Confidential

PASQAL

QUANTUM DISCOVERY

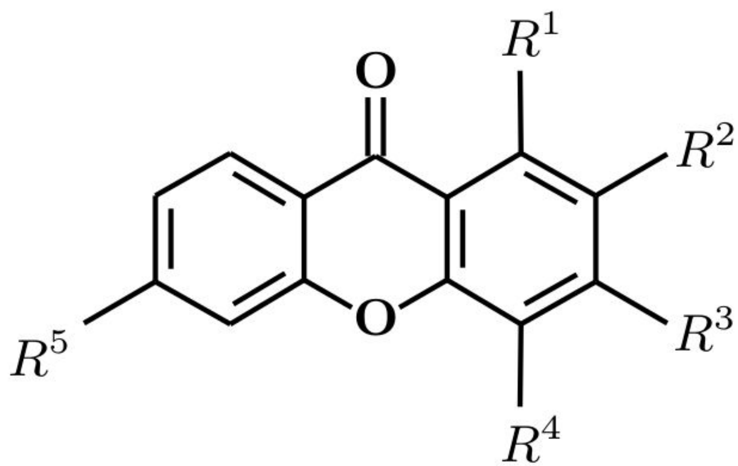
PASQAL Quantum Extremal Learning (QEL)

PASQAL
www.pasqal.com
office@pasqal.com
7 rue Léonard de Vinci
91300 Massy
France

Extremal Learning

Example problem

Find the optimal combination of substituents {R} to minimize the total energy associated with the following molecular structure…



…knowing that there exists no complete ab-initio model describing the relationship between structure and energy.

Extremal Learning

Extremal learning methodology

Extremal learning is a two-step process:

Step 1 - Modeling

The **supervised learning task** to find the optimal f(x) that adjusts some training data {x,y} such that y = f(x)

Step 2 - Extremizing

The **optimization task** to find the x value that extremizes f(x), i.e. minimizes or maximizes

Extremal Learning

Extremal learning methodology

Input / output relationship:

$$\underbrace{\boxed{y}}_{\text{Dependant variable}} = f(\overbrace{\boxed{\boldsymbol{\theta}}}^{\text{Model parameters}}; \underbrace{\boxed{\boldsymbol{x}}}_{\text{Explicative variable(s)}})$$

- **Step 1:** Train a predefined model f through varying the thetas
- **Step 2:** Freezing the thetas at optimum, find the x that extremizes y

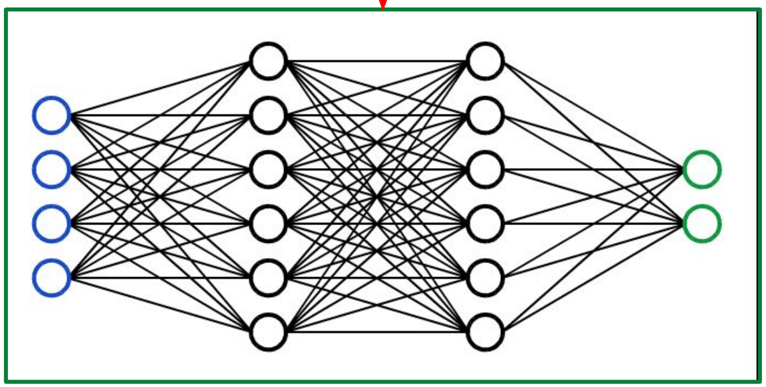
Extremal Learning

The case of classical neural networks

Schematic of a neural network:

To be adjusted through gradient descent when modeling

x



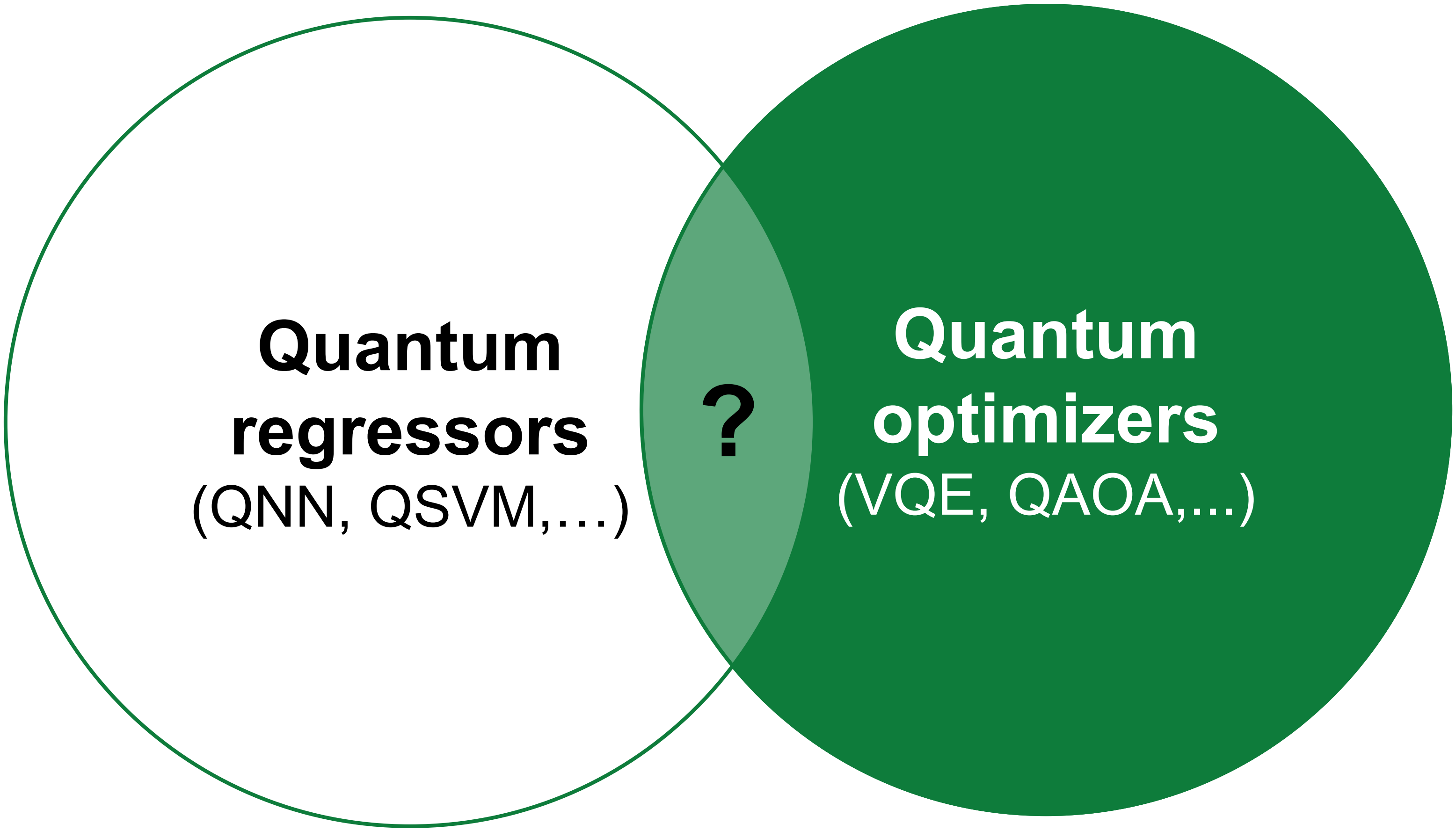
$$y = f(\boxed{\boldsymbol{\theta}}; \boxed{\boldsymbol{x}})$$

To be adjusted through gradient descent when extremizing

- **Pros**: Neural networks are universal function generators
- **Cons**: Non-convex optimization

Quantum Extremal Learning

Background

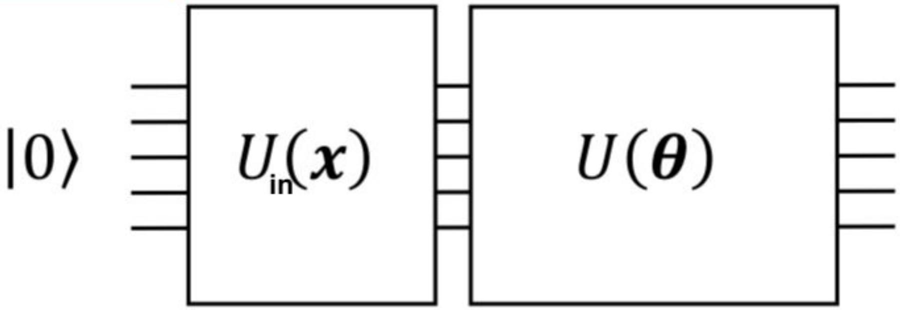


PASQAL

Quantum Extremal Learning

Quantum Circuit Learning (QCL)

QCL



$\langle B(\boldsymbol{x},\boldsymbol{\theta})\rangle \rightarrow L(f(\boldsymbol{x}),\langle \boldsymbol{B}(\boldsymbol{x},\boldsymbol{\theta})\rangle)$

: to be minimized via $\boldsymbol{\theta}$

Quantum Extremal Learning

Modeling

Step 1: Modeling

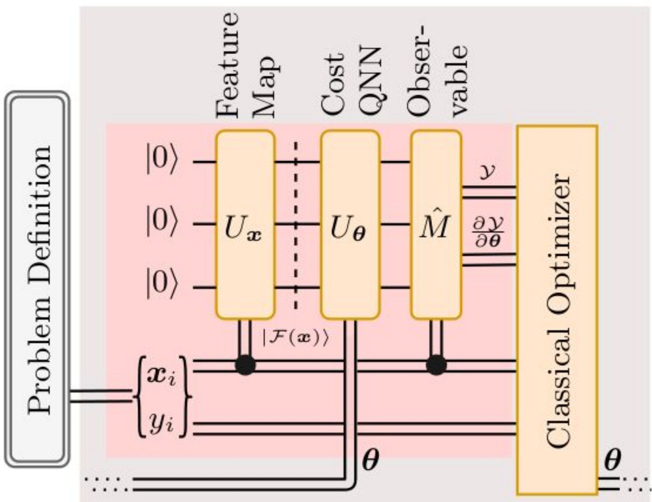


FIG. 2. One epoch of Step I is shown in the gray rectangle. The part of the circuit in the inner pink box is executed for each element of the training data $(\boldsymbol{x}_i, y_i)$ and the results are used by a classical optimizer to suggest $\boldsymbol{\theta}$ for the next epoch.

Quantum Extremal Learning

Modeling

Step 1: Modeling

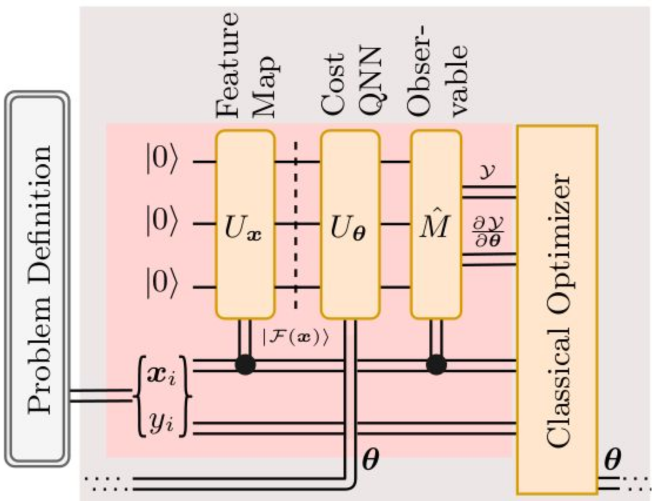


FIG. 2. One epoch of Step I is shown in the gray rectangle. The part of the circuit in the inner pink box is executed for each element of the training data $(\boldsymbol{x}_i, y_i)$ and the results are used by a classical optimizer to suggest $\boldsymbol{\theta}$ for the next epoch.

Step 1.1: Quantum feature map

$$\boldsymbol{x} \mapsto |\mathcal{F}(\boldsymbol{x})\rangle$$

$$U_{\boldsymbol{x}} |0\rangle^{\otimes N} := |\mathcal{F}(\boldsymbol{x})\rangle$$

Quantum Extremal Learning

Modeling

Step 1: Modeling

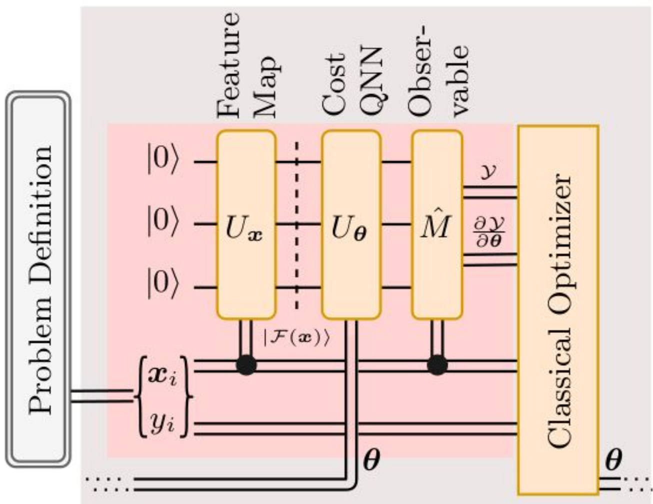


FIG. 2. One epoch of Step I is shown in the gray rectangle. The part of the circuit in the inner pink box is executed for each element of the training data $(\boldsymbol{x}_i, y_i)$ and the results are used by a classical optimizer to suggest $\boldsymbol{\theta}$ for the next epoch.

Step 1.1: Quantum feature map

$$\boldsymbol{x} \mapsto |\mathcal{F}(\boldsymbol{x})\rangle$$

$$U_{\boldsymbol{x}} |0\rangle^{\otimes N} := |\mathcal{F}(\boldsymbol{x})\rangle$$

Step 1.2: Apply a variational form / cost-QNN

$$U_{\boldsymbol{\theta}} |\mathcal{F}(\cdot)\rangle$$

Quantum Extremal Learning

Modeling

Step 1: Modeling

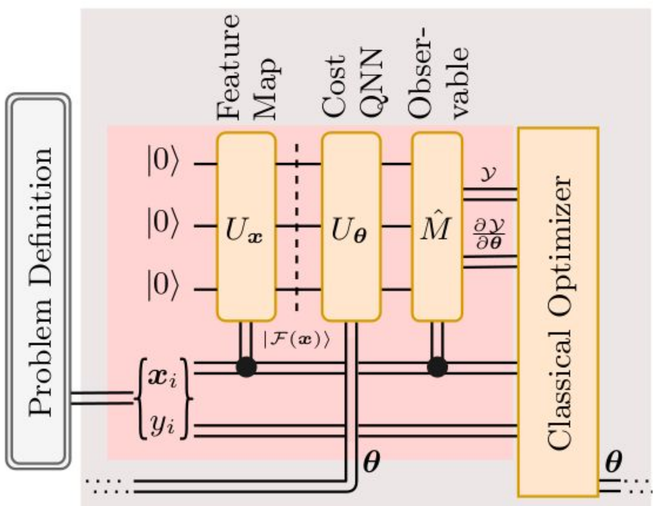


FIG. 2. One epoch of Step I is shown in the gray rectangle. The part of the circuit in the inner pink box is executed for each element of the training data $(\boldsymbol{x}_i, y_i)$ and the results are used by a classical optimizer to suggest $\boldsymbol{\theta}$ for the next epoch.

Step 1.1: Quantum feature map

$$\boldsymbol{x} \mapsto |\mathcal{F}(\boldsymbol{x})\rangle$$

$$U_{\boldsymbol{x}} |0\rangle^{\otimes N} := |\mathcal{F}(\boldsymbol{x})\rangle$$

Step 1.2: Apply a variational form / cost-QNN

$$U_{\boldsymbol{\theta}}|\mathcal{F}(\cdot)\rangle$$

Step 1.3: Measure the expectation value of an observable

$$\langle \mathcal{F}(\cdot)|U_{\boldsymbol{\theta}}^{\dagger} \hat{M} U_{\boldsymbol{\theta}}|\mathcal{F}(\cdot)\rangle : X \to \mathbb{R}$$
$$\boldsymbol{x} \mapsto \mathcal{Y}.$$

PASQAL

Quantum Extremal Learning

Modeling

Step 1: Modeling

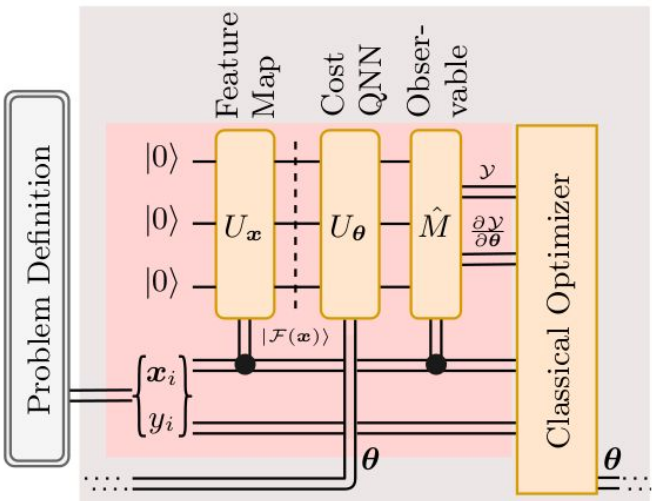


FIG. 2. One epoch of Step I is shown in the gray rectangle. The part of the circuit in the inner pink box is executed for each element of the training data $(\boldsymbol{x}_i, y_i)$ and the results are used by a classical optimizer to suggest $\boldsymbol{\theta}$ for the next epoch.

Step 1.4: Compute loss function

$$\boldsymbol{\theta} \mapsto L\Big(\boldsymbol{\theta}; \{\, (\boldsymbol{x}_i, y_i)\,\}\Big) \in \mathbb{R}$$

Quantum Extremal Learning

Modeling

Step 1: Modeling

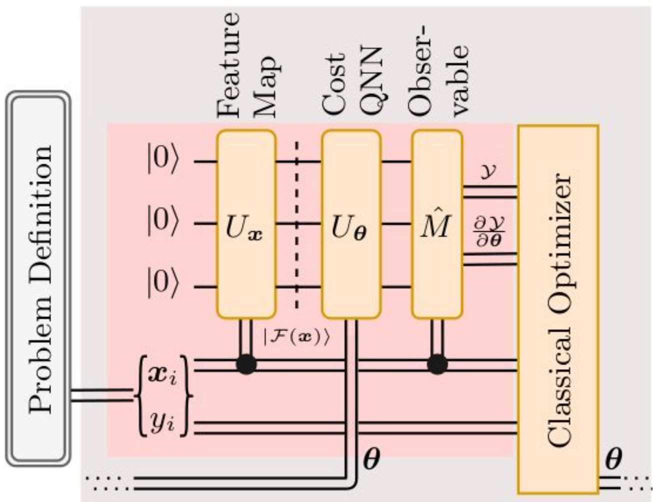


FIG. 2. One epoch of Step I is shown in the gray rectangle. The part of the circuit in the inner pink box is executed for each element of the training data $(\boldsymbol{x}_i, y_i)$ and the results are used by a classical optimizer to suggest $\boldsymbol{\theta}$ for the next epoch.

Step 1.4: Compute loss function

$$\boldsymbol{\theta} \mapsto L\Big(\boldsymbol{\theta}; \{\ (\boldsymbol{x}_i, y_i)\ \}\Big) \in \mathbb{R}$$

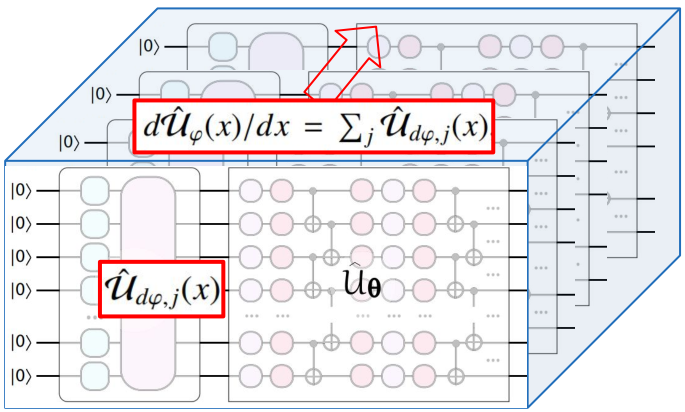
Step 1.5: Use a classical optimizer to up-date cost-QNN parameters up to conver-gence:

$$\langle \mathcal{F}(\boldsymbol{x})| U_{\boldsymbol{\theta}}^{\dagger} \hat{M} U_{\boldsymbol{\theta}} |\mathcal{F}(\boldsymbol{x})\rangle \approx f(\boldsymbol{x})$$

Quantum Extremal Learning

Extremizing

Step 2.1: Apply circuit differentiation to the feature map circuit



Step 2: Extremizing

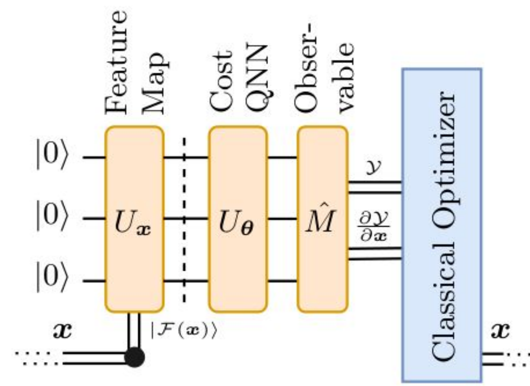
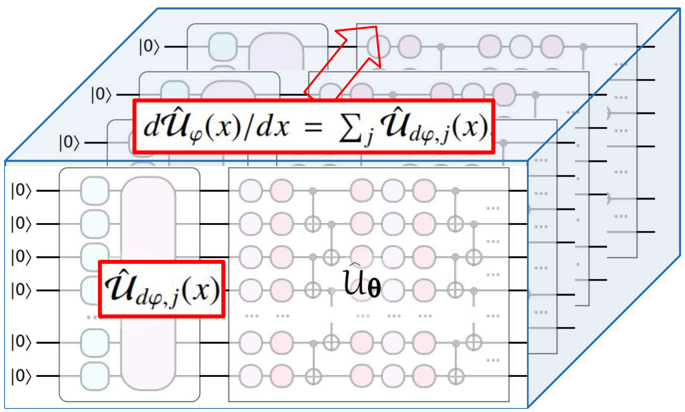


FIG. 3. Step II in the continuous case. The part of the circuit retained from Step I is shown in orange and the new elements are shown in blue. Notice that $\boldsymbol{\theta}$ is no longer updated; the optimization is performed on $\boldsymbol{x}$.

Quantum Extremal Learning

Extremizing

Step 2.1: Apply circuit differentiation to the feature map circuit



Step 2.2: Use gradient descent / ascent to extremize circuit output

Step 2: Extremizing

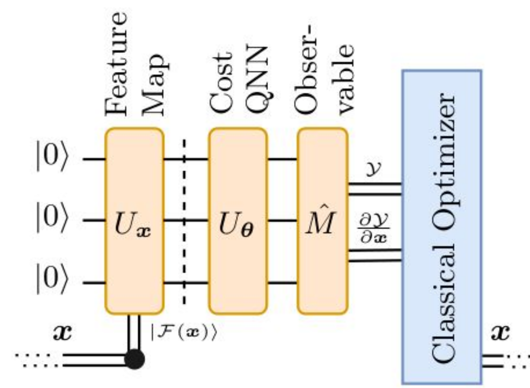
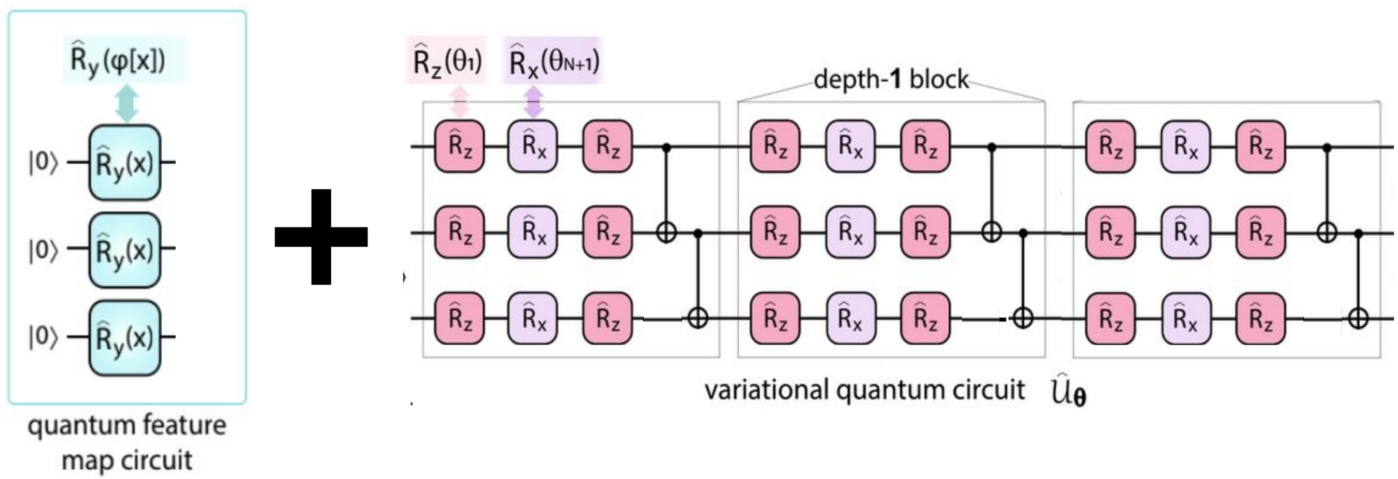


FIG. 3. Step II in the continuous case. The part of the circuit retained from Step I is shown in orange and the new elements are shown in blue. Notice that $\boldsymbol{\theta}$ is no longer updated; the optimization is performed on $\boldsymbol{x}$.

Quantum Extremal Learning

Example use

- **Function:** $f(x) = \sin(5x)$ in the range $[0, 1]$
- **Quantum feature map:** Chebyshev tower-map
- **Cost-QNN:** Hardware Efficient Ansatz
- **Measurement basis:** Z-basis
- **Loss function:** MSE
- **Optimizer:** Adam

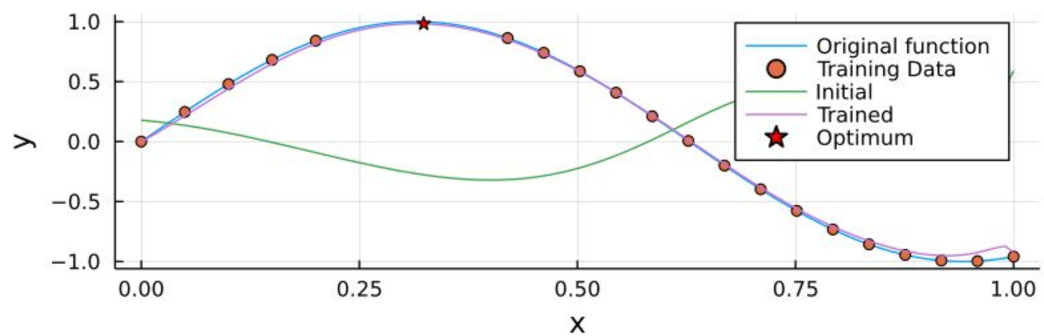


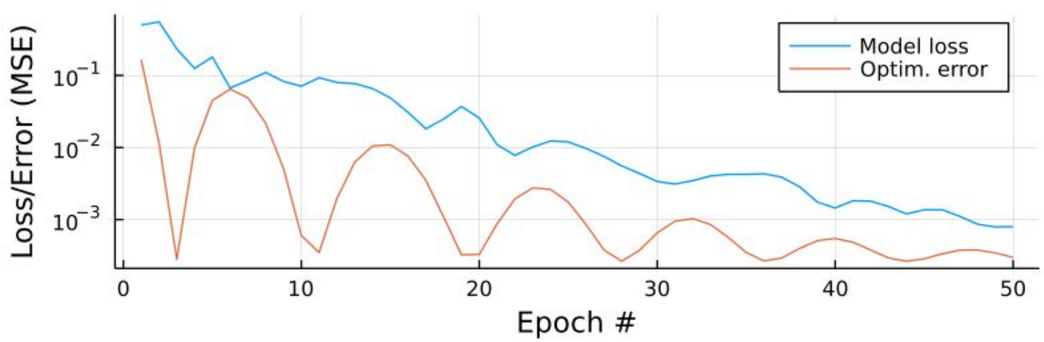
$$\hat{\mathcal{U}}_{\varphi}(x) = \bigotimes_{j=1}^{N} \hat{R}_{y,j}(2n[j] \arccos x)$$

Quantum Extremal Learning

Example use

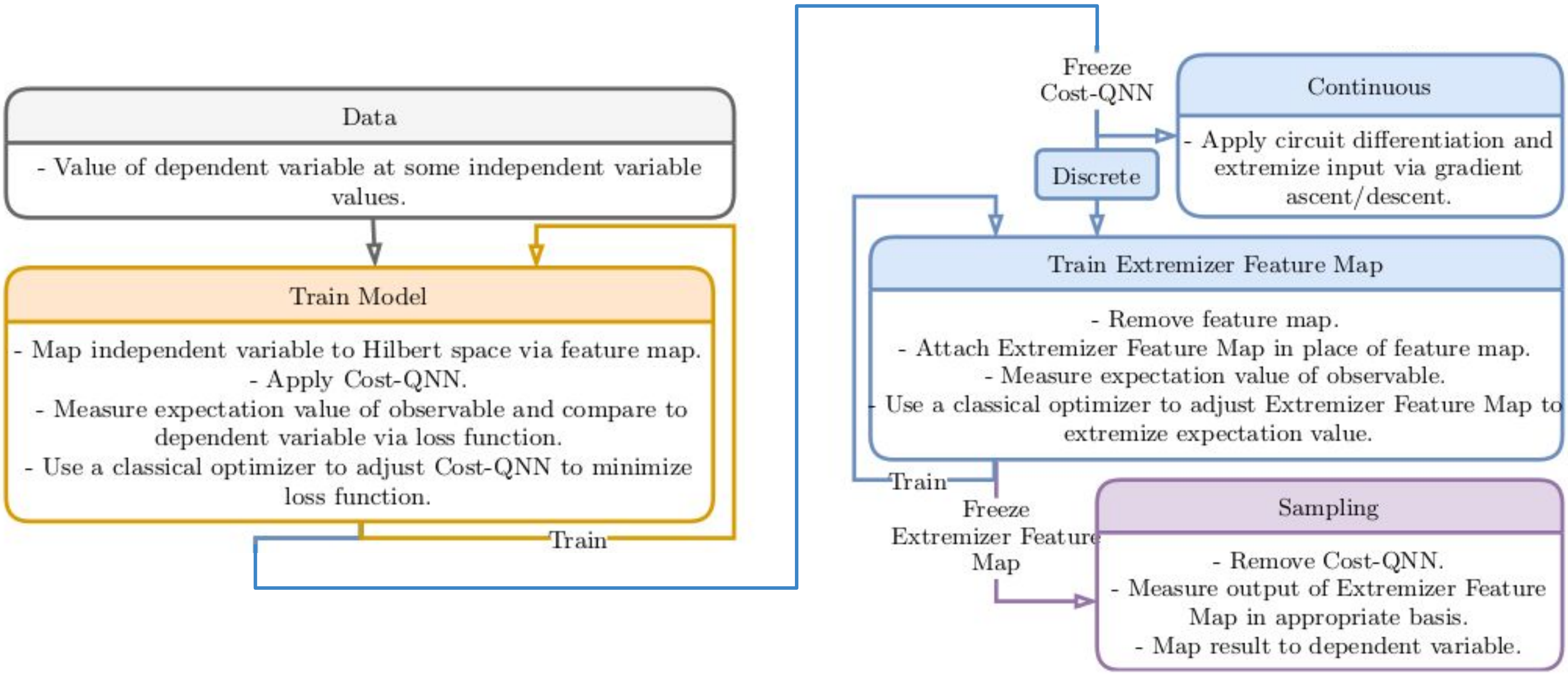
- **Function:** $f(x) = \sin(5x)$ in the range $[0, 1]$
- **Quantum feature map:** Chebyshev tower-map
- **Cost-QNN:** Hardware Efficient Ansatz
- **Measurement basis:** Z-basis
- **Loss function:** MSE
- **Optimizer:** Adam





Quantum Extremal Learning

Algorithm workflow (summary)



Conclusion

- ➔ Extremal Learning is compatible with a wide variety of optimization problem types
- ➔ Extremal Learning is the process of finding the input to a hidden function that extremizes the function output, without having direct access to the hidden function
- ➔ Quantum Extremal Learning (QEL) has been proposed has a quantum generalization of Extremal Learning and is inspired from Quantum Circuit Learning techniques.
- ➔ QEL is a two-step process that combines a quantum feature map with a cost quantum neural network. The first step is to be optimize the cost-QNN, the second step before is to extremize the circuit output by differentiating the feature map.
- ➔ QEL can be generalized to discrete inputs, extending its domain of applications