TRIUMF | THE UNIVERSITY OF BRITISH COLUMBIA | PI | DTRC-NRC | UNIVERSITY of VIRGINIA

Calo4pQVAE: A Particle-Calorimeter Surrogate Using Conditioned Quantum Annealers and Variational Autoencoders

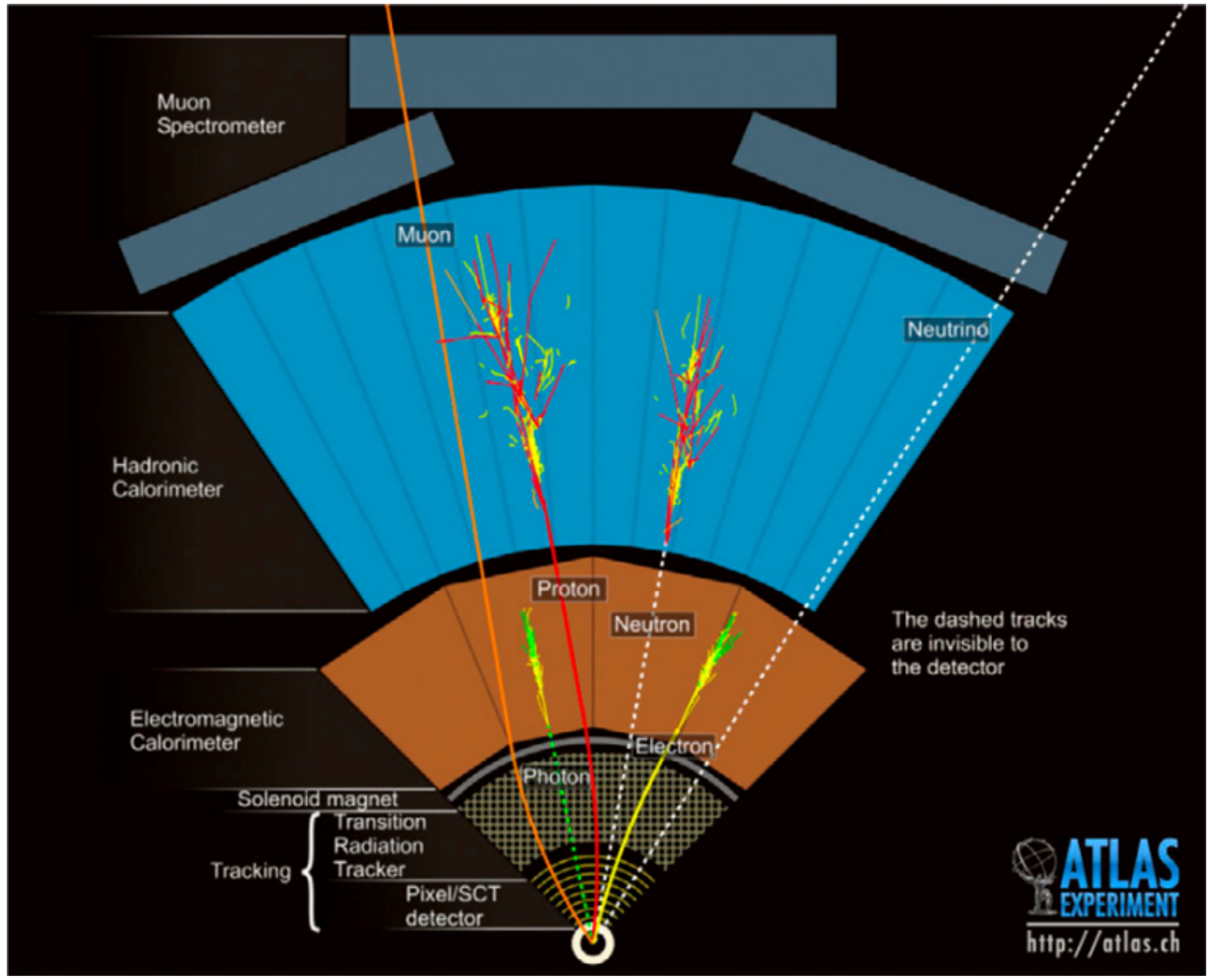
02/05/25 :: AI Seminar @ SLAC

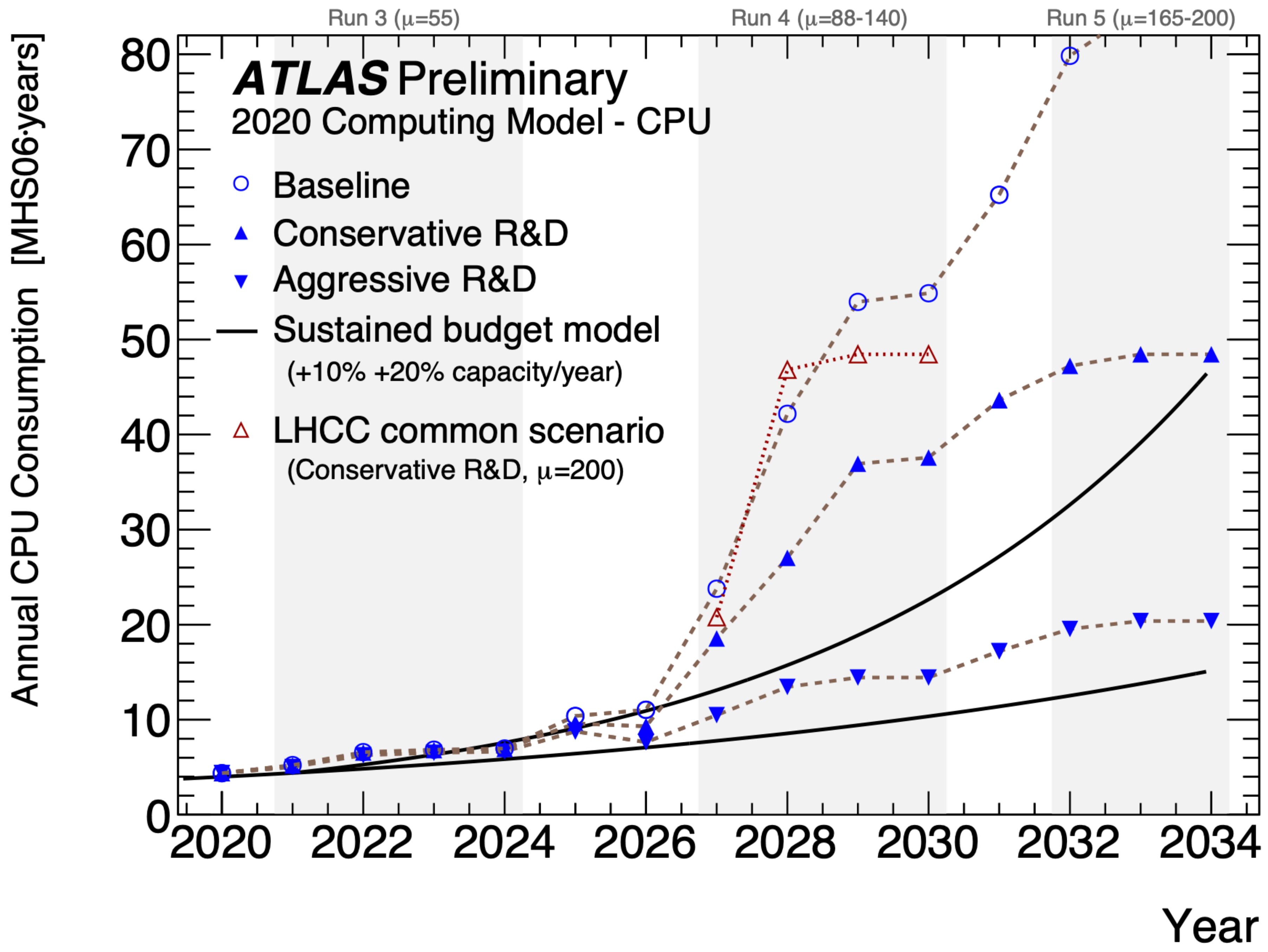
J. Quetzalcoatl Toledo-Marin
Quantum Machine Learning Research Associate

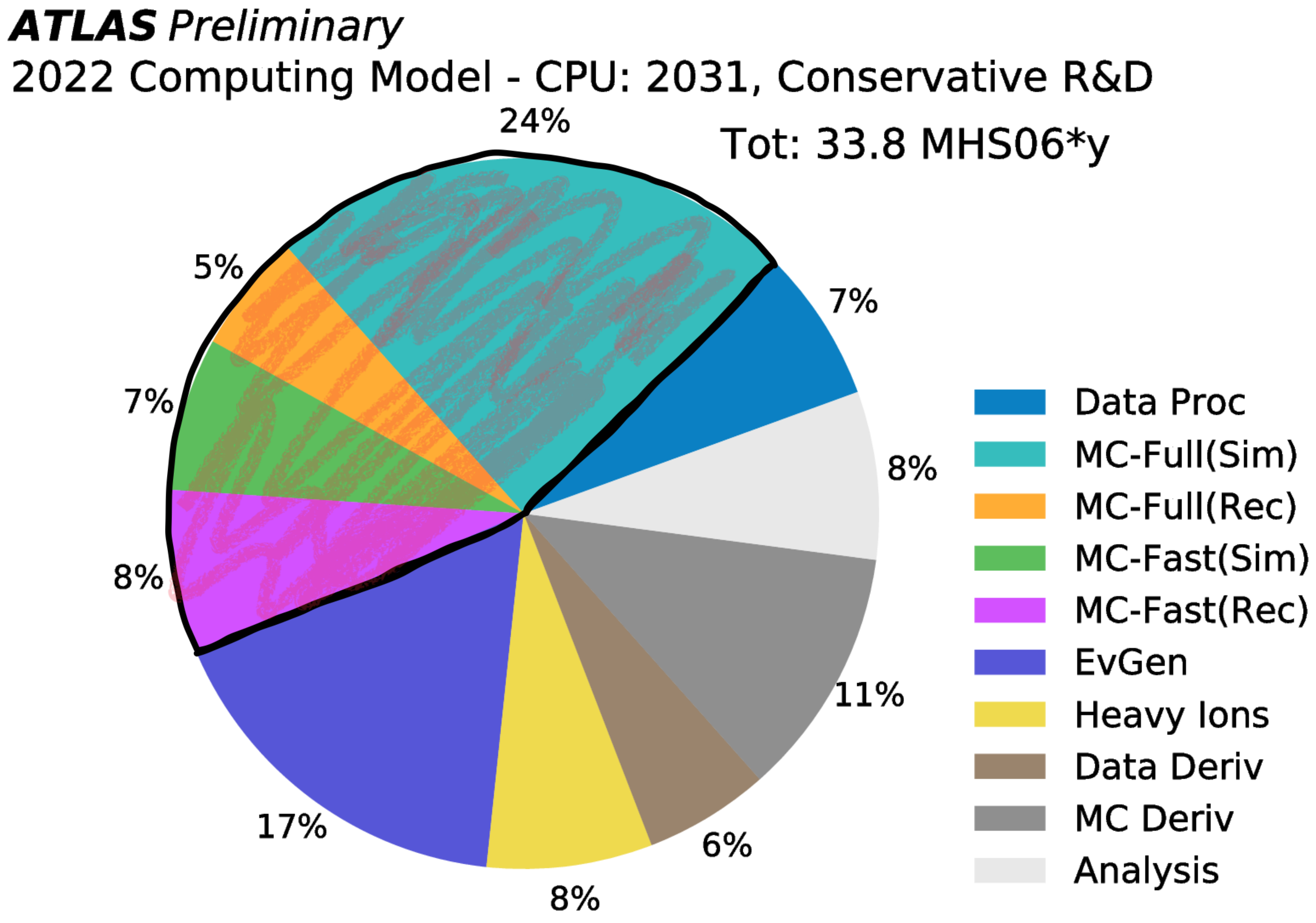
TRIUMF Canada's particle accelerator centre
Centre canadien d'accélération des particules

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.



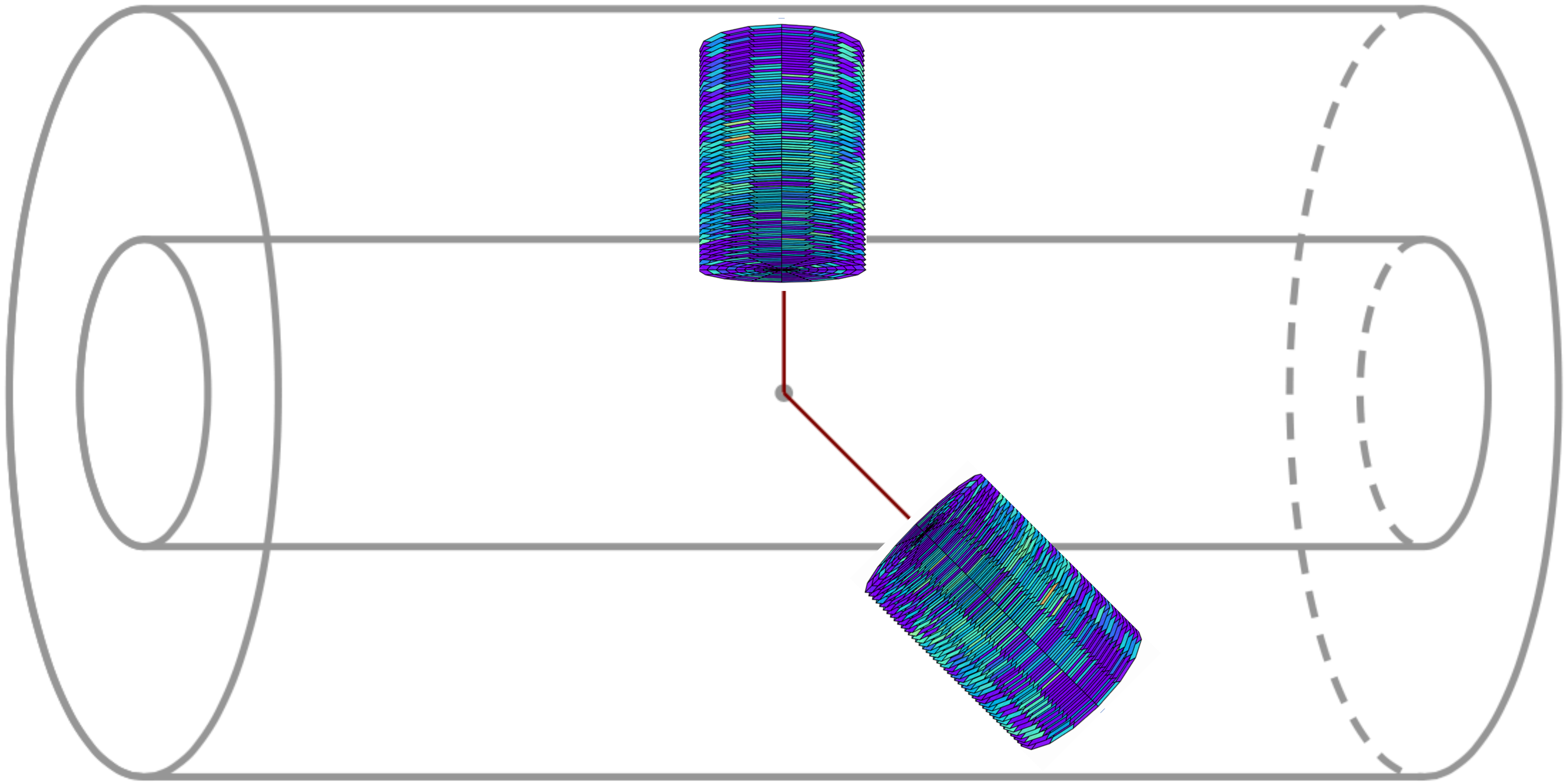




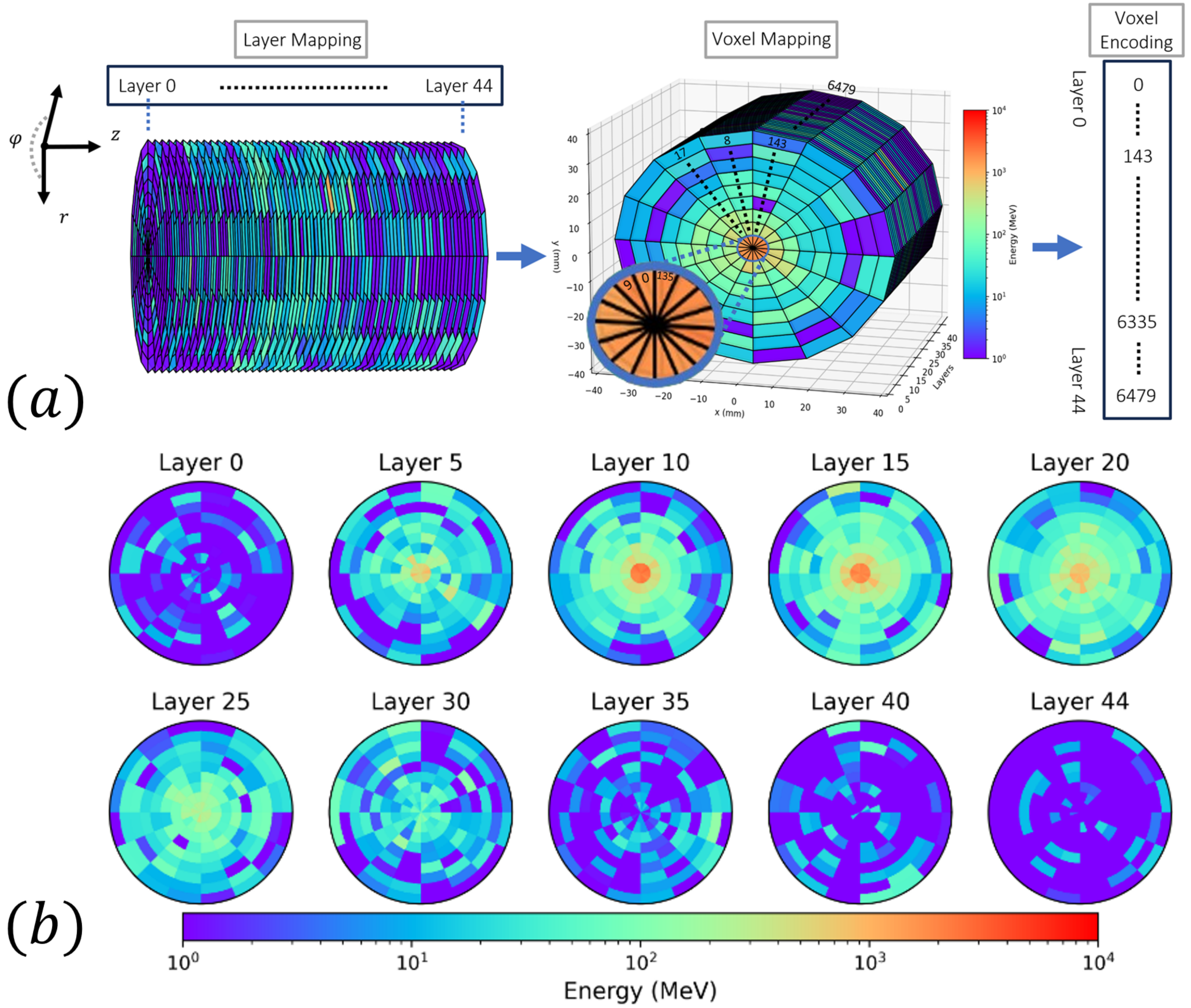
Scientific Data Lake for High Luminosity LHC project and other data-intensive particle and astro-particle physics experiments. InJournal of Physics: Conference Series 2020 Dec 1 (Vol. 1690, No. 1, p. 012166). IOP Publishing.

CaloChallenge

Dataset 2



CaloChallenge

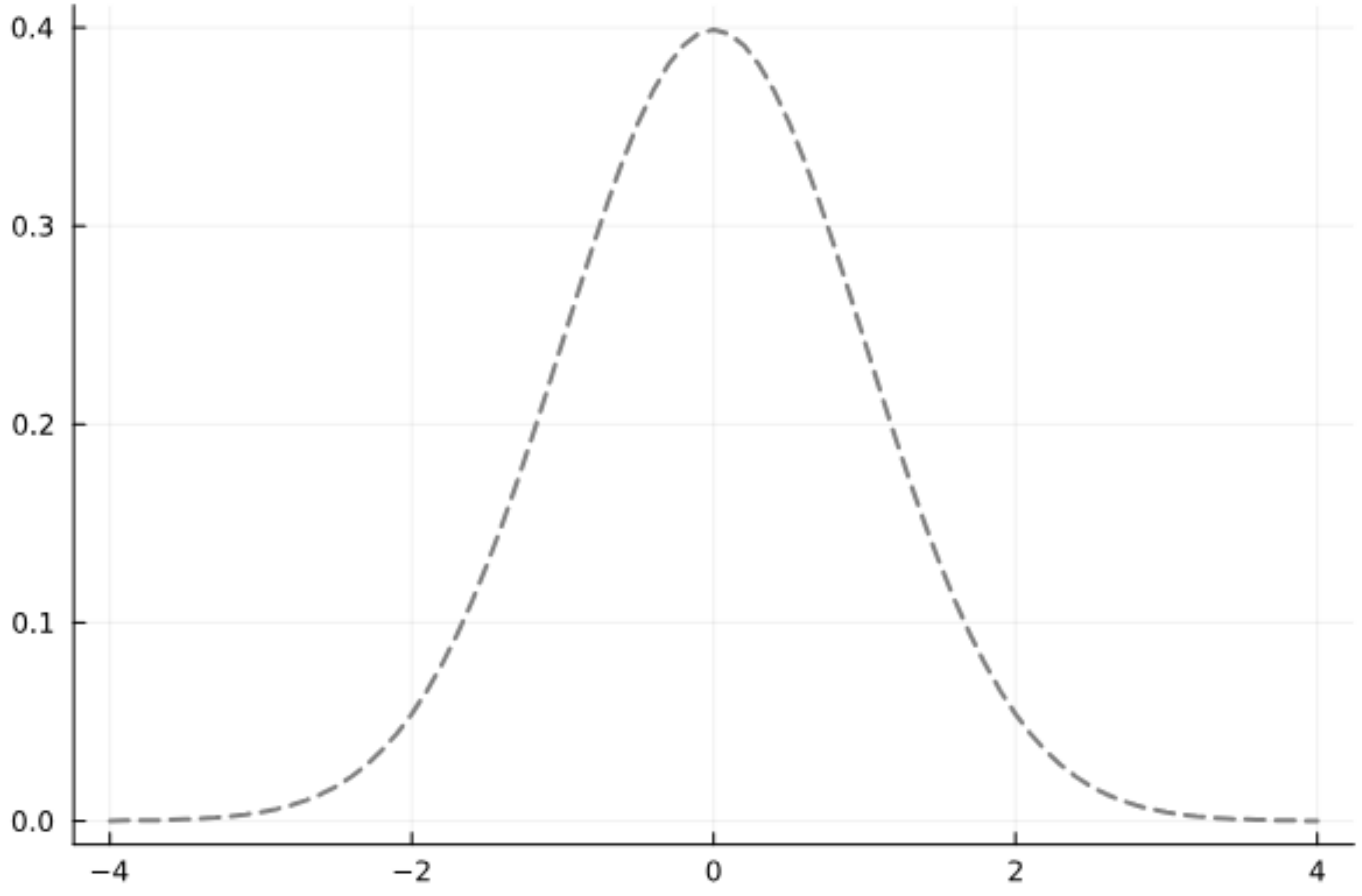


	Dataset
Particle type	Electron showers
Layers	45
Voxels per layer	9 radial * 16 angular
Incident energies	Log-uniform distribution (1GeV-1TeV)
N. of events	100,000

Generative Models

Simplest Example: Box-Muller Method

Gaussian
Distribution

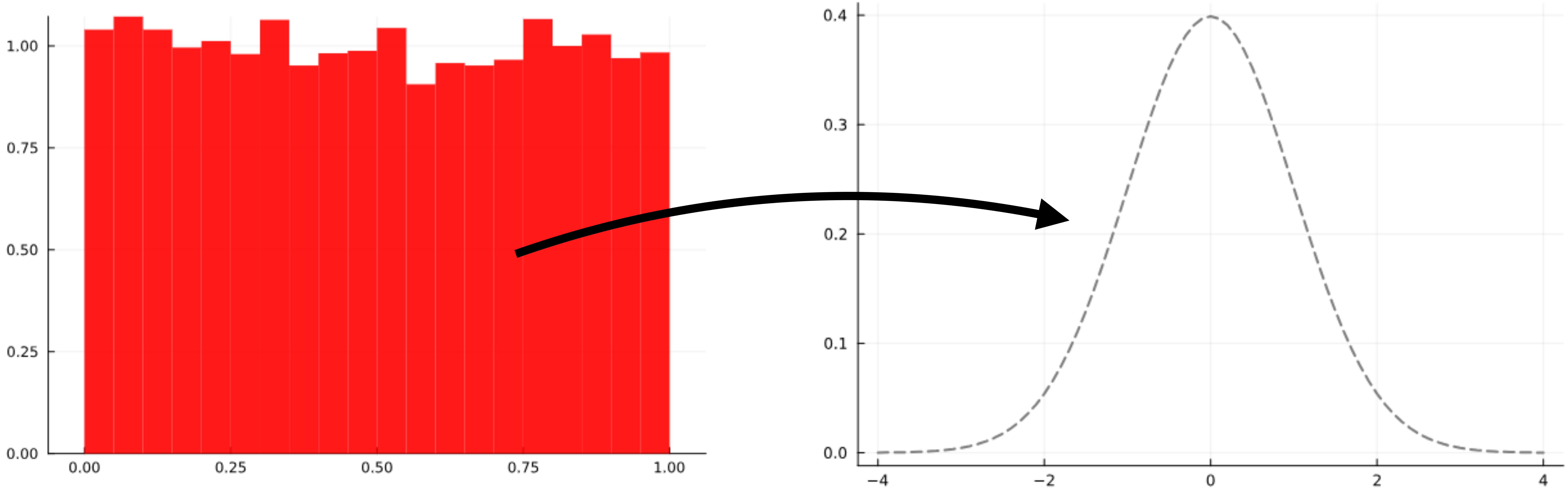


Generative Models

Simplest Example: Box-Muller Method

Uniform
Distribution

Gaussian
Distribution



Generative Models

Simplest Example: Box-Muller Method

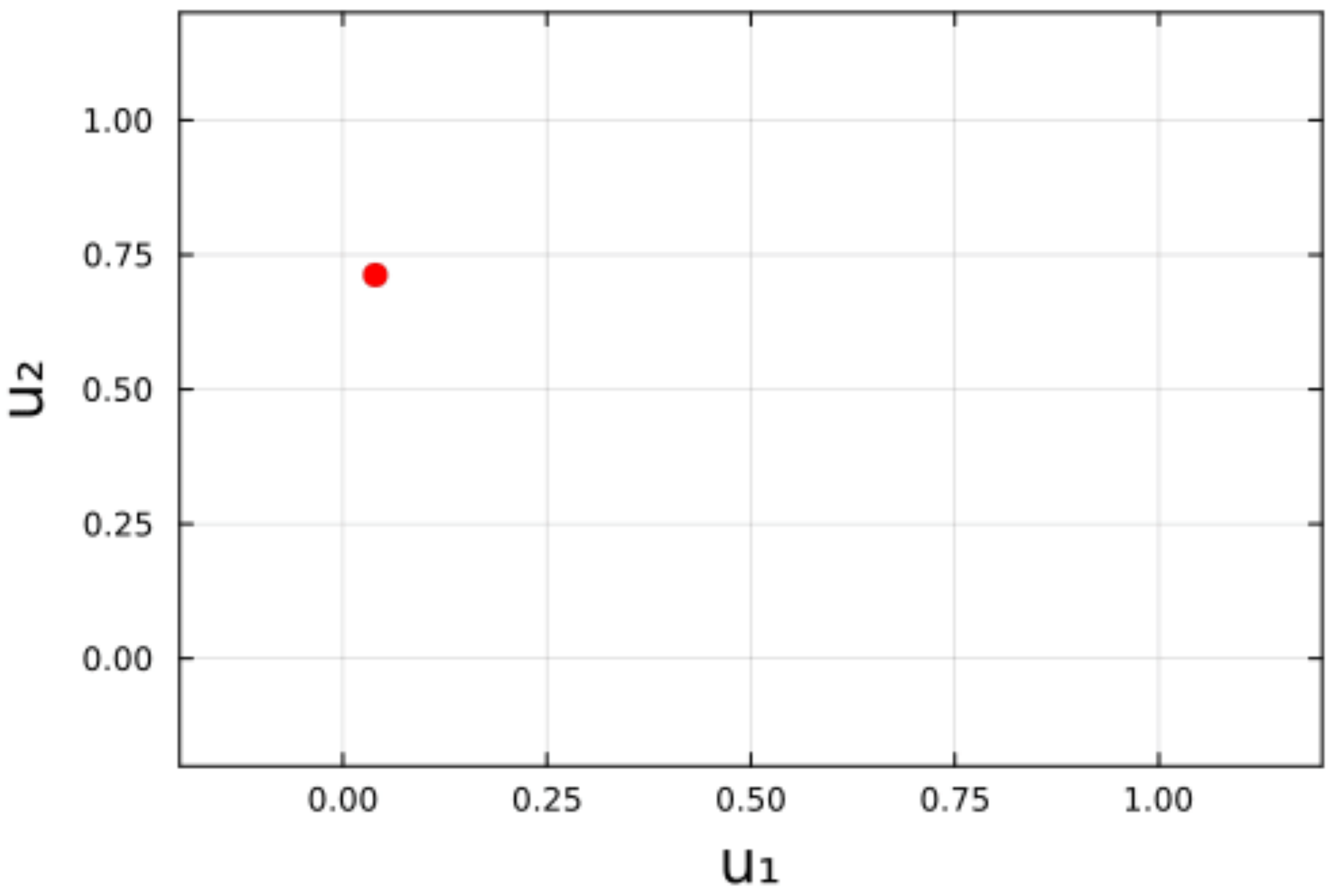
Recipe:

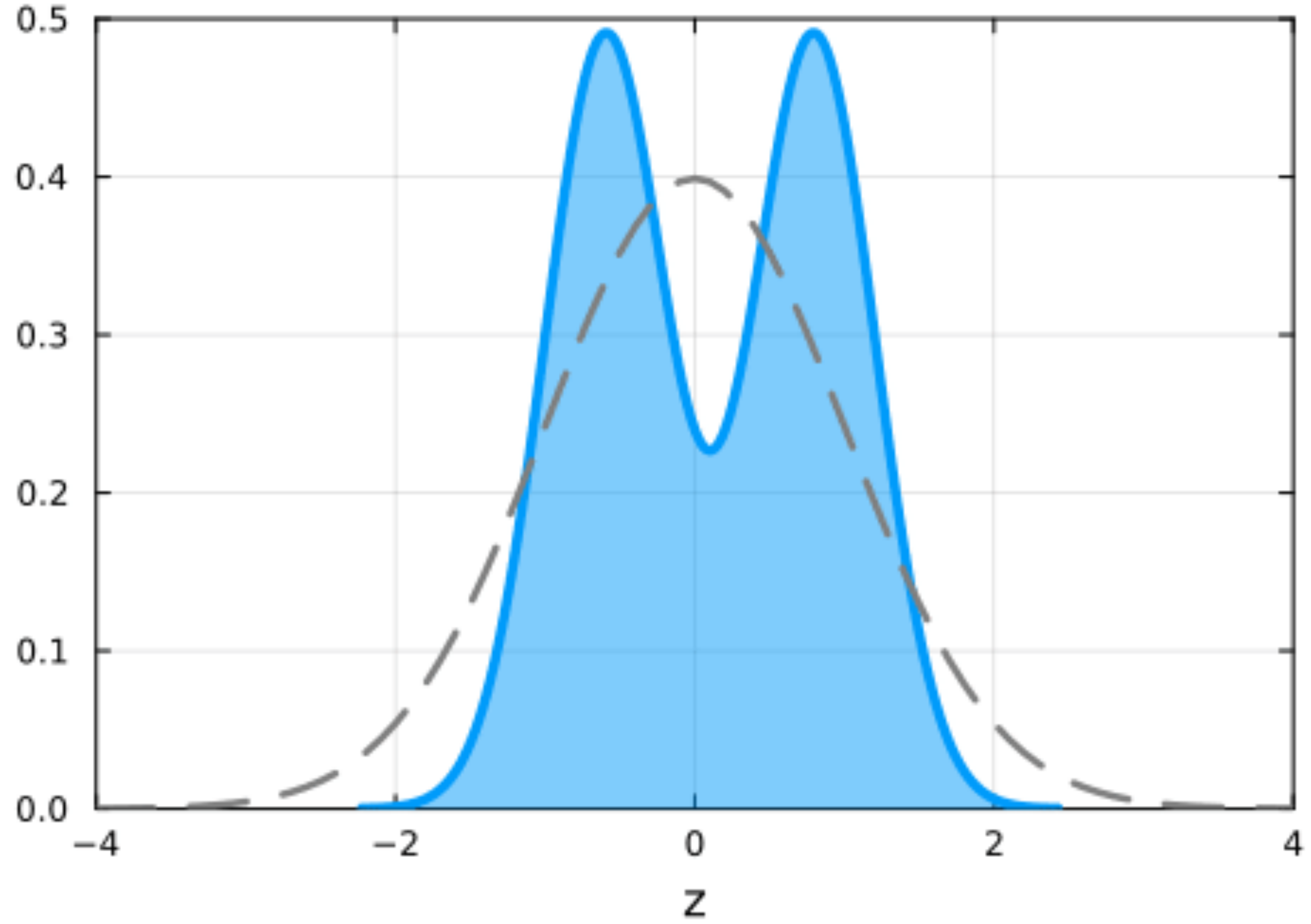
1. Generate two **uniformly** independent, identically distributed random numbers U_1 and U_2.

2. Substitute in:

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

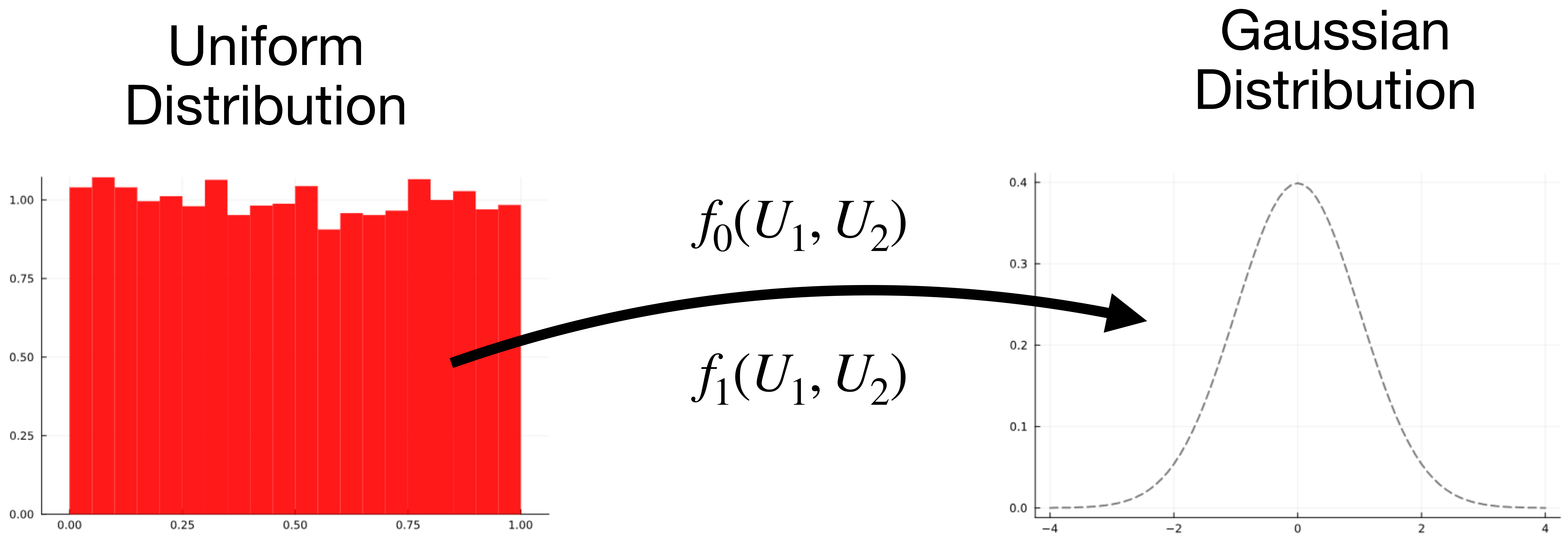
$$Z_1 = f_1(U_1, U_2) = \sqrt{-2 \ln U_1} \sin(2\pi U_2)$$





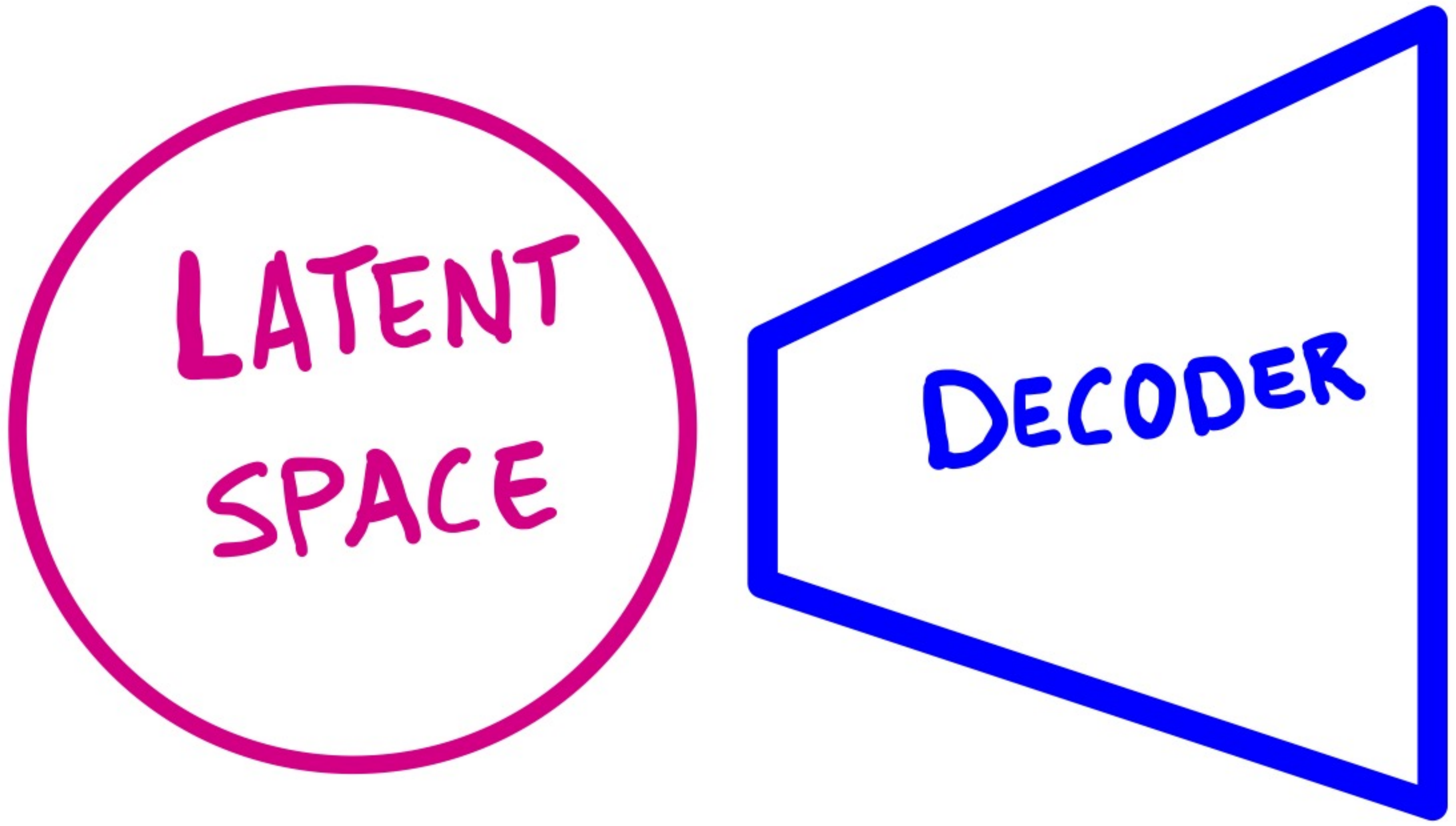
Generative Models

Simplest Example: Box-Muller Method

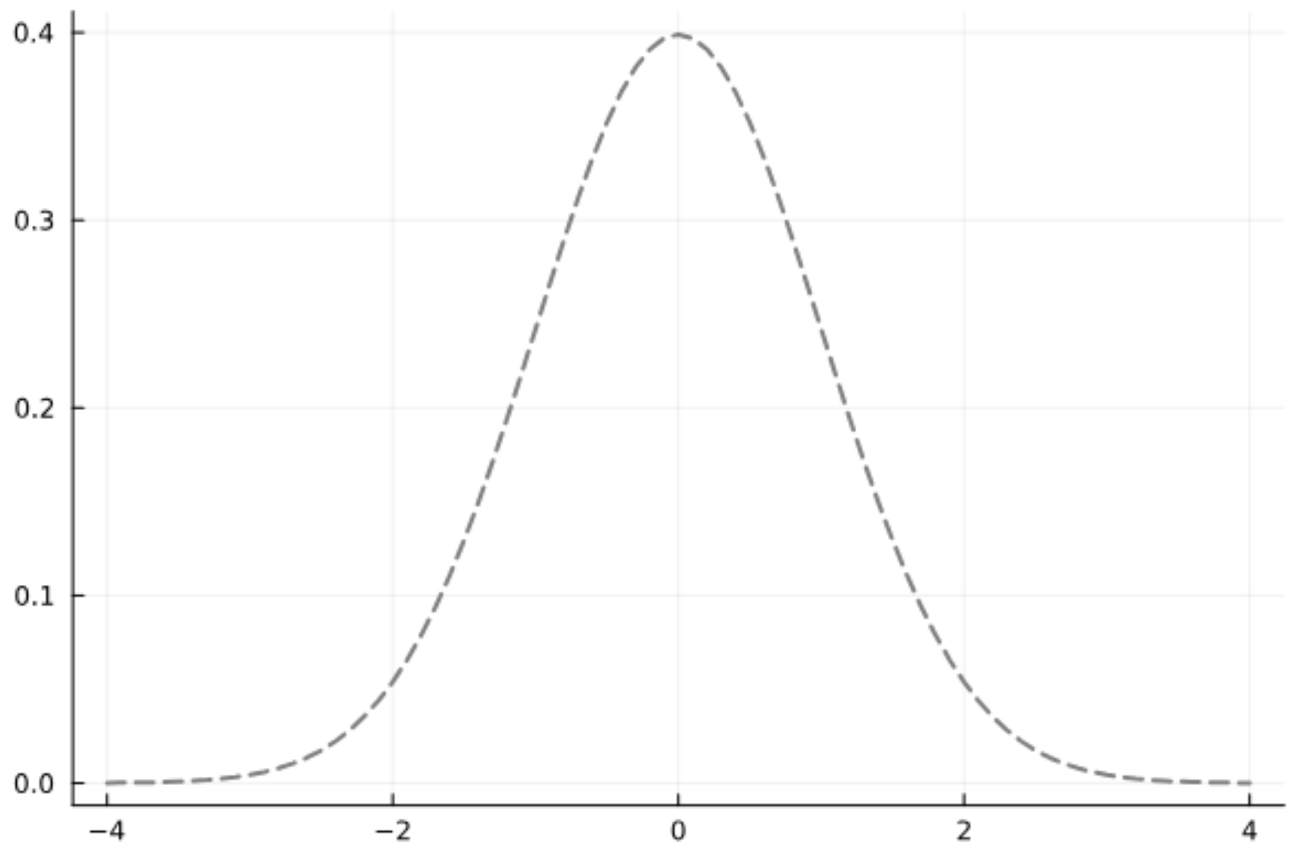


Generative Models

Uniform
Distribution

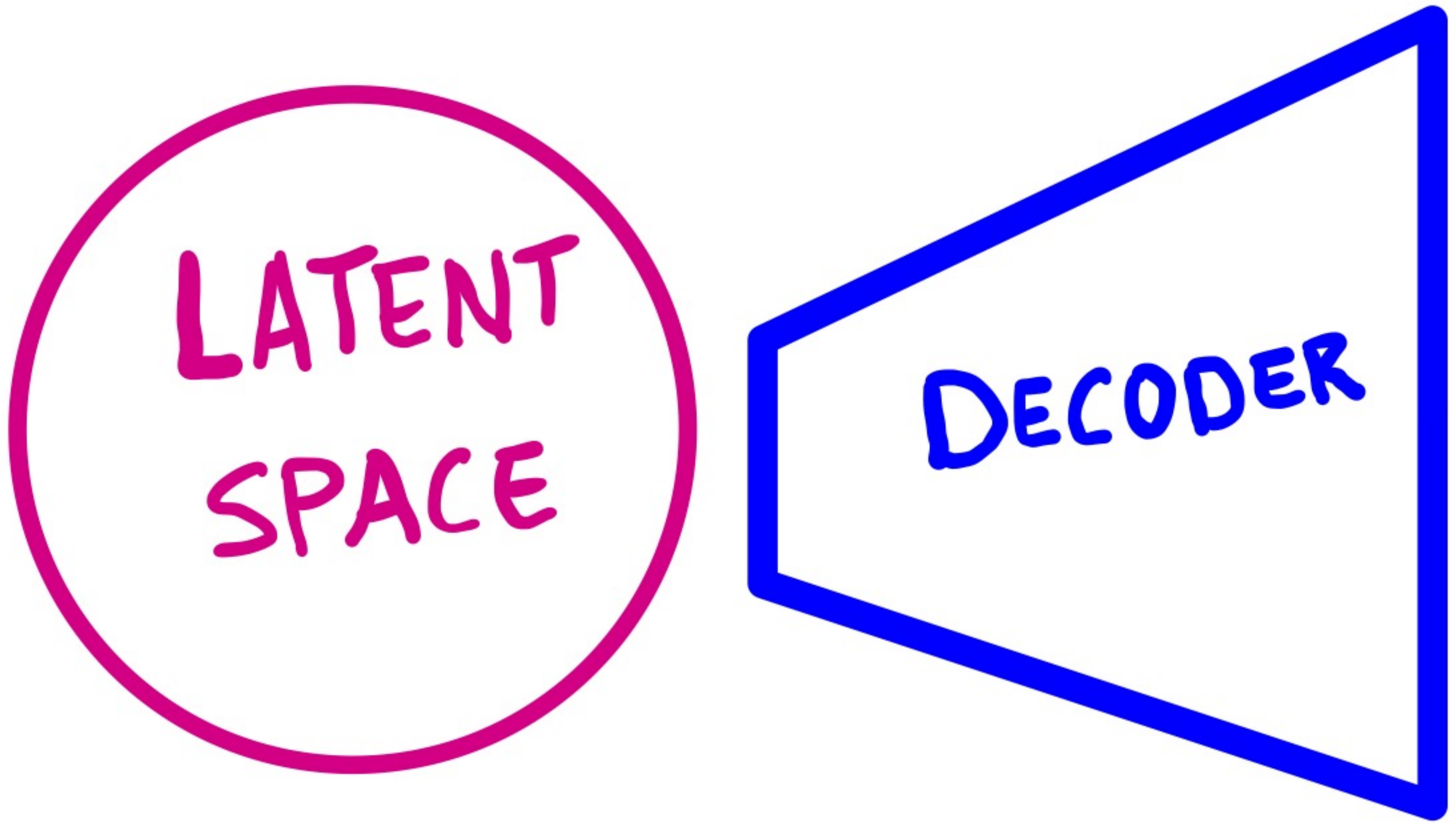


Gaussian
Distribution

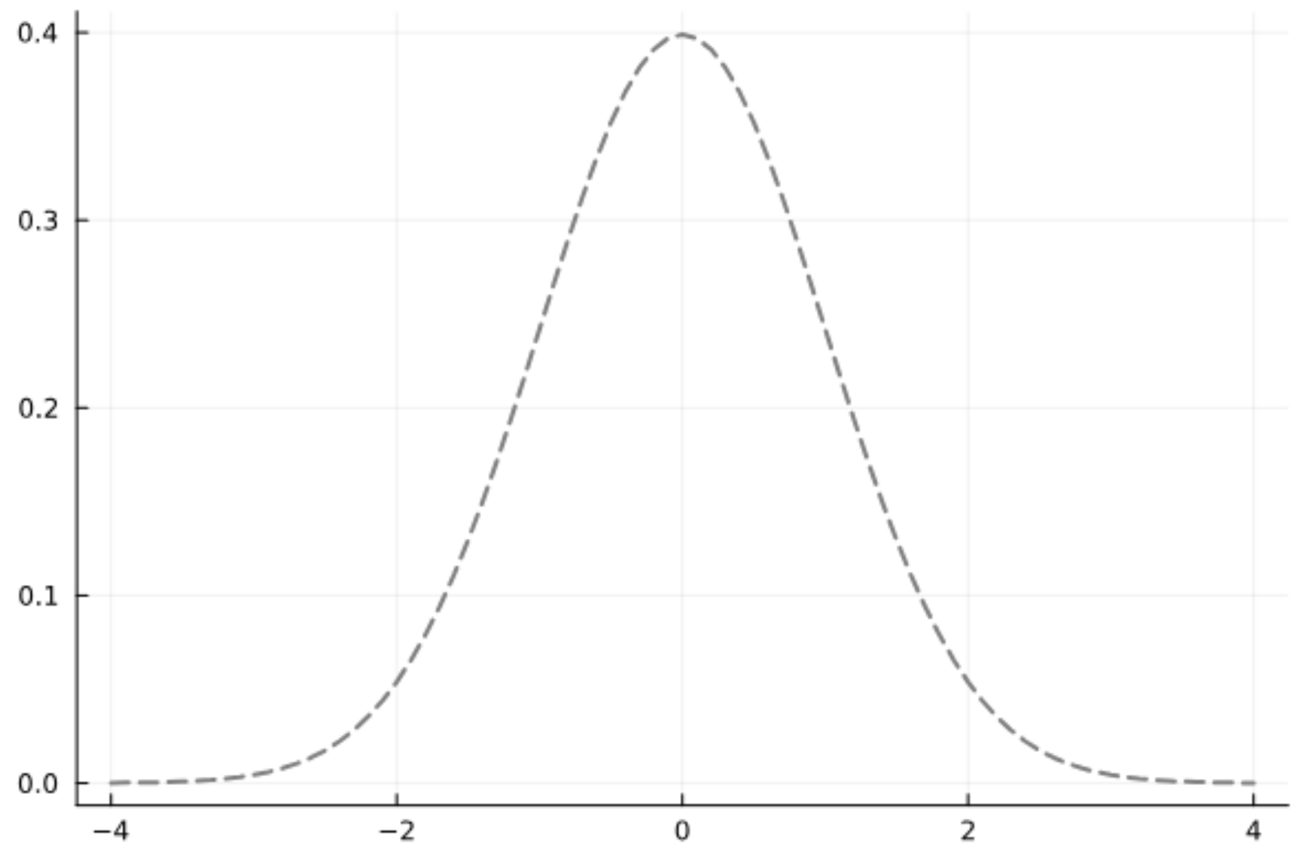


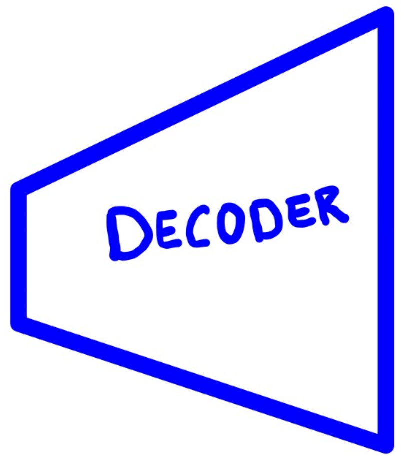
Generative Models

Uniform Distribution



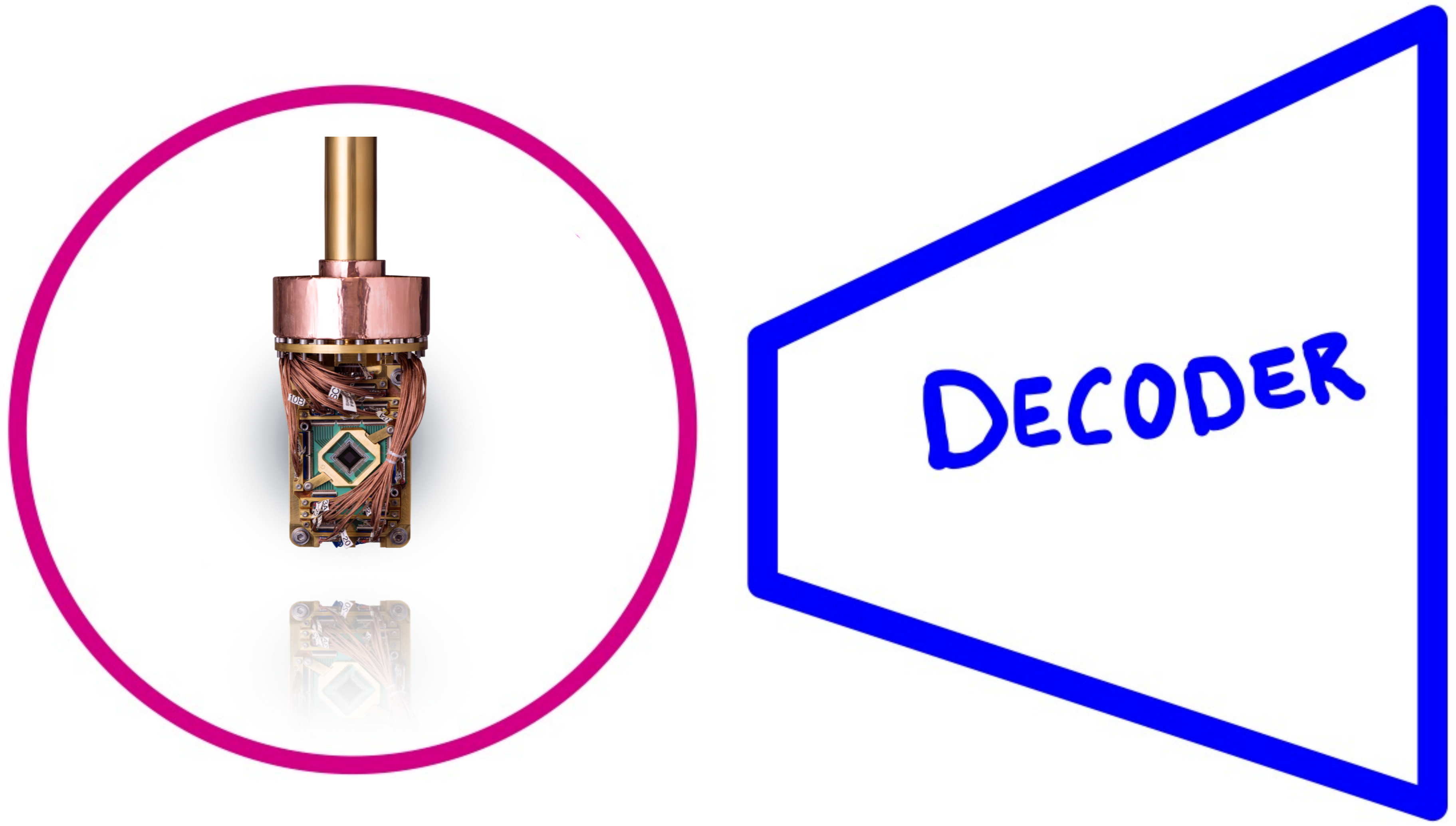
Gaussian Distribution

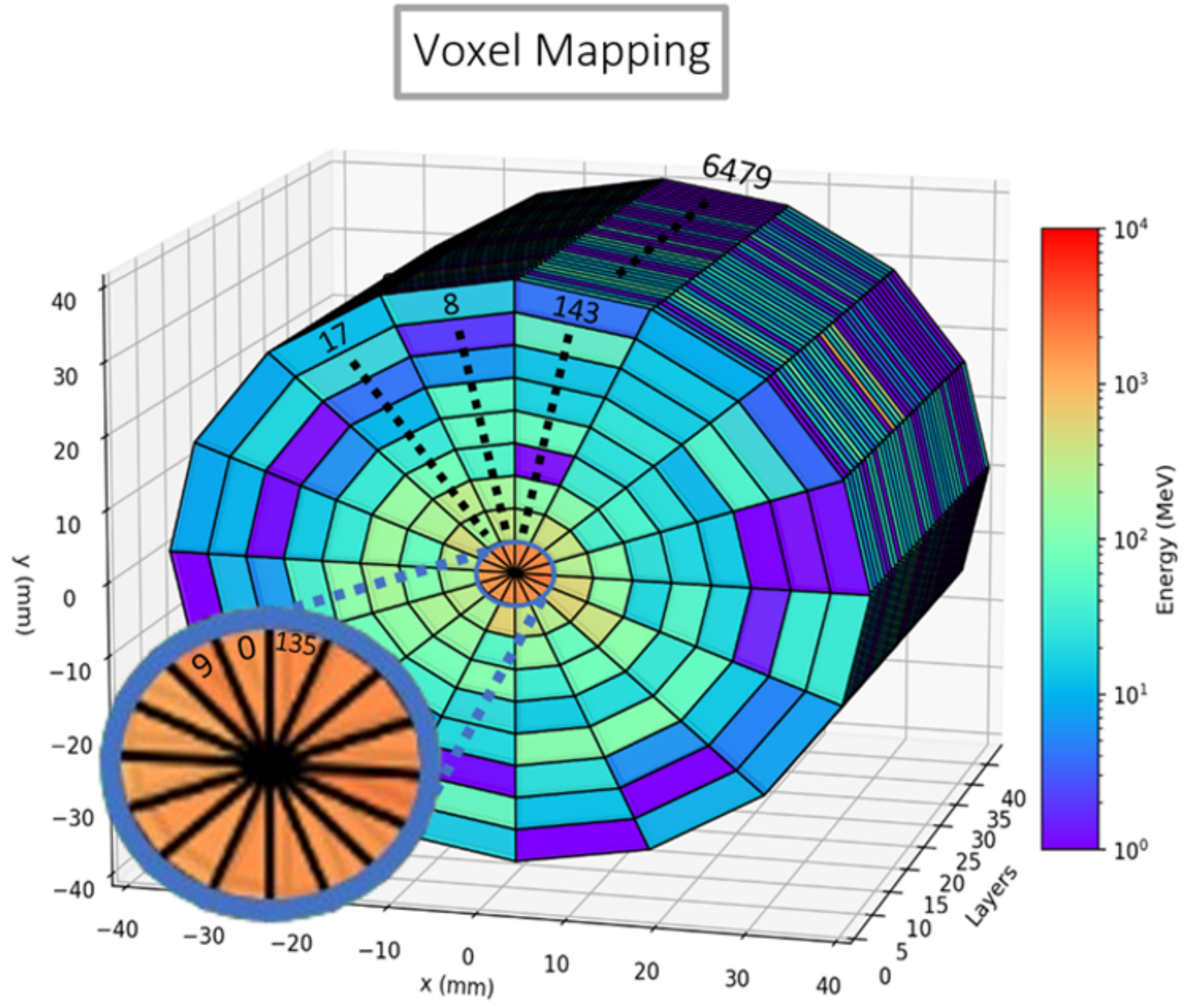


$\text{argmax}_{\theta}\langle \ln$  $\rangle$

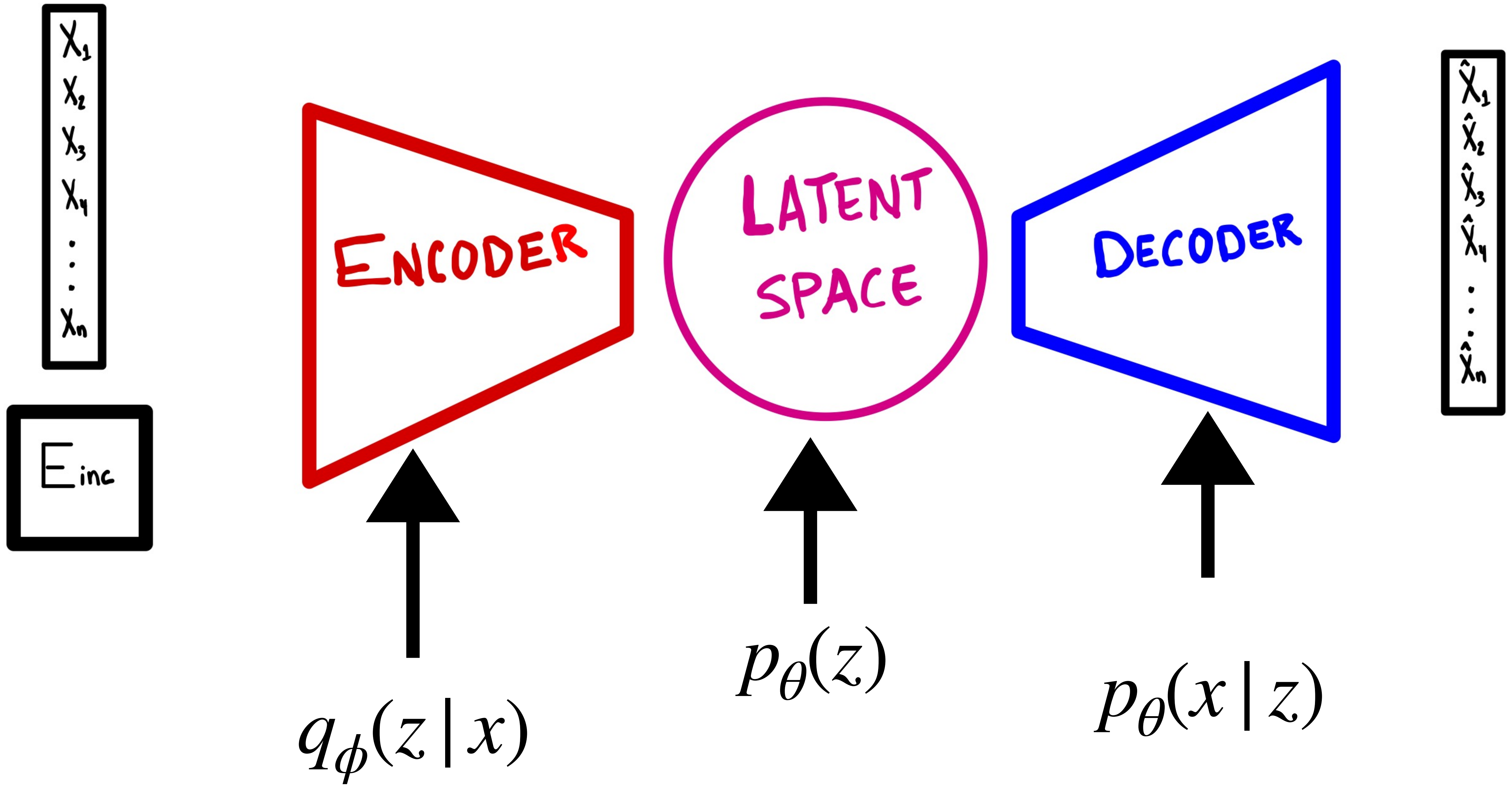
Generative Models

For particle-calorimeter interactions + quantum-assisted





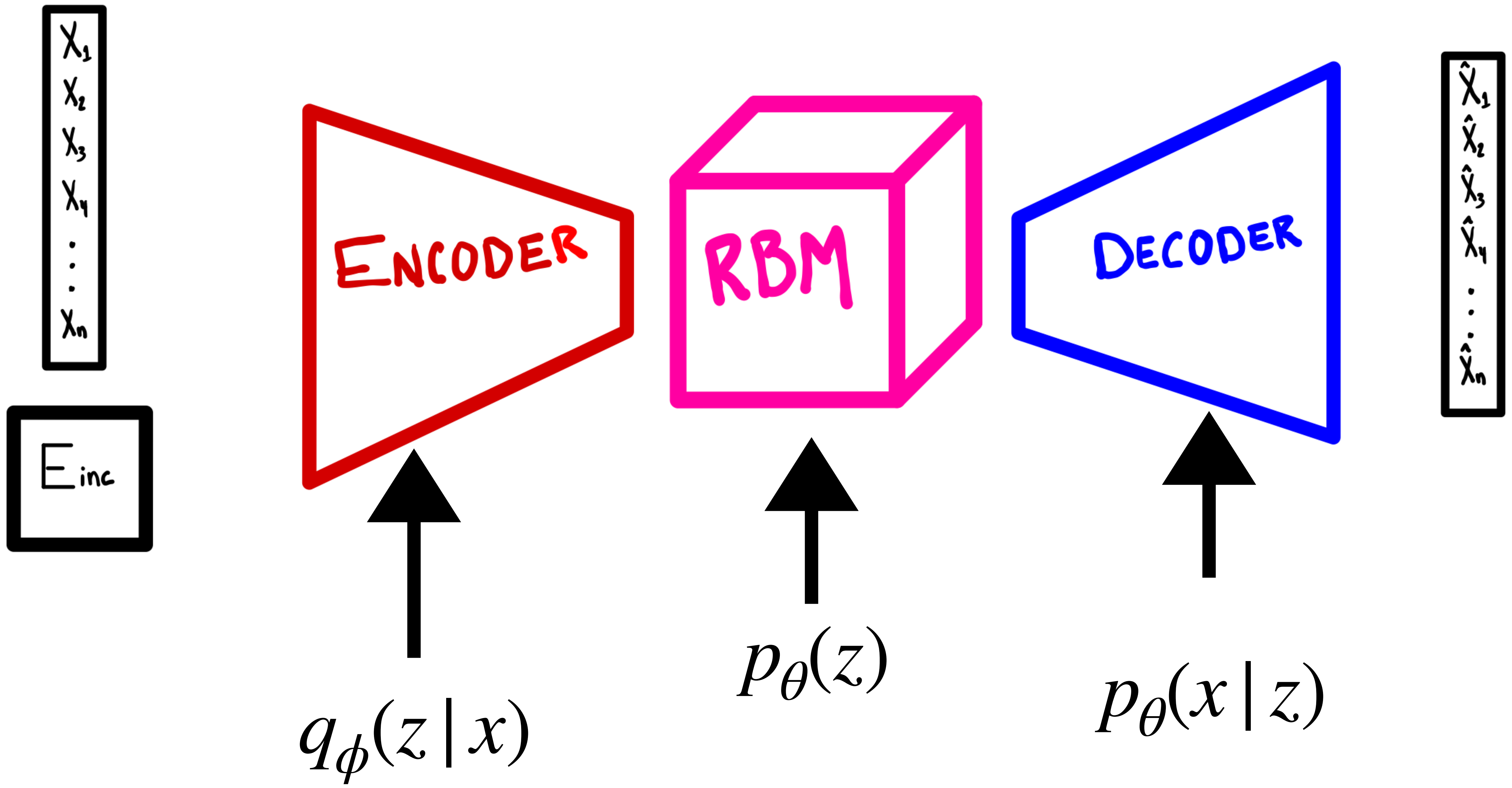
Variational Autoencoders (VAE)



- Easy to train.
- Average performance.
- *Legacy* VAE assumes a Gaussian prior.

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

VAE + Restricted Boltzmann Machine

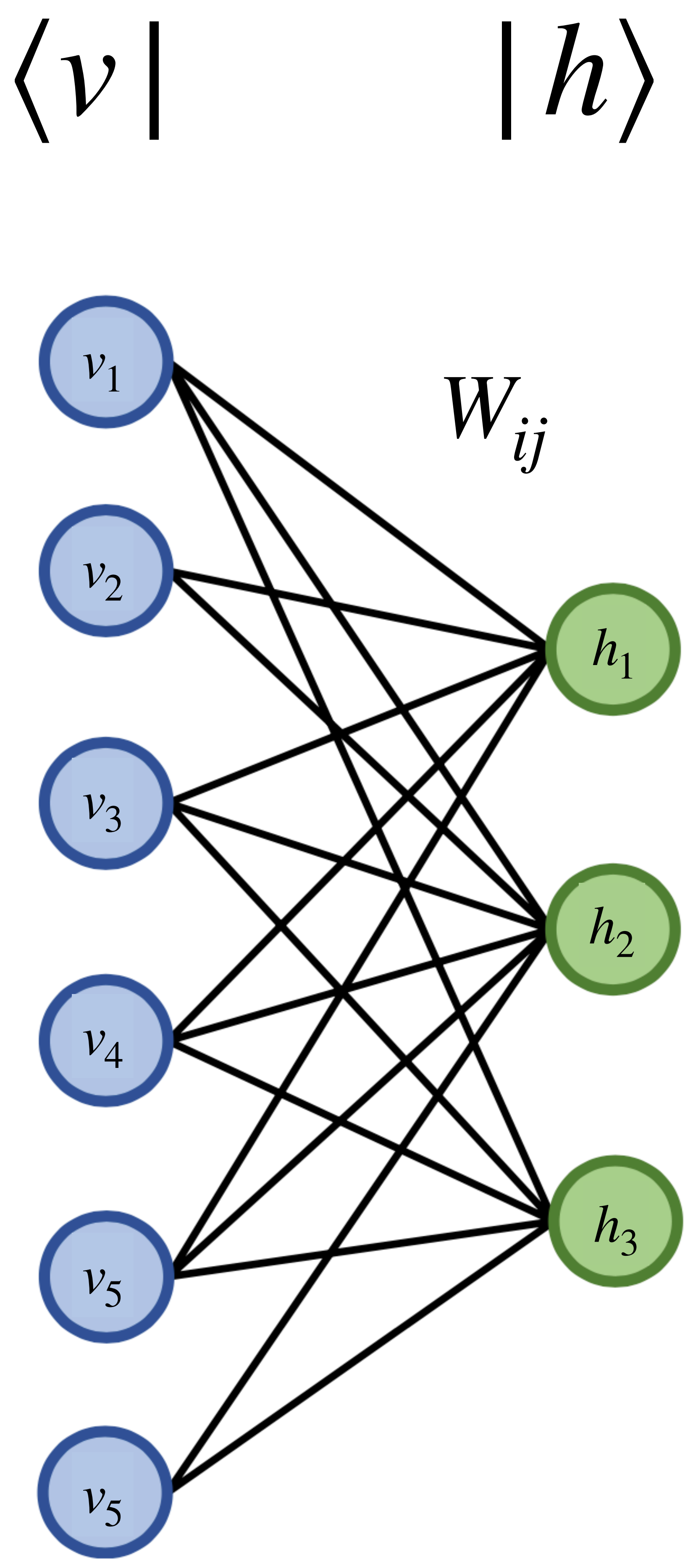


$$\mathscr{L}_{\phi,\theta}(x) = \underbrace{\langle \ln p_\theta(x|z) \rangle_{q_\phi(z|x)}}_{\text{Reconstruction}} - \underbrace{\left\langle \ln \frac{q_\phi(z|x)}{p_\theta(z)} \right\rangle_{q_\phi(z|x)}}_{\text{Regularizer}}$$

- Replace Gaussian prior with Boltzmann prior.
- Universal approximator.
- However, this comes at a cost.

Restricted Boltzmann Machine

Basics



Suppose a data set $\{v^\alpha\}_{\alpha=1}^n$, such that $v_i \in \{0,1\}$.

I) An RBM will fit a Boltzmann distribution, $p(v)$, to the data set.

II) The fitting is done by maximizing the log-likelihood, $\ln p(v)$.

III) RBMs are composed by a two-partite graph, where **v** denotes the visible layer and **h** the hidden layer.

$$p(v) = \frac{\sum_h \exp(-E(v,h))}{Z}$$

Boltzmann Dist

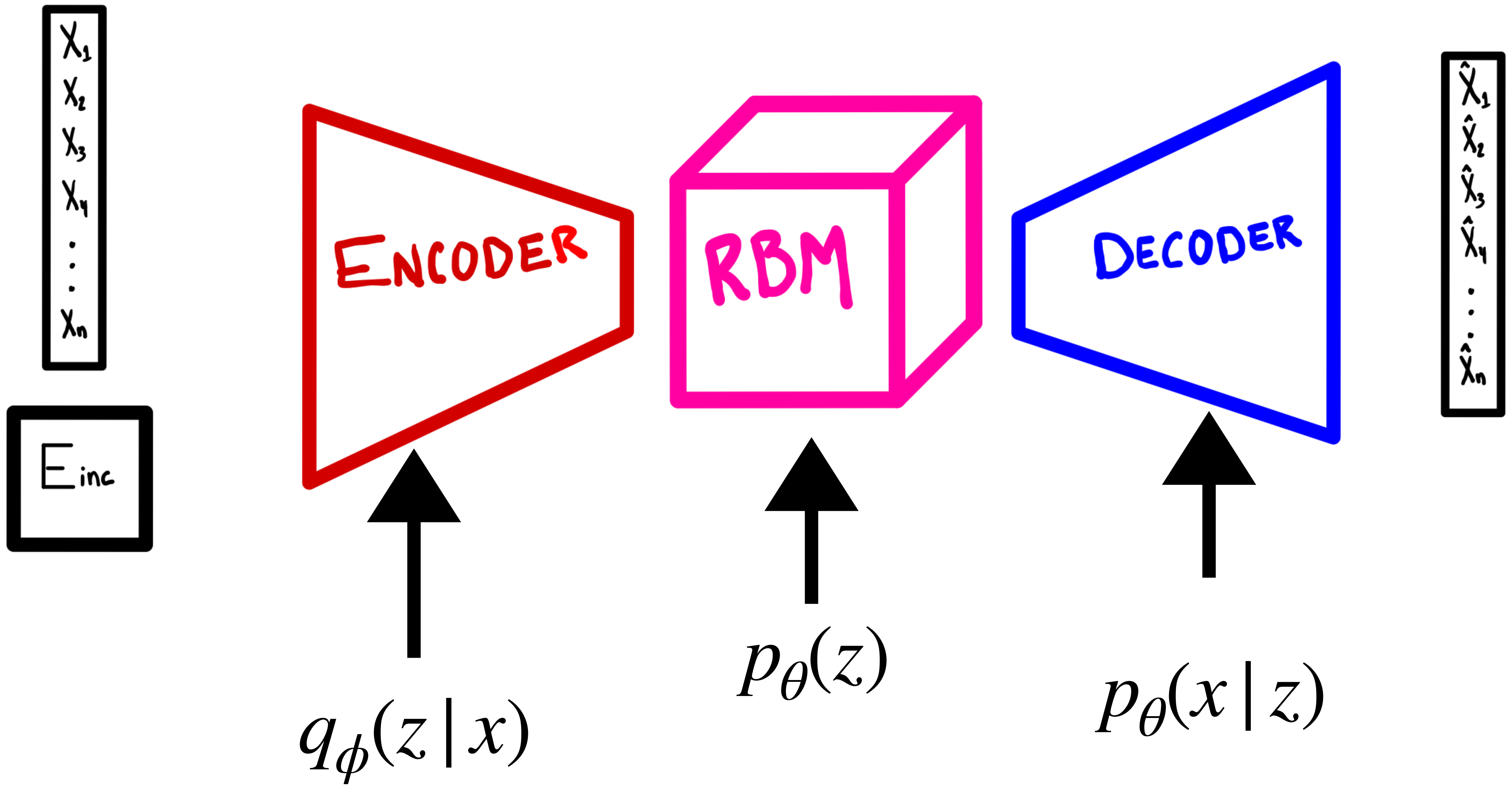
$$E(v,h) = -\sum_{i=1}^{n_v} v_i a_i - \sum_{j=1}^{n_h} b_j h_j - \sum_{i,j} v_i W_{ij} h_j$$

Energy

$$Z(W,a,b,\beta=1) = \sum_{v',h'} \exp(-E(v',h'))$$

Partition Function

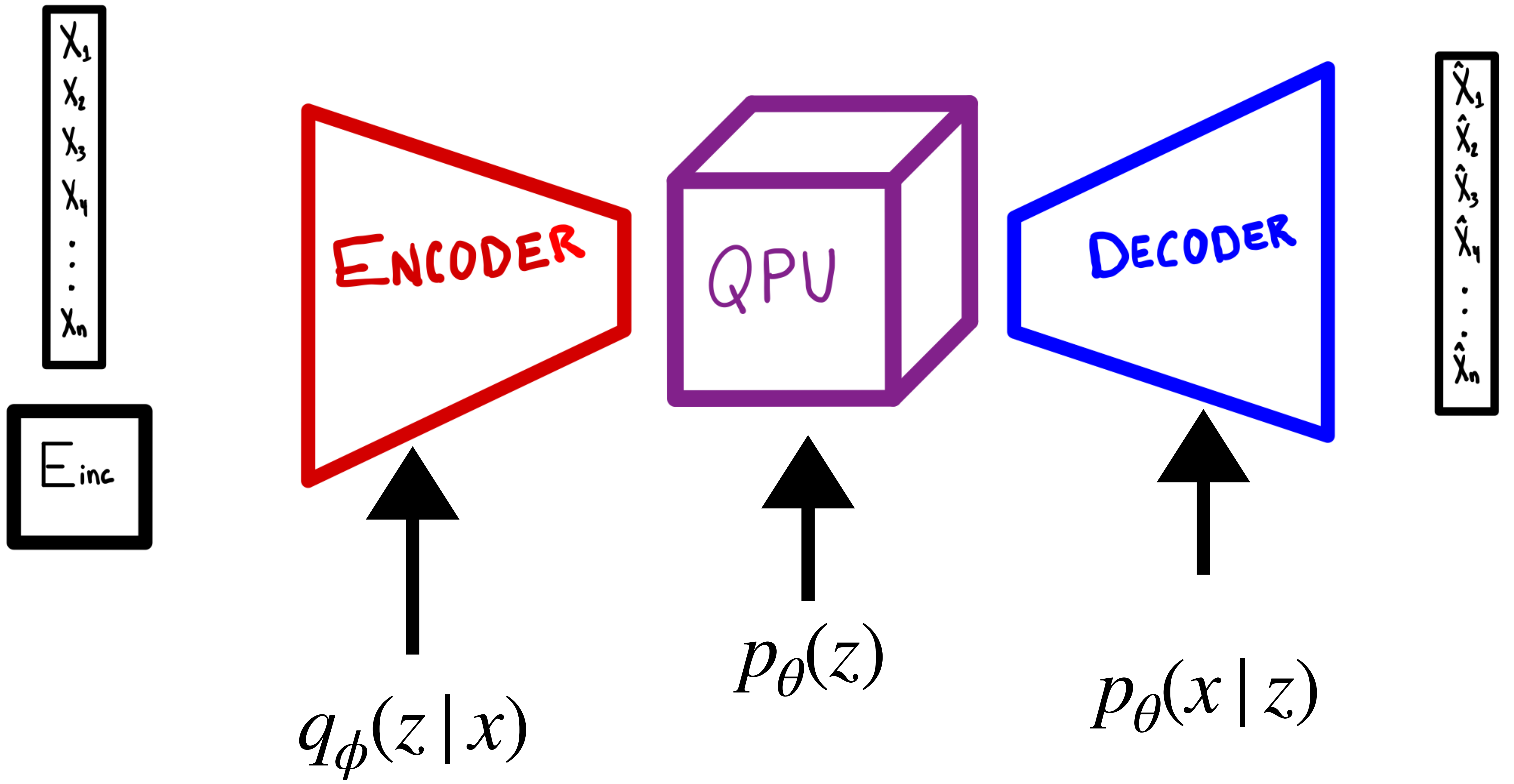
VAE + Restricted Boltzmann Machine



- ✦Replace Gaussian prior with Boltzmann prior.
- ✦Universal approximator.
- ✦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



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

- Replace Gaussian prior with Boltzmann prior.
- Universal approximator.
- However, this comes at a cost.
- But we might be able to avoid Gibbs sampling…

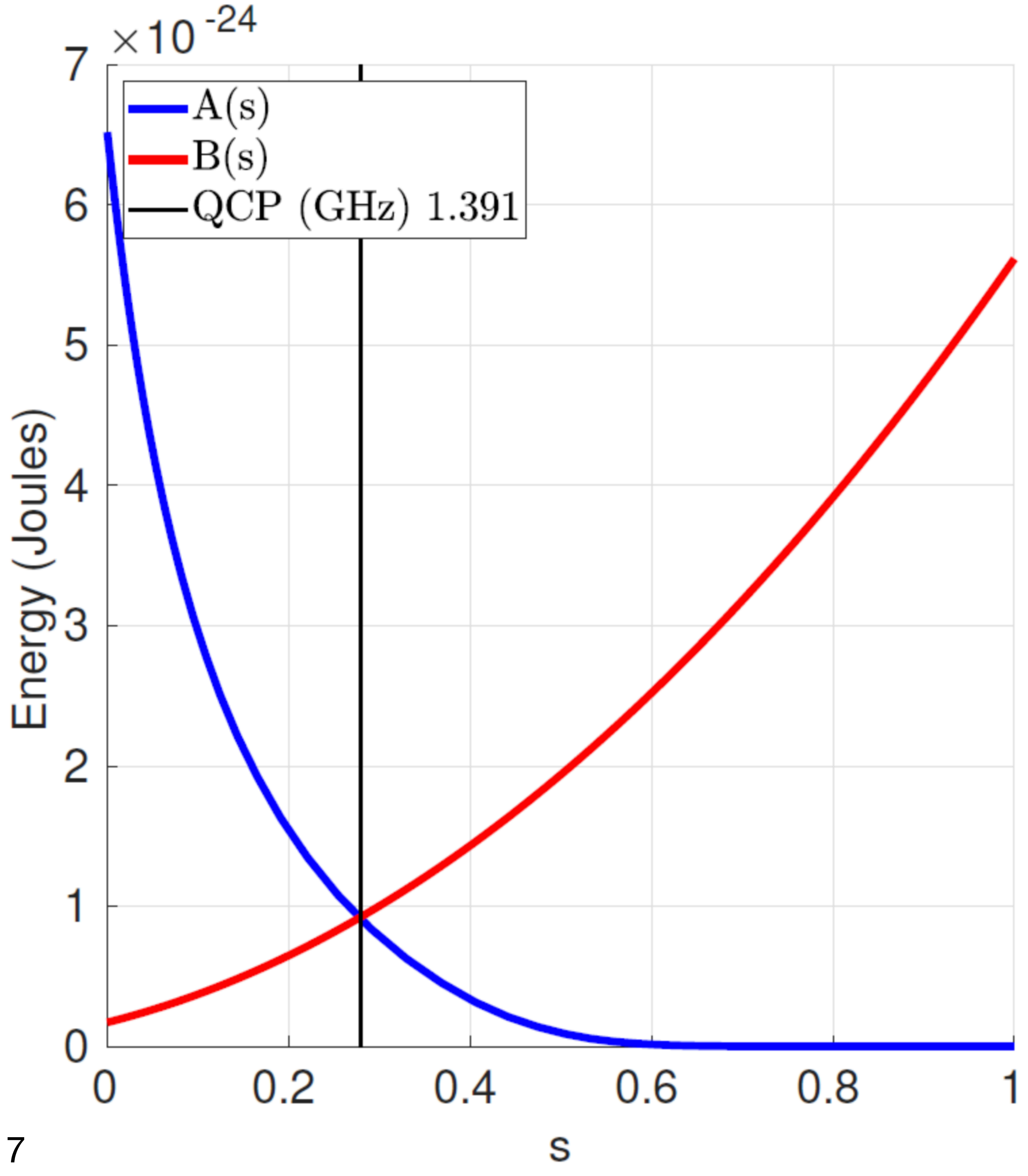
Quantum Annealer

Basics

- ✦An array of **superconducting flux quantum bits** with **programmable spin–spin couplings** and **self-fields**.
- ✦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***.

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

H_1 $\qquad$ H_0



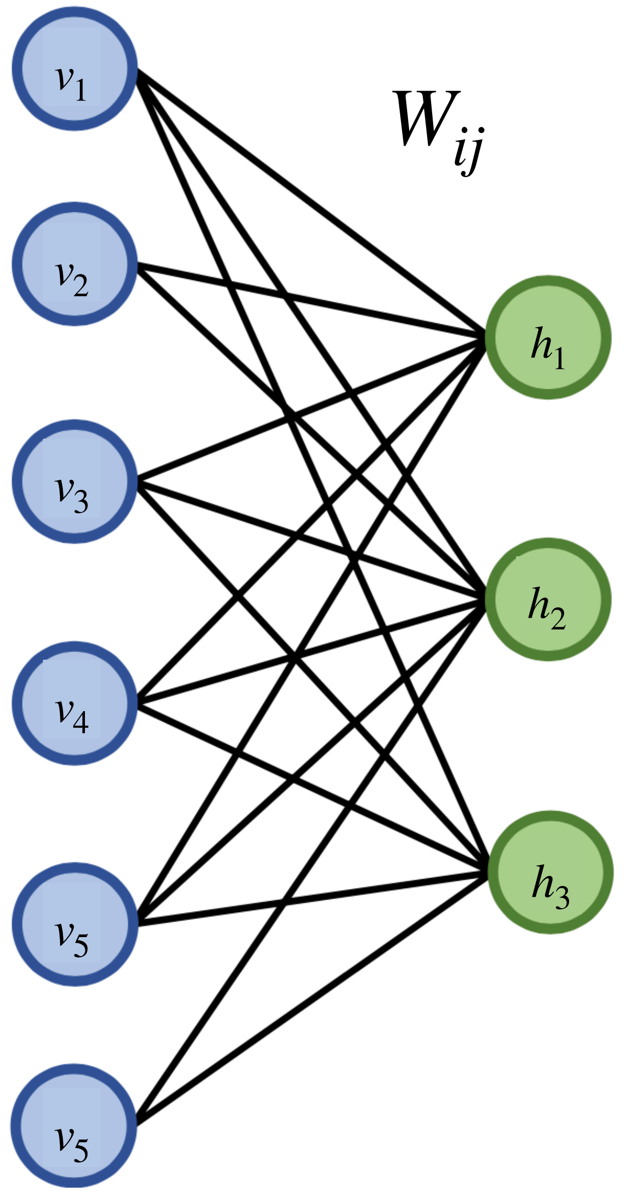
Amin MH. Searching for quantum speedup in quasistatic quantum annealers. Physical Review A. 2015 Nov 19;92(5):052323.

Quantum Annealer

Topologies

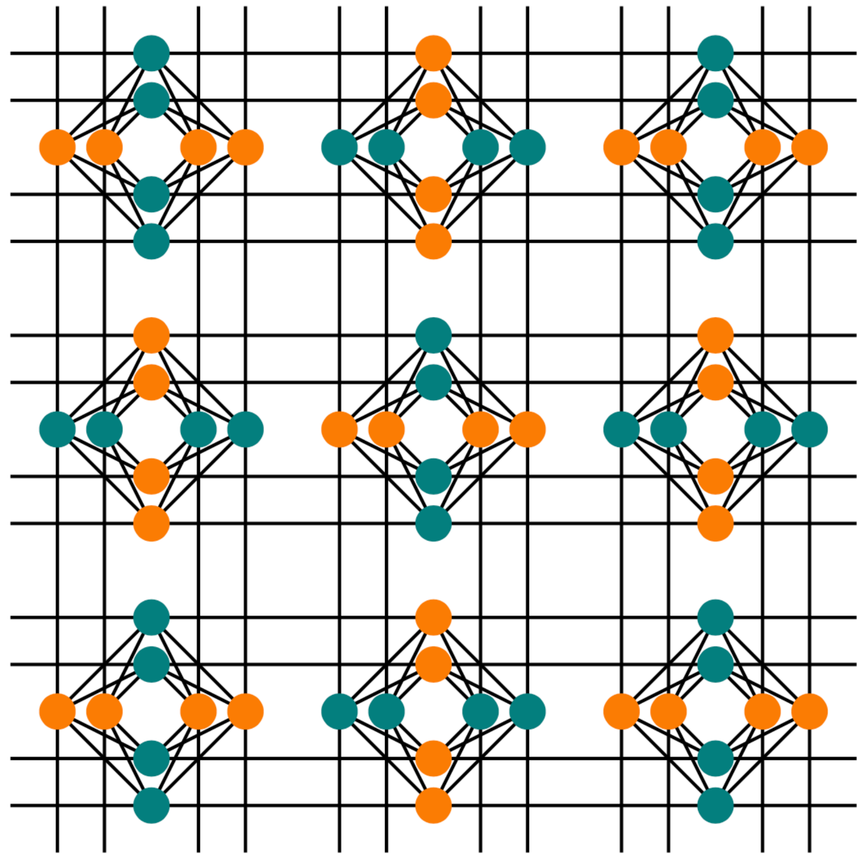
Fully Connected RBM

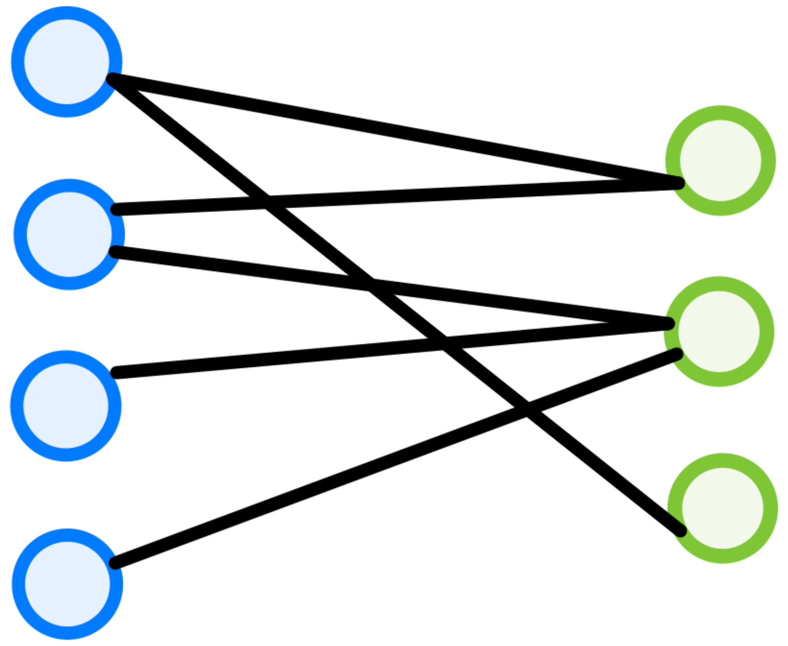
2-partite Graph



Chimera QA

2-partite Graph

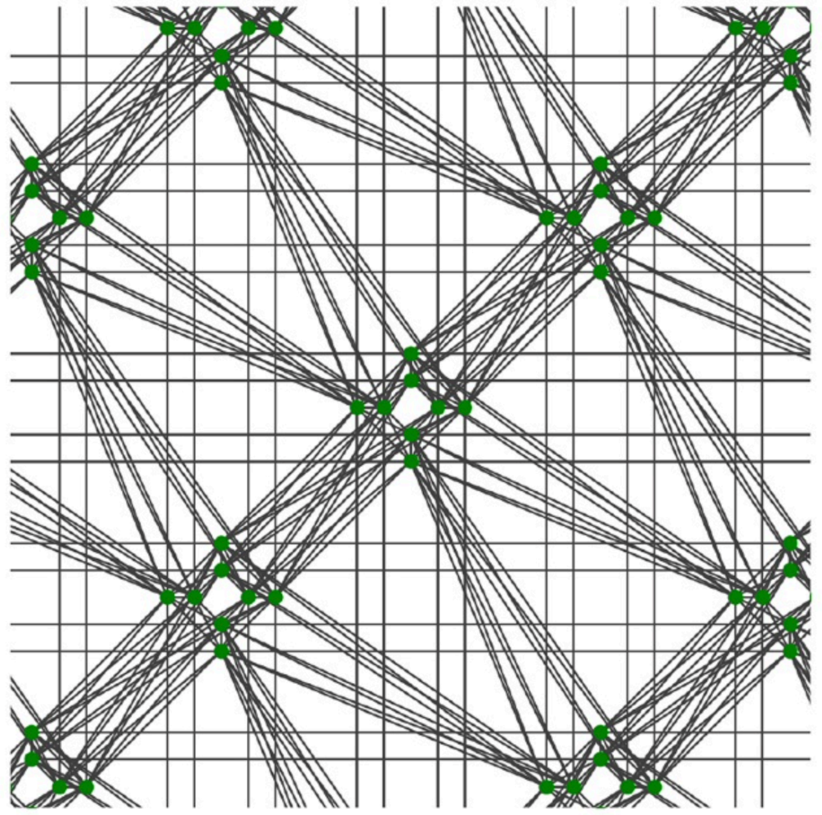




Pegasus QA

4-partite Graph

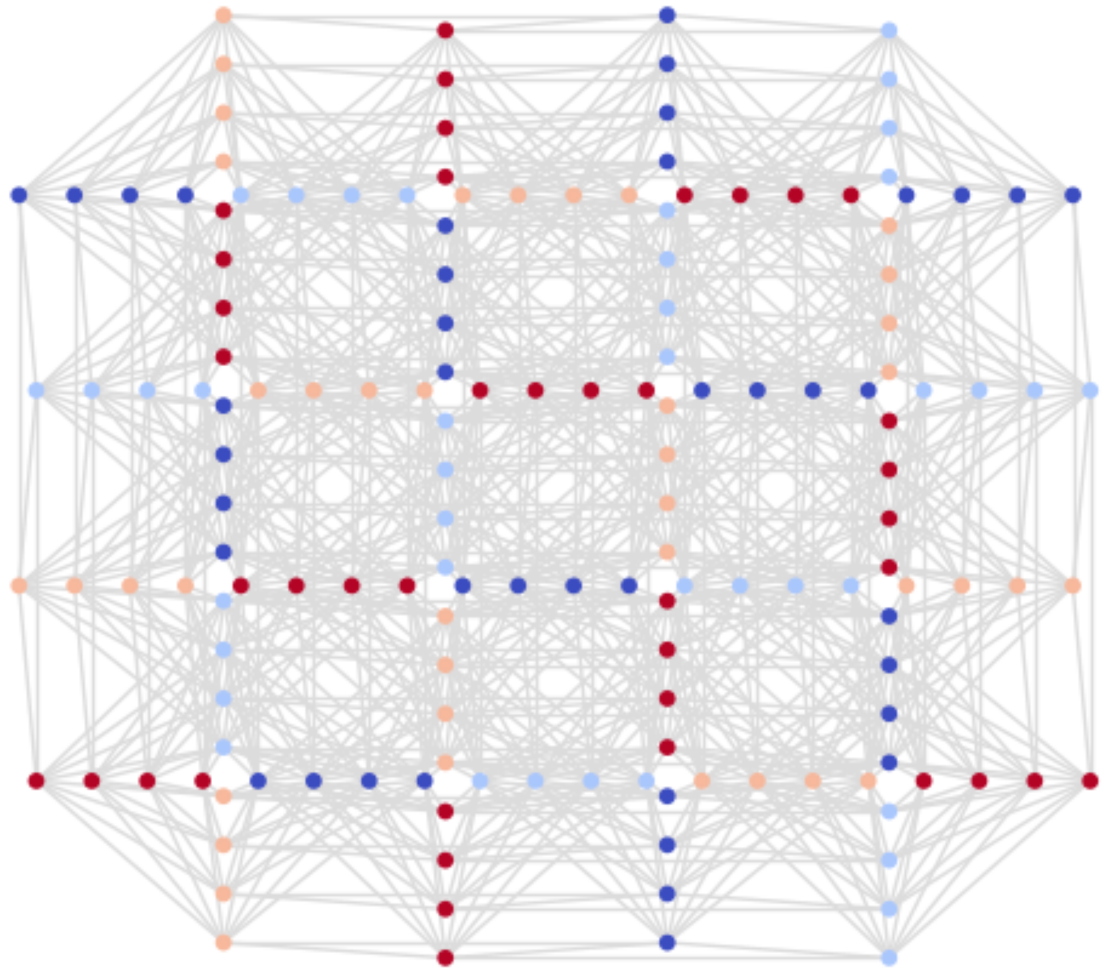
Max coord num=15

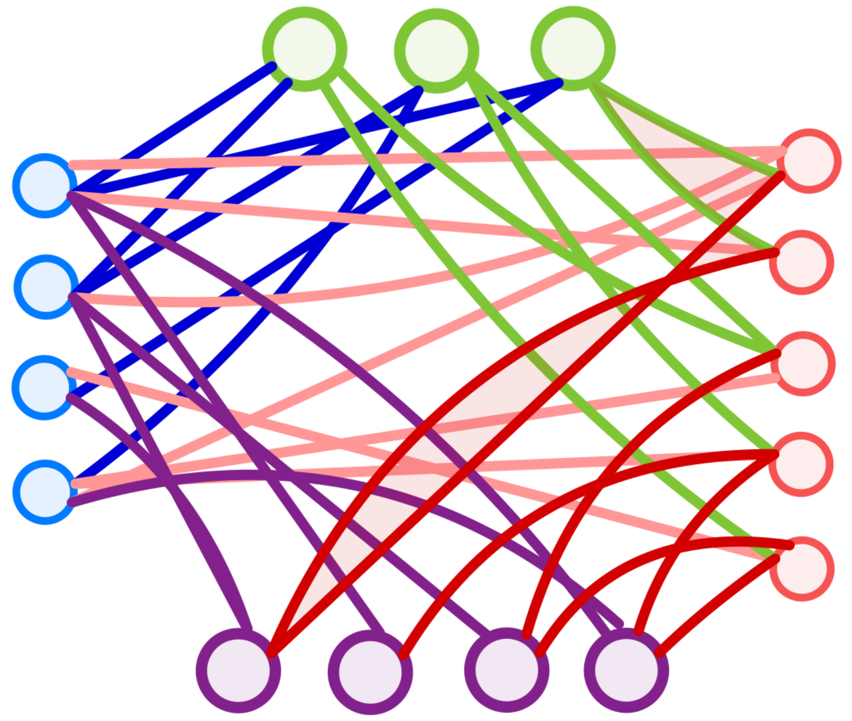


Zephyr QA

4-partite Graph

Max coord num=20



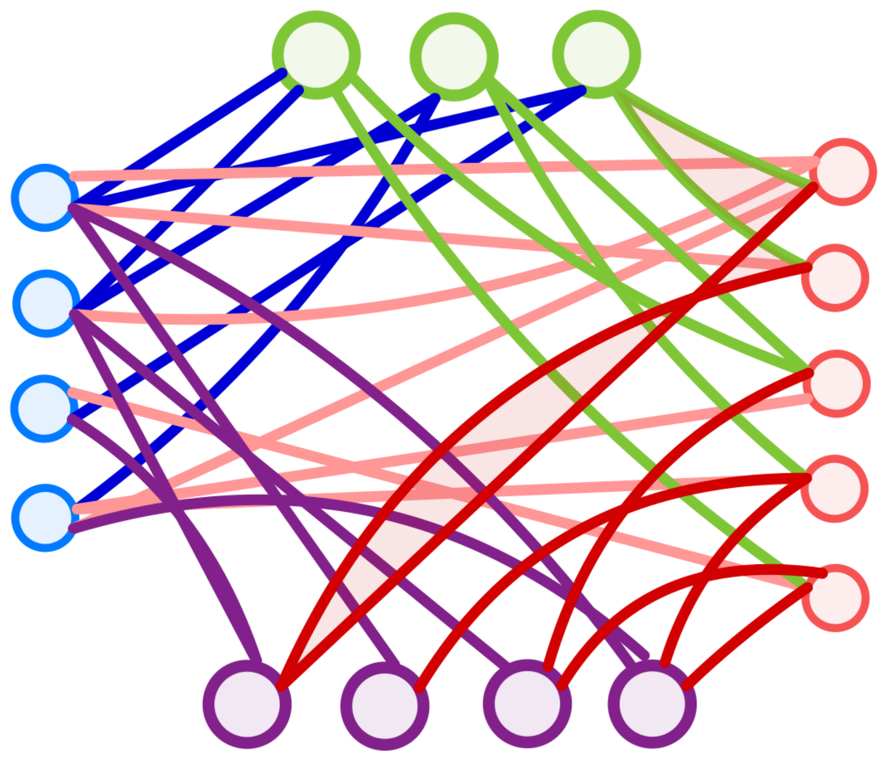


QPU conditioning

$$\sigma_z^{(i)} = \begin{cases} 1 & h_i < 0 \text{ and } |h_i| > \sum_j |J_{ij}| \\ -1 & h_i > 0 \text{ and } |h_i| > \sum_j |J_{ij}| \end{cases}$$

$k = 1,\ldots,302$ (Condition partition)

Label → f → $\sigma_z^{(k)} \in \{-1,1\}^{302}$ → g → $h_k \in \{-M, M\}^{302}$



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

QPU conditioning

$$H = -\frac{1}{2}\sum_i \Delta_q(\Phi_{CCJJ}(s))\hat{\sigma}_x^{(i)} + \sum_i h_i \,|\, I_p(\Phi_{CCJJ}(s)) \,|\, \Phi_i^x(s)\hat{\sigma}_z^{(i)} + \sum_{i>j} J_{ij} M_{AFM} I_p(\Phi_{CCJJ}(s))^2 \hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}$$

Δ_q: Energy difference between $\hat{\sigma}_x^i$ states

I_p: Current magnitude in qubit loop

M_{AFM}: Max mutual inductance

$\Phi_{CCJJ}(s)$: External flux applied J-J compound

$\Phi_i^x(s)$: External flux applied to the qubits

https://docs.dwavesys.com/docs/latest/c_qpu_annealing.html

QPU conditioning

$$H = -\frac{1}{2}\sum_i \underbrace{\Delta_q(\Phi_{CCJJ}(s))}_{A(s)}\hat{\sigma}_x^{(i)} + \sum_i h_i |I_p(\Phi_{CCJJ}(s))| \Phi_i^x(s)\hat{\sigma}_z^{(i)} + \sum_{i>j} J_{ij}\underbrace{M_{AFM}I_p(\Phi_{CCJJ}(s))^2}_{B(s)/2}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}$$

Δ_q: Energy difference between $\hat{\sigma}_x^i$ states

I_p: Current magnitude in qubit loop

M_{AFM}: Max mutual inductance

$\Phi_{CCJJ}(s)$: External flux applied J-J compound

$\Phi_i^x(s)$: External flux applied to the qubits

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

https://docs.dwavesys.com/docs/latest/c_qpu_annealing.html

QPU conditioning

$$H = -\frac{1}{2}\sum_i \underbrace{\Delta_q(\Phi_{CCJJ}(s))}_{A(s)}\hat{\sigma}_x^{(i)} + \sum_i h_i |I_p(\Phi_{CCJJ}(s))| \Phi_i^x(s)\hat{\sigma}_z^{(i)} + \sum_{i>j} J_{ij}\underbrace{M_{AFM}I_p(\Phi_{CCJJ}(s))^2}_{B(s)/2}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}$$

$$\Phi_i^x(s) = M_{AFM}I_p(s)$$

Δ_q: Energy difference between $\hat{\sigma}_x^i$ states

I_p: Current magnitude in qubit loop

M_{AFM}: Max mutual inductance

$\Phi_{CCJJ}(s)$: External flux applied J-J compound

$\Phi_i^x(s)$: External flux applied to the qubits

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

https://docs.dwavesys.com/docs/latest/c_qpu_annealing.html

Results

QA temperature estimation

★arXiv:2410.22870

System QA at Temperature $1/\beta_{QA}$

System B at Temperature $1/\beta$

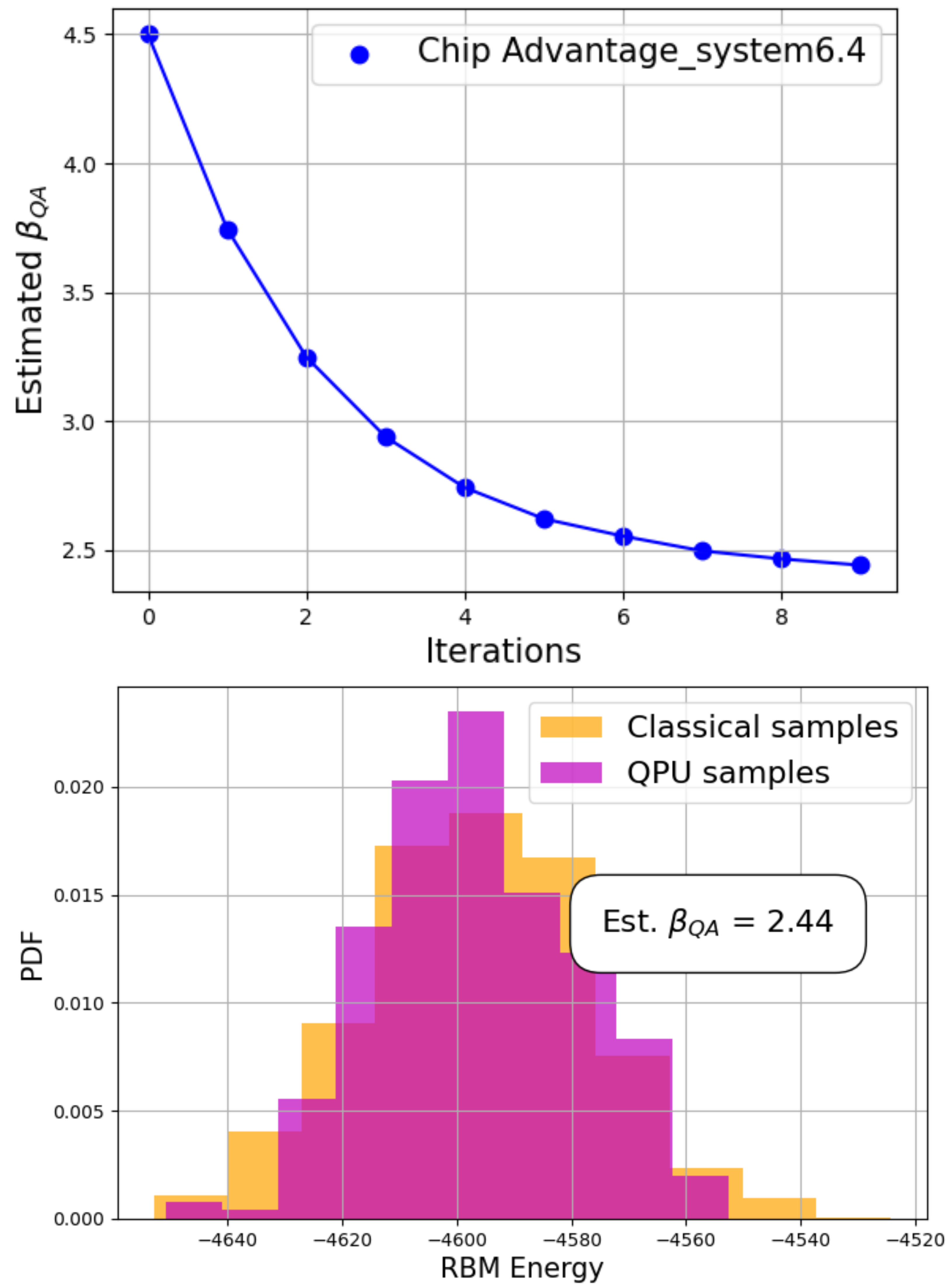
$$P_{QA}(x) = \frac{e^{-\beta_{QA} H(x)}}{Z(\beta_{QA})}$$

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

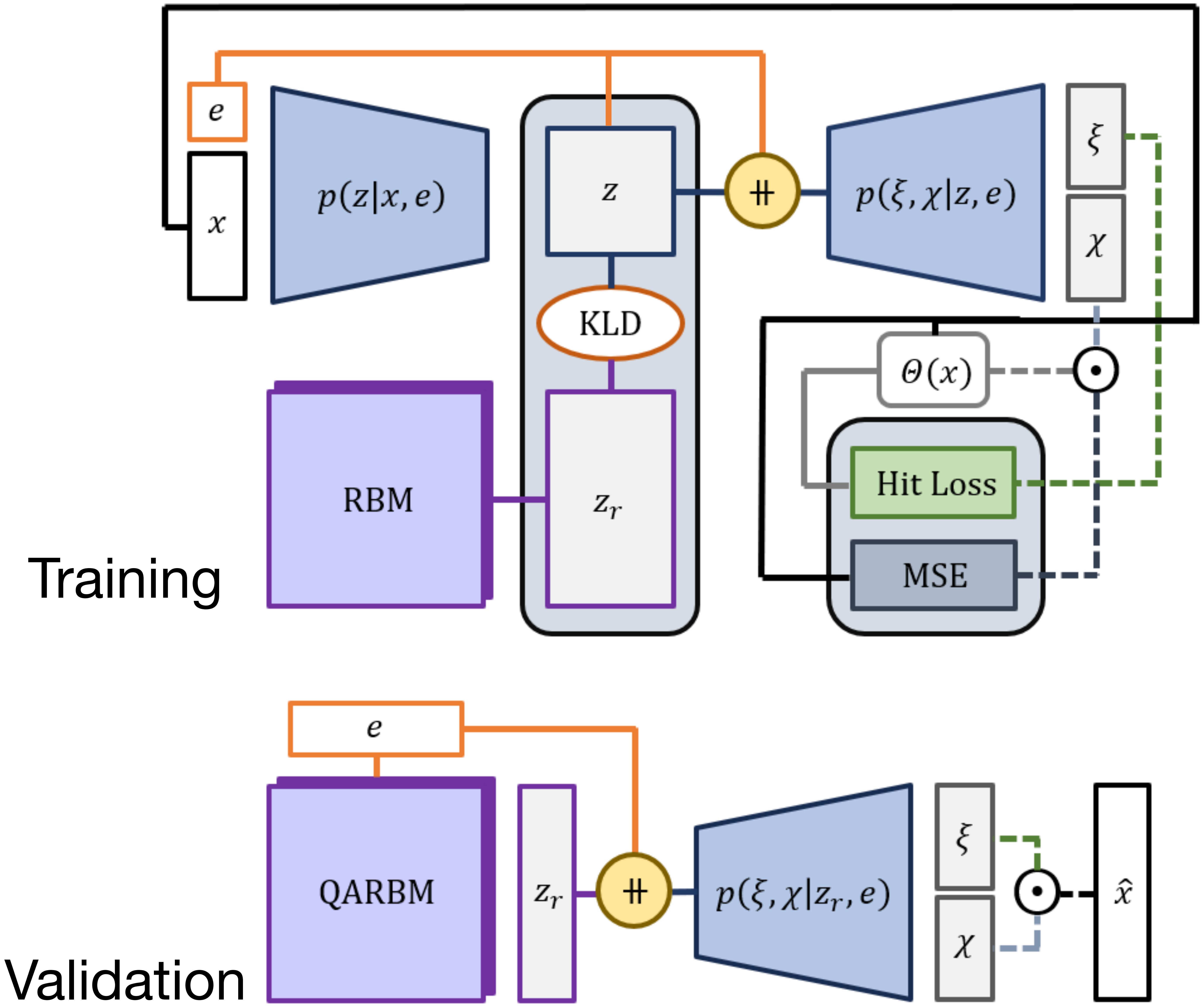
- Equate entropy of system QA to entropy of system B
- Assume $\beta = \beta_{QA} + \Delta\beta$

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

QA inverse temperature estimation



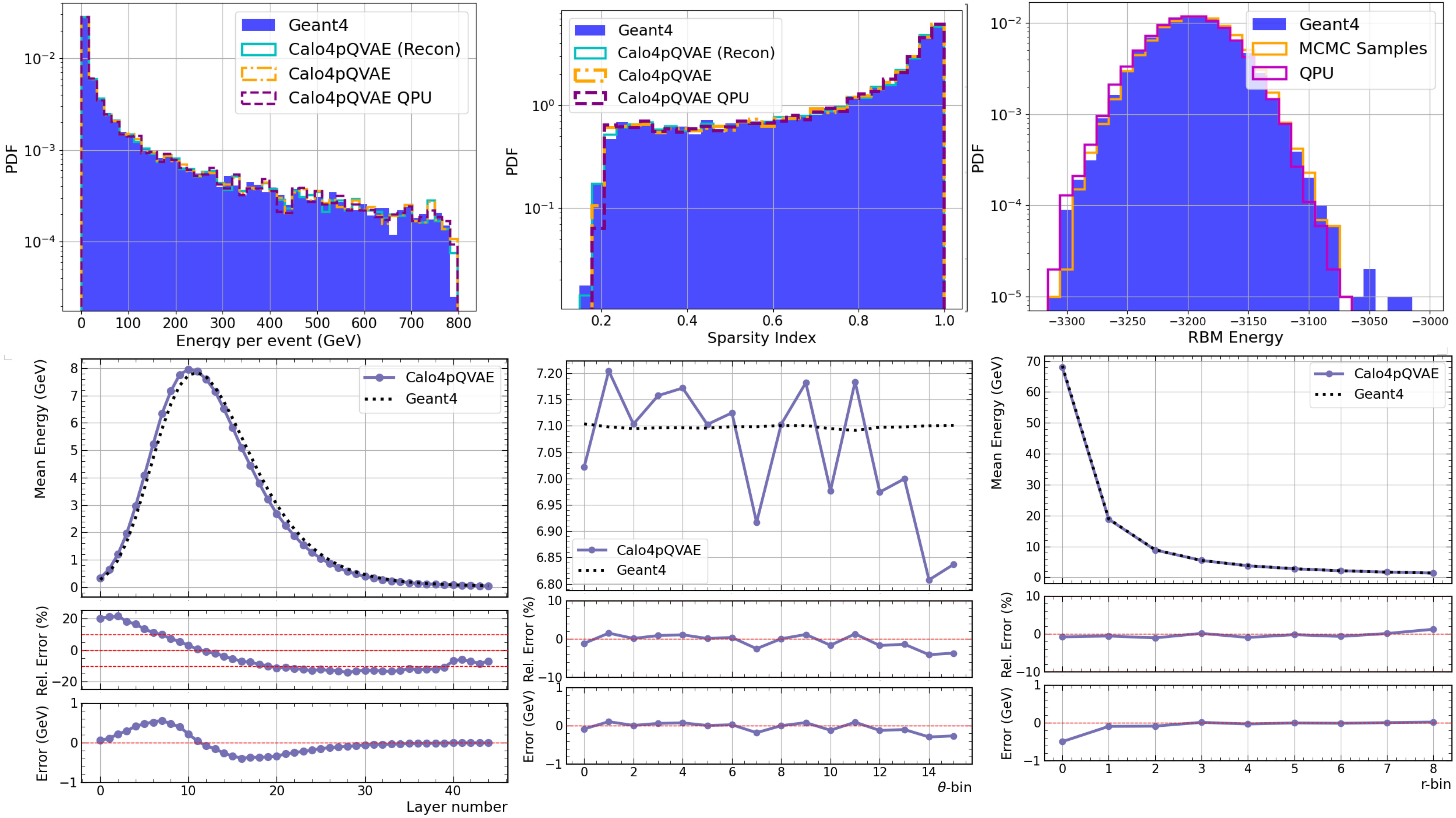
Calo4pQVAE



e
x
$p(z|x,e)$
z
KLD
RBM
z_r
$p(\xi,\chi|z,e)$
ξ
χ
$\Theta(x)$
Hit Loss
MSE
Training
e
QARBM
z_r
$p(\xi,\chi|z_r,e)$
ξ
χ
$\hat{x}$
Validation

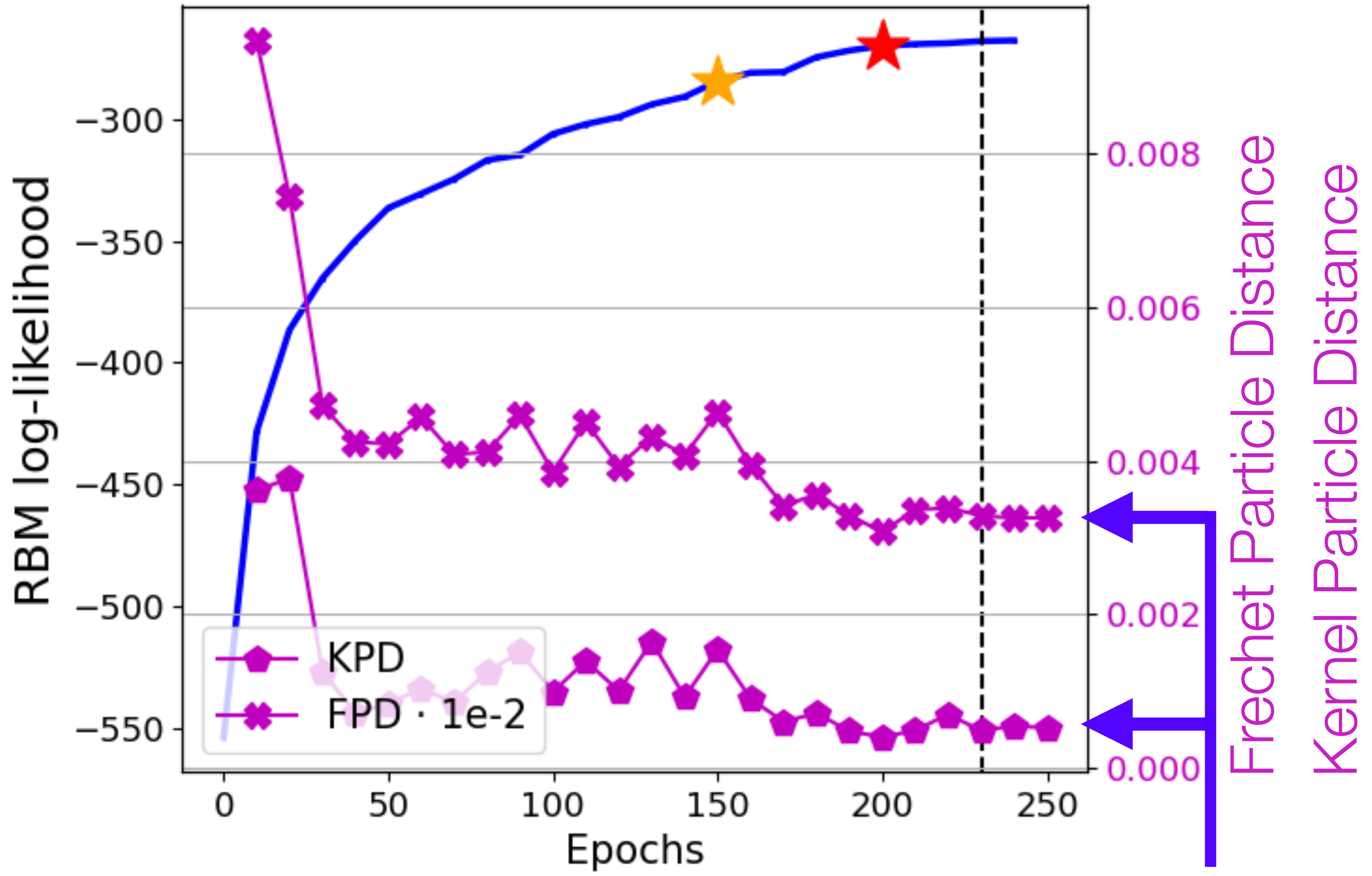
Results

★arXiv:2410.22870

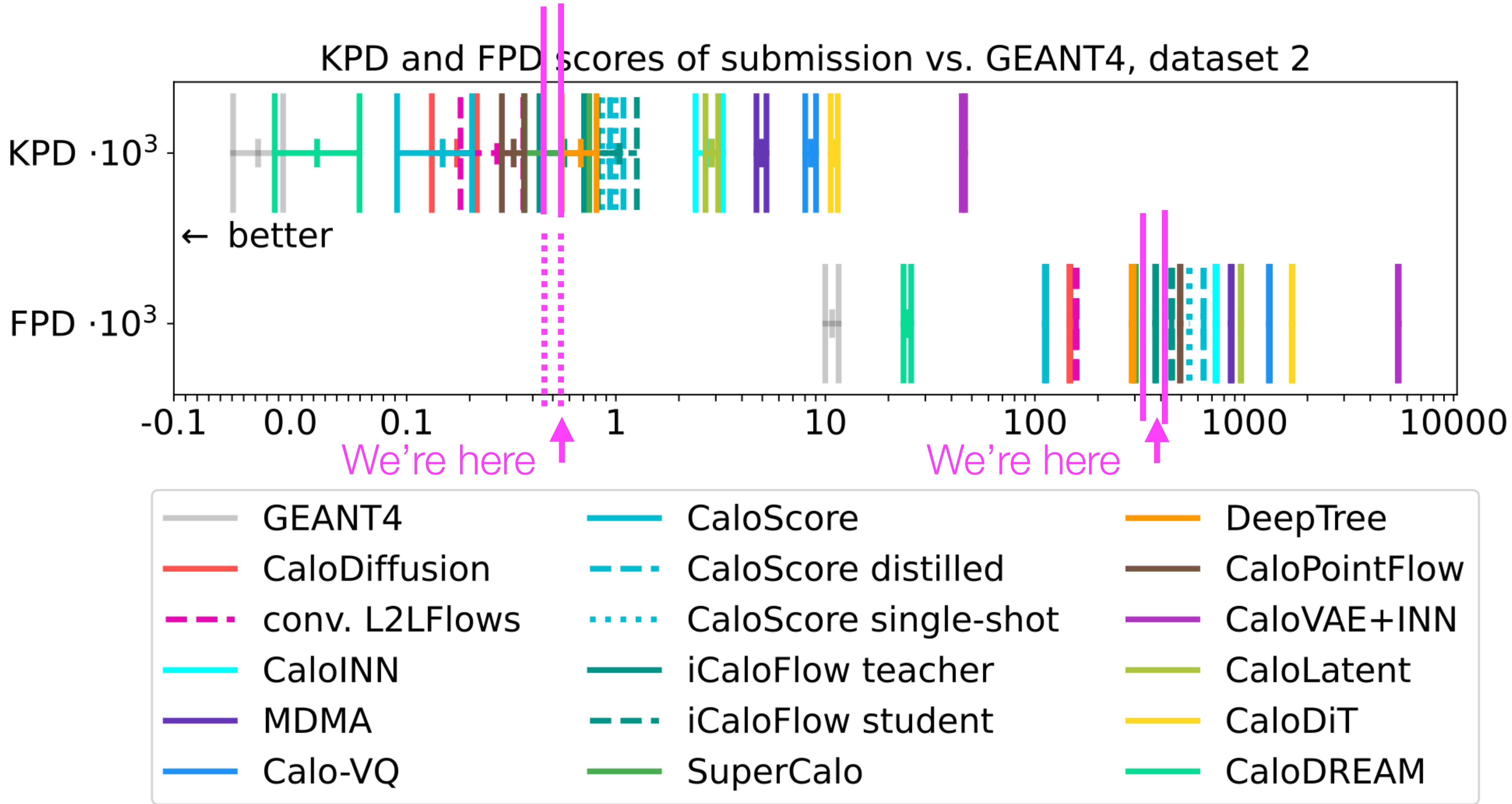


Results (NEW)

RBM Log-likelihood saturates, indicating the RBM has trained.



$$FPD(\times 10^3) = 328.7 \pm 1.1$$
$$KPD(\times 10^3) = 0.49 \pm 0.06$$



★ Slope annealing ends

★ Encoder and decoder params frozen

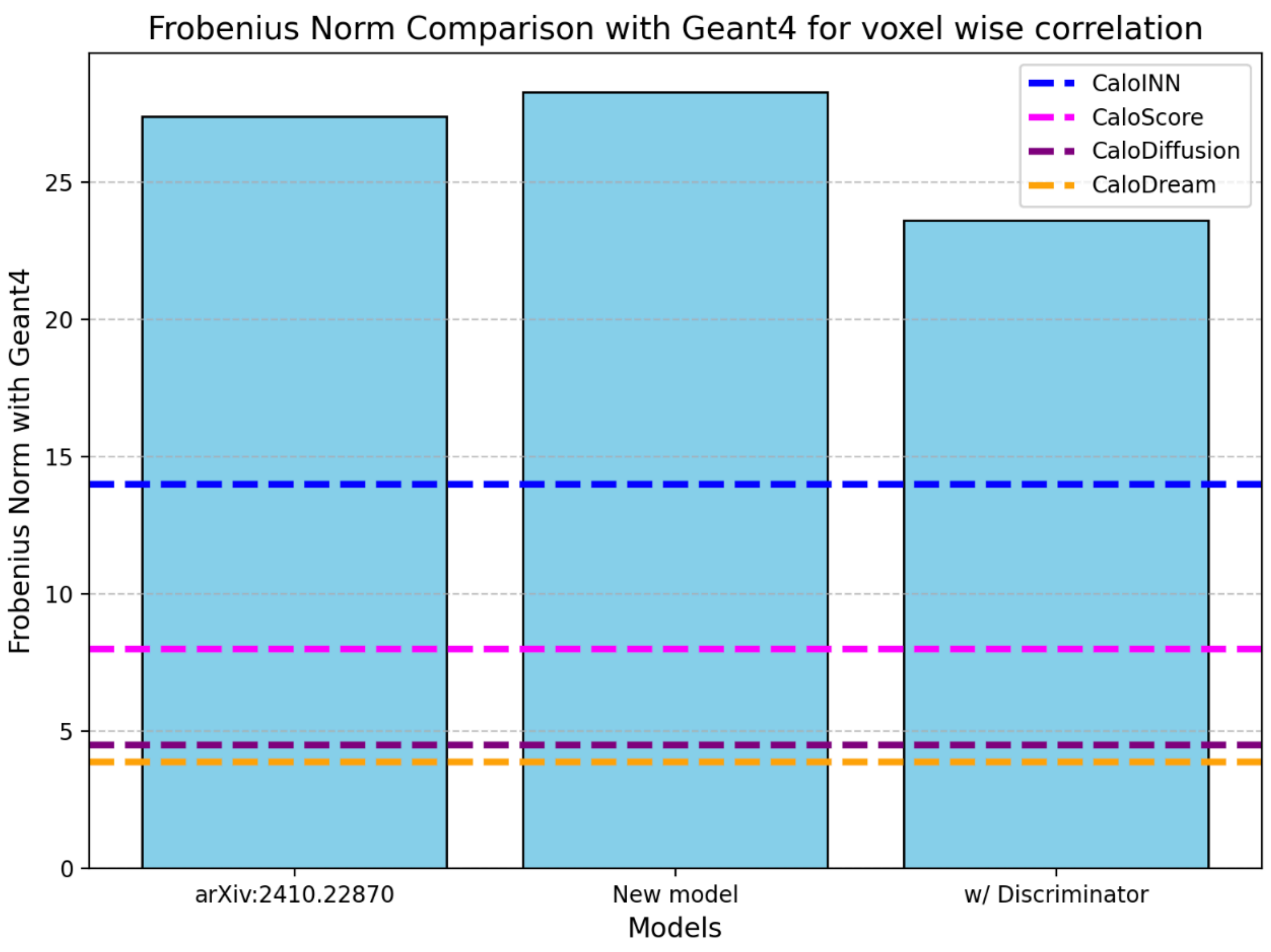
Krause C, Giannelli MF, Kasieczka G, Nachman B, Salamani D, Shih D, Zaborowska A, Amram O, Borras K, Buckley MR, Buhmann E. CaloChallenge 2022: A Community Challenge for Fast Calorimeter Simulation. arXiv preprint arXiv:2410.21611. 2024 Oct 28.

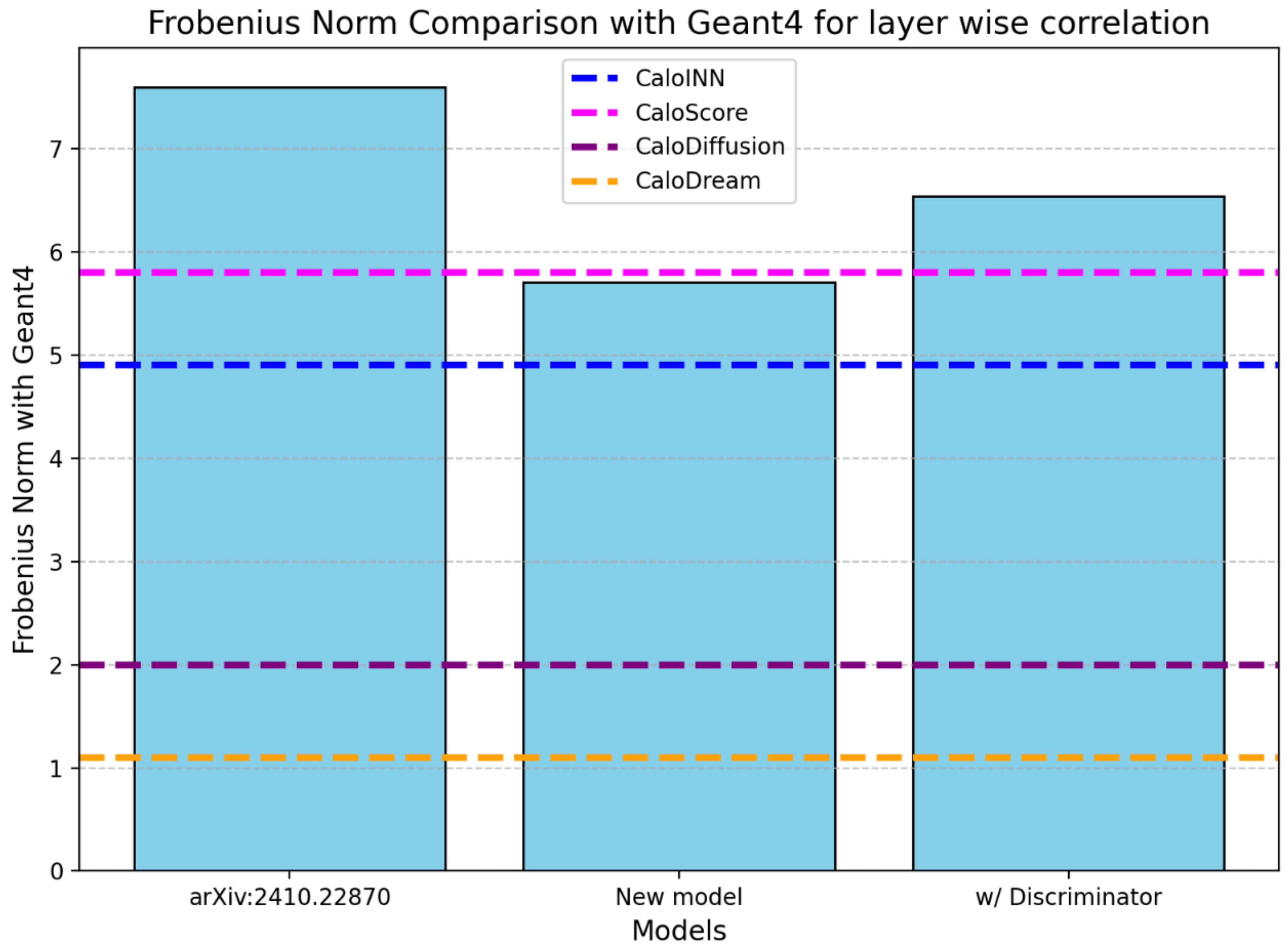
Evaluating generative models in high energy physics. Physical Review D. 2023 Apr 1;107(7):076017.

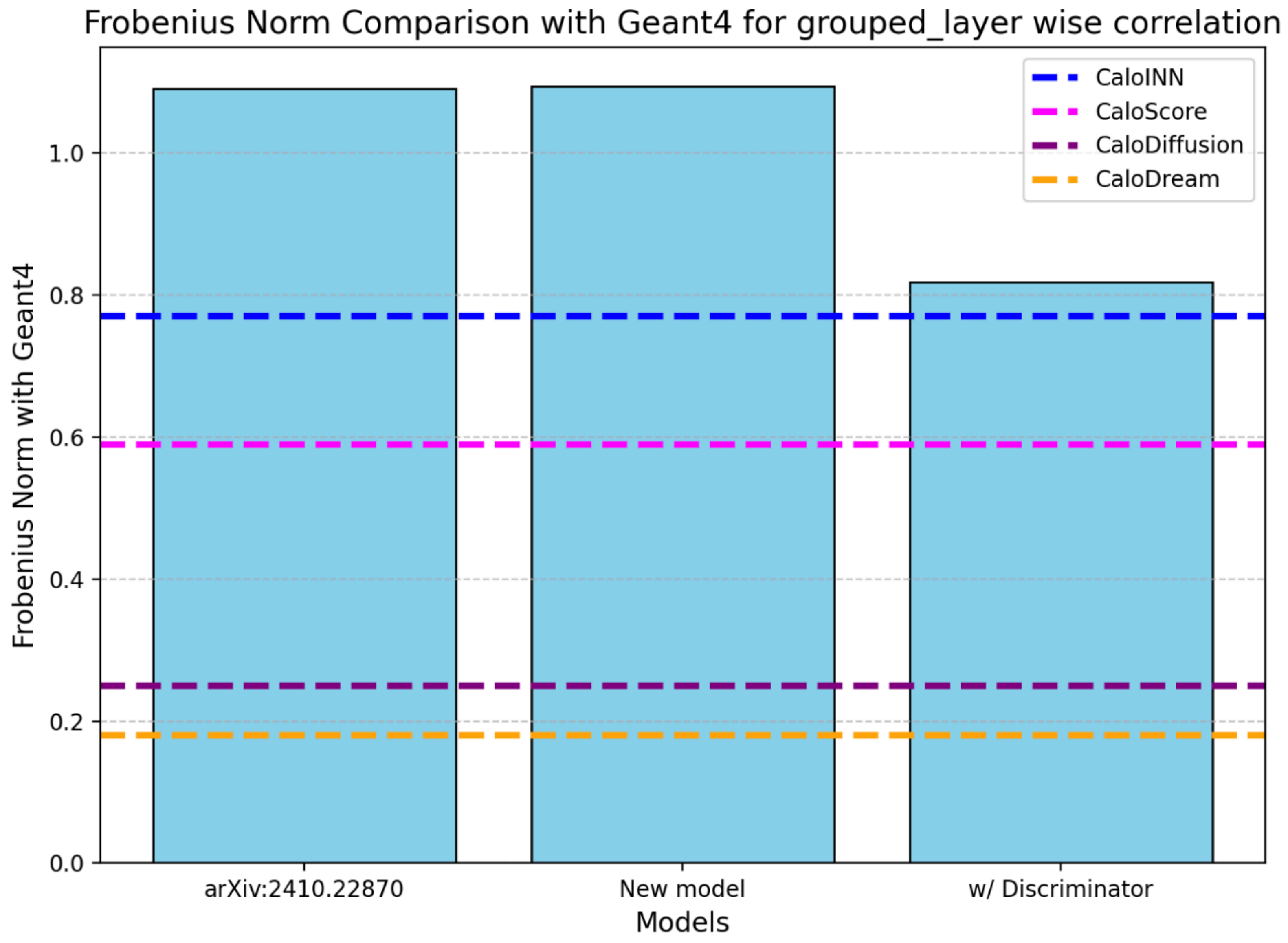
Results (NEW)

Shout out to Farzana :)

Frobenius metric

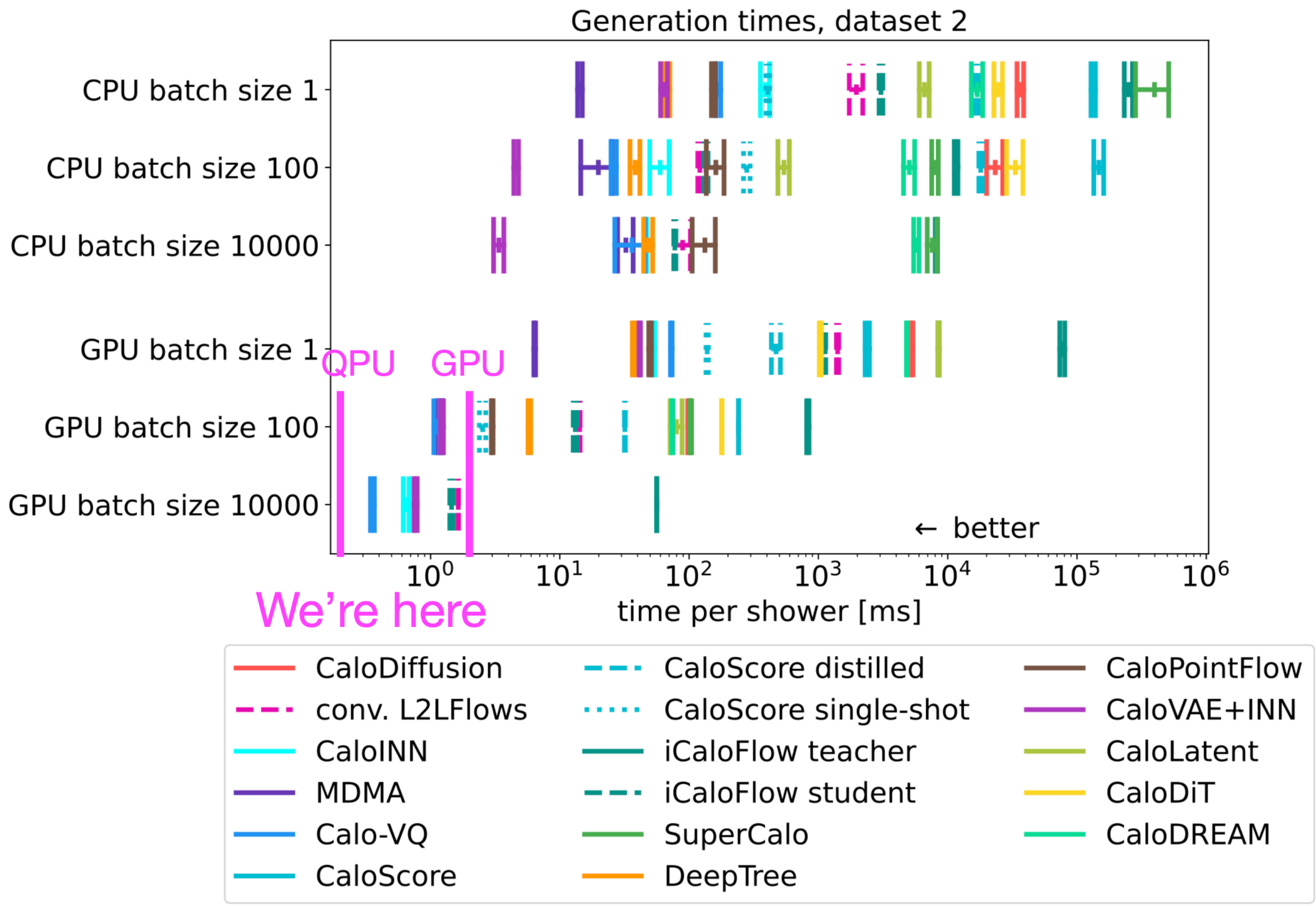






github.com/vanavreddy/Benchmarking_Calorimeter_Shower_Simulation_Generative_AI/blob/updated_2025/frob_norm_correlation.ipynb

Discussion / Conclusions / Perspectives



	GEANT4	GPU (A100)	QPU	Anneal time
Time	$\mathcal{O}(0.1) - \mathcal{O}(10^2)$ s	~ 2 ms	~ 0.2 ms	~ 0.02 ms

- ✦Improve encoder & decoder architectures.
- ✦Train model using QPU.
- ✦Train using ATLAS dataset.

Krause C, Giannelli MF, Kasieczka G, Nachman B, Salamani D, Shih D, Zaborowska A, Amram O, Borras K, Buckley MR, Buhmann E. CaloChallenge 2022: A Community Challenge for Fast Calorimeter Simulation. arXiv preprint arXiv:2410.21611. 2024 Oct 28.

Acknowledgements

Undergrads:

- Ian Lu @ UofT

PhDs:

- Hao Jia @ UBC

PIs

- Eric Paquet @ NRC
- Colin Gay @ UBC
- Roger Melko @ Perimeter Institute
- Geoffrey Fox @ University of Virginia
- Max Swiatlowski @ TRIUMF
- Wojtek Fedorko @ TRIUMF

Alumni:

- Sebastian Gonzalez @ UBC
- Sehmimul Hoque @ University of Waterloo
- Abhishek Abhishek @ UBC
- Soren Andersen @ Lund University
- Deniz Sogutlu @ UBC

- ★arXiv:2410.22870
- ★Neurips ML4Phys 2024
- ★IEEE Int Conf on QCE 2024
- ★EPJ C. 2024 Dec;84(12):1-7.
- ★arXiv:2210.07430. NeurIPS 2021

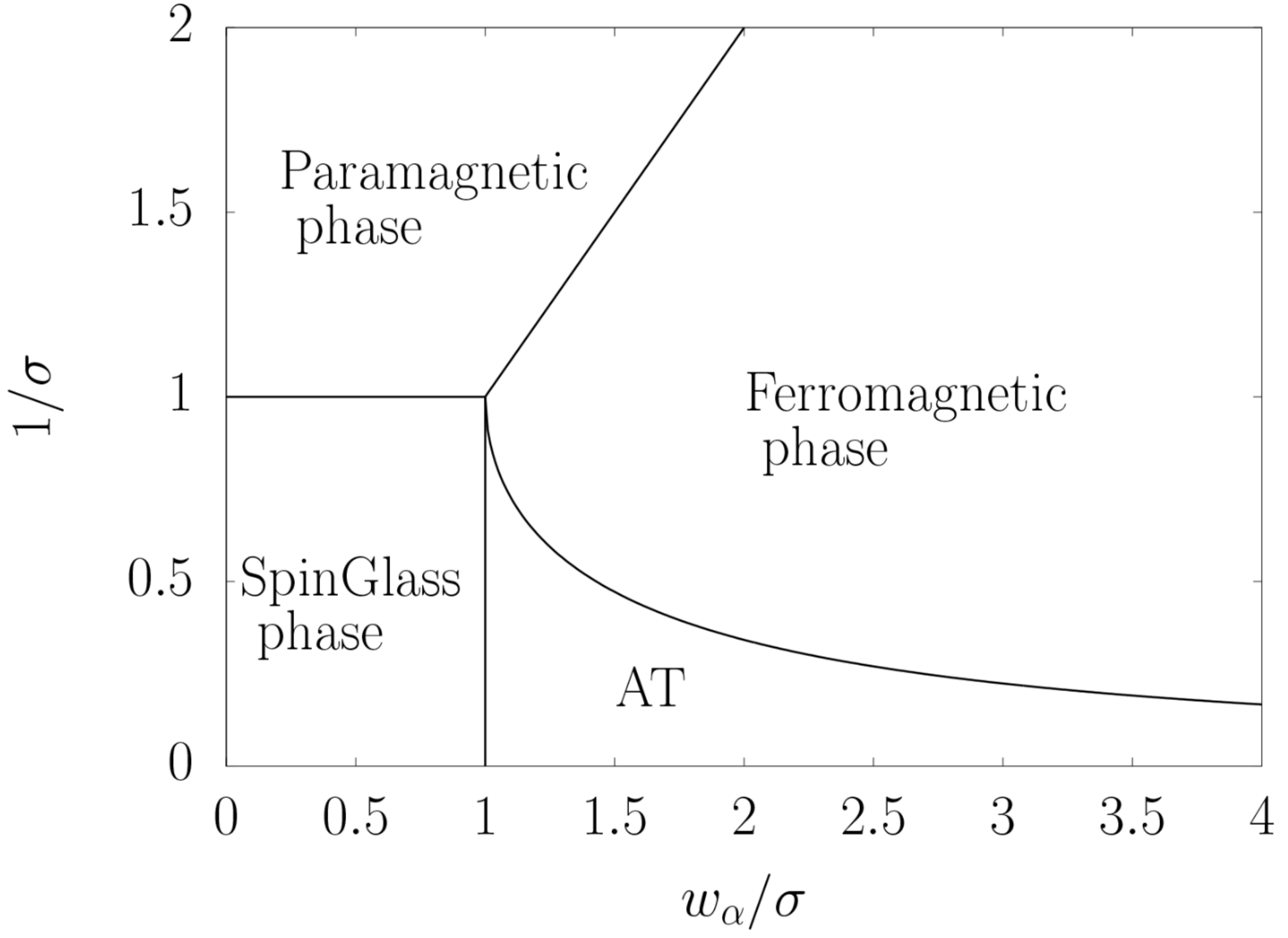
Supported by:

- ✴ NRC AQC-002
- ✴ NSERC SAPPJ-2020-00032
- ✴ SAPPJ-2022-00020
- ✴ NSF 2212550
- ✴ DOE DE-SC0023452
- ✴ Mitacs IT39533

QaloSim/CaloQVAE

jtoledo@TRIUMF.ca

Backup



Paramagnetic phase
Ferromagnetic phase
SpinGlass phase
AT
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
$1/\sigma$
w_α/σ

Results

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

KL method for beta effective calibration (Method 1).

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

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

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

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:

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

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

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

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:

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

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

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

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

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

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

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

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:

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

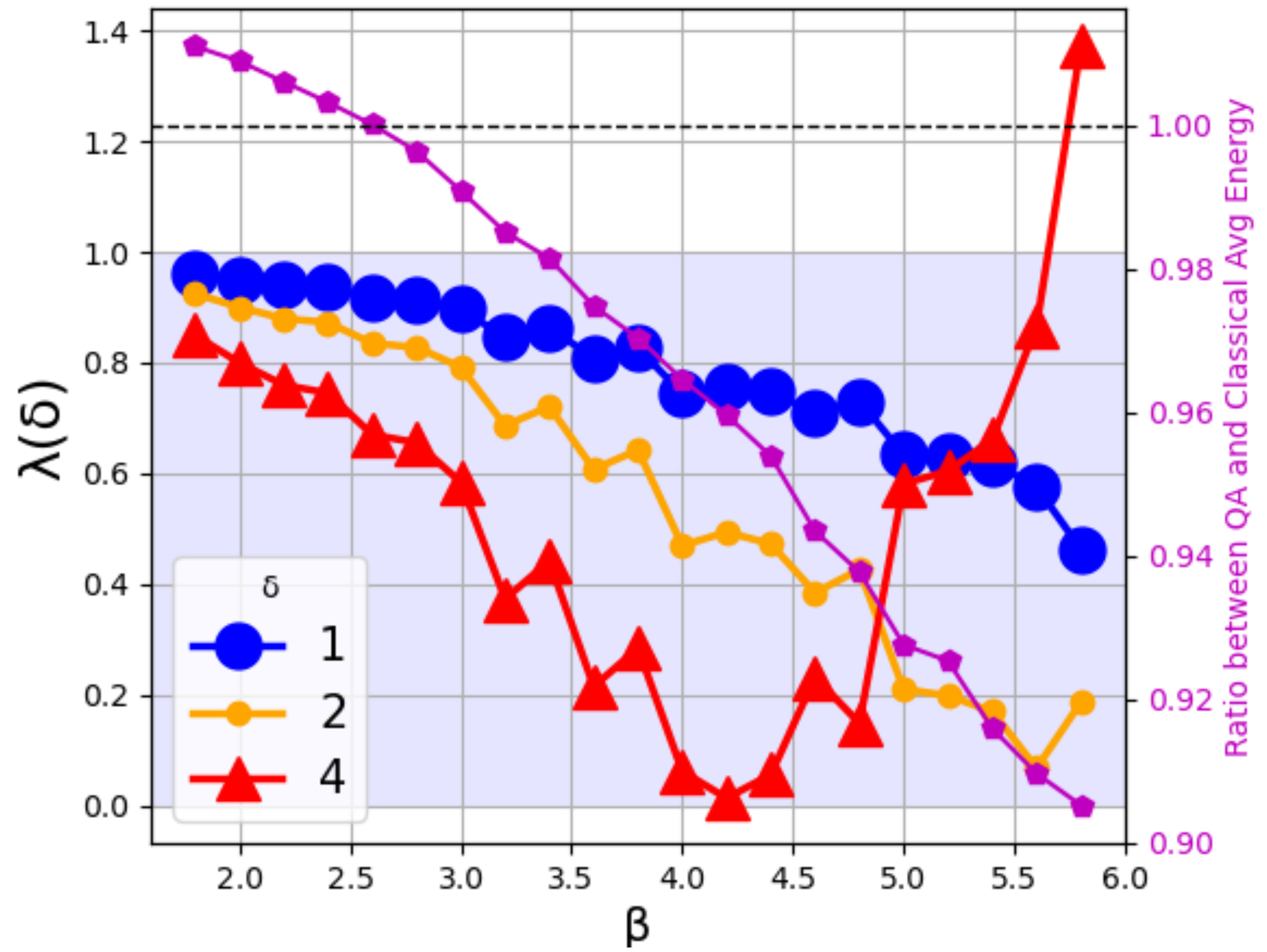
We can further simplify the previous expression by introducing the variable $\Delta\beta = \beta_{QA} - \beta$:

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

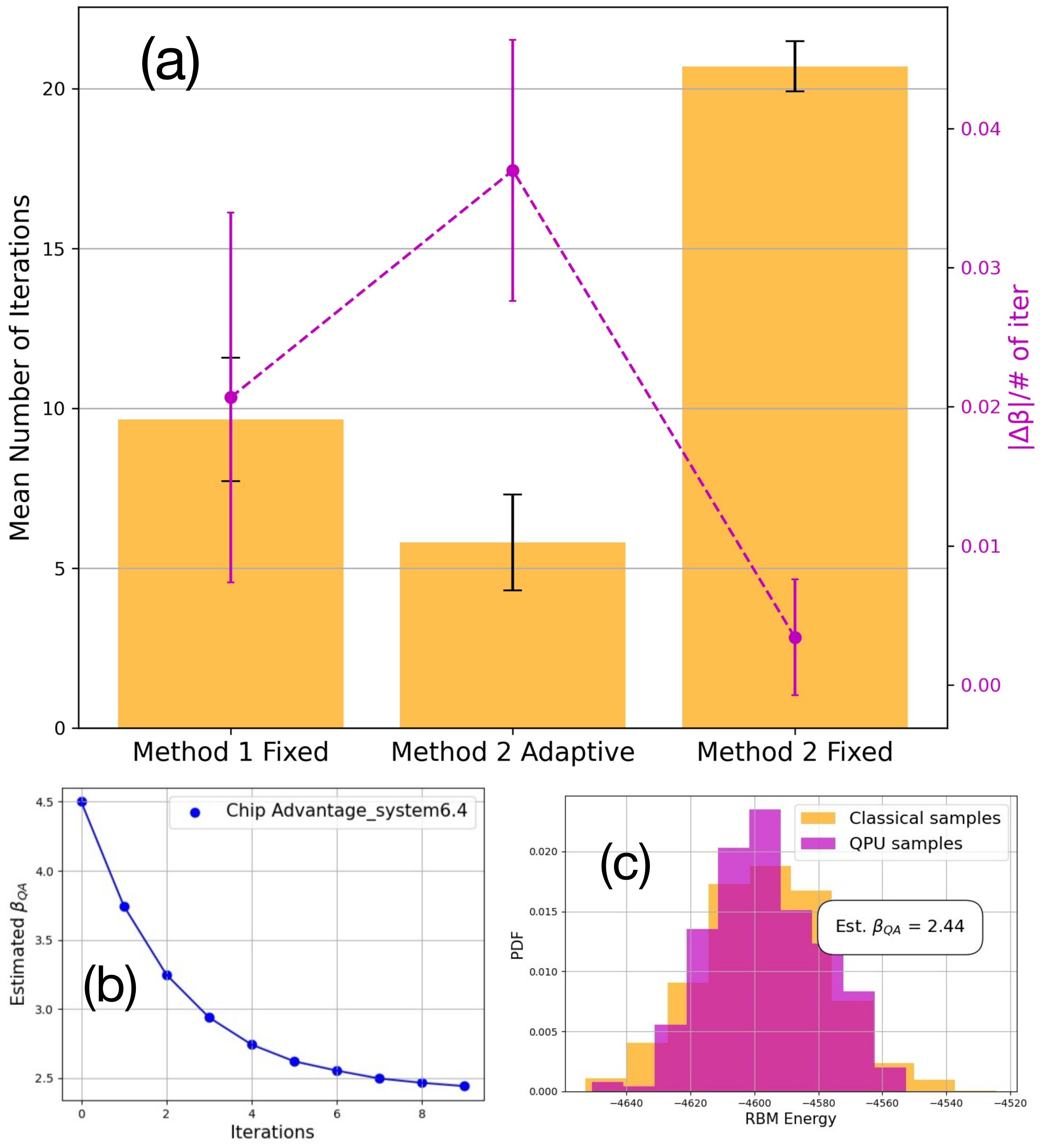
$$\beta_{t+1} = f_\delta(\beta_t) \equiv \beta_t \left(\frac{\langle H \rangle_{QA^{(r)}}}{\langle H \rangle_{B(1)}} \right)^{\delta} \quad \text{(E32)}$$

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:

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

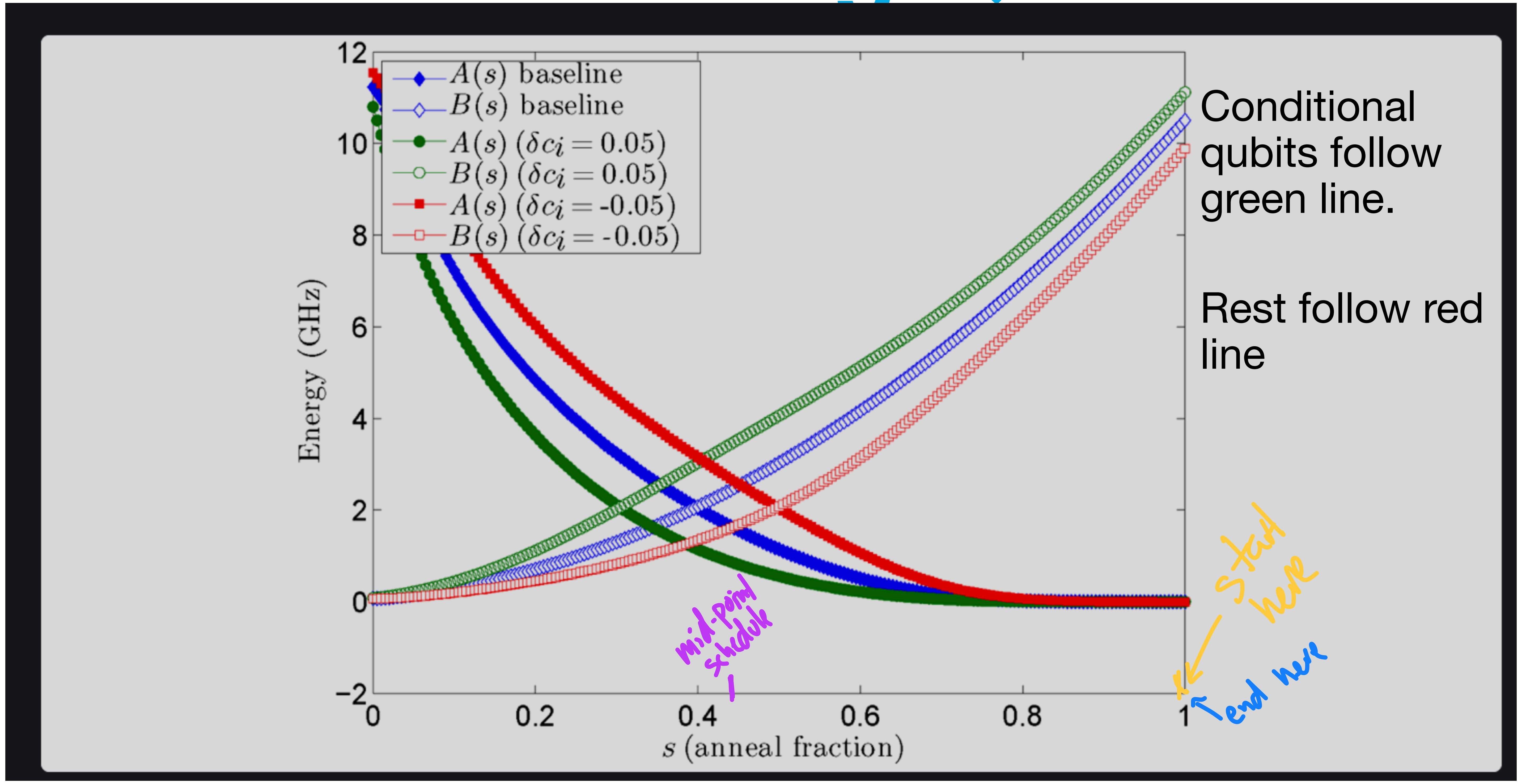


New method for beta effective calibration. (*By Hao*)



$$|\langle H\rangle_{QA} - \langle H\rangle_{RBM}| < \frac{2}{\sqrt{N}}\frac{\sigma_{QA}\sigma_{RBM}}{\sigma_{RBM}+\sigma_{QA}}$$

Conditionalizing QPU



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

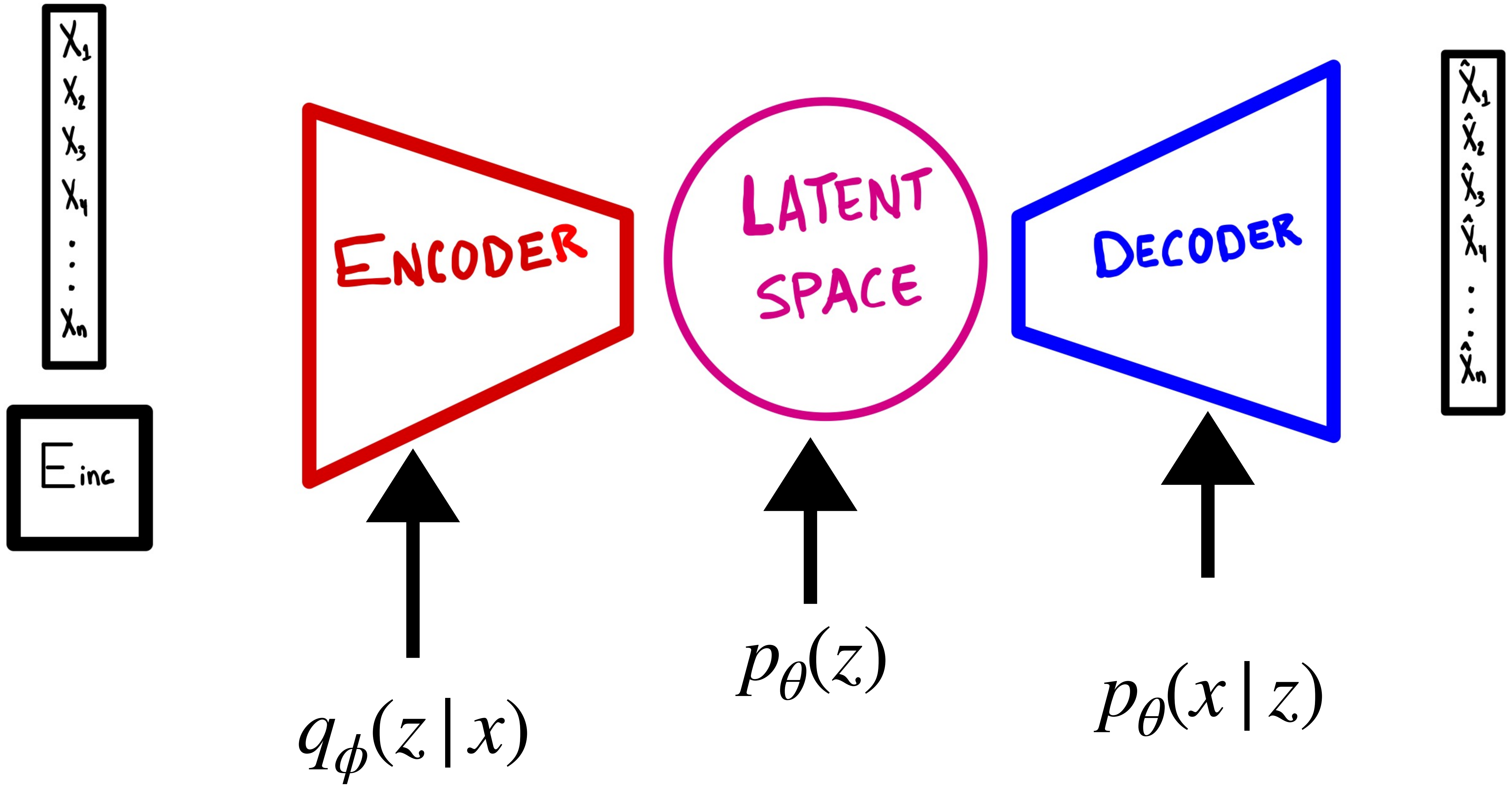
Example

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 (***currently working on this***).
- Offsetting conditionalized-qubits.
- Turning off the self-fields in transverse field associated to the conditionalized-qubits(?)

Variational Autoencoders



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

$$\langle f_\phi(z)\rangle_{q_\phi(z|x)} \sim \sum_{z\sim q_\phi(z|x)} f_\phi(z)$$

$$\nabla_\phi \langle f_\phi(z)\rangle_{q_\phi(z|x)} \sim \nabla_\phi \sum_{z\sim q_\phi(z|x)} f_\phi(z)$$

$$\nabla_\phi \sum_{\epsilon\sim\mathcal{N}(0,1)} f_\phi(z(\epsilon))$$

Reparameterization Trick

$$z = \mu_\phi(x) + \sigma_\phi(x)\cdot\epsilon$$

$$\mathcal{N}(\epsilon\,|\,0,1) = \left|\frac{dz}{d\epsilon}\right| q_\phi(z\,|\,x)$$

Discrete VAE

$$\langle f_\phi(z) \rangle_{q_\phi(z|x)} \sim \sum_{z \sim q_\phi(z|x)} f_\phi(z)$$

$$\nabla_\phi \langle f_\phi(z) \rangle_{q_\phi(z|x)} \sim \nabla_\phi \sum_{z \sim q_\phi(z|x)} f_\phi(z)$$

$$\nabla_\phi \sum_{u \sim Uni(0,1)} f_\phi(z(u))$$

Gumbel Trick

$$z = \sigma\left(\frac{l(\phi, x) + \sigma^{-1}(u)}{\tau}\right)$$

$$\rho(u) = \left|\frac{dz}{du}\right| q_\phi(z \mid x)$$

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

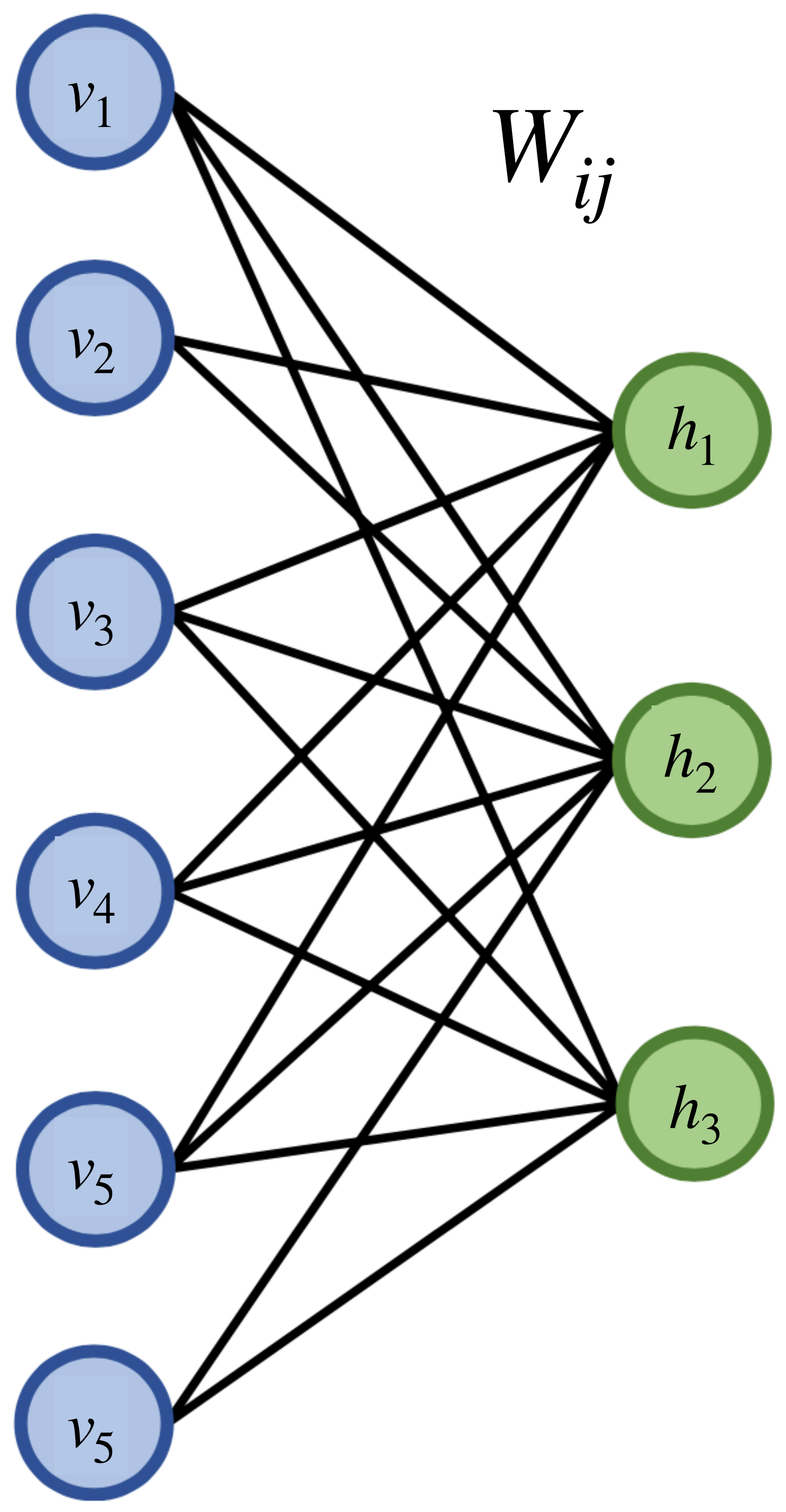
$$\mathcal{L}_{\phi,\theta}(x) = \ln p_\theta(x) - D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z \mid x)) \leq \ln p_\theta(x)$$

Restricted Boltzmann Machine

Basics

$\langle v |$ $| h \rangle$

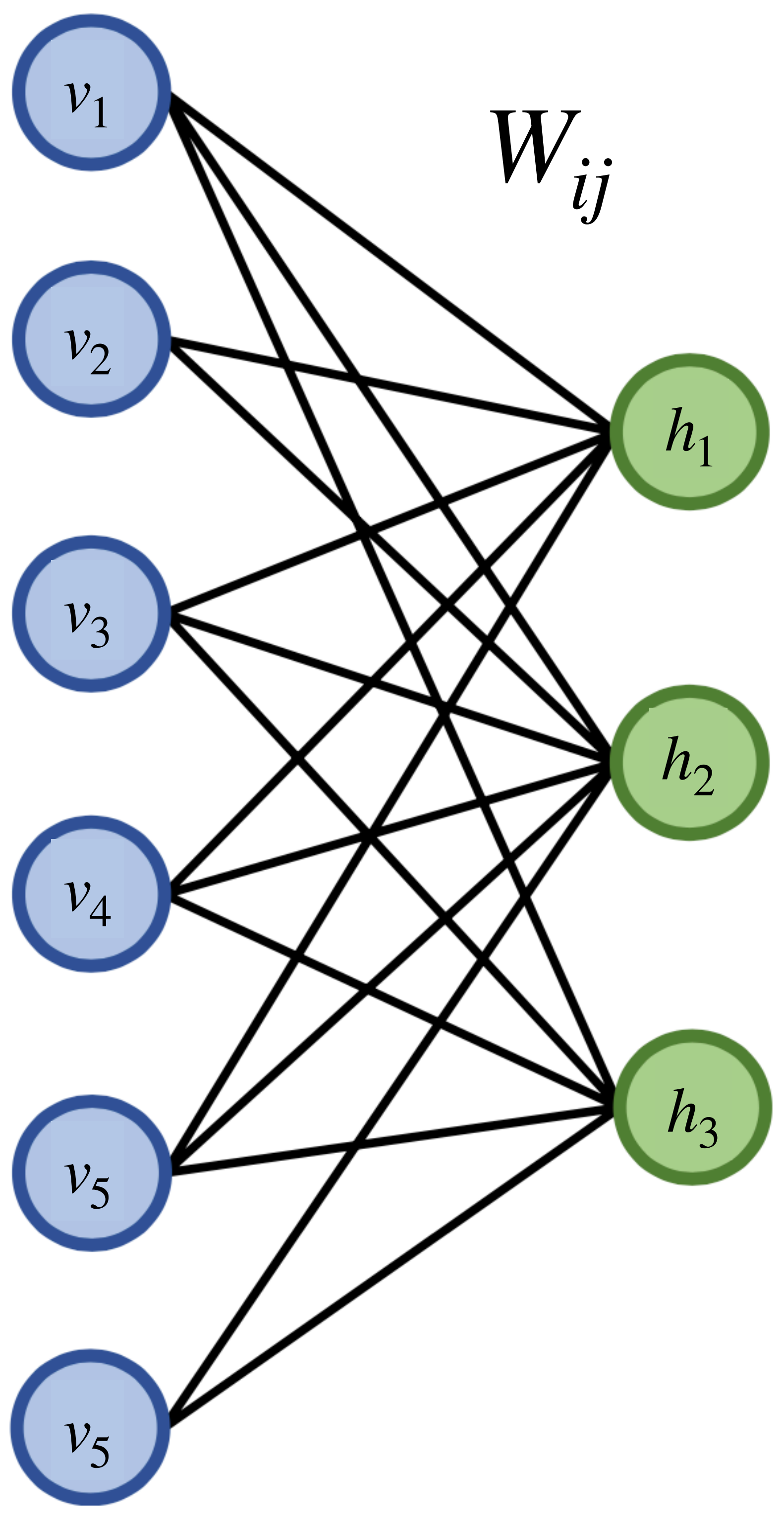
$$\frac{\partial \ln p(v)}{\partial W_{ij}} = \langle v_i h_j \rangle_{p(h|v^\alpha)} - \langle v_i h_j \rangle_{p(h',v')}$$



Restricted Boltzmann Machine

Basics

$\langle v|$ $\quad\quad$ $|h\rangle$



$$\frac{\partial \ln p(|v\rangle)}{\partial W_{ij}} = \langle v_i h_j \rangle_{p(|h\rangle||v^{(\alpha)}\rangle)} - \langle v_i h_j \rangle_{p(|h'\rangle,|v'\rangle)}$$

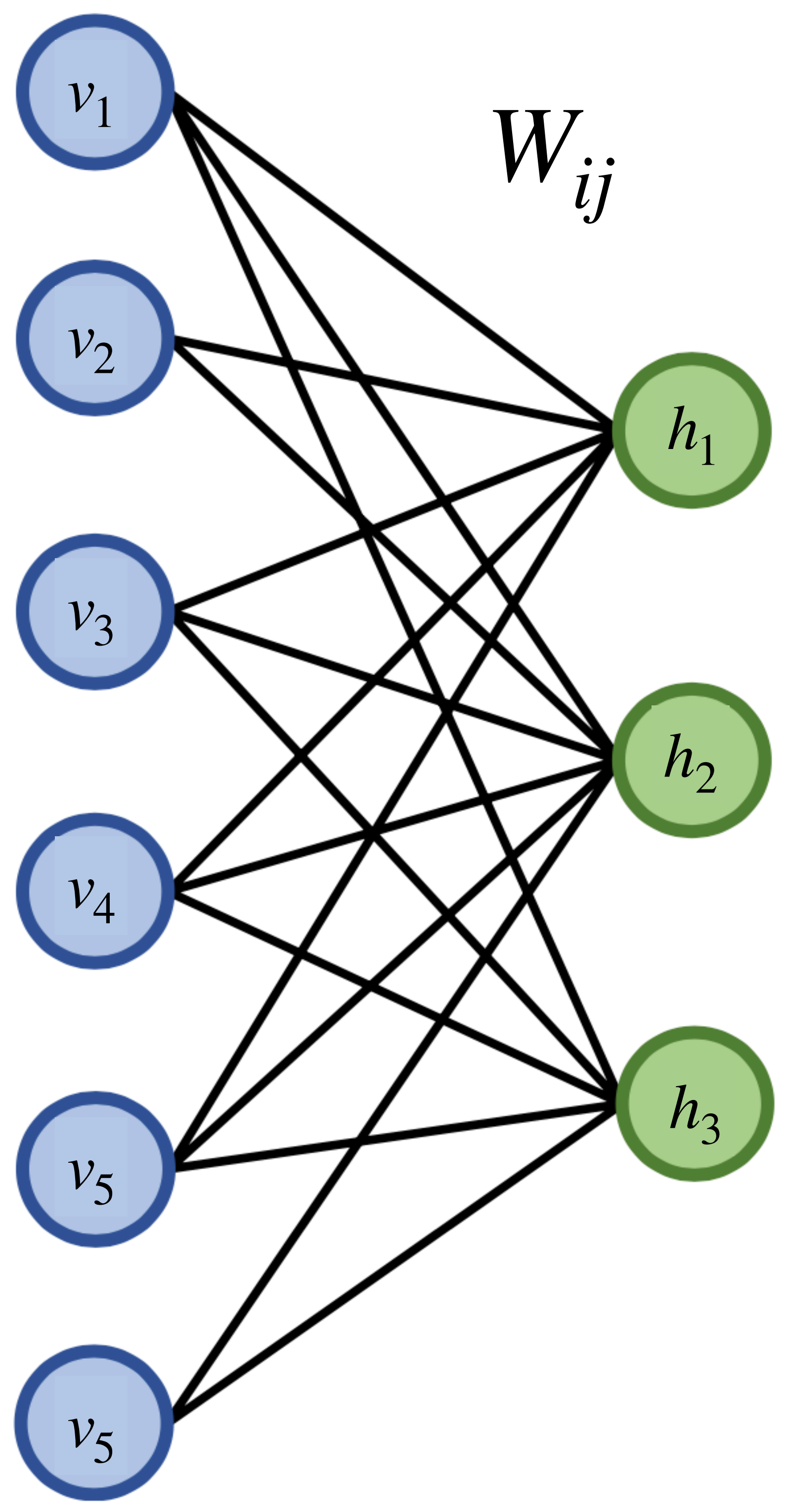
$$p(h|v) = \frac{p(v,h)}{p(v)} \quad \Rightarrow \quad p(h_j = 1|v) = \sigma(\sum_i v_i W_{ij} + b_j)$$

$$p(v|h) = \frac{p(v,h)}{p(h)} \quad \Rightarrow \quad p(v_i = 1|h) = \sigma(\sum_j W_{ij} h_j + a_i)$$

Restricted Boltzmann Machine

Basics

$\langle v |$ $\qquad | h \rangle$



$$\frac{\partial \ln p(v)}{\partial W_{ij}} = \langle v_i h_j \rangle_{p(h|v^{(\alpha)})} - \langle v_i h_j \rangle_{p(h',v')}$$

$$p(h\,|\,v) = \frac{p(v,h)}{p(v)} \quad \Rightarrow \quad p(h_j = 1\,|\,v) = \sigma(\sum_i v_i W_{ij} + b_j)$$

$$p(v\,|\,h) = \frac{p(v,h)}{p(h)} \quad \Rightarrow \quad p(v_i = 1\,|\,h) = \sigma(\sum_j W_{ij} h_j + a_i)$$

1.Start with random initial vector: $|v\rangle$

2. $|h^{(1)}\rangle \sim B[\sigma(W^t | v^{(0)}\rangle + |b\rangle)]$

3. $|v^{(1)}\rangle \sim B[\sigma(W | h^{(1)}\rangle + |a\rangle)]$

4.Repeat steps 2 and 3 n times.

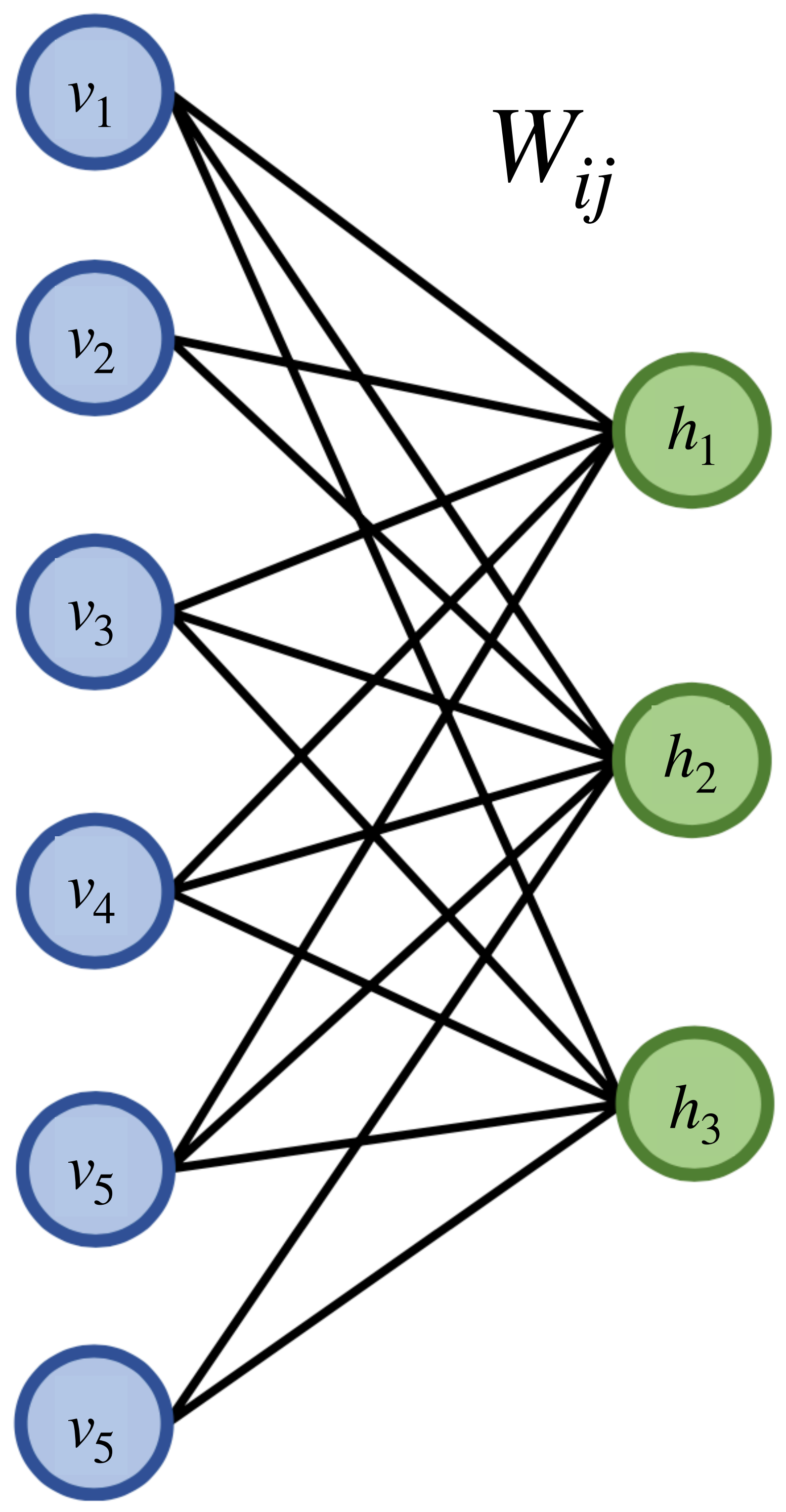
$|h^{(n)}\rangle \sim B[\sigma(W^t | v^{(n-1)}\rangle + |b\rangle)]$

$|v^{(n)}\rangle \sim B[\sigma(W | h^{(n)}\rangle + |a\rangle)]$

Restricted Boltzmann Machine

Basics

$\langle v|$ $\quad$ $|h\rangle$



$$\frac{\partial \ln p(v)}{\partial W_{ij}} = \langle v_i h_j \rangle_{p(h|v^{(\alpha)})} - \langle v_i h_j \rangle_{p(h',v')}$$

$$p(h|v) = \frac{p(v,h)}{p(v)} \Rightarrow p(h_j = 1|v) = \sigma(\sum_i v_i W_{ij} + b_j)$$

$$p(v|h) = \frac{p(v,h)}{p(h)} \Rightarrow p(v_i = 1|h) = \sigma(\sum_j W_{ij} h_j + a_i)$$

1.Start with random initial vector: $|v\rangle$
2. $|h^{(1)}\rangle \sim B[\sigma(W^t|v^{(0)}\rangle + |b\rangle)]$
3. $|v^{(1)}\rangle \sim B[\sigma(W|h^{(1)}\rangle + |a\rangle)]$
4.Repeat steps 2 and 3 n times.

$$|h^{(n)}\rangle \sim B[\sigma(W^t|v^{(n-1)}\rangle + |b\rangle)]$$
$$|v^{(n)}\rangle \sim B[\sigma(W|h^{(n)}\rangle + |a\rangle)]$$

<— Repeat this a number of times equal to batch size.

Not

$$p(\hat{x} \mid z, E_{inc}, \ldots)$$

Instead, define $\hat{x} = a \otimes h$ w/ $h_i \in \{0,1\}$

and train

$$p(a, h \mid z, E_{inc}, \ldots) = p(a \mid h, z, E_{inc}, \ldots)\, p(h \mid z, E_{inc}, \ldots)$$

$$= \left(h \frac{1}{\sqrt{2\pi x}} e^{-\frac{(a-x)^2}{2x}} + (1-h)\,\delta(a) \right) \cdot p_h^{\Theta(x)} (1-p_h)^{1-\Theta(x)}$$

$$\longrightarrow \frac{1}{\sqrt{2\pi x}} e^{-\frac{(a-x)^2}{2x}} p_h^{\Theta(x)} (1-p_h)^{1-\Theta(x)}$$

Chimera mapping

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum v_i W_{ij} h_j \quad \text{[RBM Energy]}$$

RBM variable domain: $v_i \in \{0,1\}$

QA variable domain: $s \in \{-1,1\}$

The relationship between s and v is $s = 2v - 1 \Rightarrow \frac{1}{2}(s+1) = v$

The RBM energy becomes:

$$E(v(s), h(s)) = -\sum_i a_i \tfrac{1}{2}(s_i+1) - \sum_j b_j \tfrac{1}{2}(s_j+1) - \sum_{ij} \tfrac{1}{2}(s_i+1) W_{ij} \tfrac{1}{2}(s_j+1)$$

$$= -\sum_i \frac{a_i}{2} s_i - \sum_j \frac{b_j}{2} s_j - \sum_{ij} \frac{W_{ij}}{4}(s_i s_j + s_i + s_j + 1) - \sum_i \frac{a_i}{2} - \sum_j \frac{b_j}{2}$$

$$= -\sum_i s_i \underbrace{\left(\frac{a_i}{2} + \sum_j \frac{W_{ij}}{2}\right)}_{h_i} - \sum_j s_j \underbrace{\left(b_j + \sum_i \frac{W_{ij}}{4}\right)}_{h_i} - \sum_{ij} \underbrace{\frac{W_{ij}}{4}}_{J_{ij}} s_i s_j - \underbrace{\left(\sum_i \frac{a_i}{2} + \sum_j \frac{b_j}{2} + \sum_{ij} \frac{W_{ij}}{4}\right)}_{H_G}$$

$$E(s) = -\sum_i s_i h_i - \sum_{ij} J_{ij} s_i s_j - H_G$$

$$h_{i<j} = -\left(\frac{a_i}{2} + \frac{\sum_j W_{ij}}{2}\right)$$

$$h_{j>i} = -\left(\frac{b_j}{2} + \frac{\sum_j W_{ij}}{2}\right)$$

$$J_{ij} = -\frac{W_{ij}}{4}$$

Dwave Hamiltonian

$$H_1 = \sum_i h_i s_i + \frac{1}{2}\sum_{i,j} J_{ij} s_i s_j$$

Pegasus Mapping

$$E(v,h,s,t) = -v_i a_i - h_j b_j - c_u s_u - d_l t_l - v_i W_{ij}^{(0,1)} h_j - v_i W_{iu}^{(0,2)} s_u - v_i W_{il}^{(0,3)} t_l - h_j W_{ju}^{(1,2)} s_u - h_j W_{jl}^{(1,3)} t_l - s_u W_{ul}^{(2,3)} t_l$$

$$\begin{pmatrix} v \\ h \\ s \\ t \end{pmatrix} = \frac{1}{2}\left[\begin{pmatrix} z_v \\ z_h \\ z_s \\ z_t \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\right]$$

$$E(z) = -\frac{a_i}{2} z_{i<j} - \sum \frac{a_i}{2} - \frac{b_j z_{i<j<u}}{2} - \sum_j \frac{b_j}{2} - \frac{c_u}{2} z_{j<u<l} - \sum_u \frac{c_u}{2} - d_l \frac{z_{u<l}}{2} - \sum_l \frac{d_l}{2}$$

0:
$$- z_{i<j} \frac{W_{ij}^{(0,1)}}{4} z_{i<j} - z_{i<j} \frac{W_{ij}^{(0,1)}}{4} - W_{ij}^{(0,1)} \frac{z_{i<j}}{4} - \frac{W_{ij}^{(0,1)}}{4}$$
$$- z_{i<j} \frac{W_{iu}^{(0,2)}}{4} z_{j<u} - z_{i<j} \frac{W_{iu}^{(0,2)}}{4} - \frac{W_{iu}^{(0,2)}}{4} z_{j<u} - \frac{W_{iu}^{(0,2)}}{4}$$
$$- z_{i<j} \frac{W_{il}^{(0,3)}}{4} z_{u<l} - z_{i<j} \frac{W_{il}^{(0,3)}}{4} - \frac{W_{il}^{(0,3)}}{4} z_{u<l} - \frac{W_{il}^{(0,3)}}{4}$$

1:
$$- z_j \frac{W_{ju}^{(1,2)}}{4} z_u - z_j \frac{W_{ju}^{(1,2)}}{4} - \frac{W_{ju}^{(1,2)}}{4} z_u - \frac{W_{ju}^{(1,2)}}{4}$$
$$- z_j \frac{W_{jl}^{(1,3)}}{4} z_l - z_j \frac{W_{jl}^{(1,3)}}{4} - \frac{W_{ju}^{(1,3)}}{4} z - \frac{W_{ju}}{4}$$

2 →
$$- z_u \frac{W_{ul}^{(2,3)}}{4} z_l - z_u \frac{W_{ul}^{(2,3)}}{4} - \frac{W_{ul}^{(2,3)}}{4} - \frac{W_{ul}^{(2,3)}}{4}$$

$$= -z_i\left(\frac{a_i}{2} + \frac{W_{ij}^{(0,1)}}{4} + \frac{W_{iu}^{(0,2)}}{4} + \frac{W_{il}^{(0,3)}}{4}\right) - z_j\left(\frac{b_j}{2} + \frac{W_{ij}^{(0,1)}}{4} + \frac{W_{ju}^{(1,2)}}{4} + \frac{W_{jl}^{(1,3)}}{4}\right)$$
$$- z_u\left(\frac{c_u}{2} + \frac{W_{iu}^{(0,2)}}{4} + \frac{W_{ju}^{(1,2)}}{4} + \frac{W_{ul}^{(2,3)}}{4}\right) - z_l\left(\frac{d_l}{2} + \frac{W_{il}^{(0,3)}}{4} + \frac{W_{jl}^{(1,3)}}{4} + \frac{W_{ul}^{(2,3)}}{4}\right)$$
$$- z_i \frac{W_{ij}^{(0,1)}}{4} z_j - z_i \frac{W_{iu}^{(0,2)}}{4} z_u - z_i \frac{W_{il}^{(0,3)}}{4} z_l - z_j \frac{W_{ju}^{(1,2)}}{4} z_u$$
$$- z_j \frac{W_{jl}^{(1,3)}}{4} z_l - z_u \frac{W_{ul}^{(2,3)}}{4} z_l + H_G$$