Style-based quantum generative adversarial networks for Monte Carlo events

C. Bravo-Prieto, J. Baglio, M. C‘e, A. Francis, D. M. Grabowska and S. Carrazza - Quantum 6, 777 (2022)

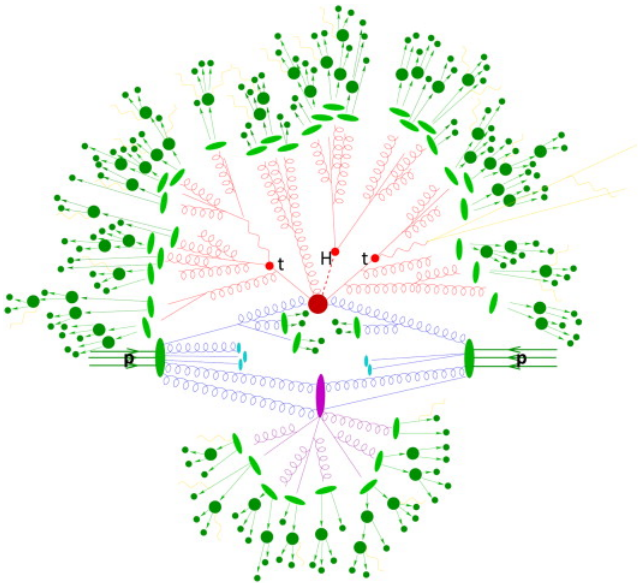
Andrea Pasquale on the behalf of Stefano Carrazza

13th October 2022

Outline

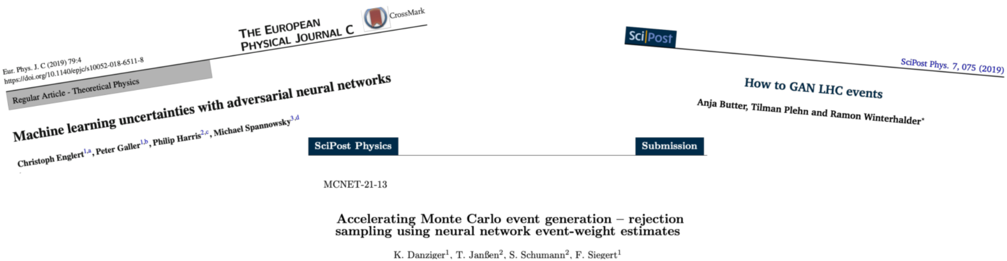
Problem: event generation at LHC

Monte Carlo event simulation is **very intensive** and requires lots of **computational power**.



Machine learning approach to event generation

Since 2018, many papers have approached event generation with machine learning techniques:



Eur. Phys. J. C (2019) 79:4
https://doi.org/10.1140/epjc/s10052-018-6511-8

THE EUROPEAN PHYSICAL JOURNAL C

CrossMark

Regular Article - Theoretical Physics

Machine learning uncertainties with adversarial neural networks

Christoph Englert[1,a], Peter Galler[1,b], Philip Harris[2,c], Michael Spannowsky[3,d]

SciPost Physics | Submission

MCNET-21-13

Accelerating Monte Carlo event generation – rejection sampling using neural network event-weight estimates

K. Danziger[1], T. Janßen[2], S. Schumann[2], F. Siegert[1]

SciPost

SciPost Phys. 7, 075 (2019)

How to GAN LHC events

Anja Butter, Tilman Plehn and Ramon Winterhalder*

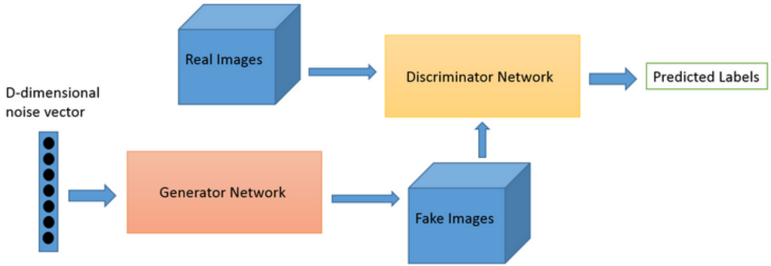
General approach:

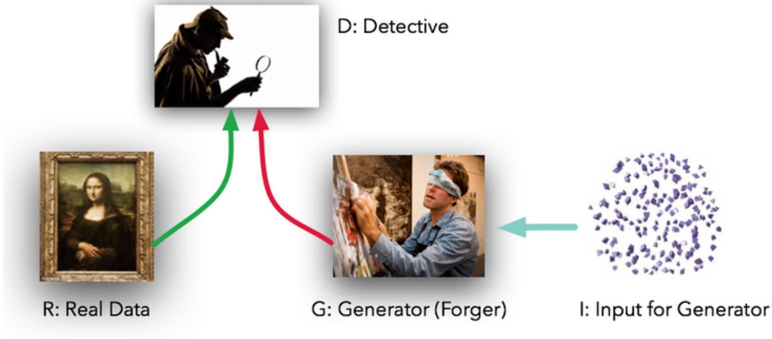
1. Train unsupervised models on **small** dataset to learn underlying pdf
2. Generate **more data** using those models $\Rightarrow$ data augmentation

Methodology

Unsupervised Learning: Generative Adversarial Networks

Two networks competing: generator produces **fake** data, while the discriminator distinguishes between **real** input data and fake data (produced by the generator).





Training: adapt alternatively the generator $G(\phi_g, z)$ and the discriminator $D(\phi_d, x)$
Metrics: binary cross-entropy for the loss functions:

- Generator loss function:

$$\mathcal{L}_G(\phi_g, \phi_d) = -\mathbb{E}_{z \sim p_{\text{prior}}(z)}[\log D(\phi_d, G(\phi_g, z))]$$

- Discriminator loss function:

$$\mathcal{L}_D(\phi_g, \phi_d) = \mathbb{E}_{x \sim p_{\text{real}}(x)}[\log D(\phi_d, x)] + \mathbb{E}_{z \sim p_{\text{prior}}(z)}[\log\big(1 - D(\phi_d, G(\phi_g, z))\big)] \,.$$

Game theory: min-max two-player game to reach Nash equilibrium

$$\min_{\phi_g} \mathcal{L}_G(\phi_g, \phi_d) \quad \max_{\phi_d} \mathcal{L}_D(\phi_g, \phi_d)$$

Machine learning with quantum computing?

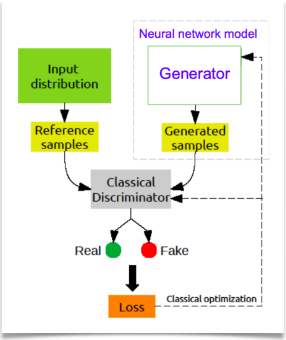
Can we develop an efficient GAN-like model for event generation using quantum hardware?

Why Quantum ML?

- Proof-of-concept, study new architectures.
    - fast inference (native representation / sampling)?
    - fast training / compact models with few parameters?
- Obtain a hardware representation (analogy with GPU and FPGA).
- Lower power consumption.

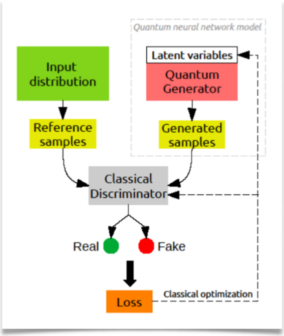
Hybrid approach

Classical setup:

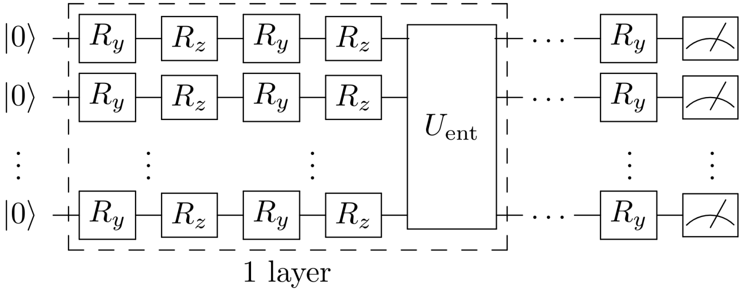


Only the generator becomes quantum

Hybrid quantum-classical setup:



Quantum generator: a series of quantum layers with rotation and entanglement gates



Style-based approach

The noise is inserted in every gate and not only in the initial quantum state

$$R^{l,m}_{y,z}(\vec{\phi}_g, \vec{z}) = R_{y,z}(\phi_g^{(l)} z^{(m)} + \phi_g^{(l-1)})$$

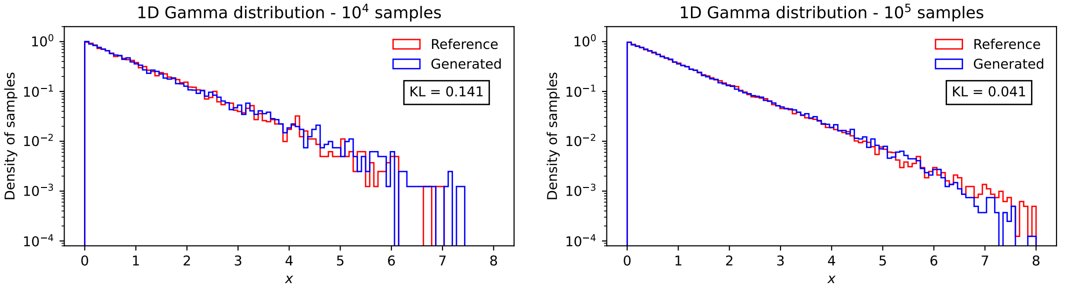
Results

Validation: 1D Gamma distribution

Assessing the validity of the approach: train and test on known toy model distribution

With 1 qubit, one layer, using 100 bins: 1D Gamma function

$$p_\gamma(x, \alpha, \beta) = x^{\alpha-1} \frac{e^{-x/\beta}}{\beta^\alpha \Gamma(\alpha)}$$



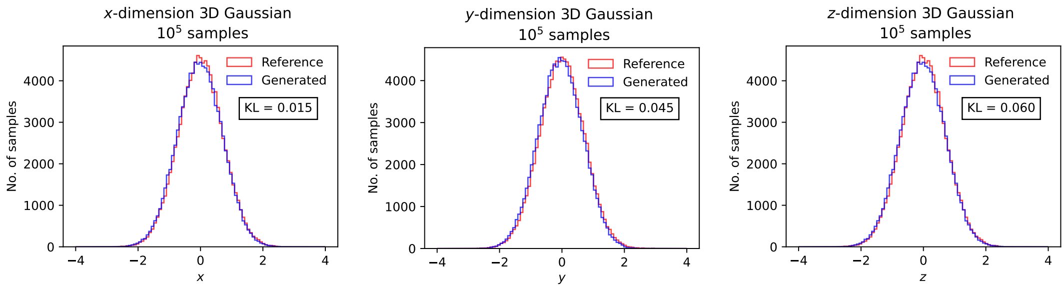
- Train on 10^4 samples until convergence is reached.
- Use generator to generate 10^4 and 10^5 samples to demonstrate data augmentation

Validation: 3D correlated Gaussian distribution

Test whether the style-qGAN captures correlations: train on 3D Gaussian distribution, with

$$p(\vec{x}) \propto \exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})\right], \quad \mu = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0.5 & 0.1 & 0.25 \\ 0.1 & 0.5 & 0.1 \\ 0.25 & 0.1 & 0.5 \end{pmatrix}.$$

Using 3 qubits, one layer, 100 bins:

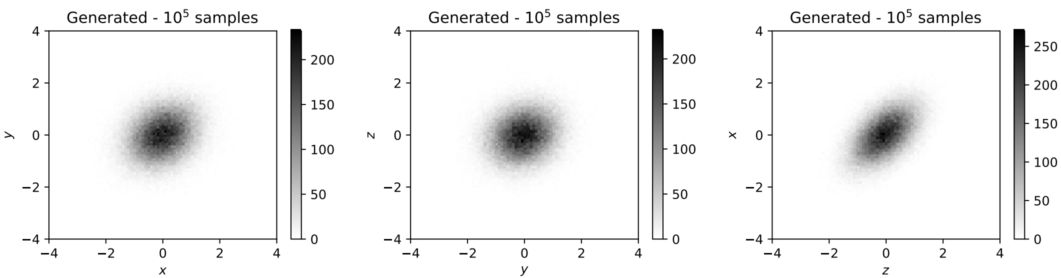


Validation: 3D correlated Gaussian distribution

Test whether the style-qGAN captures correlations: train on 3D Gaussian distribution, with

$$p(\vec{x}) \propto \exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})\right], \quad \mu = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0.5 & 0.1 & 0.25 \\ 0.1 & 0.5 & 0.1 \\ 0.25 & 0.1 & 0.5 \end{pmatrix}.$$

Using 3 qubits, one layer, 100 bins:



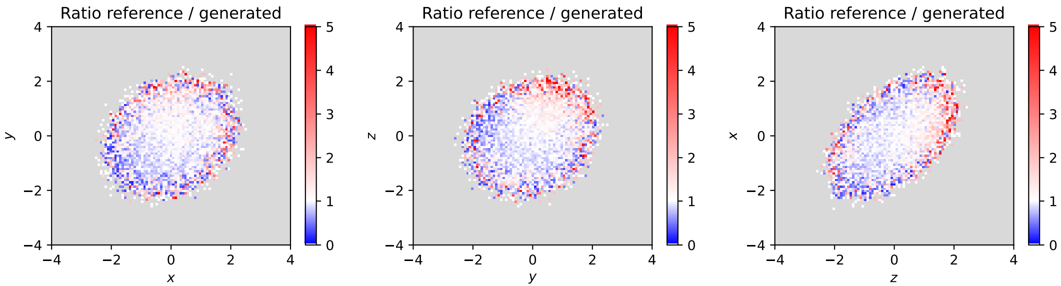
Correlations are well captured!

Validation: 3D correlated Gaussian distribution

Test whether the style-qGAN captures correlations: train on 3D Gaussian distribution, with

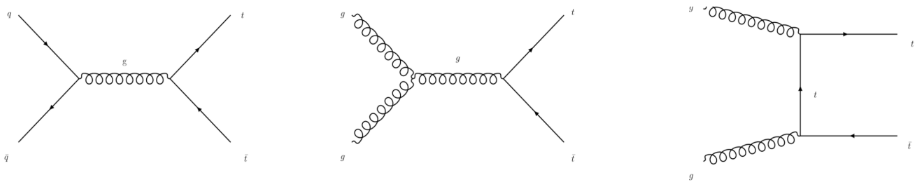
$$p(\vec{x}) \propto \exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})\right], \quad \mu = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0.5 & 0.1 & 0.25 \\ 0.1 & 0.5 & 0.1 \\ 0.25 & 0.1 & 0.5 \end{pmatrix}.$$

Using 3 qubits, one layer, 100 bins:



Correlations are well captured!

Testing the style-qGAN with real data: proton-proton collision $pp \to t\bar{t}$

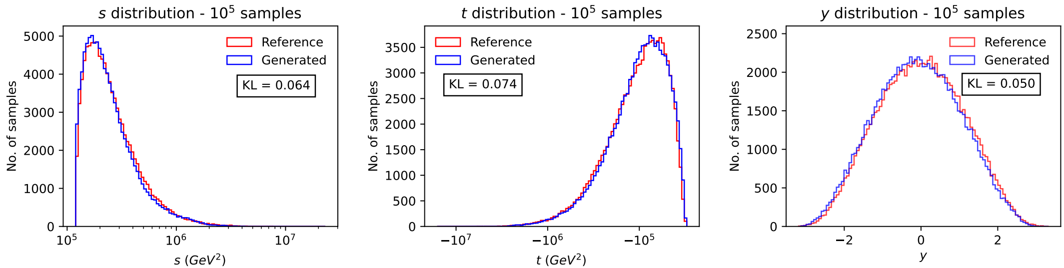


Training and reference samples generated with MadGraph5 aMC@NLO

Training set of 10^4 samples, **Mandelstam variables** (s, t) and rapidity y.

Simulation with LHC data

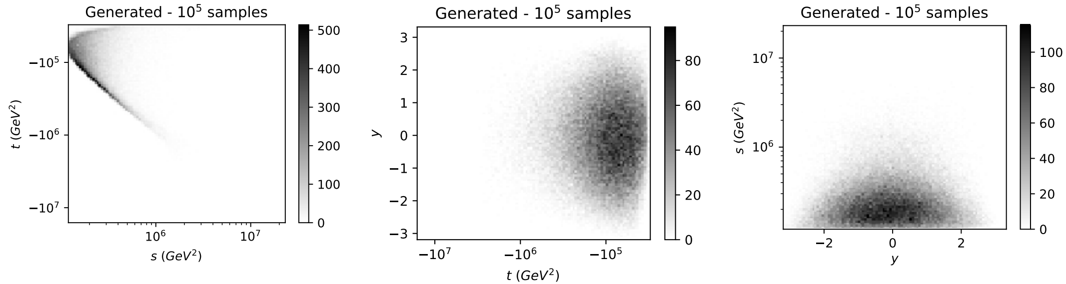
After training, we assess the performance with simulations: 3 qubits, 2 layers, 100 bins



Remarkable low KL divergences with data augmentation!

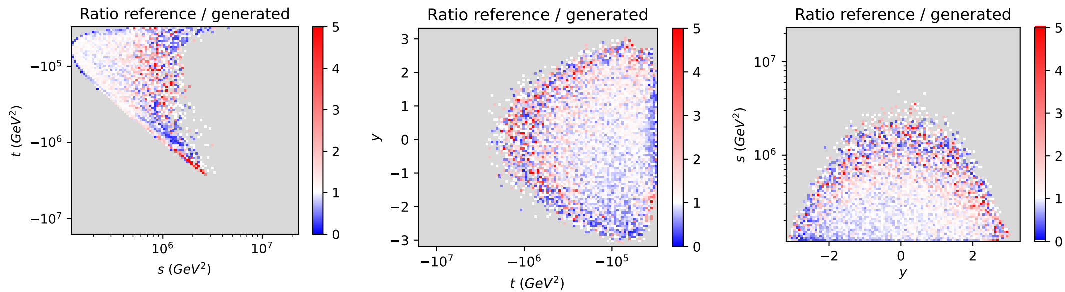
Simulation with LHC data

After training, we assess the performance with simulations: 3 qubits, 2 layers, 100 bins



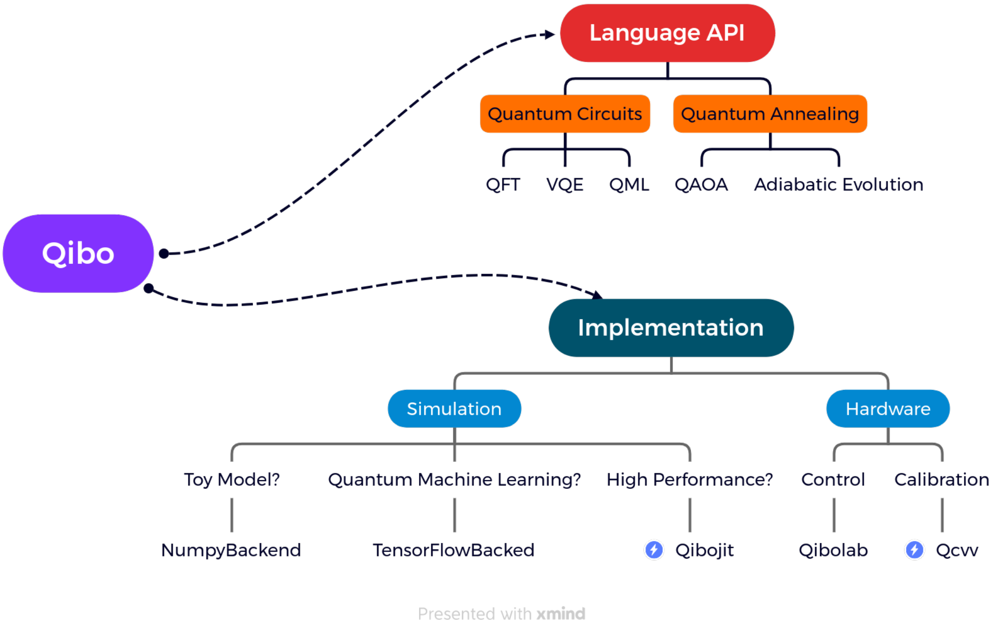
Correlations are well captured!

After training, we assess the performance with simulations: 3 qubits, 2 layers, 100 bins

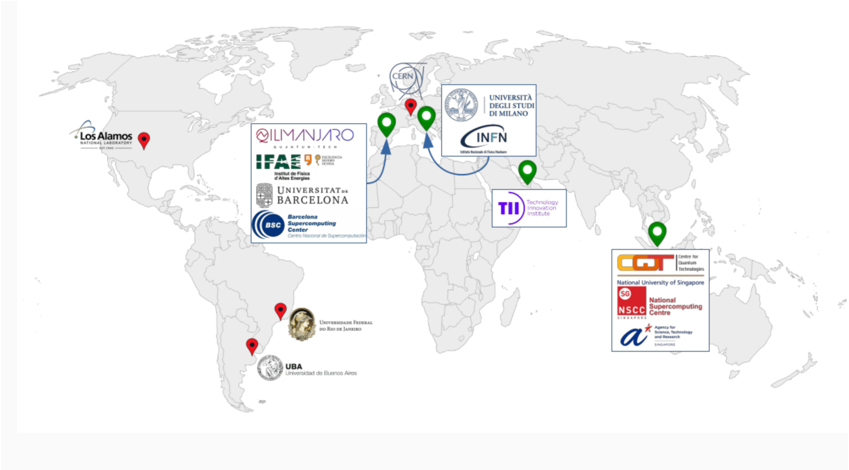


Are these results maintained on real hardware?

Qibo is an **open-source** full stack API for quantum simulation and quantum hardware control and calibration.



Qibo



Los Alamos
NATIONAL LABORATORY
CERN
QILIMANJARO
QUANTUM-TECH
IFAE
Institut de Física d'Altes Energies
UNIVERSITAT DE BARCELONA
Barcelona Supercomputing Center
Centro Nacional de Supercomputación
UNIVERSITÀ DEGLI STUDI DI MILANO
INFN
TII
Technology Innovation Institute
Centre for Quantum Technologies
National University of Singapore
NSCC
National Supercomputing Centre
Agency for Science, Technology and Research
SINGAPORE
Universidade Federal do Rio de Janeiro
UBA
Universidad de Buenos Aires

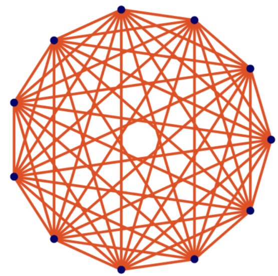
Result on Quantum Hardware

Testing different architectures

Superconducting transmon qubits: *ibmq_santiago* with 2-neighbouring site connectivity

Access via IBM Q cloud service

Trapped ion technology: *ionQ* with all-to-all connectivity

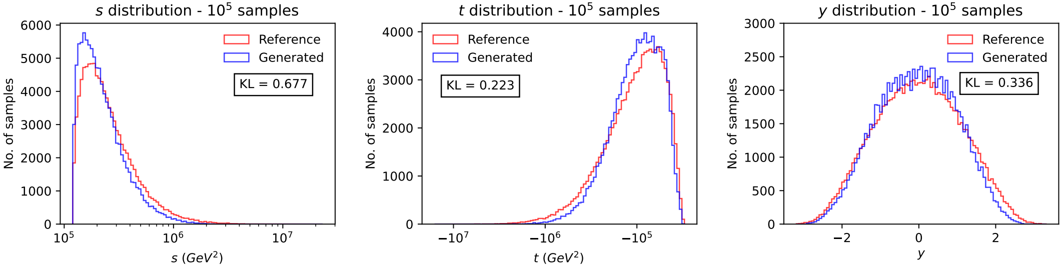


Access via Amazon Web Services

Results on IBM Q hardware

ibmq_santiago 5-qubit machine

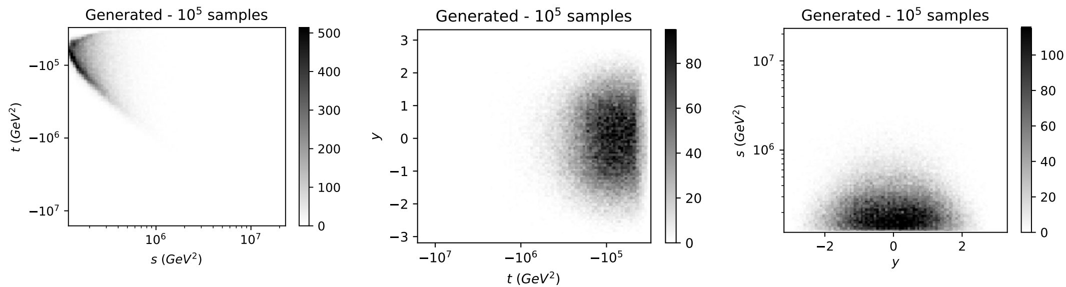
Still relatively low KL divergence!



Results on IBM Q hardware

ibmq_santiago 5-qubit machine

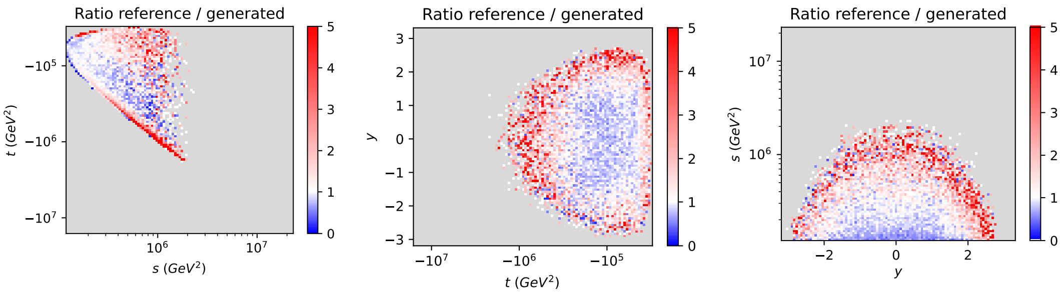
> Correlations captured on a quantum hardware!



Results on IBM Q hardware

ibmq_santiago 5-qubit machine

Still a good ratio!

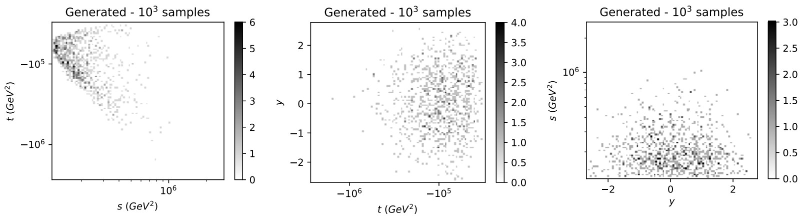


Testing different architectures

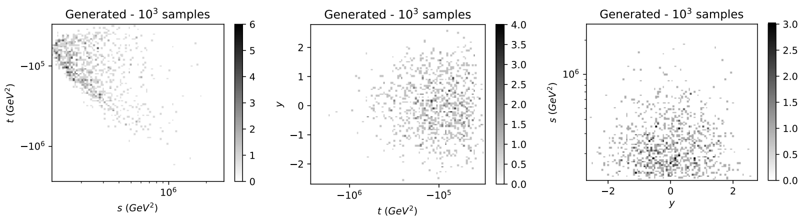
Access constraints to ionQ: test limited to 1000 samples only

> Hardware independent implementation!

IBM Q samples



ionQ samples



Outlook

Outlook

- A novel **quantum generator architecture** (style-based) has been presented.
- A test-case with real Monte Carlo event has demonstrated success: the generator has learned underlying (s, t, y) distributions and correlations for production.
- Demonstrated **data augmentation** from 10^4 training data to 10^5 generated data.
- The quantum network is **shallow**: great advantage in the current NISQ era.
- Tested on two different quantum architectures: superconducting qubits (IBM) and trapped ions (ionQ) with similar performances. The quantum generator seems quite hardware-independent.