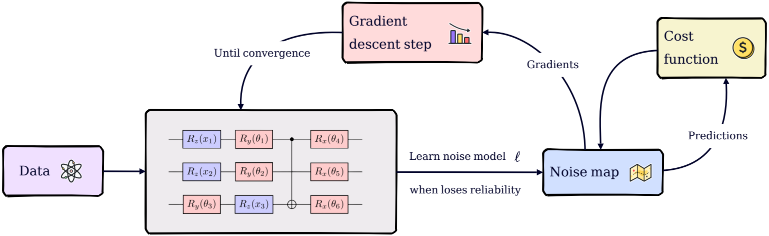
RTQEM pipeline

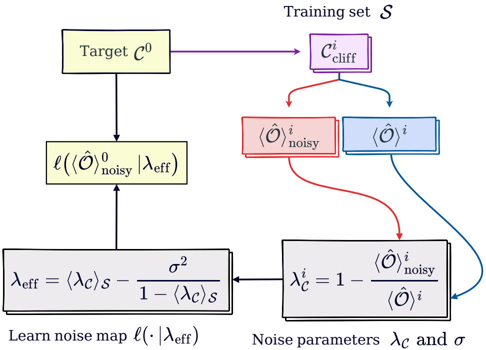
We define a Real-Time Quantum Error Mitigation (RTQEM) procedure.



1. consider a Variational Quantum Algorithm trained with gradient descent;
2. learn the noise map ℓ every time is needed over the procedure;
3. use ℓ to clean up both predictions and gradients.

Learning the noise model   arXiv:2112.06255

We use the Importance Clifford Sampling (ICS) procedure to learn the noise map ℓ.

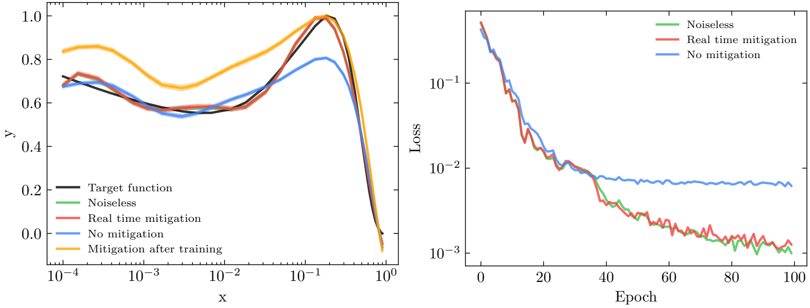


1. sample a training set of Clifford circuits $\mathcal{S}$ on top of a target $\mathcal{C}^0$;
2. process them so that their expectation values on Pauli strings is $+1$ or -1;
3. extract mitigation parameter λ_{eff} comparing $\langle\hat{\mathcal{O}}\rangle_{\text{noisy}}$ and $\langle\hat{\mathcal{O}}\rangle$;
4. build ℓ following the Phenomenological-Error-Model Inspired (PEMI) protocol:

$$\ell(\langle\hat{\mathcal{O}}\rangle|a,\sigma_a) = \frac{1-a}{[(1-a)^2+\sigma_a^2]}\langle\hat{\mathcal{O}}\rangle_{\text{noisy}}$$

One dimensional HEP target: the u-quark PDF

Parameter	N_{train}	N_{params}	N_{shots}	$\mathrm{MSE}^{\mathrm{rtqem}}_{\mathrm{best}}$	$\mathrm{MSE}^{\mathrm{unmit}}_{\mathrm{best}}$	Noise
Value	30	16	10^4	$1.1 \cdot 10^{-3}$	$6.1 \cdot 10^{-3}$	local Pauli

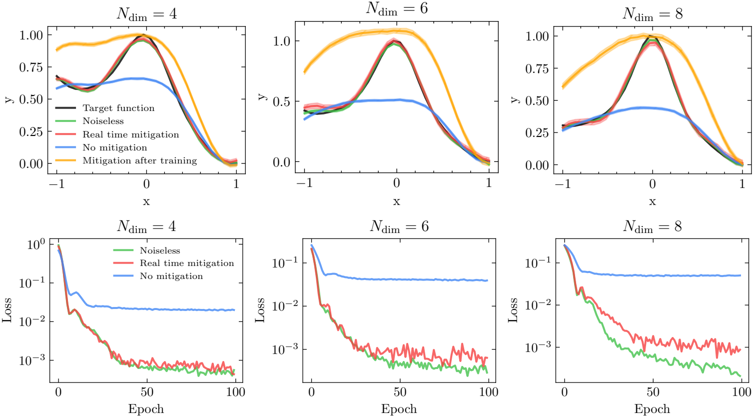


1. thanks to the RTQEM procedure, we reach a good minimum of the cost function;
2. the QEM is not effective is applied to a corrupted scenario (orange curve).

Multidimensional target

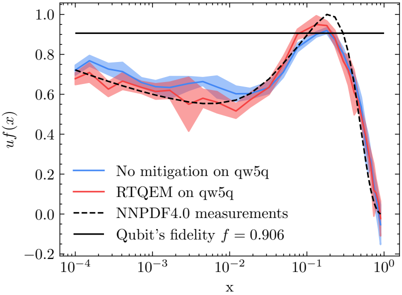
We tackle a multi-dimensional target computing predictions as expected value of a $Z^{\otimes N_{\text{dim}}}$ after executing an N_{dim} circuit.

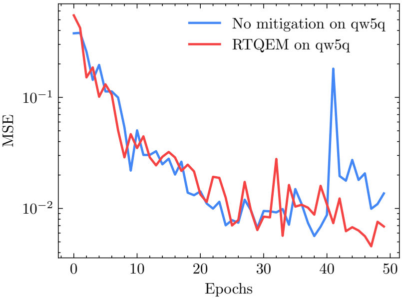
Job ID	N_{train}	N_{params}	N_{shots}	$\text{MSE}^{\text{rtqem}}_{\text{best}}$	$\text{MSE}^{\text{unmit}}_{\text{best}}$	Noise
$N_{\text{dim}} = 4$	30	48	10^4	$4.4 \cdot 10^{-4}$	$1.9 \cdot 10^{-2}$	local Pauli
$N_{\text{dim}} = 6$	30	72	10^4	$4.1 \cdot 10^{-4}$	$3.8 \cdot 10^{-2}$	local Pauli
$N_{\text{dim}} = 8$	30	96	10^4	$5.6 \cdot 10^{-4}$	$4.8 \cdot 10^{-2}$	local Pauli



RTQEM on a superconducting qubit

Parameter	N_{train}	N_{params}	N_{shots}	Noise
Value	15	16	500	real noise



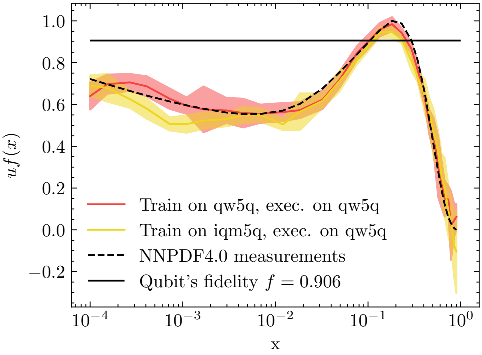


Can RTQEM generalise?

We compare two runs with the same initial conditions and $N_{\text{epochs}} = 100$ but performed with two different devices:

- `iqm5q` by IQM controlled using Zurich Instruments;
- `qw5q` by QuantWare controlled using Qblox.

We train the two devices but perform the prediction on the same chip (`qw5q`).



The $\boldsymbol{\theta}_{\text{best}}$ get on `iqm5q` perform well on `qw5q` because RTQEM clean the noise!