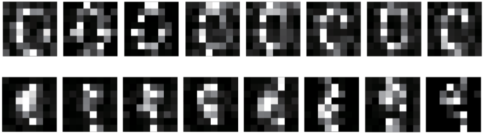
Experimental Quantum GANs

CPEN 400Q class presentation

Yuyou Lai, Juntong Luo, Sam Schweigel, Bolong Tan

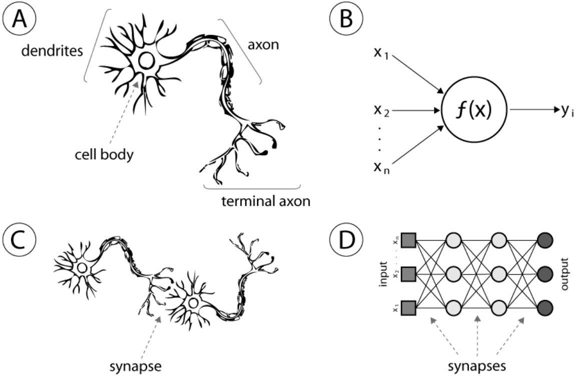


Overview

- “Experimental Quantum Generative Adversarial Networks for Image Generation”[1]
- Implemented quantum GANs on a real quantum device.
- The authors train and use a patch GAN on a superconducting quantum processor to generate the images you saw on the title slide.
- Two training strategies:
  - Batch: Training the GAN on batches of images
  - Patch: Dividing the image into smaller patches and training the GAN on each patch individually

Neural Networks

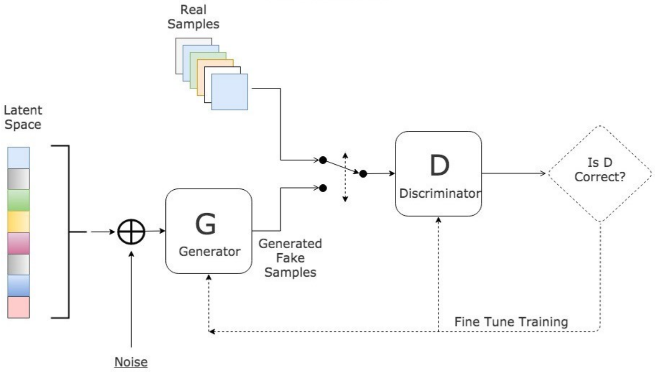
- Neural networks are a type of machine-learning model inspired by the structure of the human brain.



- Neural networks can learn to recognize patterns and make predictions based on input data, and can be used for image and speech recognition, natural language processing, etc.

Generative Adversarial Network (GAN)

- GAN is a specific type of neural network that are used for generating new data that is similar to a given dataset.
- GAN consists of two neural networks: a generator and a discriminator.



Partial trace

In a mixed state ρ, the expected value looks like:

$$\langle A \rangle_\psi = \mathrm{Tr}(A\rho) = \sum_i \langle i | A\rho | i \rangle$$

If a measurement traces over a complete set of basis states for the space $\mathcal{H}$, what if we want to discard the subsystem $\mathcal{A}$ of $\mathcal{H}_\mathcal{A} \otimes \mathcal{H}_\mathcal{B}$?

$$\mathrm{Tr}_\mathcal{A}(\rho) = \sum_i (\langle i|_\mathcal{A} \otimes I_\mathcal{B})\ \rho\ (|i\rangle_\mathcal{A} \otimes I_\mathcal{B})$$

By *tracing out* the system $\mathcal{A}$, we project onto a basis for $\mathcal{A}$ while leaving $\mathcal{B}$ untouched ($I_\mathcal{B}$).
Each term of the sum leaves us with an *operator* instead of a scalar. This is the **partial trace**[2].

Partial measurement

Let's try the partial trace:

$$\mathrm{Tr}_{\mathcal{A}}((\Pi_i \otimes I_{\mathcal{B}})\rho) = \sum_j (\langle j| \otimes I)(\Pi_i \otimes I)\rho(|j\rangle \otimes I)$$

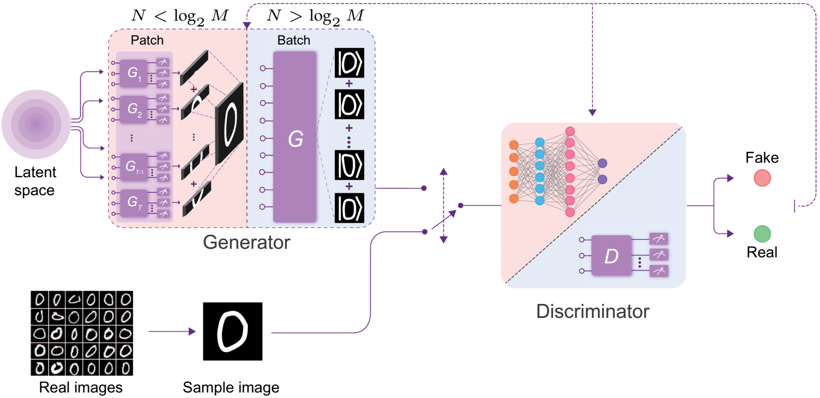
Notice that $\langle j|\,\Pi_j$ is 0 if $i \neq j$, and $\langle j|$ otherwise!

$$\rho' = \frac{\mathrm{Tr}_{\mathcal{A}}((\Pi_i \otimes I_{\mathcal{B}})\rho)}{\underbrace{\mathrm{Tr}((\Pi_i \otimes I_{\mathcal{B}})\rho)}_{\text{magic}}}$$

Making a partial measurement on our system transforms the density matrix *non-linearly*!

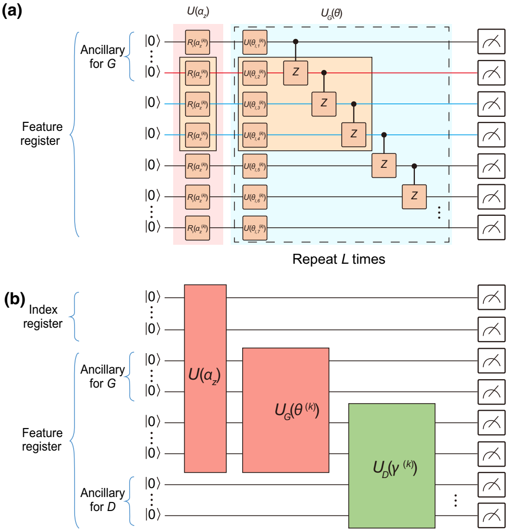
Quantum GANs - Overview

N = # of qubits, M = # of features



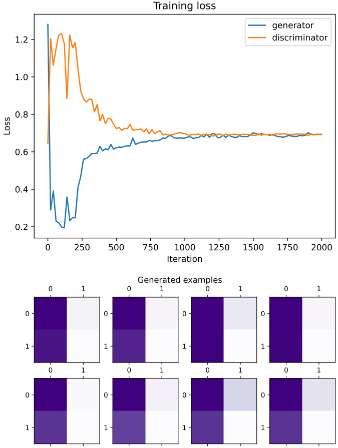
Quantum GANs - Details

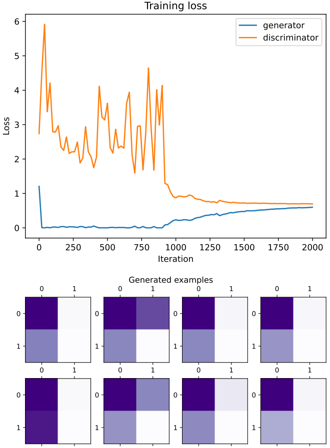
Top: Patch GAN Sub-Generator, Bottom: Batch GAN Generator and Discriminator



Reproducibility

Identical parameters—different seeds. Performance from the paper is “best-case” only.

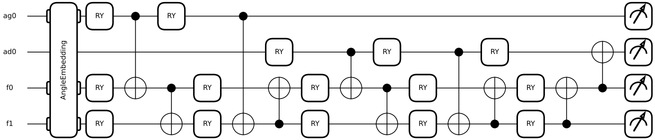




Software design and implementation

- JAX throughout: just take gradients of the loss methods.
- Simple, but extensible API implementing completely quantum GANs (generator and discriminator):

```
g = gan.BatchGAN(4, 1, 4, 1, 5,
                 entanglers=qml.CNOT, layout="random")
g.draw(...)
```



- Deterministic randomness for repeatable experiments!

References I

[1] He-Liang Huang et al. “Experimental Quantum Generative Adversarial Networks for Image Generation”. In: *Physical Review Applied* 16.2 (Aug. 2021). DOI: 10.1103/physrevapplied.16.024051. URL: https://doi.org/10.1103%2Fphysrevapplied.16.024051.

[2] Ryan LaRose. “Quantum States and Partial Trace”. In: *QuIC Seminar*. Semester I. 2018. URL: https://www.ryanlarose.com/uploads/1/1/5/8/115879647/quic06-states-trace.pdf.