The Calo4pQVAE: Challenges and opportunities

3/20/24 :: D-wave HQ

J. Quetzalcoatl Toledo-Marin
Quantum Machine Learning Research Associate

TRIUMF Canada's particle accelerator centre
Centre canadien d'accélération des particules

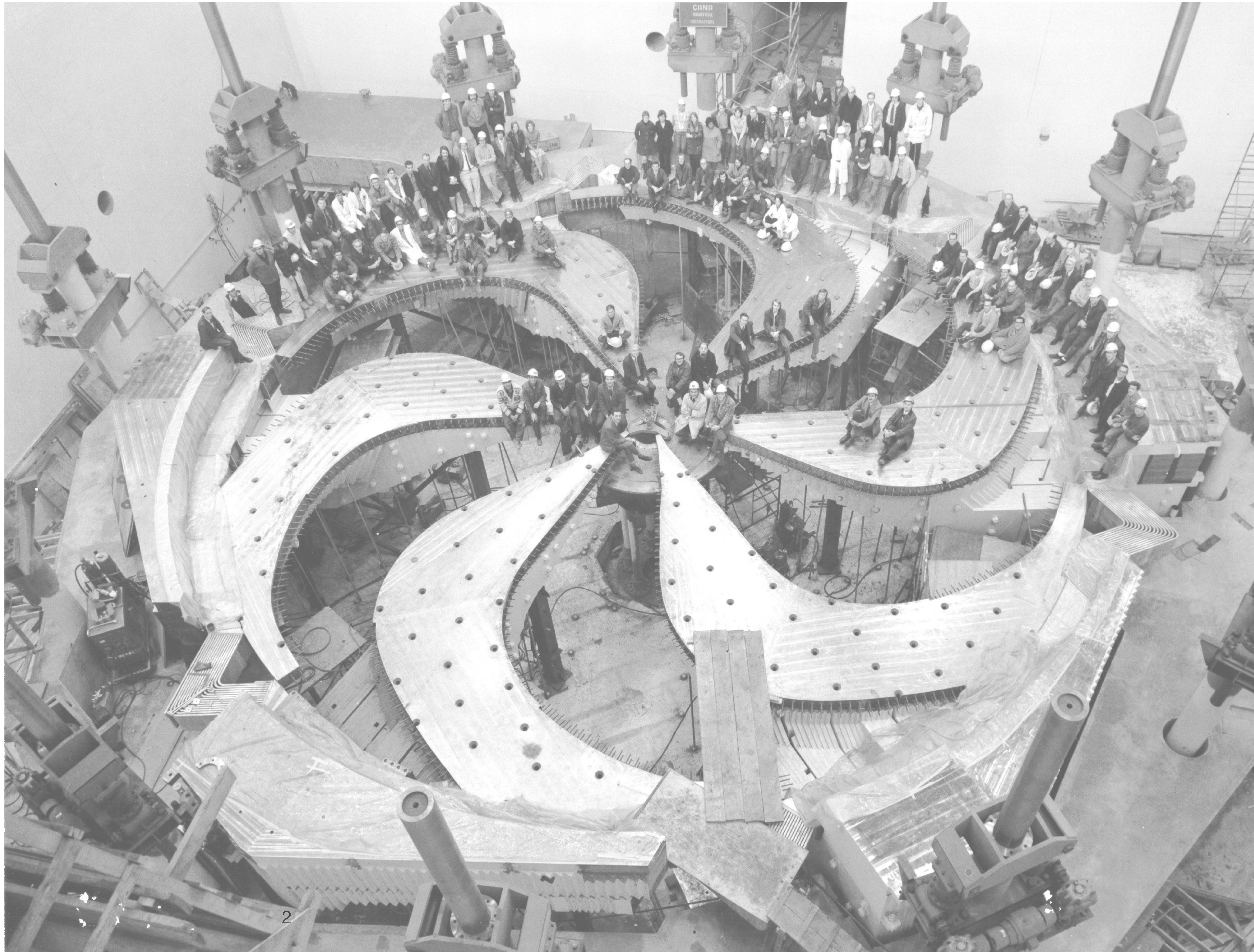
Acknowledgements

Team:

- Sebastian Gonzalez @ UBC
- Hao Jia @ UBC
- Sehmimul Hoque @ Waterloo University
- Abhishek Abhishek @ UBC
- Tiago Vale @ Simon Fraser Uni
- Soren Andersen @ Lund University
- Eric Paquet @ NRC
- Roger Melko @ Perimeter Institute
- Geoffrey Fox @ University of Virginia
- Max Swiatlowski @ TRIUMF
- Wojtek Fedorko @ TRIUMF

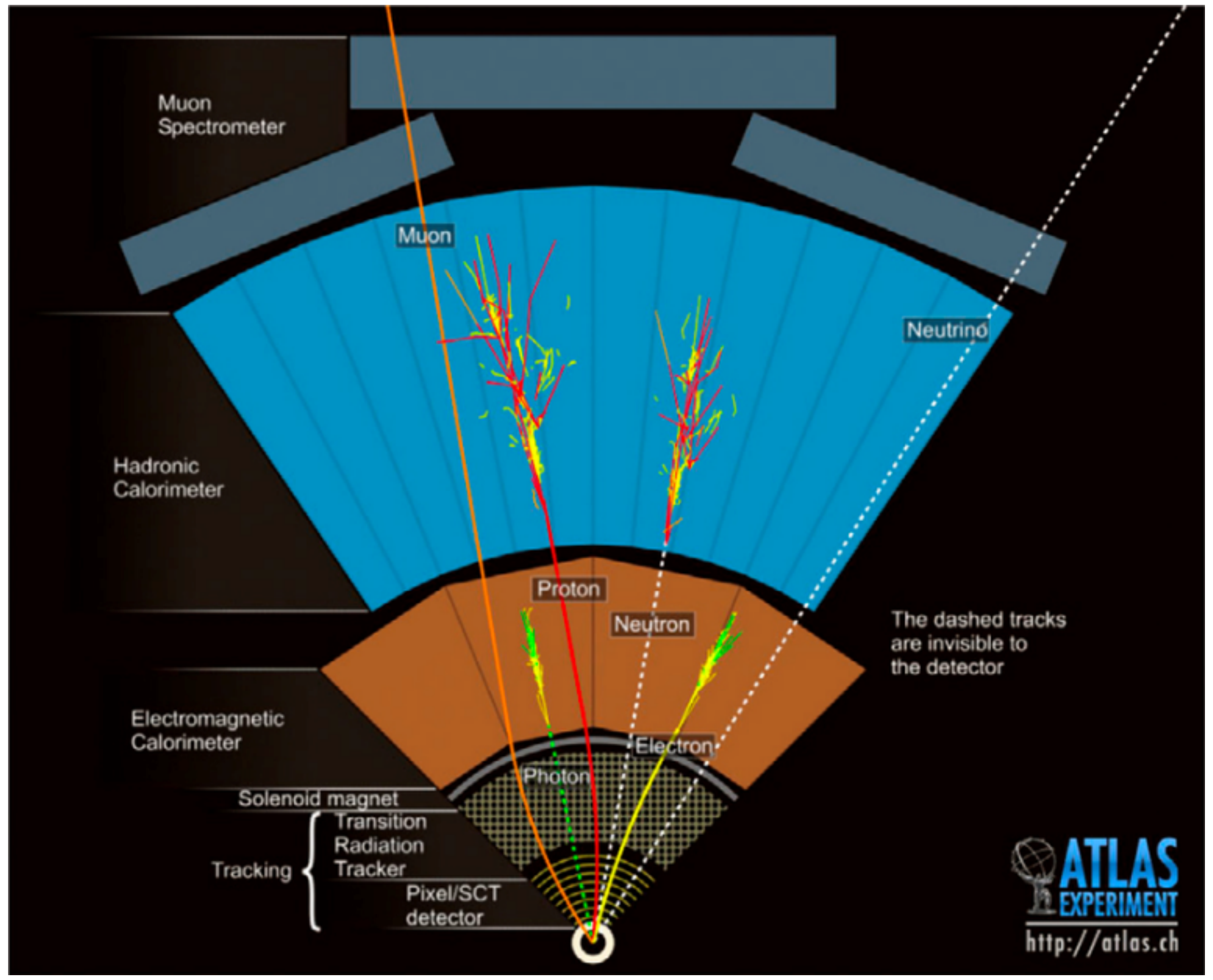
jtoledo@TRIUMF.ca

arXiv preprint arXiv:2312.03179 (2023).
arXiv preprint arXiv:2210.07430 (2022). NeurIPS 2021
Current work to be submitted



Motivation

- As we approach the launch of the High Luminosity Large Hadron Collider (HL-LHC) by the decade's end, the computational demands of traditional collision simulations have become untenably high.
- Current methods, relying heavily on Monte Carlo simulations for event showers in calorimeters, are projected to require millions of CPU-years annually, a demand far beyond current capabilities.
- This bottleneck presents a unique opportunity for breakthroughs in computational physics through the integration of generative AI with quantum computing technologies.



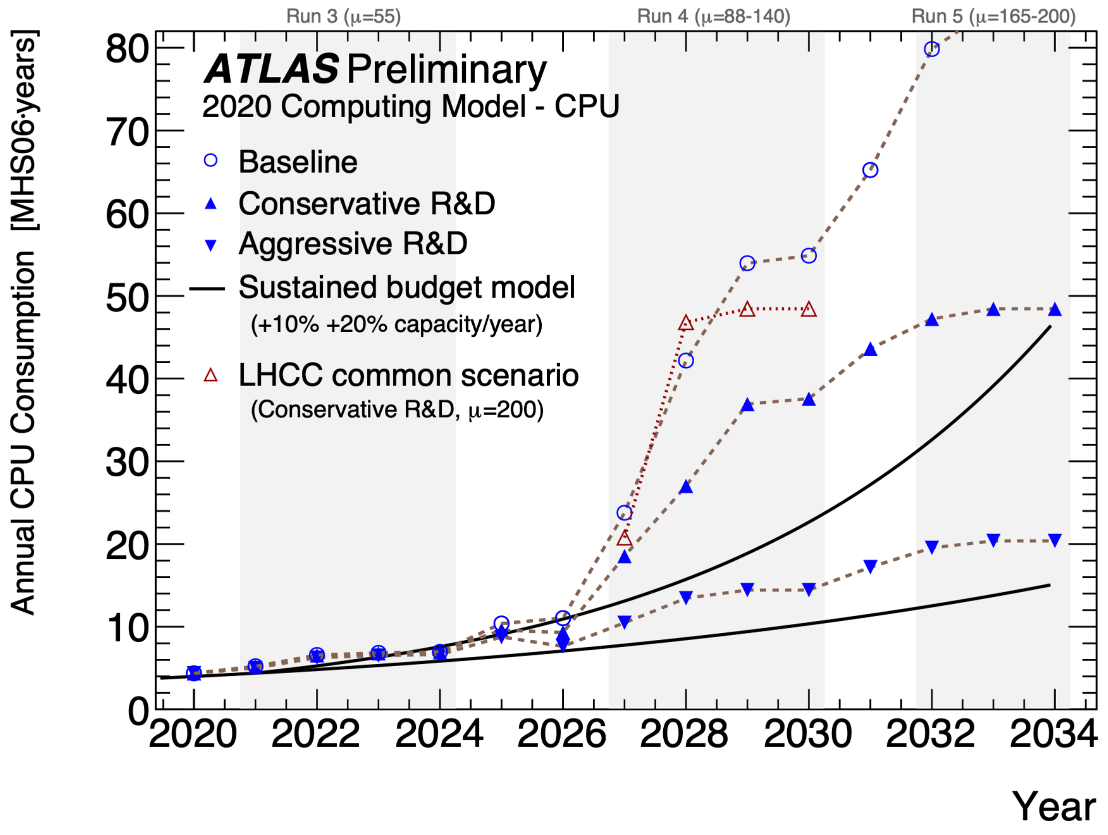
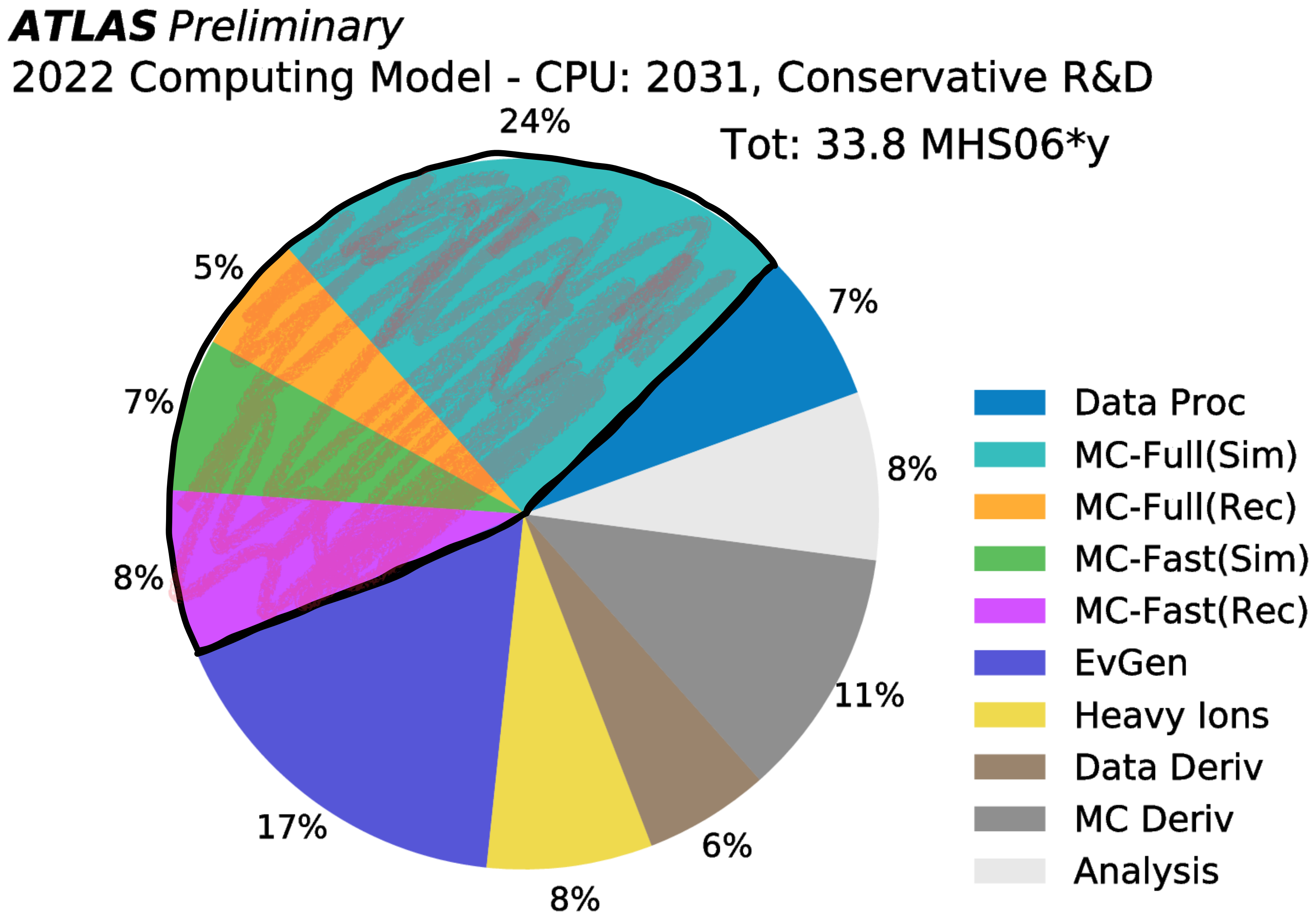


Figure 1. Projected CPU requirements of ATLAS experiment between 2020 and 2034 based on 2020 assessment. Three scenarios are shown, corresponding to an ambitious ("aggressive"), modest ("conservative") and minimal ("baseline") development program. The black lines indicate annual improvements of 10% and 20% in the computational capacity of new hardware for a given cost, assuming a sustained level of annual investment. The blue dots with the brown lines represent the 3 ATLAS scenarios following the present LHC schedule. The red triangles indicate the Conservative R&D scenario under an assumption of the LHC reaching in average 200 primary vertexes per one bunch crossing (μ) in Run4 (2028-2030).



Scientific Data Lake for High Luminosity LHC project and other data-intensive particle and astro-particle physics experiments. InJournal of Physics: Conference Series 2020 Dec 1 (Vol. 1690, No. 1, p. 012166). IOP Publishing.

Generative Models

Simplest Example: Box-Muller Method

$$\int_0^1 dU_1 Uni(U_1) \int_0^1 dU_2 Uni(U_2) = \int_{-\infty}^{\infty} dZ_1 \mathcal{N}(Z_1 | 0,1) \int_{-\infty}^{\infty} dZ_2 \mathcal{N}(Z_2 | 0,1) = 1$$

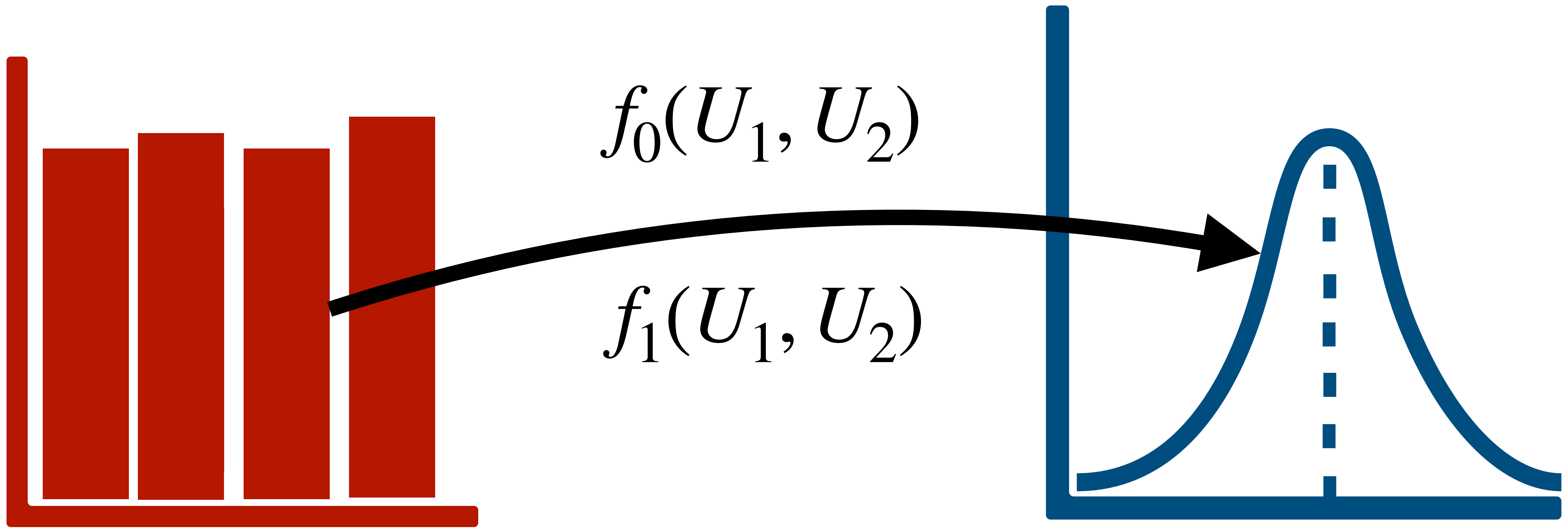
$$\int_0^{u_1} dU_1 Uni(U_1) \int_0^{u_2} dU_2 Uni(U_2) = \int_a^b \int_c^d dZ_0 dZ_1 \underbrace{\left| \frac{\partial(U_1, U_2)}{\partial(Z_0, Z_1)} \right| Uni(U_1(Z_0, Z_1)) Uni(U_2(Z_0, Z_1))}_{\mathcal{N}(Z_0 | 0,1) \mathcal{N}(Z_1 | 0,1)}$$

1. Generate two **uniformly** independent, identically distributed random numbers U_1 and U_2.

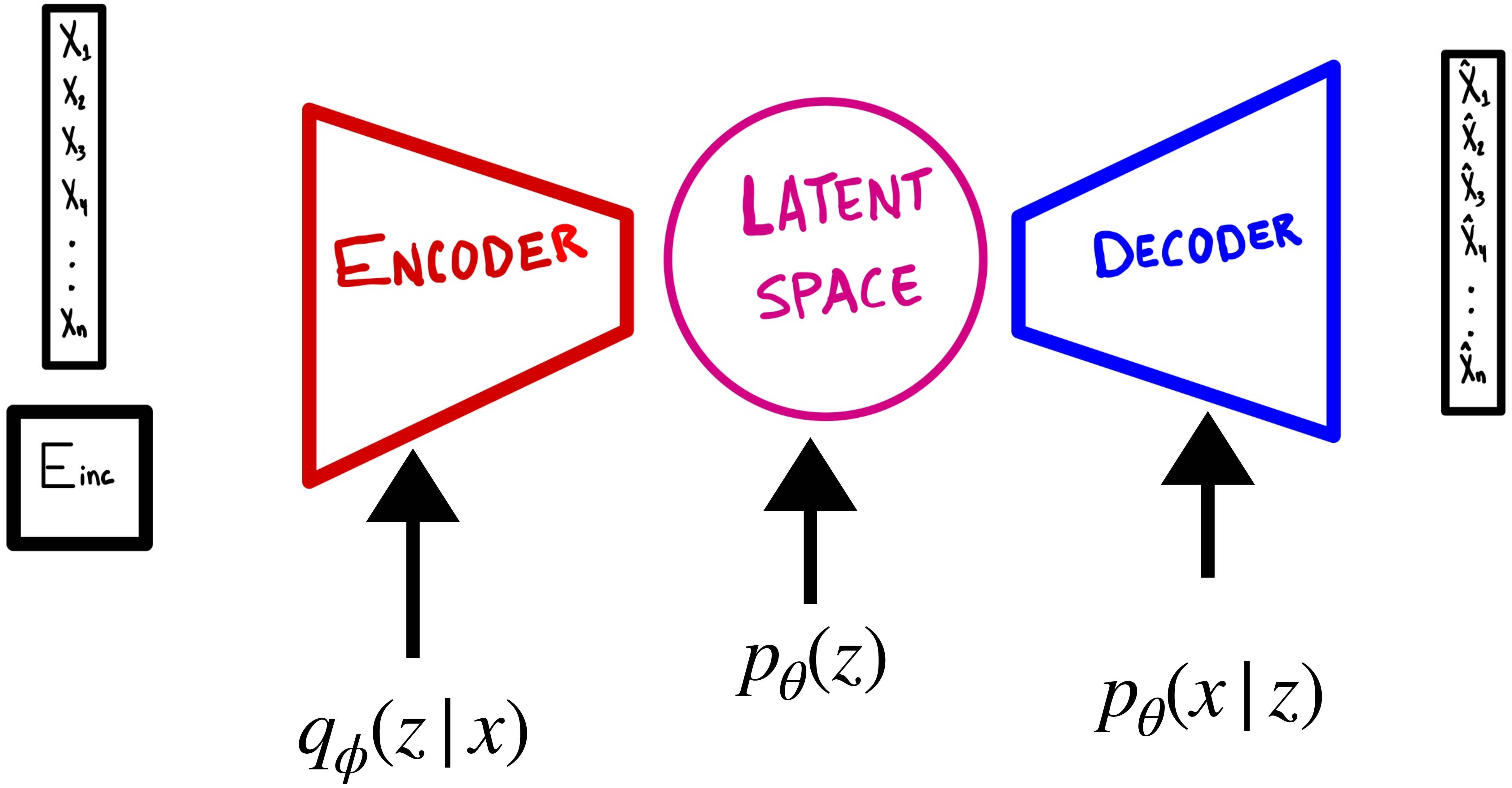
2. Substitute in:

$$Z_0 = f_0(U_1, U_2) = \sqrt{-2 \ln U_1} \cos(2\pi U_2)$$

$$Z_1 = f_1(U_1, U_2) = \sqrt{-2 \ln U_1} \sin(2\pi U_2)$$



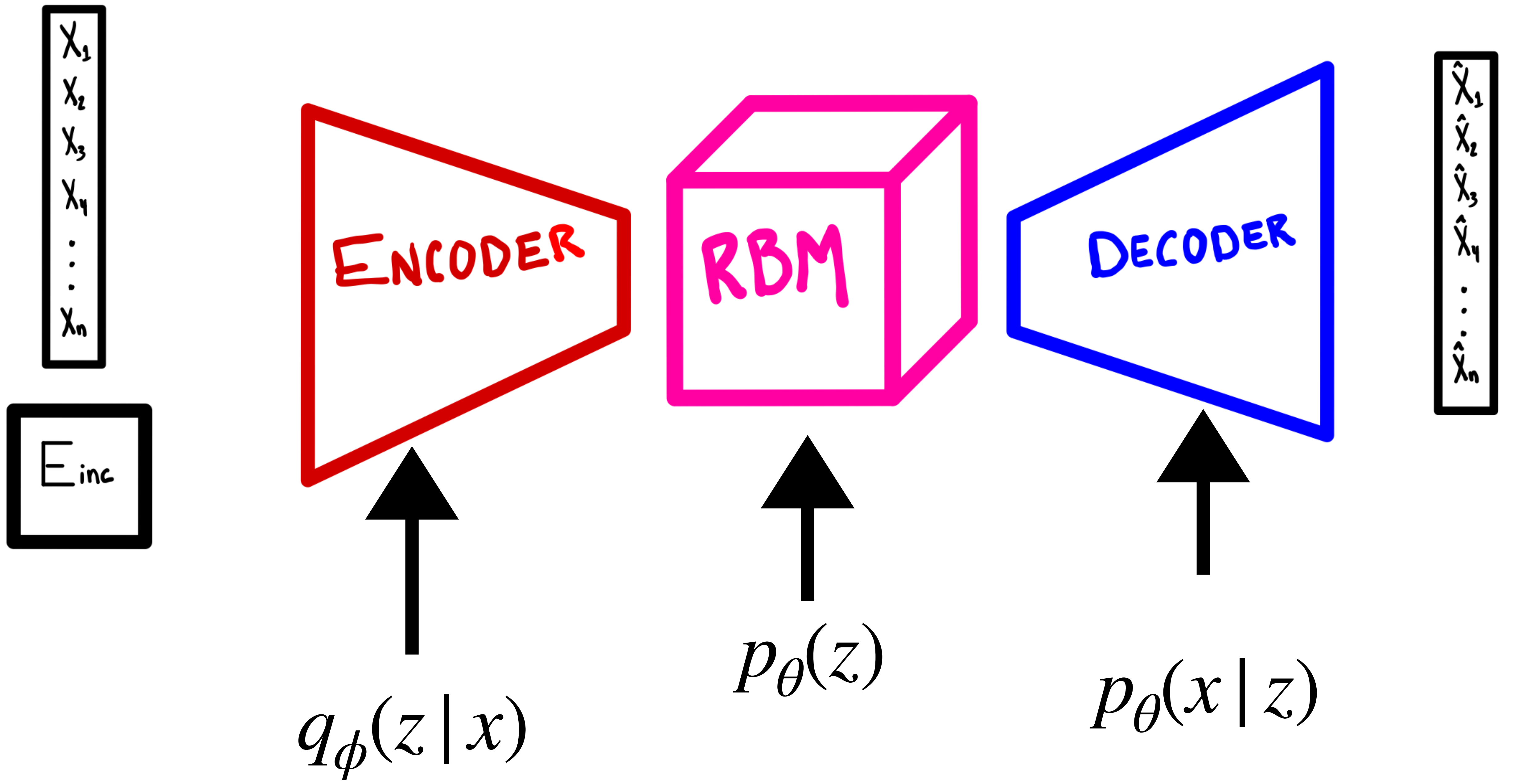
Variational Autoencoders (VAE)



$$\mathcal{L}_{\phi,\theta}(x) = \underbrace{\langle \ln p_\theta(x\,|\,z) \rangle_{q_\phi(z|x)}}_{\text{Reconstruction}} - \underbrace{\left\langle \ln \frac{q_\phi(z\,|\,x)}{p_\theta(z)} \right\rangle_{q_\phi(z|x)}}_{\text{Regularizer}}$$

VAE + Restricted Boltzmann Machine

Why?

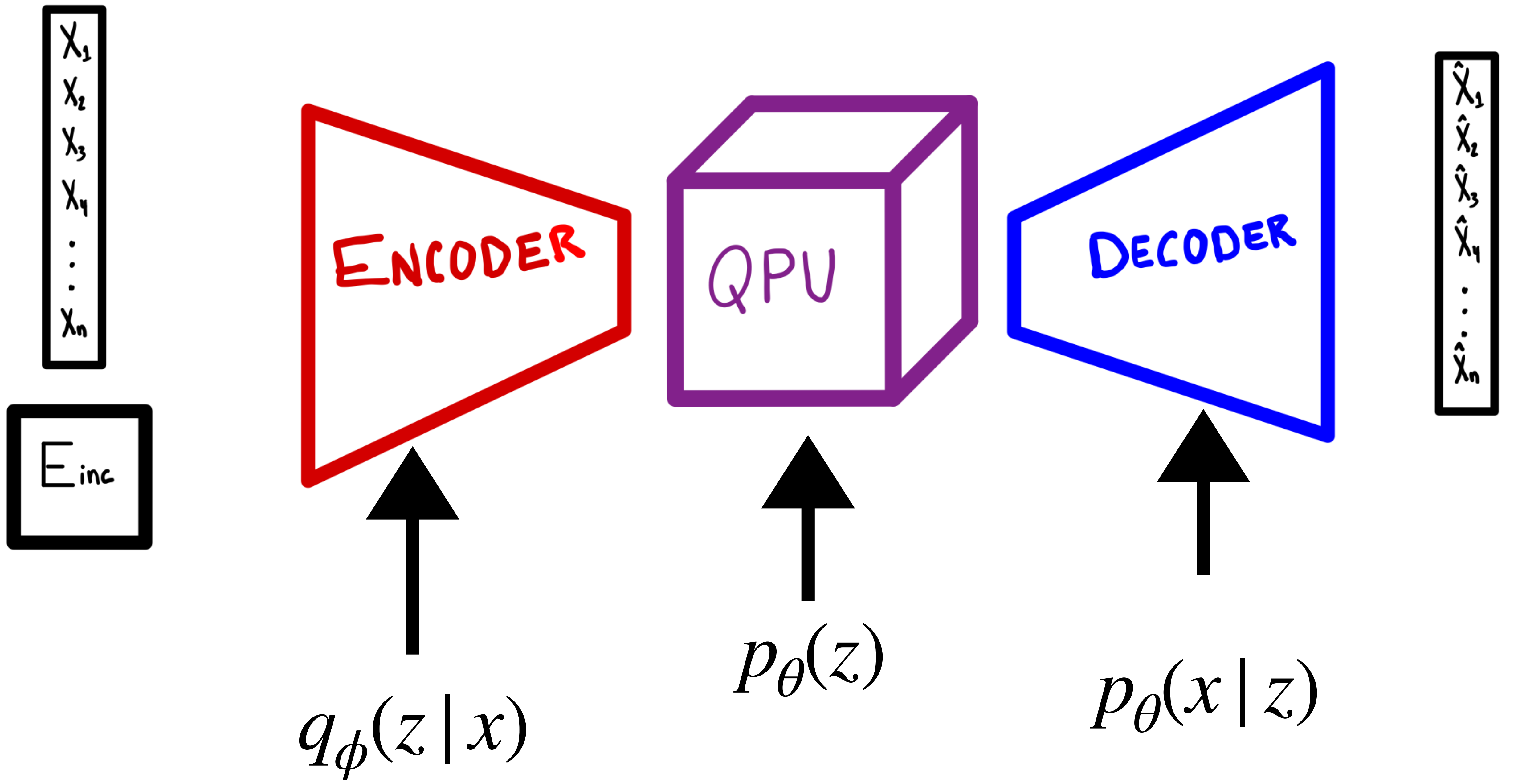


- More expressiveness
- However, this comes at a cost.

$$\mathscr{L}_{\phi,\theta}(x) = \underbrace{\langle \ln p_\theta(x\,|\,z) \rangle_{q_\phi(z|x)}}_{\text{Reconstruction}} - \underbrace{\left\langle \ln \frac{q_\phi(z\,|\,x)}{p_\theta(z)} \right\rangle_{q_\phi(z|x)}}_{\text{Regularizer}}$$

Quantum-Assisted Discrete VAE

Why?



- More expressiveness
- However, this comes at a cost.
- But we might be able to avoid Gibbs sampling…

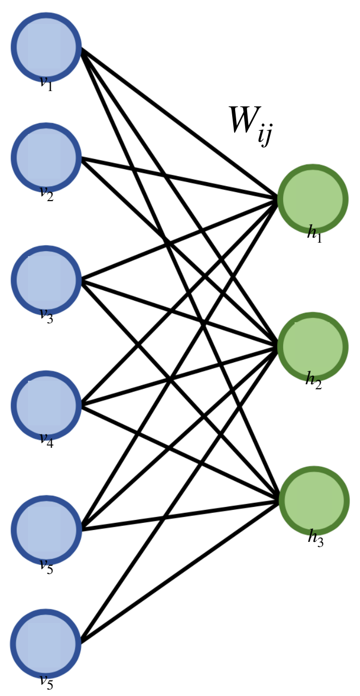
$$\mathscr{L}_{\phi,\theta}(x) = \underbrace{\langle \ln p_\theta(x\,|\,z)\rangle_{q_\phi(z|x)}}_{\text{Reconstruction}} - \underbrace{\left\langle \ln \frac{q_\phi(z\,|\,x)}{p_\theta(z)} \right\rangle_{q_\phi(z|x)}}_{\text{Regularizer}}$$

Quantum Annealer

Topologies

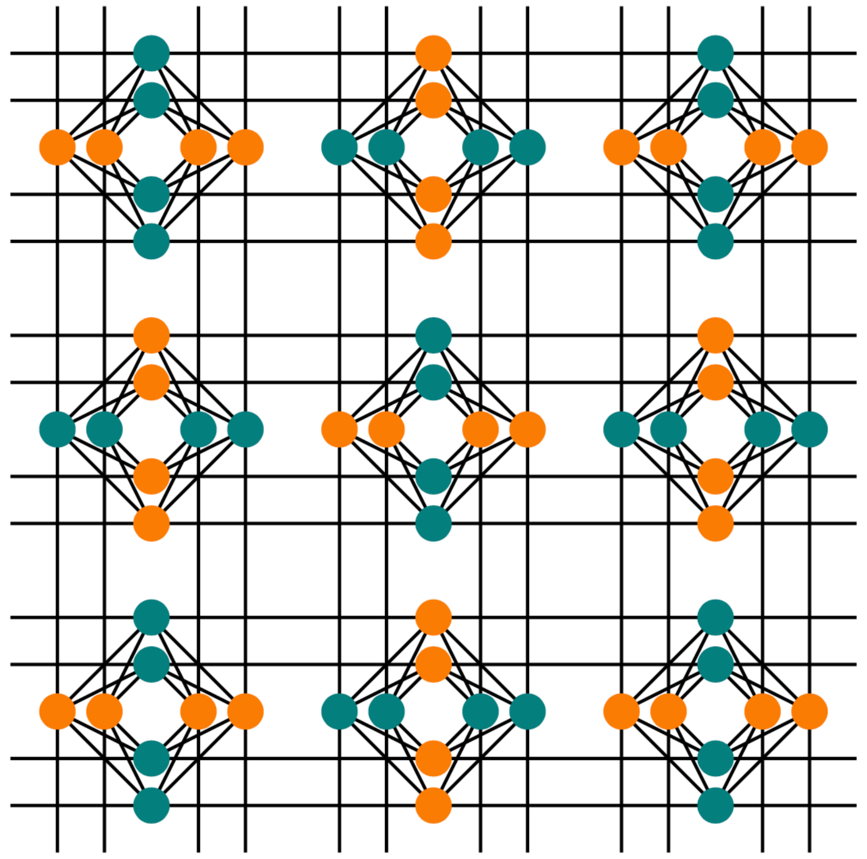
Fully Connected RBM

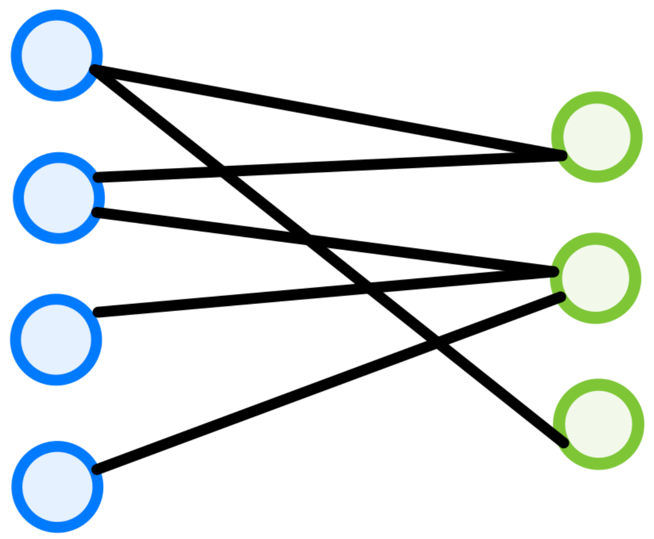
2-partite Graph



Chimera QA

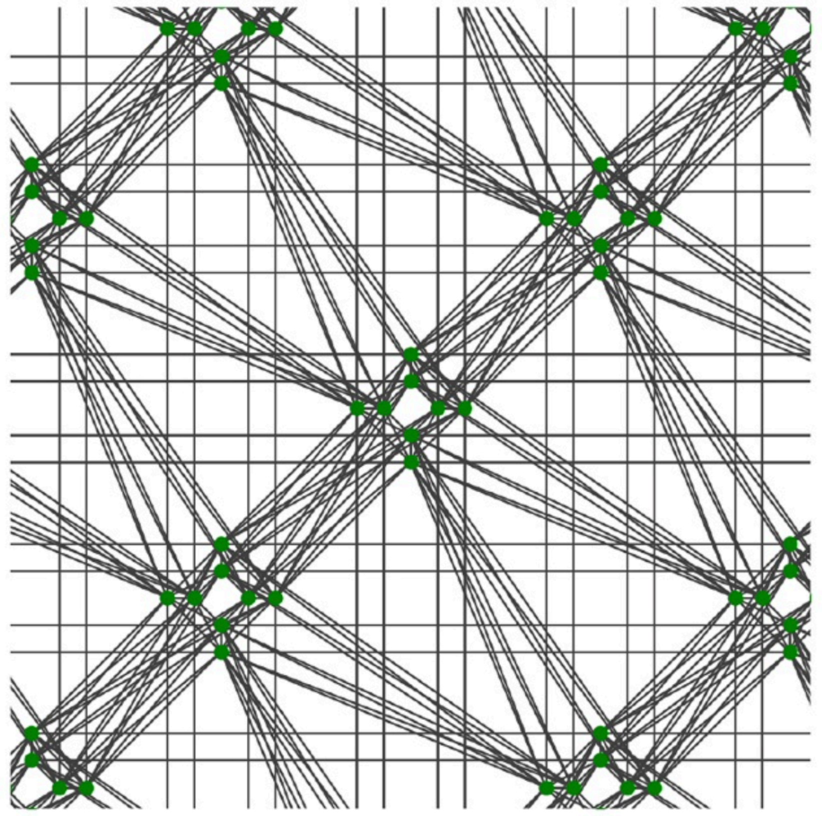
2-partite Graph

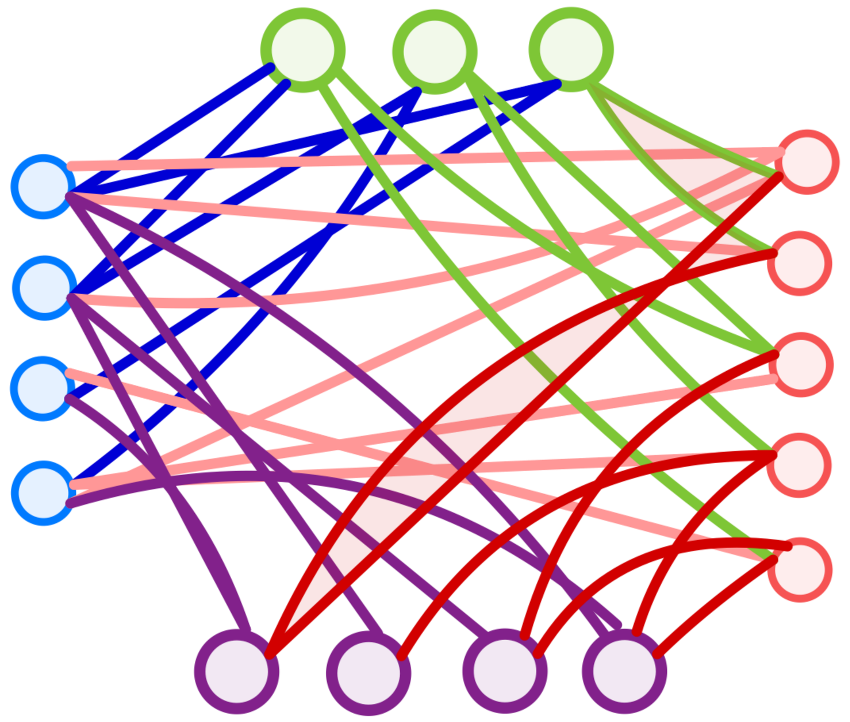




Pegasus QA

4-partite Graph





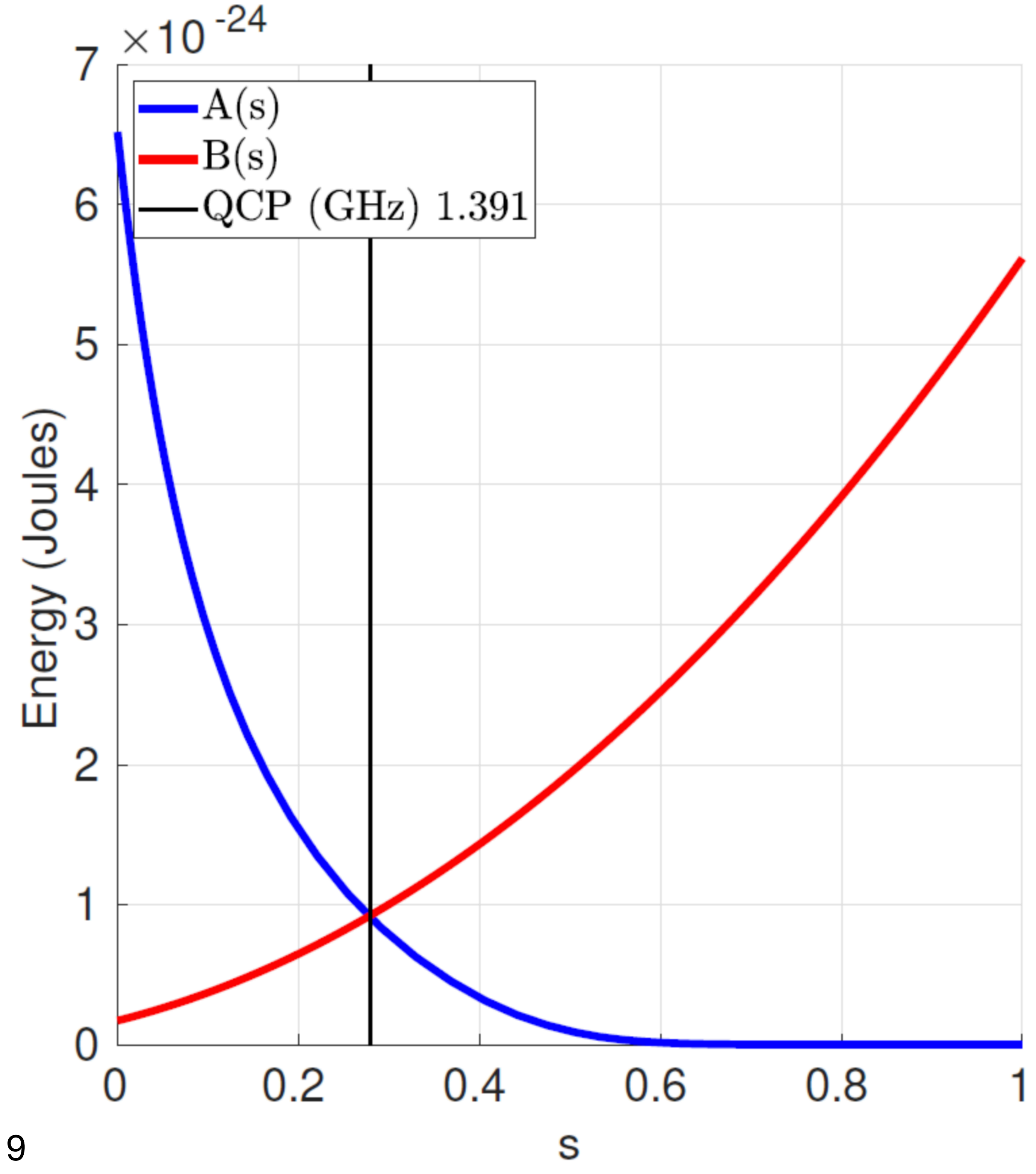
Winci W, Buffoni L, Sadeghi H, Khoshaman A, Andriyash E, Amin MH. A path towards quantum advantage in training deep generative models with quantum annealers. Machine Learning: Science and Technology. 2020 Oct 29;1(4):045028.

Quantum Annealer

Basics

- A QA is an array of superconducting flux quantum bits with programmable spin–spin couplings.
- QA relies on the Adiabatic Approximation.
- The goal is to find the ground state of a Hamiltonian H_0.
- In practice, quantum annealers have a strong interaction with the environment which lead to **thermalization** and **decoherence**. It can also reach a *dynamical arrest*.

$$\mathcal{H}_{ising} = \underbrace{-\frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right)}_{\substack{\text{Initial Hamiltonian} \\ H_1}} + \underbrace{\frac{B(s)}{2}\left(\sum_i c_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\right)}_{\substack{\text{Final Hamiltonian} \\ H_0}}$$



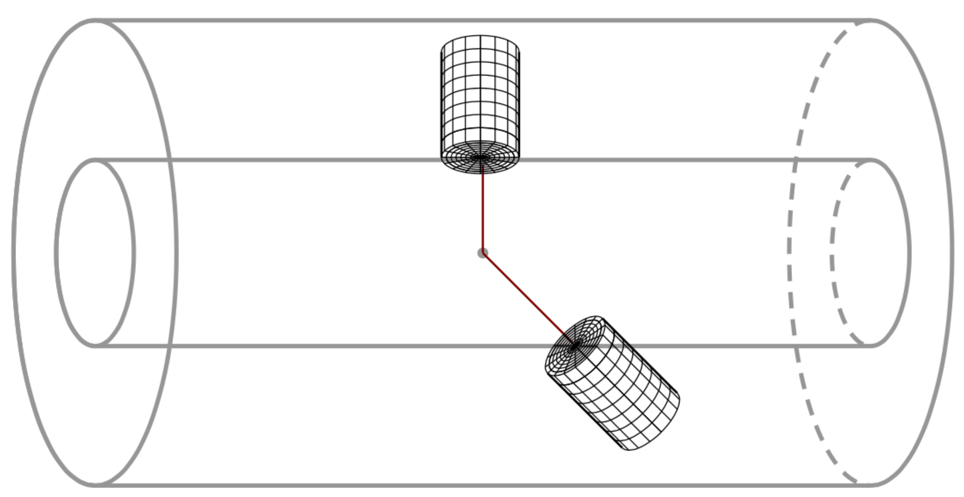
Amin MH. Searching for quantum speedup in quasistatic quantum annealers. Physical Review A. 2015 Nov 19;92(5):052323.

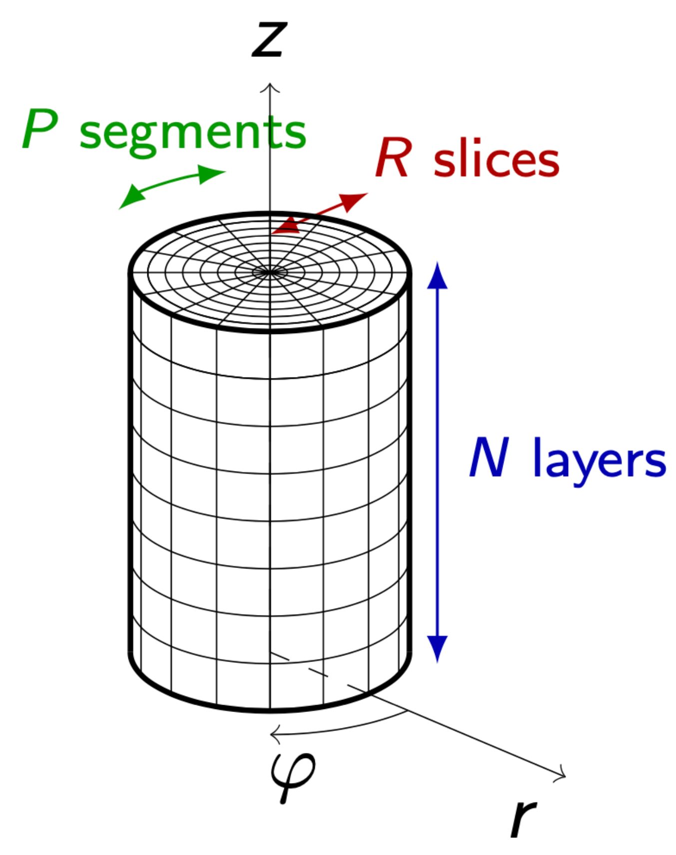
Calo4p-QVAE

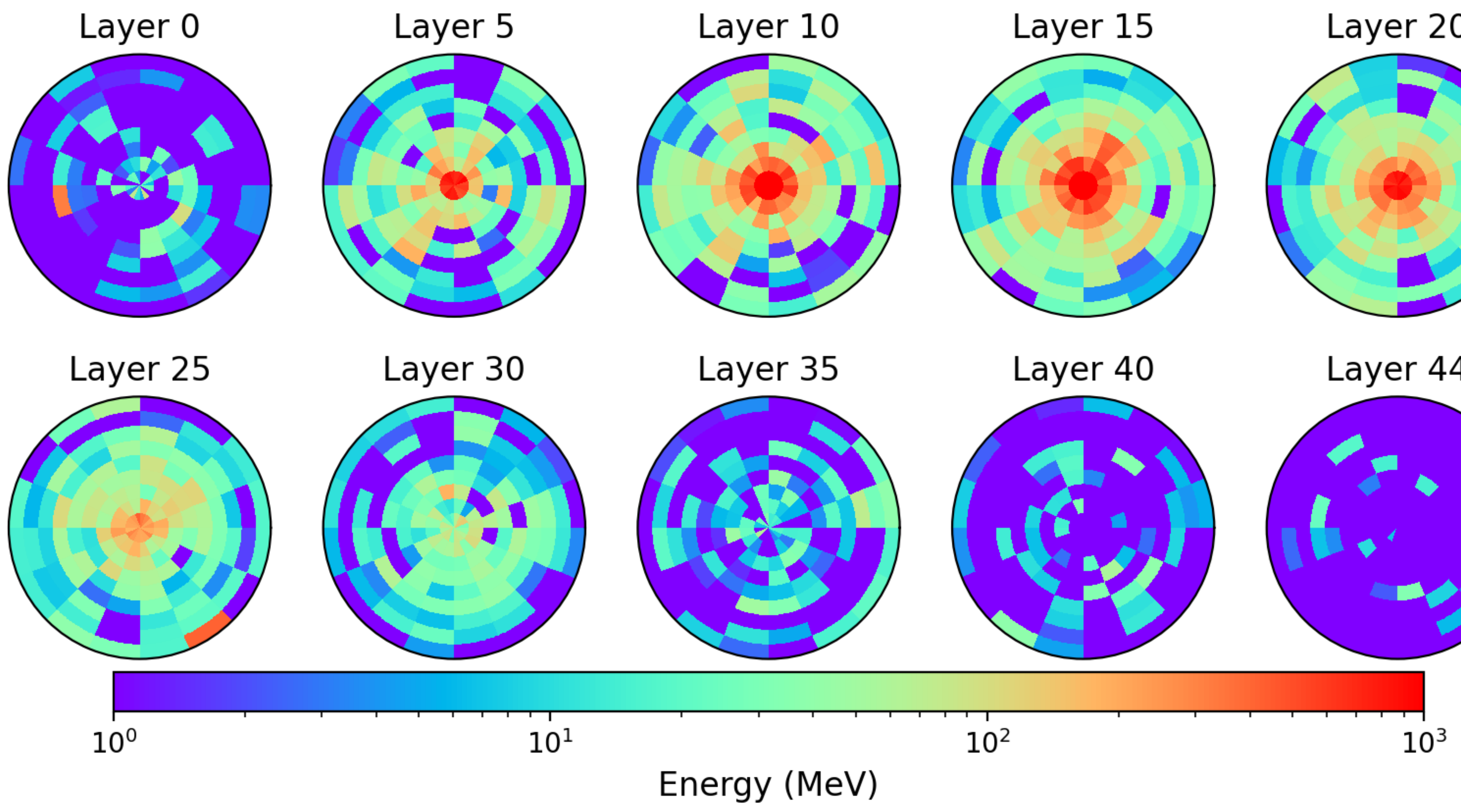

KLD
RBM
z'
P(ξ|Z,E)
ξ
P(χ|Z,E)
χ
x
P(Z|X,E)
z
E
Heaviside(X)
Hit Loss
MSE
Training
QA-RBM
z
E
P(ξ|Z,E)
ξ
P(χ|Z,E)
χ
x
Inference

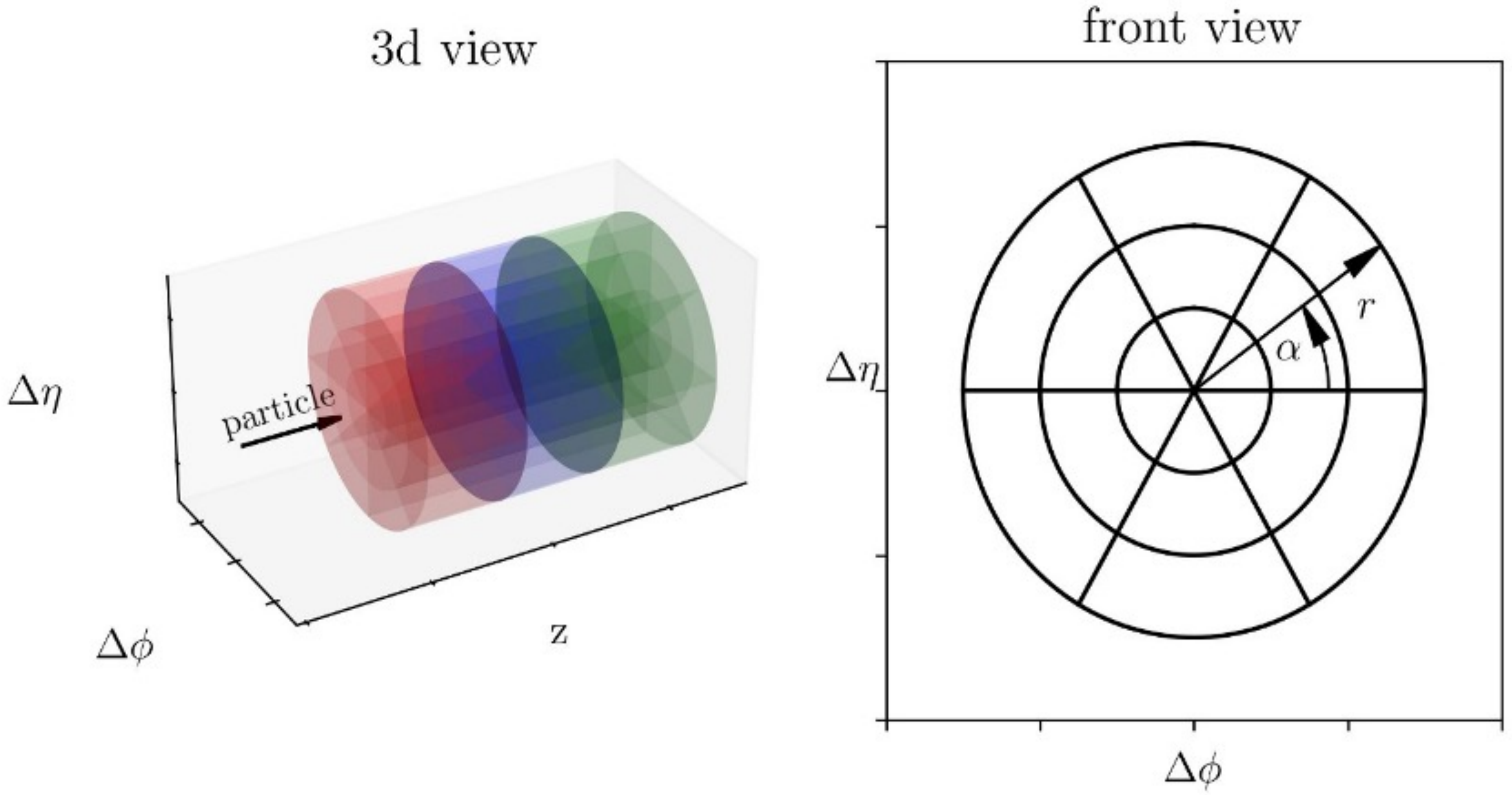
CaloChallenge Dataset

Dataset	
Particle type	Electron showers
Layers	45
Voxels per layer	9 radial * 16 angular
Incident energies	Log-uniform distribution (1GeV-1TeV)
N. of events	100,000

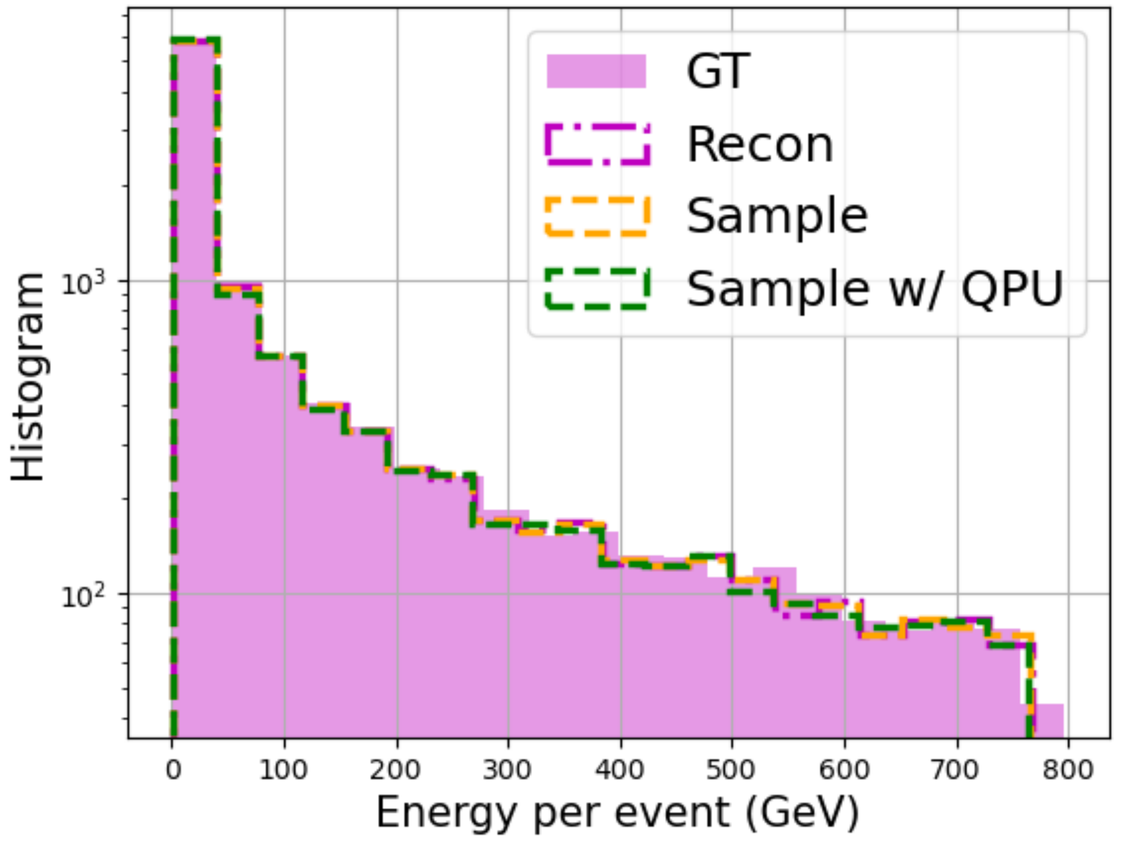


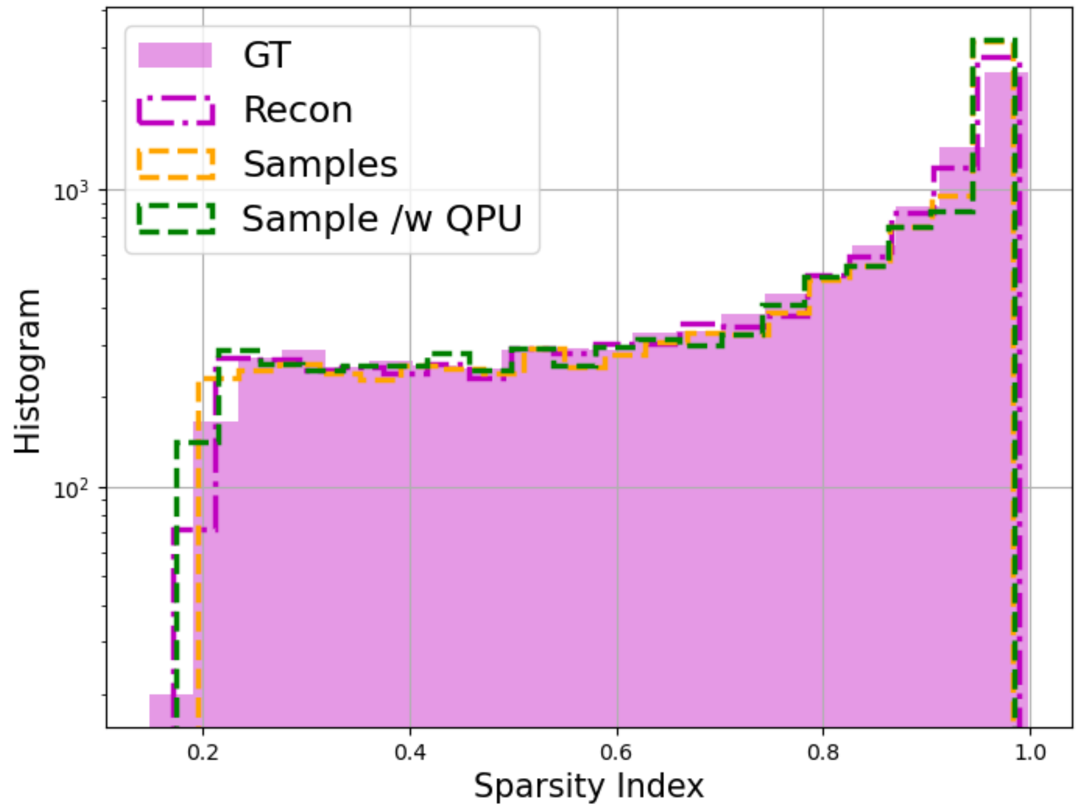


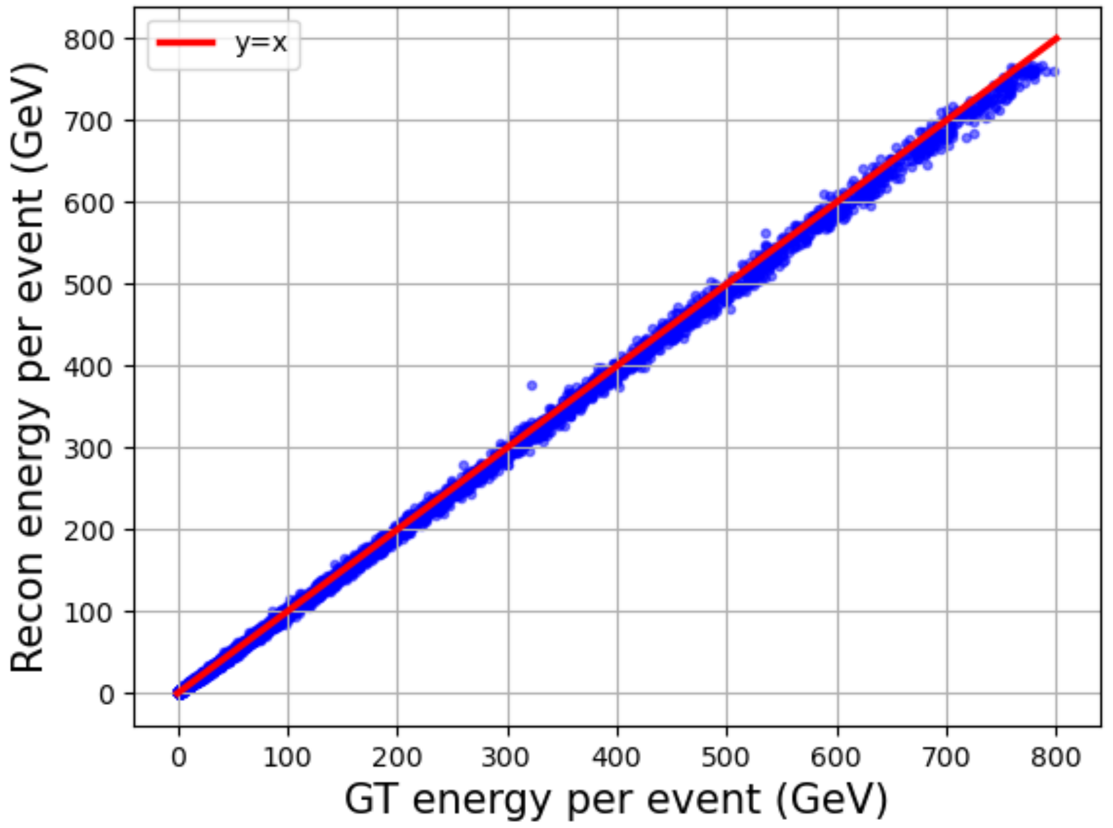


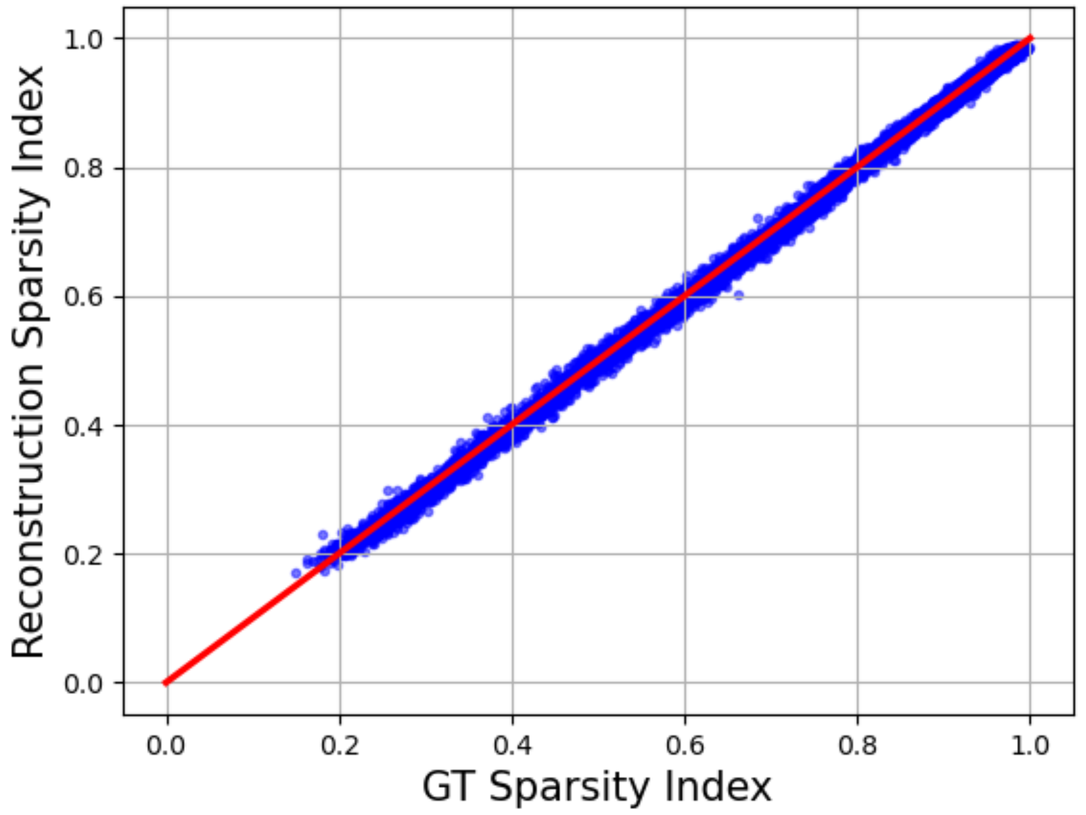


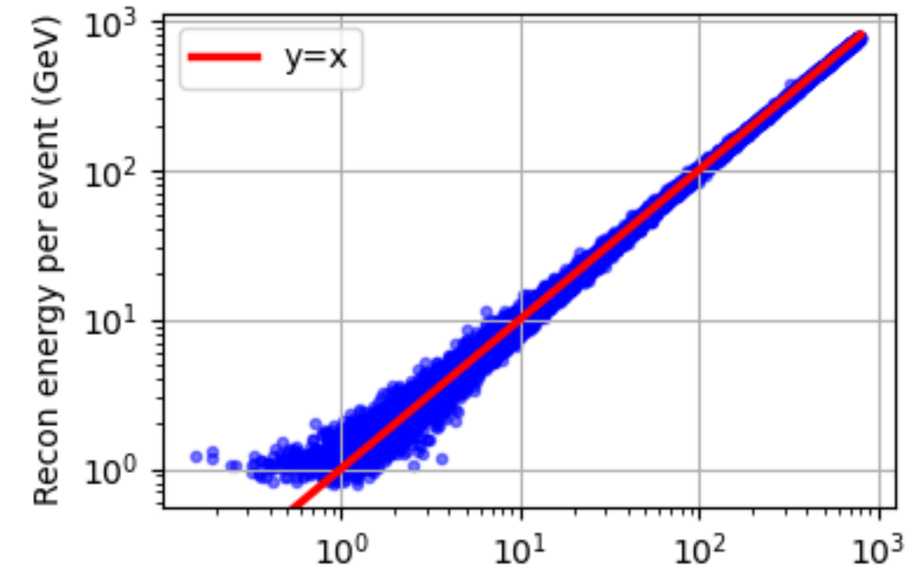
Results

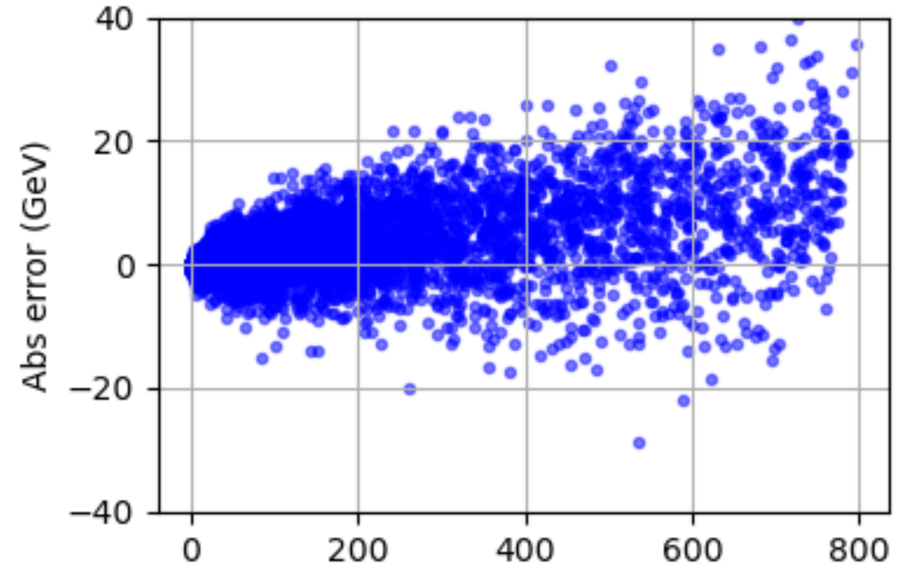

GT
Recon
Sample
Sample w/ QPU
Histogram
Energy per event (GeV)

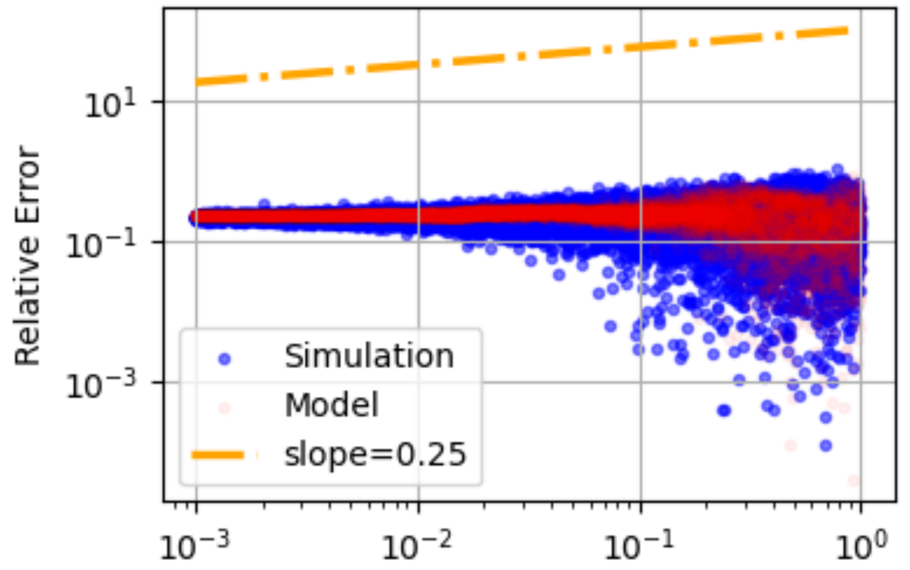

GT
Recon
Samples
Sample /w QPU
Histogram
Sparsity Index

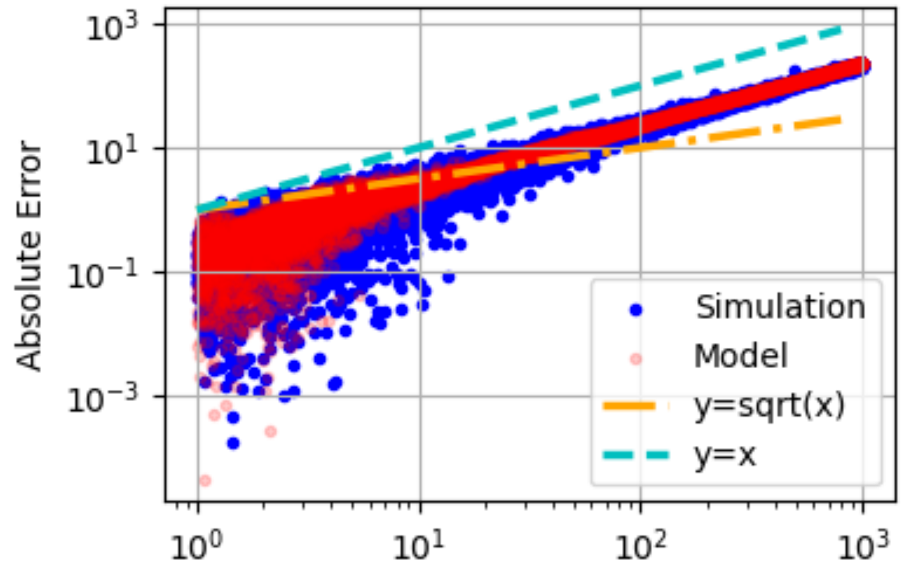

y=x
Recon energy per event (GeV)
GT energy per event (GeV)

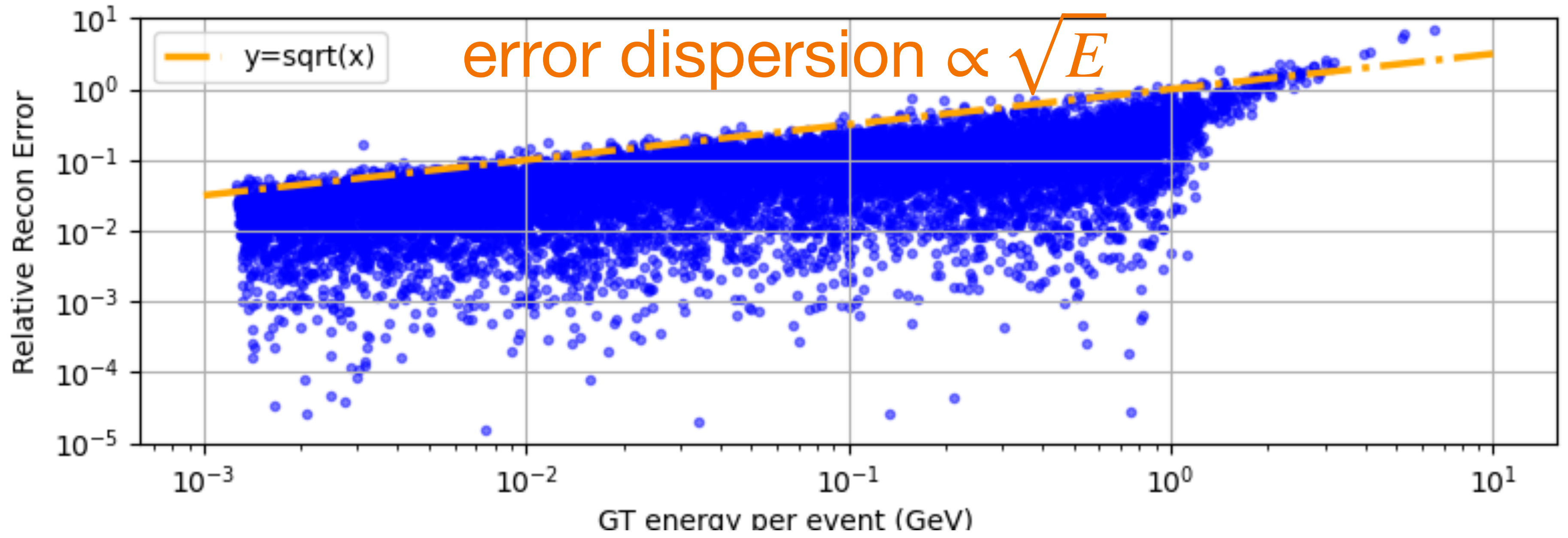

Reconstruction Sparsity Index
GT Sparsity Index

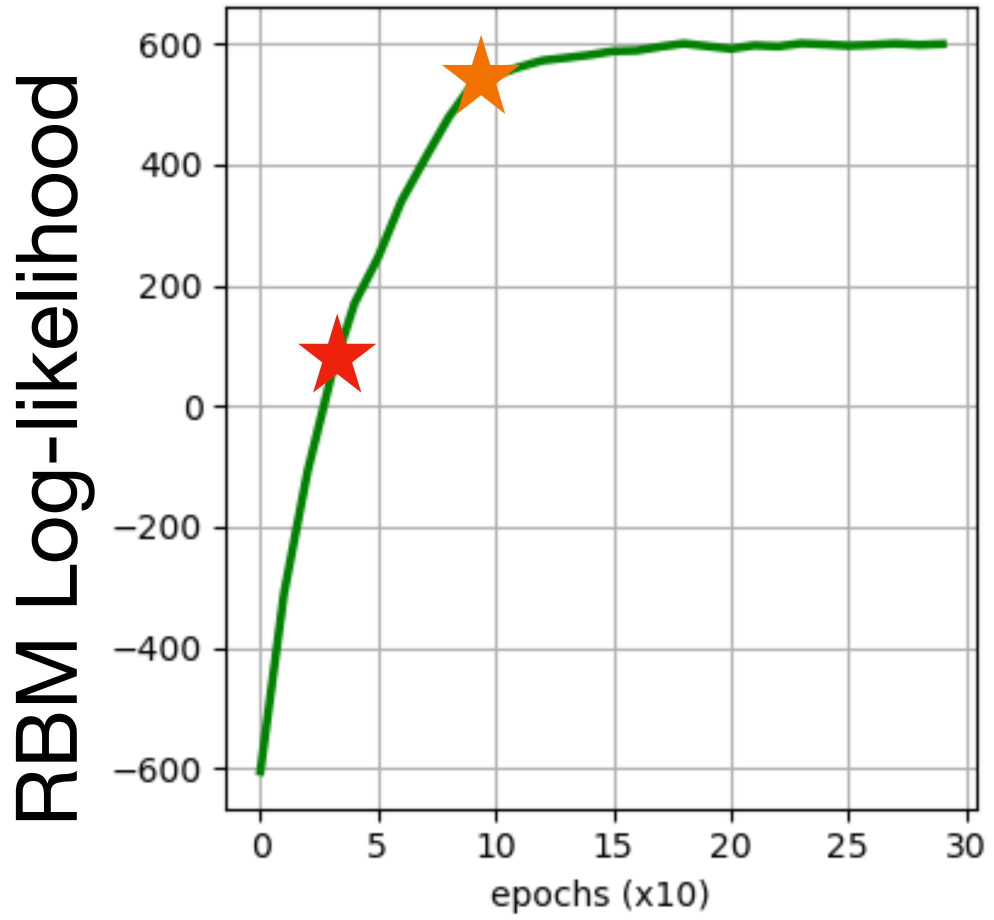

y=x
Recon energy per event (GeV)


Abs error (GeV)


Simulation
Model
slope=0.25
Relative Error


Simulation
Model
y=sqrt(x)
y=x
Absolute Error


y=sqrt(x)
error dispersion ∝ $\sqrt{E}$
Relative Recon Error
GT energy per event (GeV)


RBM Log-likelihood
epochs (x10)

Results

	Wall time to generate 1024 samples
Calorimeter Geant4	$\sim 400\ s$
GPU A100	$2.19 \pm 0.14\ s$
QPU	$\sim 0.180\ s$
Decoder	$\sim 0.01\ s$

QPU_ANNEAL_TIME_PER_SAMPLE
20 µs

QPU_READOUT_TIME_PER_SAMPLE
136 µs

QPU_DELAY_TIME_PER_SAMPLE
21 µs

Geant4 time per sample
O(1) s

QPU ~12x faster than GPU

QPU pipeline ~$2 \cdot 10^3$x faster than Geant4

Future directions and caveats

- In the process of getting dataset from ATLAS.
- New method for beta effective estimation.
- Training using QPU.
- Conditionalizing QPU.

KL method for beta effective calibration (Method 1).

Suppose two RBMs, QA and B described by the same Hamiltonian…

$$P_{QA}(x) = \frac{e^{-\beta_{QA} H(x)}}{Z(\beta_{QA})} , \tag{E22}$$

$$P_B(x) = \frac{e^{-\beta H(x)}}{Z(\beta)} . \tag{E23}$$

We denote as β_{QA} and β the inverse temperatures of system QA and B, respectively. The Kullback-Liebler divergence associated to these two system yields:

$$D_{KL}(P_{QA}||P_B) = (\beta - \beta_{QA})\langle H\rangle_{QA} + \ln \frac{Z(\beta)}{Z(\beta_{QA})} , \tag{E24}$$

from which it is trivial to show that $\beta = \beta_{QA}$ yields zero in the KL divergence. The KL divergence derivative w.r.t. β yields

$$\frac{\partial D_{KL}}{\partial \beta} = \langle H\rangle_{QA} - \langle H\rangle_{B(\beta)} , \tag{E25}$$

where we have made explicit the β dependence of system B. We can fit β through gradient descent using the KL divergence, which leads to:

$$\beta_{t+1} = \beta_t - \eta \left(\langle H(x)\rangle_{QA} - \langle H(x)\rangle_{B(\beta)}\right) \tag{E26}$$

$$H(x) \to H(x)/\beta$$

$$\beta_{t+1} = \beta_t - \frac{\eta}{\beta_t} \left(\langle H(x)\rangle_{QA^{(r)}} - \langle H(x)\rangle_{B(1)}\right) \tag{E27}$$

New method for beta effective calibration (Method 2 aka *Hao's Method*)

Suppose two RBMs, QA and B described by the same Hamiltonian…

$$P_{QA}(x) = \frac{e^{-\beta_{QA} H(x)}}{Z(\beta_{QA})}, \tag{E28}$$

$$P_B(x) = \frac{e^{-\beta H(x)}}{Z(\beta)}. \tag{E29}$$

We denote as β_{QA} and β the inverse temperatures of system QA and B, respectively. Now, let us denote as S_{QA} and S_B as the entropy of QA and B, respectively, and assume $S_{QA} = S_B$, from which after some straightforward algebra:

$$\beta = \beta_{QA} \frac{\langle H \rangle_{QA}}{\langle H \rangle_{B(\beta)}} + \frac{\ln \frac{Z(\beta_{QA})}{Z(\beta)}}{\langle H \rangle_{B(\beta)}}. \tag{E30}$$

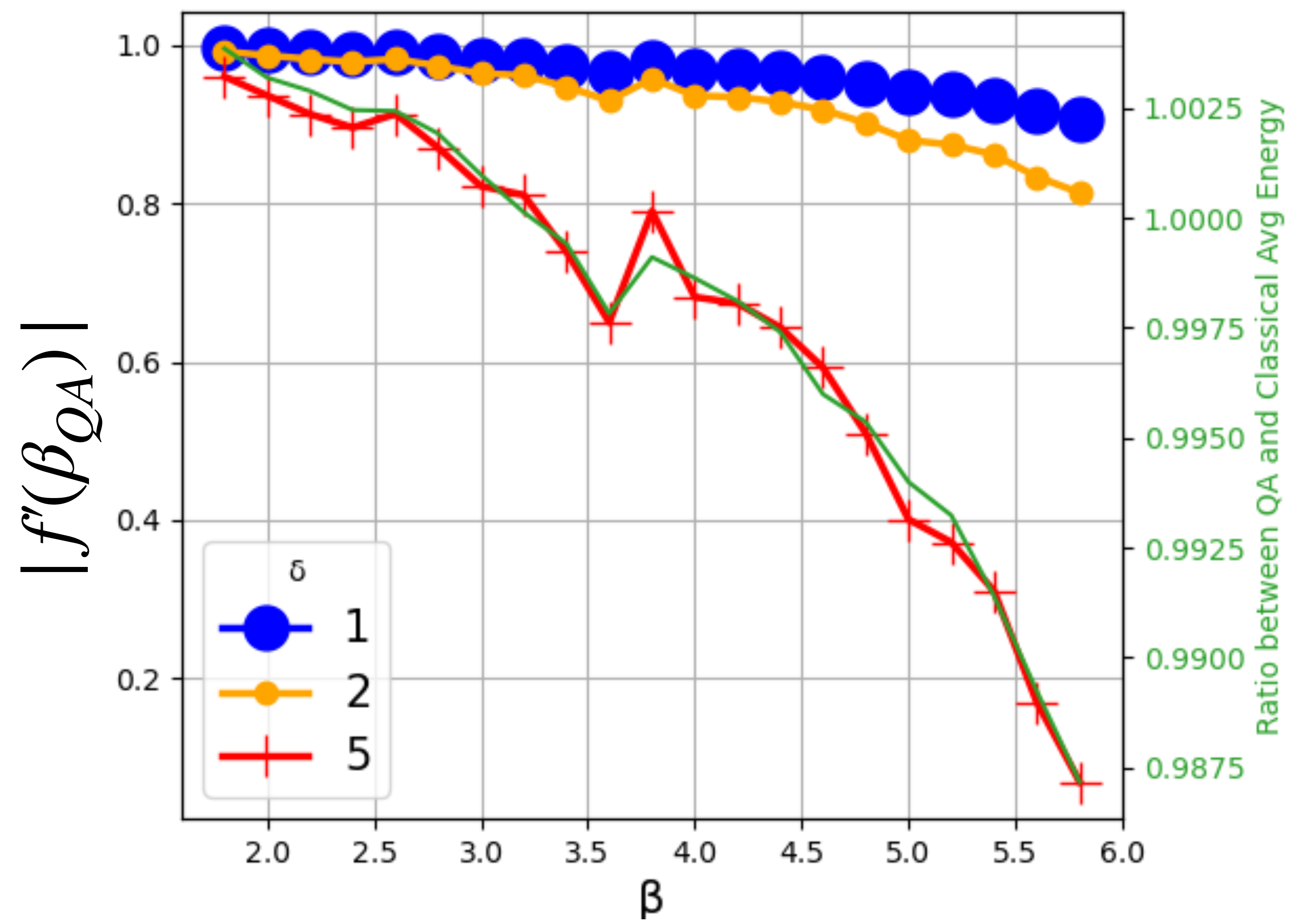
We can further simplify the previous expression by introducing the variable $\Delta\beta = \beta_{QA} - \beta$:

$$\beta = \beta_{QA} \frac{\langle H \rangle_{QA}}{\langle H \rangle_{B(\beta)}} + \frac{\ln \langle e^{-\Delta\beta H} \rangle_{B(\beta)}}{\langle H \rangle_{B(\beta)}}. \tag{E31}$$

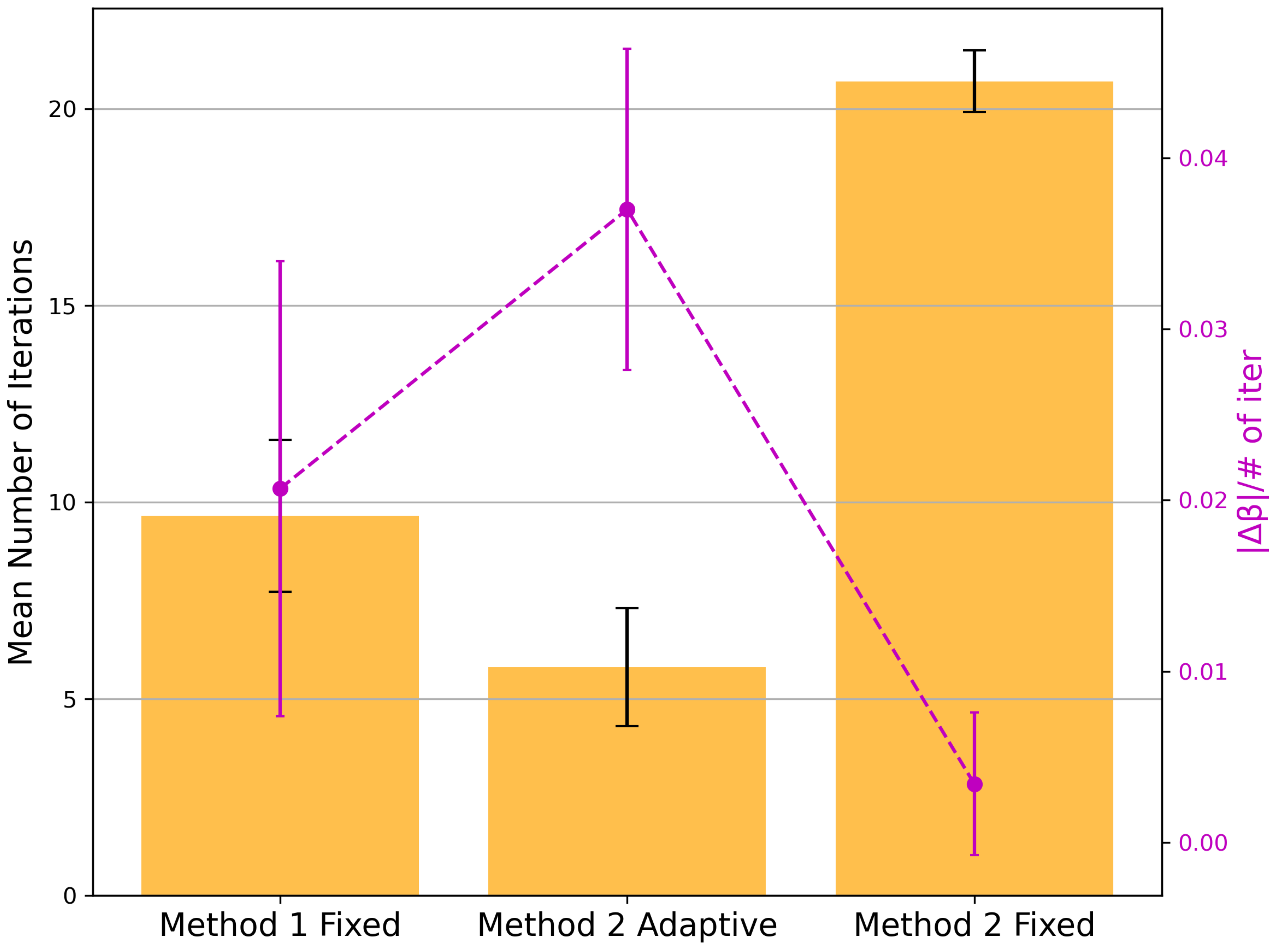
$$\beta_{t+1} = f_\delta(\beta_t) \equiv \beta_t \left(\frac{\langle H \rangle_{QA^{(r)}}}{\langle H \rangle_{B(1)}} \right)^\delta \tag{E32}$$

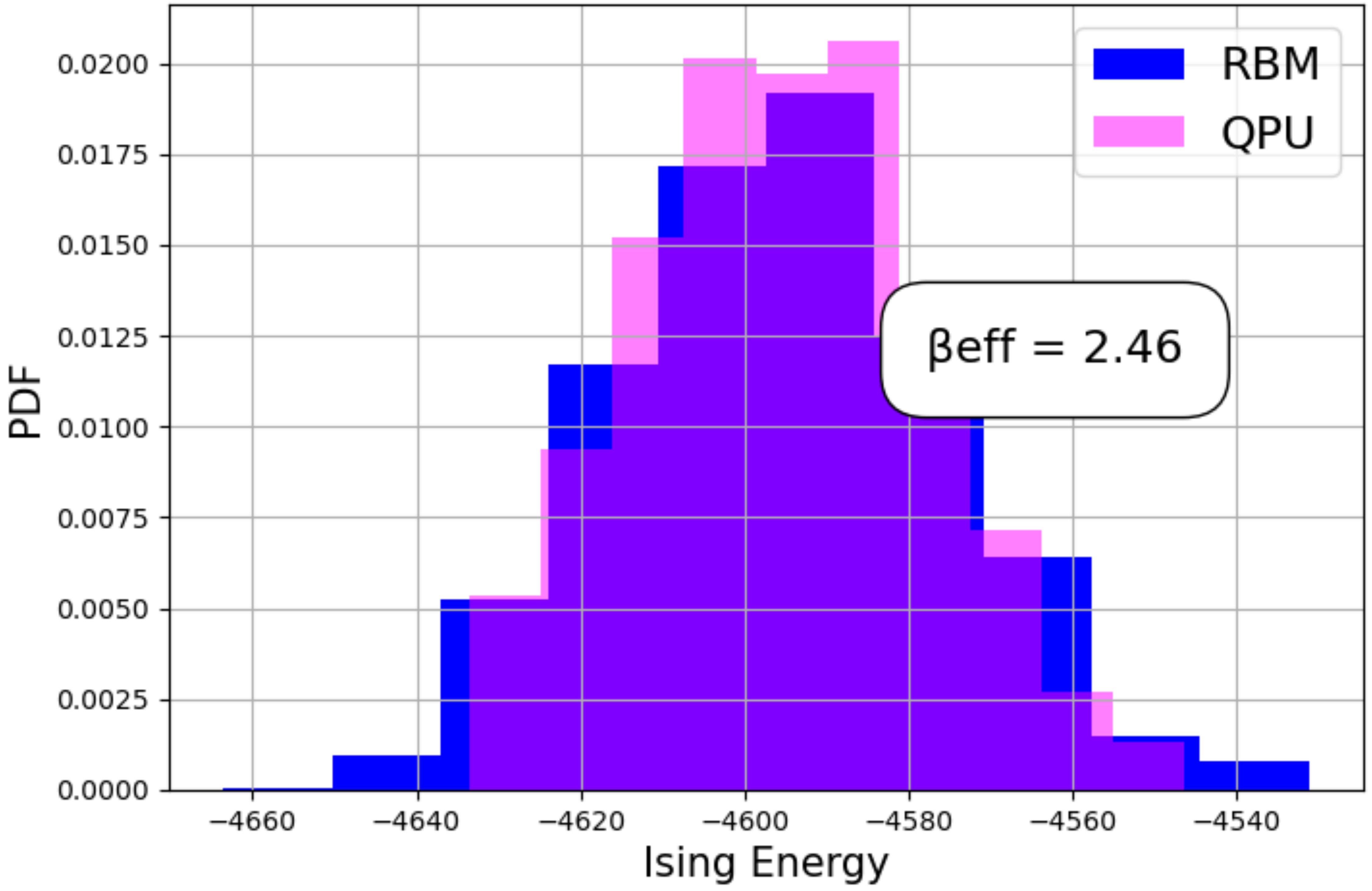
The function f_δ has a fixed point at $\beta = \beta_{QA}$. The stability condition close to the fixed point correspond to $|f'_\delta(\beta_{QA})| < 1$. The first derivative at the fixed point yields:

$$|f'_\delta(\beta_{QA})| = \begin{cases} |1 + \frac{\sigma^2_{QA}}{\langle H \rangle_{B(1)}}|, & \delta = 1 \\ |1 + \delta \frac{\sigma^2_{QA}}{\langle H \rangle_{QA}}|, & \delta \neq 1 \,. \end{cases} \tag{E33}$$

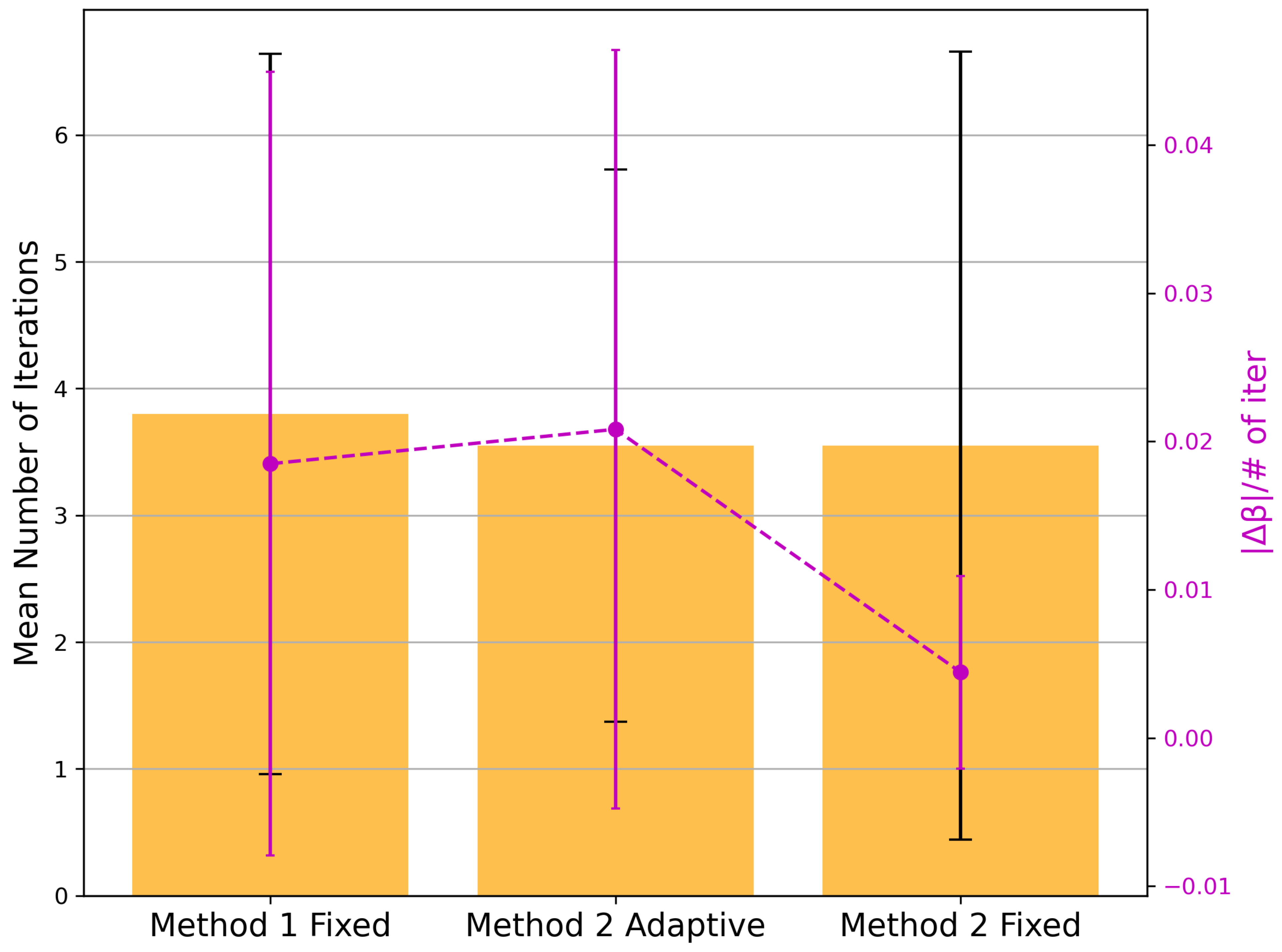


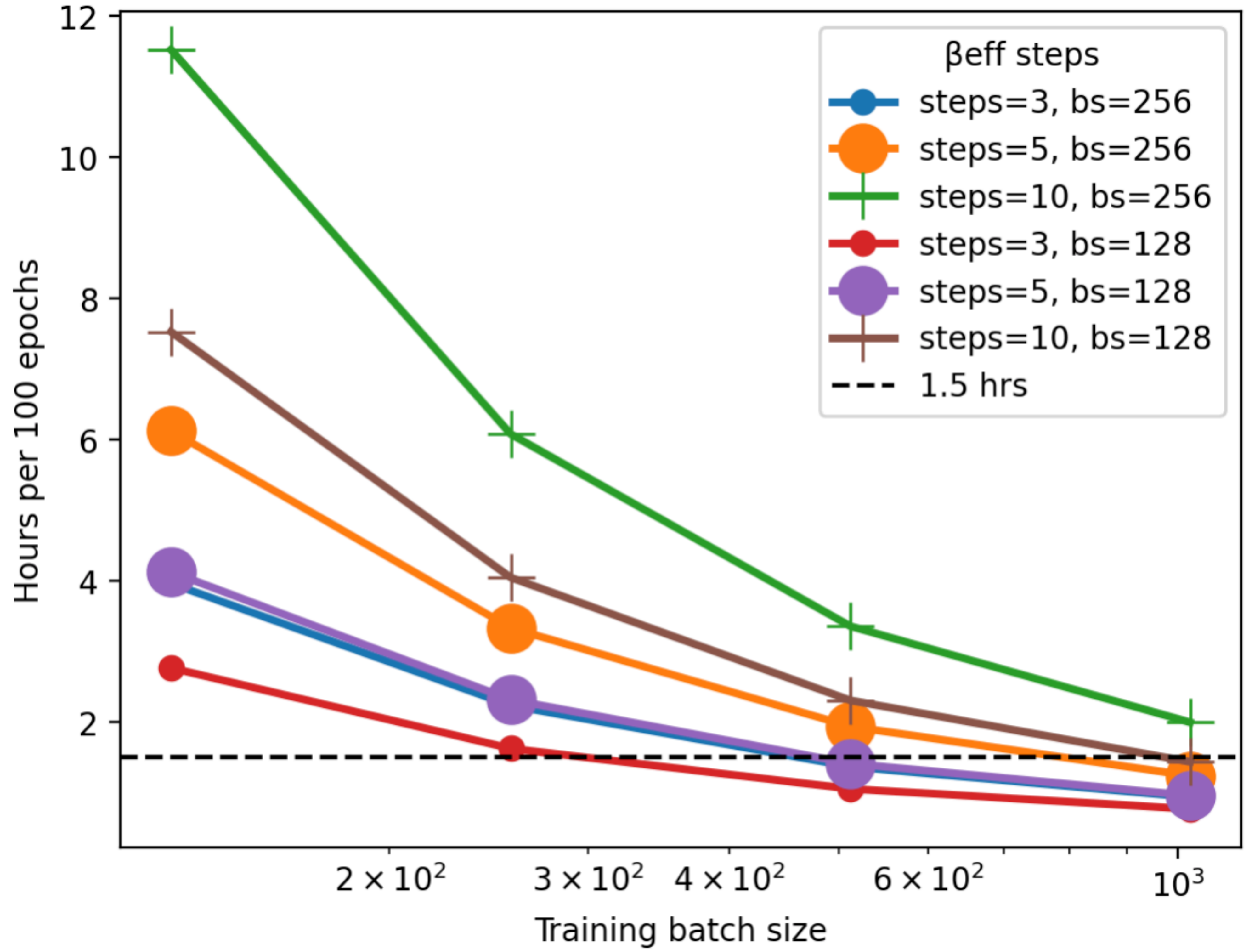
New method for beta effective calibration. (*By Hao*)





Training using QPU

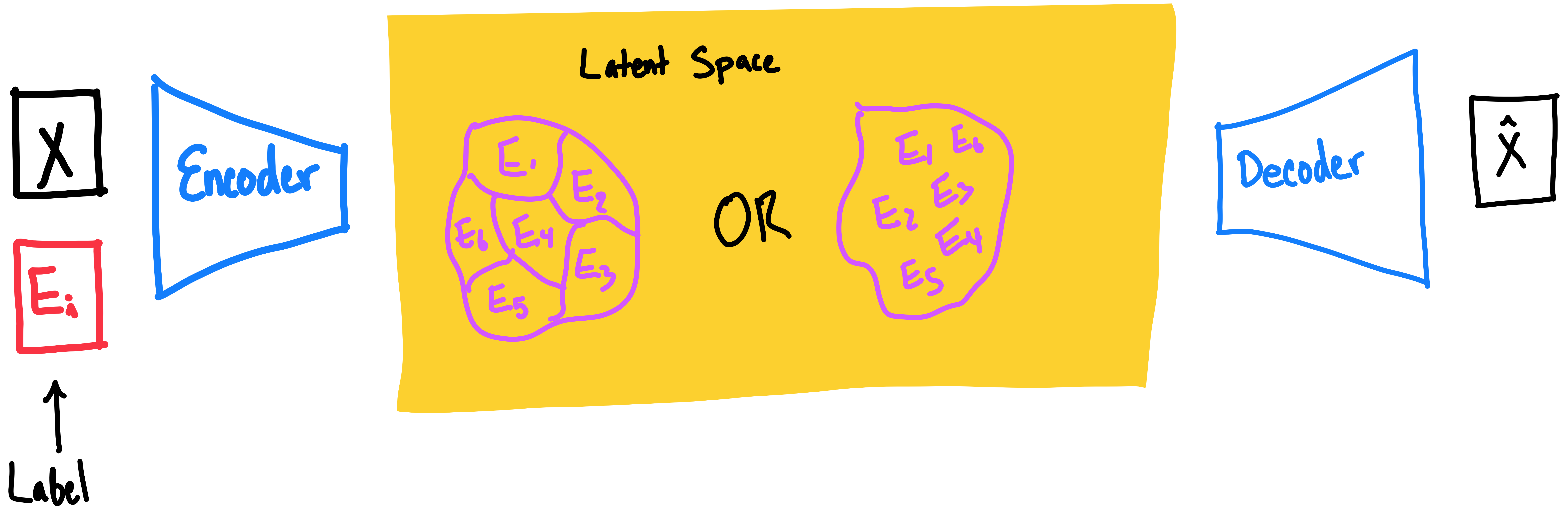




Conditionalizing QPU

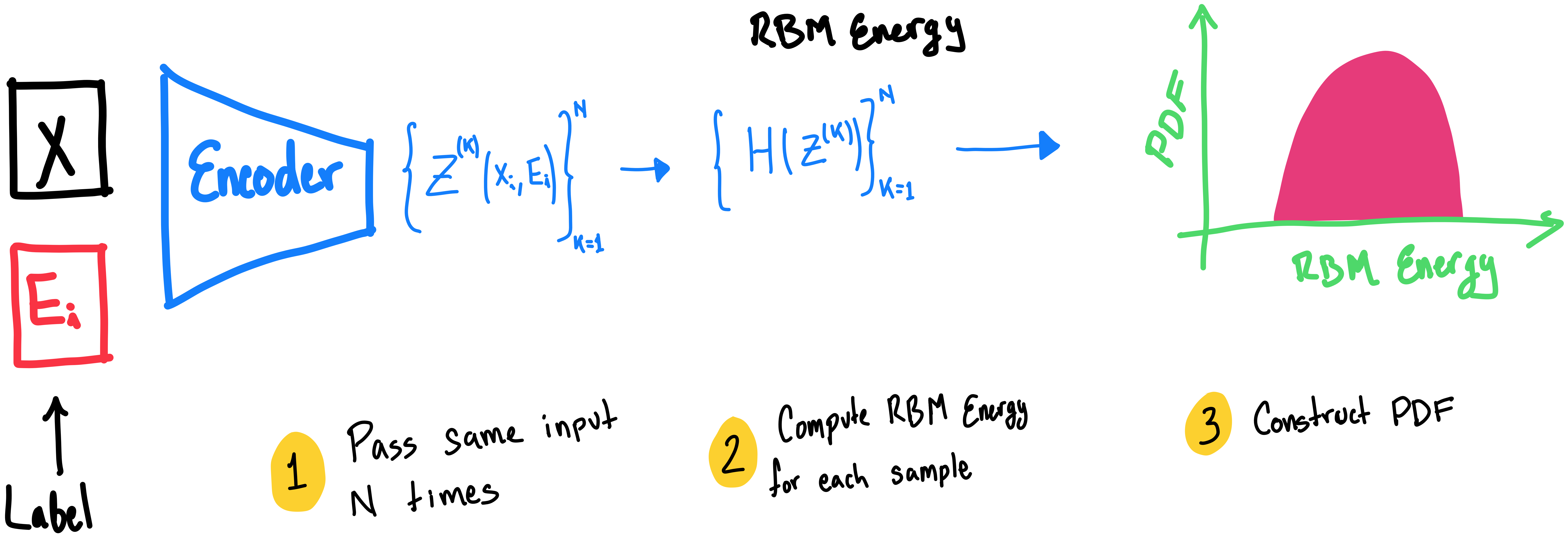
Conditionalizing QPU

Latent space clustering



Conditionalizing QPU

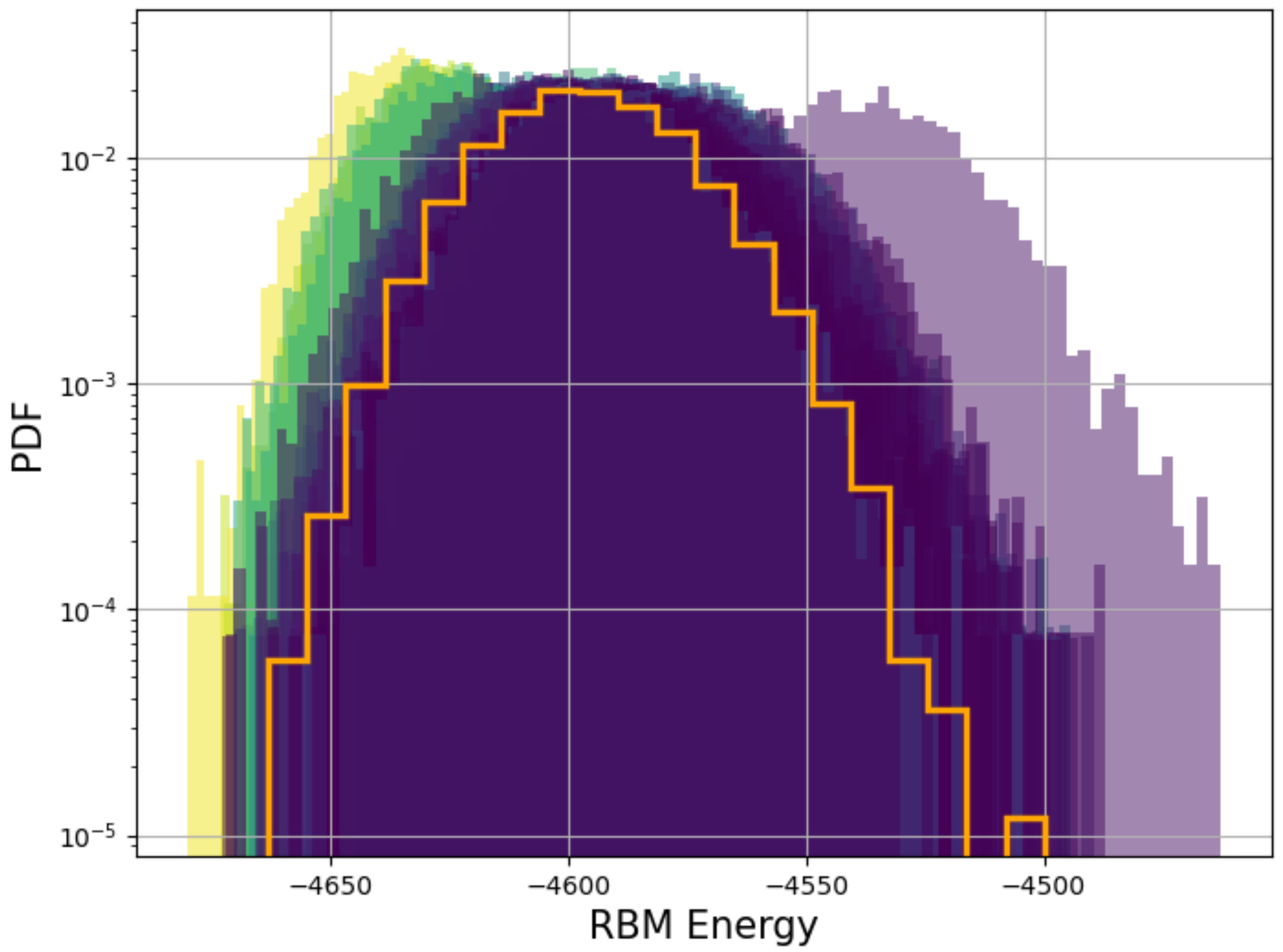
Latent space clustering

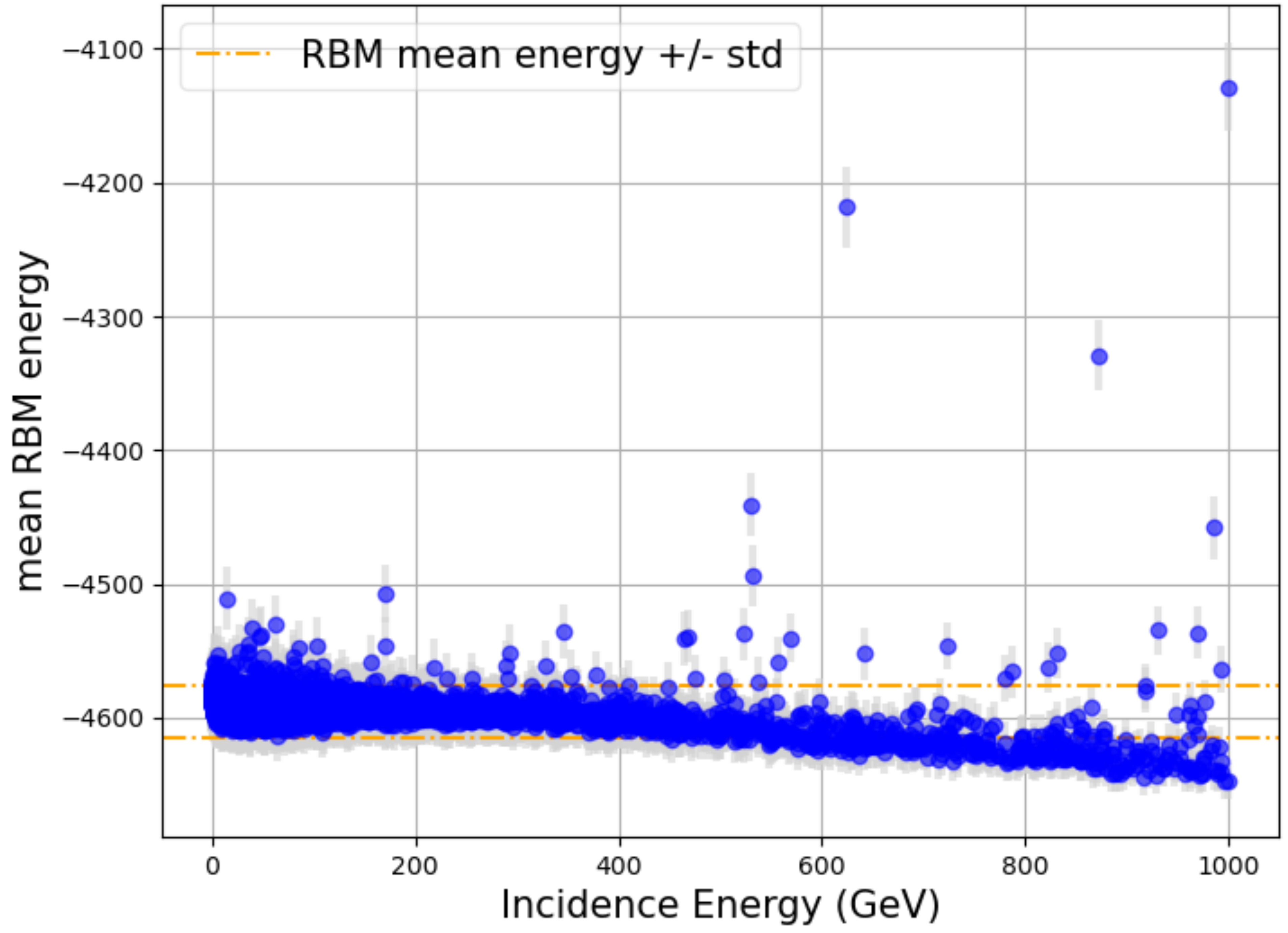


Conditionalizing QPU

Latent space clustering

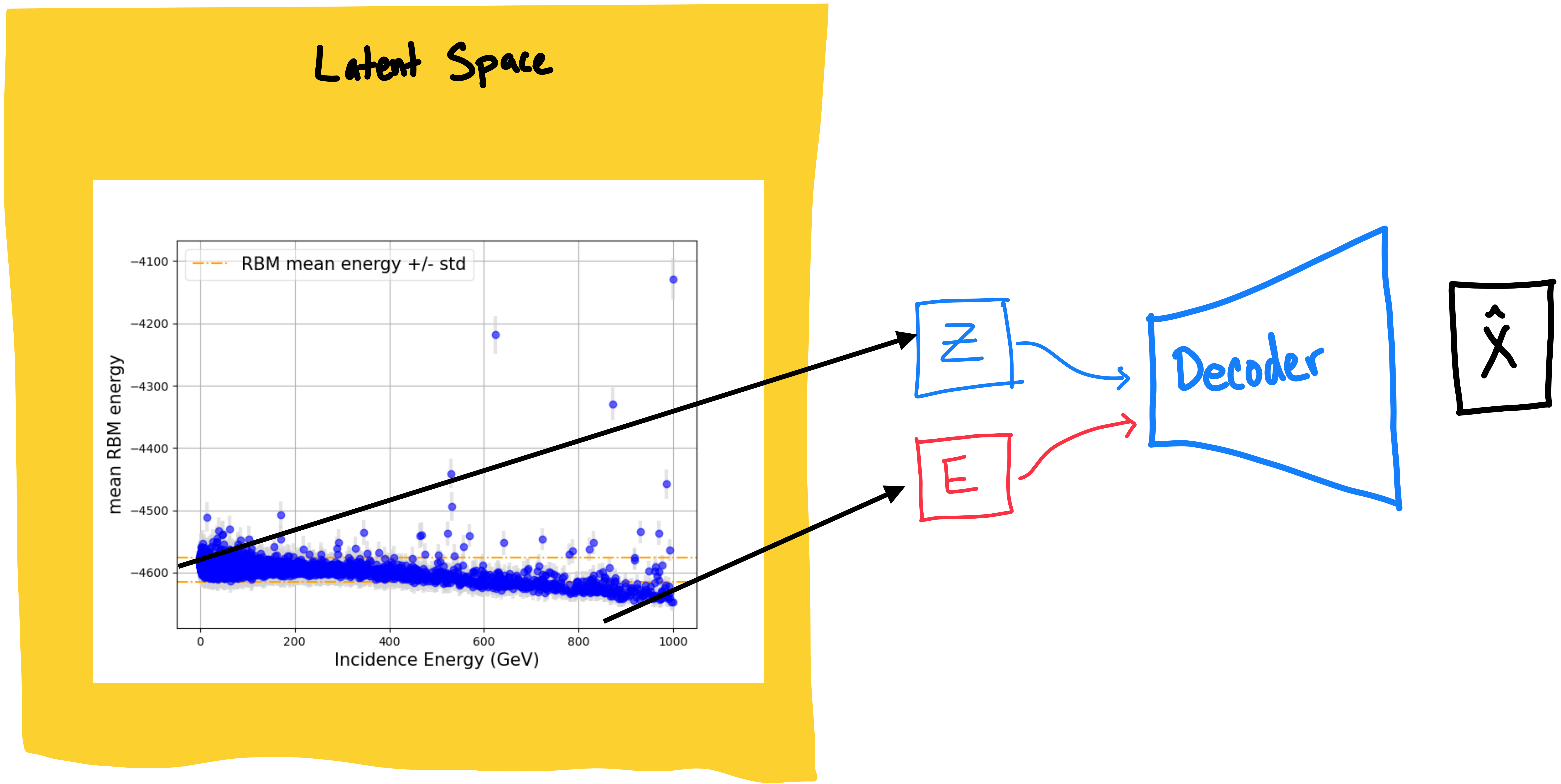
- We repeat this process for multiple events in the validation dataset and color each histogram. Low incidence energy correspond to dark colours, whereas high incidence energy correspond to light colours.



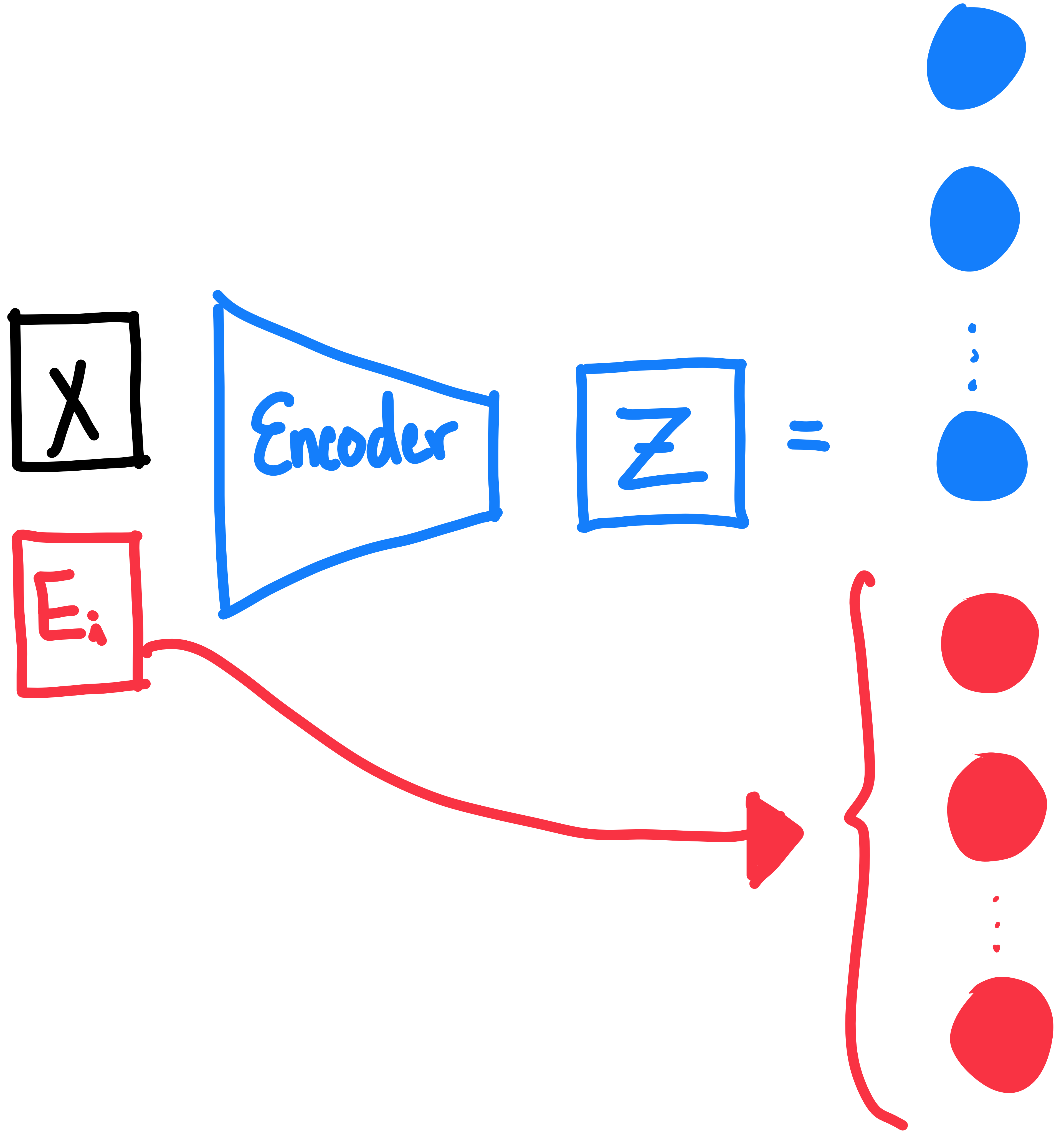


Conditionalizing QPU

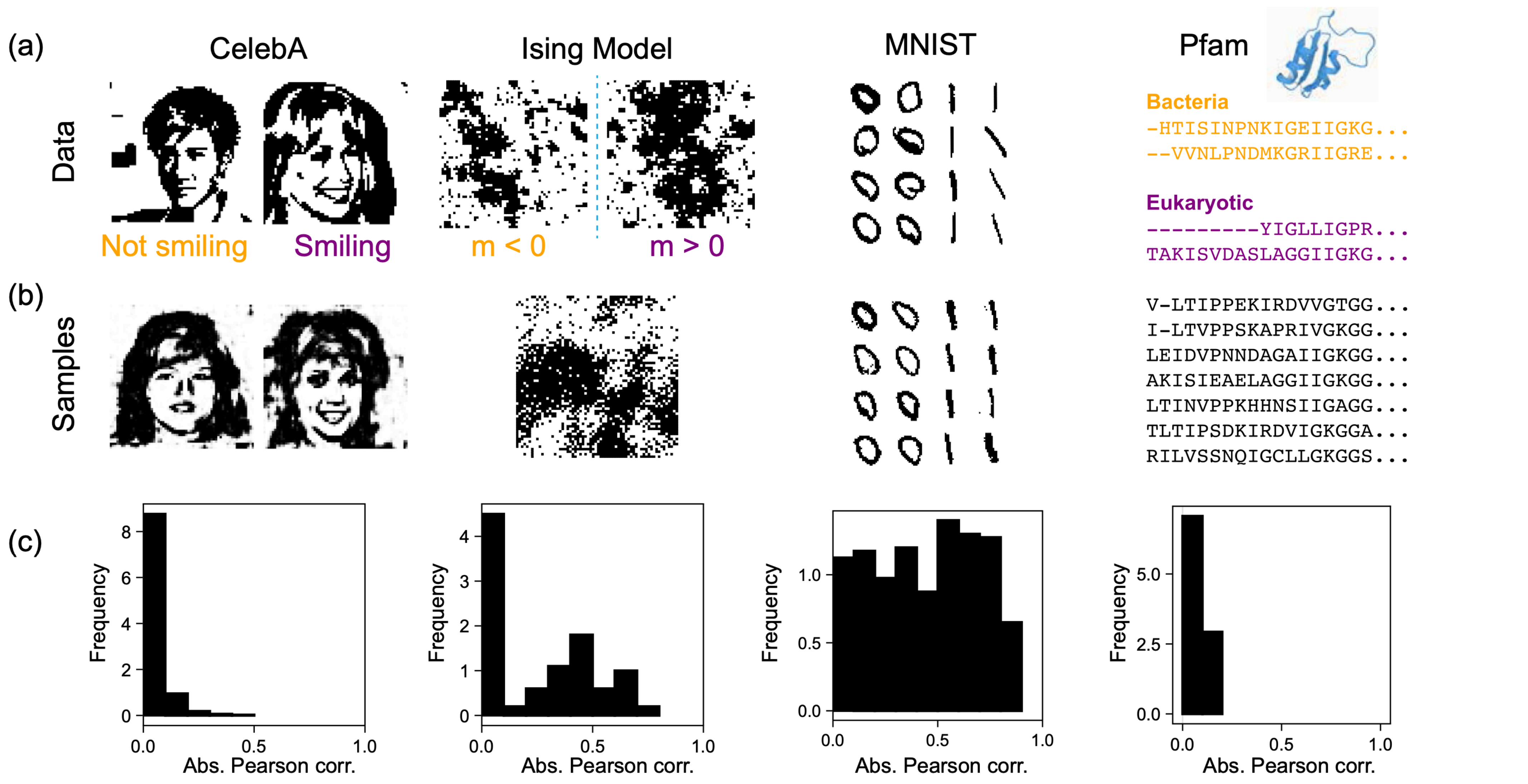
Latent space clustering



Conditionalizing QPU

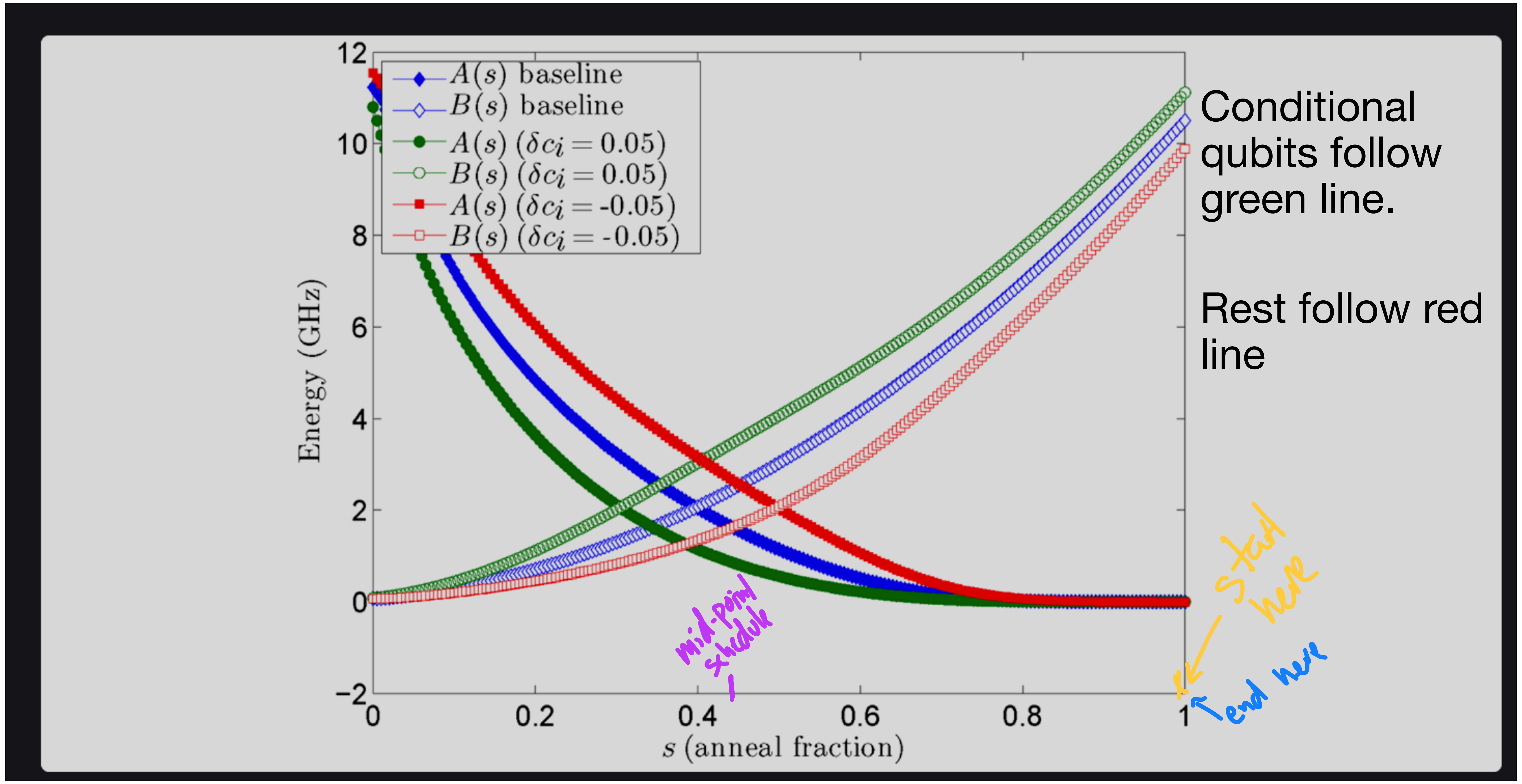


Conditionalized-qubits



Fernandez-de-Cossio-Diaz, Jorge, Simona Cocco, and Rémi Monasson. "Disentangling representations in restricted boltzmann machines without adversaries." *Physical Review X* 13.2 (2023): 021003.

Conditionalizing QPU



$$\mathcal{H}_{ising} = \underbrace{\frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right)}_{\text{Initial Hamiltonian}} + \underbrace{\frac{B(s)}{2}\left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\right)}_{\text{Final Hamiltonian}}$$

Example

This illustrative example configures a reverse-anneal schedule on a random native problem.

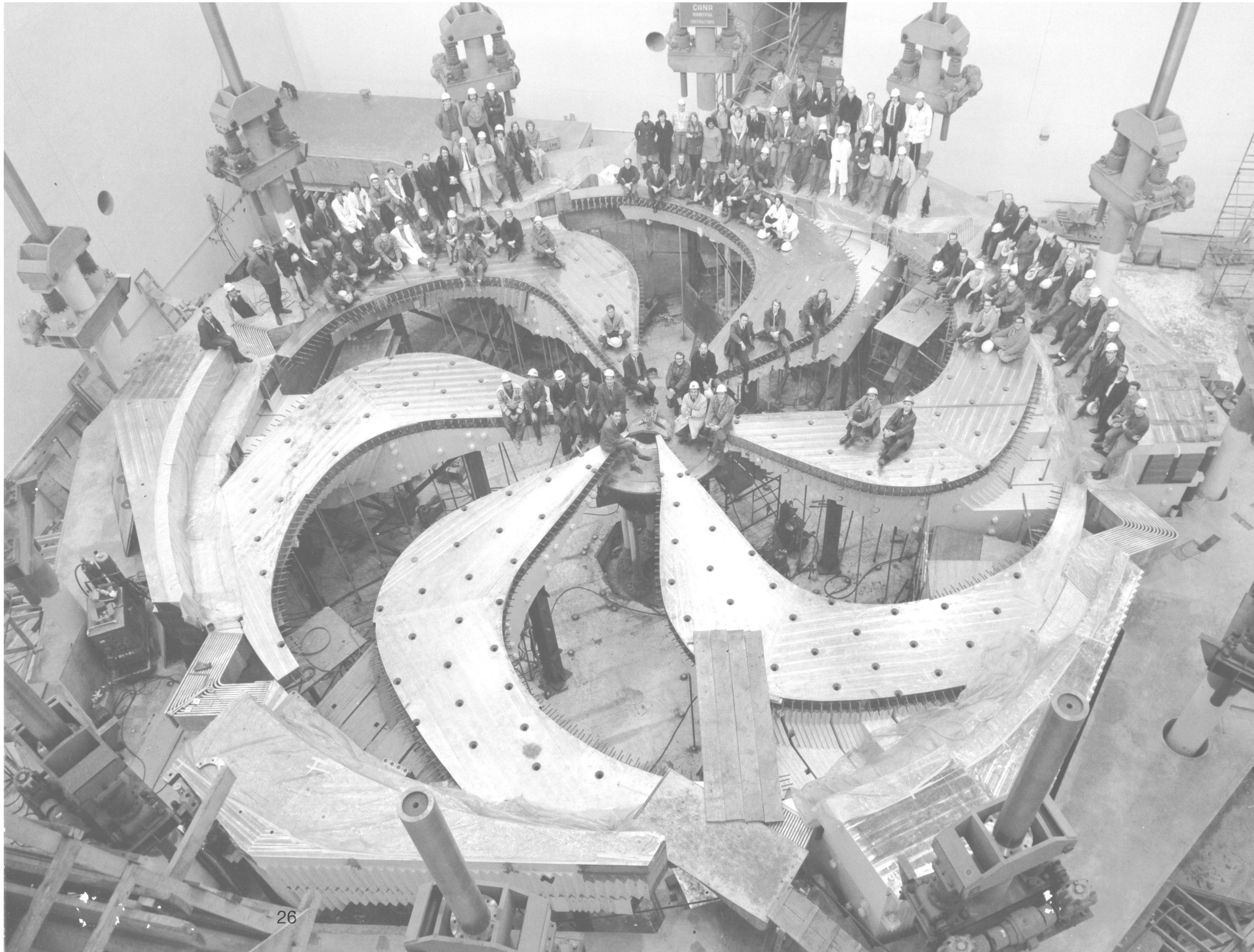
```
>>> from dwave.system import DWaveSampler
>>> import random
>>> qpu = DWaveSampler()
>>> J = {coupler: random.choice([-1, 1]) for coupler in qpu.edgelist}
>>> initial = {qubit: random.randint(0, 1) for qubit in qpu.nodelist}
>>> reverse_schedule = [[0.0, 1.0], [5, 0.45], [99, 0.45], [100, 1.0]]
>>> reverse_anneal_params = dict(anneal_schedule=reverse_schedule,
...                              initial_state=initial,
...                              reinitialize_state=True)
>>> sampleset = qpu.sample_ising({}, J, num_reads=1000, **reverse_anneal_params)
```

- Fixing the conditionalized-qubits' self-fields to max/min value.
- Offsetting conditionalized-qubits.
- Turning off the self-fields in transverse field associated to the conditionalized-qubits(?)

Acknowledgements

Team:

- Sebastian Gonzalez @ UBC
- Hao Jia @ UBC
- Sehmimul Hoque @ Waterloo University
- Abhishek Abhishek @ UBC
- Tiago Vale @ Simon Fraser Uni
- Soren Andersen @ Lund University
- Eric Paquet @ NRC
- Roger Melko @ Perimeter Institute
- Geoffrey Fox @ University of Virginia
- Max Swiatlowski @ TRIUMF
- Wojtek Fedorko @ TRIUMF



jtoledo@TRIUMF.ca

arXiv preprint arXiv:2312.03179 (2023).
arXiv preprint arXiv:2210.07430 (2022). NeurIPS 2021
Current work to be submitted