

University of Stuttgart
Institute for Computational Physics

Freie Universität



Berlin

SimTech

Simulating Stochastic Processes with Variational Quantum Circuits

Daniel
Fink

QC Seminar

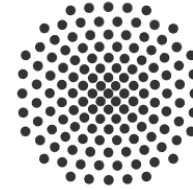
@

School of Physical &
Mathematical Sciences

(Nanyang Technological University Singapore)

August 11th, 2022

- 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**

Can we predict the future based on past observations?



Can we predict the future based on past observations?



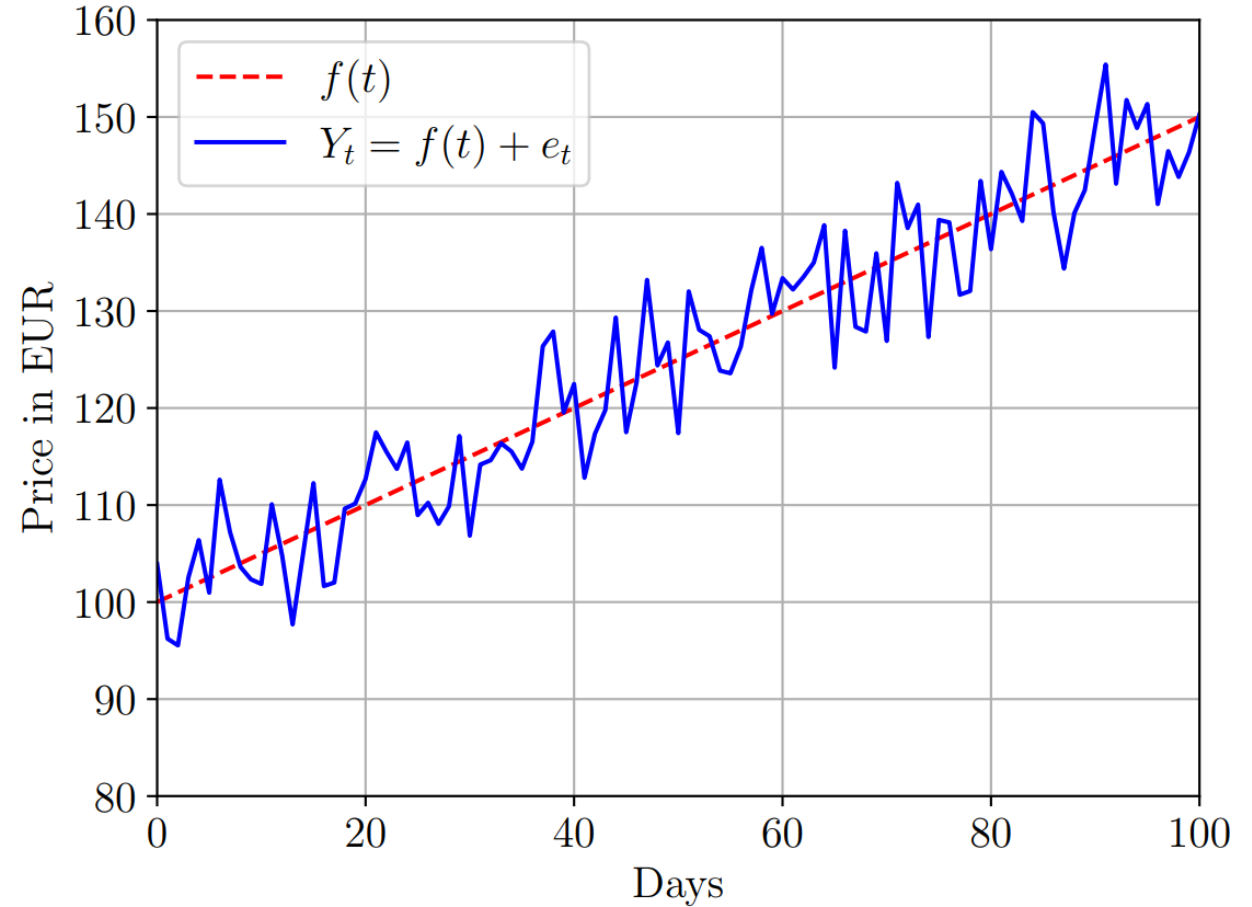
Simulations → show possible futures

Assume linear trend $f(t)$

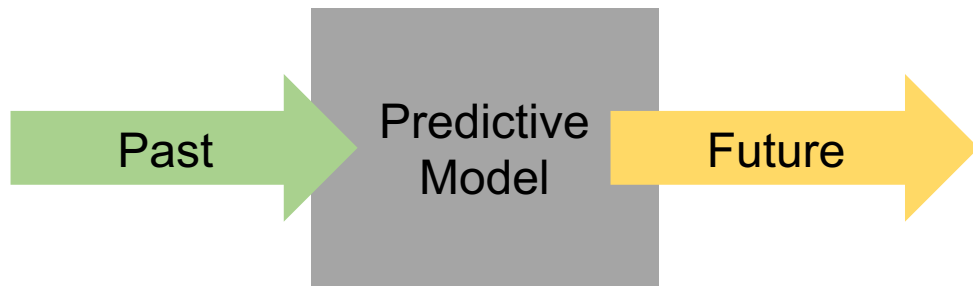
Add some noise e_t

→ e_t is a stochastic process

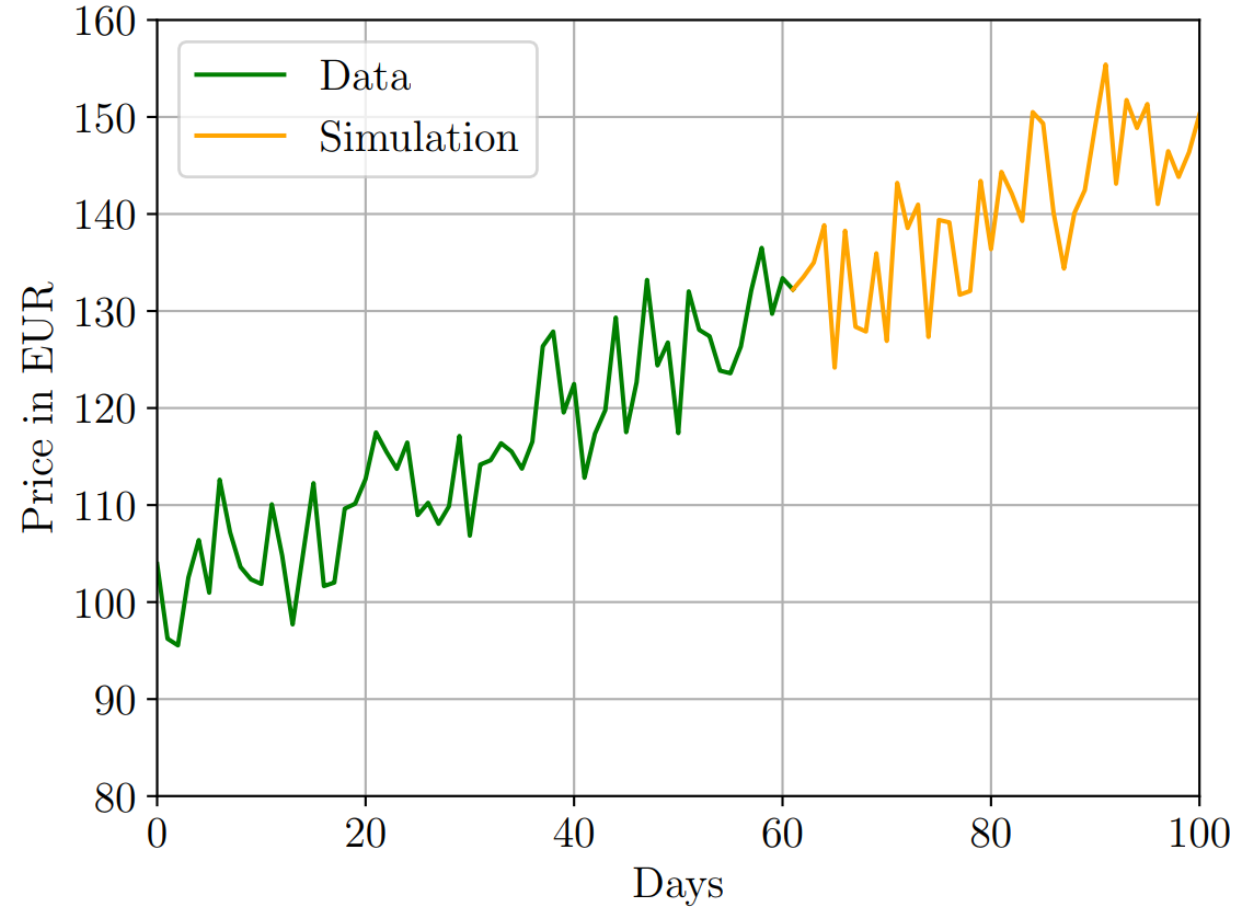
Stock Price Trend



Assume data drawn
by a stochastic process



Stock Price Trend

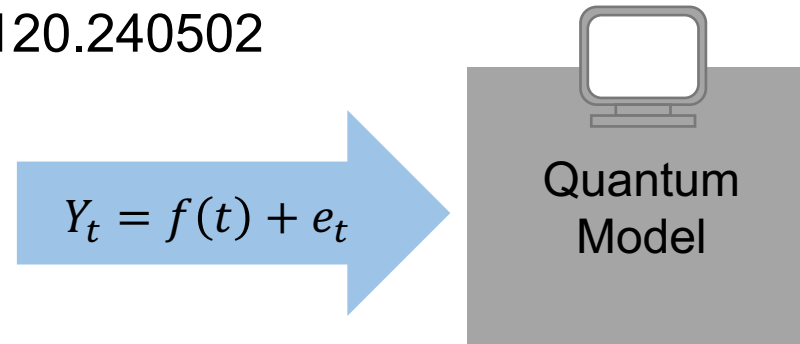




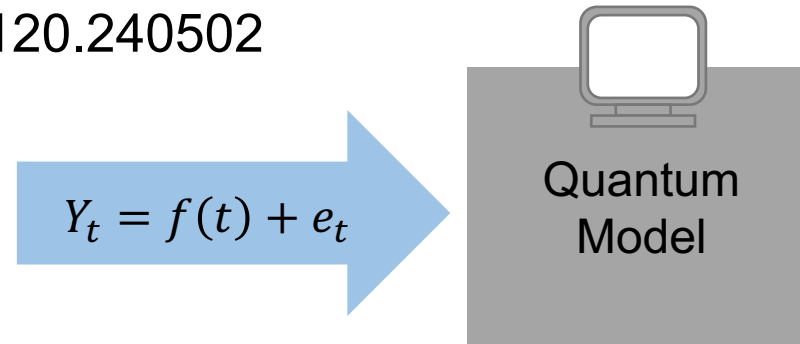
Classical Models \leq Quantum Models

How to get a quantum model?

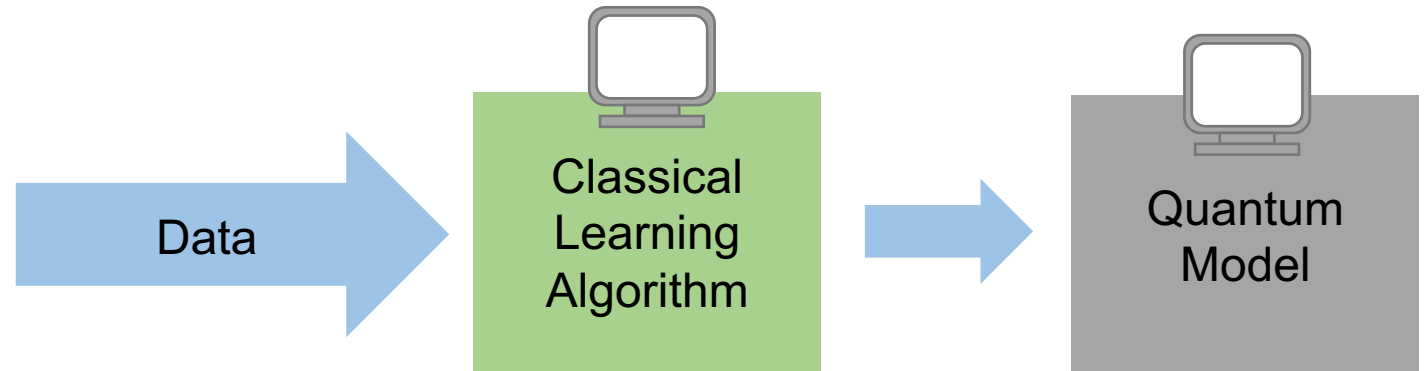
- Classical **description of the process** \rightarrow q -simulator
 - Binder et al., 10.1103/PhysRevLett.120.240502



- Classical **description of the process** \rightarrow q -simulator
 - Binder et al., 10.1103/PhysRevLett.120.240502

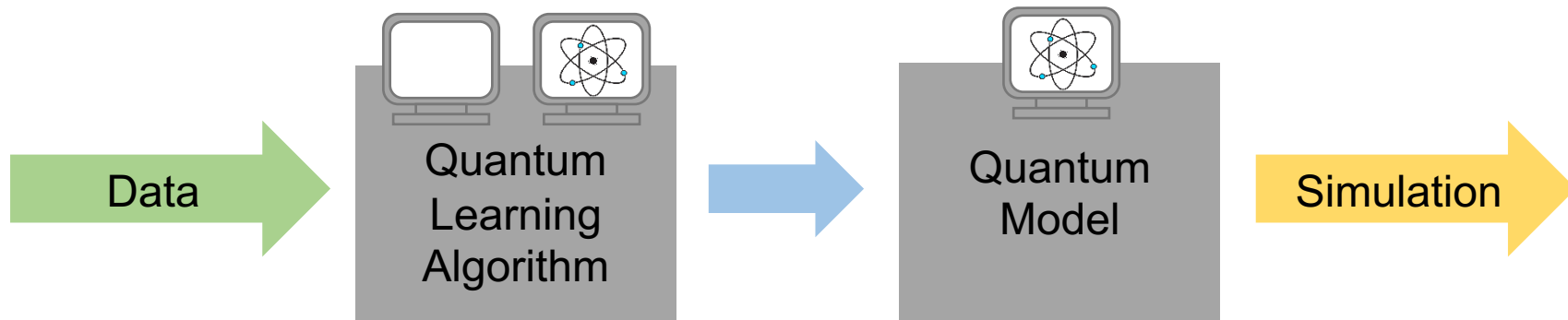


- Data from the process \rightarrow **classical** discovery algorithm
 - Yang et al., arXiv:2105.14434




Goal

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

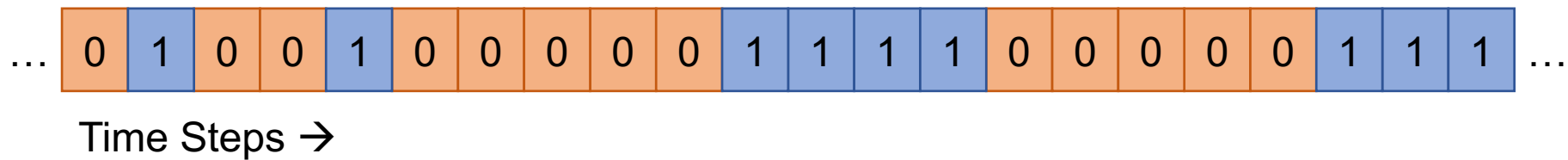


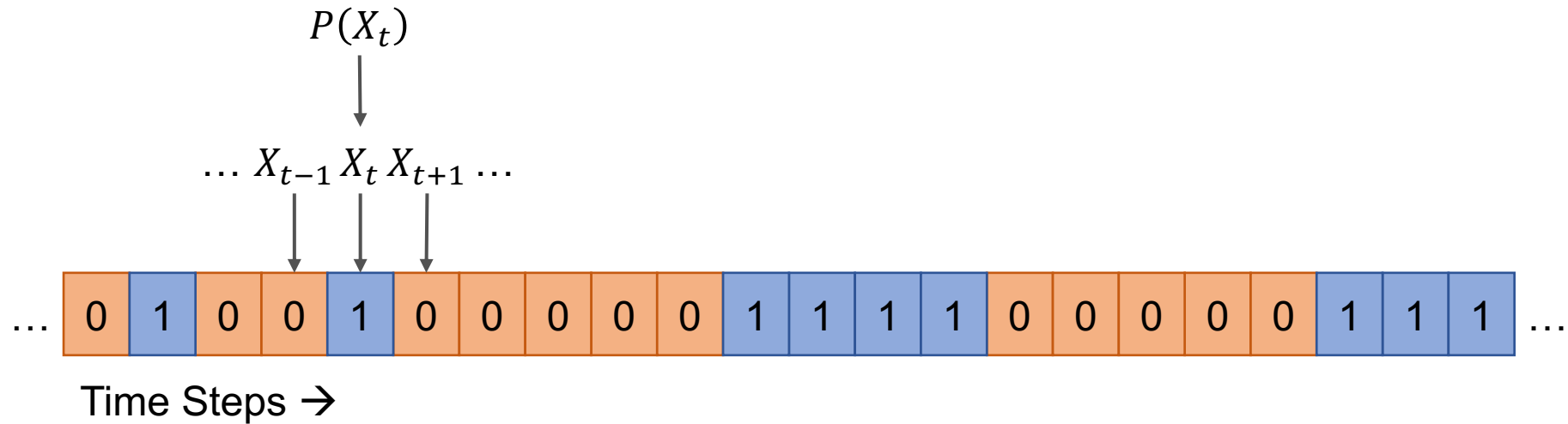
- Stochastic Processes
- ϵ -machine
- Quantum Circuits
- q -simulator
- Quantum Learning Algorithm
- Results
- Conclusion

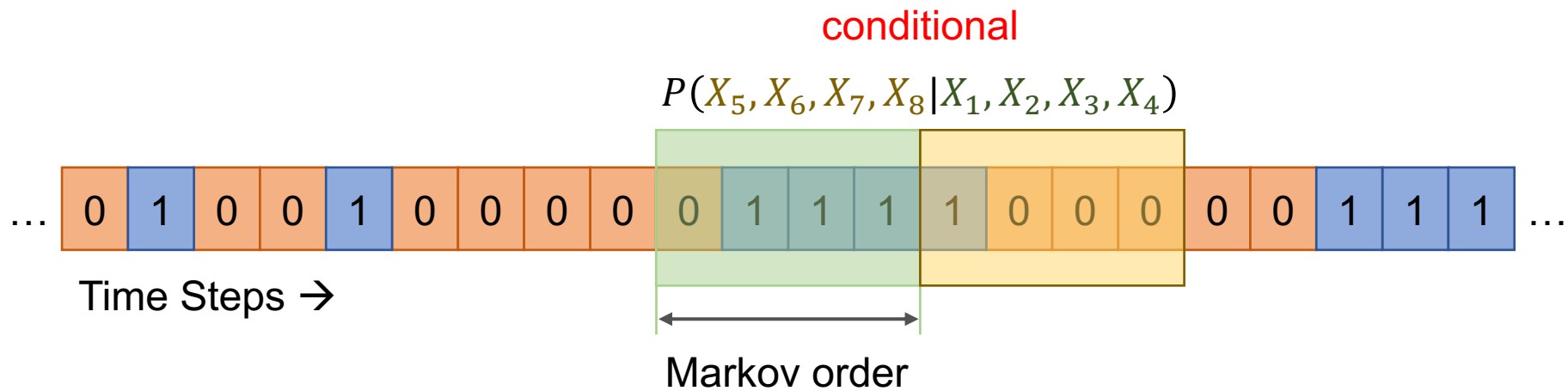


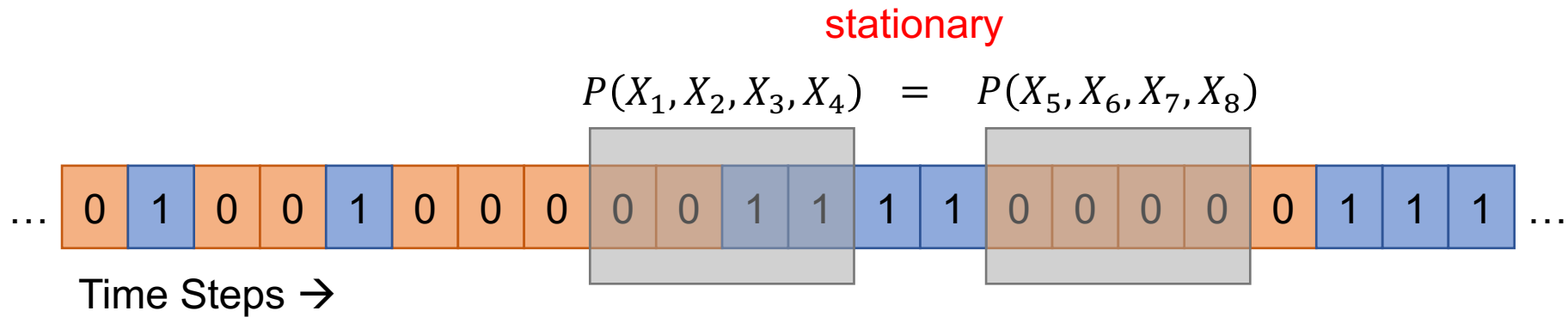


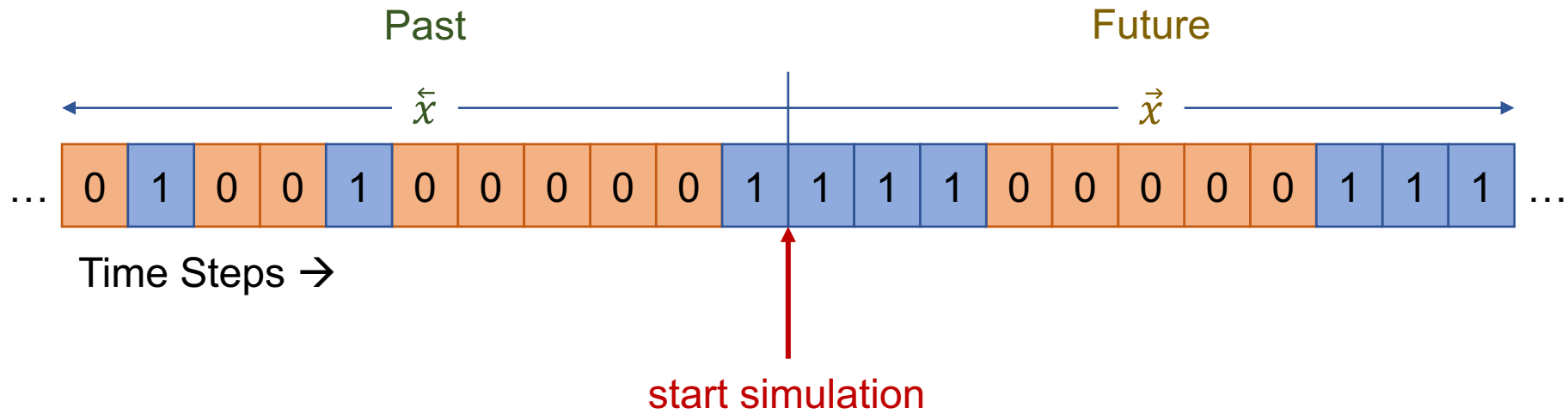
Stochastic Processes

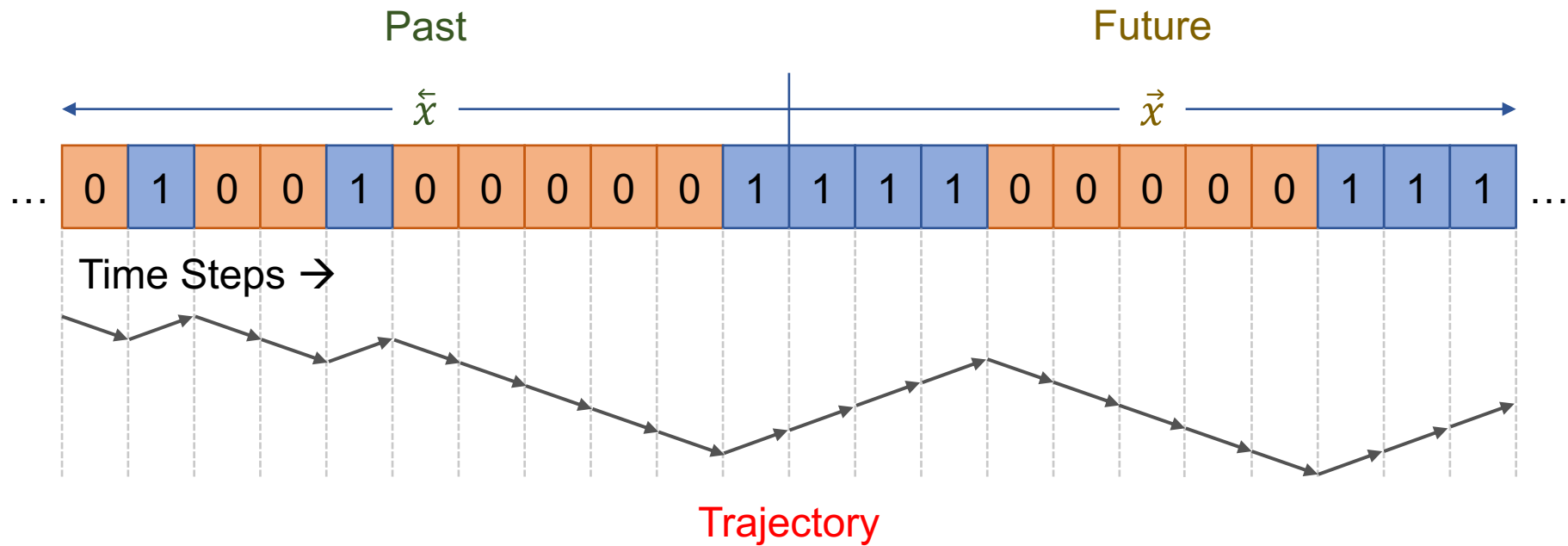




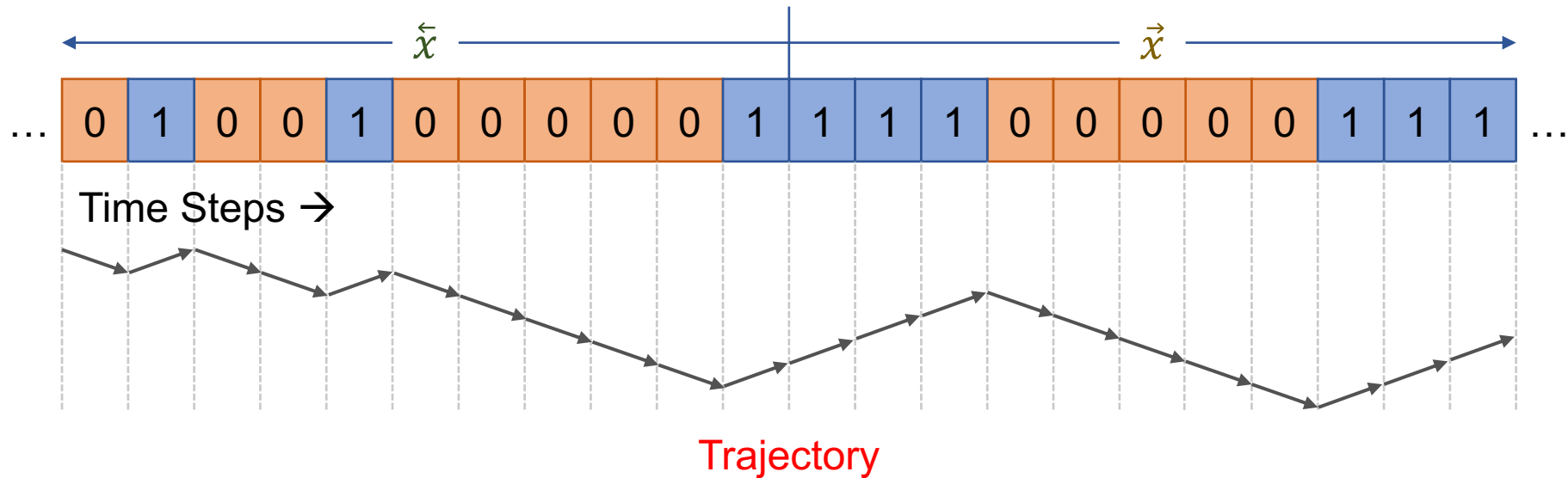






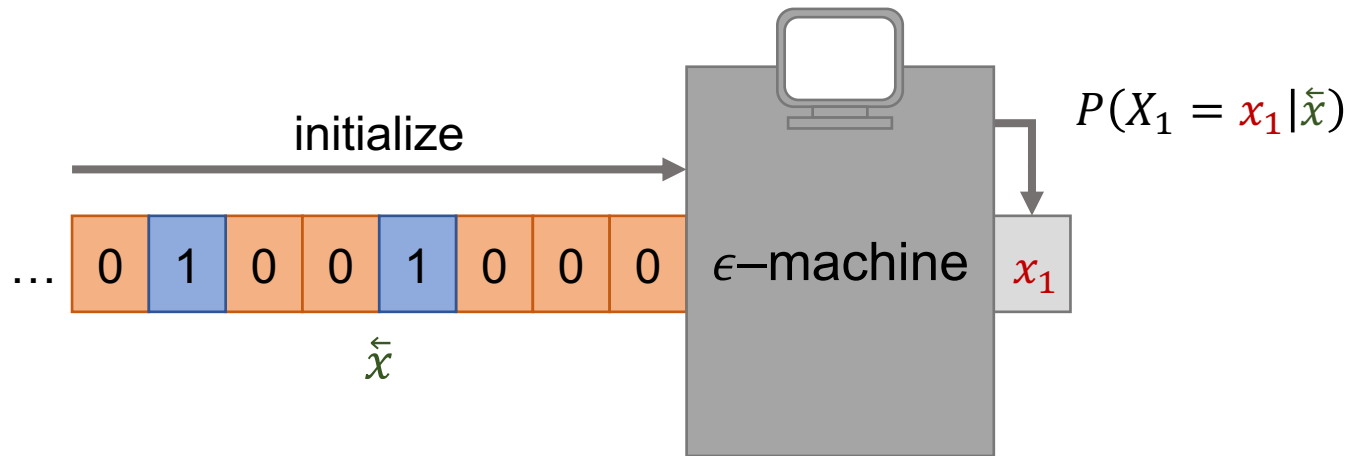


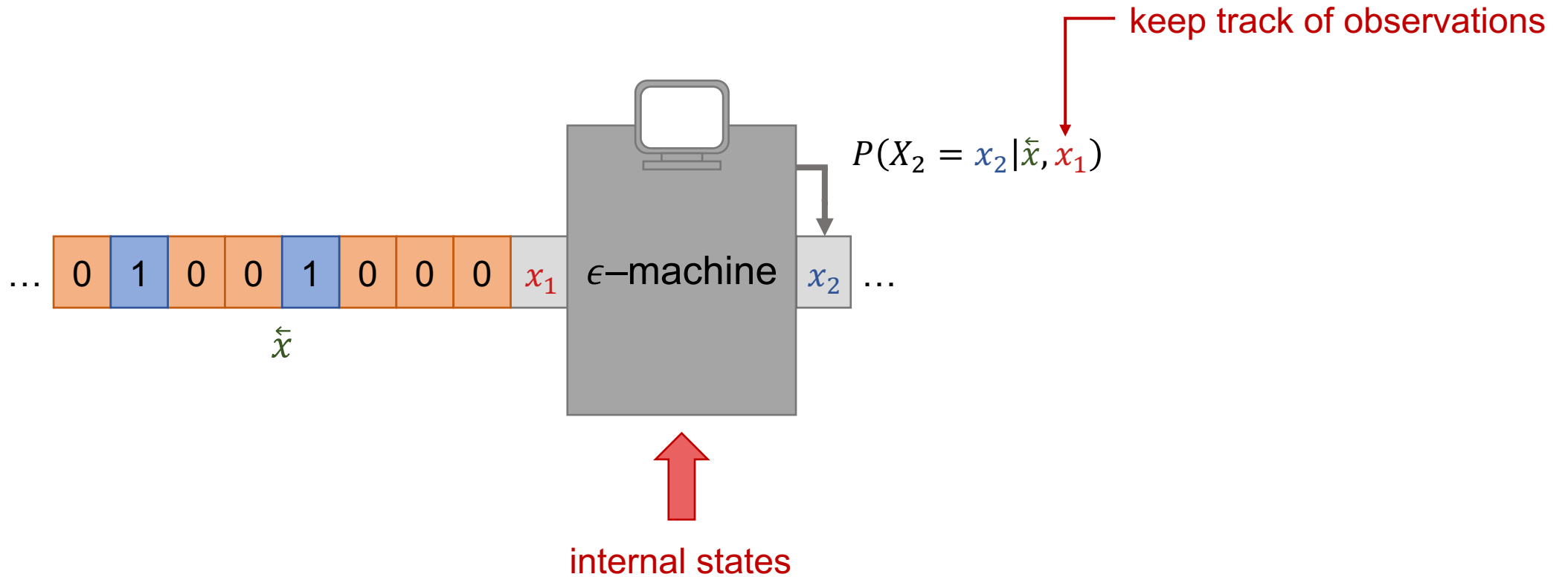
- Simulating = sampling trajectories
- Trajectory is governed by $P(\vec{X}|\hat{X})$

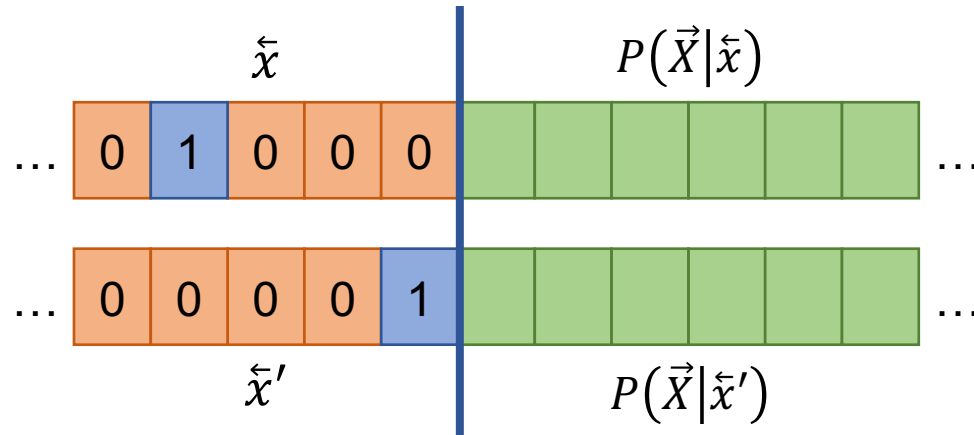


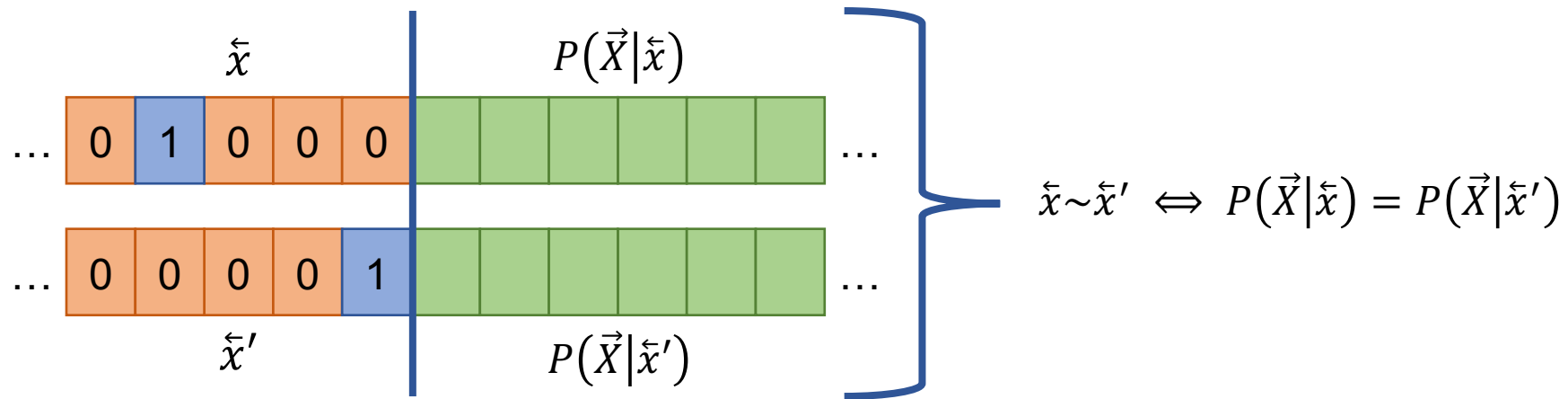


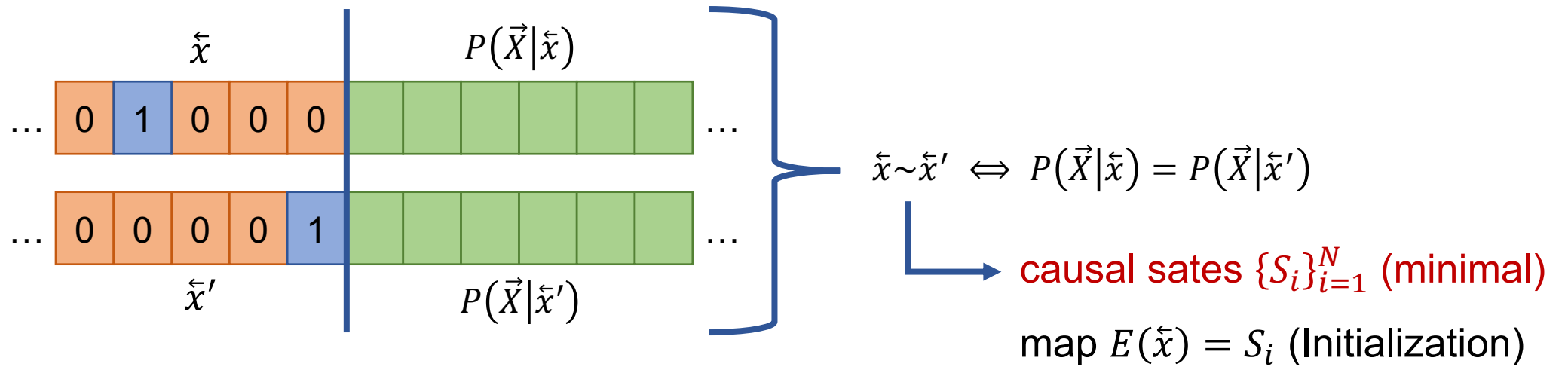
ϵ -machine





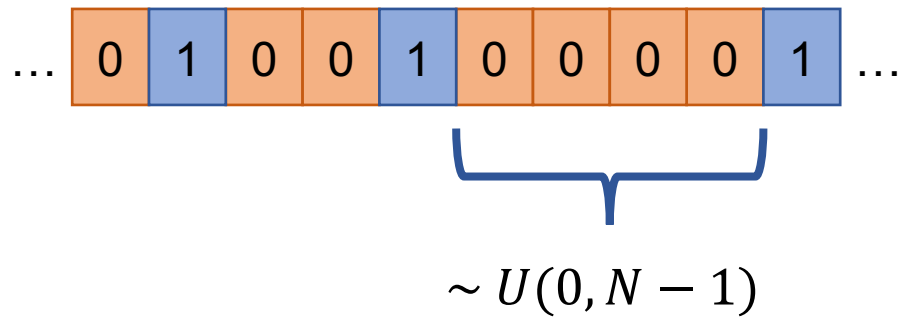





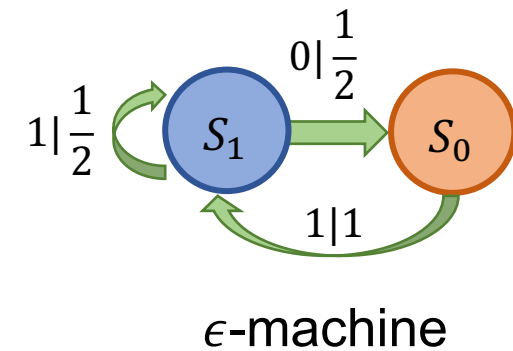


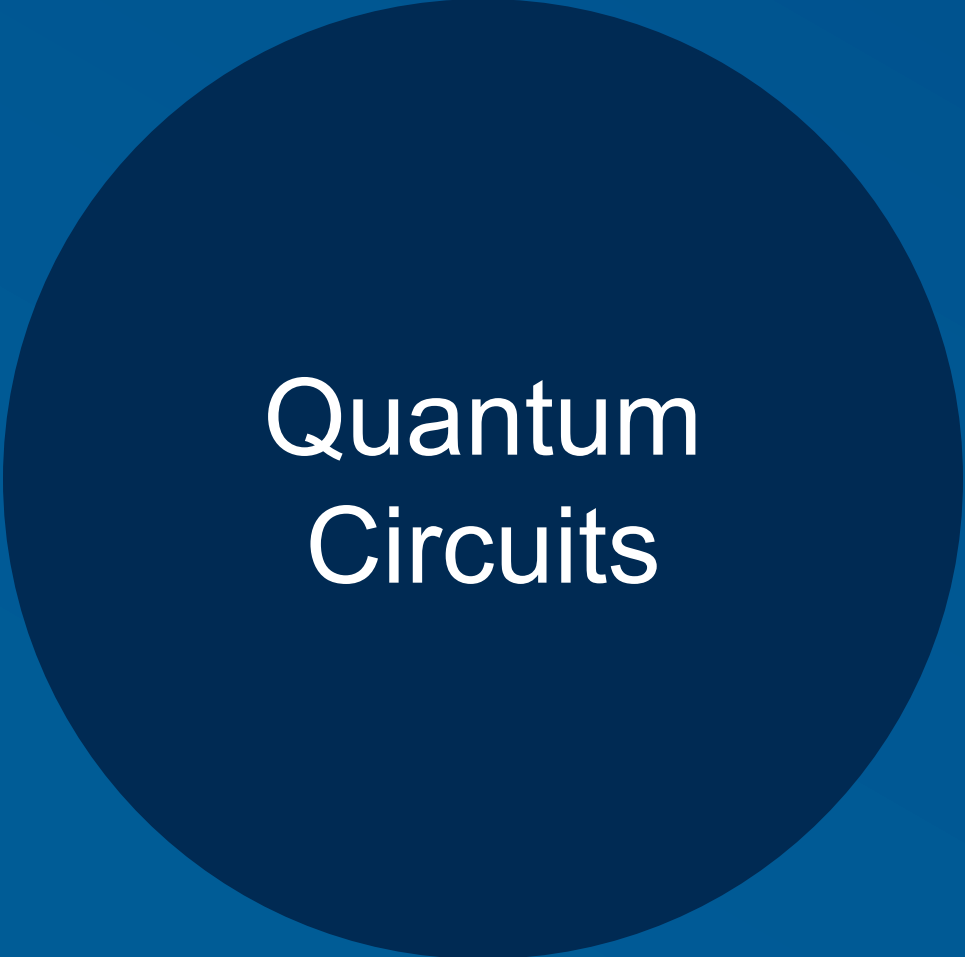
Classical Topological Complexity: $d_c = \log_2 N$

Period- N Uniform Renewal Process

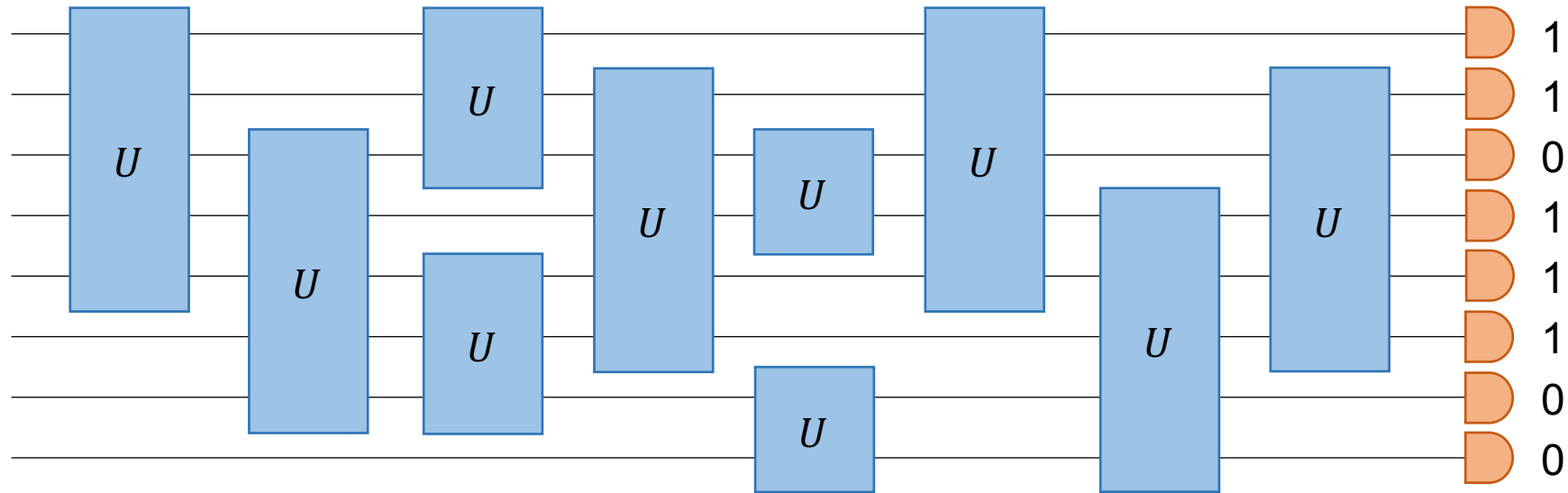


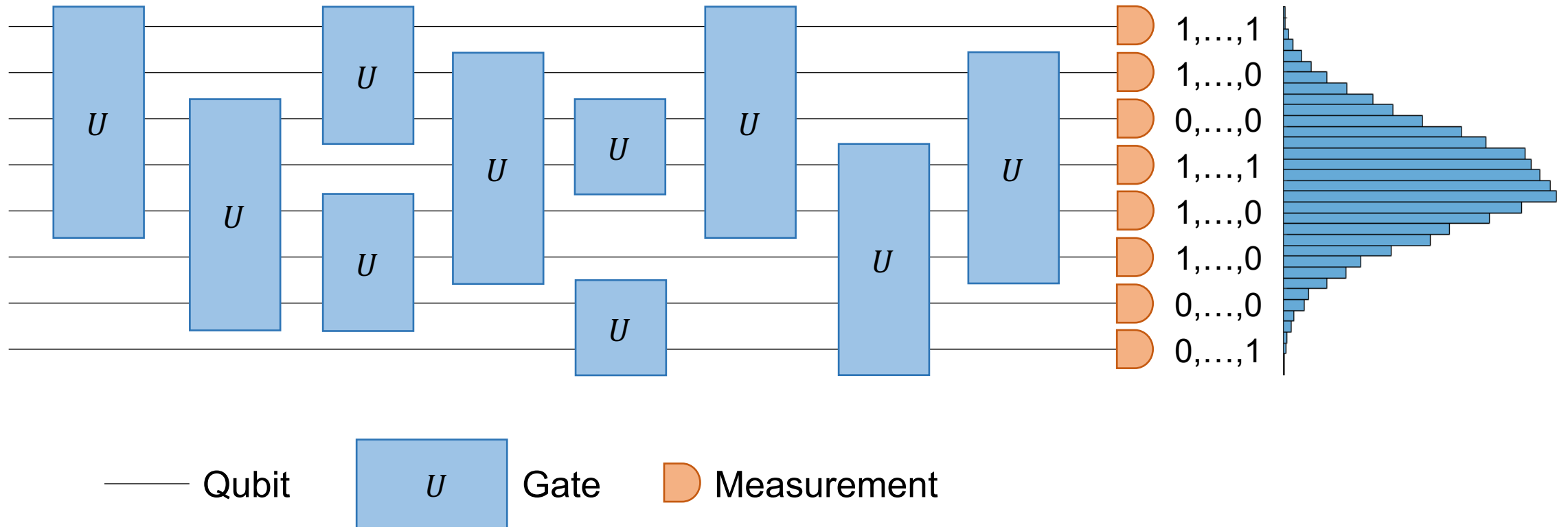
Period = 2



Quantum Circuits



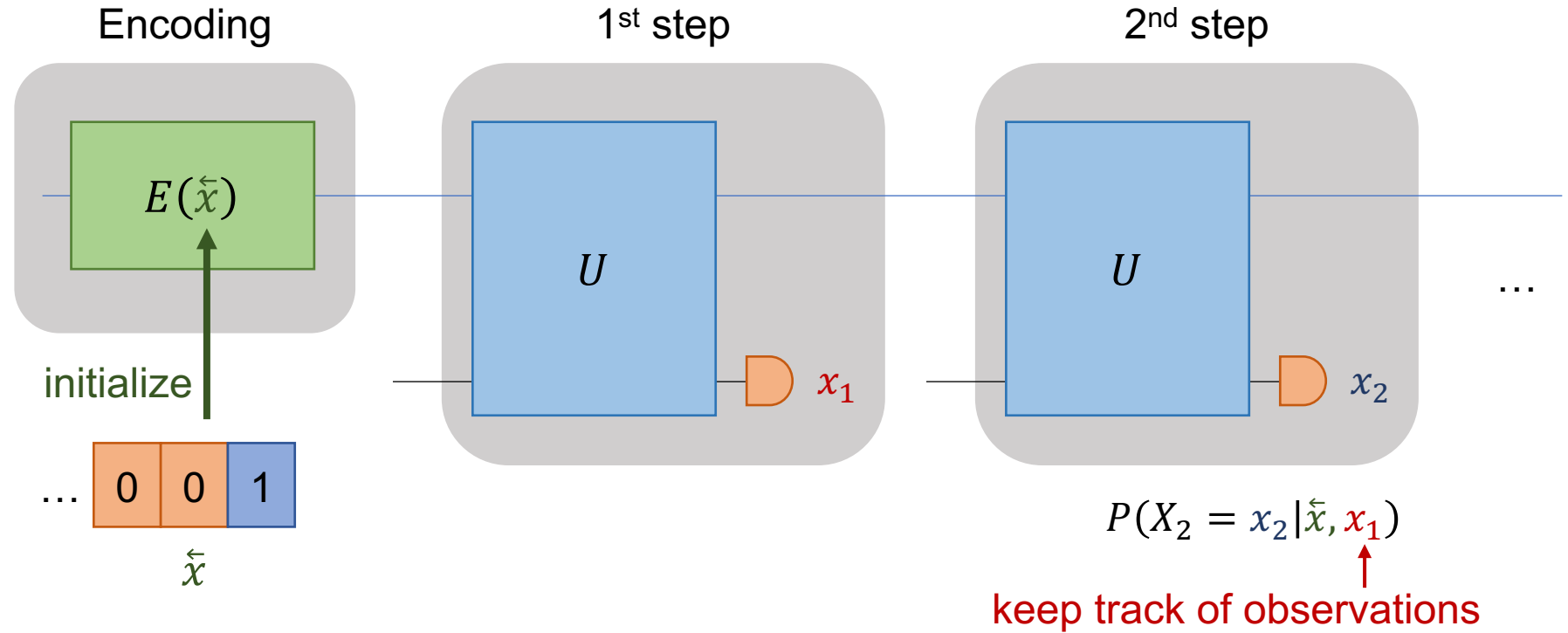


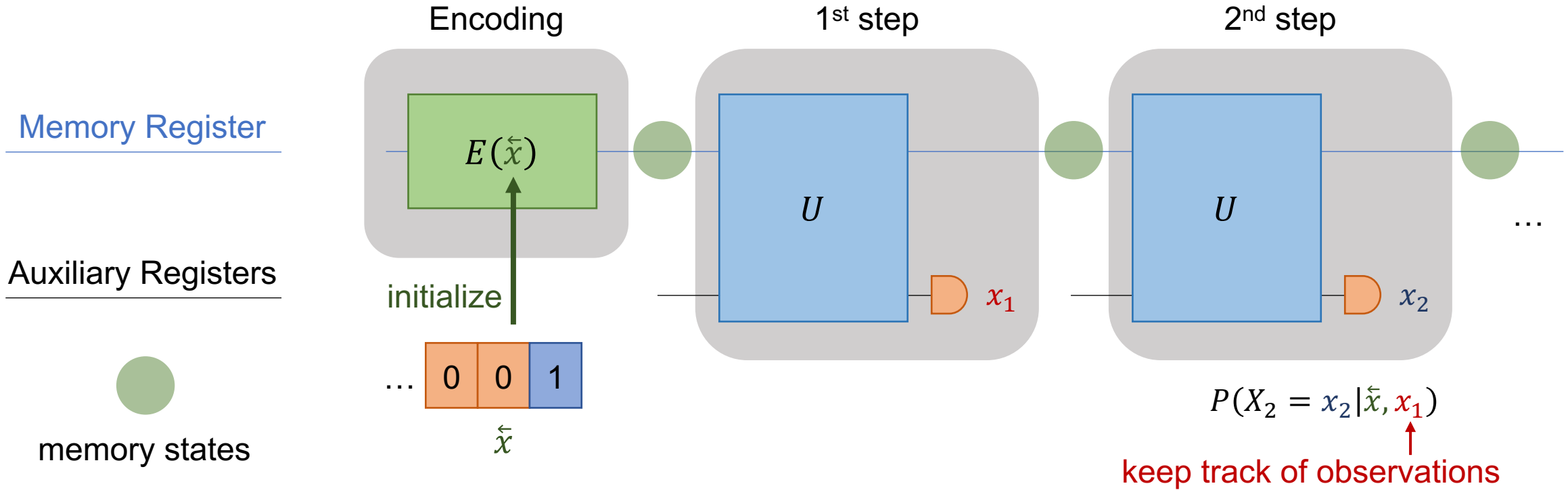


q -simulator

Memory Register

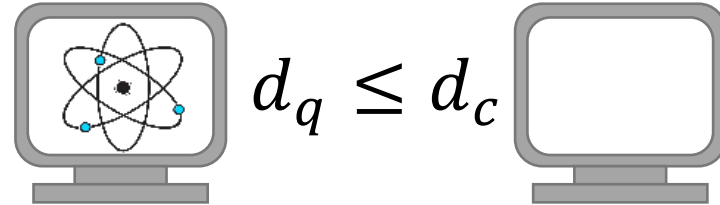
Auxiliary Registers



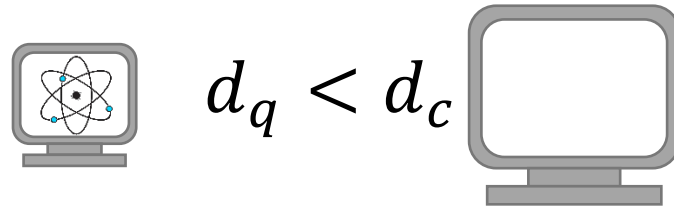


Quantum Topological Complexity: $d_q = \text{\#qubits}$

In general:

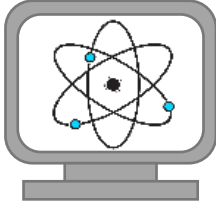



For some processes:



Thompson et al.,
10.1103/PhysRevX.8.031013

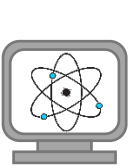
In general:




$$d_q \leq d_c$$


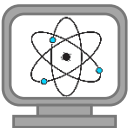
Thompson et al.,
10.1103/PhysRevX.8.031013


For some processes:



$$d_q < d_c$$


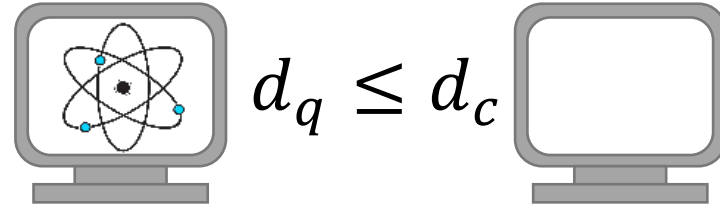
Approximate models:



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


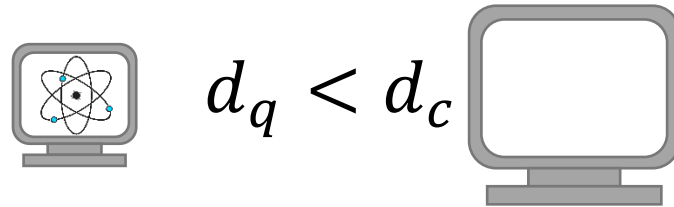
Q-Models can have better accuracy
Yang et al., arXiv:2105.14434

In general:

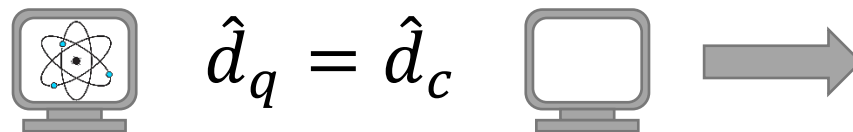


Thompson et al.,
10.1103/PhysRevX.8.031013

For some processes:

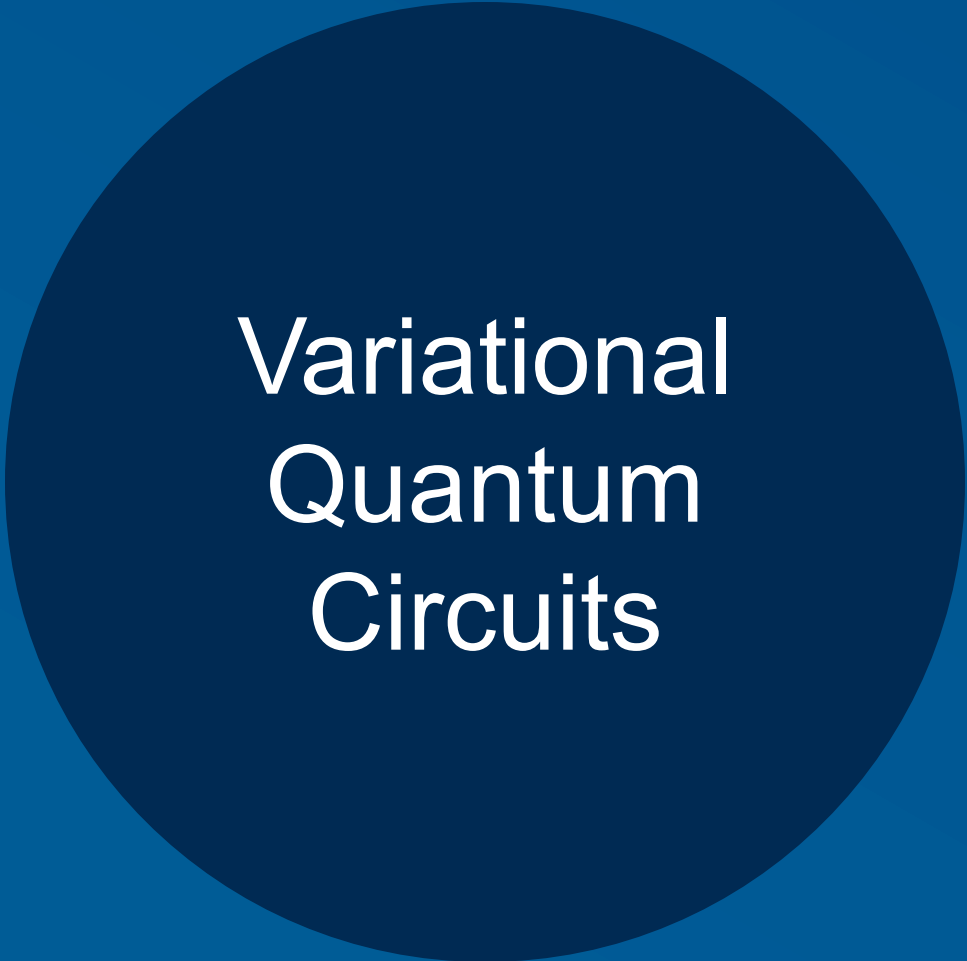


Approximate models:



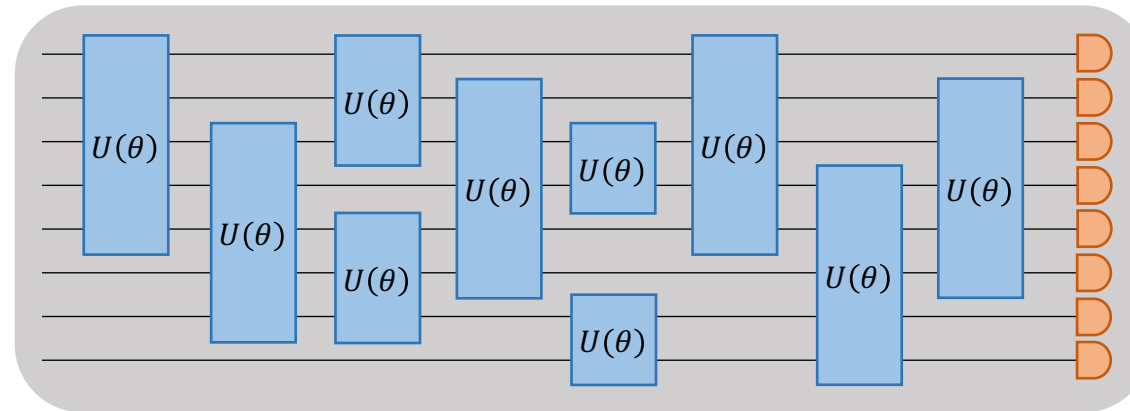
Q-Models can have better accuracy
Yang et al., arXiv:2105.14434

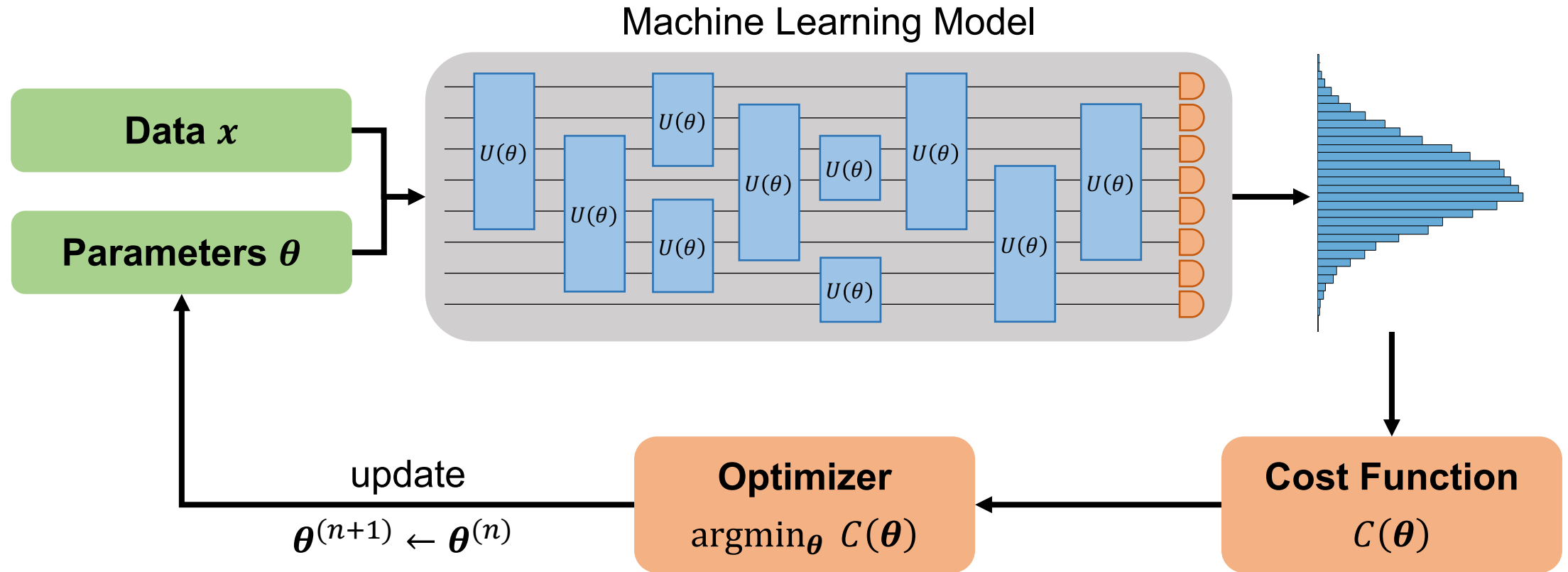
How to get a **quantum representation** of a quantum model?

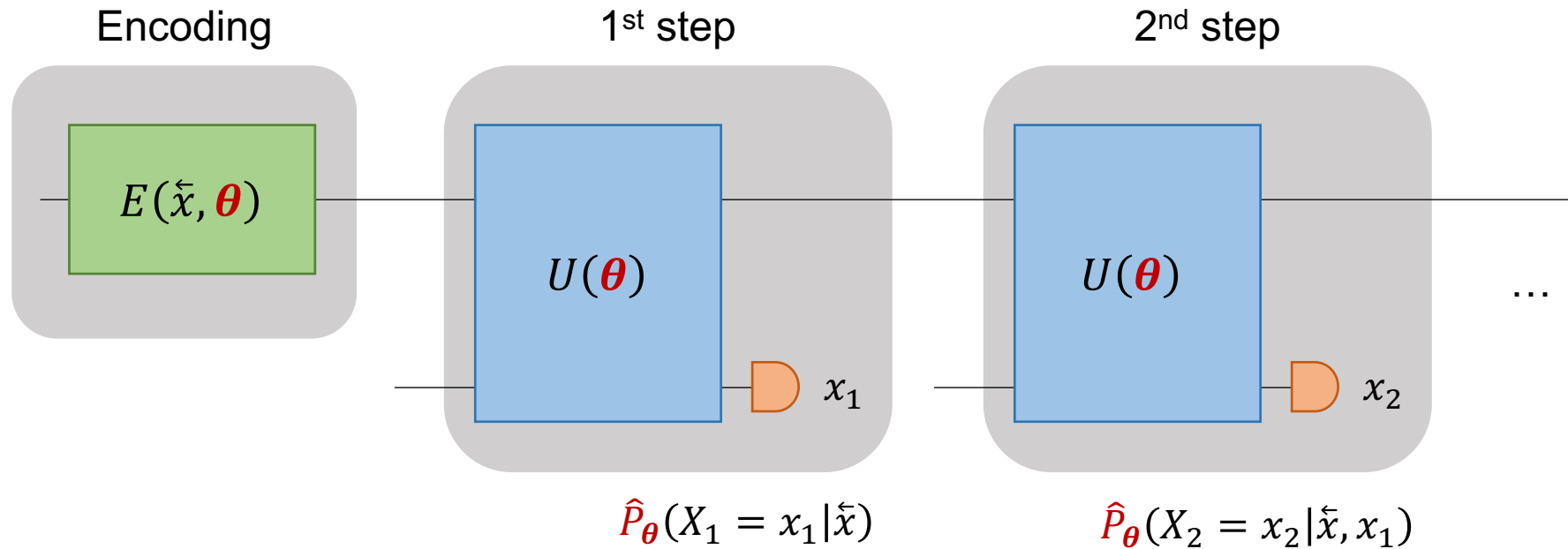


Variational Quantum Circuits

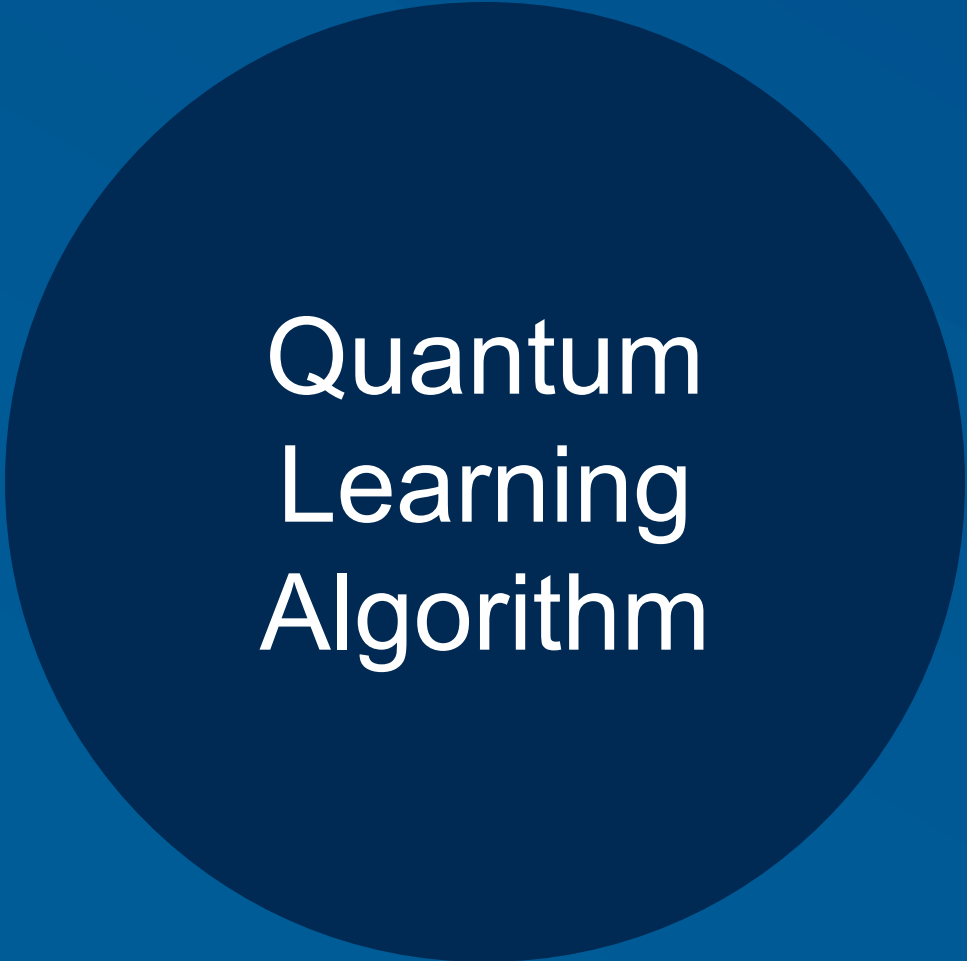
Machine Learning Model



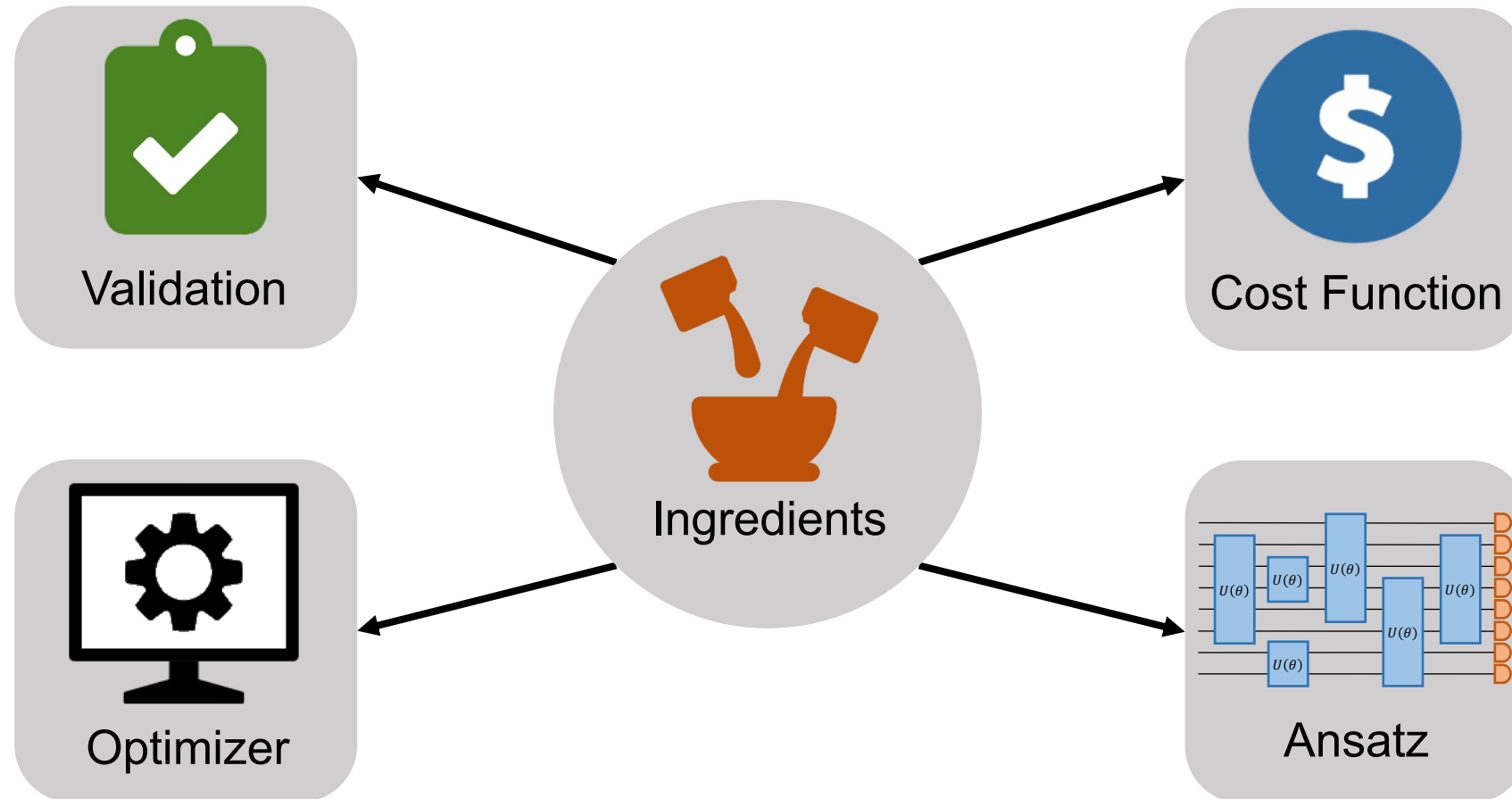


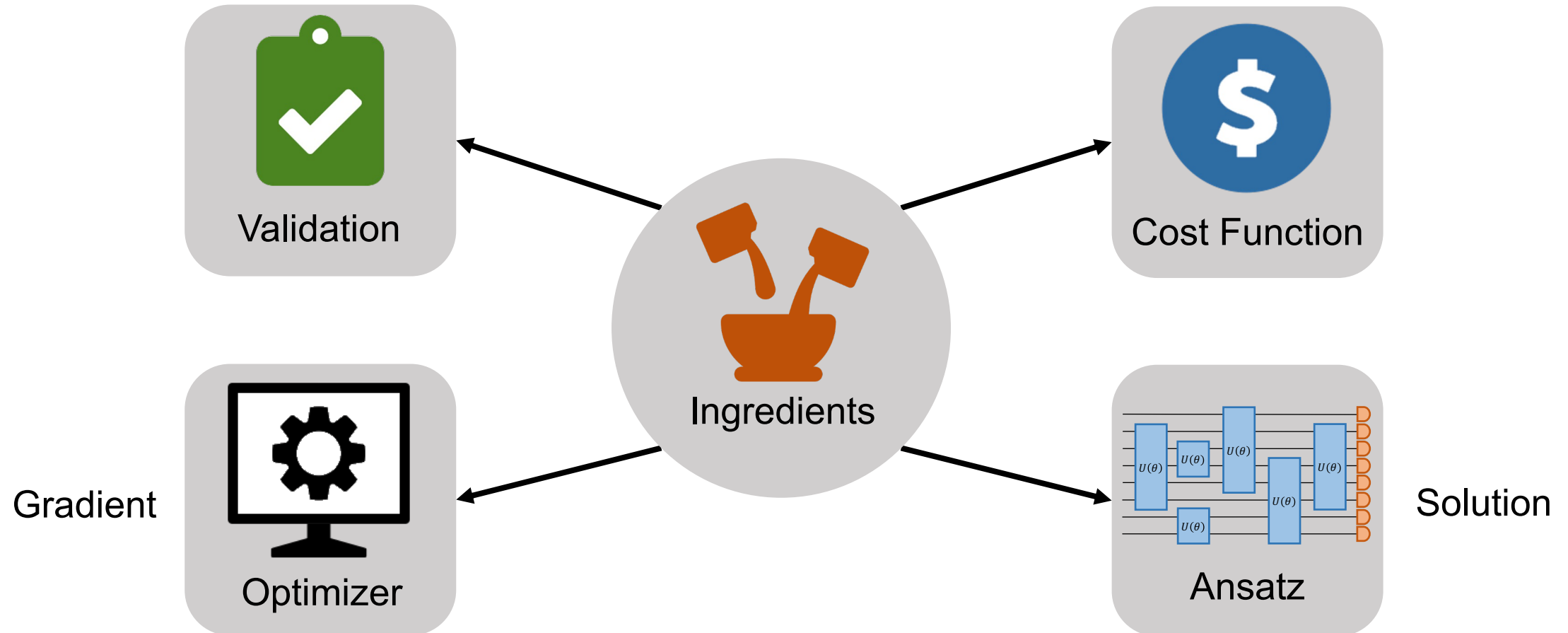


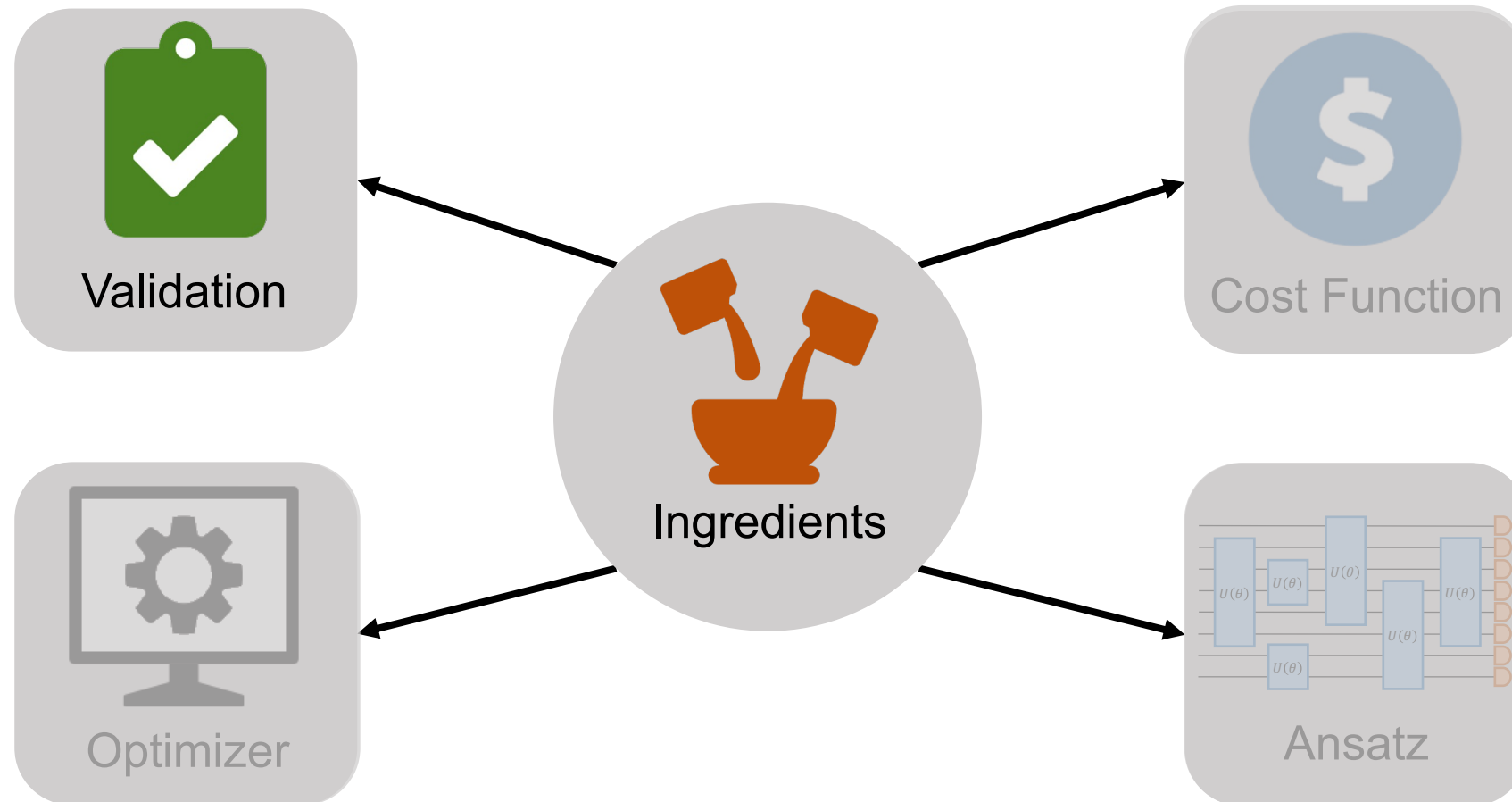
$$\text{Approximate } P \rightarrow |P - \hat{P}_\theta| < \delta$$



Quantum Learning Algorithm

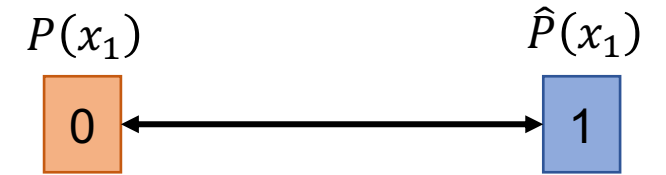






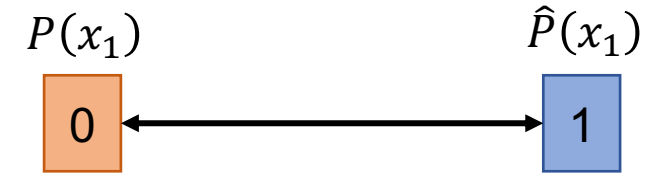
Kullback-Leibler divergence:
(KL)

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



Kullback-Leibler divergence:
(KL)

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

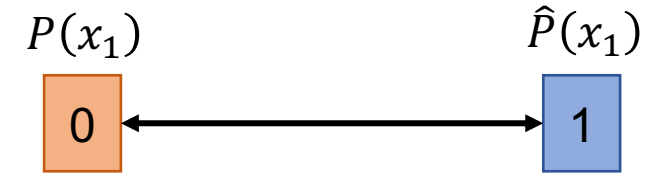


mean over time steps

average over pasts

Kullback-Leibler divergence:
(KL)

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

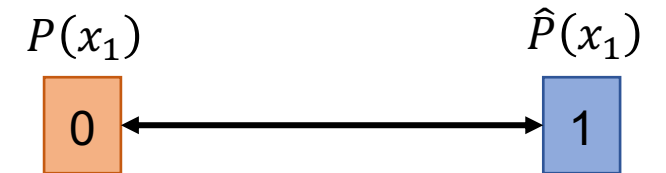


mean over time steps

average over pasts

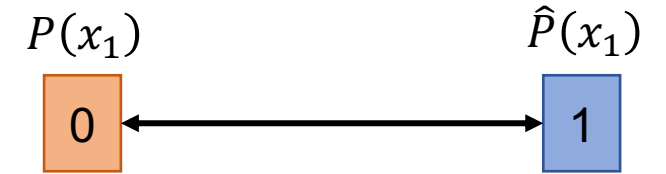
Total Variation Distance:
(TV)

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



Kullback-Leibler divergence:
(KL)

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

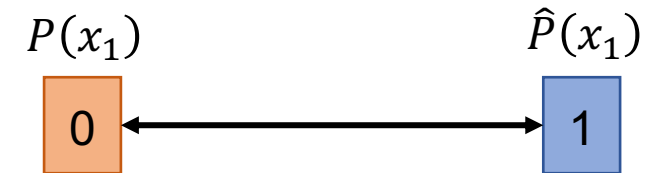


mean over time steps

average over pasts

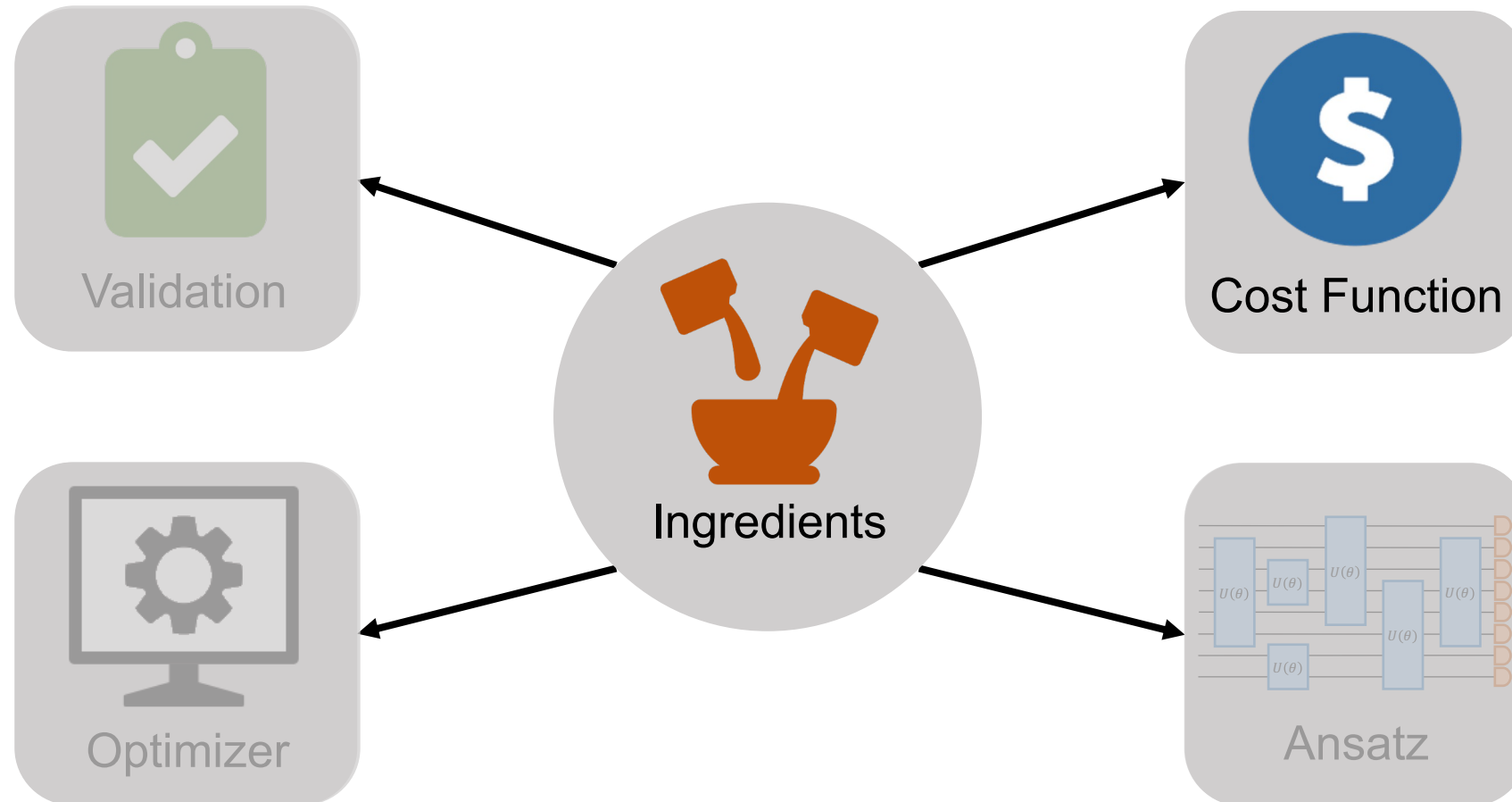
Total Variation Distance:
(TV)

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



sum up time steps

sum up pasts



Ideally, use validation metric:

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

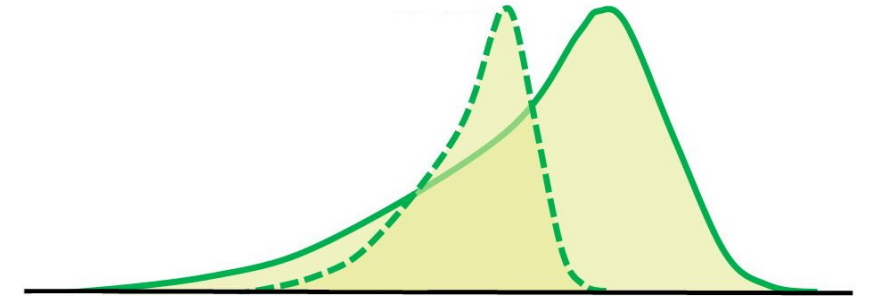
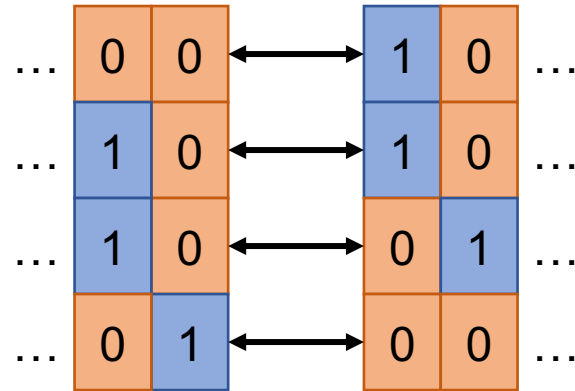
Ideally, use validation metric:

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

unknown

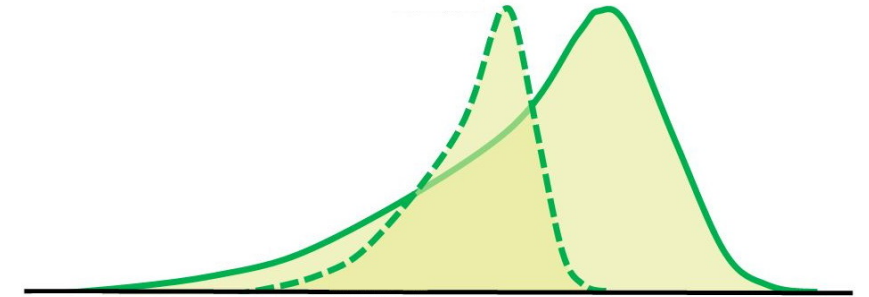
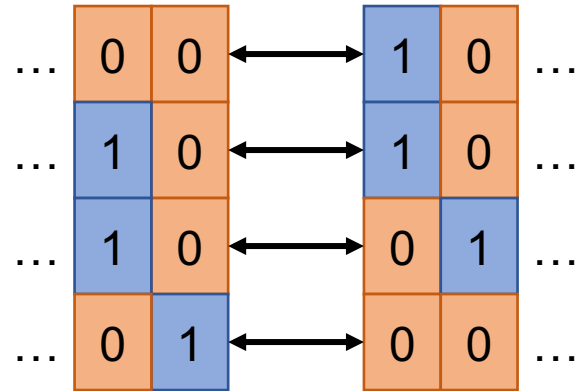
inefficient

Maximum Mean Discrepancy:
(MMD)

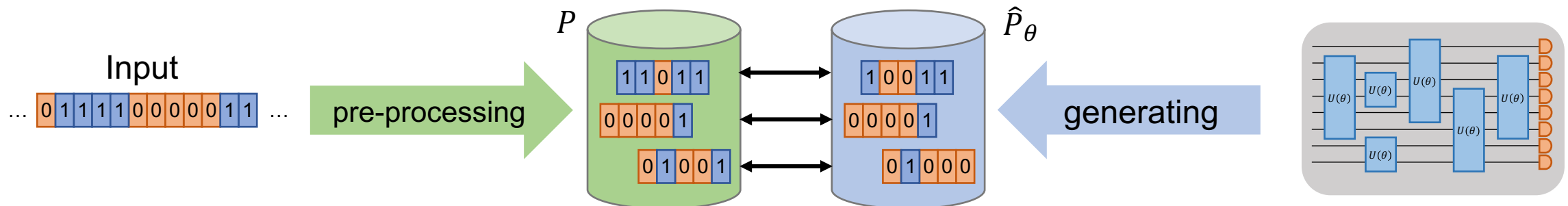


$$\text{MMD}[P, \hat{P}] = 0 \iff P = \hat{P}$$

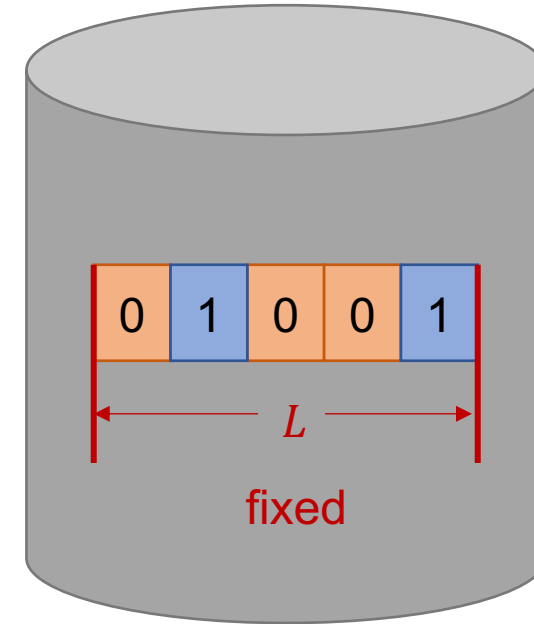
Maximum Mean Discrepancy:
(MMD)



$$\text{MMD}[P, \hat{P}] = 0 \iff P = \hat{P}$$



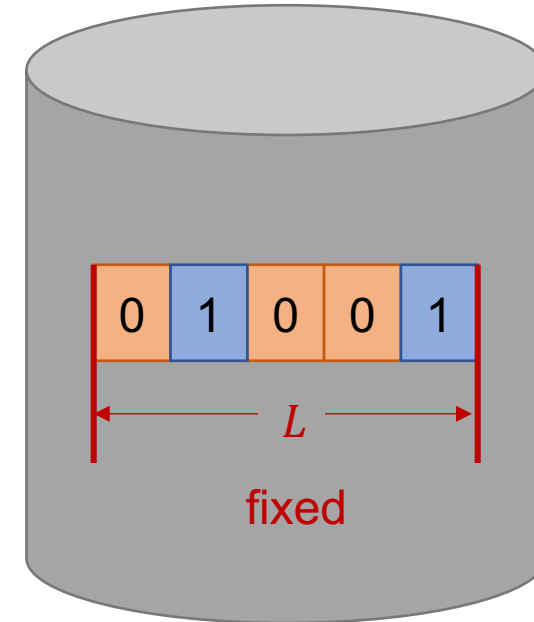
$$C(\boldsymbol{\theta}) = \sum_{\tilde{x}} w_{\tilde{x}} \cdot \text{MMD}^2[\mathbf{P}, \hat{\mathbf{P}}_{\boldsymbol{\theta}} | \tilde{x}]$$



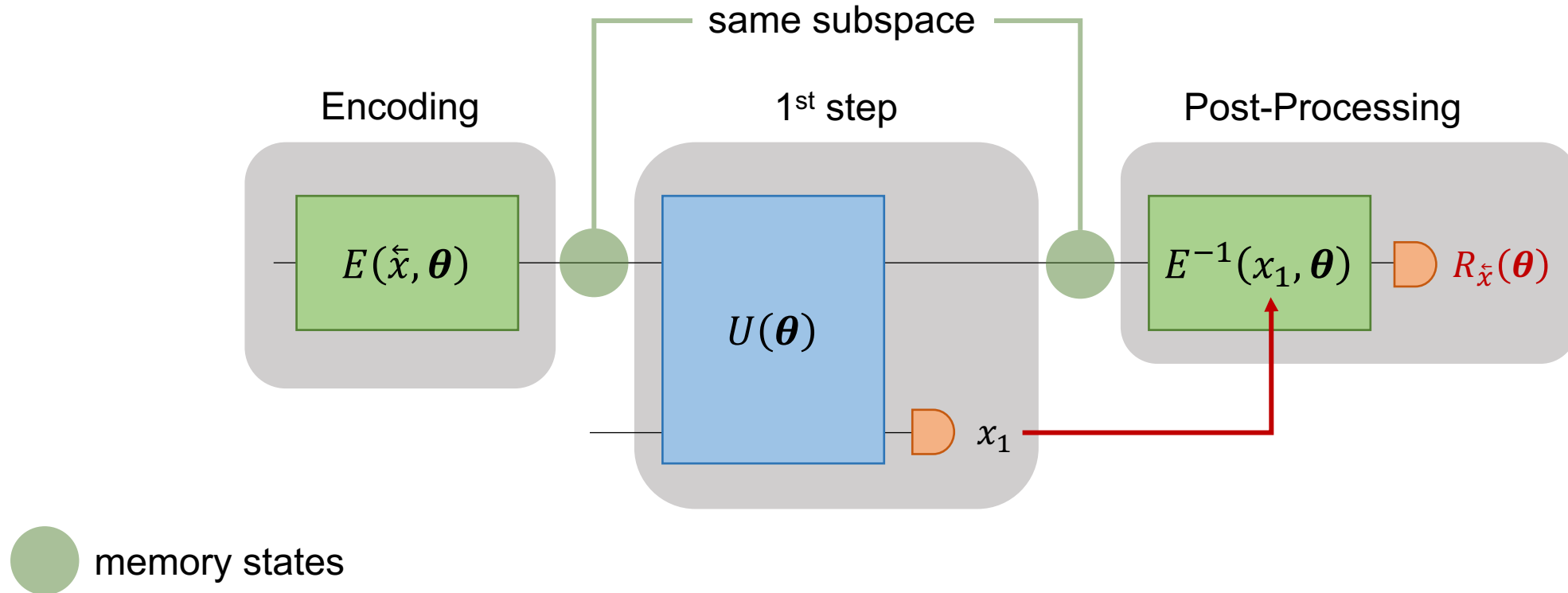
$$C(\theta) = \sum_{\tilde{x}} w_{\tilde{x}} \cdot \text{MMD}^2[P, \hat{P}_{\theta}|\tilde{x}]$$



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

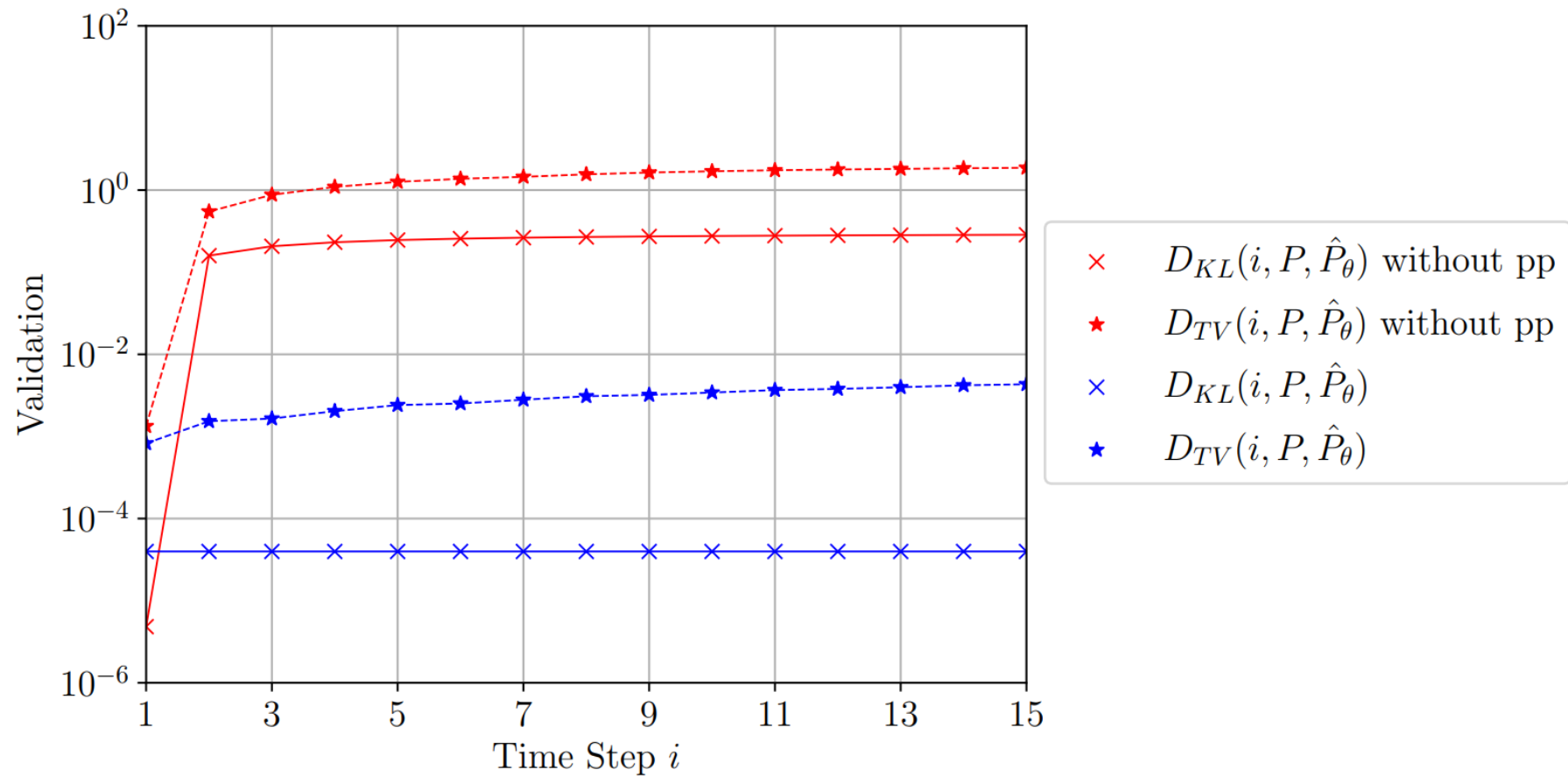


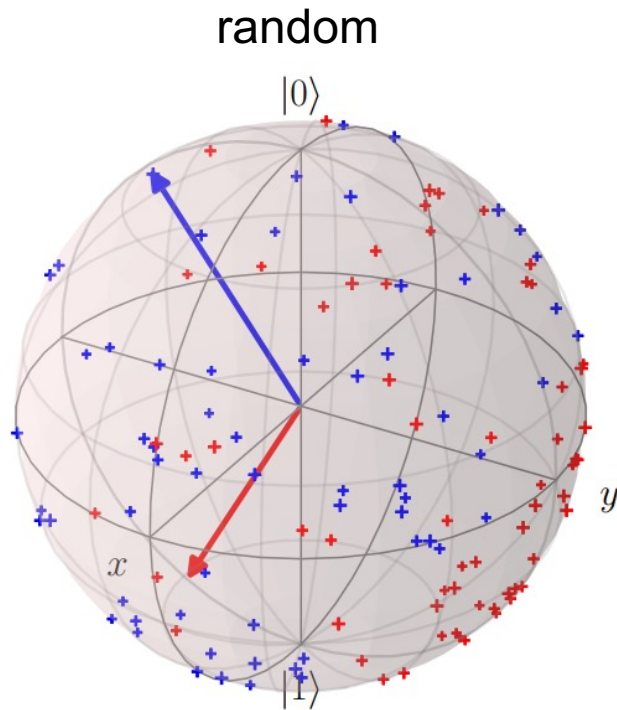
Regularization = penalizes models with a large set of memory states





Results

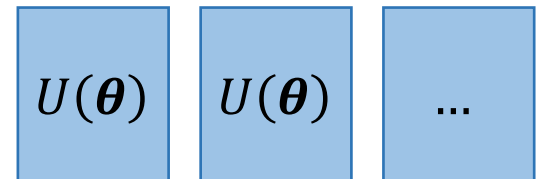


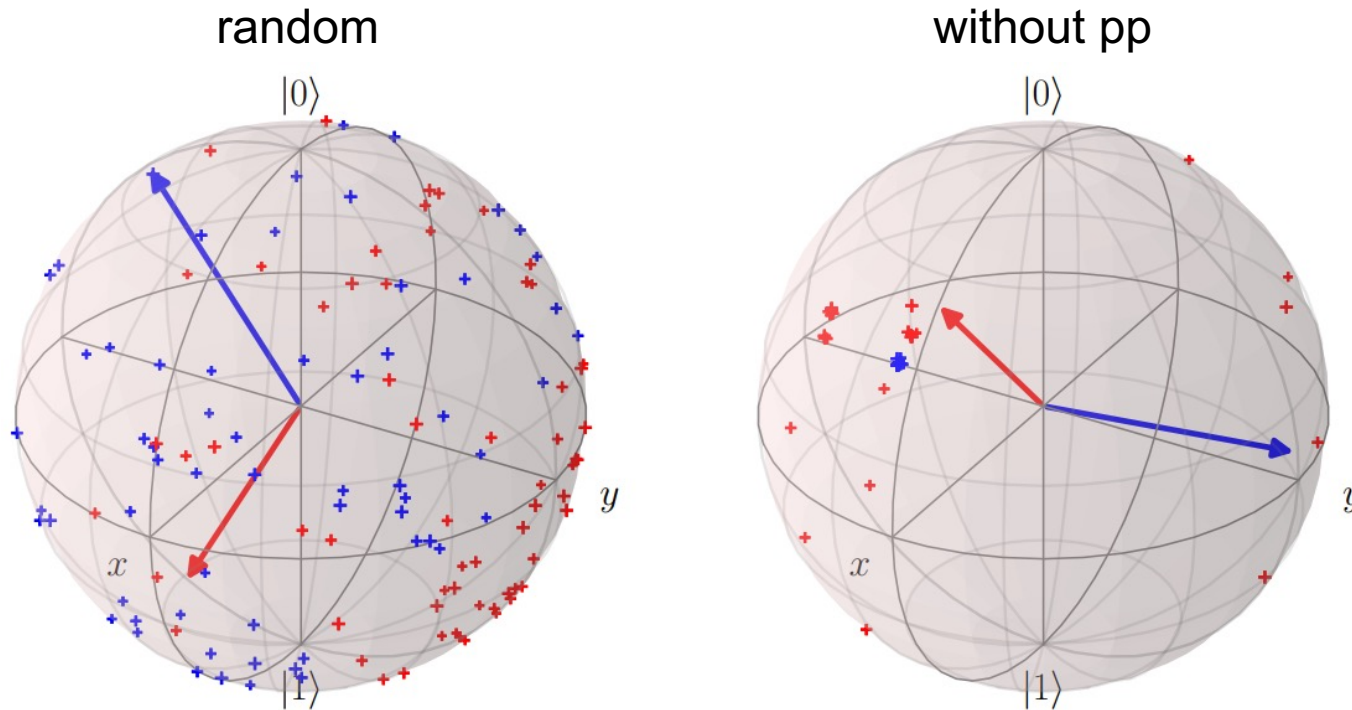


$$E(\tilde{x}, \theta)$$

\uparrow initial state for $\tilde{x} = 0$
 \uparrow initial state for $\tilde{x} = 1$

$+$ memory states for $x_i = 0$
 $+$ memory states for $x_i = 1$

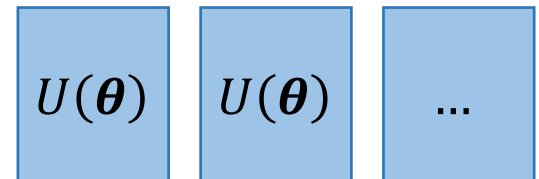


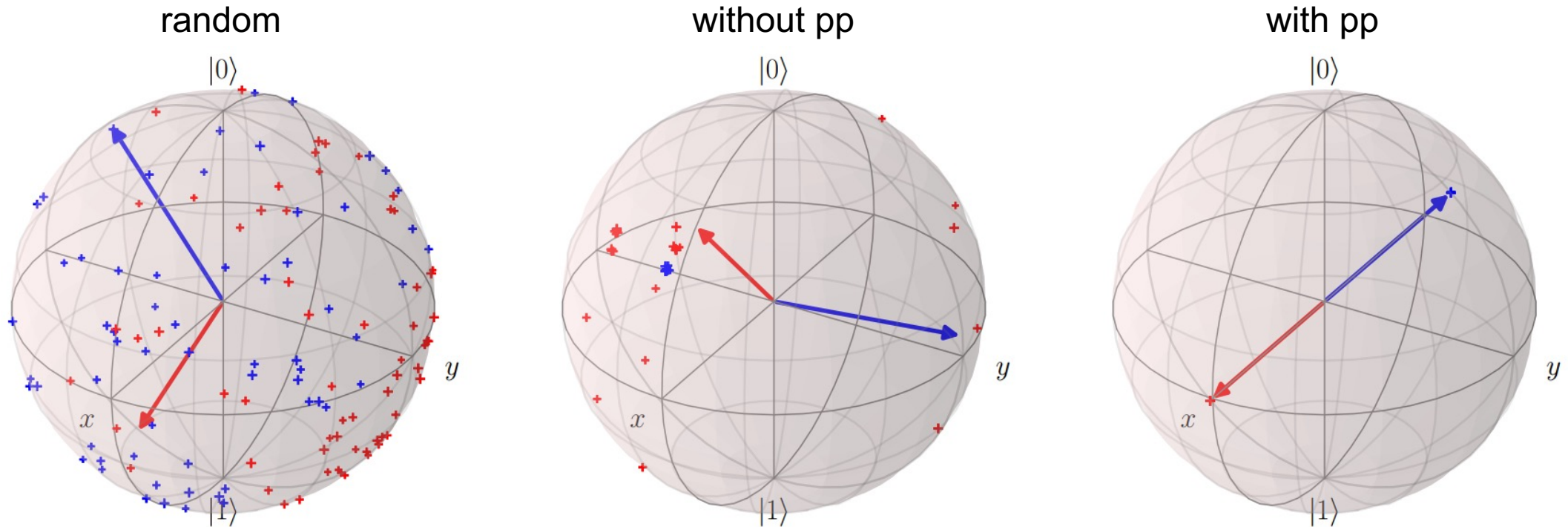


$$E(\tilde{x}, \theta)$$

\uparrow initial state for $\tilde{x} = 0$
 \uparrow initial state for $\tilde{x} = 1$

$+$ memory states for $x_i = 0$
 $+$ memory states for $x_i = 1$

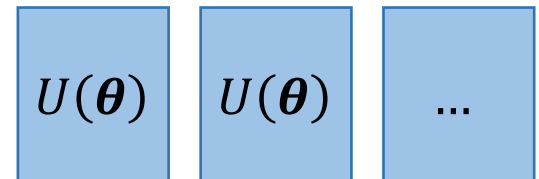





$$E(\tilde{x}, \theta)$$

\uparrow initial state for $\tilde{x} = 0$
 \uparrow initial state for $\tilde{x} = 1$

$+$ memory states for $x_i = 0$
 $+$ memory states for $x_i = 1$



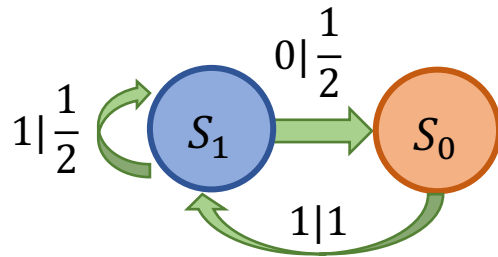
- Developed a hybrid quantum learning algorithm for predictive models
 - Learning algorithm is memory efficient
 - Extended MMD for predictive models \rightarrow Decrease KL and TV
 - Regularization \rightarrow small set of memory states
 - Learned models show constantly good simulation performance
-
- Apply the algorithm to the period-3 uniform renewal process
 - Consider processes with outcomes $\subseteq \mathbb{N}$



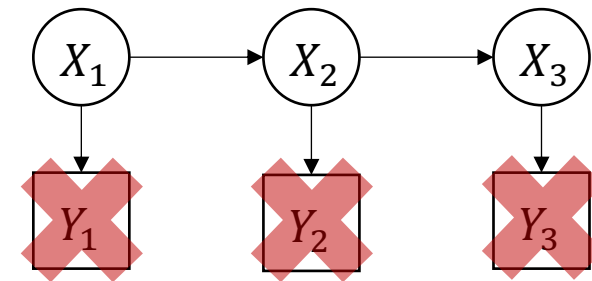
Thank you
very much!

Questions
are highly
welcome.

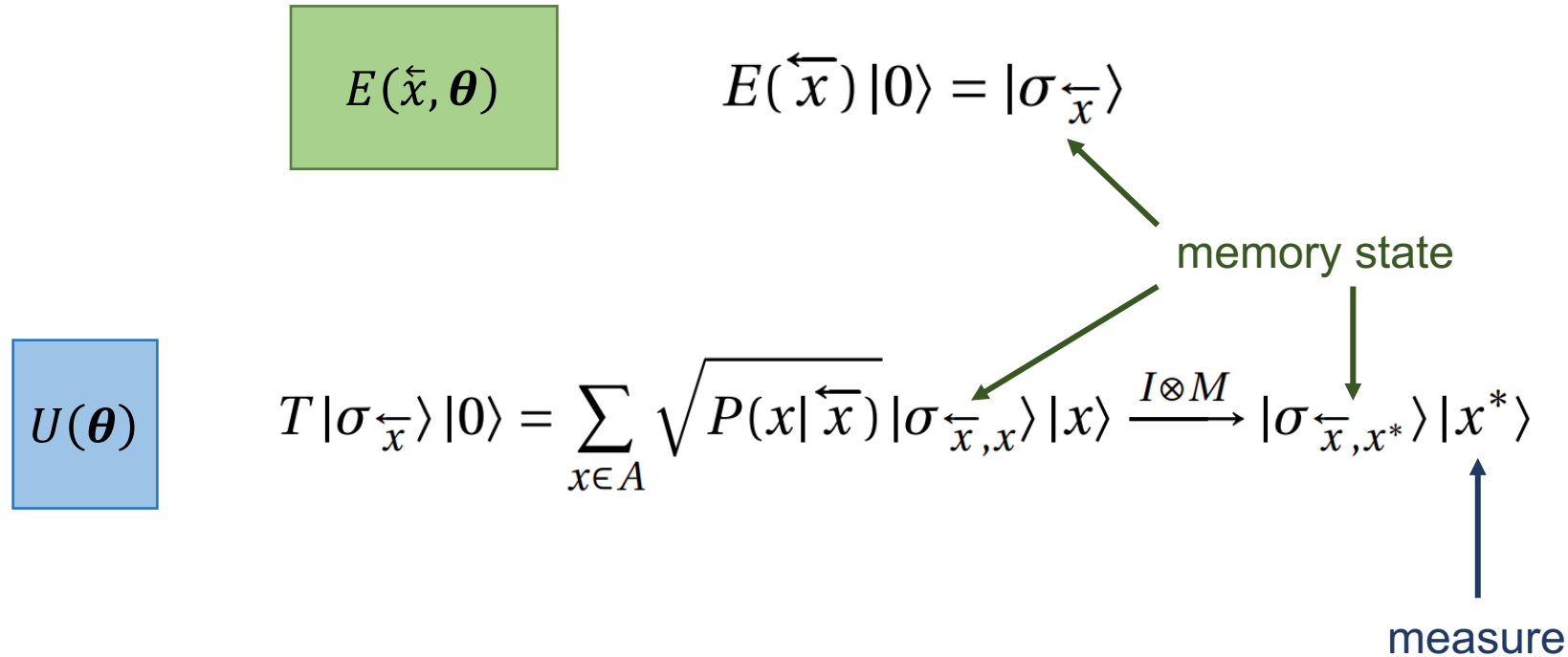
ϵ -machine



HMM



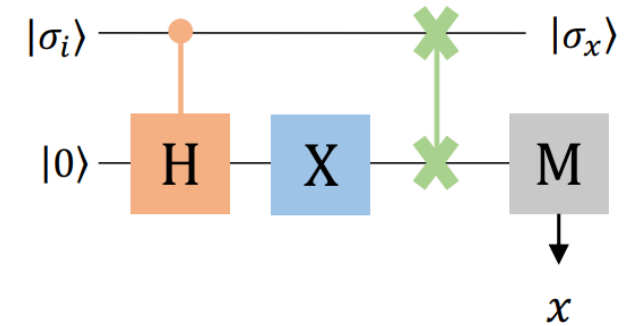
unifilar



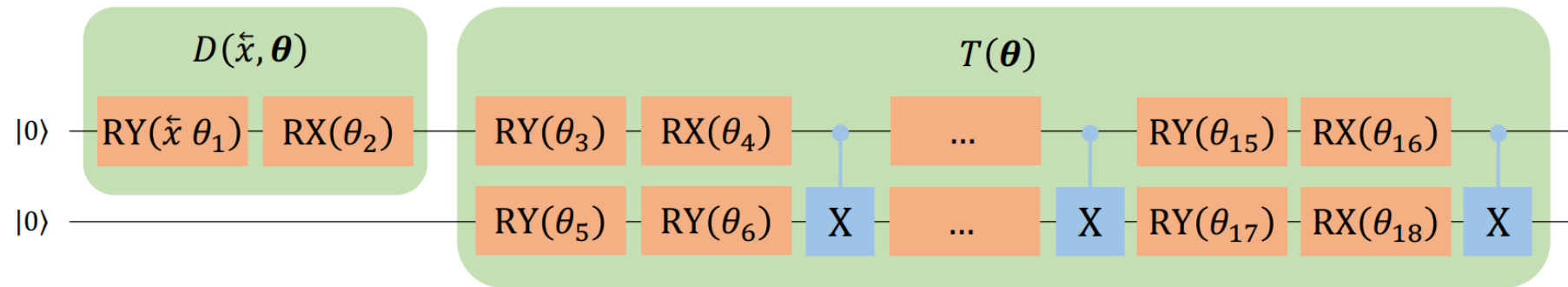
Analytical solution:

$$|0\rangle \xrightarrow{X} |1\rangle = |\sigma_0\rangle$$

$$|0\rangle \xrightarrow{H} |+\rangle = |\sigma_1\rangle$$



Ansatz:



Kullback-Leibler divergence:
(KL)

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

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

Total Variation Distance:
(TV)

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

$$D_{TV}(L, P, \hat{P}) = \frac{1}{2} \sum_{\mathbf{x}_{1:L}} \sum_{\tilde{\mathbf{x}}} |P(\mathbf{x}_{1:L} | \tilde{\mathbf{x}}) - \hat{P}(\mathbf{x}_{1:L} | \tilde{\mathbf{x}})|$$

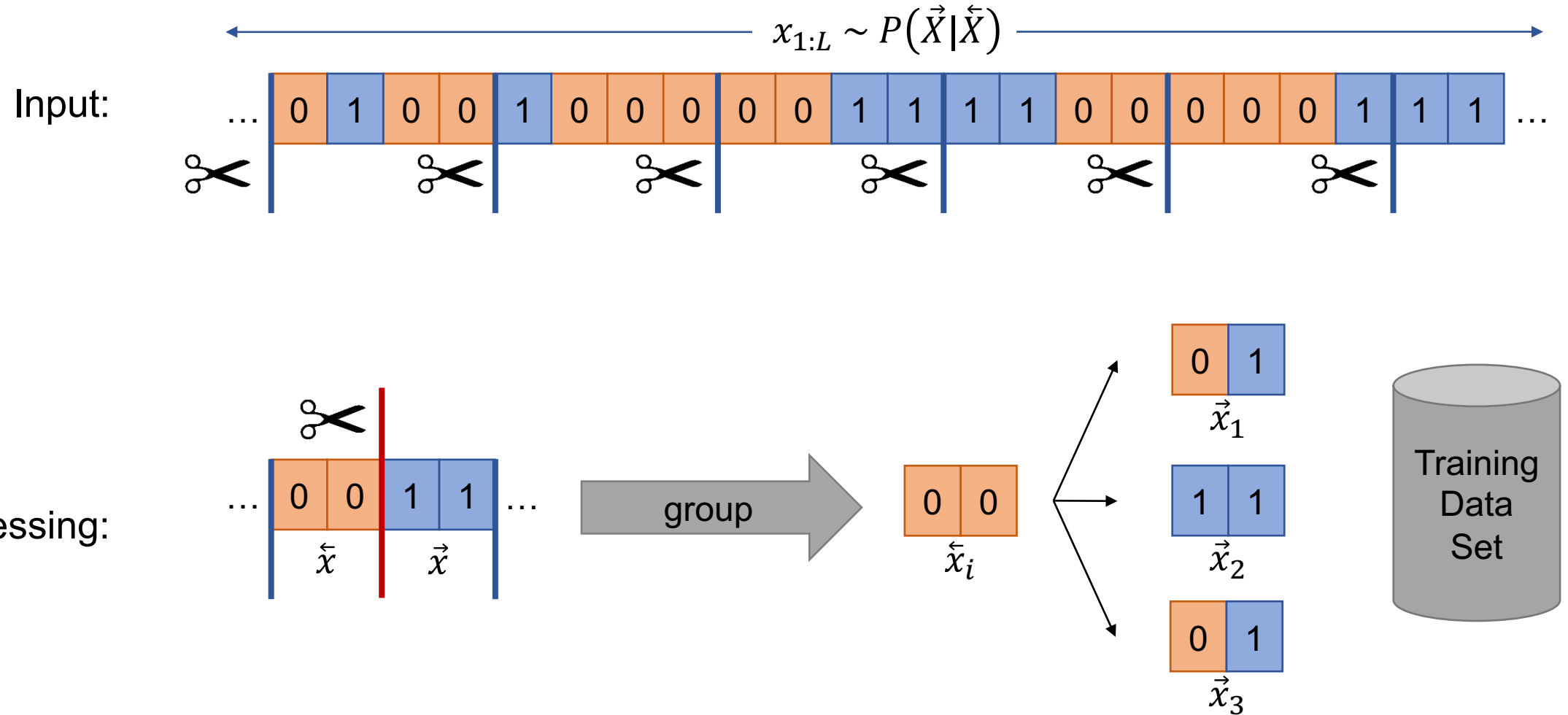
Kernel

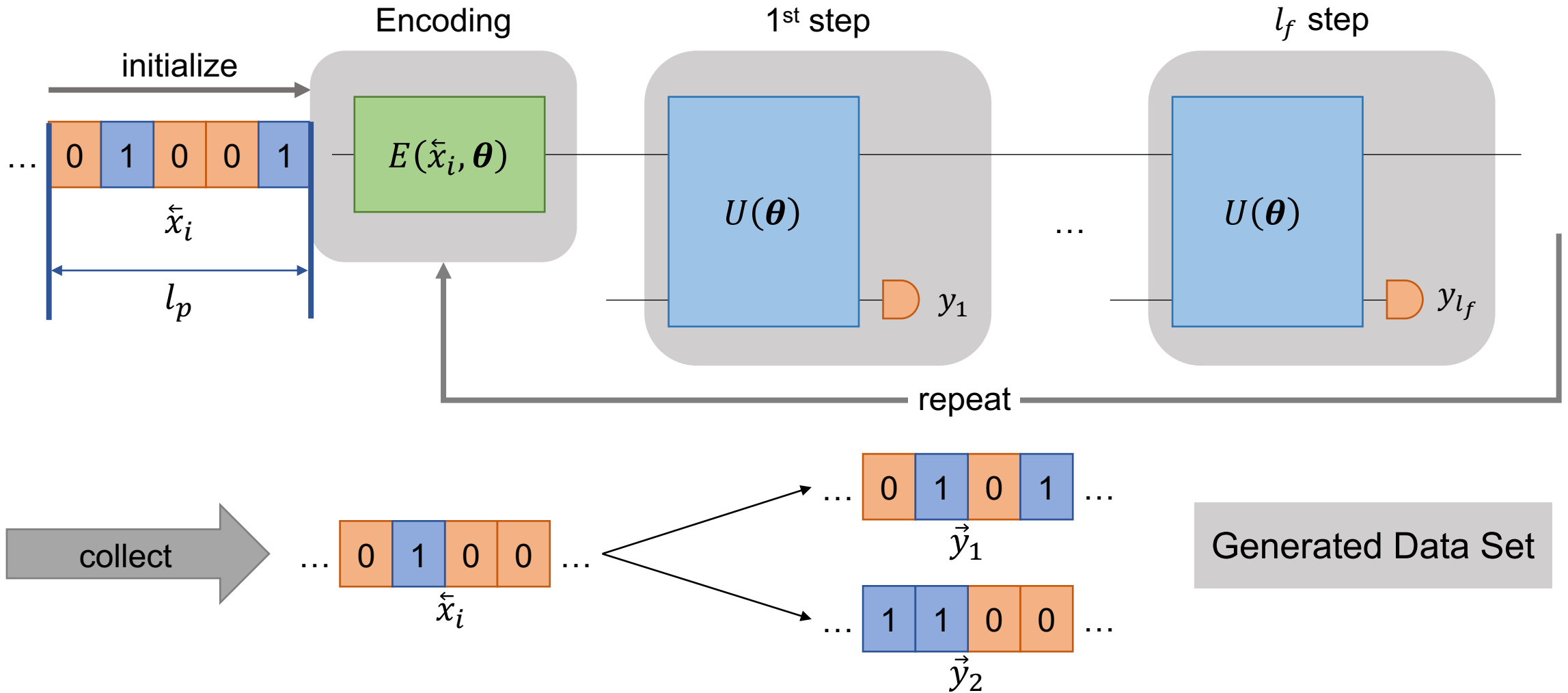


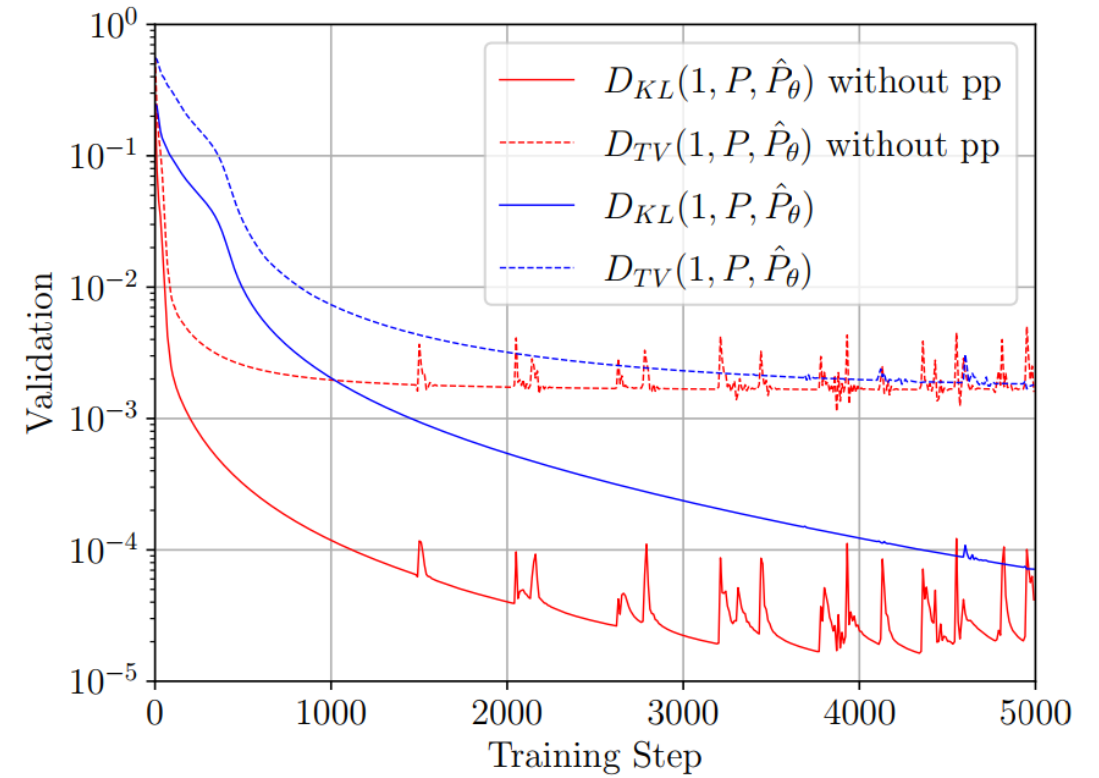
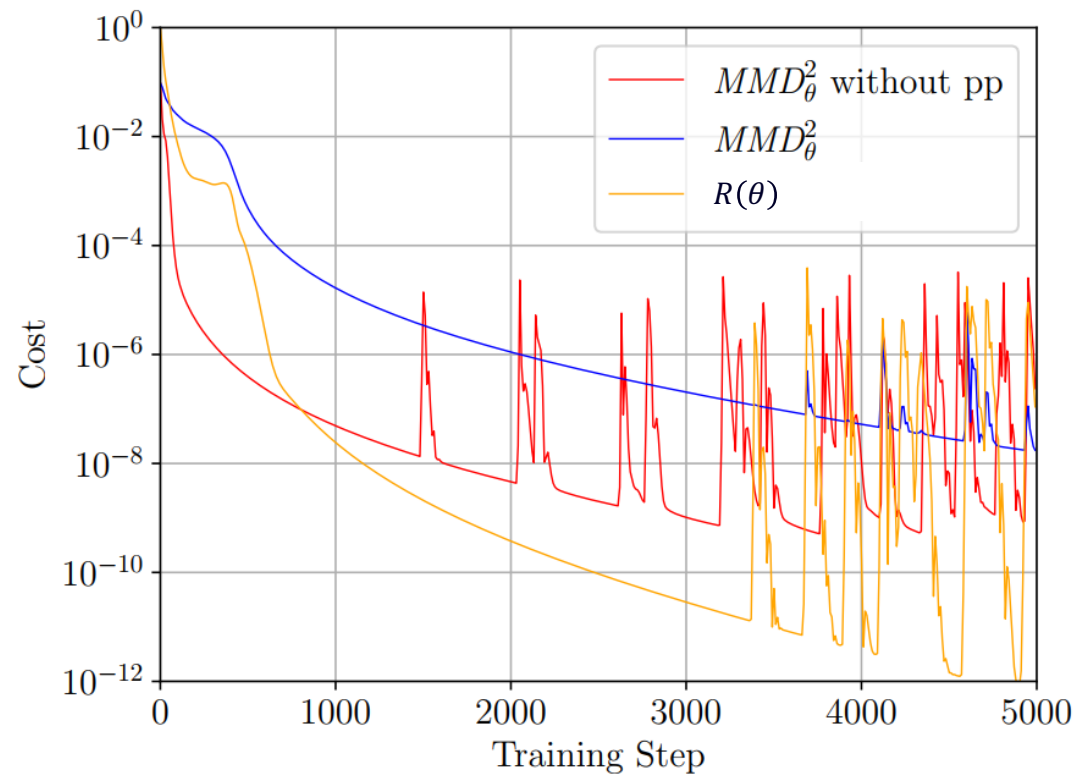
$$MMD^2[F, P, \hat{P}] = \mathbb{E}_{x \sim P, x' \sim P}[k(x, x')] - 2 \cdot \mathbb{E}_{x \sim P, y' \sim \hat{P}}[k(x, y)] + \mathbb{E}_{y \sim \hat{P}, y' \sim \hat{P}}[k(y, y')]$$

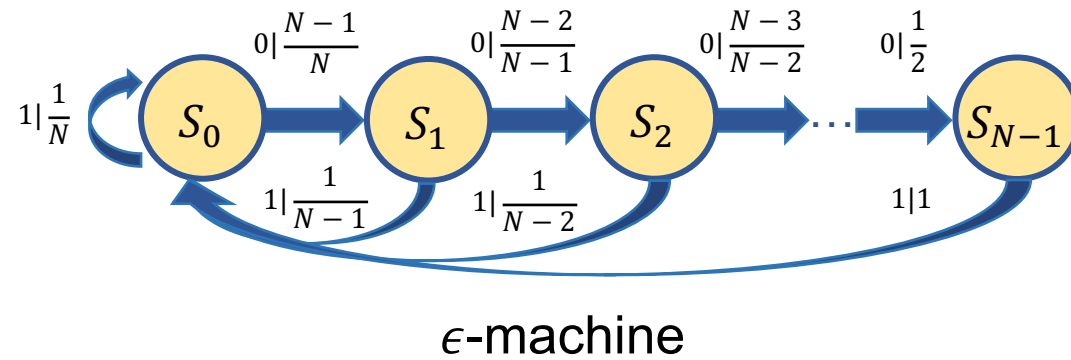
$$\frac{\partial MMD^2[P, \hat{P}_{\theta}]}{\partial \theta_i} = \sum_{j=1}^m \alpha'_{i,j}(\theta_i) \left[\mathbb{E}_{\substack{x \sim \hat{P}_{\theta_{i,j}^+} \\ y \sim \hat{P}_{\theta}}} [k(x, y)] - \mathbb{E}_{\substack{x \sim \hat{P}_{\theta_{i,j}^-} \\ y \sim \hat{P}_{\theta}}} [k(x, y)] - \mathbb{E}_{\substack{x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

Matrix depending on the ansatz



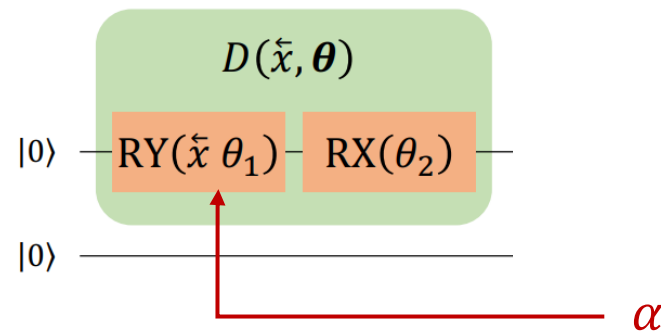







$$f(\boldsymbol{\theta}) = \langle 0 | U(\boldsymbol{\theta})^\dagger \hat{O} U(\boldsymbol{\theta}) | 0 \rangle$$

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





Thank you
very much!

Questions
are highly
welcome.