Quantum Generative Adversarial Networks

CERN openlab Technical Workshop 2021

Su Yeon Chang (CERN, EPFL)

Steven Herbert (CQC), Sofia Vallecorsa (CERN) , Elias Combarro (U. Oviedo, CERN)

11/03/2021

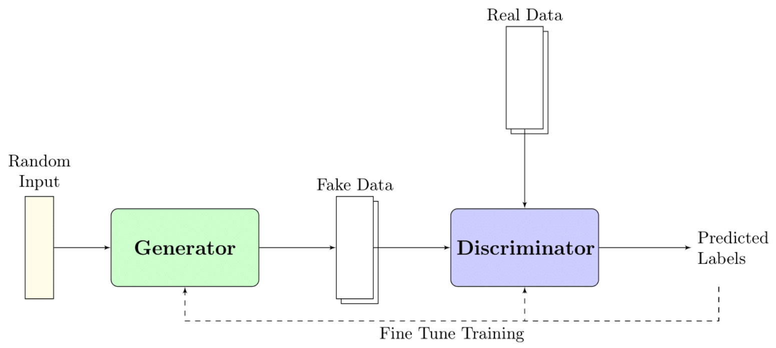
MOTIVATION

Why Quantum Generative Adversarial Networks (GAN)?

Detectors simulation :

- Tremendous amount of time required by Monte Carlo based simulation

→ **Generative Adversarial Networks**



Quantum Machine Learning :

- Compressed data representation in quantum states
- Expect faster training with less number of parameters

→ Potential advantage of Quantum GAN

- Initial work using qGAN model constructed by IBM

→ limited in reproducing a probability distribution over discrete variables

➡ **Explore different prototypes of quantum GAN to improve the model**

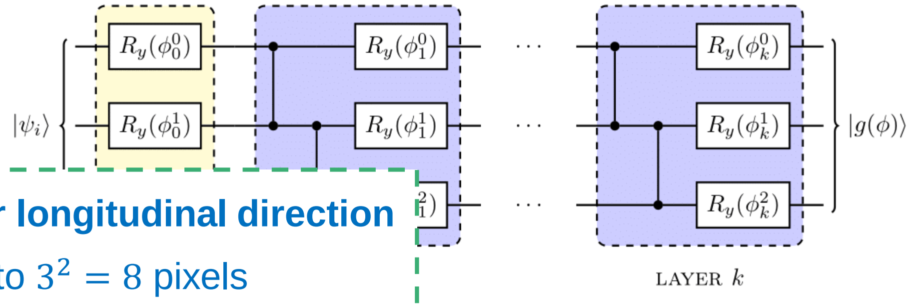
https://dl.acm.org/doi/proceedings/10.5555/2969033

Quantum GAN

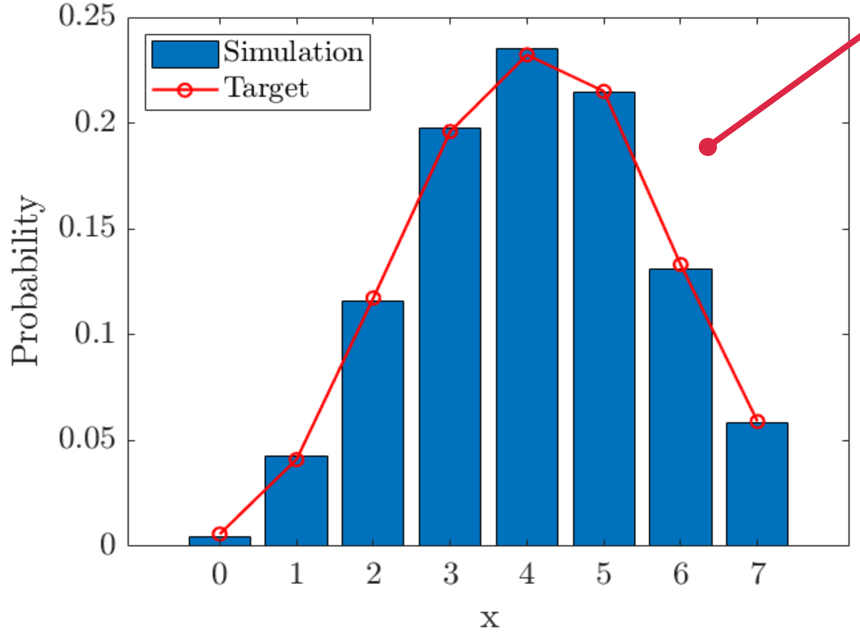
Practical qGAN model constructed by IBM

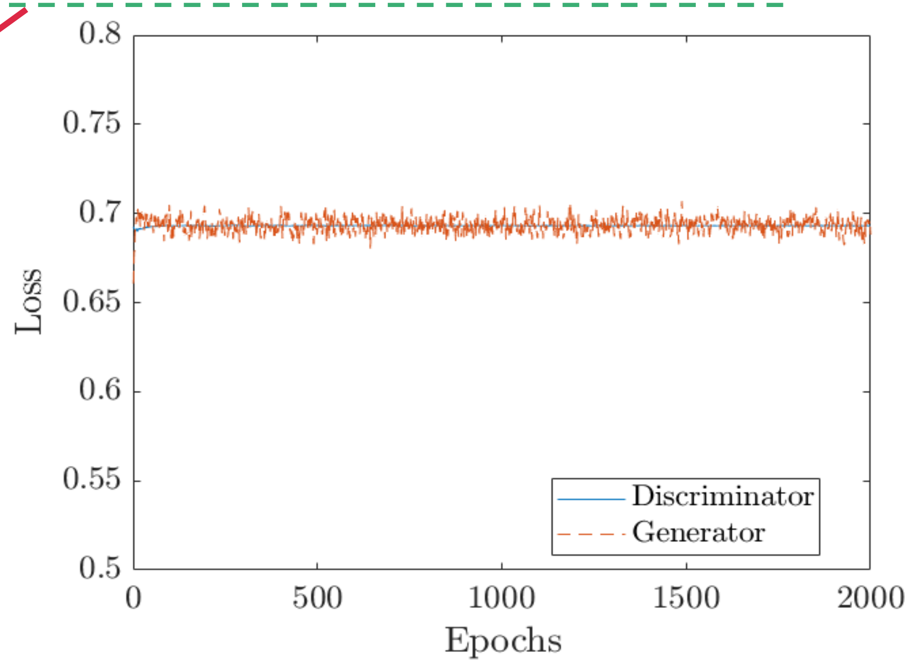
- Hybrid model : **Quantum** Generator + **Classical** Discriminator
- Efficient in loading and learning a probability over discrete values

$\rightarrow p_g(\phi)$ to approach p_{real}



- ✓ 2D image summed **over longitudinal direction**
- ✓ Normalized & Binned into $3^2 = 8$ pixels
- ✓ Averaged over 20,000 samples





Limitation

IBM qGAN model

- Limited in reproducing an average probability distribution over pixels
- Aim to reproduce a distribution over continuous variables

Need to find alternative ways to reproduce a “set” of images

Dual-PQC GAN model (in collaboration with Cambridge Quantum Computing)

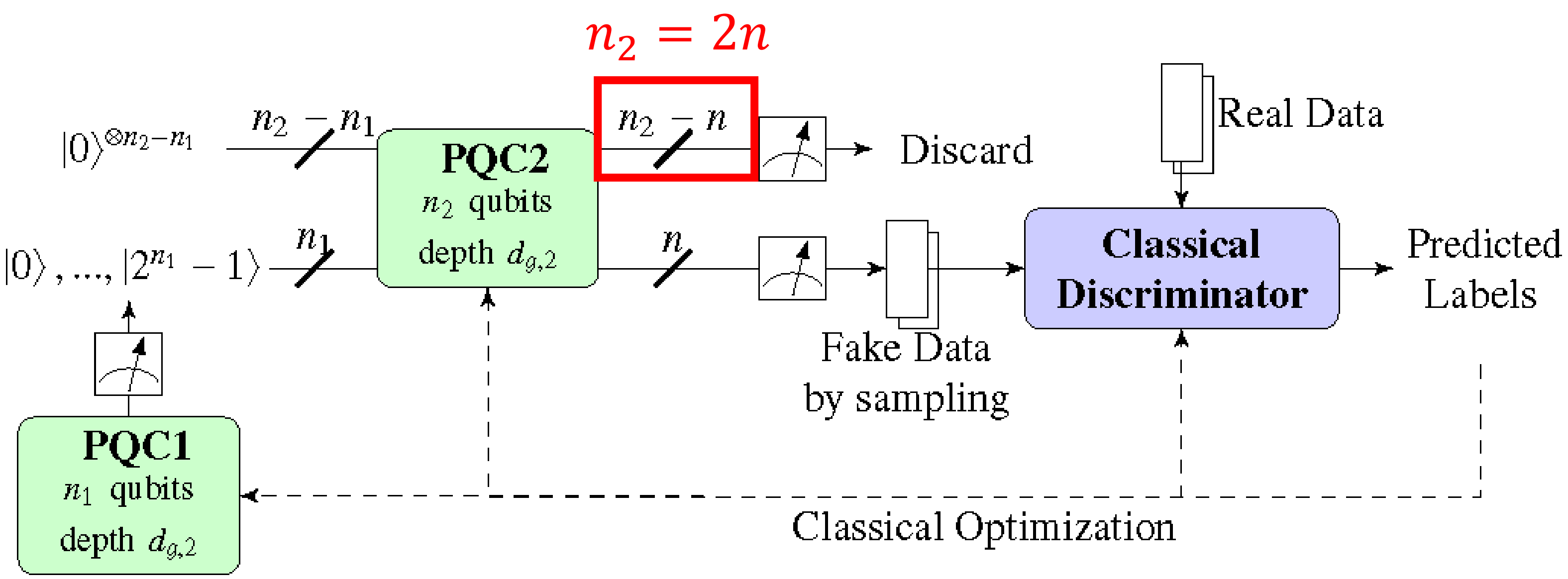
Continuous Variable Quantum GAN

Dual-PQC GAN model

Role of single generator shared by two parameterized quantum circuits (pqc)

- **PQC1** – Reproduce the distribution over 2^{n_1} images of size 2^n
- **PQC2** – Reproduce amplitudes over 2^n pixels on one image

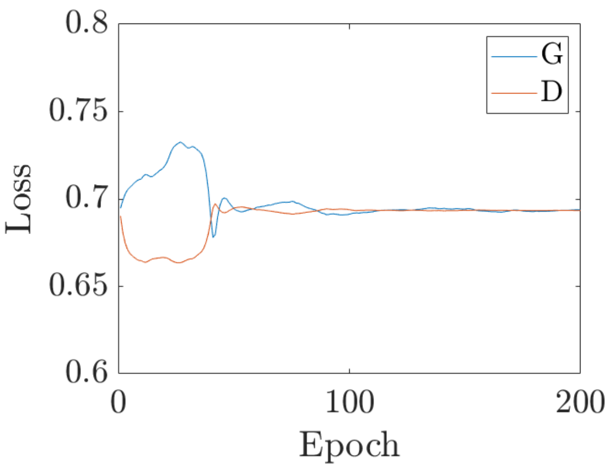
➡ **2^{n_1} images of size 2^n**

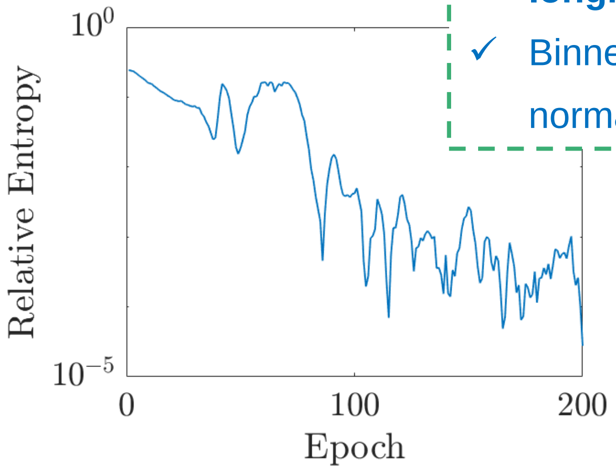


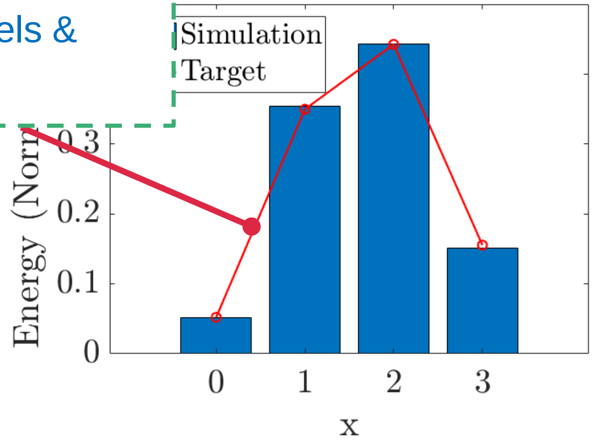
Application of Dual-PQC GAN in HEP

$n = 2, n_1 = 4, n_2 = 4, depth_{g1} = 2, depth_{g2} = 16$

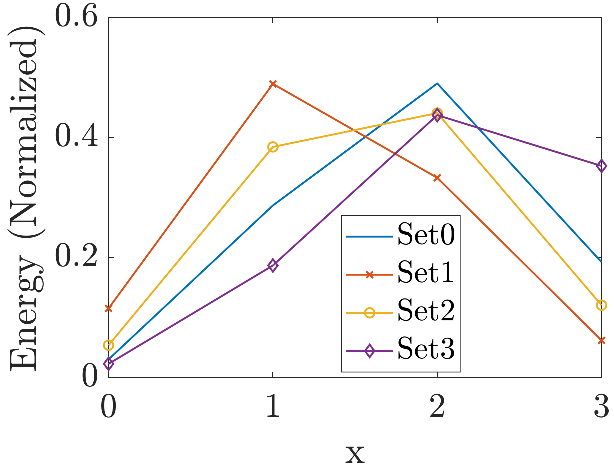
- ✓ 2D image summed **over longitudinal direction**
- ✓ Binned into 4 pixels & normalized



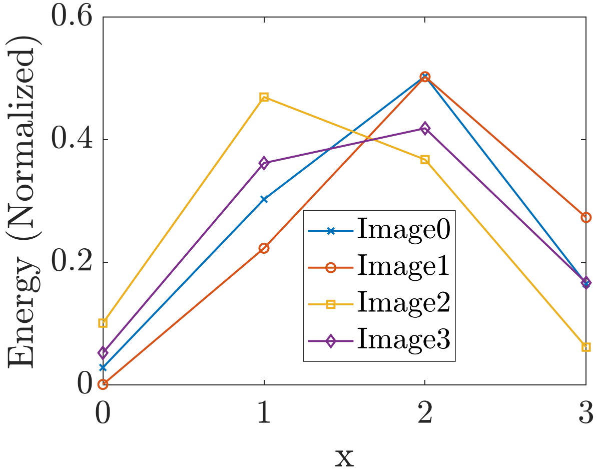




Real



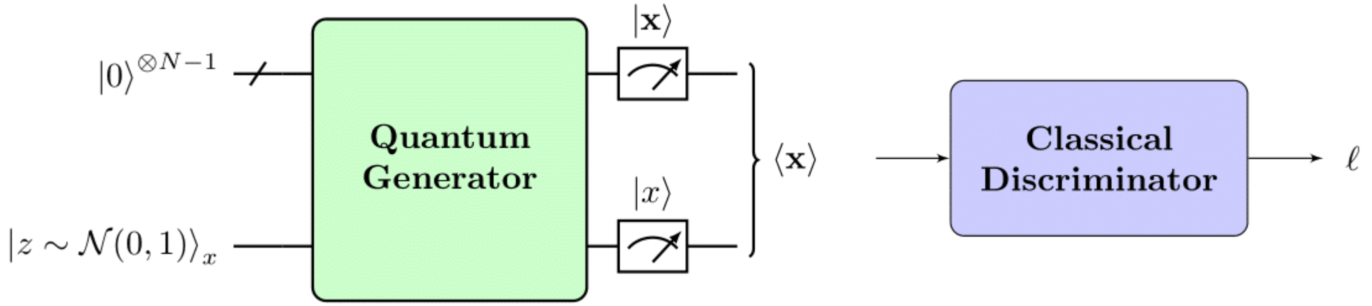
Generated



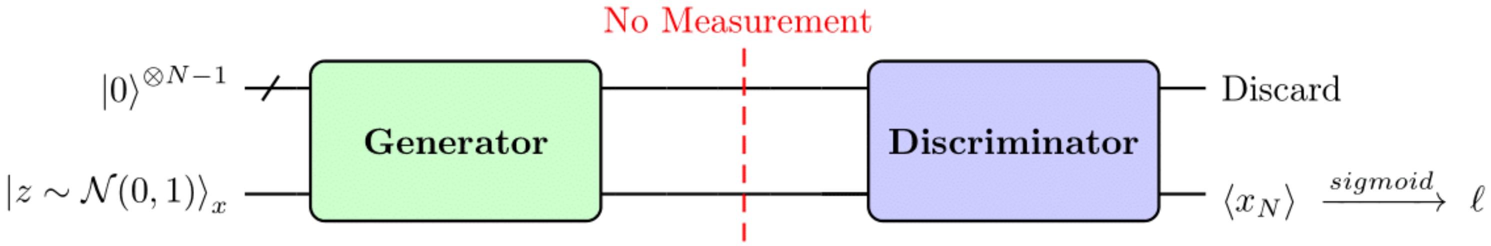
CV qGAN

Quantum GAN with a generator constructed by Continuous-variable NN

- Continuous-variable QC : Fundamental information-carrying units = **Qumodes**
- CV Neural network with CV gates (N. Killoran et al. 2019) → **Construct CV qGAN**



Hybrid model : Quantum Generator & Classical Discriminator



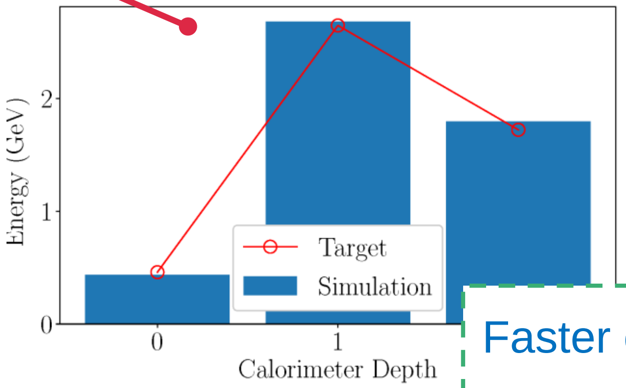
Fully Quantum model : Quantum Generator & Quantum Discriminator https://arxiv.org/abs/2101.11132

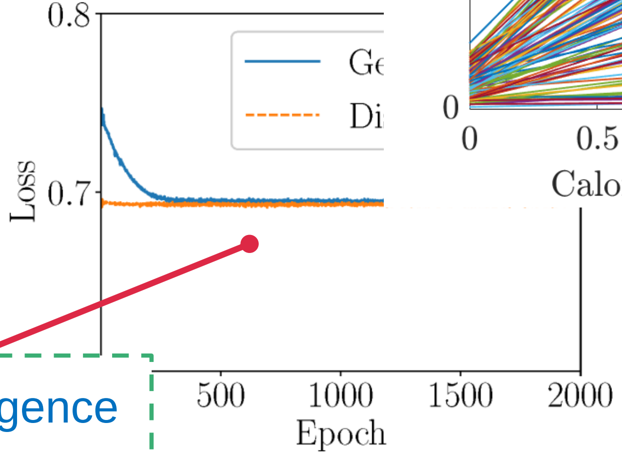
…sing latent sp… …nsion

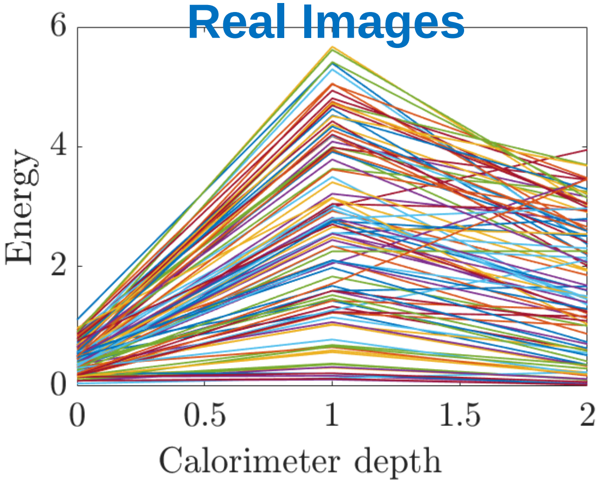
- ✓ 2D image summed **over longitudinal direction**
- ✓ Binned into 3 pixels
- ✓ No normalization required

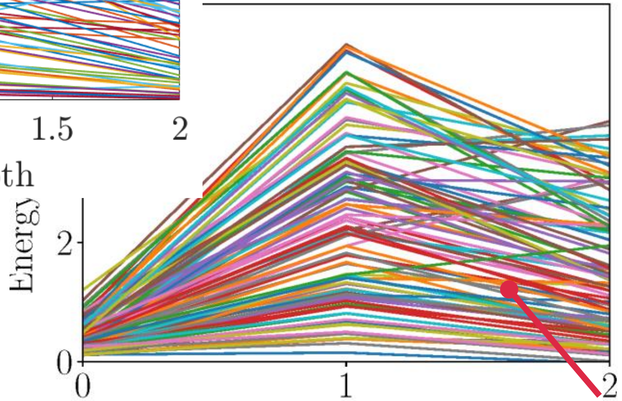
…dimension = 3

- Classical GAN, $n_{params} \approx 45000$





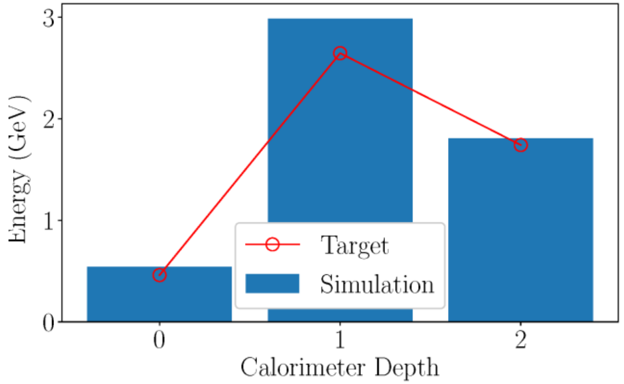


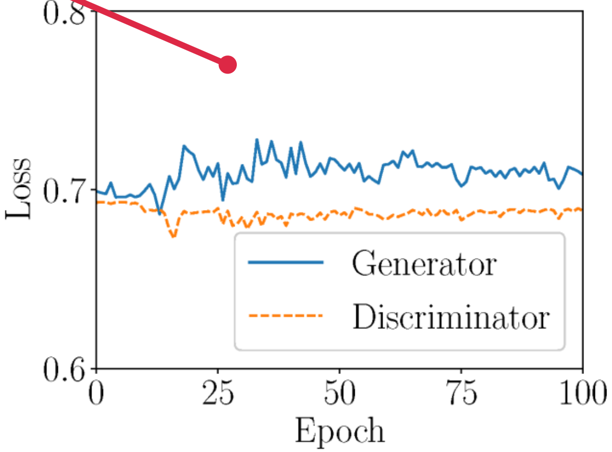


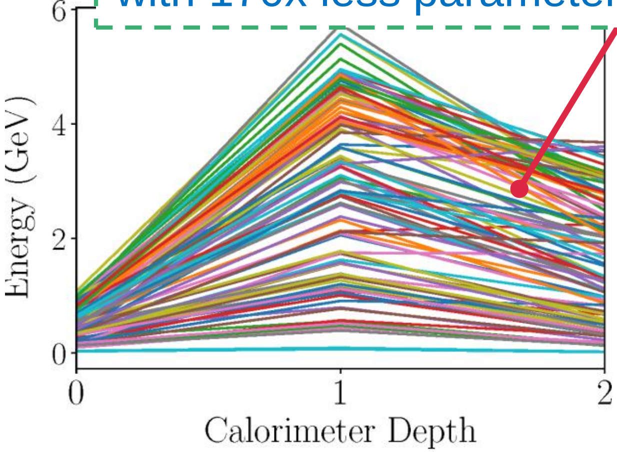
Faster convergence for CV qGAN

Can achieve similar performance with 170x less parameters

- Hybrid CV qGAN, $n_{params} \approx 260$







Conclusion

Dual-PQC GAN & CV qGAN

- Two different prototypes of quantum GAN to reproduce a set of images
  1) Dual-PQC GAN, 2) CV qGAN
- Able to reproduce images with reduced size (3~4 pixels)

Future works

- Test fully quantum CV qGAN model
- Increase problem size
- Extend to other use-cases (e.g. Image generation for Earth Observation)

QUESTIONS?

su.yeon.chang@cern.ch

Appendix A : qGAN in HEP (details)

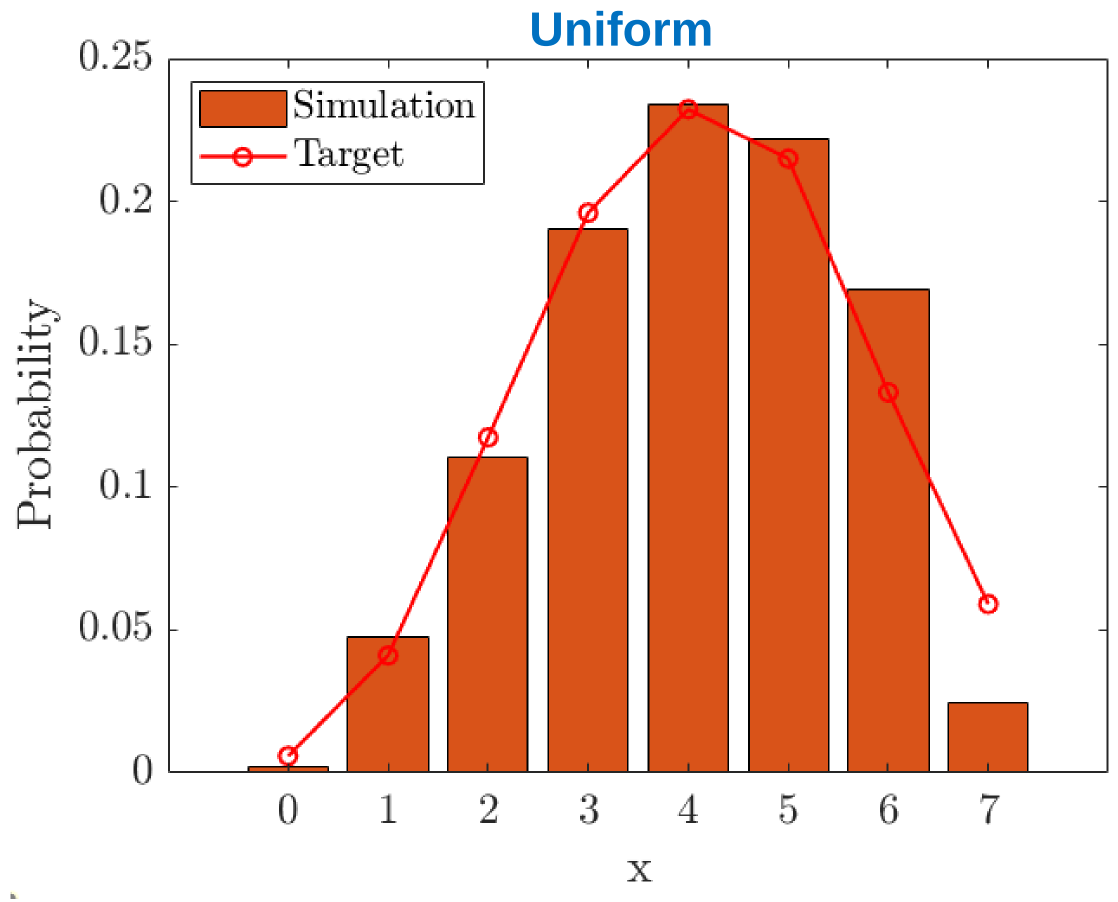
Preparation of Initial State

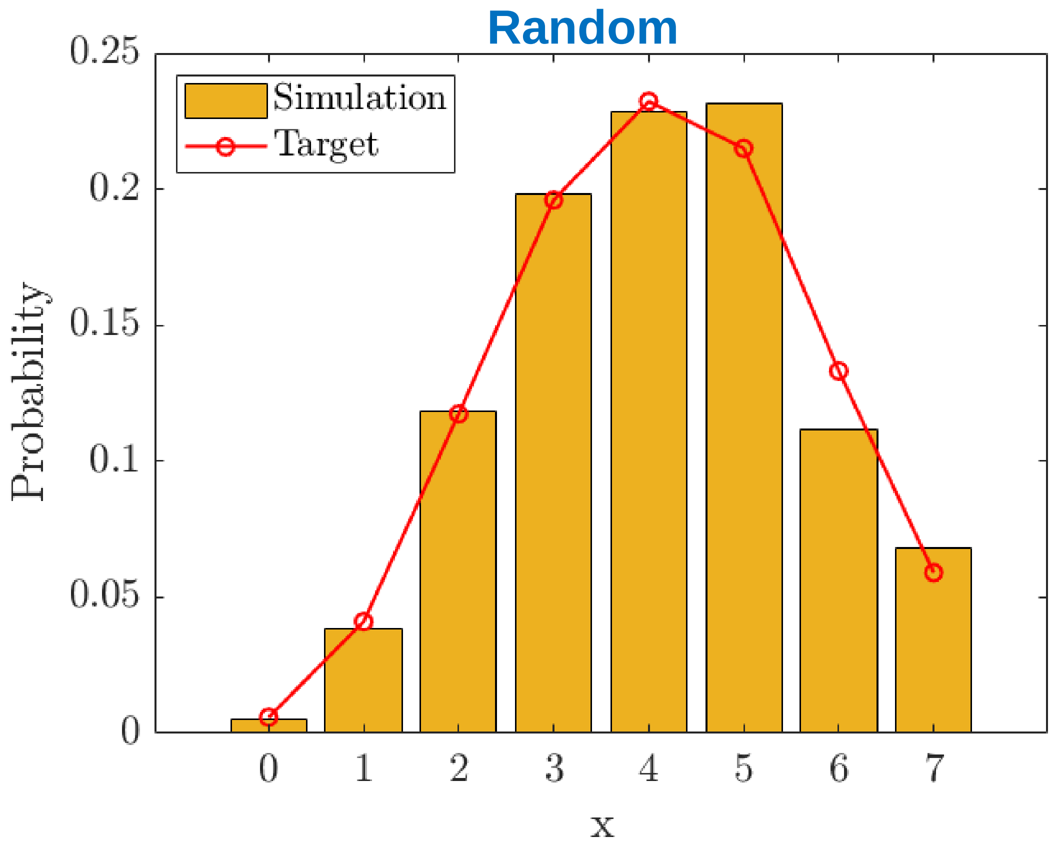
1. **Uniform** : Equiprobable Superposition of $|0\rangle, \ldots, |N-1\rangle$
2. **Normal** : Normally distributed with empirical mean and std of training set
3. **Random** : Randomly distributed over $|0\rangle, \ldots, |N-1\rangle$

Classical Discriminator

- ✓ PyTorch Discriminator
- ✓ 512 nodes + Leaky ReLU → 256 nodes + Leaky ReLU → single-node + sigmoid
- ✓ AMSGRAD optimizer for both generator and discriminator

Appendix B : qGAN in HEP (Results)





Appendix C : Why $n_2 > n$?

$$M(j) = \begin{pmatrix} |I_{0j}|^{\frac{1}{2}} e^{i\phi_{0j}} \\ \vdots \\ |I_{2^n-1j}|^{\frac{1}{2}} e^{i\phi_{2^n-1j}} \end{pmatrix}, \quad \phi_{ij} \in [0, 2\pi[$$ where I_{ij} = Amplitude at pixel i for image j → Normalized

Case $n_2 = n$

- Quantum Circuit consists of reversible gates → **Unitary matrix**
- Inputs = computational basis → $M(j) =$ j$^{\text{th}}$ column at M_{PQC_2}

→ Cannot train PQC2 with n qubits if M(j) do not form an orthonormal basis

Case $n_2 = 2n$

- First 2^n columns of PQC2 is constructed as : $M_{PQC_2}(i) = |i\rangle \otimes |M(i)\rangle$ where $|i\rangle \in \{|0\rangle, \ldots, |2^n - 1\rangle\}$,

→ $\langle M_{PQC_2}(i) | M_{PQC_2}(j)\rangle = \langle i|j\rangle\langle M(i)|M(j)\rangle = \begin{cases} 1 & if\ i = j \\ 0 & otherwise \end{cases}$

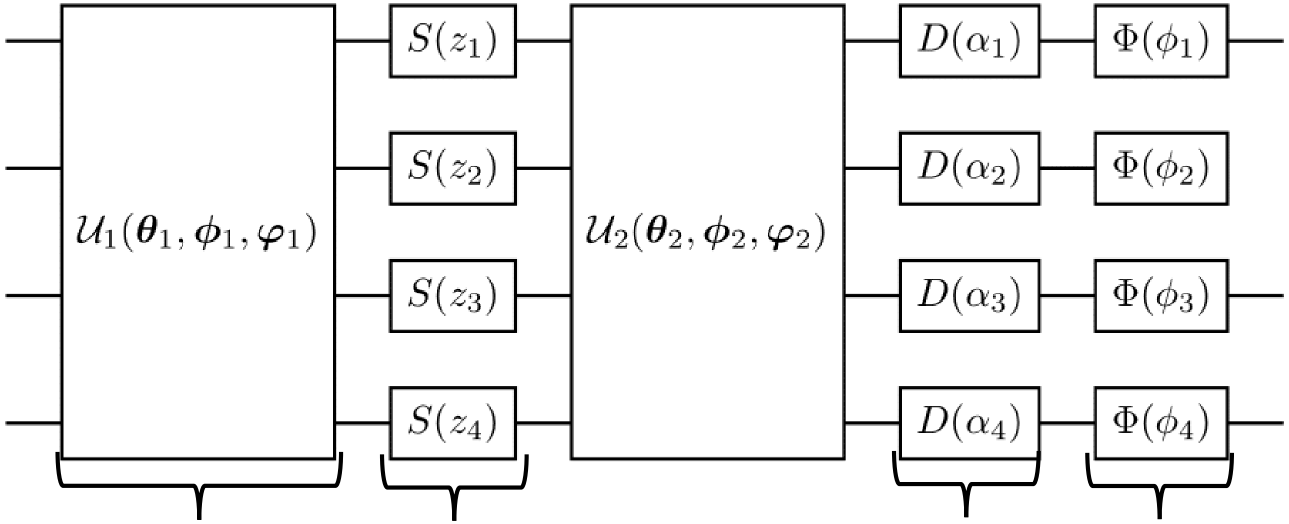
→ $2^{2n} - 2^n$ columns can be chosen freely to construct a unitary matrix

Appendix D : Qubit vs. CV

	CV	Qubit
Fundamental Unit	Qumodes $\{\|x\rangle\}_{x \in R}$, $\|\psi\rangle = \int dx\, \psi(x)\|x\rangle dx$	Qubits $\|0/1\rangle$, $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$
Relevant Operators	Position $\hat{x}$, Momentum $\hat{p}$ Mode operators $\hat{a}, \hat{a}^t$	Pauli Operators $\sigma_x, \sigma_y, \sigma_z$
Common Gates	Displacement $D_i(\alpha) = \exp\left(\alpha \hat{a}_i^t - \alpha^* \hat{a}_i\right)$ Rotation $R_i(\phi) = \exp(i\phi \hat{n}_i)$ Squeezing $S_i(z) = \exp\left(\frac{1}{2}\,(z^* \hat{a}_i^2 - z\hat{a}_i^{t2})\right)$ Beam Splitters $BS_{ij}(\theta, \phi) = \exp\left(\theta(e^{i\phi}\hat{a}_i^t \hat{a}_j - e^{-i\phi}\hat{a}_i \hat{a}_j^t)\right)$ Kerr $K_i(\kappa) = \exp(i\kappa n_i^2)$	Phase Shift, Rotation, Hadamard, Controlled-U gate
Measurements	Homodyne $\|x_\phi\rangle\langle x_\phi\|$, $\hat{x}_\phi = \cos(\phi)\hat{x} + \sin(\phi)\hat{p}$ Heterodyne $\frac{1}{\pi}\|\alpha\rangle\langle\alpha\|$ Photon Counting $\|n\rangle\langle n\|$	Pauli Measurements $\|0/1\rangle\langle 0/1\|, \|\pm\rangle\langle\pm\|, \|\pm i\rangle\langle\pm i\|$

https://doi.org/10.22331/q-2019-03-11-129

Appendix E : CVNN



https://doi.org/10.1038/ncomms13795

- Fully connected layer : $\boldsymbol{x} \rightarrow \phi(W\boldsymbol{x} + \boldsymbol{b})$ W= Weight matrix, $\boldsymbol{b}$ = bias, $\phi(x)$ = Activation function
- Weight matrix W decomposed using **singular value decomposition** : $W = O_2 \Sigma O_1$

1. Multiplication by an orthogonal matrix O_1 → Apply an **interferometer** U_1
2. Multiplication by a diagonal matrix Σ → Apply a **squeezing gate** $S(\mathbf{r})|\mathbf{x}\rangle = e^{-\frac{1}{2}\Sigma_i r_i}|\Sigma\mathbf{x}\rangle$
3. Multiplication by another orthogonal matrix O_2 → Apply an **interferometer** U_2
4. Addition of bias $\boldsymbol{b}$ → Apply a **displacement gate** $D(\boldsymbol{\alpha})|\mathbf{x}\rangle = |\mathbf{x} + \boldsymbol{\alpha}\rangle$
5. Non-linear function $\phi(x)$ → Apply a **Kerr gate** $\Phi|\mathbf{x}\rangle = |\phi(\mathbf{x})\rangle$

$L|\boldsymbol{x}\rangle \propto |\phi(W\boldsymbol{x} + \boldsymbol{b})\rangle$