Title: 4-partite Quantum-Assisted VAE as a calorimeter surrogate

Speakers: Javier Toledo MarÃn

Series: Machine Learning Initiative

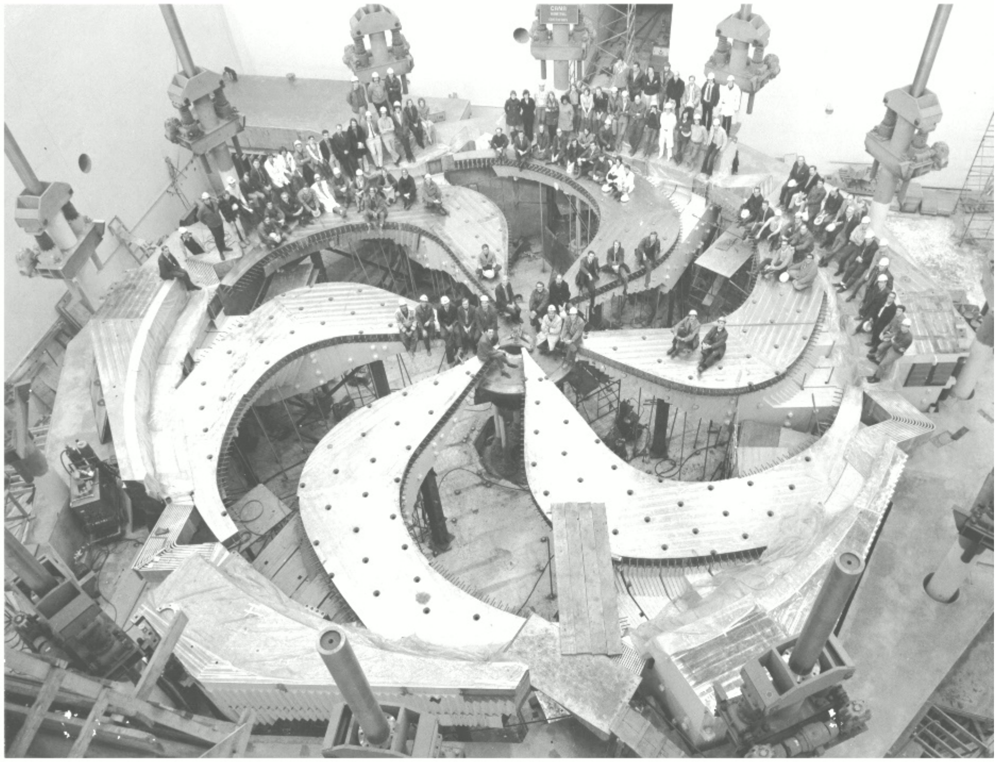
Date: October 27, 2023 - 2:30 PM

URL: https://pirsa.org/23100113

Abstract: Numerical simulations of collision events within the ATLAS experiment have played a pivotal role in shaping the design of future experiments and analyzing ongoing ones. However, the quest for accuracy in describing Large Hadron Collider (LHC) collisions comes at an imposing computational cost, with projections estimating the need for millions of CPU-years annually during the High Luminosity LHC (HL-LHC) run. Simulating a single LHC event with Geant4 currently devours around 1000 CPU seconds, with calorimeter simulations imposing substantial computational demands. To address this challenge, we propose a Quantum-Assisted deep generative model. Our model marries a variational autoencoder (VAE) on the exterior with a Restricted Boltzmann Machine (RBM) in the latent space, delivering enhanced expressiveness compared to conventional VAEs. The RBM nodes and connections are meticulously engineered to enable the use of qubits and couplers on D-Wave's Pegasus Quantum Annealer. We also provide preliminary insights into the requisite infrastructure for large-scale deployment.

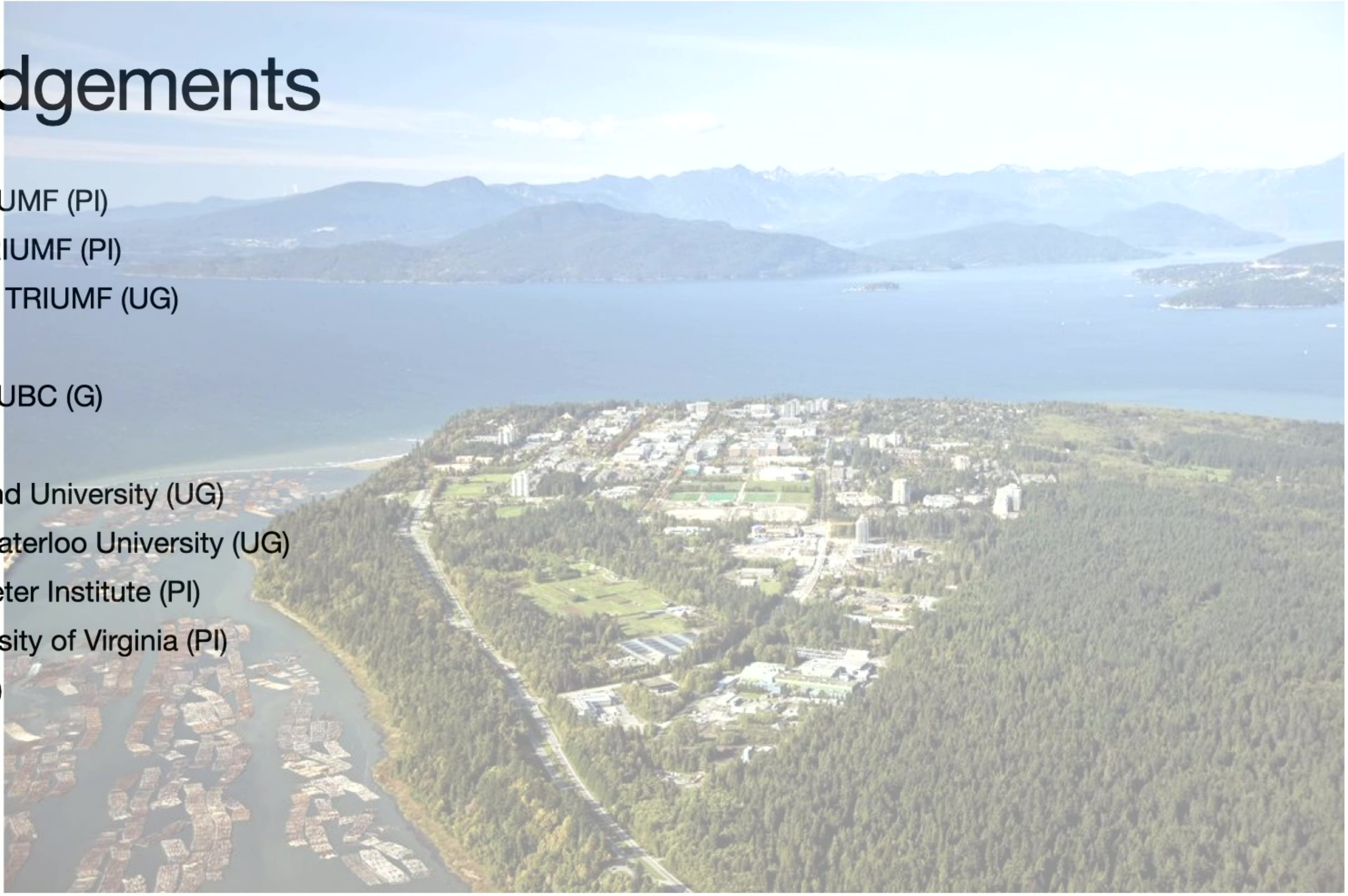
Zoom link https://pitp.zoom.us/j/97724484247?pwd=Witua1lKcHlrc3JDNHNDWXpHYkVvQT09

4-partite Quantum-Assisted VAE as a calorimeter surrogate



J. Quetzalcoatl Toledo-Marín - Research Associate @ TRIUMF :: 10/27/23 :: PI

Acknowledgements

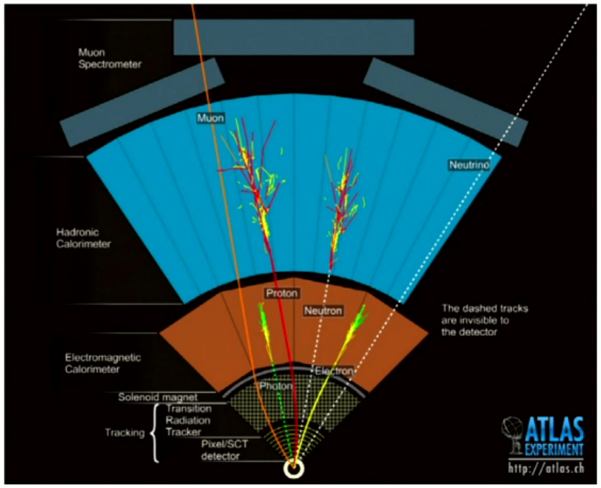


Project Team:

- Wojtek Fedorko @ TRIUMF (PI)
- Max Swiatlowski @ TRIUMF (PI)
- Sebastian Gonzalez @ TRIUMF (UG)
- Hao Jia @ UBC (G)
- Abhishek Abhishek @ UBC (G)
- Tiago Vale @ SFU (PD)
- Soren Andersen @ Lund University (UG)
- Sehmimul Hoque @ Waterloo University (UG)
- Roger Melko @ Perimeter Institute (PI)
- Geoffrey Fox @ University of Virginia (PI)
- Eric Paquet @ NRC (R)

Motivation

- One particle impacting a calorimeter can lead to thousands of secondary particles (called the shower) to be tracked through the detector, while only the total energy deposit per sensitive element (a cell) is useful.
- Can we go directly from the impacting particle parameter to the cell energy deposits?
- Could this speed up the simulation framework?



Motivation

- Simulation plays a significant role in the design of future experiments but also in the analysis of the current ones.
- One single event fully simulated with Geant4 in an LHC experiment requires about O(1) CPU seconds.
- The calorimeter simulation is by far dominating the total simulation time.
- **AI generator models are being developed in particular for the simulation of calorimeters.**

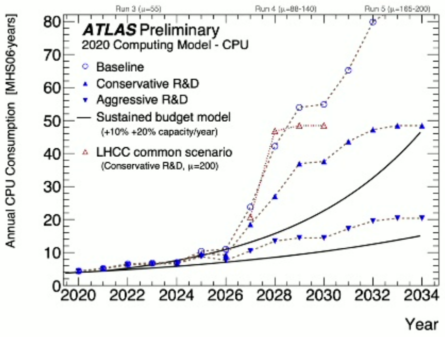
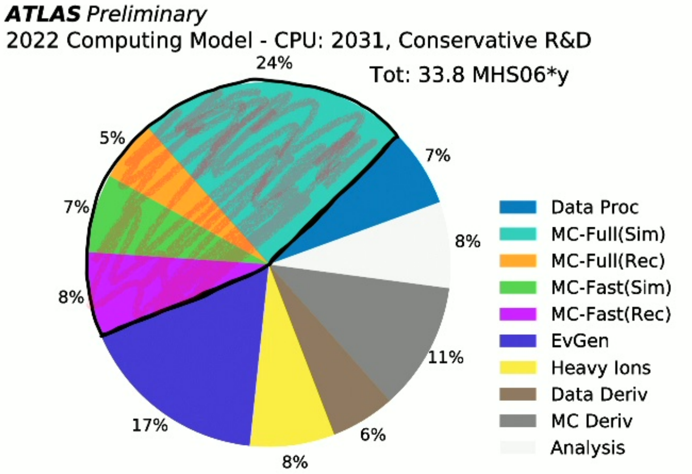
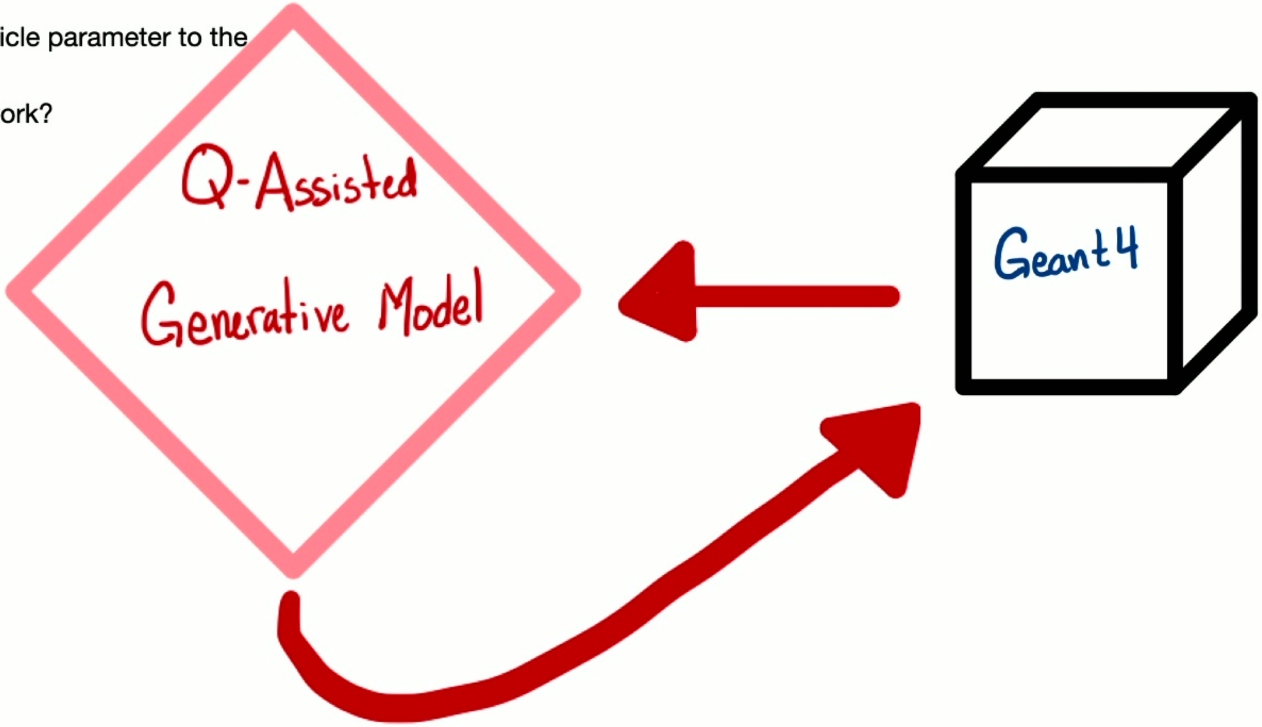


Figure 1. Projected CPU requirements of ATLAS experiment between 2020 and 2034 based on 2020 assessment. Three scenarios are shown, corresponding to an ambitious ("aggressive"), modest ("conservative") and minimal ("baseline") development program. The black lines indicate annual improvements of 10% and 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).



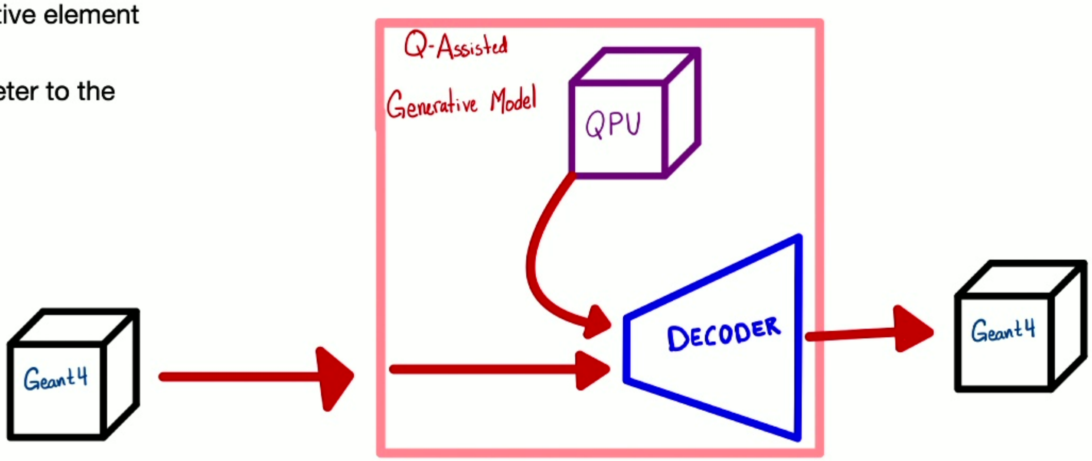
Motivation

- One particle impacting a calorimeter can lead to thousands of secondary particles (called the shower) to be tracked through the detector, while only the total energy deposit per sensitive element (a cell) is useful.
- Can we go directly from the impacting particle parameter to the cell energy deposits?
- Could this speed up the simulation framework?



Motivation

- One particle impacting a calorimeter can lead to thousands of secondary particles (called the shower) to be tracked through the detector, while only the total energy deposit per sensitive element (a cell) is useful.
- Can we go directly from the impacting particle parameter to the cell energy deposits?
- Could this speed up the simulation framework?



Contents

- Generative Models
  - Variational Autoencoders (VAE)
  - Restricted Boltzmann Machines (RBM)
  - Discrete VAE
- Quantum Annealers (QA)
- Dataset
- Results
- Conclusions

Generative Models

Simplest Example: Box-Muller Method

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

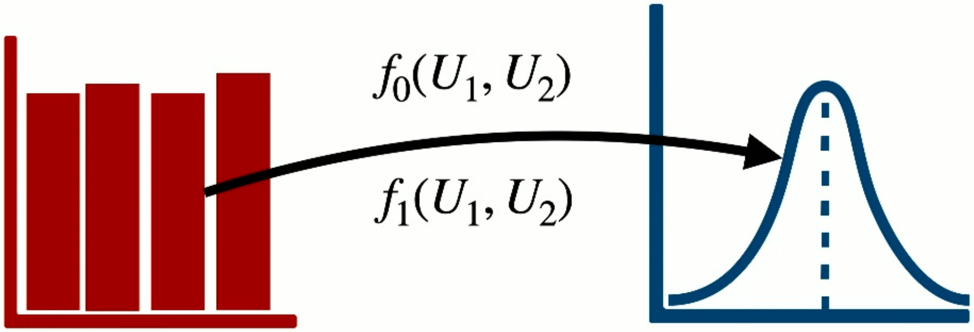
2. Substitute in:

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

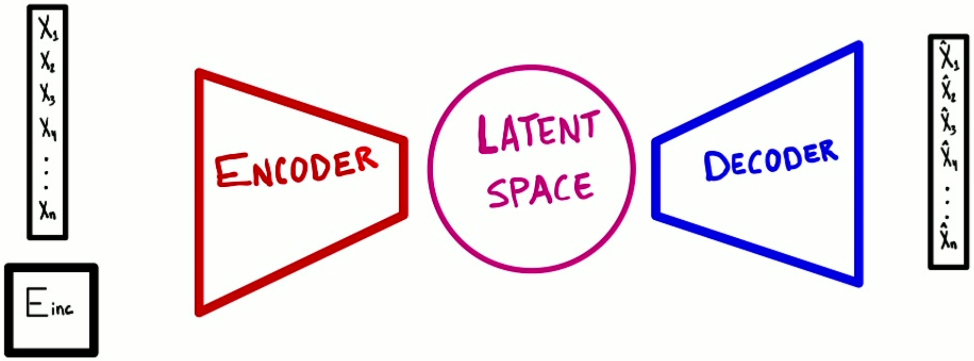
   2.$Z_1 = f_1(U_1, U_2) = \sqrt{-2\ln U_1}\sin(2\pi U_2)$

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

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



Variational Autoencoders

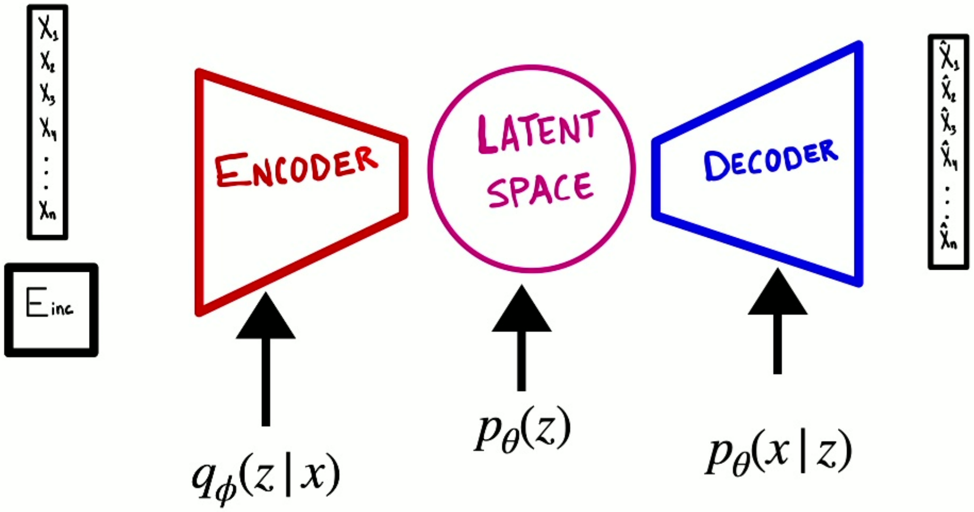


Variational Autoencoders

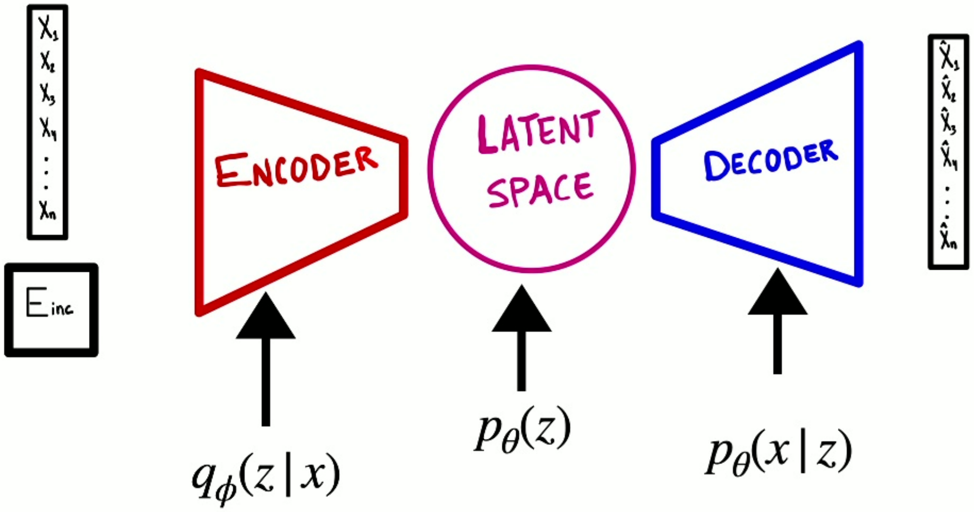
X: event

Z: Encoded data

Phi and theta are fitting parameters


X_1 X_2 X_3 X_4 ⋮ X_n
E_{inc}
ENCODER
LATENT SPACE
DECODER
$\hat{X}_1$ $\hat{X}_2$ $\hat{X}_3$ $\hat{X}_4$ ⋮ $\hat{X}_n$
$q_\phi(z|x)$
$p_\theta(z)$
$p_\theta(x|z)$

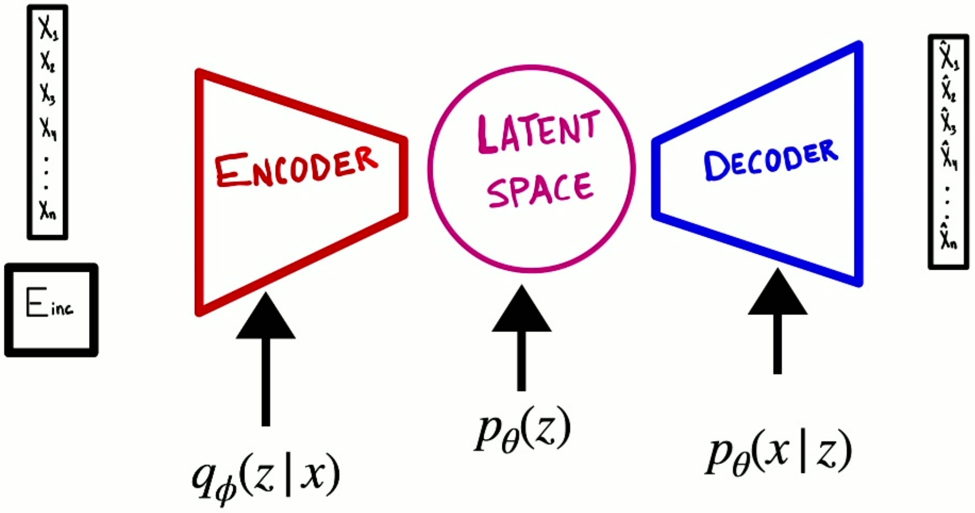
Variational Autoencoders



$$\mathscr{L}_{\phi,\theta}(x) = \langle \ln p_\theta(x\,|\,z) \rangle_{q_\phi(z|x)} - \langle \ln \frac{q_\phi(z\,|\,x)}{p_\theta(z)} \rangle_{q_\phi(z|x)}$$

Variational Autoencoders

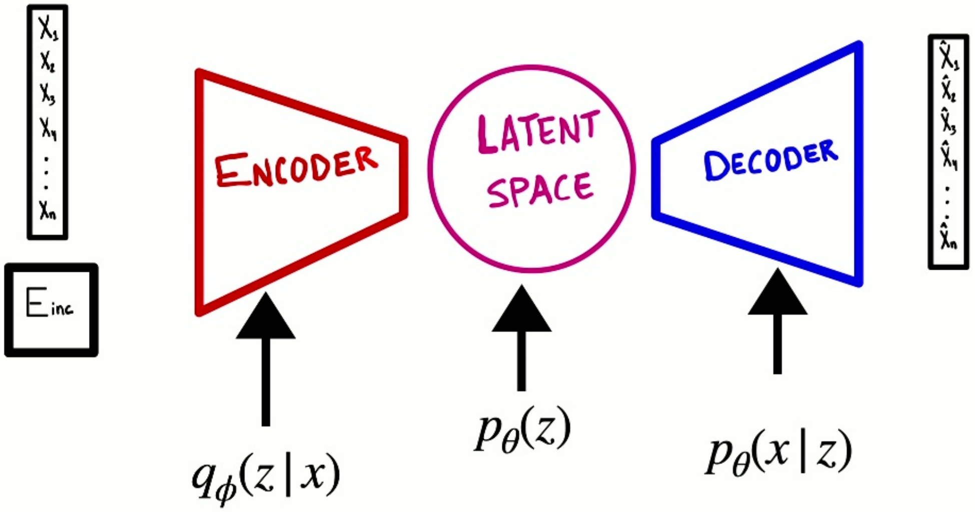
$$\langle f_\phi(z) \rangle_{q_\phi(z|x)} \sim \sum_{z \sim q_\phi(z|x)} f_\phi(z)$$



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

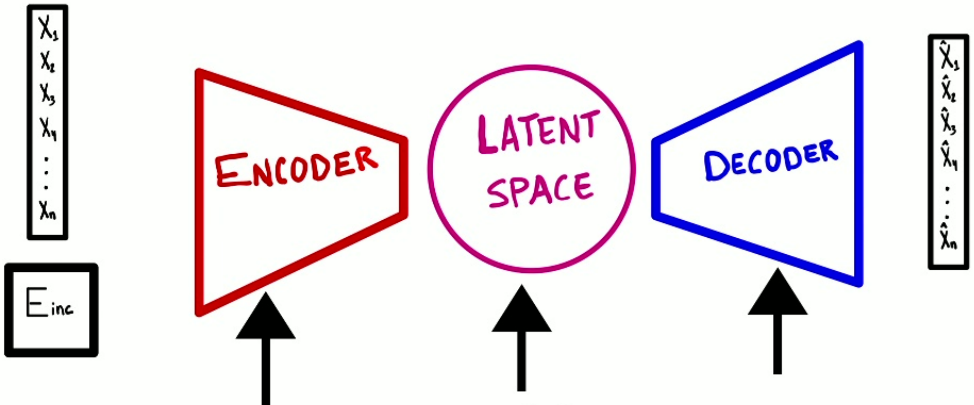
Variational Autoencoders

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



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

Variational Autoencoders



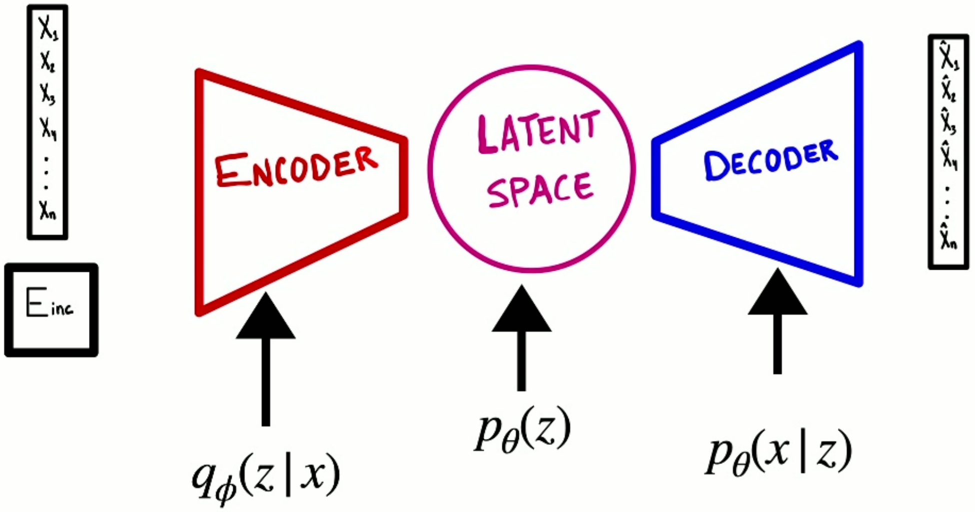
$q_\phi(z\,|\,x)$ $\quad p_\theta(z)$ $\quad p_\theta(x\,|\,z)$

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

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

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

Variational Autoencoders



X_1 X_2 X_3 X_4 ⋮ X_n
E_{inc}
ENCODER
LATENT SPACE
DECODER
$\hat{X}_1$ $\hat{X}_2$ $\hat{X}_3$ $\hat{X}_4$ ⋮ $\hat{X}_n$

$q_\phi(z\,|\,x)$ $p_\theta(z)$ $p_\theta(x\,|\,z)$

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

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

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

$$\Big\downarrow$$

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

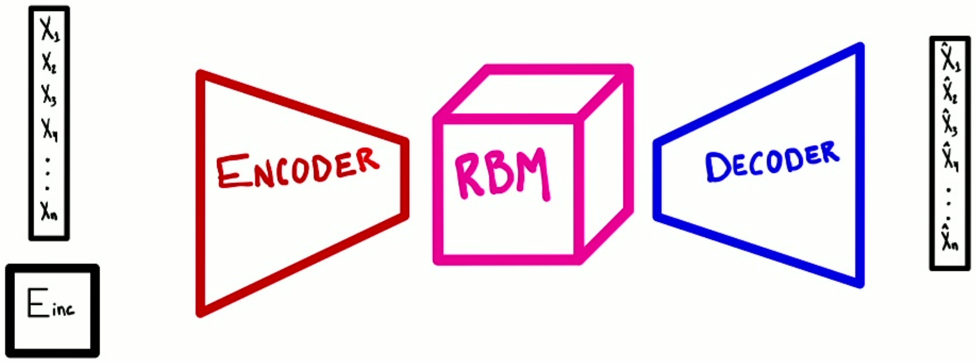
Reparameterization Trick

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

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

Restricted Boltzmann Machine

Why?

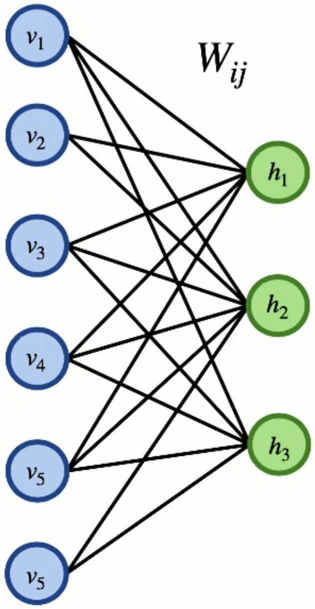


- More expressiveness
- However, this comes at a cost.

Restricted Boltzmann Machine

Basics

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



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.

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

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

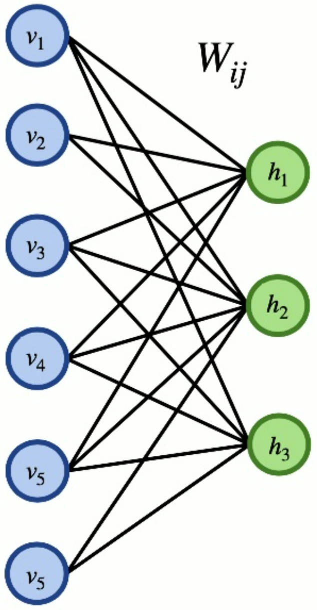
$$Z(W,a,b,\beta=1) = \sum_{v',h'} \exp(-E(v',h'))$$

Partition Function

Restricted Boltzmann Machine

Basics

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



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

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

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

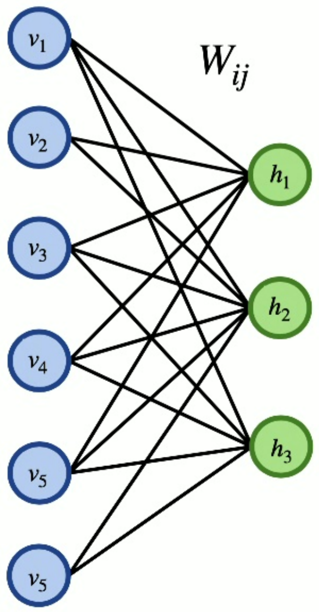
1.Start with random initial vector: $|v\rangle$
2.$|h^{(1)}\rangle \sim B[\sigma(W^t|v^{(0)}\rangle + |b\rangle)]$
3.$|v^{(1)}\rangle \sim B[\sigma(W|h^{(1)}\rangle + |a\rangle)]$
4.Repeat steps 2 and 3 n times.

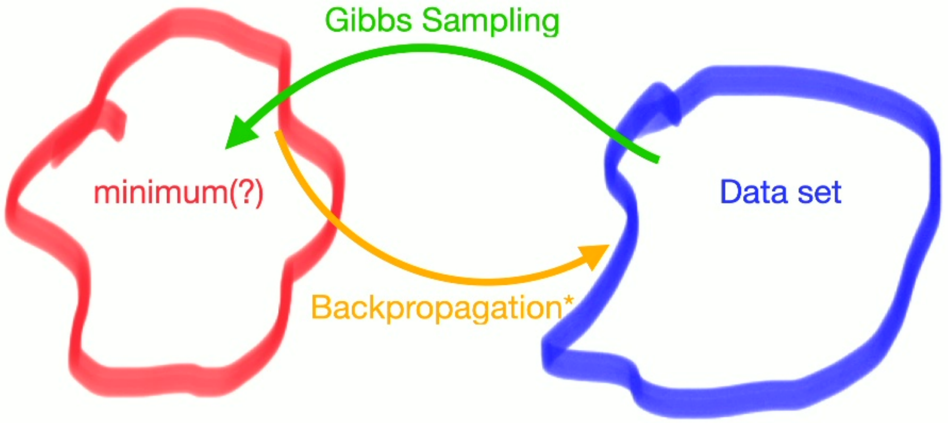
$|h^{(n)}\rangle \sim B[\sigma(W^t|v^{(n-1)}\rangle + |b\rangle)]$
$|v^{(n)}\rangle \sim B[\sigma(W|h^{(n)}\rangle + |a\rangle)]$

Restricted Boltzmann Machine

Basics

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




Gibbs Sampling

minimum(?)

Data set

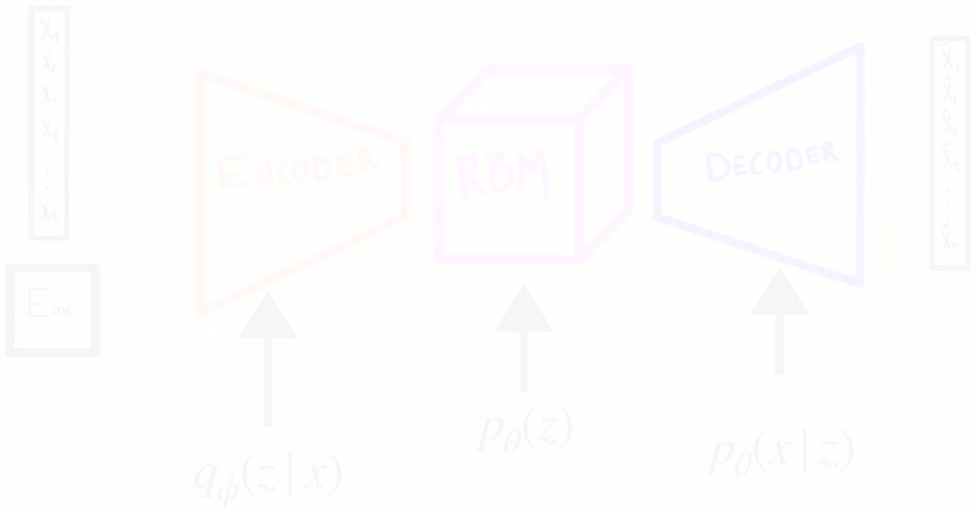
Backpropagation*

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.

Discrete VAE



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

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

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

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

$$\Big\downarrow$$

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

Gumbel Trick

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

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

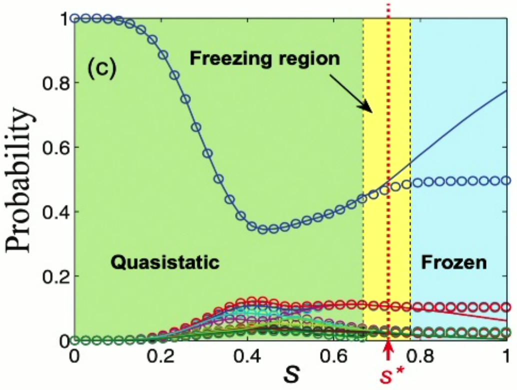
22

Quantum Annealer

Basics

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

- 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**.



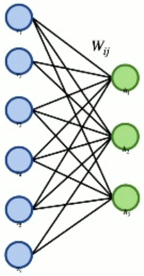
Occupation probabilities during the annealing calculated by using the Redfield formalism (circles) and the Boltzmann distribution (solid lines), assuming T = 40 mK and t_a = 20 μs. All probabilities follow the Boltzmann distribution in the quasistatic region (green) until they start freezing in the freezing region (yellow) and stay constant in the frozen region (blue). All final probabilities are close to the Boltzmann probabilities at the freeze-out point s∗, marked by the vertical (red) dashed line.

Quantum Annealer

Topologies

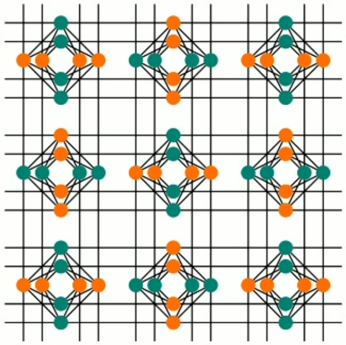
Fully Connected RBM

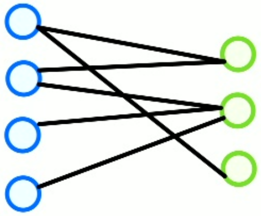
2-partite Graph



Chimera QA

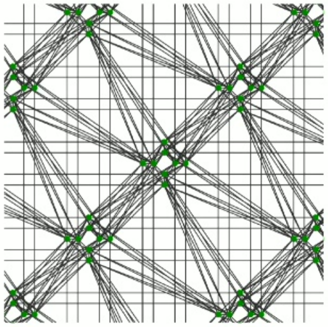
2-partite Graph

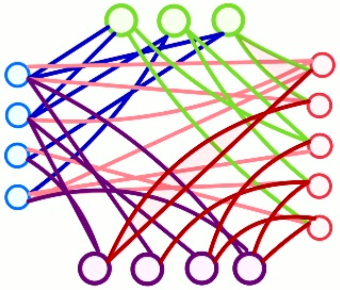




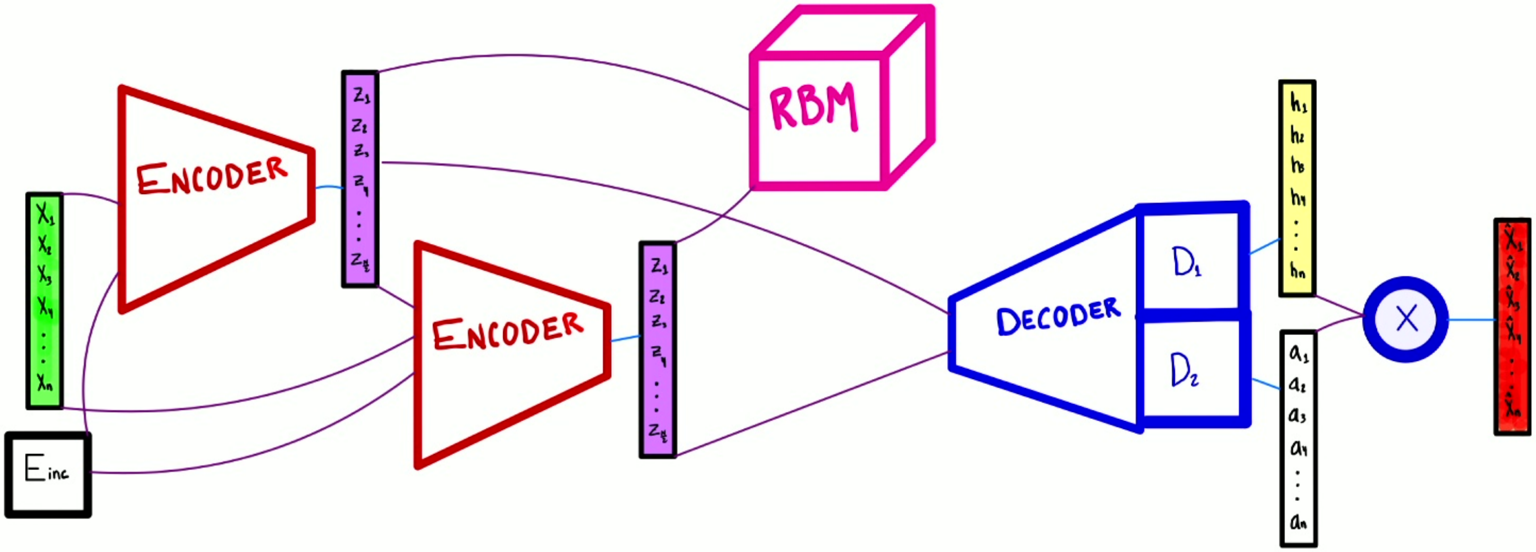
Pegasus QA

4-partite Graph





QVAE



ENCODER
ENCODER
RBM
DECODER
D_1
D_2
x_1 x_2 x_3 x_4 ⋮ x_n
E_{inc}
z_1 z_2 z_3 z_4 ⋮ z_k
z_1 z_2 z_3 z_4 ⋮ z_k
h_1 h_2 h_3 h_4 ⋮ h_n
a_1 a_2 a_3 a_4 ⋮ a_n
X
$\hat{x}_1$ $\hat{x}_2$ $\hat{x}_3$ $\hat{x}_4$ ⋮ $\hat{x}_n$

Loss = MSE(🟩🟥) + BCE(🟨 🟩) + Div$_{KL}$(🟥🟪)

Not

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

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

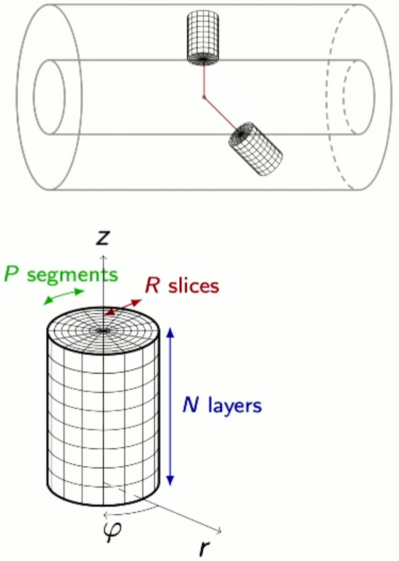
and train

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

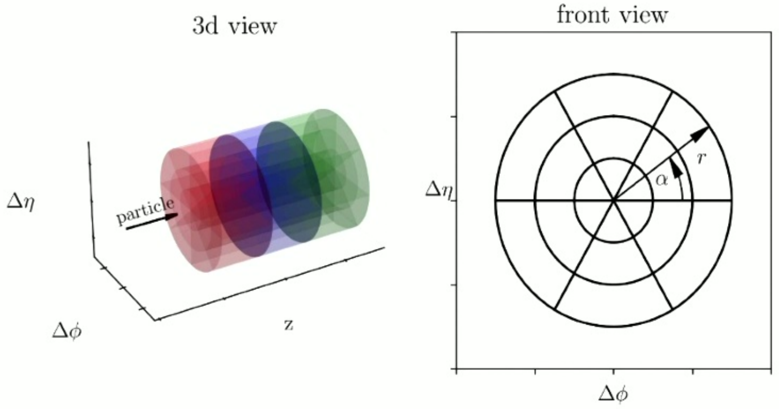
$$= \left(h \frac{1}{\sqrt{2\pi x}} e^{-\frac{(a-x)^2}{2x}} + (1-h)\, \delta(a) \right) \cdot p_h^{\Theta(x)} (1 - p_h)^{1-\Theta(x)}$$

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

CaloChallange Dataset

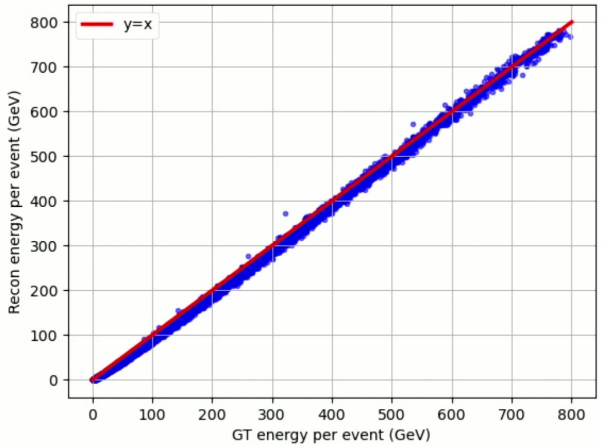


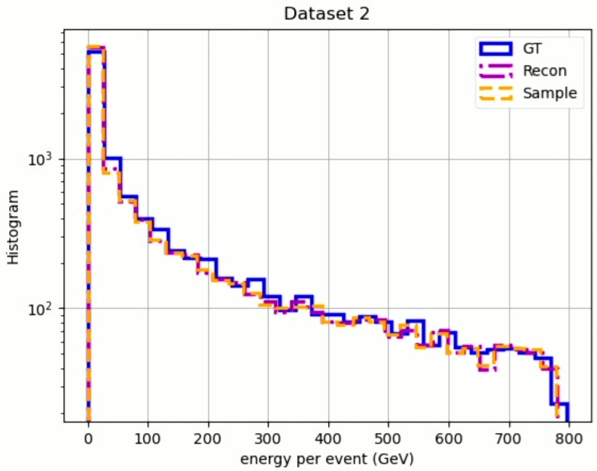
Dataset	
Particle type	Electron showers
Layers	45
Voxels per layer	9 radial * 16 angular
Incident energies	Log-uniform distribution (1GeV-1TeV)
N. of events	100,000

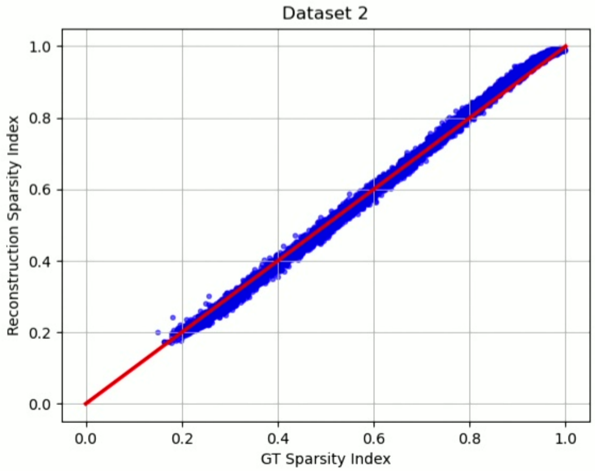


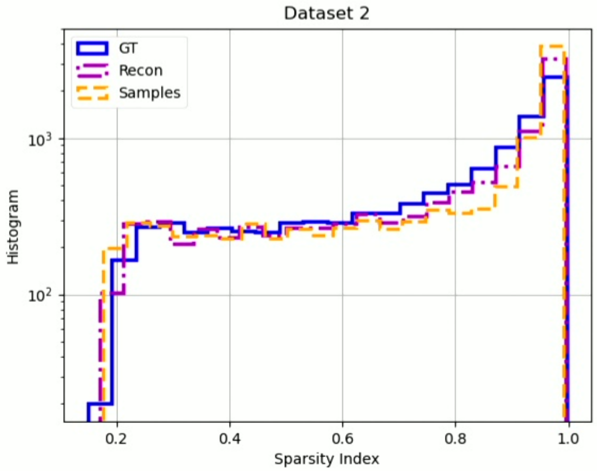
Results

- Chimera Topology
- RBM



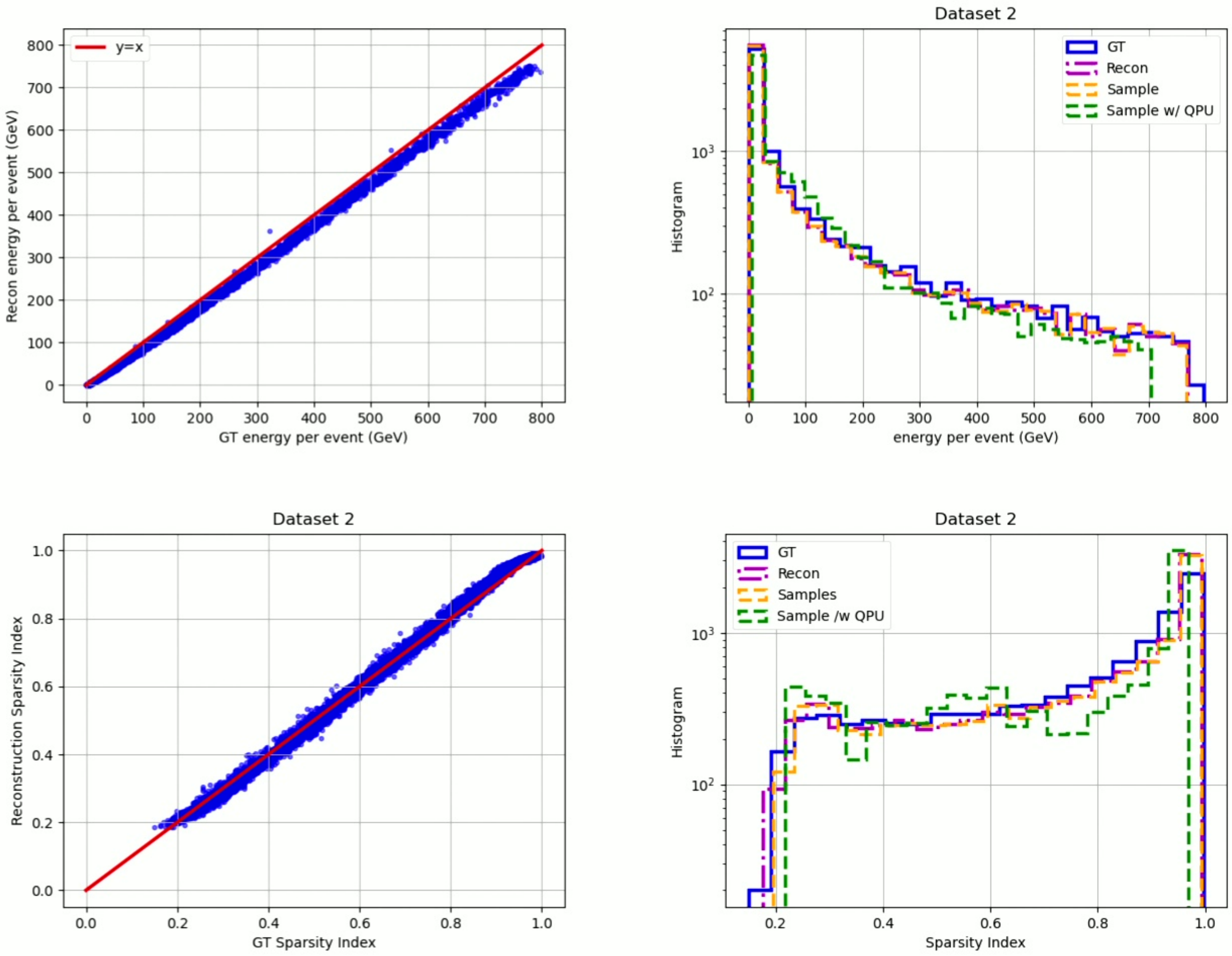






Results

- Pegasus Topology
- QPU & RBM



Recipe:

1.- Generate N samples via Gibbs sampling in RBM.

2.- Generate N samples using the QA.

3.- Compute the arithmetic mean over the energy samples, both, $\langle E_{RBM} \rangle$ and $\langle E_{QPU} \rangle$.

4.- Obtain $\beta_{QPU} = \frac{\langle E_{RBM} \rangle}{\langle E_{QPU} \rangle}$

$$\beta_{QPU} \simeq \frac{1}{8}$$

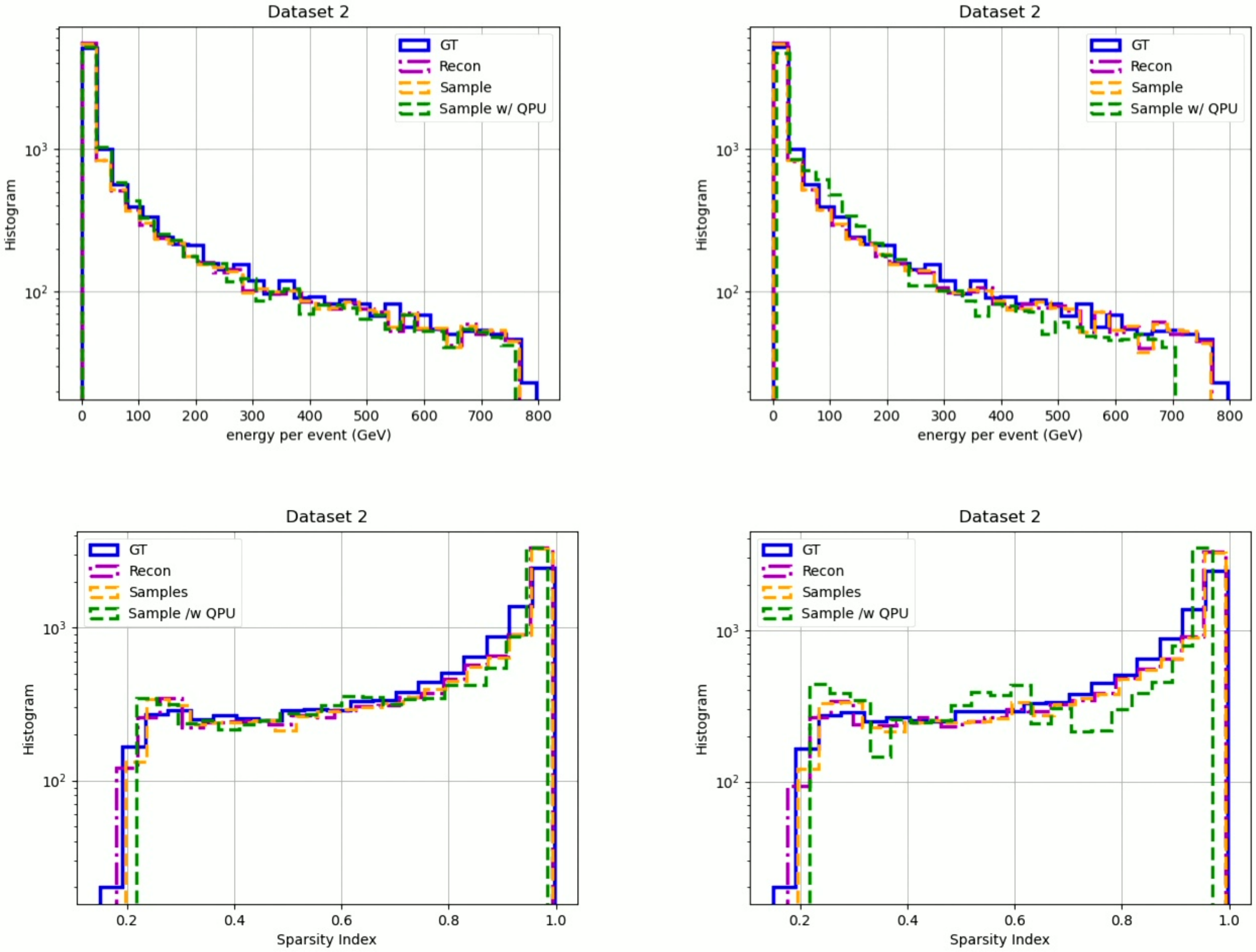
5.- Rescale $E_{QPU} = \beta_{QPU} E_{RBM}$

Results

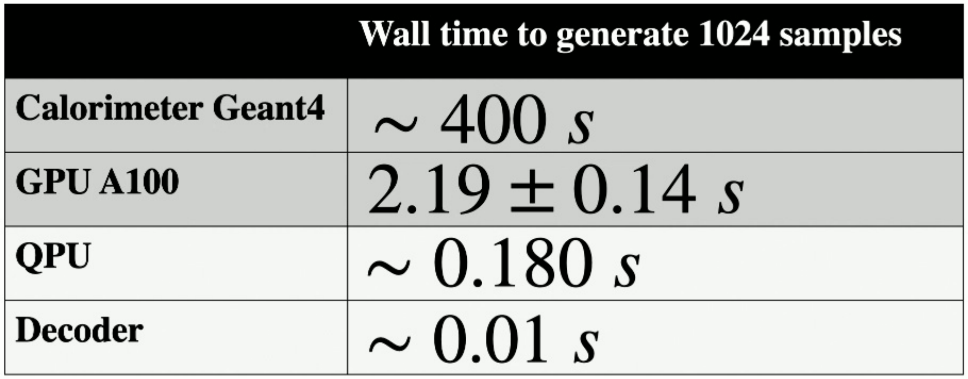
- Pegasus Topology
- QPU & RBM

After rescaling

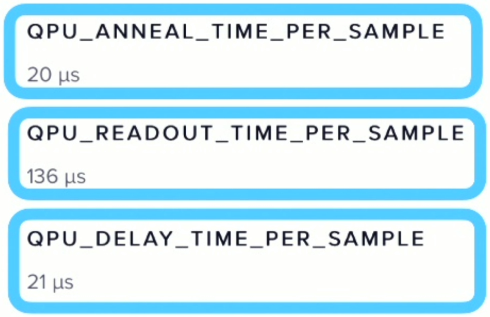
Before rescaling



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

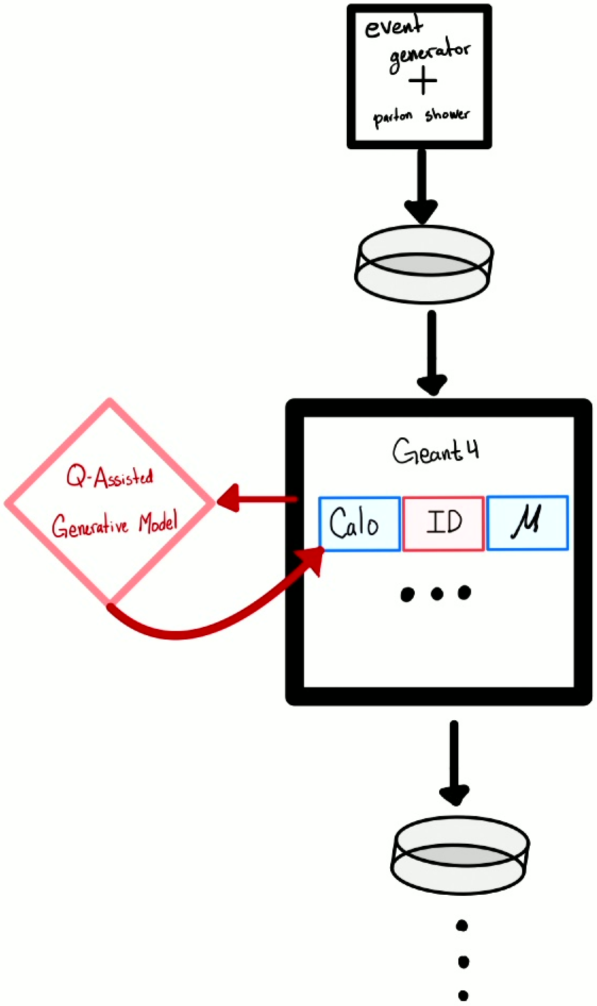
QPU ~12x faster than GPU

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

Conclusions

First thoughts on infrastructure

- Task specific partial information routing.
  - Particle type
  - Energy of incidence
  - Location
  - Etc.
- Dedicated QPU + GPU resources + networking.
- Event merging back to Geant4 record.
- Batch zipping



Conclusions

The remaining challenges to address are as follows:

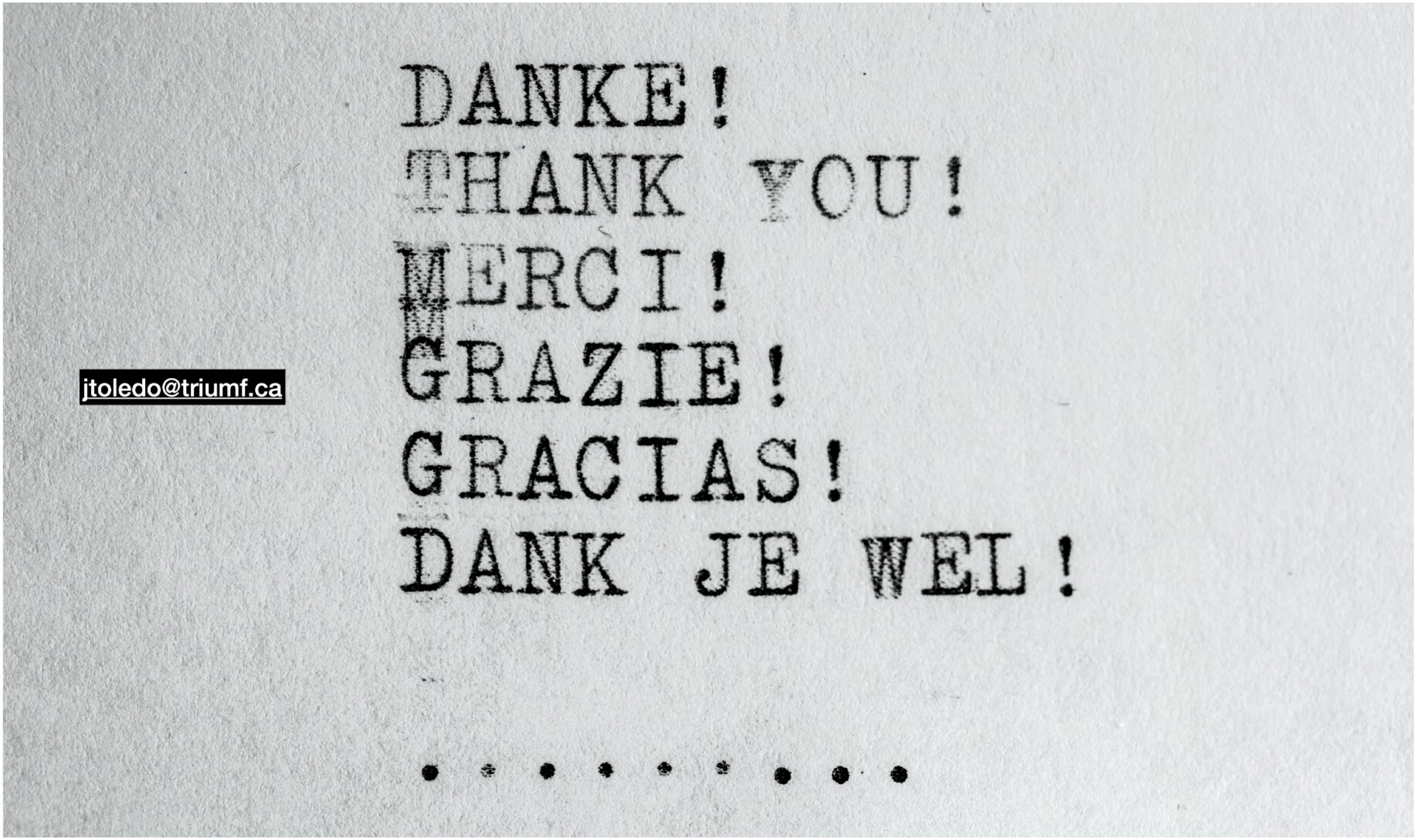
- Defining the Level of Accuracy.
- Enhancing Tail Distribution Modeling: Improving the representation of long tails in distributions is essential. Occasional fluctuations, although rare in the training sample, must be integrated into the model, as they can be significantly magnified during the event selection process.
- Modeling Complex Detector Geometries: Real-world calorimeters are intricate 3D structures, and their cells can be likened to irregularly shaped voxels. While proof of concepts often focuses on regions with translation symmetry, a production model should have the capacity to handle intricate features like edges and variations in granularity within the detector's full complexity.

[1] Rousseau D. Experimental Particle Physics and Artificial Intelligence. InArtificial Intelligence for Science: A Deep Learning Revolution 2023 (pp. 447-464).

[2] Alekseev A, Kiryanov A, Klimentov A, Korchuganova T, Mitsyn V, Oleynik D, Smirnov A, Smirnov S, Zarochentsev A. 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.

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

[4] Abhishek A, Drechsler E, Fedorko W, Stelzer B. CaloDVAE: Discrete Variational Autoencoders for Fast Calorimeter Shower Simulation. arXiv preprint arXiv:2210.07430. 2022 Oct 14.



DANKE!
THANK YOU!
MERCI!
GRAZIE!
GRACIAS!
DANK JE WEL!

. . . .

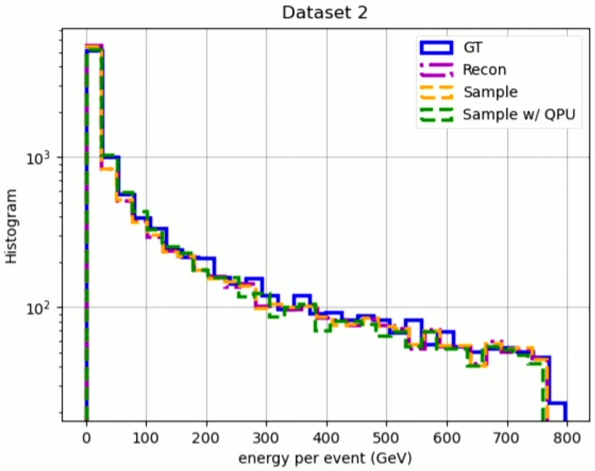
jtoledo@triumf.ca

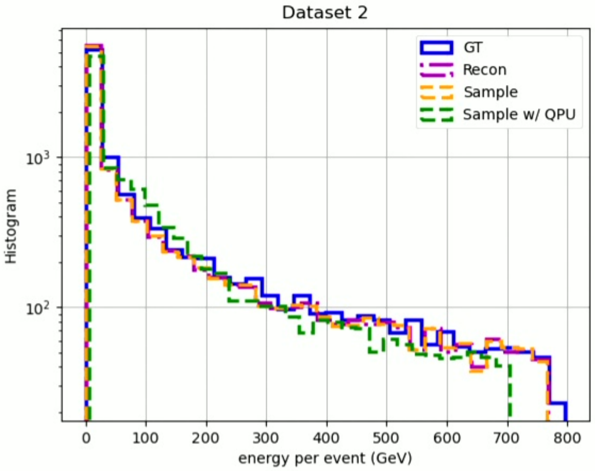
Results

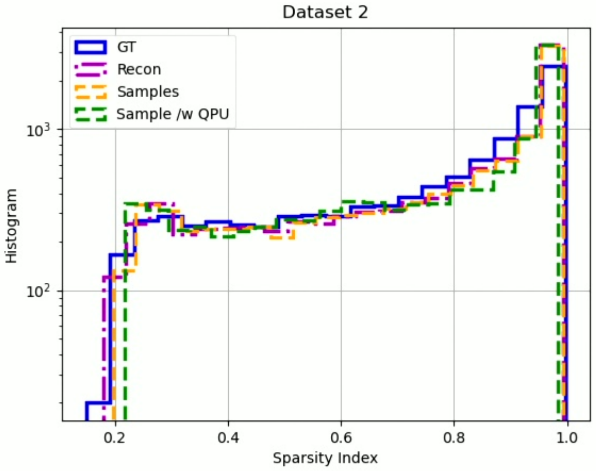
- Pegasus Topology
- QPU & RBM

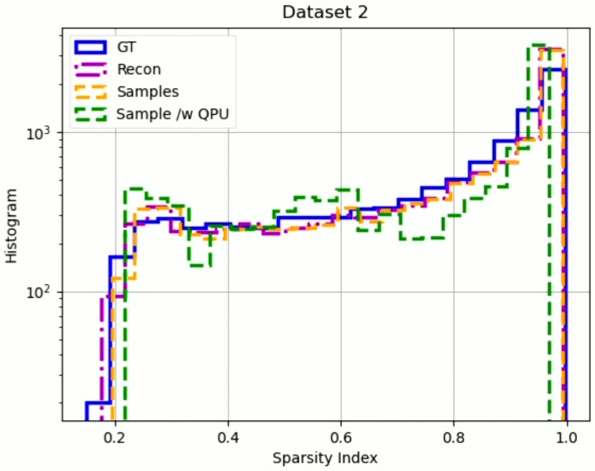
After rescaling

Before rescaling

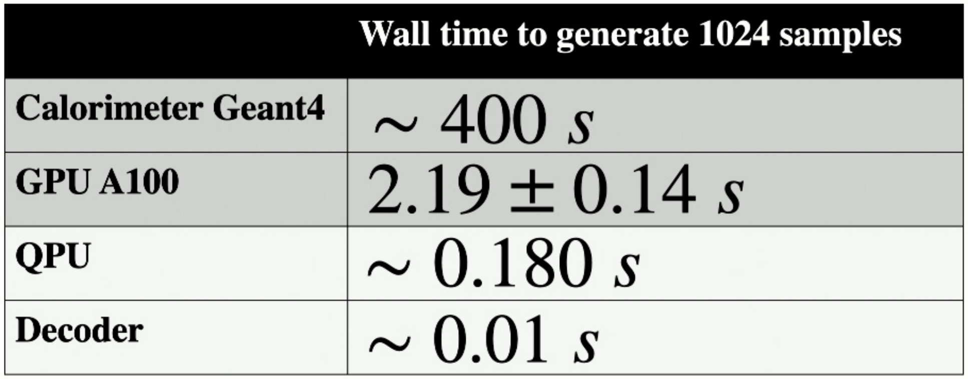




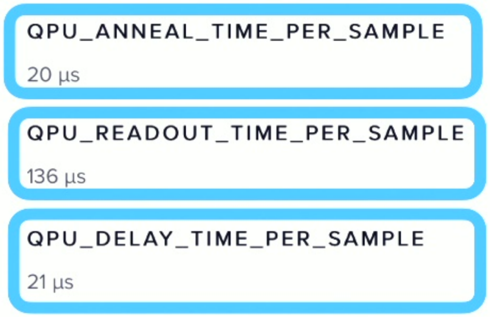




Results



	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

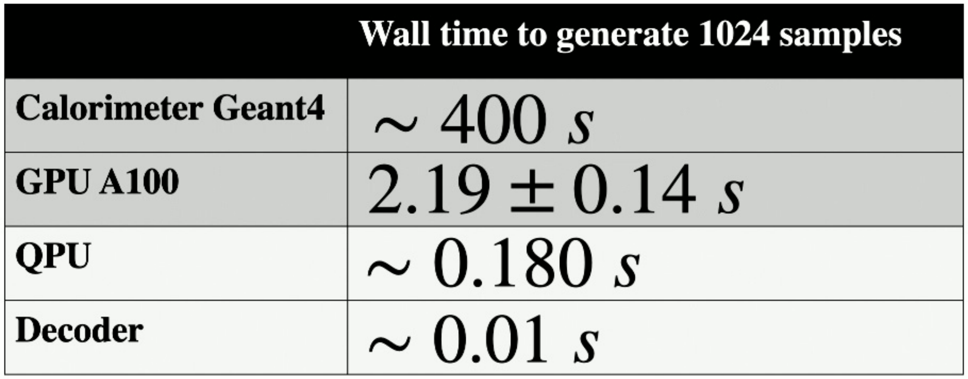
Geant4 time per sample

O(1) s

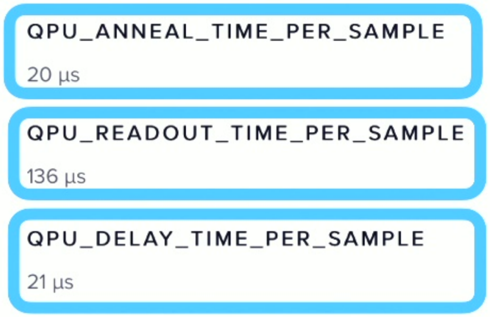
QPU ~12x faster than GPU

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

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

Conclusions

The remaining challenges to address are as follows:

- Defining the Level of Accuracy.
- Enhancing Tail Distribution Modeling: Improving the representation of long tails in distributions is essential. Occasional fluctuations, although rare in the training sample, must be integrated into the model, as they can be significantly magnified during the event selection process.
- Modeling Complex Detector Geometries: Real-world calorimeters are intricate 3D structures, and their cells can be likened to irregularly shaped voxels. While proof of concepts often focuses on regions with translation symmetry, a production model should have the capacity to handle intricate features like edges and variations in granularity within the detector's full complexity.

[1] Rousseau D. Experimental Particle Physics and Artificial Intelligence. InArtificial Intelligence for Science: A Deep Learning Revolution 2023 (pp. 447-464).

[2] Alekseev A, Kiryanov A, Klimentov A, Korchuganova T, Mitsyn V, Oleynik D, Smirnov A, Smirnov S, Zarochentsev A. 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.

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

[4] Abhishek A, Drechsler E, Fedorko W, Stelzer B. CaloDVAE: Discrete Variational Autoencoders for Fast Calorimeter Shower Simulation. arXiv preprint arXiv:2210.07430. 2022 Oct 14.

Motivation

- Simulation plays a significant role in the design of future experiments but also in the analysis of the current ones.
- One single event fully simulated with Geant4 in an LHC experiment requires about O(1) CPU seconds.
- The calorimeter simulation is by far dominating the total simulation time.
- **AI generator models are being developed in particular for the simulation of calorimeters.**

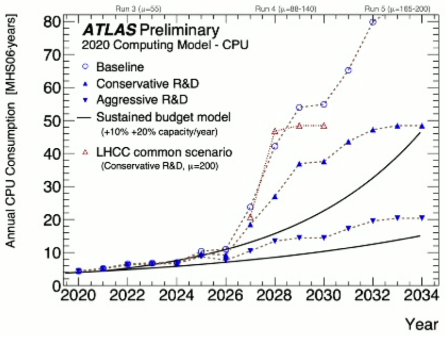
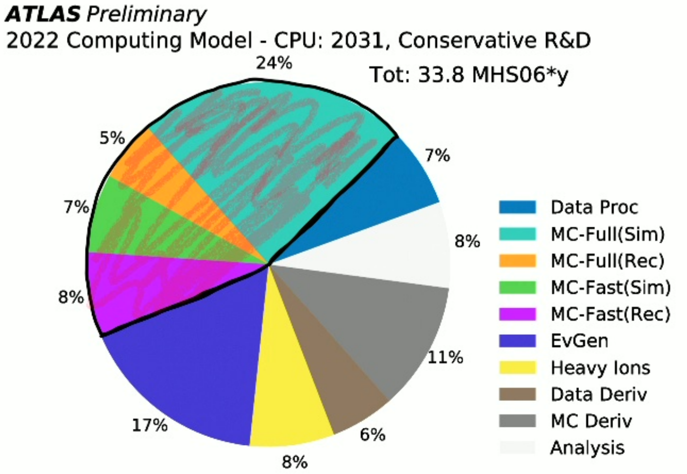


Figure 1. Projected CPU requirements of ATLAS experiment between 2020 and 2034 based on 2020 assessment. Three scenarios are shown, corresponding to an ambitious ("aggressive"), modest ("conservative") and minimal ("baseline") development program. The black lines indicate annual improvements of 10% and 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).


ATLAS Preliminary
2022 Computing Model - CPU: 2031, Conservative R&D
Tot: 33.8 MHS06*y
Data Proc
MC-Full(Sim)
MC-Full(Rec)
MC-Fast(Sim)
MC-Fast(Rec)
EvGen
Heavy Ions
Data Deriv
MC Deriv
Analysis