LATTICE 2024
LIVERPOOL

UNIVERSITÄT BONN
RHEINISCHE
FRIEDRICH-WILHELMS-
UNIVERSITÄT BONN

TRA MATTER
TRANSDISCIPLINARY
RESEARCH AREA

Machine Learning Enhanced Optimization of Variational Quantum Eigensolvers

Kim A. Nicoli
University of Bonn, TRA Matter, HISKP (Helmholtz Institute for Radiation and Nuclear Physics)

Talk based on: K.A. Nicoli, et al, NeurIPS '23

HISKP

Lattice Field Theories on Quantum Computers

PROCEEDINGS OF SCIENCE

Review on Quantum Computing for Lattice Field Theory

Lena Funcke,[a,b,*] Tobias Hartung,[c] Karl Jansen[d] and Stefan Kühn[d,e]

[1] Funcke L. et al., PoS (LATTICE2022)

PROCEEDINGS OF SCIENCE

Quantum simulations of lattice field thoeries

Dorota M Grabowska*

[2] Grabowska D., PoS (LATTICE2023)

Lattice Field Theories on Quantum Computers

PROCEEDINGS OF SCIENCE

Review on Quantum Computing for Lattice Field Theory

Lena Funcke,[a,b,*] **Tobias Hartung,**[c] **Karl Jansen**[d] **and Stefan Kühn**[d,e]

[1] Funcke L. et al., PoS (LATTICE2022)

PROCEEDINGS OF SCIENCE

Quantum simulations of lattice field thoeries

Dorota M Grabowska*

[2] Grabowska D., PoS (LATTICE2023)

Recent progress holds promise for successful deployment of quantum simulations of lattice field theories:

[3] Banuls et al., *Simulating lattice gauge theories within quantum technologies*, Eur. Phys. J. D (2020)

[4] Klco et al., *Standard model physics and the digital quantum revolution: thoughts about the interface*, Rep. Prog. Phys. (2020)

[5] Atas et al., *SU(2) hadrons on a quantum computer via a variational approach*, Nat. Comms. (2021)

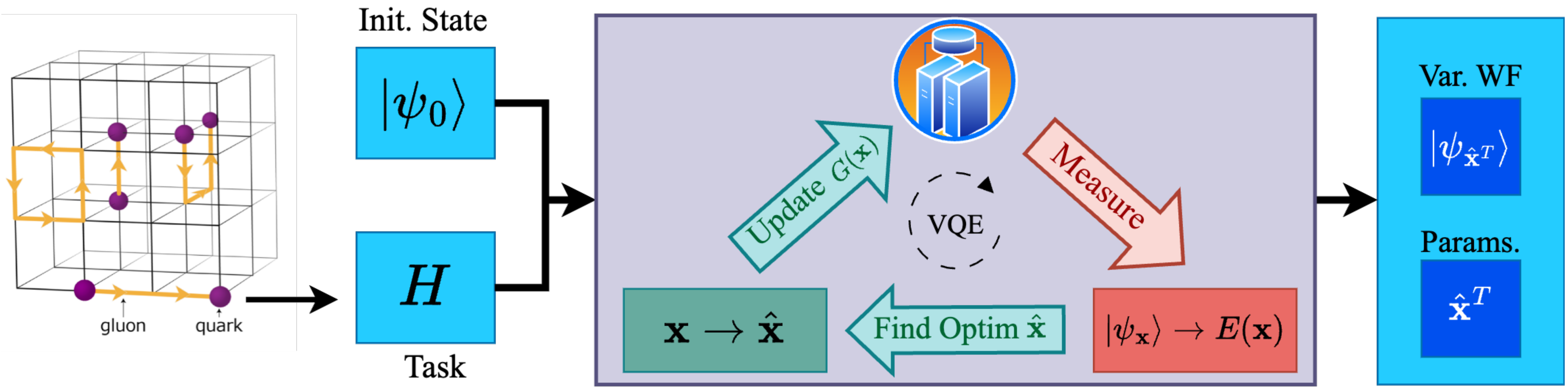
[6] Farrell et al., *Scalable Circuits for Preparing Ground States on Digital Quantum Computers: The Schwinger Model Vacuum on 100 Qubits,* arXiv:2307.03236 (2024)

[7] Crippa et al., *Towards determining the (2+1)-dimensional Quantum Electrodynamics running coupling with Monte Carlo and quantum computing methods,* arXiv: 2404.17545 (2024)

…

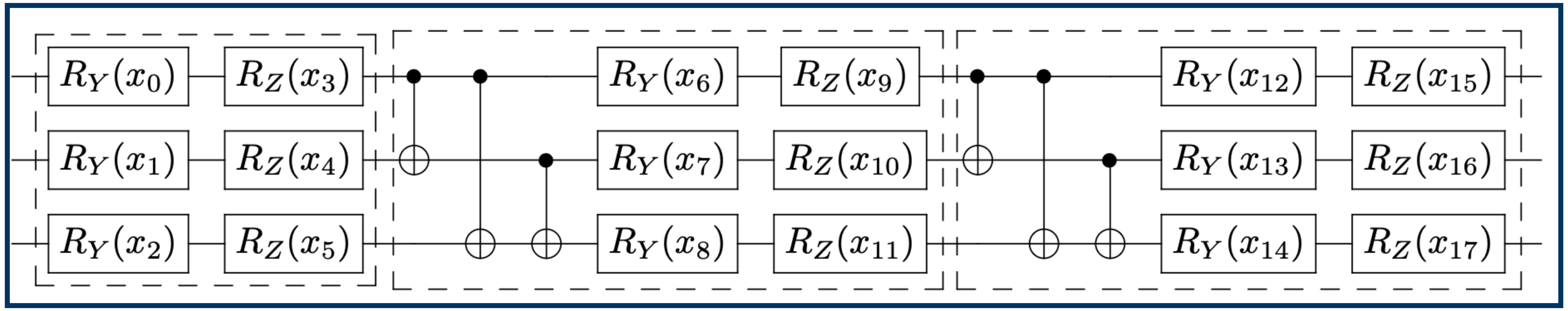
Variational Quantum Algorithms

⭐ **Goal: Find optimal parameters of a variational circuit to minimize a given cost function, being the expectation value of a given Hamiltonian H.**



Variational Quantum Eigensolvers (VQEs)

VQE: Use of a feedback loop between a classical computer and a quantum processor, where the latter is used to efficiently evaluate a cost function.

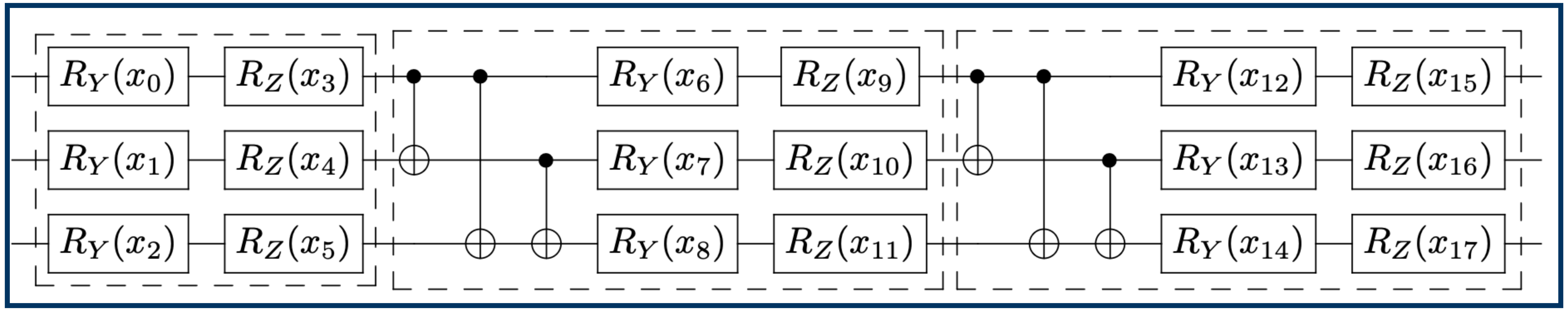
$|\psi_0\rangle$ →  → $|\psi_{\mathbf{x}}\rangle$

$$G(\mathbf{x})$$

Variational Quantum Eigensolvers (VQEs)

VQE: Use of a feedback loop between a classical computer and a quantum processor, where the latter is used to efficiently evaluate a cost function.

$|\psi_0\rangle$ →



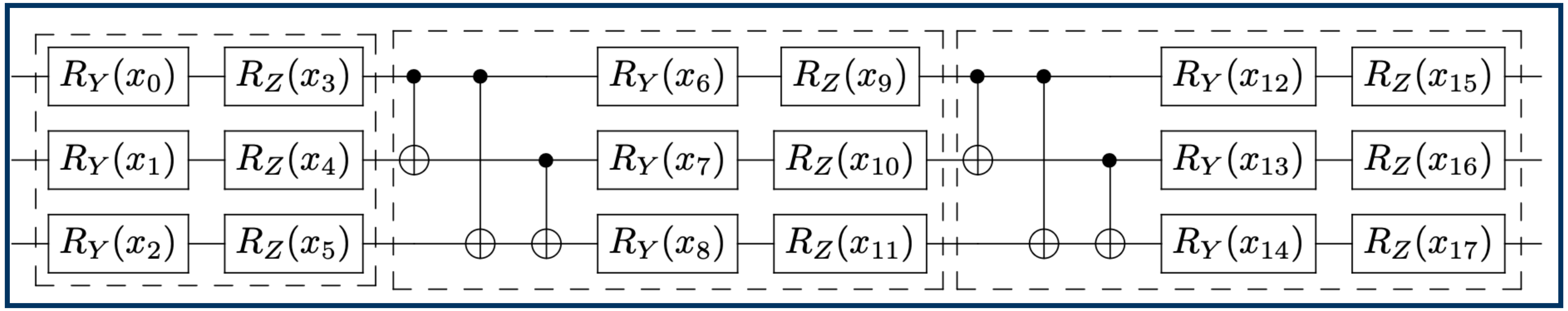
→ $|\psi_{\mathbf{x}}\rangle$

$$G(\mathbf{x})$$

(1) Initial state preparation → $|\psi_0\rangle$

Variational Quantum Eigensolvers (VQEs)

VQE: Use of a feedback loop between a classical computer and a quantum processor, where the latter is used to efficiently evaluate a cost function.

$|\psi_0\rangle$ →  → $|\psi_{\mathbf{x}}\rangle$

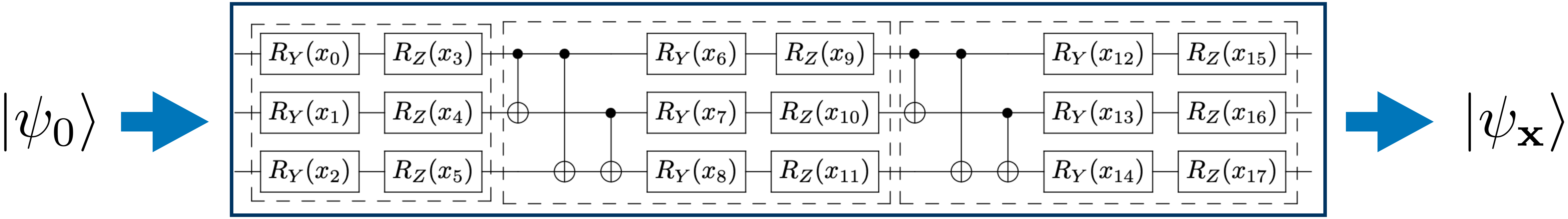
$$G(\mathbf{x})$$

(1) Initial state preparation → $|\psi_0\rangle$

(2) Quantum state transformation → $|\psi_{\mathbf{x}}\rangle = G(\mathbf{x})|\psi_0\rangle$

Variational Quantum Eigensolvers (VQEs)

VQE: Use of a feedback loop between a classical computer and a quantum processor, where the latter is used to efficiently evaluate a cost function.



$$G(\mathbf{x})$$

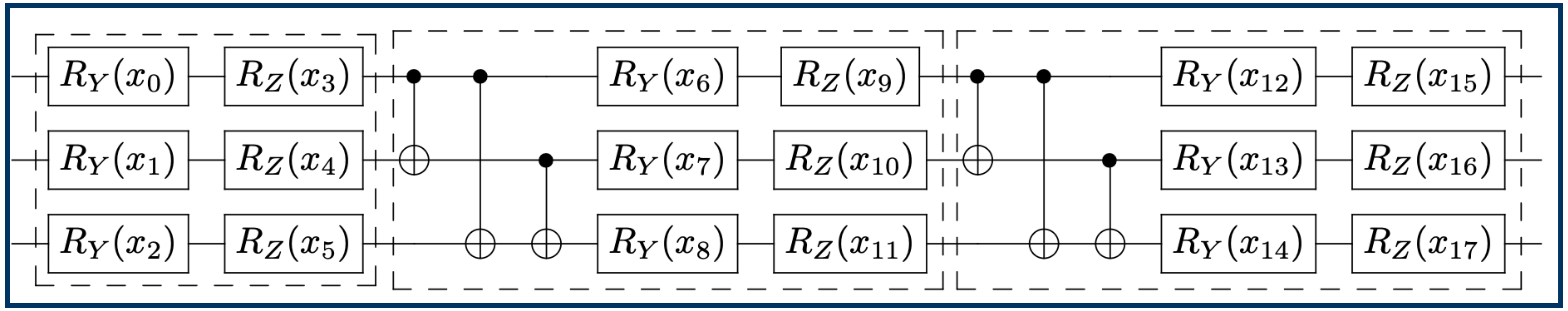
(1) Initial state preparation → $|\psi_0\rangle$

(2) Quantum state transformation → $|\psi_{\mathbf{x}}\rangle = G(\mathbf{x})|\psi_0\rangle$

(3) Measure final energy → $E(\mathbf{x}) = \langle\psi_{\mathbf{x}}|H|\psi_{\mathbf{x}}\rangle = \langle\psi_0|G(\mathbf{x})^\dagger H G(\mathbf{x})|\psi_0\rangle$

Variational Quantum Eigensolvers (VQEs)

VQE: Use of a feedback loop between a classical computer and a quantum processor, where the latter is used to efficiently evaluate a cost function.

$|\psi_0\rangle$ →  → $|\psi_{\mathbf{x}}\rangle$

$$G(\mathbf{x})$$

(1) Initial state preparation → $|\psi_0\rangle$

(2) Quantum state transformation → $|\psi_{\mathbf{x}}\rangle = G(\mathbf{x})|\psi_0\rangle$

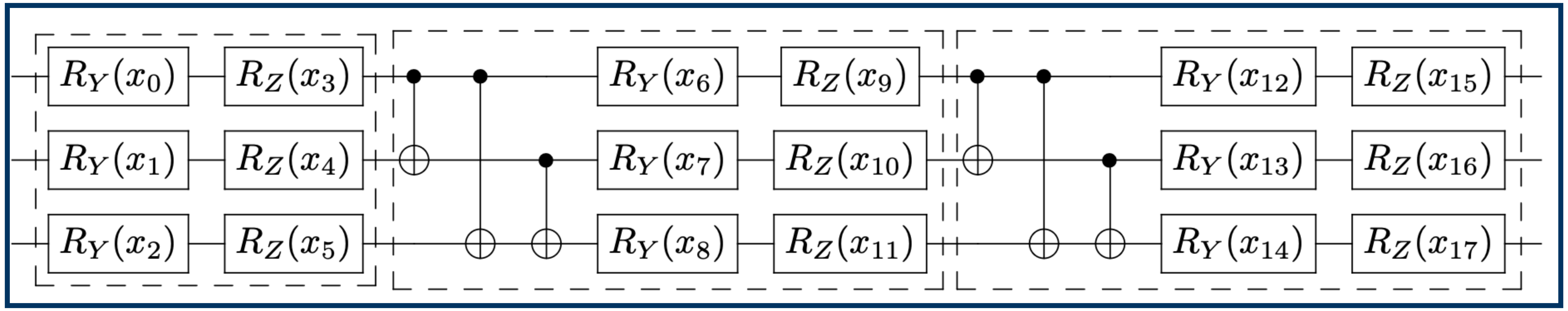
(3) Measure final energy → $E(\mathbf{x}) = \langle\psi_{\mathbf{x}}|H|\psi_{\mathbf{x}}\rangle = \langle\psi_0|G(\mathbf{x})^\dagger H G(\mathbf{x})|\psi_0\rangle$

(4) Find $\mathbf{x}$ that minimizes E → $\underset{\mathbf{x}}{\operatorname{argmin}}\ E(\mathbf{x})$

Variational Minimization Problem

Variational Quantum Eigensolvers (VQEs)

VQE: Use of a feedback loop between a classical computer and a quantum processor, where the latter is used to efficiently evaluate a cost function.

$|\psi_0\rangle$ →  → $|\psi_{\mathbf{x}}\rangle$

$G(\mathbf{x})$ **Quantum operation**: measurement on QC

(1) Initial state preparation → $|\psi_0\rangle$

(2) Quantum state transformation → $|\psi_{\mathbf{x}}\rangle = G(\mathbf{x})|\psi_0\rangle$

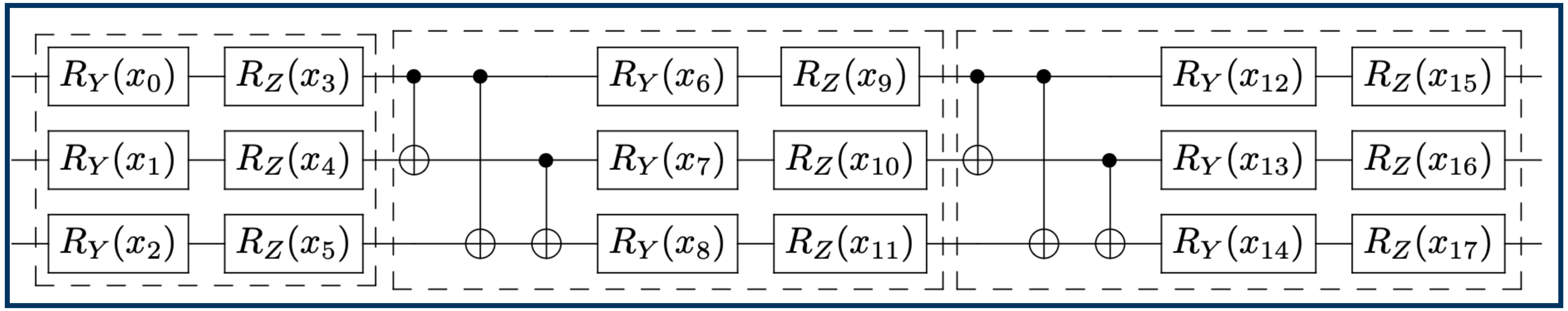
(3) Measure final energy → $E(\mathbf{x}) = \langle\psi_{\mathbf{x}}|H|\psi_{\mathbf{x}}\rangle = \langle\psi_0|G(\mathbf{x})^\dagger H G(\mathbf{x})|\psi_0\rangle$

(4) Find $\mathbf{x}$ that minimizes E → $\underset{\mathbf{x}}{\operatorname{argmin}}\ E(\mathbf{x})$

Variational Minimization Problem

Variational Quantum Eigensolvers (VQEs)

<u>VQE</u>: Use of a feedback loop between a classical computer and a quantum processor, where the latter is used to efficiently evaluate a cost function.

$|\psi_0\rangle$ → 

→ $|\psi_{\mathbf{x}}\rangle$

$G(\mathbf{x})$ **Quantum operation**: measurement on QC

(1) Initial state preparation → $|\psi_0\rangle$

(2) Quantum state transformation → $|\psi_{\mathbf{x}}\rangle = G(\mathbf{x})|\psi_0\rangle$

(3) Measure final energy → $E(\mathbf{x}) = \langle\psi_{\mathbf{x}}|H|\psi_{\mathbf{x}}\rangle = \langle\psi_0|G(\mathbf{x})^\dagger H G(\mathbf{x})|\psi_0\rangle$

(4) Find $\mathbf{x}$ that minimizes E → $\underset{\mathbf{x}}{\operatorname{argmin}}\ E(\mathbf{x})$

Classical operation: runs on classical computer

Variational Minimization Problem

Previous work: NFT Algorithm

Nakanishi et al., Phys. Rev. Res 2, 043158 (2020)

Nakanishi et al., (2020) show that the VQE objective $E(\,\cdot\,)$ obeys

$$\exists \boldsymbol{b} \in \mathbb{R}^{3^D} \quad \text{s.t.} \qquad E(\boldsymbol{x}) = \boldsymbol{b}^\top \cdot \mathrm{vec}(\otimes_{d=1}^{D} (1, \cos x_d,\, \sin x_d)^\top), \qquad \forall \boldsymbol{x} \in [0, 2\pi)^D$$

i.e., **for unitary gates**, the energy function is a tensor product of sin and cos.

Previous work: NFT Algorithm

Nakanishi et al., Phys. Rev. Res 2, 043158 (2020)

Nakanishi et al., (2020) show that the VQE objective $E(\cdot)$ obeys

$$\exists \boldsymbol{b} \in \mathbb{R}^{3^D} \quad \text{s.t.} \qquad E(\boldsymbol{x}) = \boldsymbol{b}^\top \cdot \mathrm{vec}(\otimes_{d=1}^{D} (1, \cos x_d, \sin x_d)^\top), \qquad \forall \boldsymbol{x} \in [0, 2\pi)^D$$

i.e., **for unitary gates**, the energy function is a tensor product of sin and cos.

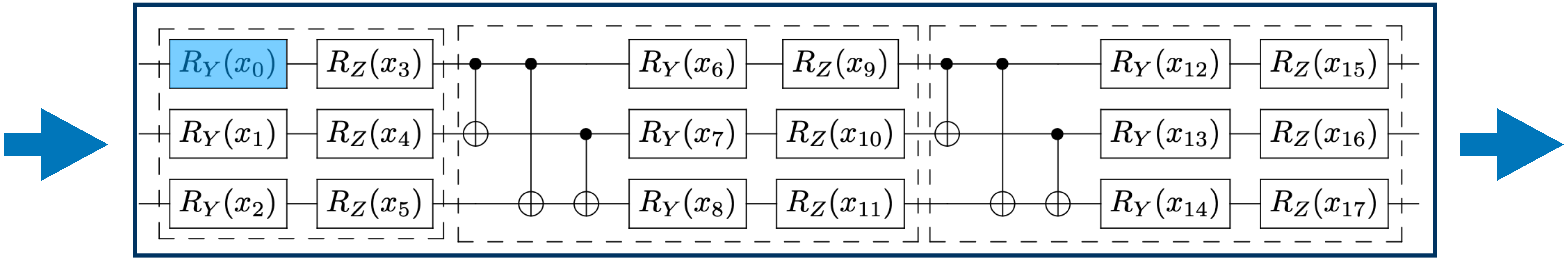
↓

Optimization of circuit parameters, i.e., sequentially (randomly) choose one parameter and optimize on 1-D submanifolds, keeping the other parameters fixed.

Previous work: Sequential Minimal Optimization

Nakanishi et al., Phys. Rev. Res 2, 043158 (2020)

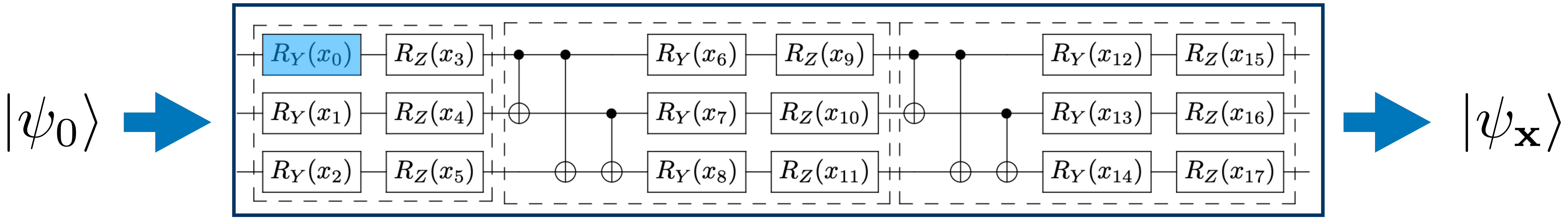
$|\psi_0\rangle$

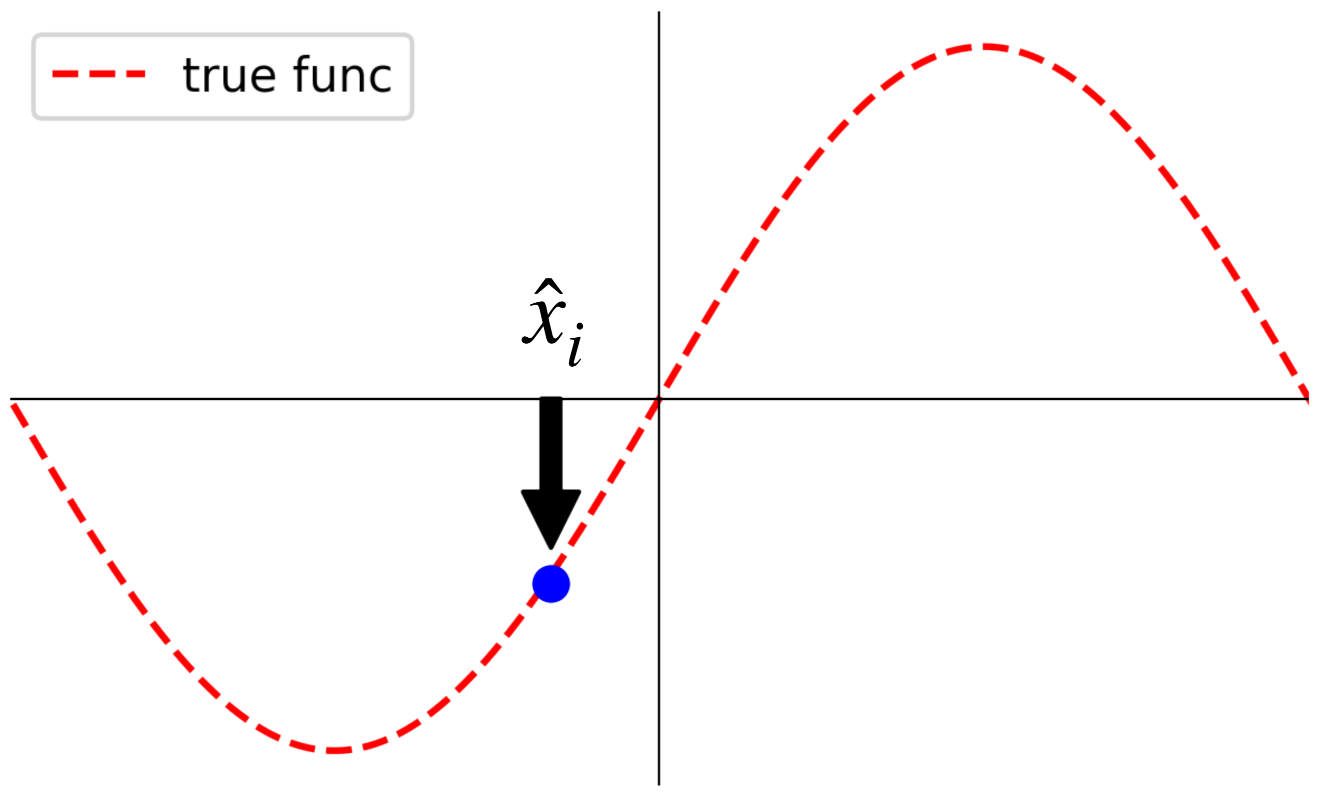


$|\psi_{\mathbf{x}}\rangle$

Previous work: Sequential Minimal Optimization

Nakanishi et al., Phys. Rev. Res 2, 043158 (2020)

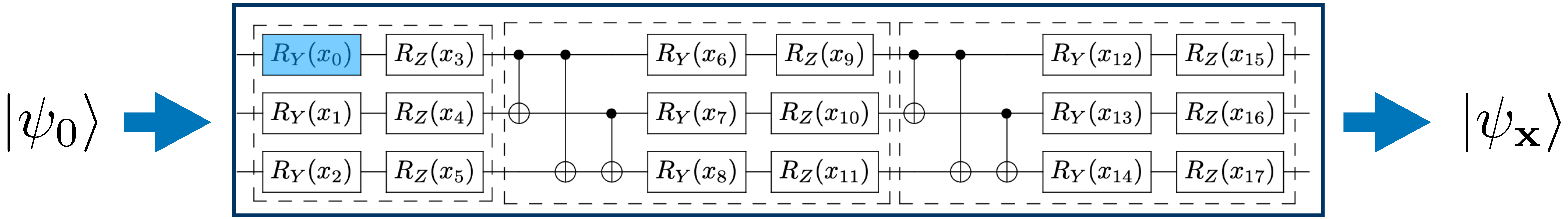


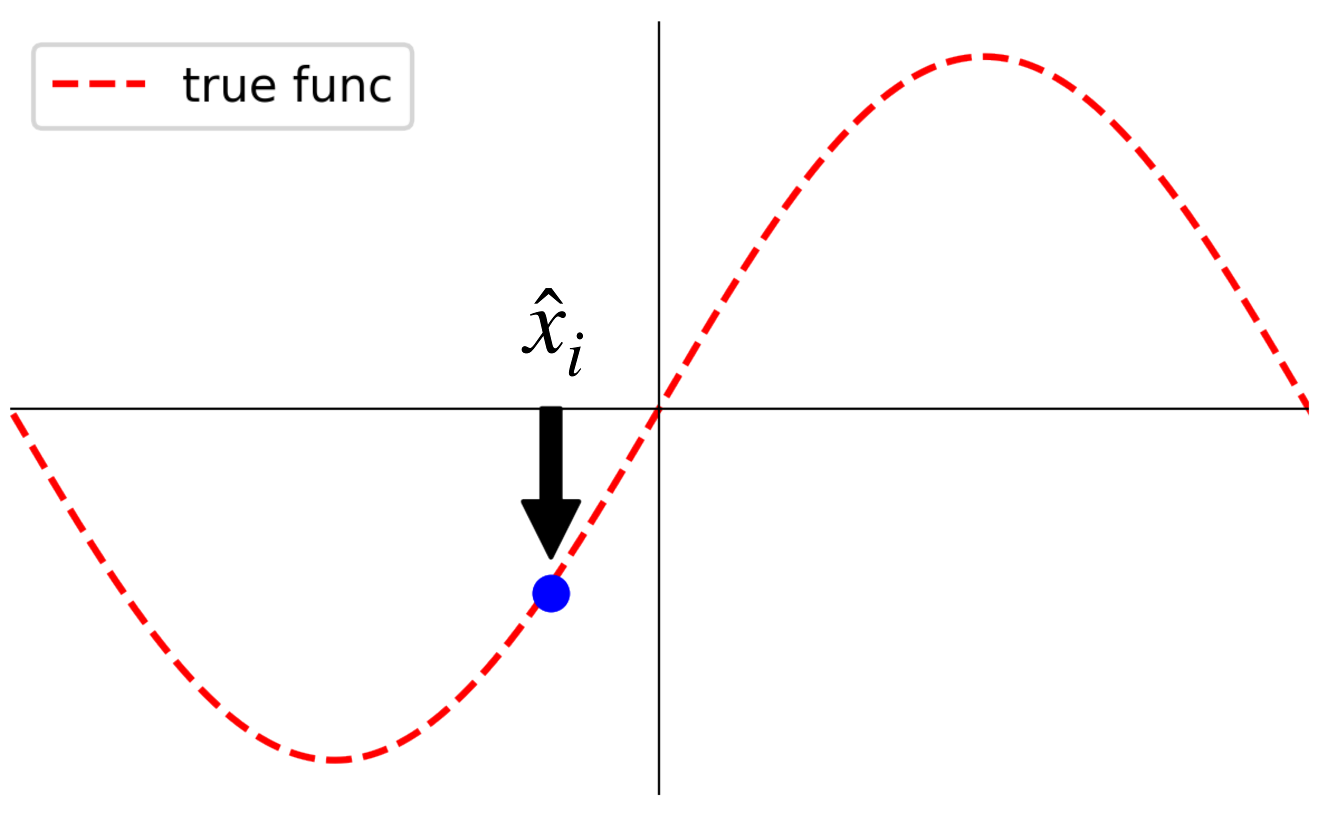


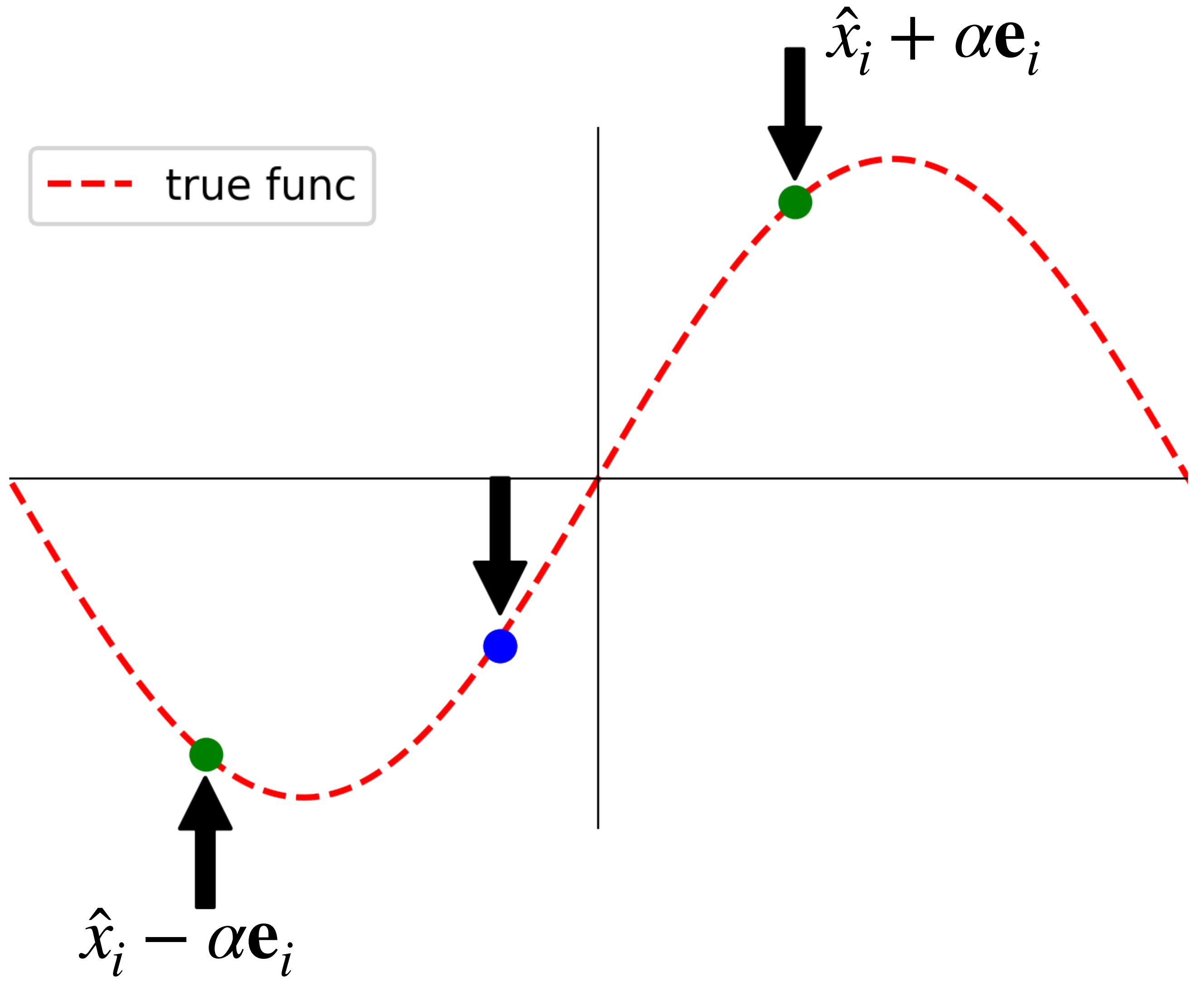
Start from the current best point
on **subspace** identified by i

Previous work: Sequential Minimal Optimization

Nakanishi et al., Phys. Rev. Res 2, 043158 (2020)





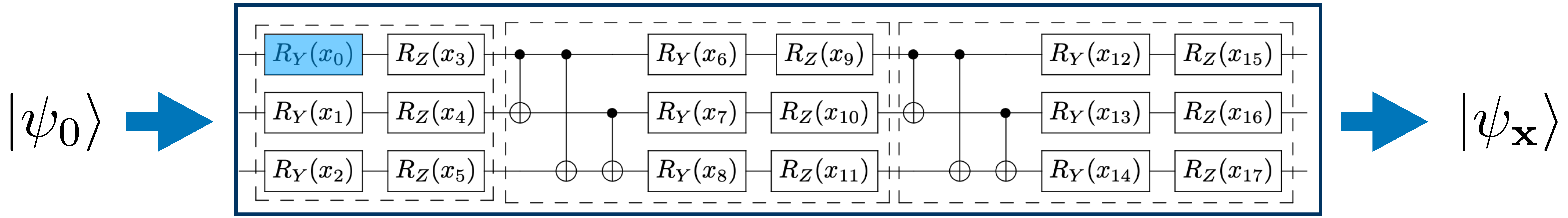


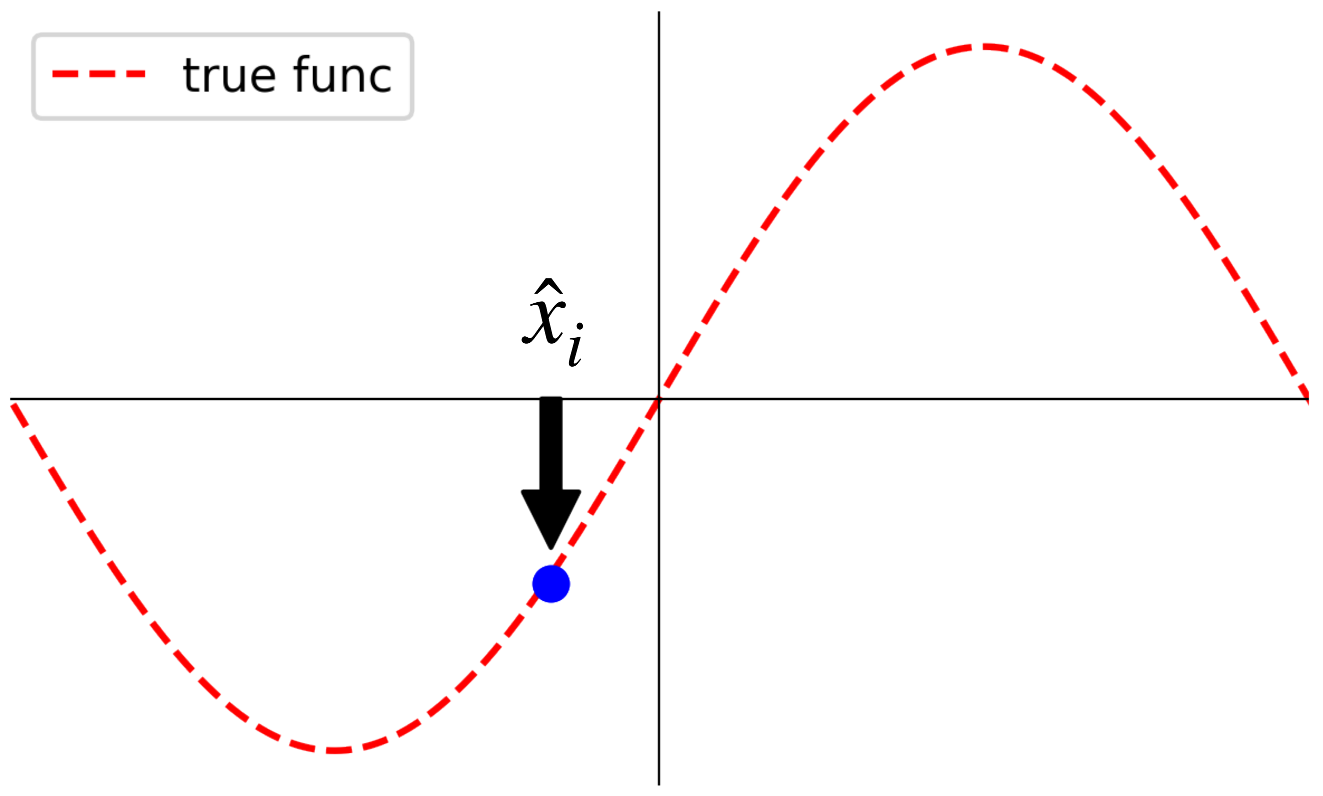
Start from the current best point on **subspace** identified by i

Measure new points on the quantum computer at $\hat{x}_i \pm \alpha \mathbf{e}_i$

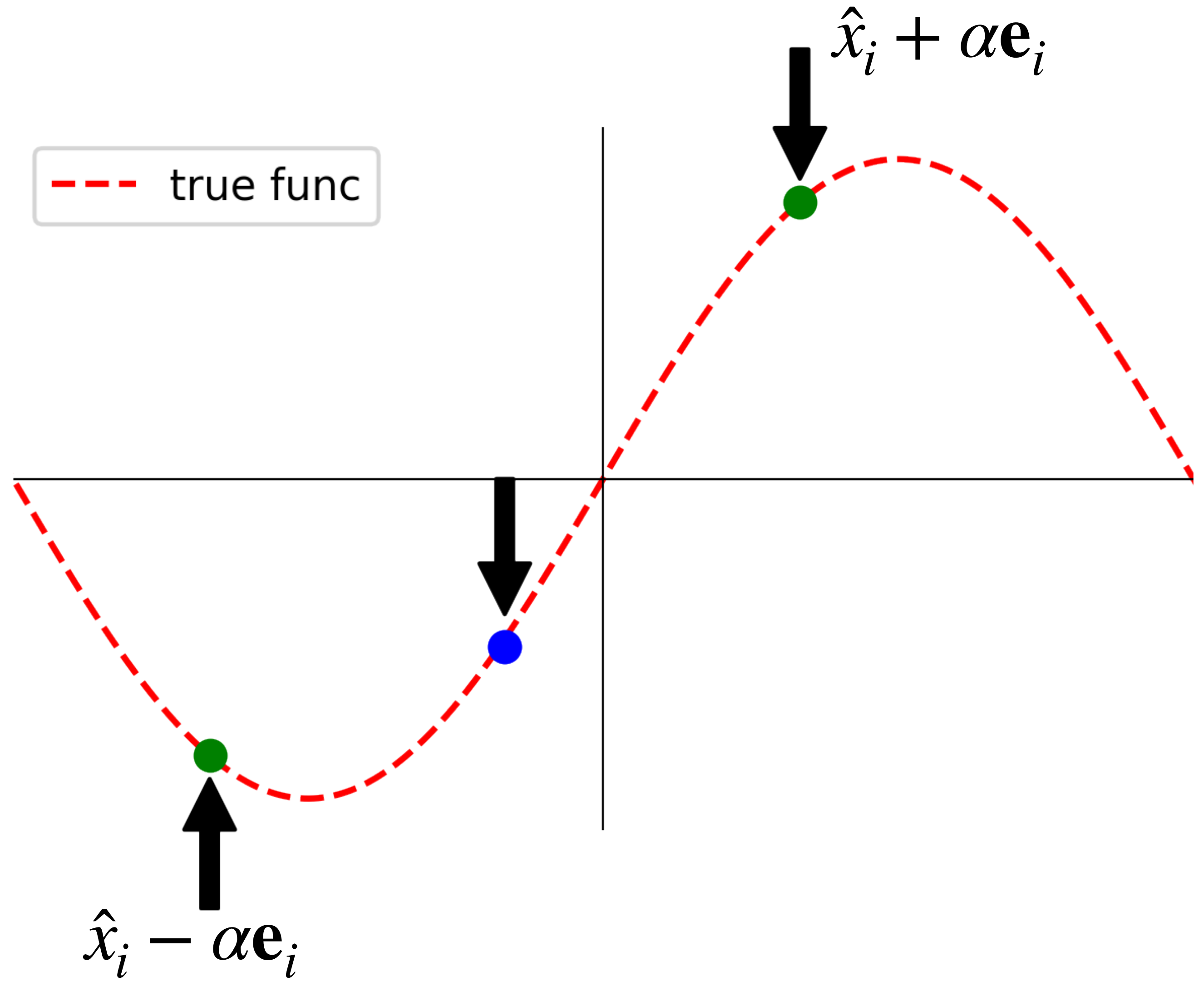
Previous work: Sequential Minimal Optimization

Nakanishi et al., Phys. Rev. Res 2, 043158 (2020)

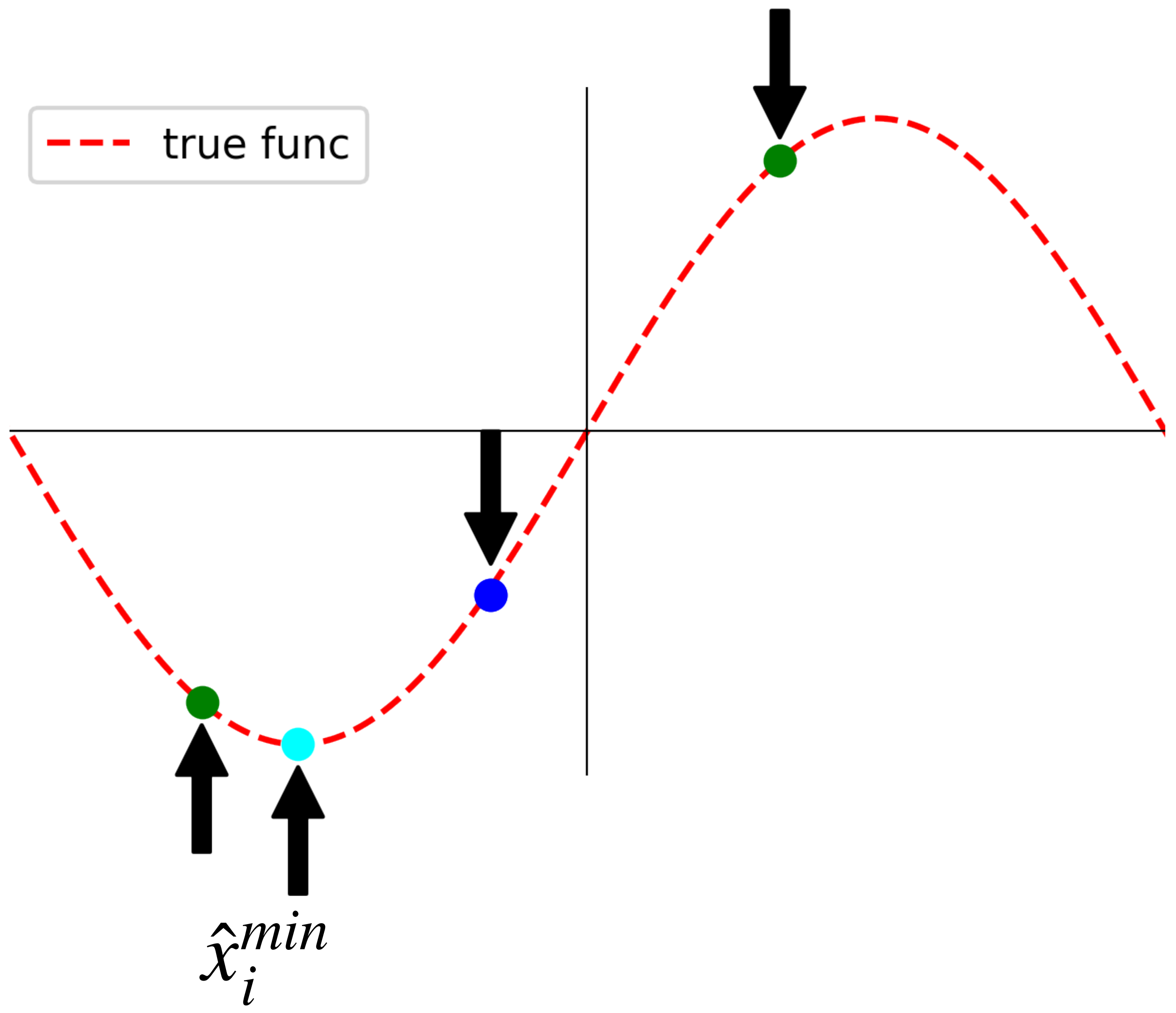




Start from the current best point on **subspace** identified by i



Measure new points on the quantum computer at $\hat{x}_i \pm \alpha \mathbf{e}_i$



Do **least square minimization** and find the new best point on the line s.t. $\hat{x}_i^{min} \equiv \hat{x}_{i+1}$

Previous work: Sequential Minimal Optimization

Problems 😕

$\boldsymbol{\alpha} \rightarrow$ Fixed to $\dfrac{2\pi}{3}$

Measurement Noise
Hardware Noise

Previous work: Sequential Minimal Optimization

Problems 😕

$\boldsymbol{\alpha} \rightarrow$ Fixed to $\dfrac{2\pi}{3}$ →

Measurement Noise
Hardware Noise

Questions 🤔

Learn optimal $\boldsymbol{\alpha}$ from previous measurements?

Previous work: Sequential Minimal Optimization

Problems 😕

$\boldsymbol{\alpha} \rightarrow$ Fixed to $\frac{2\pi}{3}$ → Learn optimal $\boldsymbol{\alpha}$ from previous measurements?

Measurement Noise
Hardware Noise → Deal with noisy measurements?

Questions 🤔

Previous work: Sequential Minimal Optimization

Problems 😕

Questions 🤔

$\boldsymbol{\alpha} \rightarrow$ Fixed to $\frac{2\pi}{3}$ → Learn optimal $\boldsymbol{\alpha}$ from previous measurements?

Measurement Noise
Hardware Noise → Deal with noisy measurements?

YES!! 🥳

Physics Informed Bayesian Optimization

We tackle the classical optimization problem from a Bayesian Optimization standpoint.

↓

Given a set of (**costly**) measurements and a **surrogate** model, BO helps to identify at which points are worth measuring next.

Physics Informed Bayesian Optimization

We tackle the classical optimization problem from a Bayesian Optimization standpoint.

Given a set of (**costly**) measurements and a **surrogate** model, BO helps to identify at which points are worth measuring next.

Our research question:

Which point should we ***measure next****, on the quantum computer, to* ***maximize the information gain*** *and* ***minimize the quantum computer calls*** *needed to minimize the objective?*

Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Gaussian Process Regression (GPR) uses a GP surrogate model

$$p(E(\cdot)|\boldsymbol{X}, \boldsymbol{y}) = \mathrm{GP}(E(\cdot); \mu_{\boldsymbol{X}}(\cdot), s_{\boldsymbol{X}}(\cdot, \cdot))$$

to infer a **target function** $E(\cdot)$ from a set of observations $\{\boldsymbol{X}, \boldsymbol{y}\}$

Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Gaussian Process Regression (GPR) uses a GP surrogate model

$$p(E(\cdot)|\boldsymbol{X}, \boldsymbol{y}) = \mathrm{GP}(E(\cdot); \mu_{\boldsymbol{X}}(\cdot), s_{\boldsymbol{X}}(\cdot, \cdot))$$

to infer a **target function** $E(\cdot)$ from a set of observations $\{\boldsymbol{X}, \boldsymbol{y}\}$

The **mean** $\mu_{\mathbf{X}}(\cdot)$ and **covariance** $s_{\mathbf{X}}(\cdot)$ of the **GPR** depend on a **kernel function** $k(\cdot, \cdot)$

Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Gaussian Process Regression (GPR) uses a GP surrogate model

$$p(E(\cdot)|\boldsymbol{X}, \boldsymbol{y}) = \mathrm{GP}(E(\cdot); \mu_{\boldsymbol{X}}(\cdot), s_{\boldsymbol{X}}(\cdot, \cdot))$$

to infer a **target function** $E(\cdot)$ from a set of observations $\{\boldsymbol{X}, \boldsymbol{y}\}$

The **mean** $\mu_{\mathbf{X}}(\cdot)$ and **covariance** $s_{\mathbf{X}}(\cdot)$ of the **GPR** depend on a **kernel function** $k(\cdot, \cdot)$

Choosing the right kernel function is crucial in order to leverage the learning capabilities of the GP and of GP Regression

The VQE Kernel

Nakanishi et al., (2020) show that the VQE objective $E(\,\cdot\,)$ obeys

$$\exists \boldsymbol{b} \in \mathbb{R}^{3^D} \quad \text{s.t.} \qquad E(\boldsymbol{x}) = \boldsymbol{b}^\top \cdot \operatorname{vec}(\otimes_{d=1}^{D} (1, \cos x_d,\, \sin x_d)^\top), \qquad \forall \boldsymbol{x} \in [0, 2\pi)^D$$

We thus derive a **covariance function** $k(\,\cdot\,,\cdot\,)$ fulfilling the same functional

$$k^{\mathrm{VQE}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \prod_{d=1}^{D} \left(\frac{\gamma^2 + 2\cos\left(x_d - x'_d\right)}{\gamma^2 + 2} \right)$$

See Nicoli et al., (2023) for detailed proofs.

EMICoRe

Expected Maximum Improvement over Confident Regions

➡ Special **acquisition function** using the VQE kernel and the concept of **confident regions**

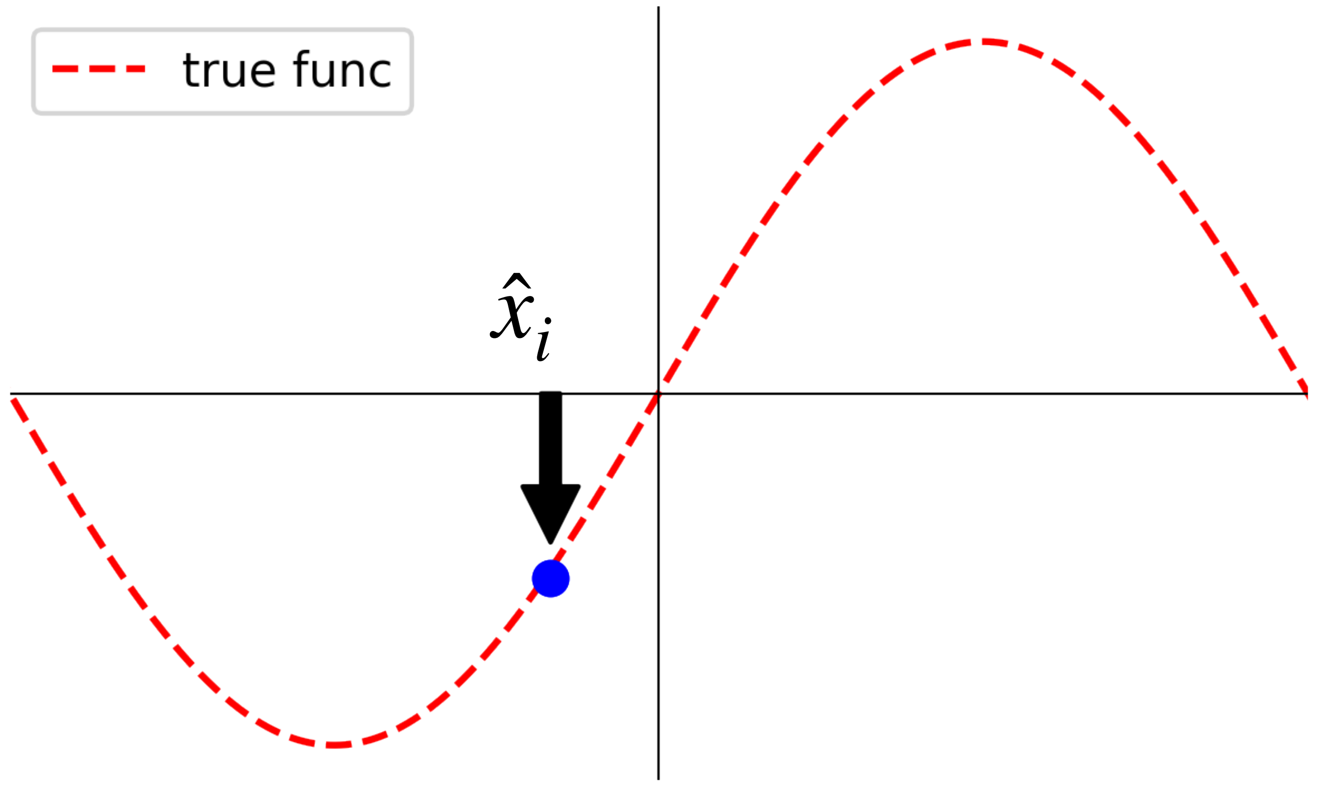
➡ Use EMICoRe to perform a grid search and find the **best pair of shifts** $\{\hat{\alpha}_1^t, \hat{\alpha}_2^t\}_{d^t}$.

EMICoRe

Expected Maximum Improvement over Confident Regions

➡ Special **acquisition function** using the VQE kernel and the concept of **confident regions**

➡ Use EMICoRe to perform a grid search and find the **best pair of shifts** $\{\hat{\alpha}_1^t, \hat{\alpha}_2^t\}_{d^t}$.



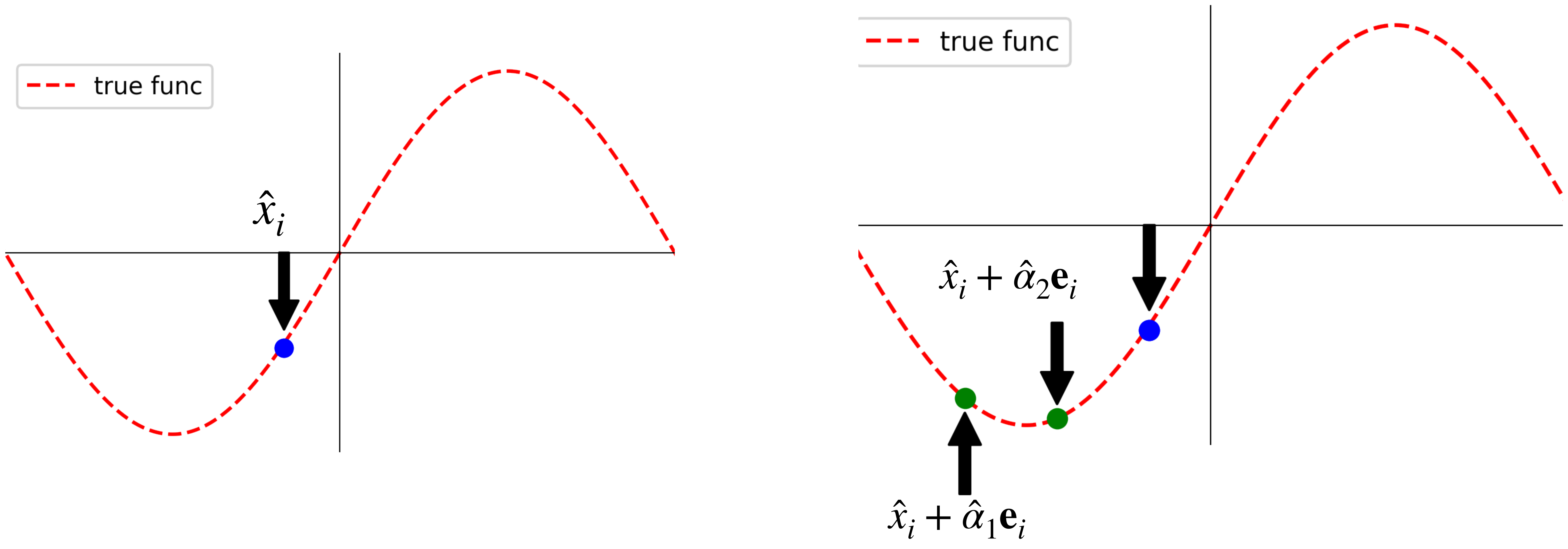
Start from the current best point
on **subspace** identified by i

EMICoRe

Expected Maximum Improvement over Confident Regions

➡ Special **acquisition function** using the VQE kernel and the concept of **confident regions**

➡ Use EMICoRe to perform a grid search and find the **best pair of shifts** $\{\hat{\alpha}_1^t, \hat{\alpha}_2^t\}_{d^t}$.



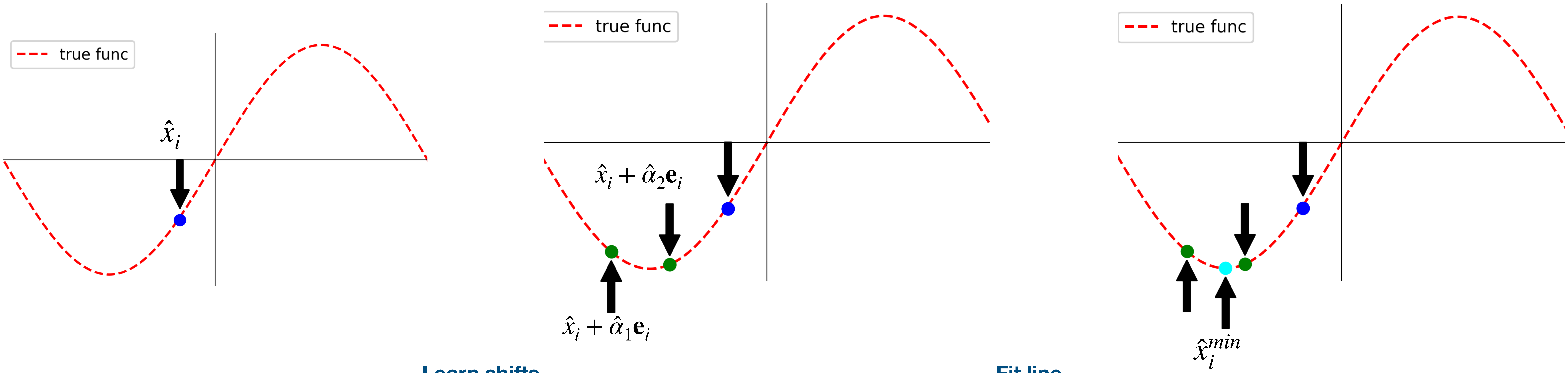
Start from the current best point on **subspace** identified by i → **Learn shifts** → **Measure** shifted points (green) on the quantum computer

EMICoRe

Expected Maximum Improvement over Confident Regions

➡ Special **acquisition function** using the VQE kernel and the concept of **confident regions**

➡ Use EMICoRe to perform a grid search and find the **best pair of shifts** $\{\hat{\alpha}_1^t, \hat{\alpha}_2^t\}_{d^t}$.



Start from the current best point on **subspace** identified by i

Learn shifts →

Measure shifted points (green) on the quantum computer

Fit line →

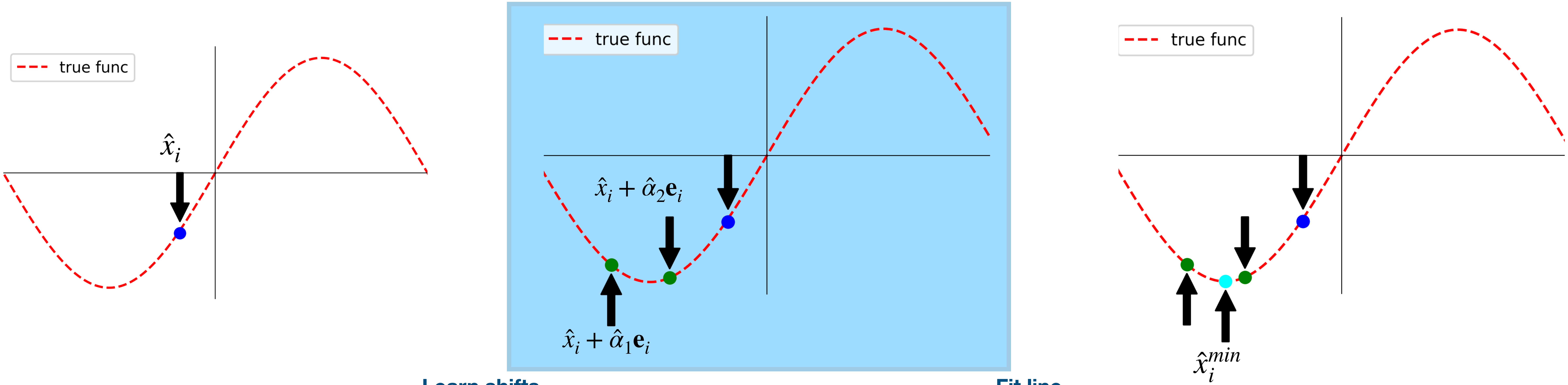
Least square minimization and find the new min on the line s.t. $\hat{x}_i^{min} \equiv \hat{x}_{i+1}$

EMICoRe

Expected Maximum Improvement over Confident Regions

- ➡ Special **acquisition function** using the VQE kernel and the concept of **confident regions**
- ➡ Use EMICoRe to perform a grid search and find the **best pair of shifts** $\{\hat{\alpha}_1^t, \hat{\alpha}_2^t\}_{d^t}$.

In contrast to NFT shifts are not equidistant but learned using BO and GP



Start from the current best point on **subspace** identified by i

Learn shifts →

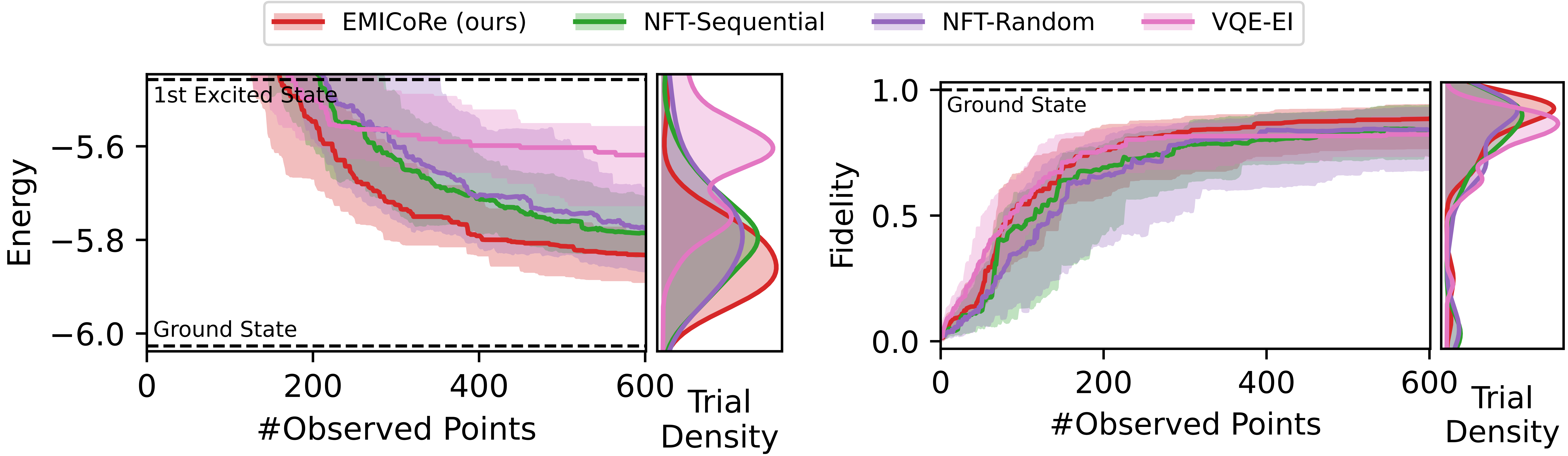
Measure shifted points (green) on the quantum computer

Fit line →

Least square minimization and find the new min on the line s.t. $\hat{x}_i^{min} \equiv \hat{x}_{i+1}$

Results: ✅ Shot Noise, ❌ Hardware Noise

$$H = -\left[\sum_{j=1}^{Q-1}(J_X\sigma_j^X\sigma_{j+1}^X + J_Y\sigma_j^Y\sigma_{j+1}^Y + J_Z\sigma_j^Z\sigma_{j+1}^Z) + \sum_{j=1}^{Q}(h_X\sigma_j^X + h_Y\sigma_j^Y + h_Z\sigma_j^Z)\right]$$



Critical Ising $\boldsymbol{\sigma} = (0.0,\ 0.0,\ -1.0)$, $\boldsymbol{J} = (-1,\ 0.0,\ 0.0)$

Setting Qubits $= 5$, Layers $= 3$, Circuit $=$ ESU(2) with OBC

Results: ✅ Shot Noise, ✅ Hardware Noise

Critical Ising

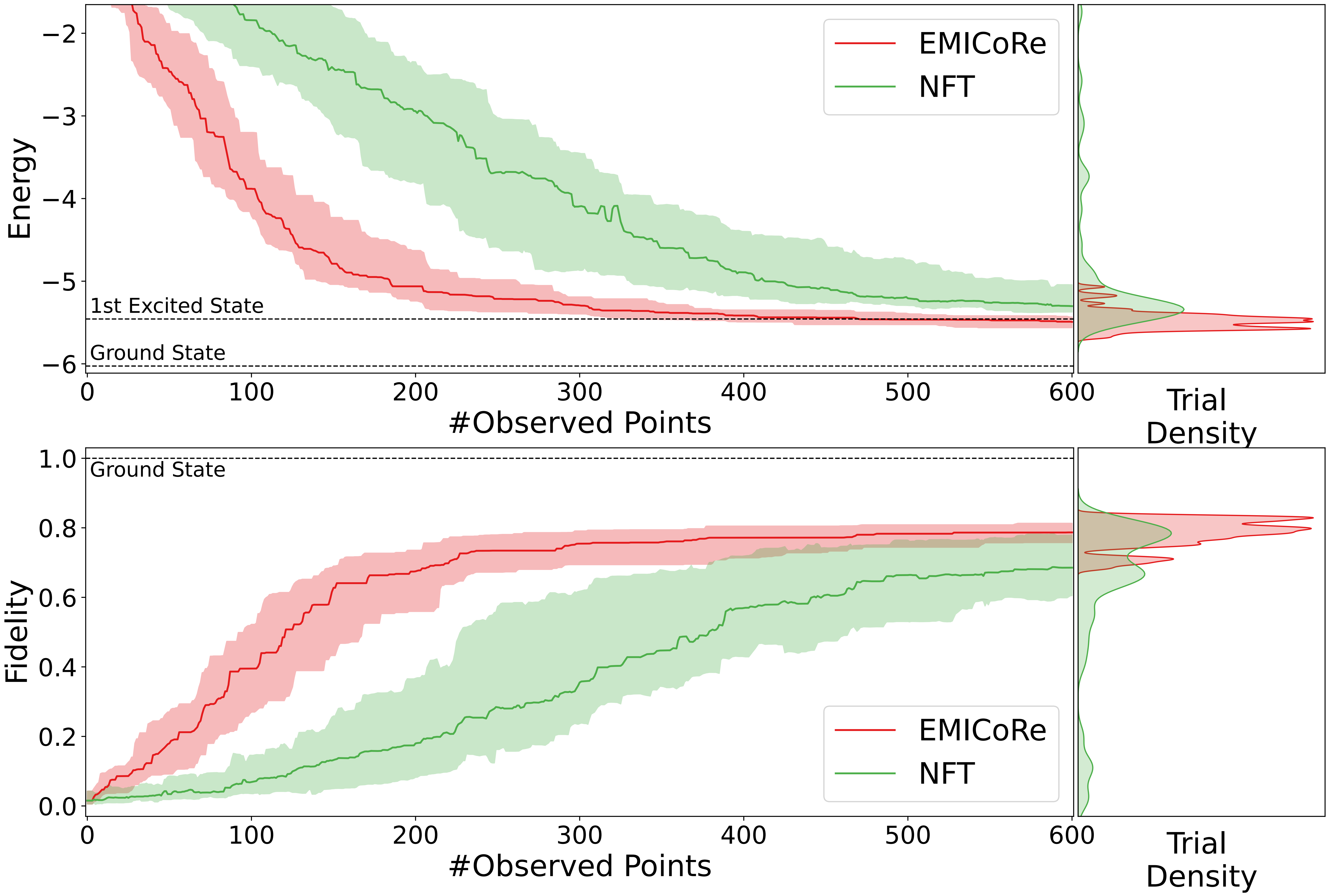
$$\boldsymbol{\sigma} = (0.0,\ 0.0,\ -1.0)$$
$$\boldsymbol{J} = (-1,\ 0.0,\ 0.0)$$

Setting

Qubits = 5
Layers = 3
Circuit = ESU(2) with OBC

Noise Type

- Simulated Hardware Noise
- No Error Mitigation



⚠️ For details on error mitigation see the poster by **Luca Wagner**

Summary and outlook

Summary:

- Proposed a physics-informed **VQE-kernel** fulfilling VQEs' functional form.
- Proposed novel acquisition function **EMICoRe**.
- EMICoRe combined with the VQE-kernel can
  - ⋆ **Outperform** baselines on standard benchmarks.
  - ⋆ **Approximate** the target function as more points are measured.

Summary and outlook

Summary:

- Proposed a physics-informed **VQE-kernel** fulfilling VQEs' functional form.
- Proposed novel acquisition function **EMICoRe**.
- EMICoRe combined with the VQE-kernel can
  - ⋆ **Outperform** baselines on standard benchmarks.
  - ⋆ **Approximate** the target function as more points are measured.

Outlook:

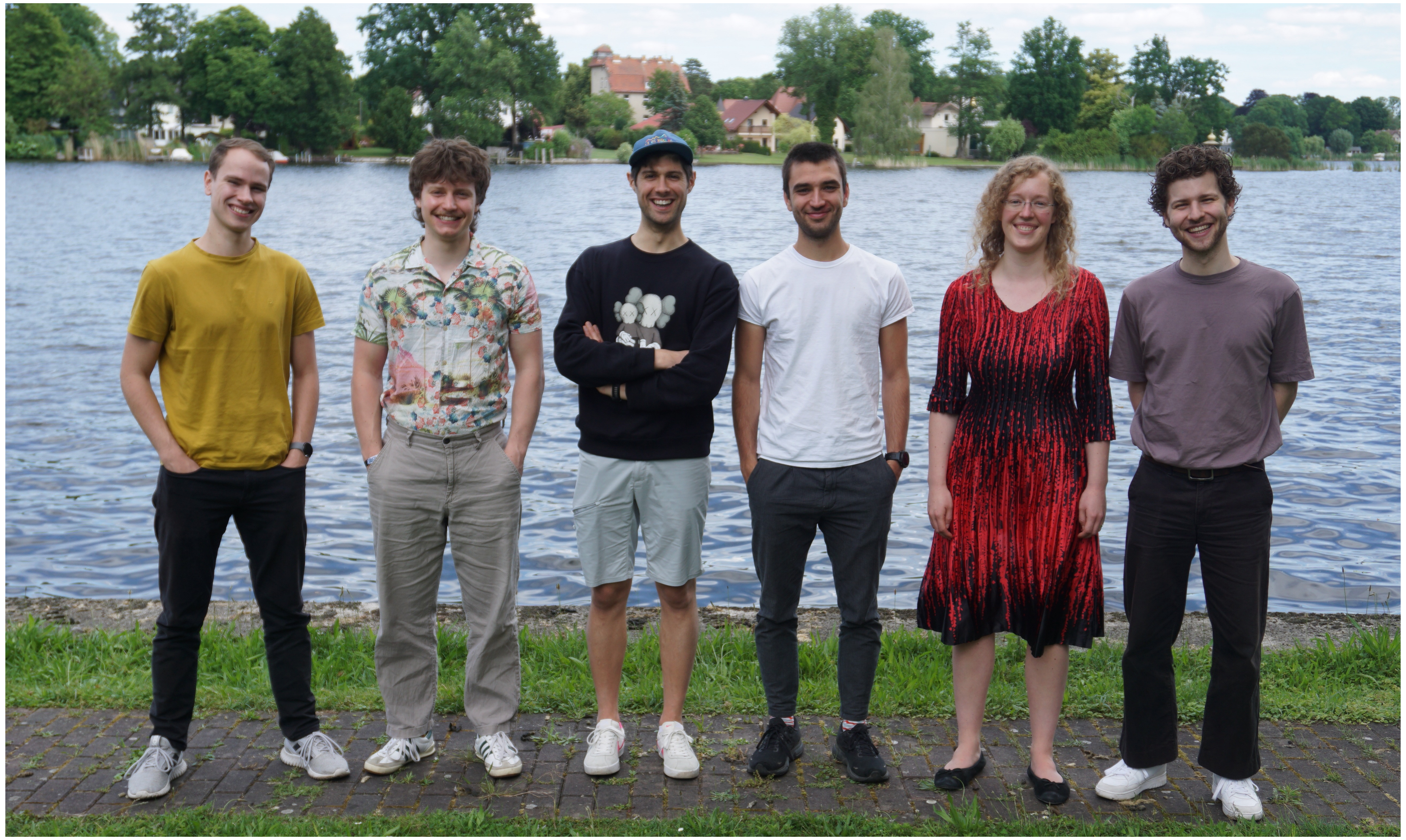
- Hardware noise and error mitigation → see poster by **Luca Wagner**
- Quantum chemistry benchmark → see poster by **Luca Wagner**
- Learn to optimize measurement shots → see **Anders C., Nicoli K.A. et al., ICML (2024)**
- Application in LQFT (work in progress, i.e., 2+1 QED)

Time for some Advertisement

Registrations are open:

https://indico.hiskp.uni-bonn.de/event/443/page/147-home

Thank You!



Link to the paper and code:

https://t.ly/fYWbx https://t.ly/IET-l

Link to the workshop registration:

Back Up Slides

Backup: Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Backup: Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Data: Collection of observations $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$

Task: Provide predictive distribution at new test points $\{\boldsymbol{x}'_j\}_{j=1}^{M}$

Backup: Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Data: Collection of observations $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$

Task: Provide predictive distribution at new test points $\{\boldsymbol{x}'_j\}_{j=1}^{M}$

Assume: Noisy observations of some true function $f^*(\boldsymbol{x})$, i.e.,

$$y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i$$

Backup: Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Data: Collection of observations $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$

Task: Provide predictive distribution at new test points $\{\boldsymbol{x}'_j\}_{j=1}^{M}$

Assume: Noisy observations of some true function $f^*(\boldsymbol{x})$, i.e., $y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i$

We introduce:

Variance of observation noise ε_i

$$p(y \,|\, \boldsymbol{x}, f(\cdot)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$$

GP **regression model** with 1-D Gaussian Likelihood

Backup: Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Data: Collection of observations $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$

Task: Provide predictive distribution at new test points $\{\boldsymbol{x}'_j\}_{j=1}^{M}$

Assume: Noisy observations of some true function $f^*(\boldsymbol{x})$, i.e., $$y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i$$

We introduce:

Variance of observation noise ε_i

$$p(y \,|\, \boldsymbol{x}, f(\,\cdot\,)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$$

GP **regression model** with 1-D Gaussian Likelihood

$$p(f(\,\cdot\,)) = \mathrm{GP}(f(\,\cdot\,); \nu(\,\cdot\,), k(\,\cdot\,,\,\cdot\,)),$$

GP prior

Prior mean $\nu(\,\cdot\,)$

Prior covariance function $k(\,\cdot\,,\,\cdot\,)$

Backup: Gaussian Processes and Bayesian Optimization

A GP is an infinite-dimensional generalization of multivariate Gaussian distribution.

Data: Collection of observations $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$

Task: Provide predictive distribution at new test points $\{\boldsymbol{x}'_j\}_{j=1}^{M}$

Assume: Noisy observations of some true function $f^*(\boldsymbol{x})$, i.e.,

$$y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i$$

We introduce:

$$p(y \,|\, \boldsymbol{x}, f(\cdot)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$$

Variance of observation noise ε_i

GP **regression model** with 1-D Gaussian Likelihood

$$p(f(\cdot)) = \mathrm{GP}(f(\cdot); \nu(\cdot), k(\cdot\,,\cdot)),$$

GP prior

Prior mean $\nu(\cdot)$

Prior covariance function $k(\cdot\,,\cdot)$

Backup: Gaussian Processes and Bayesian Optimization

➡ **Prior covariance function** or **kernel function** $k(\boldsymbol{x}, \boldsymbol{x}' \,|\, \theta)$

Function measuring the similarity between any two inputs $\{\boldsymbol{x}, \boldsymbol{x}'\}$

- Implicitly determines which functions are likely to be sampled.
- It needs to be carefully designed.
- Technical restrictions apply (symmetry, positive-semidefitness).

Backup: Gaussian Processes and Bayesian Optimization

➡ **Prior covariance function** or **kernel function** $k(\boldsymbol{x}, \boldsymbol{x}' \mid \theta)$

Function measuring the similarity between any two inputs $\{\boldsymbol{x}, \boldsymbol{x}'\}$

- Implicitly determines which functions are likely to be sampled.
- It needs to be carefully designed.
- Technical restrictions apply (symmetry, positive-semidefitness).

➡ **The GP regressor can be used in the context of Bayesian Optimization (BO)**

- M set of new **candidate points** $\boldsymbol{X}' \in \mathcal{X}^M$ are selected.
- The **mean** and **variance** of the GP (trained on $D = \{\boldsymbol{X}, \boldsymbol{y}\},\ \boldsymbol{X} \in \mathcal{X}^N$) can be computed at $\boldsymbol{X}'$.
- Those information are used to compute (maximize) a general **acquisition function**

Backup: Gaussian Processes and Bayesian Optimization

➡ **Prior covariance function** or **kernel function** $k(\boldsymbol{x}, \boldsymbol{x}' \,|\, \theta)$

Function measuring the similarity between any two inputs $\{\boldsymbol{x}, \boldsymbol{x}'\}$

- Implicitly determines which functions are likely to be sampled.
- It needs to be carefully designed.
- Technical restrictions apply (symmetry, positive-semidefitness).

➡ **The GP regressor can be used in the context of Bayesian Optimization (BO)**

- M set of new **candidate points** $X' \in \mathcal{X}^M$ are selected.
- The **mean** and **variance** of the GP (trained on $D = \{\boldsymbol{X}, \boldsymbol{y}\},\ \boldsymbol{X} \in \mathcal{X}^N$) can be computed at $\boldsymbol{X}'$.
- Those information are used to compute (maximize) a general **acquisition function**

$$\max_{\boldsymbol{X}'} a_{\boldsymbol{X}^{t-1}}(\boldsymbol{X}')$$

Backup: Gaussian Processes and Bayesian Optimization

➡ **Prior covariance function** or **kernel function** $k(x, x' \mid \theta)$

Function measuring the similarity between any two inputs $\{x, x'\}$

- Implicitly determines which functions are likely to be sampled.
- It needs to be carefully designed.
- Technical restrictions apply (symmetry, positive-semidefitness).

➡ **The GP regressor can be used in the context of Bayesian Optimization (BO)**

- M set of new **candidate points** $X' \in \mathcal{X}^M$ are selected.
- The **mean** and **variance** of the GP (trained on $D = \{X, y\}, \; X \in \mathcal{X}^N$) can be computed at X'.
- Those information are used to compute (maximize) a general **acquisition function**

$$\max_{X'} a_{X^{t-1}}(X') \longrightarrow$$ **Where should I measure next s.t. the information gain is maximized?**

Physics Informed Bayesian Optimization

We tackle the classical optimization problem from a Bayesian Optimization standpoint.

Physics Informed Bayesian Optimization

We tackle the classical optimization problem from a Bayesian Optimization standpoint.

Approximate the true energy $E^*(\cdot) : \mathcal{X} \mapsto \mathbb{R}$

Using noisy observations from the QC $y = E^*(\mathbf{x}) + \varepsilon$ $\quad \varepsilon \sim$ QC noise

Physics Informed Bayesian Optimization

We tackle the classical optimization problem from a Bayesian Optimization standpoint.

Approximate the true energy $E^*(\cdot) : \mathcal{X} \mapsto \mathbb{R}$

Using noisy observations from the QC $y = E^*(\mathbf{x}) + \varepsilon$ $\varepsilon \sim$ QC noise

We train a GP surrogate model $p(f(\cdot)|\mathbf{X}, \mathbf{y}) = \mathrm{GP}(f(\cdot); \mu_{\mathbf{X}}(\cdot), s_{\mathbf{X}}(\cdot, \cdot))$

Physics Informed Bayesian Optimization

We tackle the classical optimization problem from a Bayesian Optimization standpoint.

Approximate the true energy $E^*(\cdot) : \mathcal{X} \mapsto \mathbb{R}$

Using noisy observations from the QC $y = E^*(\mathbf{x}) + \varepsilon$ $\varepsilon \sim$ QC noise

We train a GP surrogate model $p(f(\cdot)|\mathbf{X}, \mathbf{y}) = \mathrm{GP}(f(\cdot); \mu_{\mathbf{X}}(\cdot), s_{\mathbf{X}}(\cdot, \cdot))$

One question remains to be answered:

*At which point in parameter space we should perform the **next measurement**, on the quantum computer, to **maximize the information gain** and **minimize the quantum computer calls** needed to minimize the objective?*

The VQE Kernel

Computing derivatives on the quantum computer is challenging.

They can be computed using the so-called **parameter shift rule (PSR)** Schuld et al., (2019)

$$2\frac{\partial}{\partial x_d} f^*(\mathbf{x}) = f^*\left(\mathbf{x} + \tfrac{\pi}{2}\mathbf{e}_d\right) - f^*\left(\mathbf{x} - \tfrac{\pi}{2}\mathbf{e}_d\right)$$

The VQE Kernel

Computing derivatives on the quantum computer is challenging.

They can be computed using the so-called **parameter shift rule (PSR)** Schuld et al., (2019)

$$2\frac{\partial}{\partial x_d} f^*(\mathbf{x}) = f^*\left(\mathbf{x} + \tfrac{\pi}{2}\mathbf{e}_d\right) - f^*\left(\mathbf{x} - \tfrac{\pi}{2}\mathbf{e}_d\right)$$

Starting from this Nakanishi et al., (2020) show that the VQE objective $f^*(\,\cdot\,)$ obeys

$$\exists \mathbf{b} \in \mathbb{R}^{3^D} \quad \text{s.t.} \quad f^*(\mathbf{x}) = \mathbf{b}^\top \cdot \mathbf{vec}\left(\otimes_{d=1}^{D}(1, \cos x_d, \sin x_d)^\top\right)$$

The VQE Kernel

Computing derivatives on the quantum computer is challenging.

They can be computed using the so-called **parameter shift rule (PSR)** Schuld et al., (2019)

$$2\frac{\partial}{\partial x_d} f^*(\mathbf{x}) = f^*\left(\mathbf{x} + \tfrac{\pi}{2}\mathbf{e}_d\right) - f^*\left(\mathbf{x} - \tfrac{\pi}{2}\mathbf{e}_d\right)$$

Starting from this Nakanishi et al., (2020) show that the VQE objective $f^*(\,\cdot\,)$ obeys

$$\exists \mathbf{b} \in \mathbb{R}^{3^D} \quad \text{s.t.} \qquad f^*(\mathbf{x}) = \mathbf{b}^\top \cdot \mathbf{vec}\left(\otimes_{d=1}^{D} (1, \cos x_d, \sin x_d)^\top\right)$$

We thus derive a **covariance function** $k(\,\cdot\,,\,\cdot\,)$ fulfilling the same functional

$$k^{\mathrm{VQE}}(\mathbf{x}, \mathbf{x}') = \sigma_0^2 \prod_{d=1}^{D} \left(\frac{\gamma^2 + 2\cos\left(x_d - x'_d\right)}{\gamma^2 + 2}\right)$$

See Nicoli et al., (2023) for detailed proofs.

EMICoRe

Expected **M**aximum **I**mprovement

EMICoRe

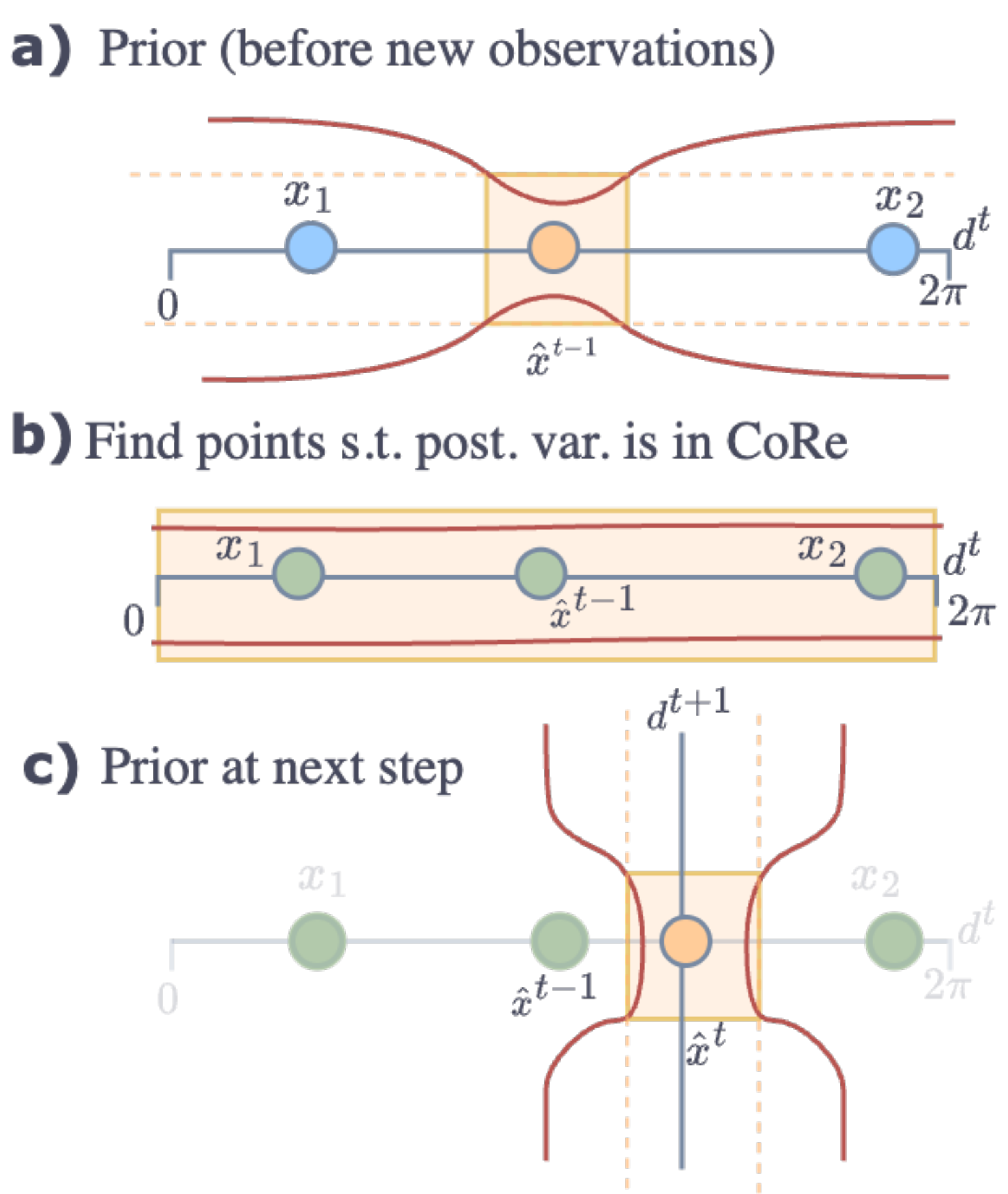
Confident **Re**gions

EMICoRe

Expected Maximum Improvement **EMICoRe** **Confident Regions**

For a set of samples, $\boldsymbol{X}$ we compute the Confident Region

$$\mathcal{Z}_{\boldsymbol{X}} = \{\boldsymbol{x} \in \mathcal{X};\ s_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{x}) \leq \kappa^2\}$$



EMICoRe

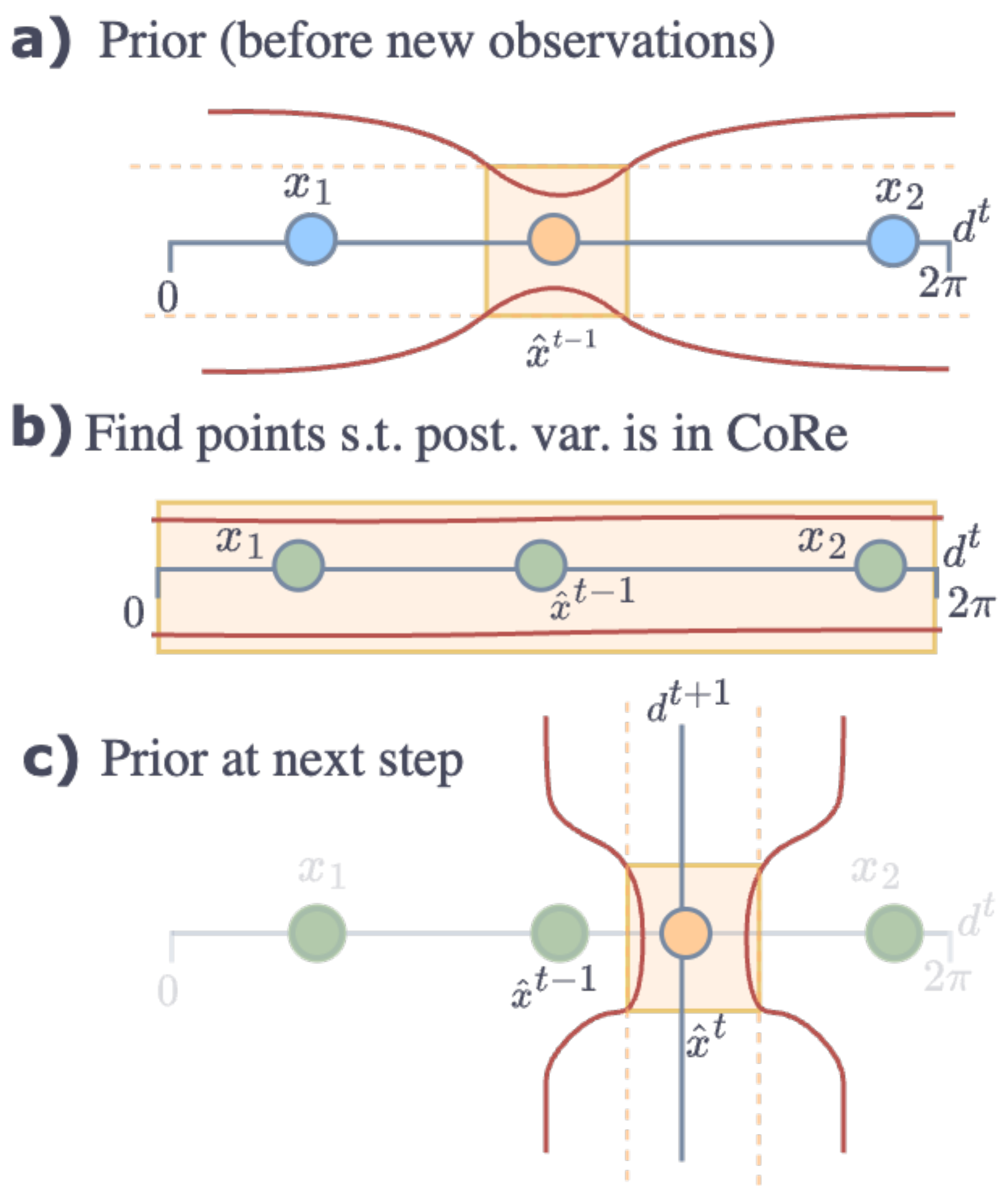
Expected Maximum Improvement **EMICoRe** **Confident Regions**

For each set of candidate pairs $\boldsymbol{X}' \in \mathcal{X}^M$, i.e., $M = 2$, we compute

$$a_{\boldsymbol{X}}(\boldsymbol{X}') = \frac{1}{M} \langle \max(0, \min_{\boldsymbol{x} \in \mathcal{Z}_{\boldsymbol{X}}} f(\boldsymbol{x}) - \min_{\boldsymbol{x} \in \mathcal{Z}_{(\boldsymbol{X}, \boldsymbol{X}')}} f(\boldsymbol{x})) \rangle_{p(f(\cdot)|\boldsymbol{X}, \boldsymbol{y})}$$

For a set of samples, $\boldsymbol{X}$ we compute the Confident Region

$$\mathcal{Z}_{\boldsymbol{X}} = \{\boldsymbol{x} \in \mathcal{X};\, s_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{x}) \leq \kappa^2\}$$



EMICoRe

Expected Maximum Improvement **EMICoRe** **Confident Regions**

For each set of candidate pairs $\boldsymbol{X}' \in \mathcal{X}^M$, i.e., $M = 2$, we compute

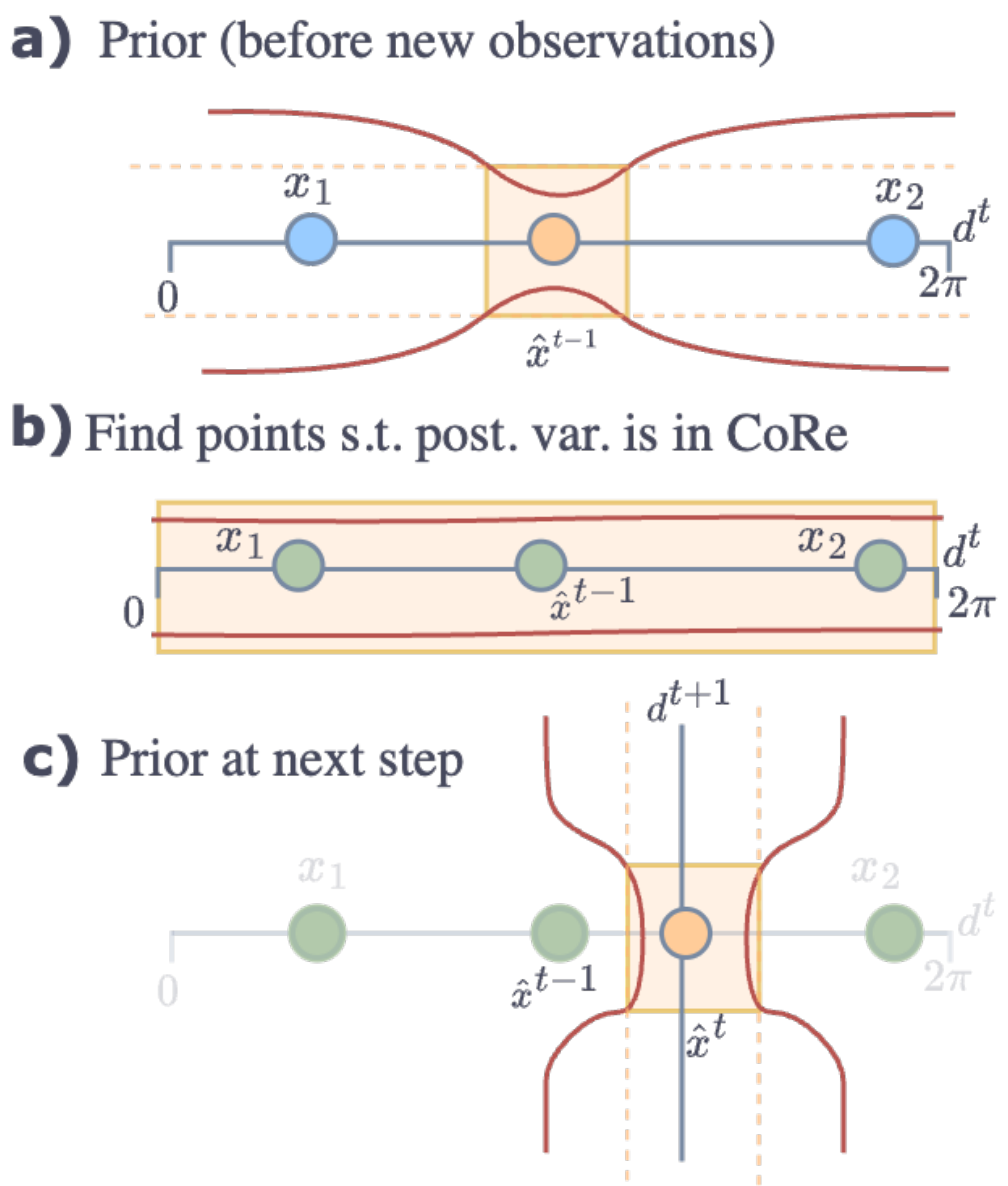
$$a_{\boldsymbol{X}}(\boldsymbol{X}') = \frac{1}{M}\langle \max(0, \min_{\boldsymbol{x}\in\mathcal{Z}_{\boldsymbol{X}}} f(\boldsymbol{x}) - \min_{\boldsymbol{x}\in\mathcal{Z}_{(\boldsymbol{X},\boldsymbol{X}')}} f(\boldsymbol{x}))\rangle_{p(f(\cdot)|\boldsymbol{X},\boldsymbol{y})}$$

The **best candidate pair** $\tilde{X}$ to measure at next step is

$$\tilde{\boldsymbol{X}} = \mathrm{argmax}_{\boldsymbol{X}'\in\mathcal{X}^M}\ a_{\boldsymbol{X}^{t-1}}(\boldsymbol{X}')$$

For a set of samples, $\boldsymbol{X}$ we compute the Confident Region

$$\mathcal{Z}_{\boldsymbol{X}} = \{\boldsymbol{x} \in \mathcal{X};\ s_{\boldsymbol{X}}(\boldsymbol{x},\boldsymbol{x}) \leq \kappa^2\}$$



EMICoRe

Expected Maximum Improvement — **Confident Regions**

For each set of candidate pairs $\boldsymbol{X}' \in \mathcal{X}^M$, i.e., $M = 2$, we compute

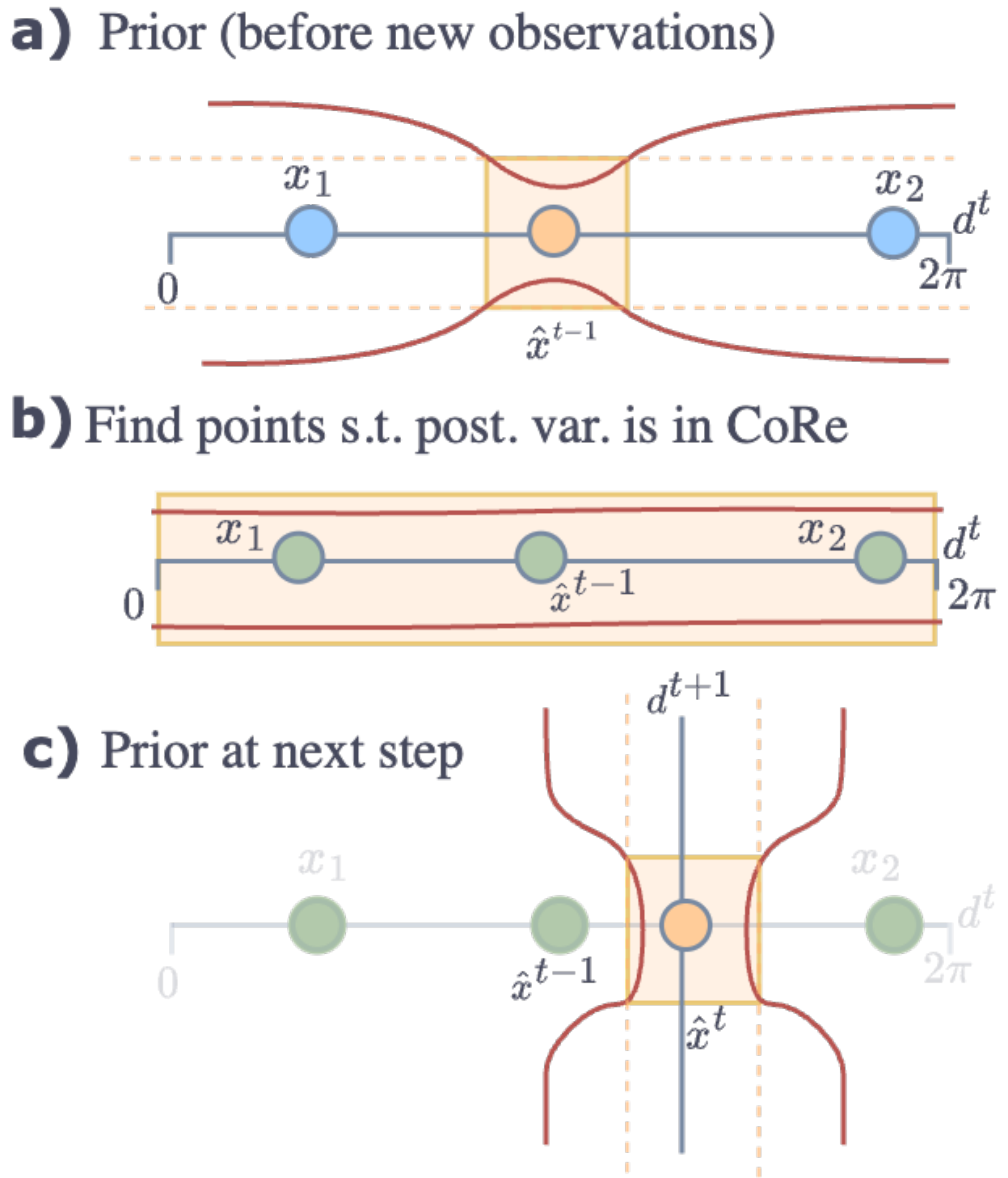
$$a_{\boldsymbol{X}}(\boldsymbol{X}') = \frac{1}{M}\langle \max(0, \min_{\boldsymbol{x}\in\mathcal{Z}_{\boldsymbol{X}}} f(\boldsymbol{x}) - \min_{\boldsymbol{x}\in\mathcal{Z}_{(\boldsymbol{X},\boldsymbol{X}')}} f(\boldsymbol{x}))\rangle_{p(f(\cdot)|\boldsymbol{X},\boldsymbol{y})}$$

The **best candidate pair** $\tilde{X}$ to measure at next step is

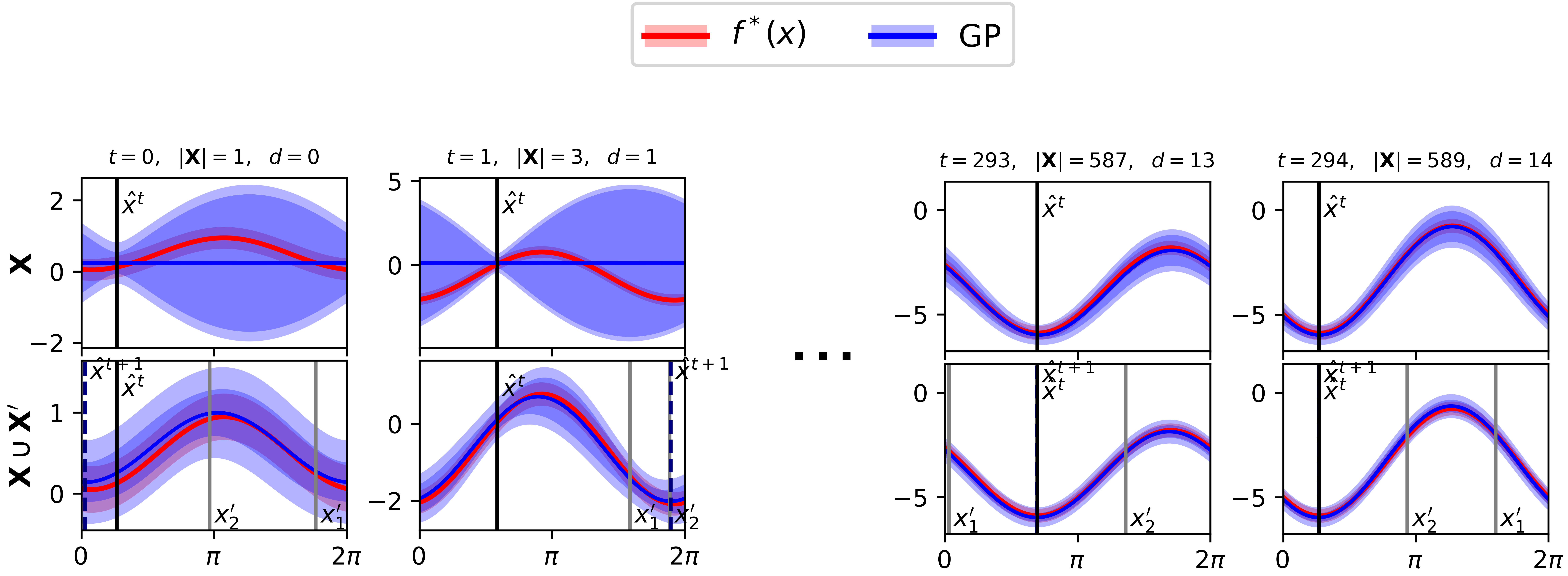
$$\tilde{\boldsymbol{X}} = \mathrm{argmax}_{\boldsymbol{X}'\in\mathcal{X}^M}\ a_{\boldsymbol{X}^{t-1}}(\boldsymbol{X}')$$

For a set of samples, $\boldsymbol{X}$ we compute the Confident Region

$$\mathcal{Z}_{\boldsymbol{X}} = \{\boldsymbol{x} \in \mathcal{X};\ s_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{x}) \leq \kappa^2\}$$



Backup: GP Visualization



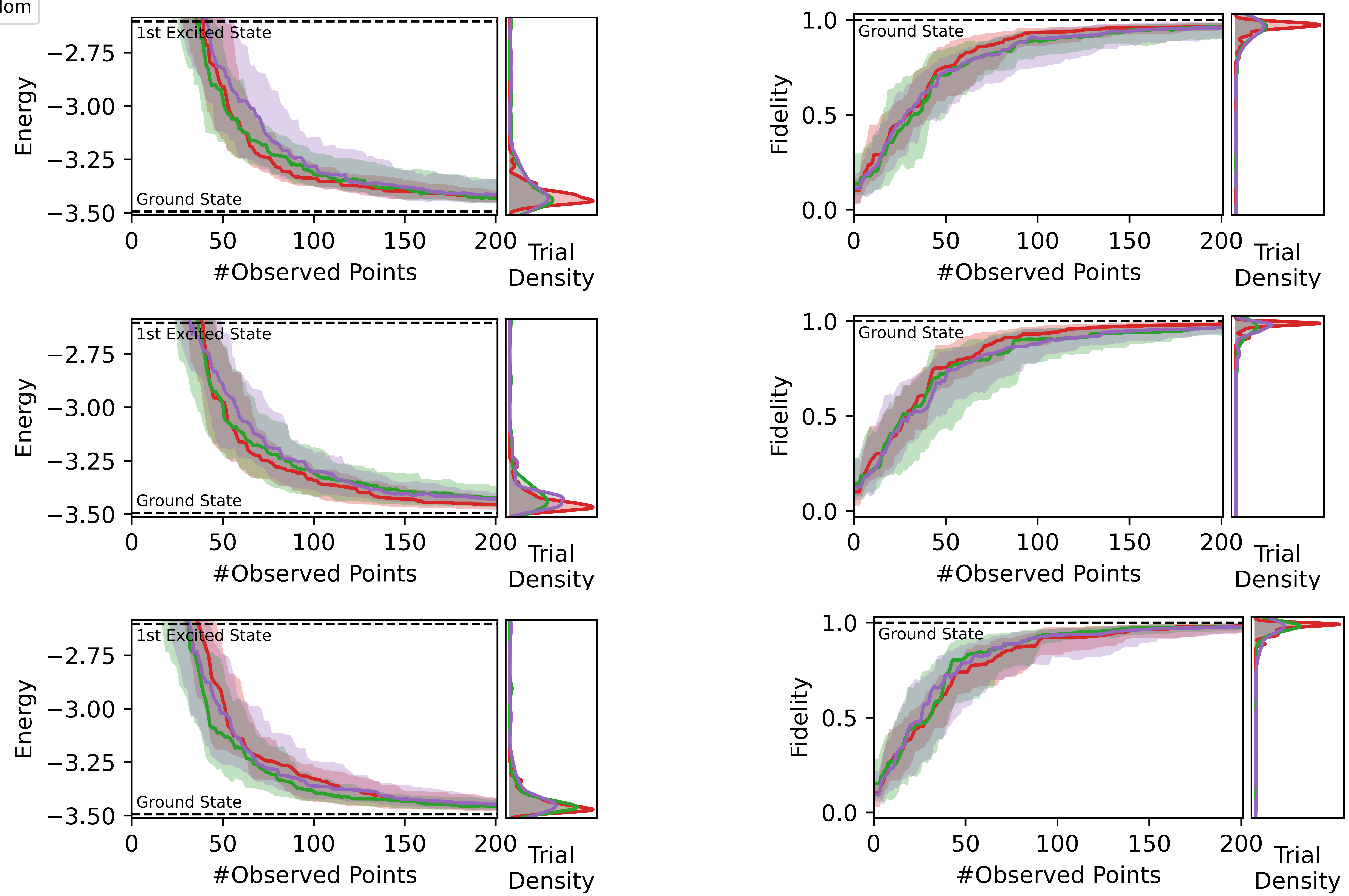
Backup: 3 qubits (Critical Ising)

EMICoRe (ours) NFT-Sequential NFT-Random

$N_{\text{Shots}} = 256$

$N_{\text{Shots}} = 512$

$N_{\text{Shots}} = 1024$

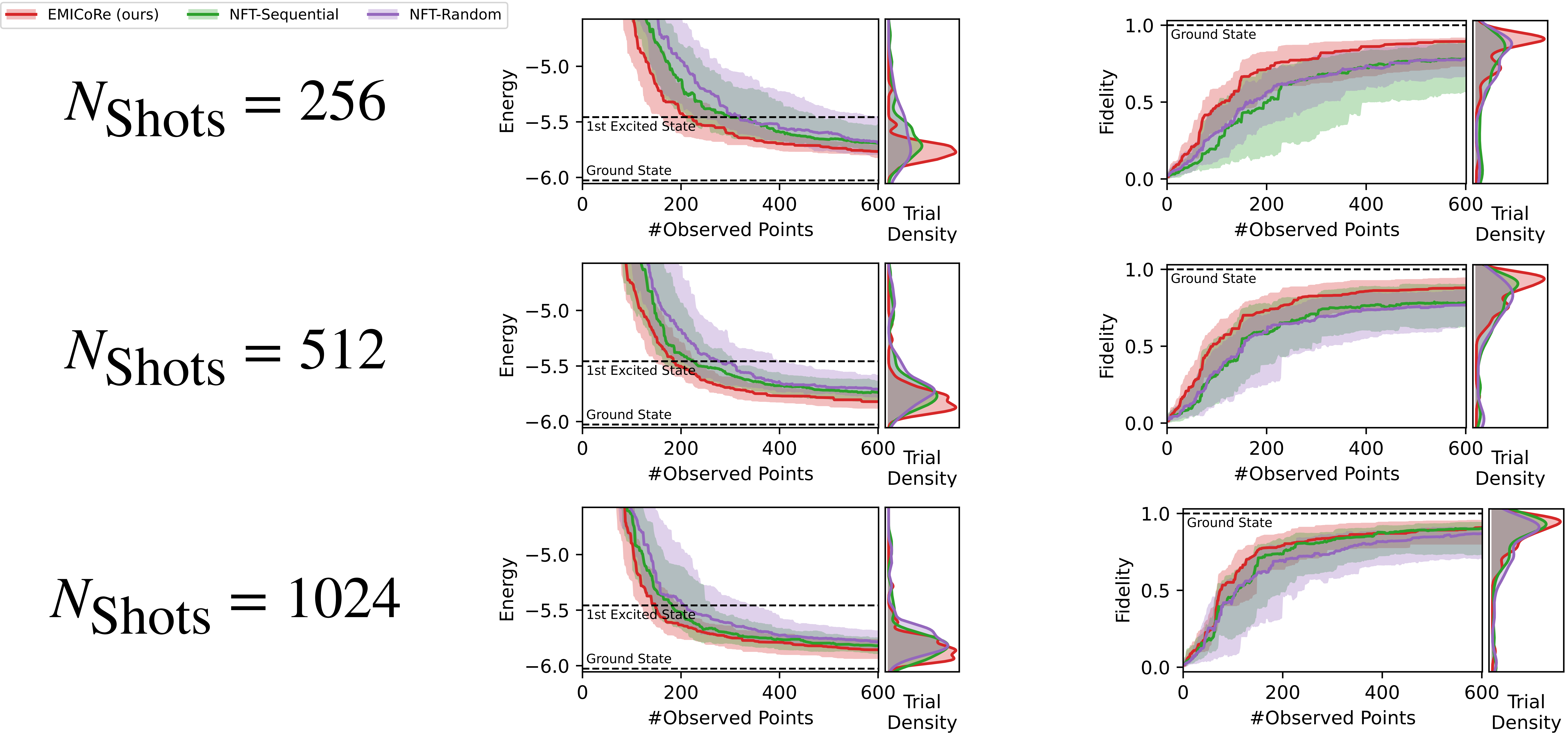


Backup: 5 qubits (Critical Ising)

$$N_{\text{Shots}} = 256$$

$$N_{\text{Shots}} = 512$$

$$N_{\text{Shots}} = 1024$$

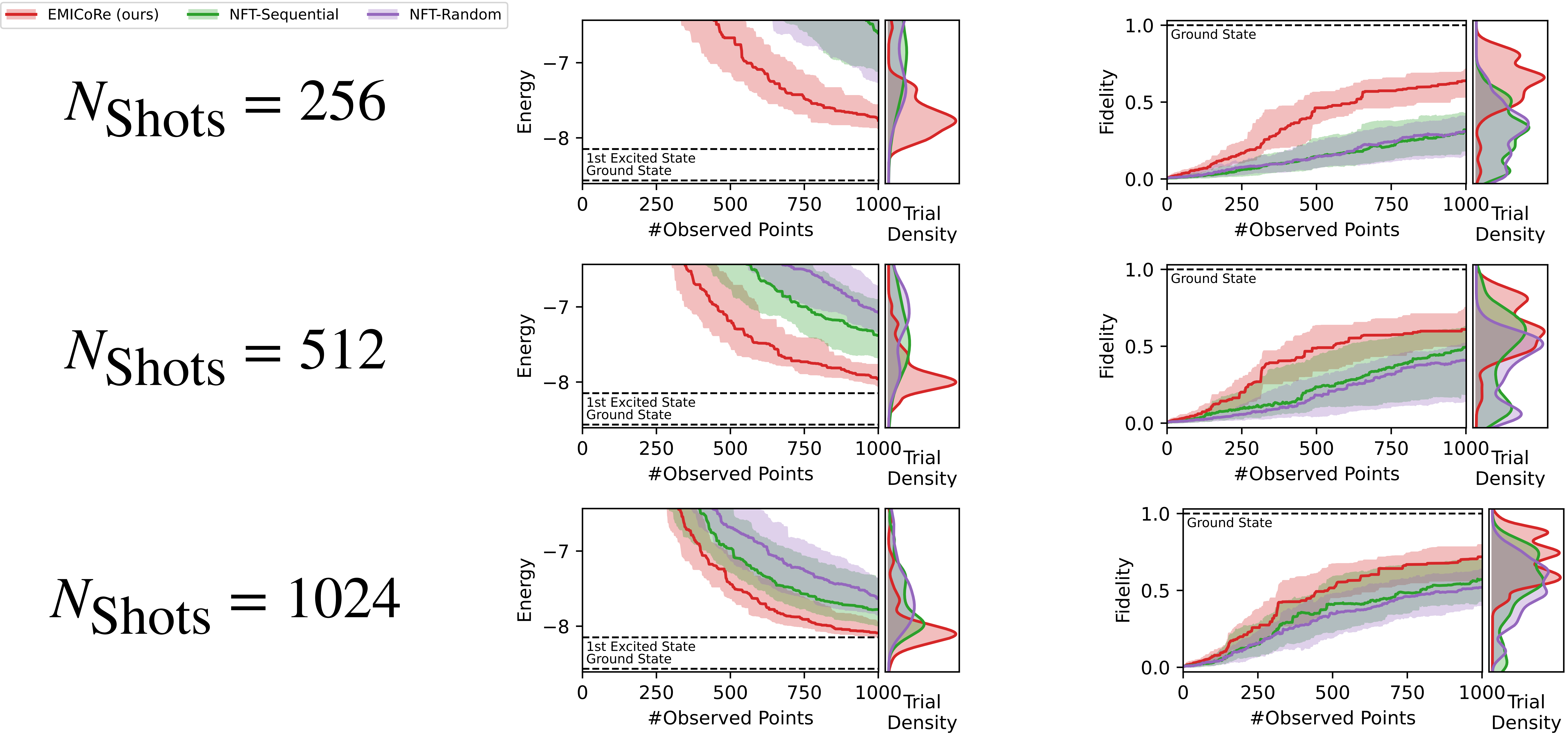


Backup: 7 qubits (Critical Ising)

$N_{\text{Shots}} = 256$

$N_{\text{Shots}} = 512$

$N_{\text{Shots}} = 1024$

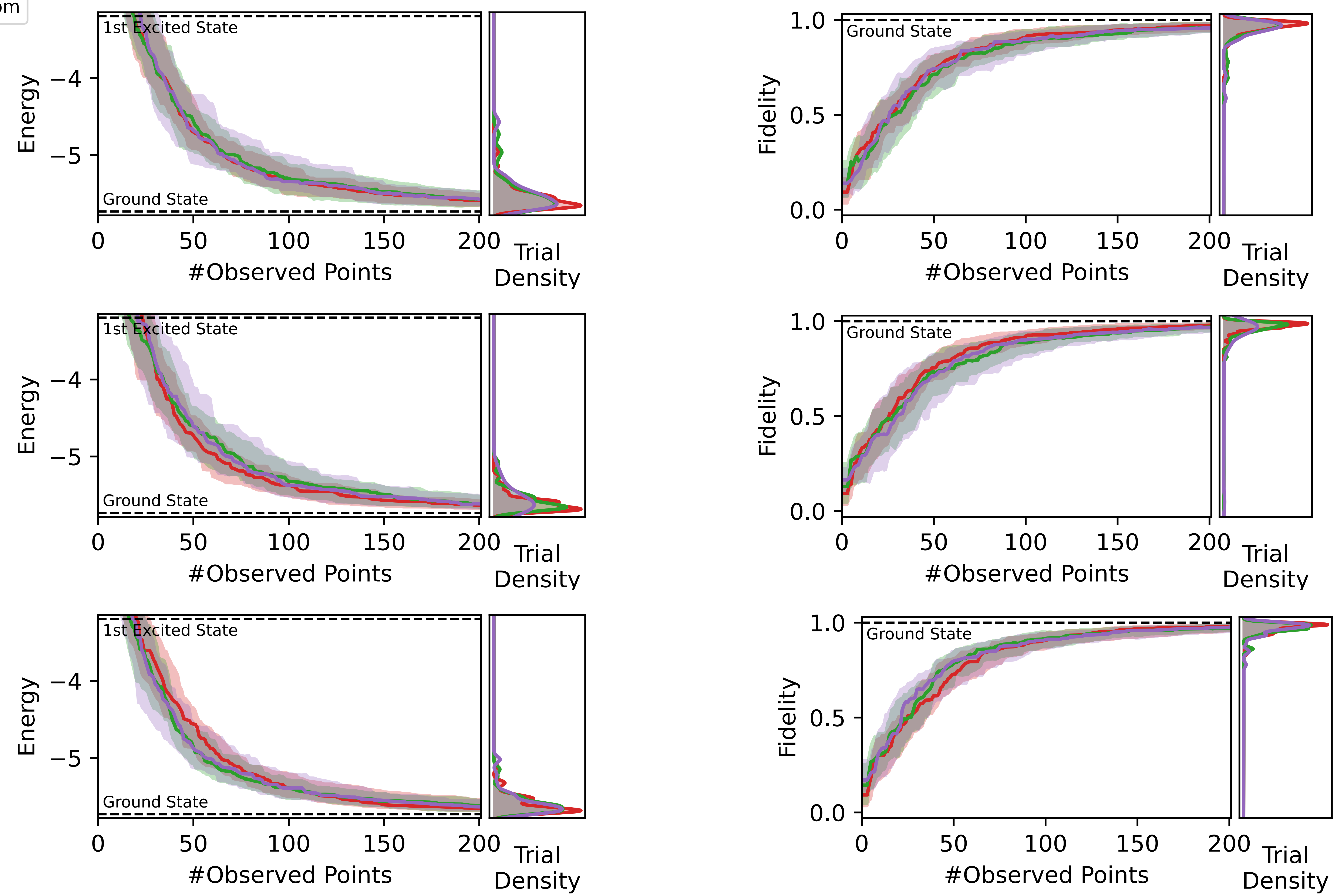


Backup: 3 qubits (Heisenberg)

$N_{\text{Shots}} = 256$

$N_{\text{Shots}} = 512$

$N_{\text{Shots}} = 1024$

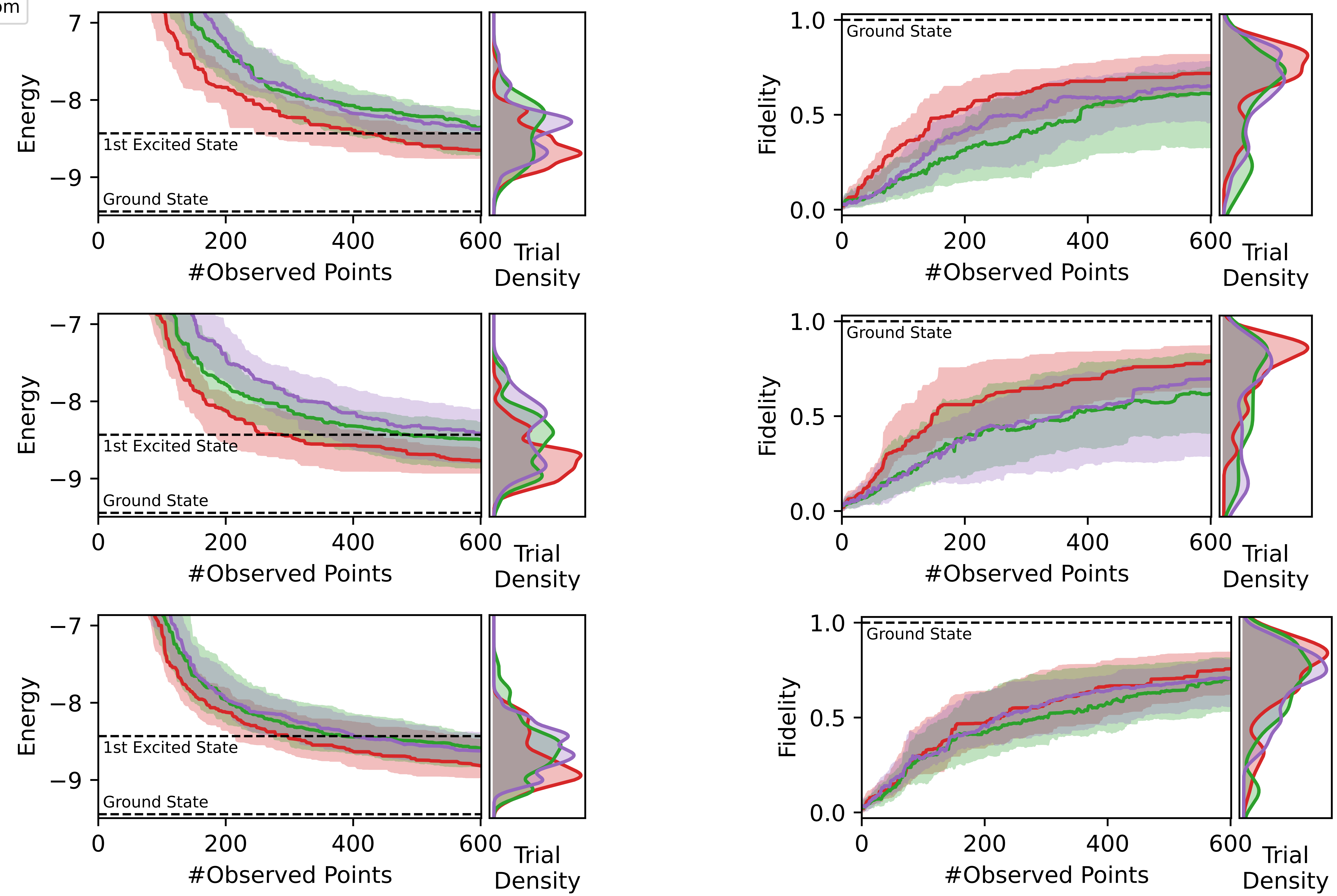


Backup: 5 qubits (Heisenberg)

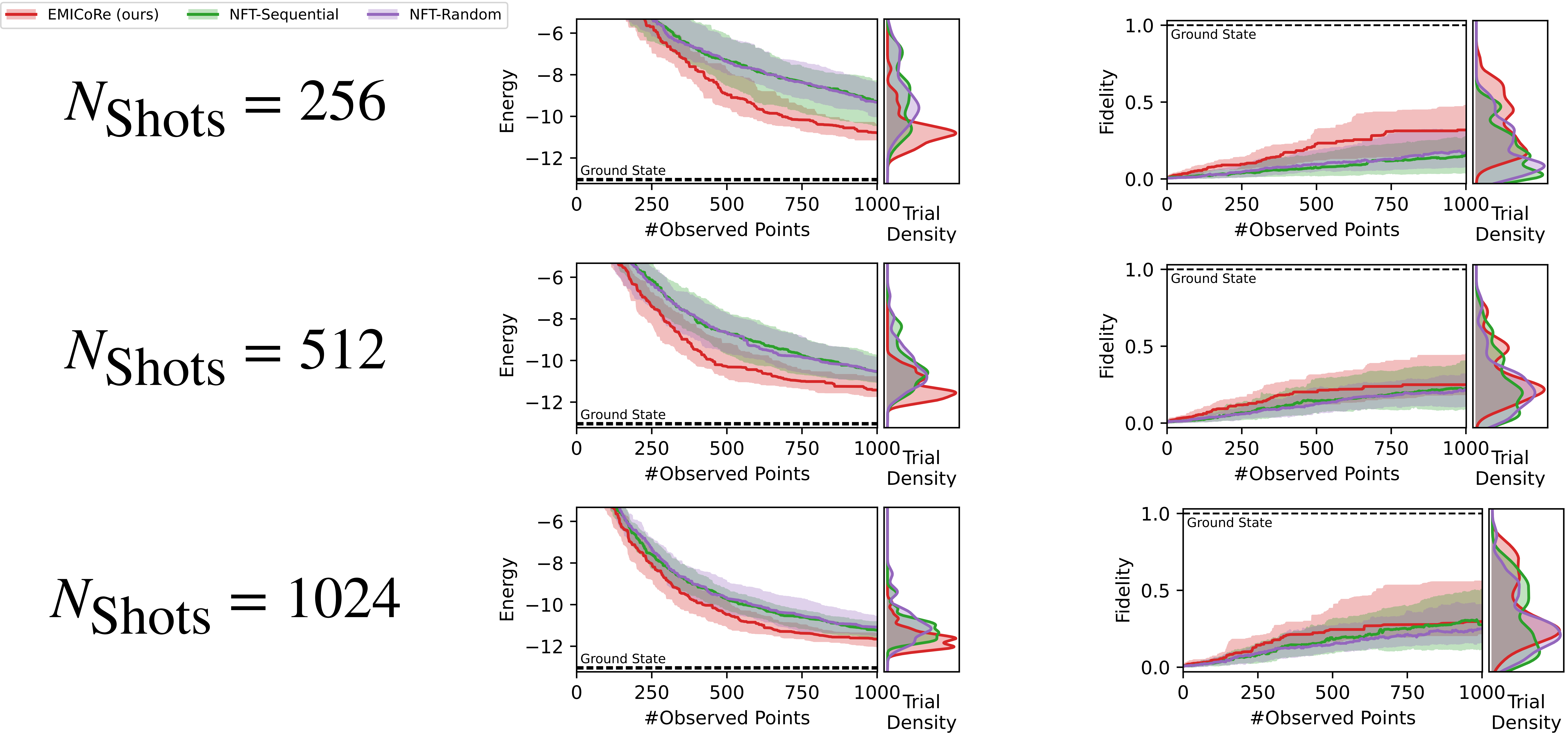
$N_{\text{Shots}} = 256$

$N_{\text{Shots}} = 512$

$N_{\text{Shots}} = 1024$



Backup: 7 qubits (Heisenberg)

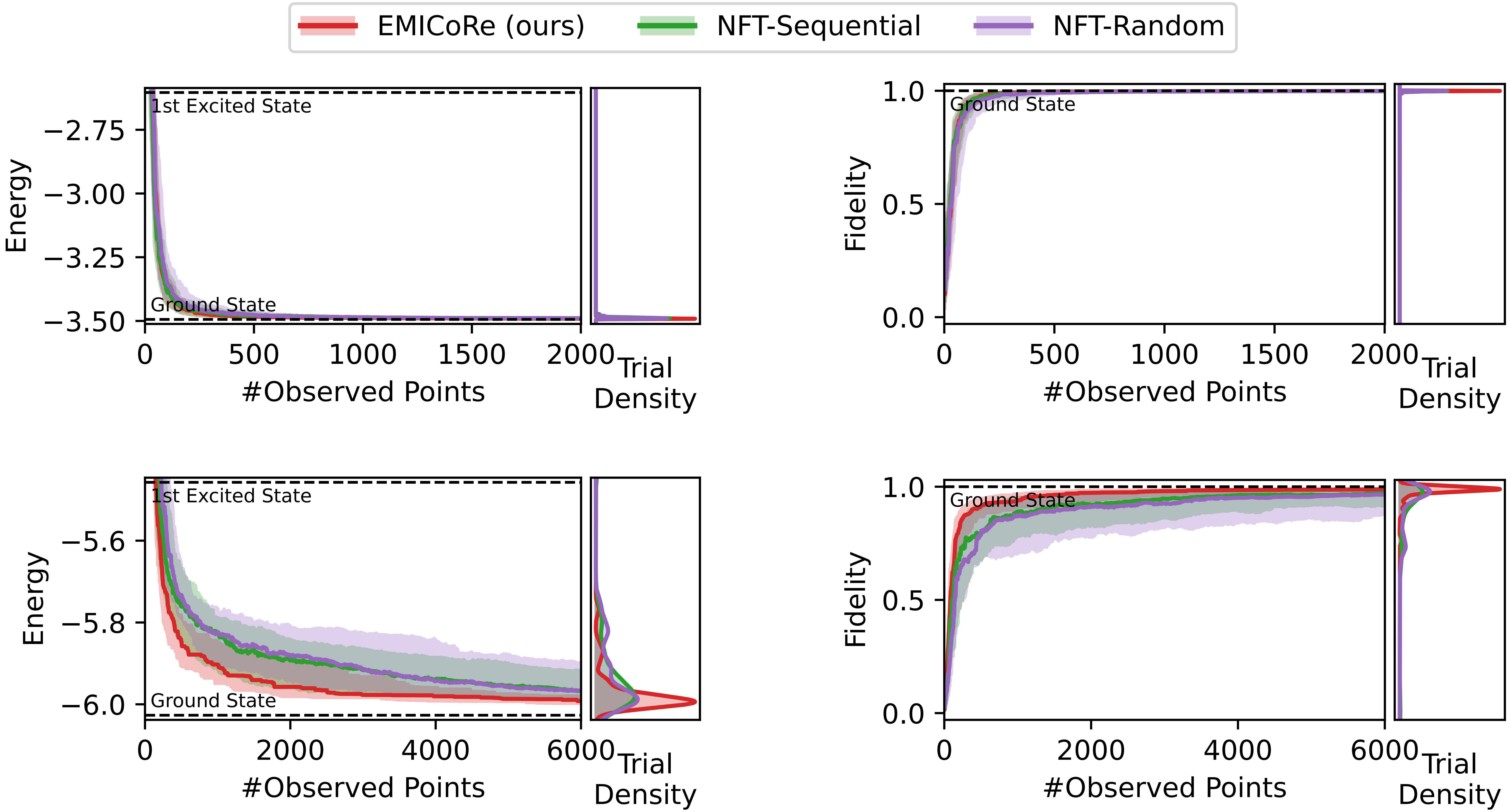


$$N_{\text{Shots}} = 256$$

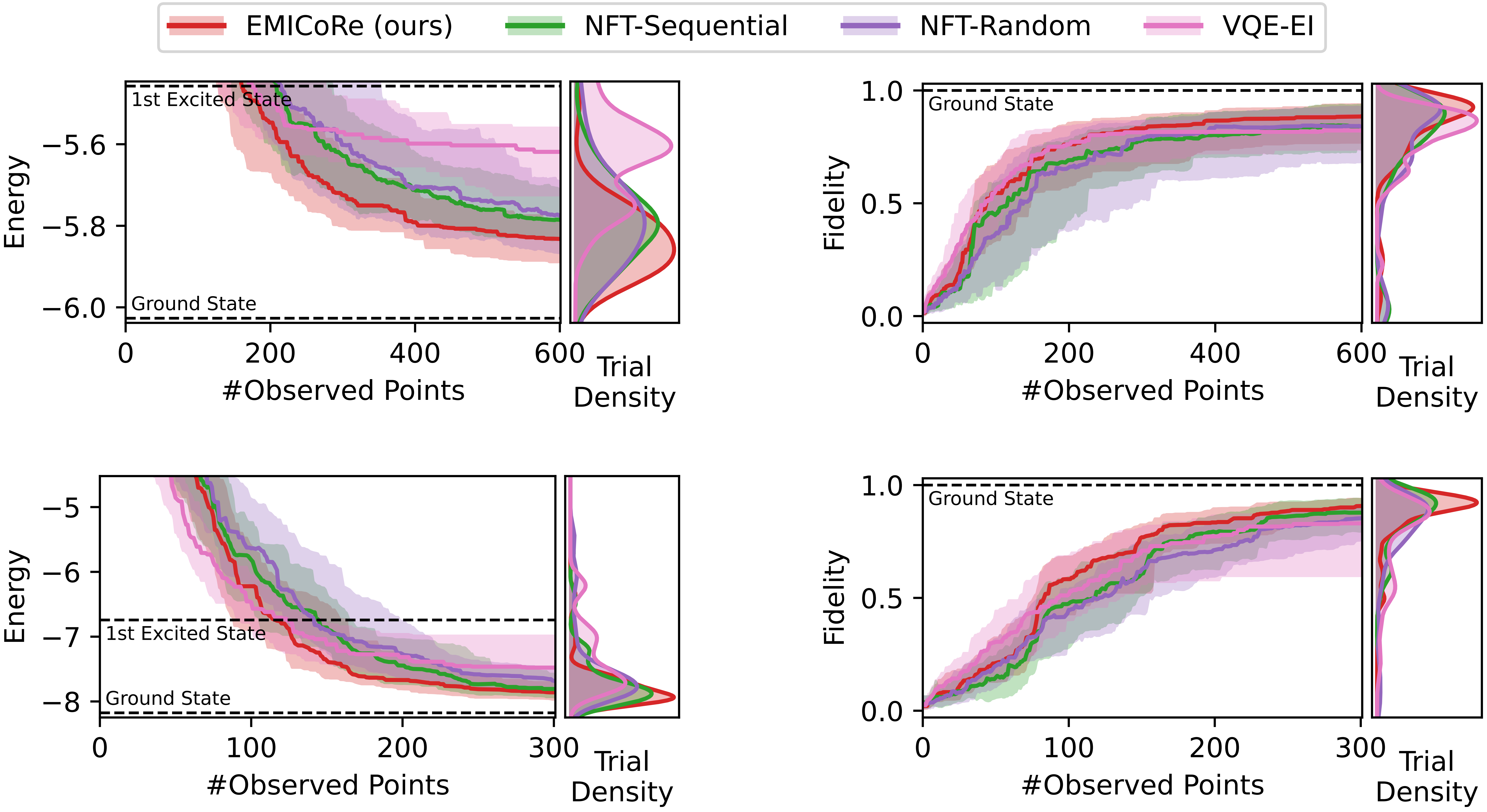
$$N_{\text{Shots}} = 512$$

$$N_{\text{Shots}} = 1024$$

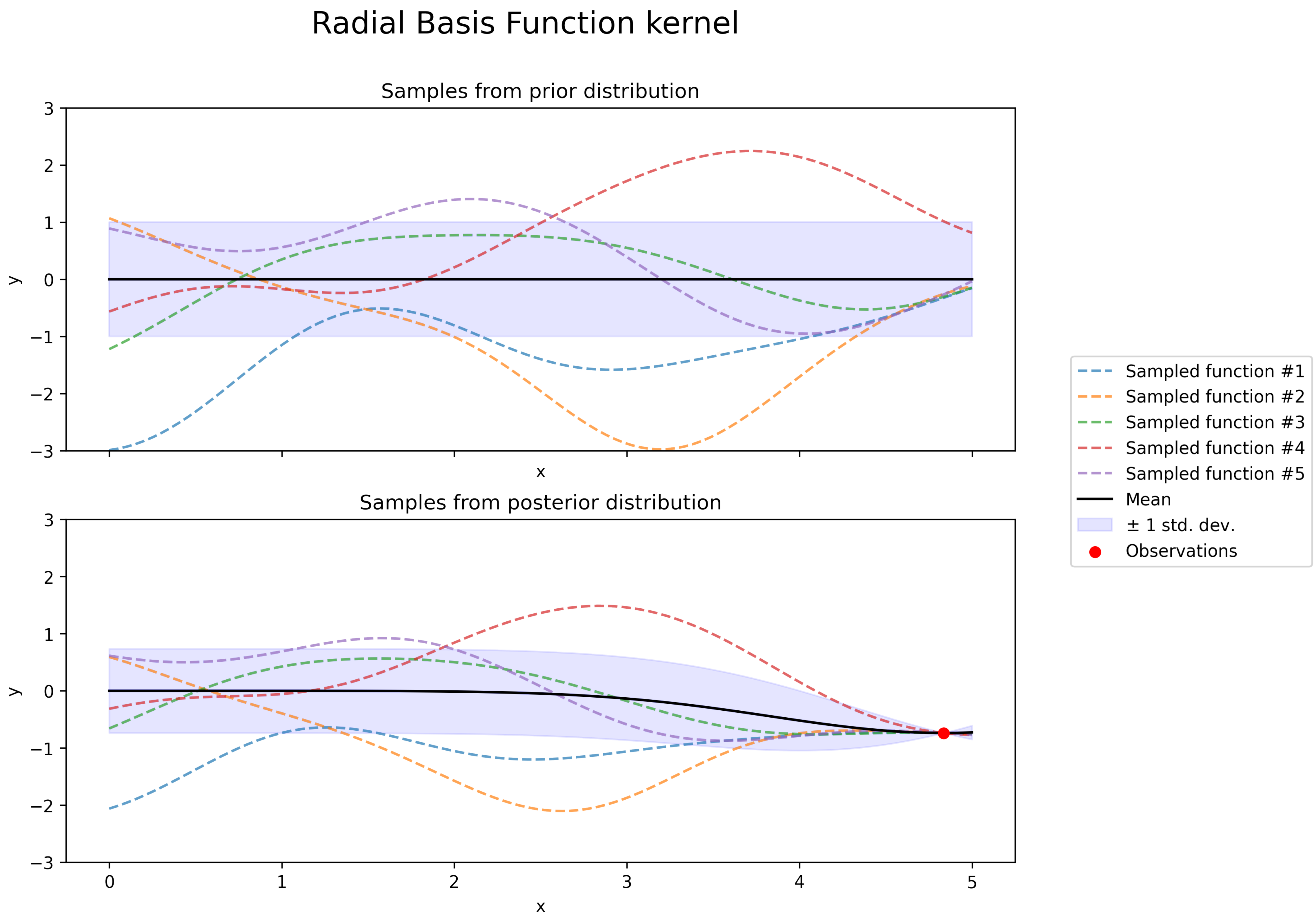
Backup: Convergence for Longer Runs



Backup: Ablation Study (5 qubits)

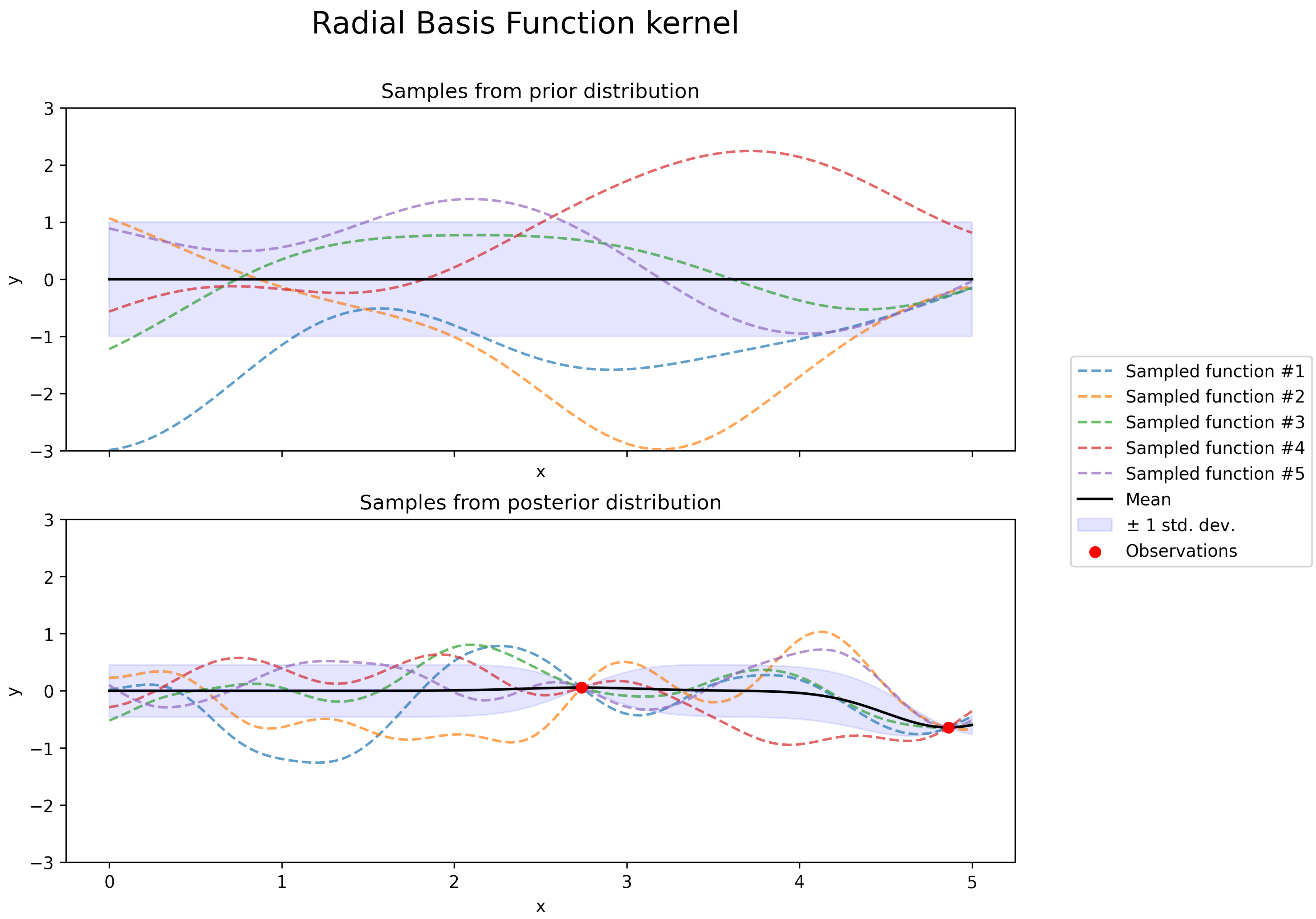


Backup: Gaussian Processes Regression



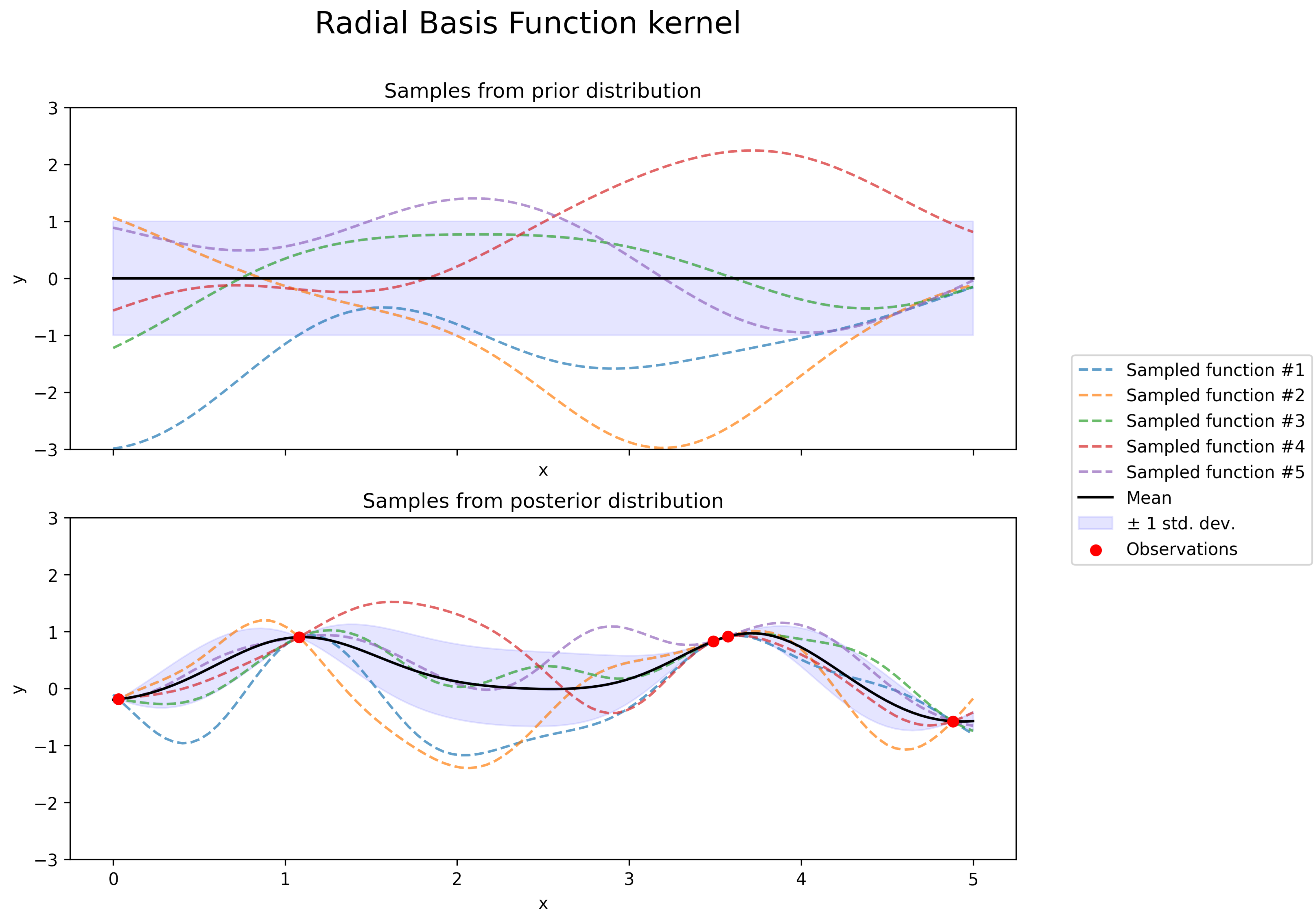
$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \exp\left(-\frac{\| \boldsymbol{x} - \boldsymbol{x}' \|^2}{2\gamma^2}\right)$$

Backup: Gaussian Processes Regression



$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \exp\left(-\frac{\| \boldsymbol{x} - \boldsymbol{x}' \|^2}{2\gamma^2}\right)$$

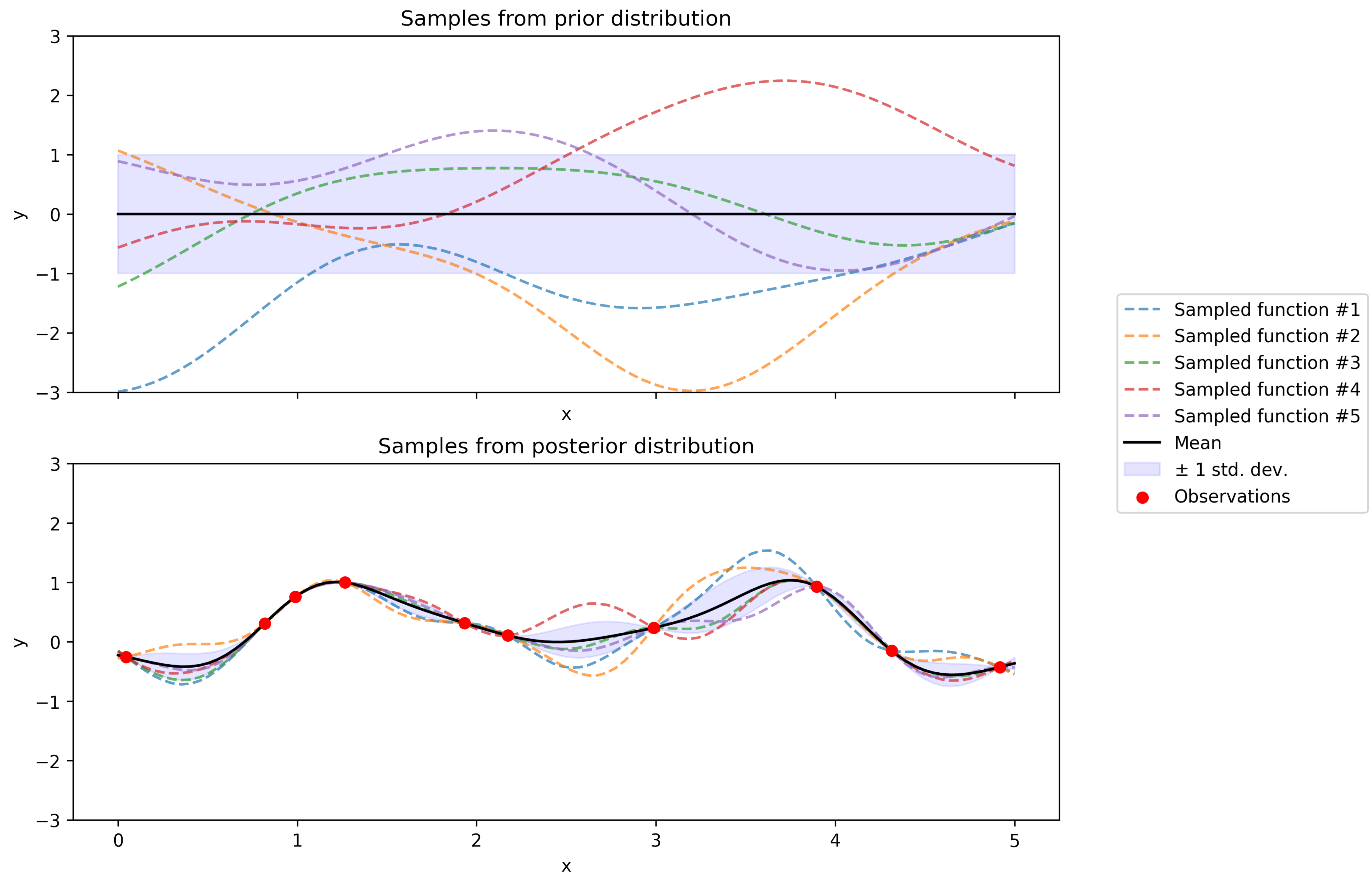
Backup: Gaussian Processes Regression



$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \exp\left(-\frac{\| \boldsymbol{x} - \boldsymbol{x}' \|^2}{2\gamma^2}\right)$$

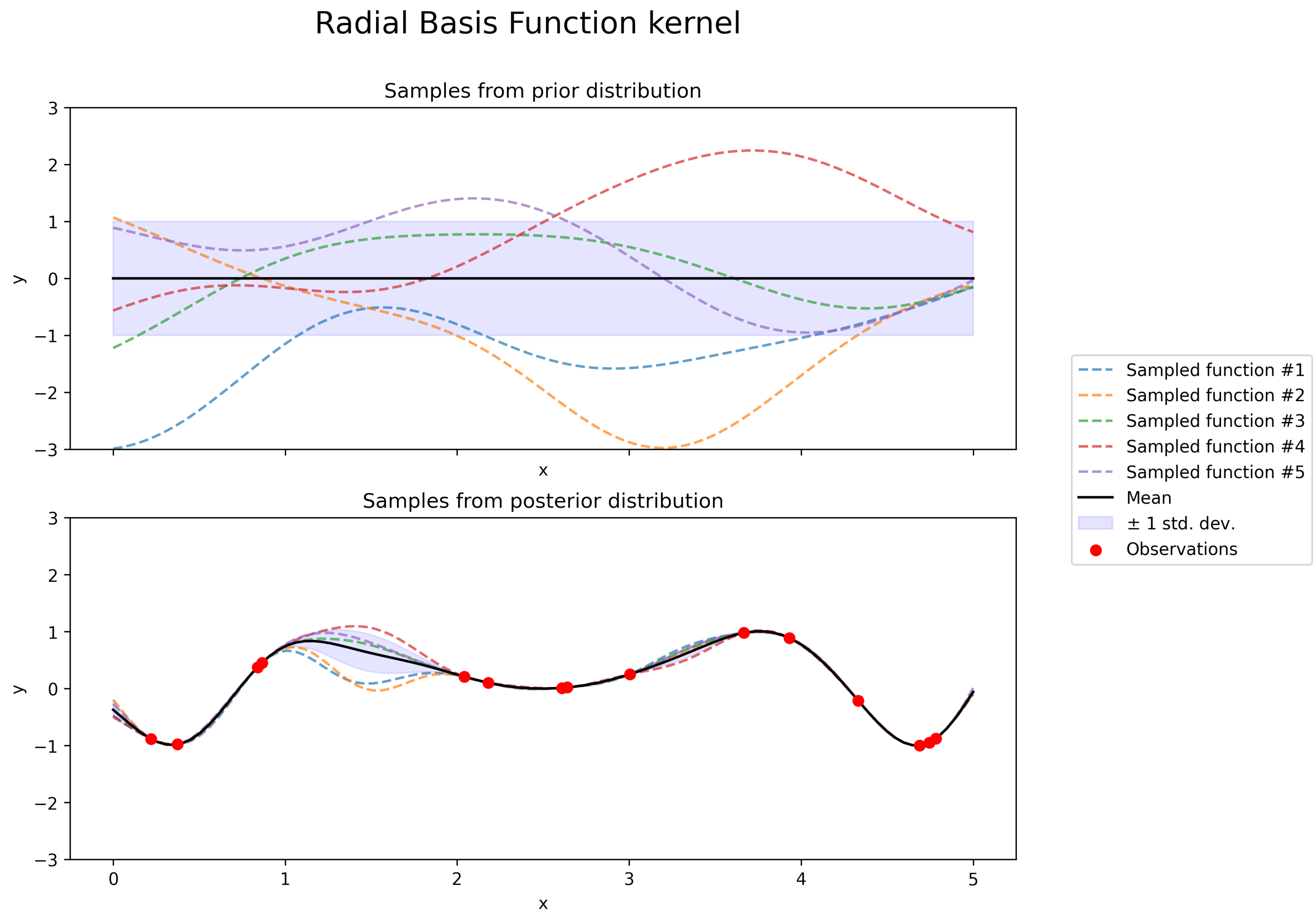
Backup: Gaussian Processes Regression

Radial Basis Function kernel



$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \exp\left(-\frac{\| \boldsymbol{x} - \boldsymbol{x}' \|^2}{2\gamma^2}\right)$$

Backup: Gaussian Processes Regression



$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \exp\left(-\frac{\| \boldsymbol{x} - \boldsymbol{x}' \|^2}{2\gamma^2}\right)$$