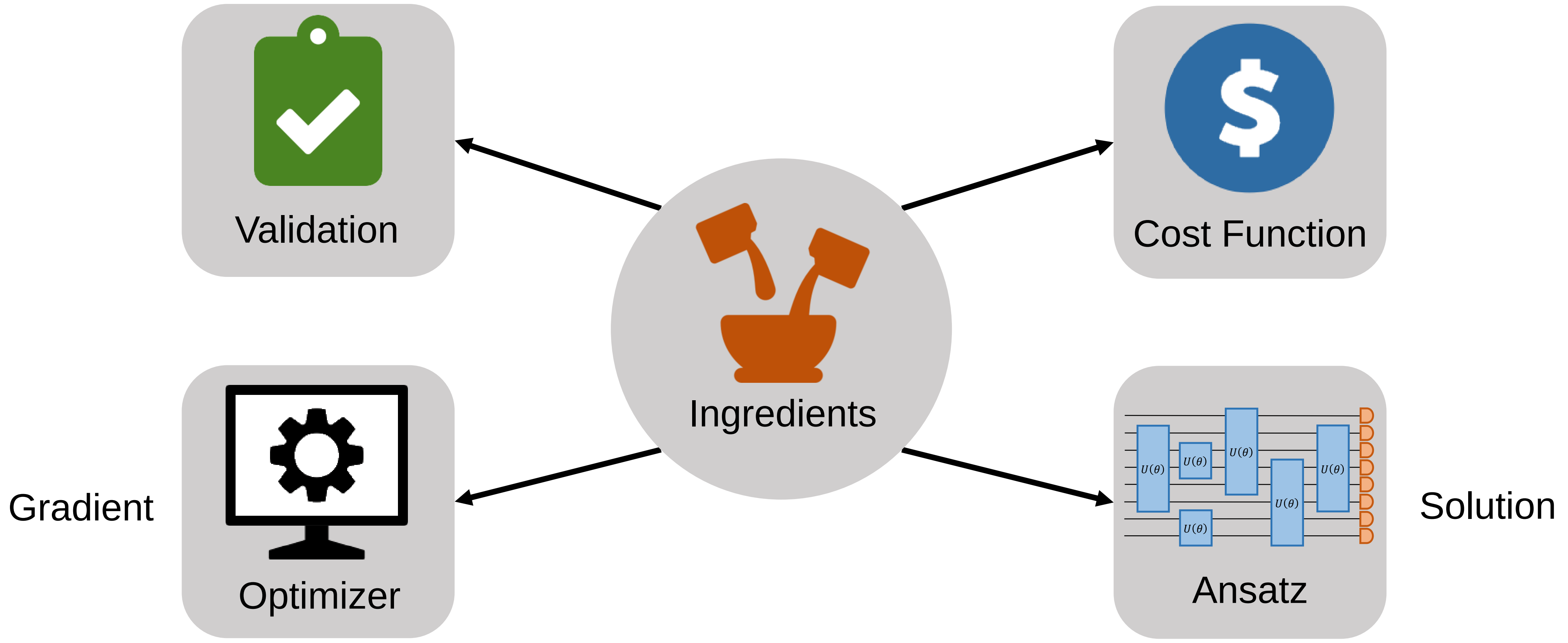
Approximate P → $\left| P - \hat{P}_{\boldsymbol{\theta}} \right| < \delta$

Quantum Learning Algorithm

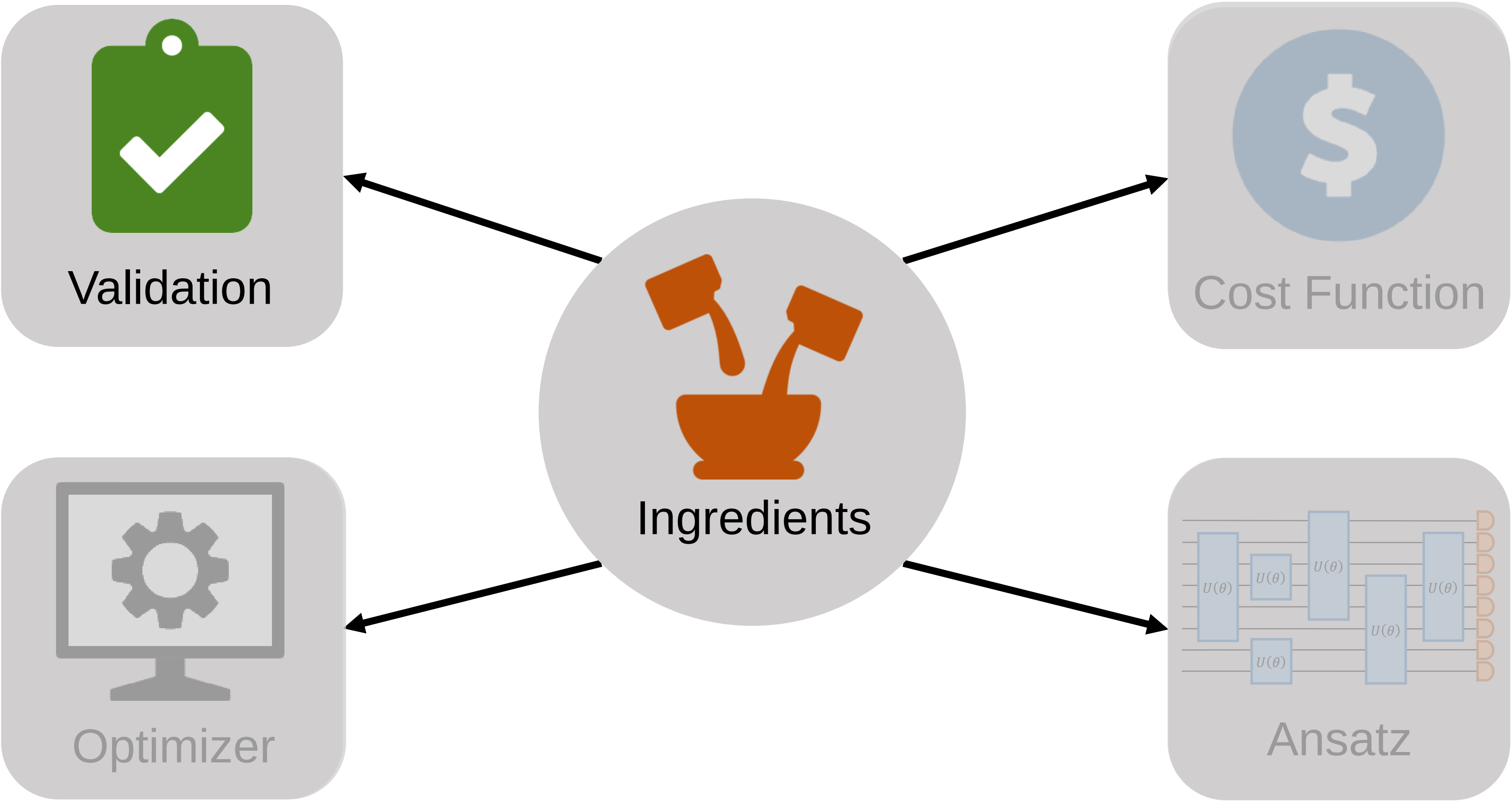
Quantum Learning Algorithm



Quantum Learning Algorithm



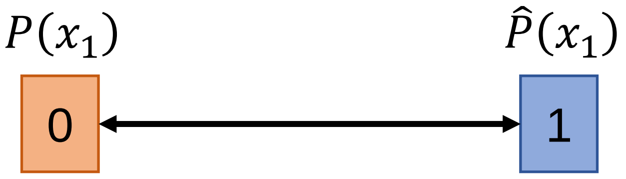
Quantum Learning Algorithm



Validation

Kullback-Leibler divergence:
(KL)

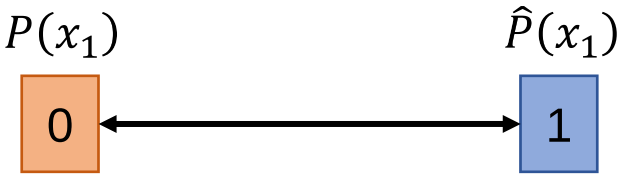
$$D_{KL}(P, \hat{P}) = \sum_x P(x) \log_2 \frac{P(x)}{\hat{P}(x)}$$



Validation

Kullback-Leibler divergence:
(KL)

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



mean over time steps

average over pasts

Validation

Kullback-Leibler divergence:
(KL)

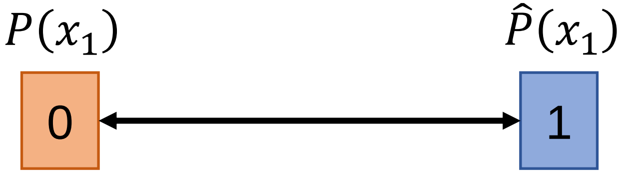
$$D_{KL}(P,\hat{P}) = \sum_x P(x) \log_2 \frac{P(x)}{\hat{P}(x)}$$

mean over time steps

average over pasts

Total Variation Distance:
(TV)

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



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

Total Variation Distance: (TV)

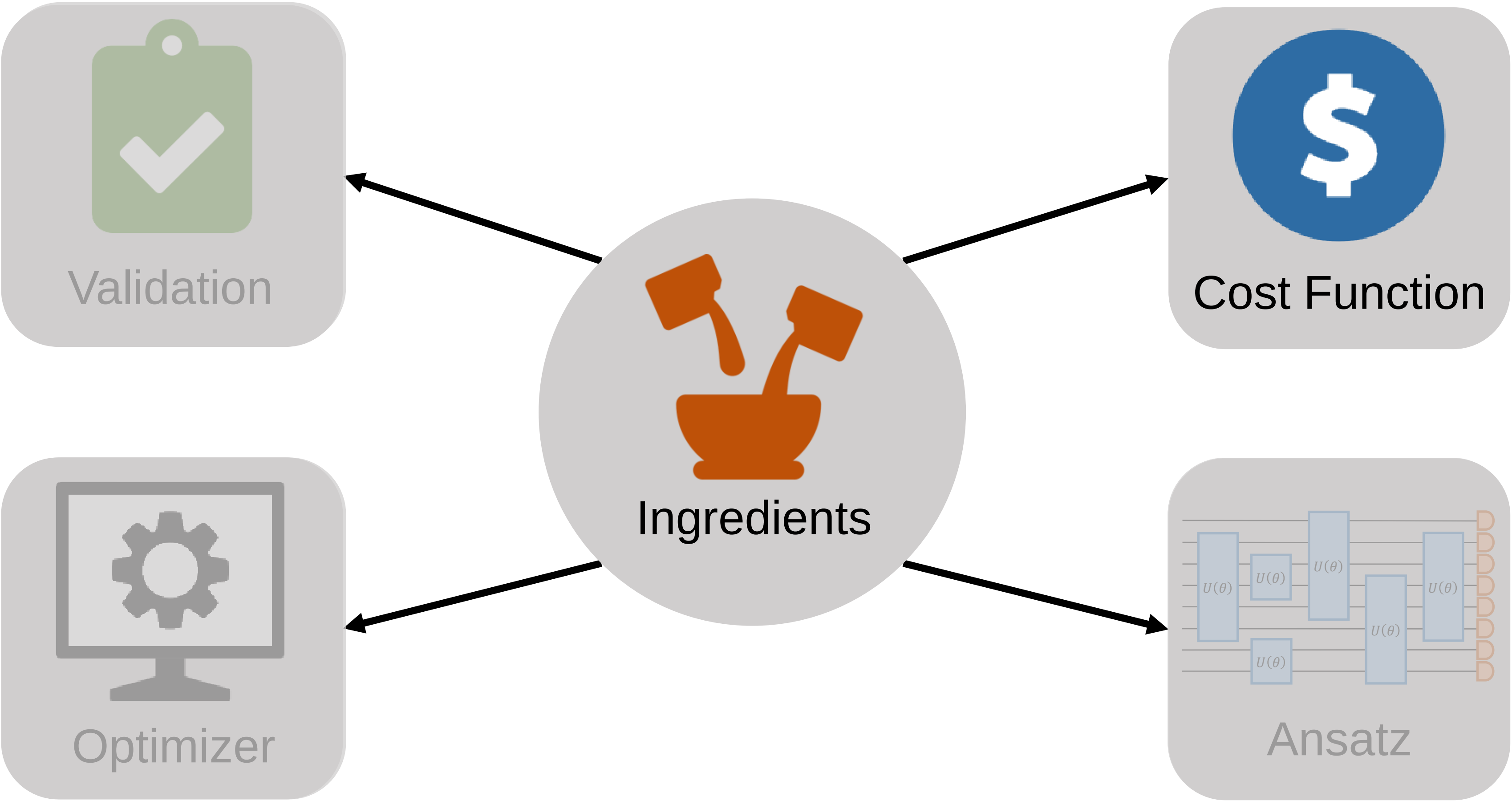
$$D_{TV}(P,\hat{P}) = \frac{1}{2}\sum_x \left|P(x)-\hat{P}(x)\right|$$

$P(x_1)$ 0 ⟷ 1 $\hat{P}(x_1)$

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

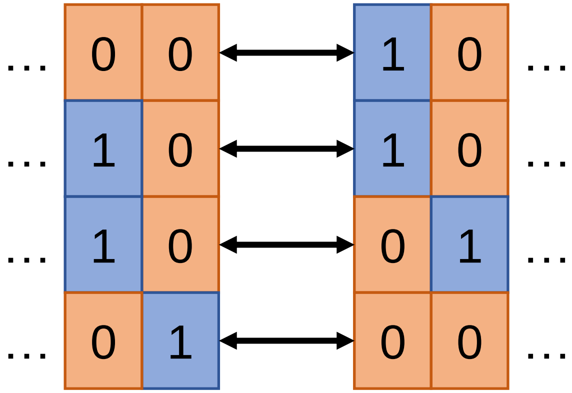
Ideally, use validation metric: $$D_{KL}(P, \hat{P}) = \sum_x P(x) \log_2 \frac{P(x)}{\hat{P}(x)}$$

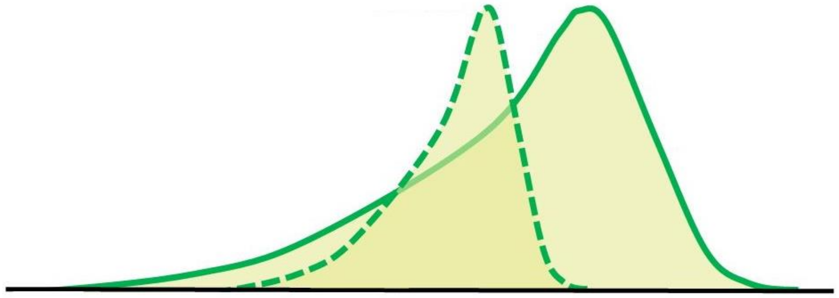
unknown

inefficient

Cost Function

Maximum Mean Discrepancy:
(MMD)

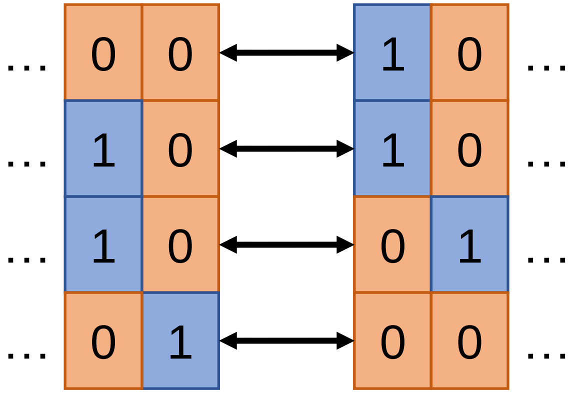


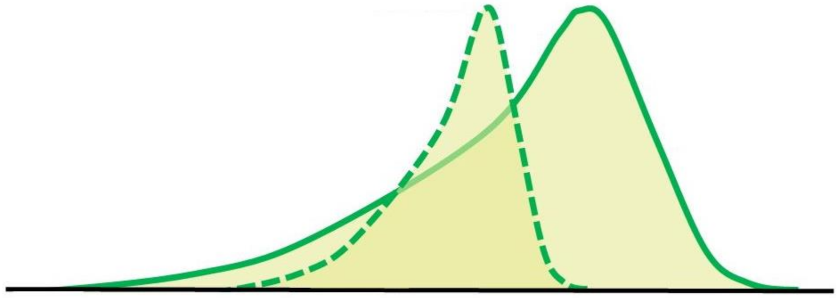


$$\mathrm{MMD}\left[P, \hat{P}\right] = 0 \iff P = \hat{P}$$

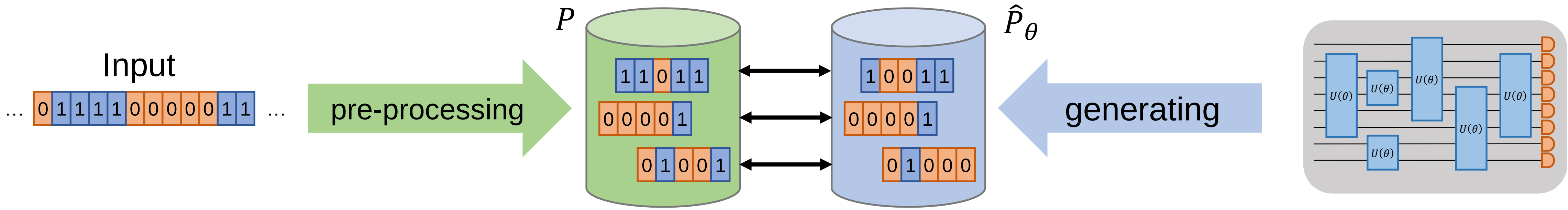
Cost Function

Maximum Mean Discrepancy:
(MMD)



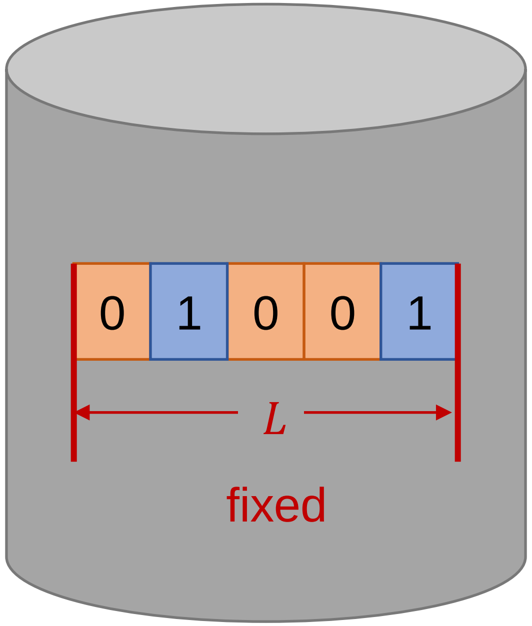


$$\mathrm{MMD}\left[P, \hat{P}\right] = 0 \iff P = \hat{P}$$



Cost Function

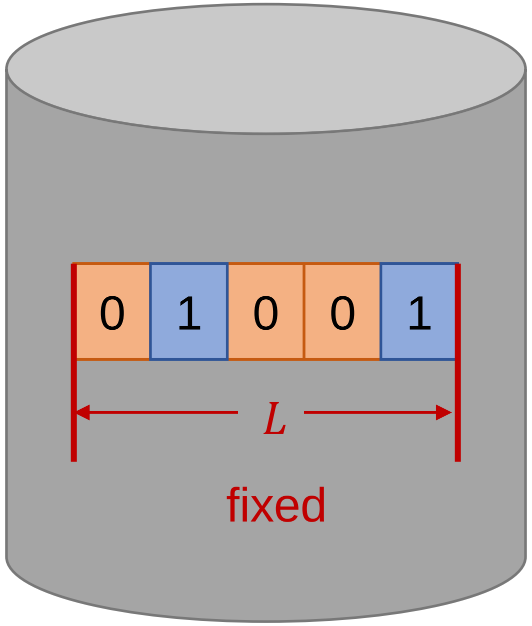
$$C(\boldsymbol{\theta}) = \sum_{\overleftarrow{x}} w_{\overleftarrow{x}} \cdot \mathrm{MMD}^2[\mathrm{P}, \hat{P}_{\boldsymbol{\theta}} | \overleftarrow{x}]$$



Cost Function

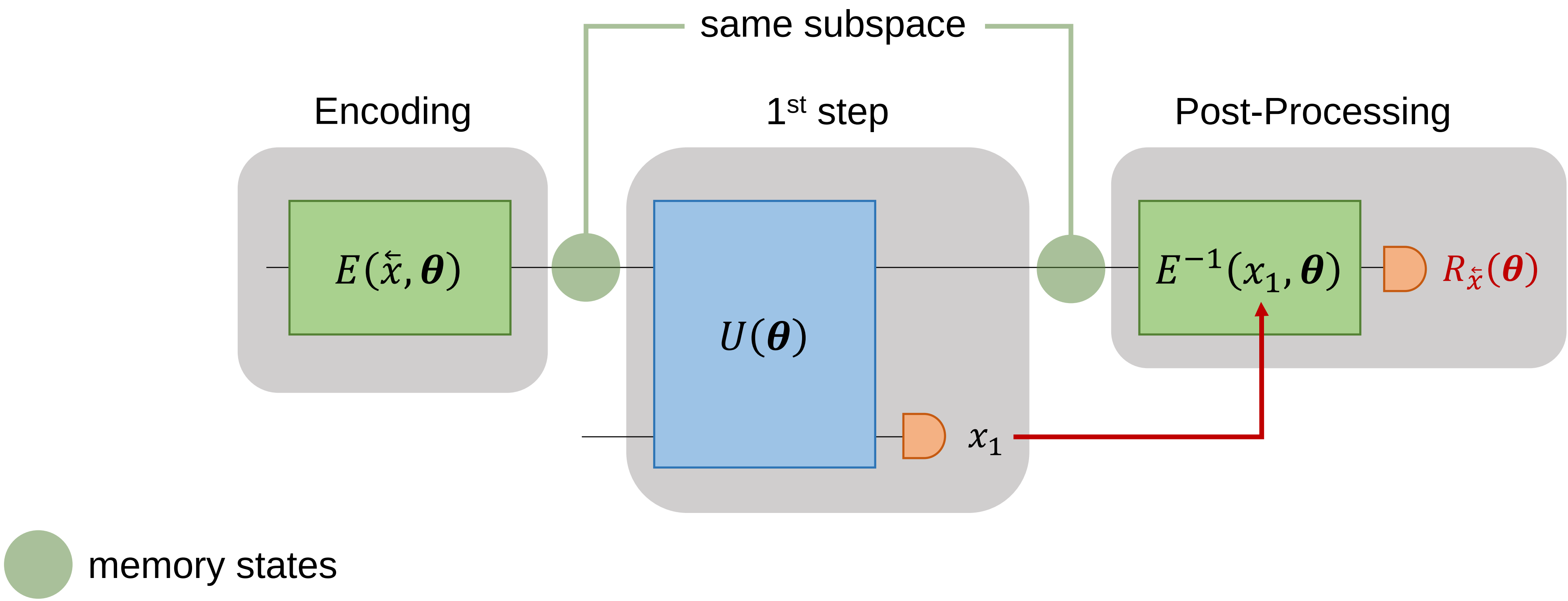
$$C(\boldsymbol{\theta}) = \sum_{\overleftarrow{x}} w_{\overleftarrow{x}} \cdot \mathrm{MMD}^2[\mathrm{P}, \hat{P}_{\boldsymbol{\theta}} | \overleftarrow{x}]$$

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



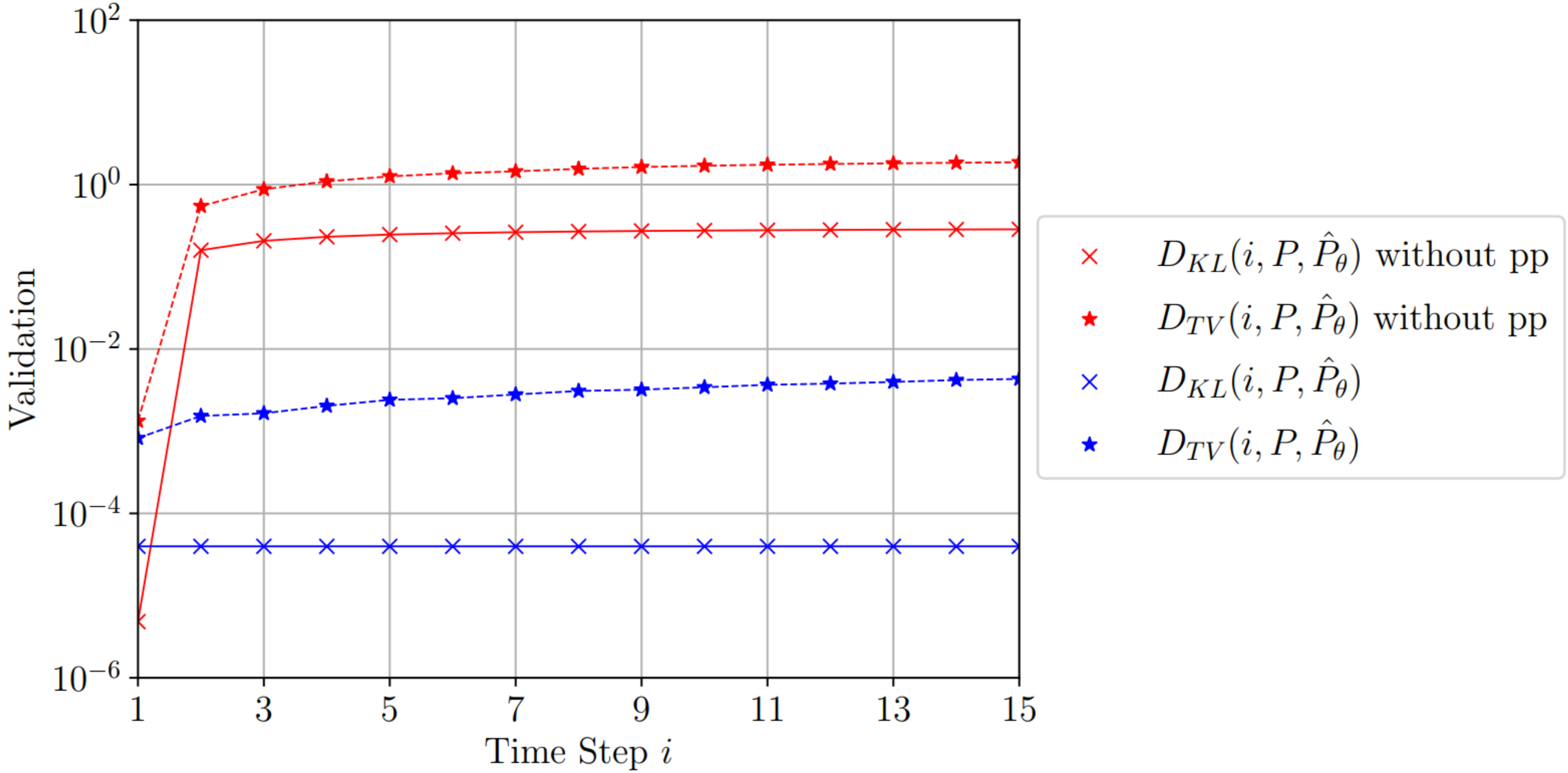
Regularization = penalizes models with a large set of memory states

Cost Function



Results

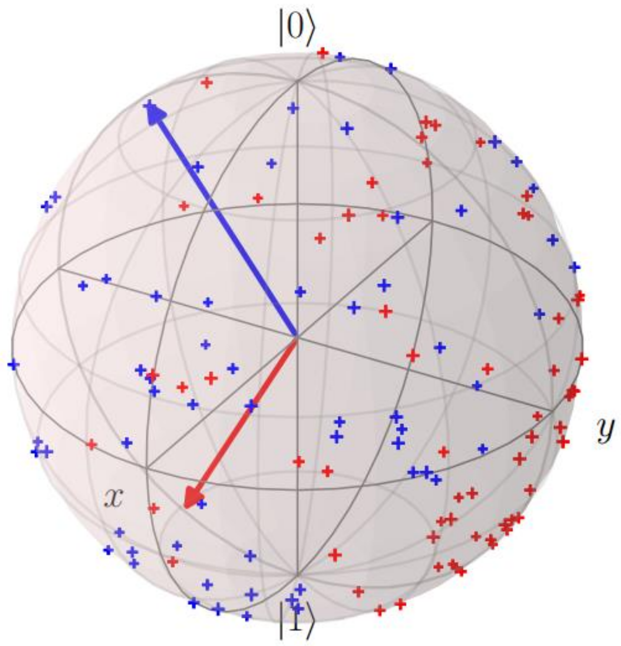
Results



10^2
10^0
10^{-2}
10^{-4}
10^{-6}
Validation
1
3
5
7
9
11
13
15
Time Step i
$D_{KL}(i, P, \hat{P}_\theta)$ 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 $\overleftarrow{x} = 1$

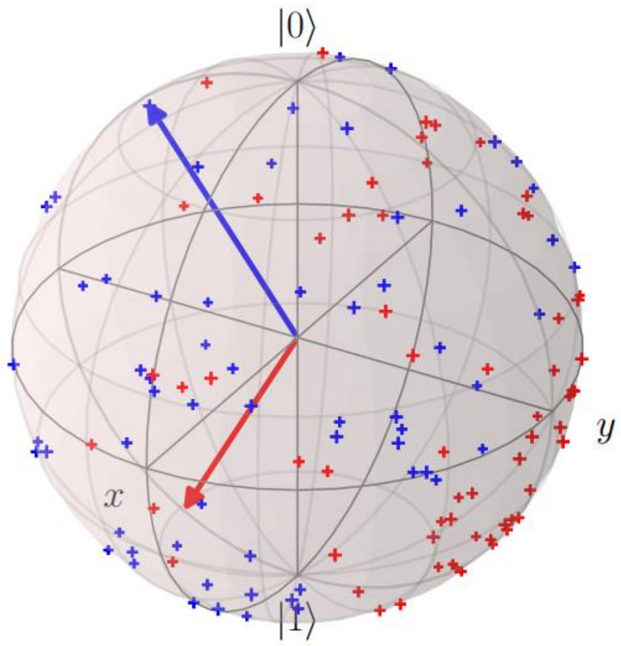
memory states for $x_i = 0$

memory states for $x_i = 1$

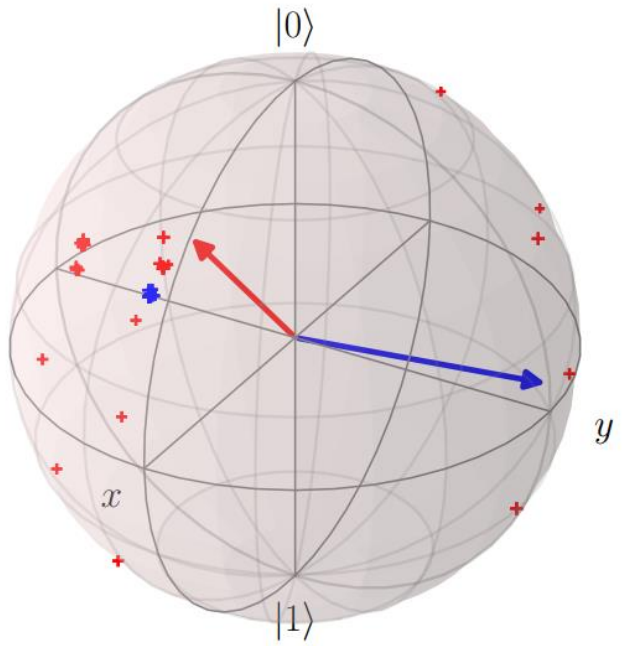
$U(\boldsymbol{\theta})$ | $U(\boldsymbol{\theta})$ | …

Results

random



without pp



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

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

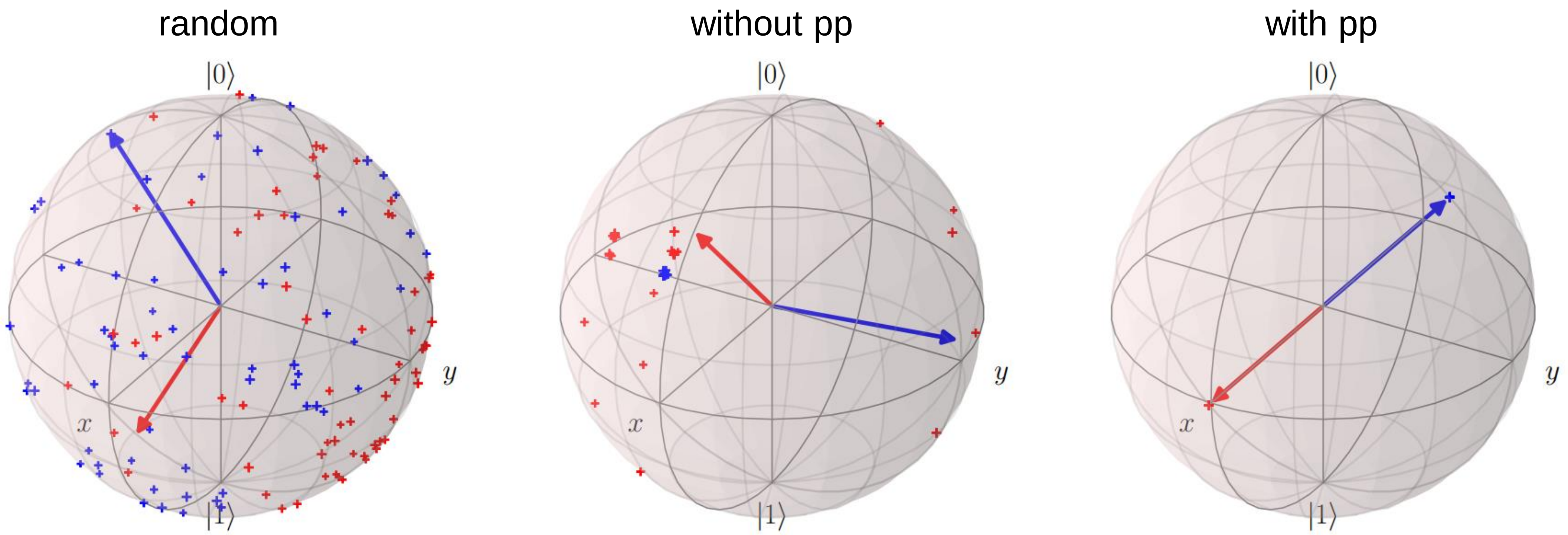
initial state for $\overleftarrow{x} = 1$

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

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

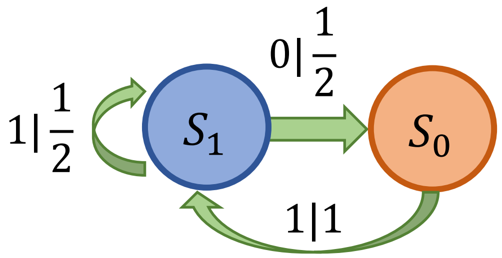
- Apply the algorithm to the period-3 uniform renewal process
- Consider processes with outcomes $\subseteq \mathbb{N}$

Thank you very much!

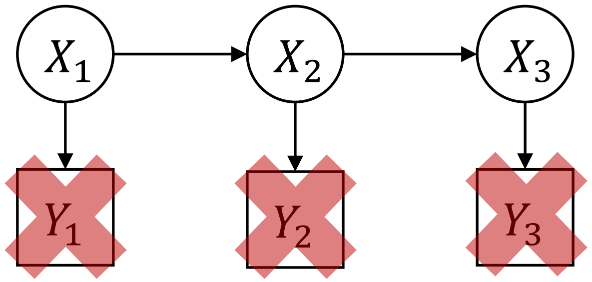
Questions are highly welcome.

ϵ–machine vs. HMM

ϵ-machine



HMM



unifilar

q–simulator

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

$$E(\overleftarrow{x})\,|0\rangle = |\sigma_{\overleftarrow{x}}\rangle$$

memory state

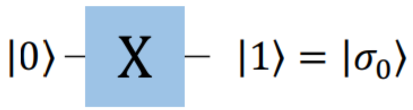
$U(\boldsymbol{\theta})$

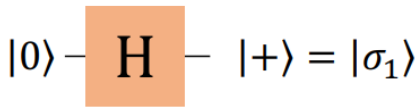
$$T\,|\sigma_{\overleftarrow{x}}\rangle\,|0\rangle = \sum_{x \in A} \sqrt{P(x|\overleftarrow{x})}\,|\sigma_{\overleftarrow{x},x}\rangle\,|x\rangle \xrightarrow{I \otimes M} |\sigma_{\overleftarrow{x},x^*}\rangle\,|x^*\rangle$$

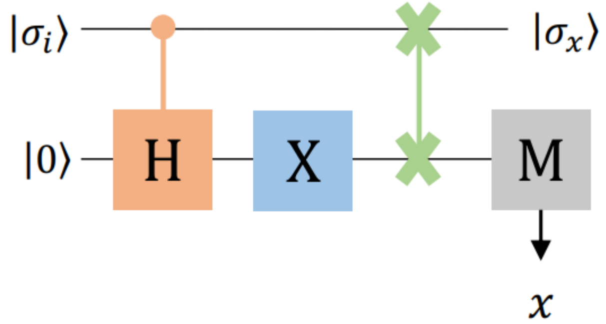
measure

Ansatz

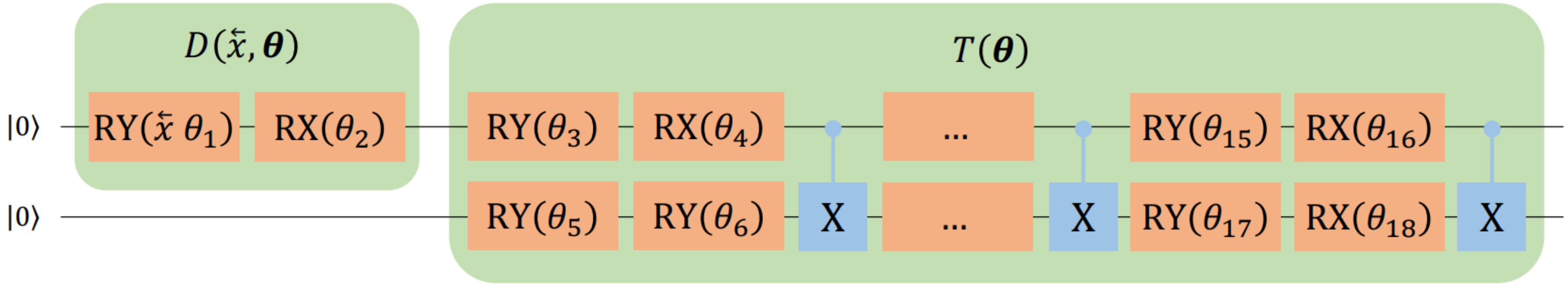
Analytical solution:



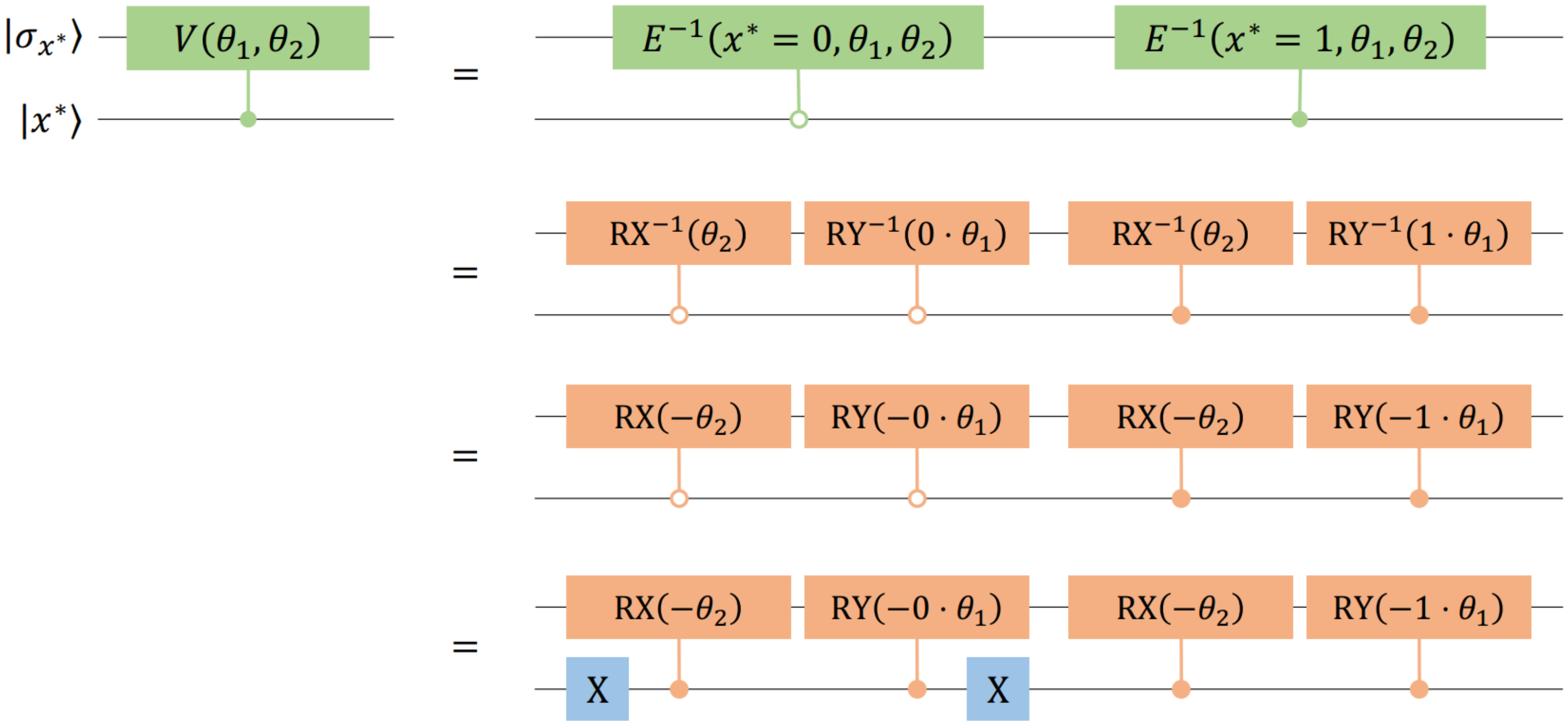




Ansatz:



Post Processing



Validation

Kullback-Leibler divergence: (KL)

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

$$D_{KL}(L,P,\hat{P}) = \sum_{x_{1:L}}\frac{1}{L}\sum_{\overleftarrow{x}} P(\overleftarrow{x})\cdot P(x_{1:L}|\overleftarrow{x})\log_2\frac{P(x_{1:L}|\overleftarrow{x})}{\hat{P}(x_{1:L}|\overleftarrow{x})}$$

Total Variation Distance: (TV)

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

$$D_{TV}(L,P,\hat{P}) = \frac{1}{2}\sum_{x_{1:L}}\sum_{\overleftarrow{x}}\left|P(x_{1:L}|\overleftarrow{x})-\hat{P}(x_{1:L}|\overleftarrow{x})\right|$$

Maximum Mean Discrepancy

Kernel

$$MMD^2[F, P, \hat{P}] = \mathbb{E}_{x\sim P, x'\sim P}[k(x, x')] - 2\cdot\mathbb{E}_{x\sim P, y'\sim\hat{P}}[k(x, y)] + \mathbb{E}_{y\sim\hat{P}, y'\sim\hat{P}}[k(y, y')]$$

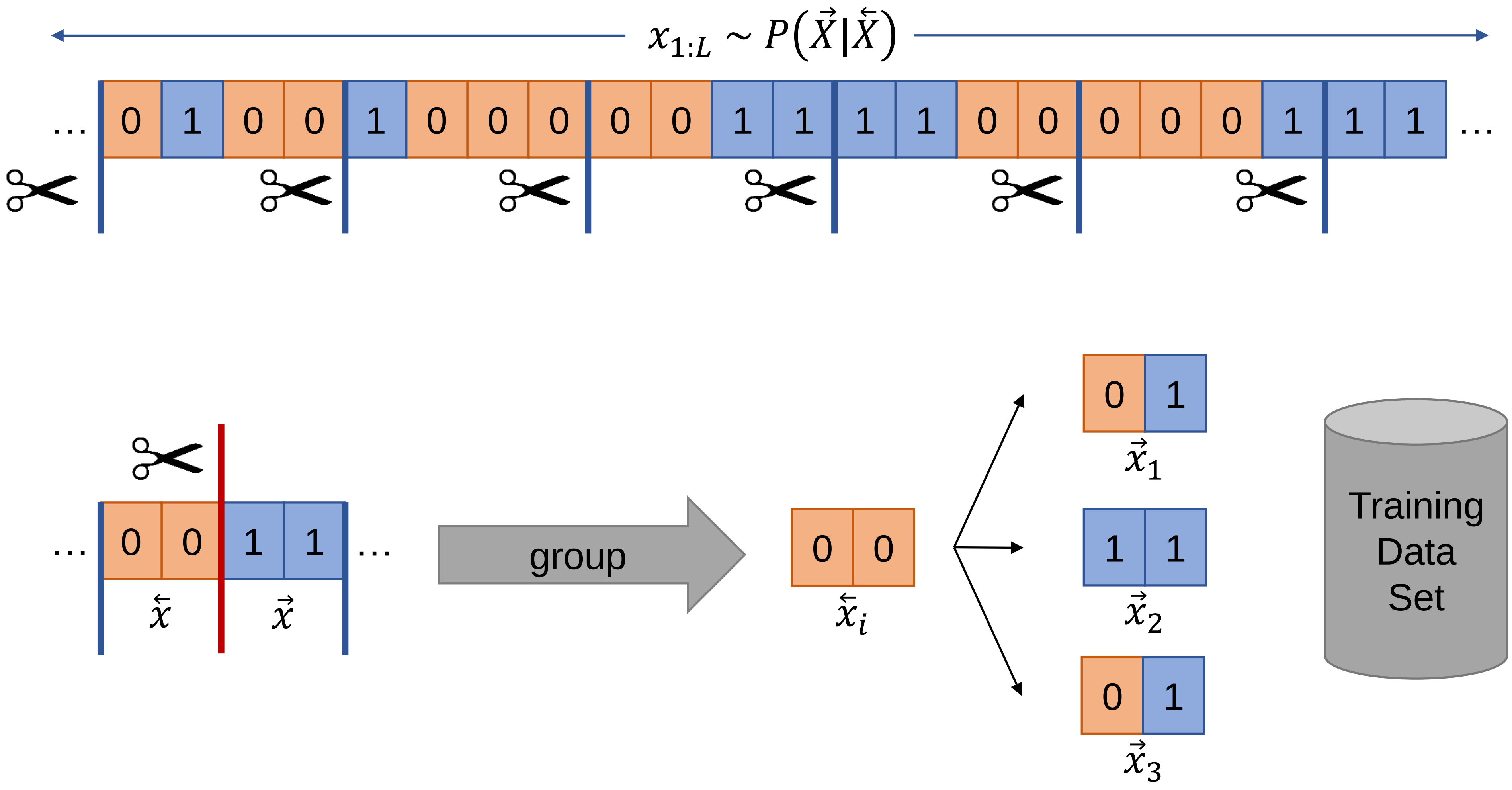
$$\frac{\partial MMD^2[P, \hat{P}_{\boldsymbol{\theta}}]}{\partial\theta_i} = \sum_{j=1}^{m} \alpha'_{i,j}(\theta_i)\left[\underset{\substack{x\sim\hat{P}_{\boldsymbol{\theta}^+_{i,j}} \\ y\sim\hat{P}_{\boldsymbol{\theta}}}}{\mathbb{E}}[k(x, y)] - \underset{\substack{x\sim\hat{P}_{\boldsymbol{\theta}^-_{i,j}} \\ y\sim\hat{P}_{\boldsymbol{\theta}}}}{\mathbb{E}}[k(x, y)] - \underset{\substack{x\sim\hat{P}_{\boldsymbol{\theta}^+_{i,j}} \\ y\sim P}}{\mathbb{E}}[k(x, y)] + \underset{\substack{x\sim\hat{P}_{\boldsymbol{\theta}^-_{i,j}} \\ y\sim P}}{\mathbb{E}}[k(x, y)]\right]$$

Matrix depending on the ansatz

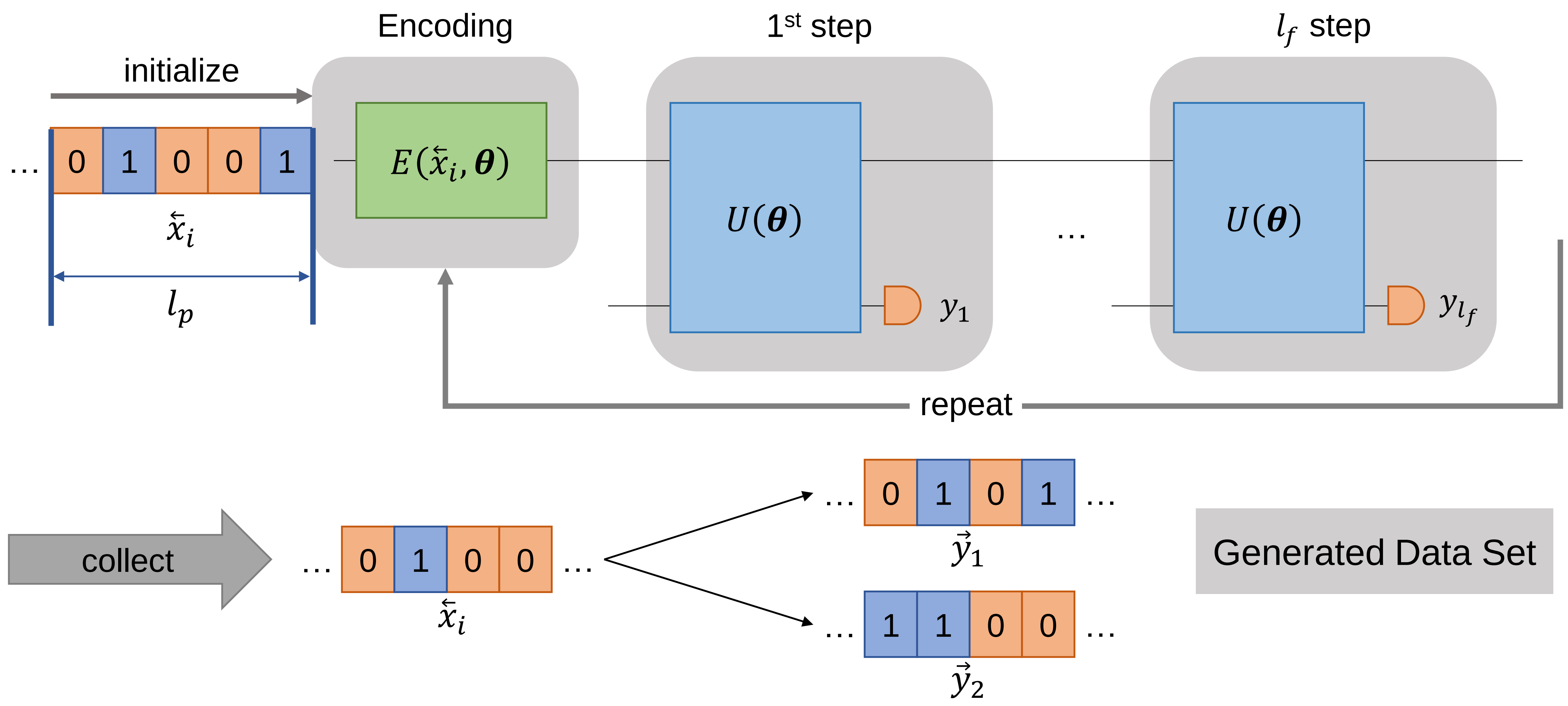
Cost Function

Input:

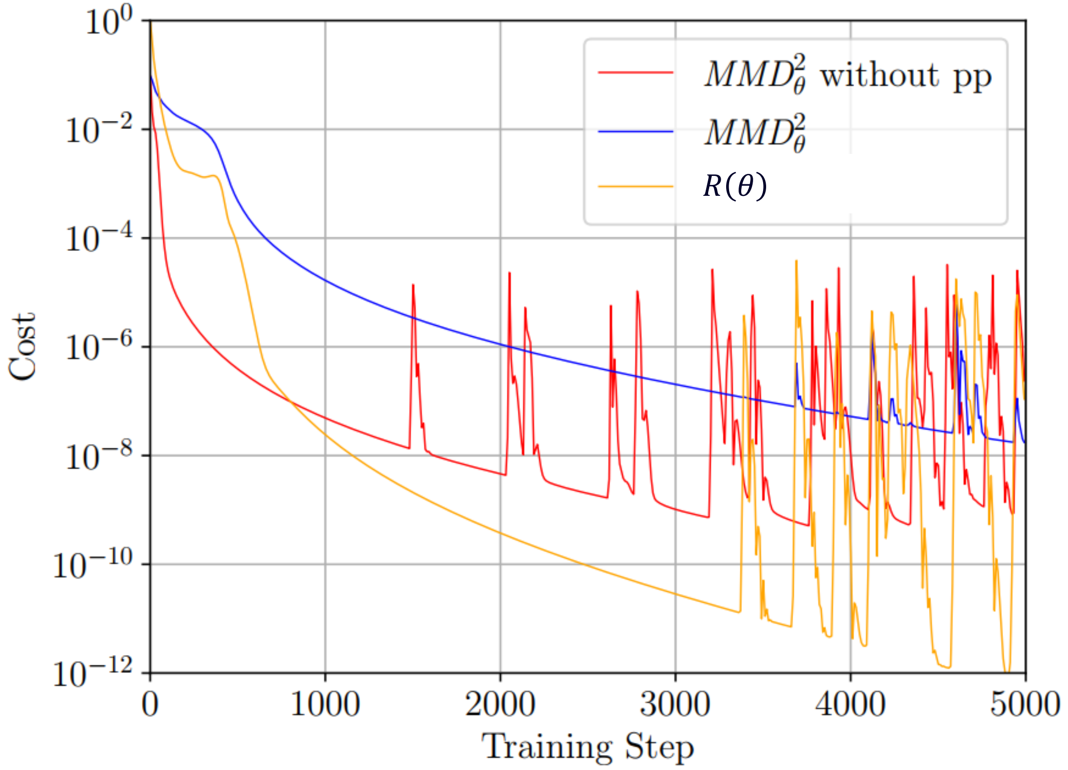
Pre-Processing:



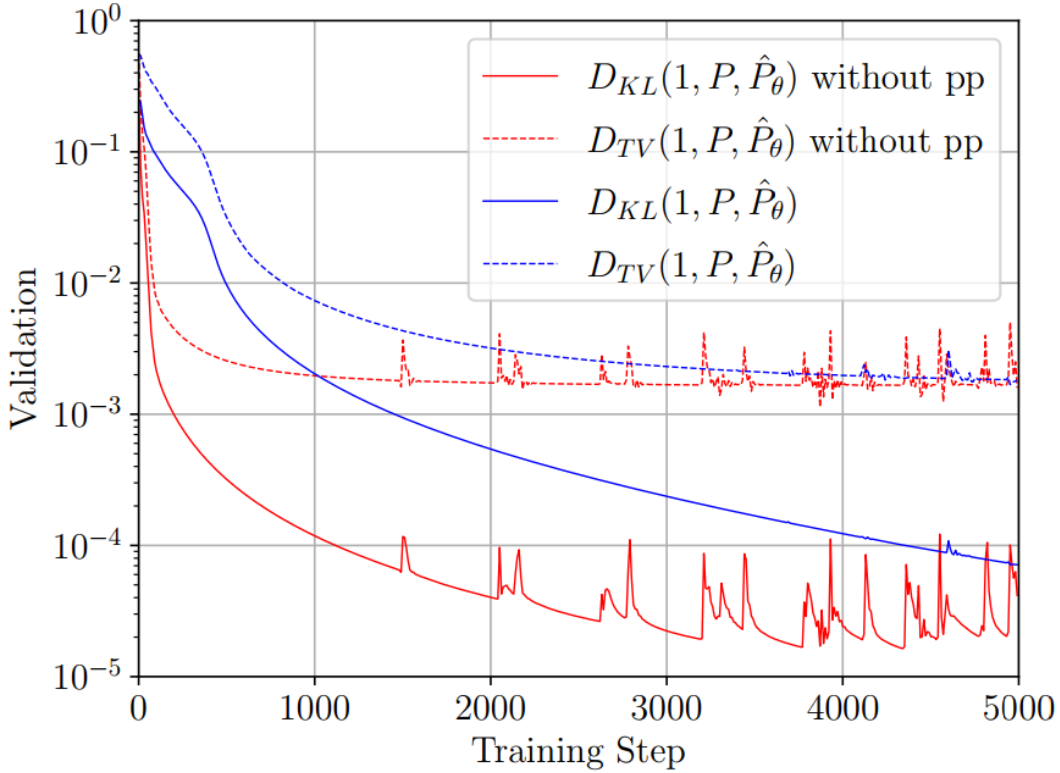
Cost Function



Training

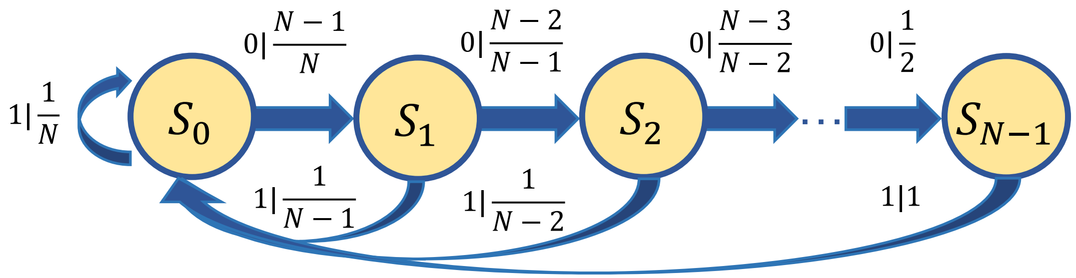


10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
Cost
0
1000
2000
3000
4000
5000
Training Step
MMD_θ^2 without pp
MMD_θ^2
$R(\theta)$



10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
Validation
0
1000
2000
3000
4000
5000
Training Step
$D_{KL}(1, P, \hat{P}_\theta)$ without pp
$D_{TV}(1, P, \hat{P}_\theta)$ without pp
$D_{KL}(1, P, \hat{P}_\theta)$
$D_{TV}(1, P, \hat{P}_\theta)$

Period-N Uniform Renewal Process

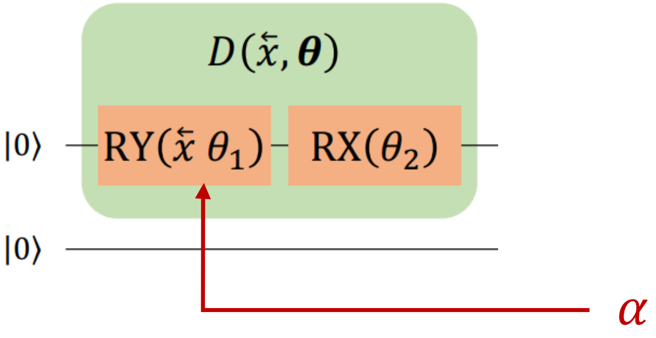


ϵ-machine

Gradients

$$f(\boldsymbol{\theta}) = \langle 0 | U(\boldsymbol{\theta})^{\dagger} \hat{O} U(\boldsymbol{\theta}) | 0 \rangle$$

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \mu} = \frac{1}{2} \sum_{j=1}^{m} \alpha'_j(\mu) \left[f_j\left(\alpha_j(\mu), +\frac{\pi}{2}\right) - f_j\left(\alpha_j(\mu), -\frac{\pi}{2}\right) \right]$$



Thank you very much!

Questions are highly welcome.