Quantum generative modelling and Born machines

Train a circuit-defined probability distribution while comparing it with matched classical samplers.

Undergraduate 21 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- Fit a one-qubit Bernoulli distribution.
- Explain why likelihood access differs from sampling.
- Distinguish entanglement from generative advantage.

Let measurement define a generative distribution

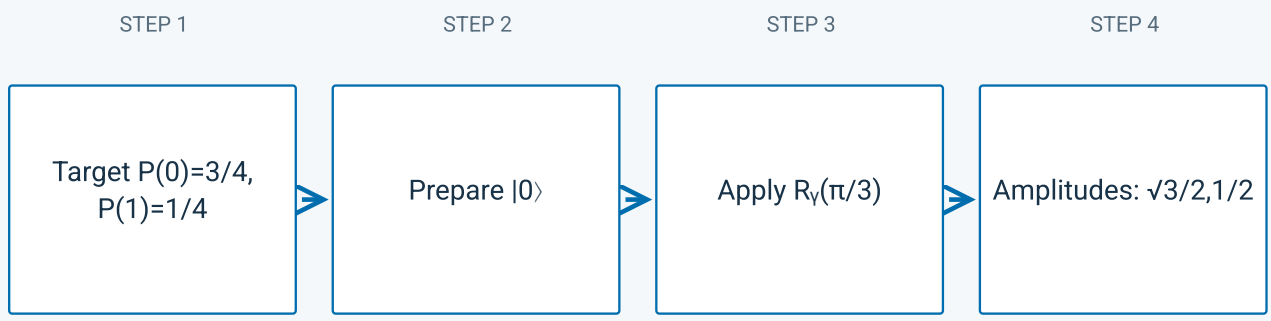
A generative model produces samples resembling a target dataset or distribution. In a quantum circuit Born machine, a parameterised circuit prepares a state and computational-basis measurement produces bitstrings. Their probabilities are the squared magnitudes of the state amplitudes. The circuit is therefore a model of a probability distribution, even when the amplitudes themselves are complex.

The output is a sample, not necessarily a tractable probability value for every bitstring. This makes the model similar to other implicit generative models, where sampling can be easier than evaluating likelihoods. Training must choose a loss that can be estimated from the information the device actually provides. A simulator's ability to print all probabilities should not be assumed on hardware.

Review Born probabilities, entanglement and sampling evidence. A generative task should specify both the target distribution and how sample quality is evaluated.

WORKED DIAGRAM · AA22

A rotation fits a simple target



The one probability is $\sin^2(\pi/6)=1/4$. A separate Bell preparation samples 00 and 11 equally, but a copied classical coin produces that same easy distribution.

State the model setup

Choose an input reference state, circuit family, parameter set and measurement basis. The model distribution is the probability of each measured bitstring. Supply training data or a target distribution, a loss function, an optimisation method and a shot budget. The output is a trained sampler with a stated evaluation protocol.

If the target has continuous variables, explain how they are discretised into bitstrings and how samples are mapped back. If some bitstrings represent invalid objects, define whether the circuit prevents them, penalises them or filters them after measurement. Rejection changes the effective sample cost and should be included in performance comparisons.

Work a one-qubit fit

Suppose the target produces zero with probability three quarters and one with probability one quarter. A Y rotation by theta on zero gives one probability equal to sine squared of half theta. Choosing theta equal to pi over three fits the target exactly. The amplitudes are square root of three over two and one half, so the probabilities are three quarters and one quarter.

This simple fit makes the parameter-to-distribution connection explicit. Many states can produce the same Z distribution because phases are not observed in that setting. A phase gate after the final rotation would not change this sampler's probabilities. It may still matter if later gates mix components before measurement, so the circuit architecture determines which parameters are identifiable from the chosen loss.

Now consider a two-bit target that produces 00 and 11 equally and never produces 01 or 10. H on q0 followed by CX prepares a Bell state with exactly that Z distribution. But a classical fair coin copied into two output bits generates the same samples. Preparing an entangled model does not establish a sampling advantage for this easy target.

Choose the rotation from the target measurement probability.

$$|\psi(\theta)\rangle = R_y(\theta)|0\rangle, \qquad P_\theta(1) = \sin^2(\theta/2), \qquad \theta = \frac{\pi}{3}: \quad |\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle.$$

Squaring the two magnitudes gives target probabilities $3/4$ and $1/4$. The Z sampler does not identify a final relative phase unless later mixing makes that phase affect probabilities.

Training without full likelihoods

One option compares moments or selected features of generated and training samples. Another uses a kernel discrepancy such as maximum mean discrepancy, which compares averages of pairwise kernel values. The chosen kernel determines which distribution differences are visible at a given sample budget. A small empirical discrepancy is meaningful only with a suitable evaluation and uncertainty analysis.

Liu and Wang developed differentiable training of quantum circuit Born machines using a kernel-based discrepancy and studied small generative examples. Their work provides a concrete model and training proposal, not proof that every generative task gains a quantum advantage. Read the Born-machine training paper.

When exact probabilities are unavailable, estimating a log-likelihood for rare strings can be difficult. Repeated sampling may never observe them. A training method should not quietly use statevector probabilities in simulation while claiming the same cost on hardware. State which loss terms are measured, classically computed or available only in small validation experiments.

Follow a complete learning workflow

Define the data representation and split training, validation and test sets. Choose a circuit with enough structure to represent useful correlations but a manageable measurement and optimisation cost. Select a sample-based loss, estimate gradients or use another optimiser, and monitor both fit and uncertainty. Evaluate held-out samples using metrics matched to the application.

For the two-bit target, inspect all four probabilities in a tiny simulator to validate the circuit, then repeat with finite-shot samples to understand the statistical task. As the number of bits grows, full distribution enumeration becomes infeasible. The evaluation should therefore identify which properties can be estimated without reconstructing every outcome probability.

Expressivity is not trainability

A circuit may represent complicated distributions while being difficult to train. Deep random circuits can have small gradient signals in relevant regimes, and hardware noise can wash out useful correlations. A shallow structured circuit may train more reliably but represent a narrower family. The appropriate trade-off depends on the data and loss.

Classical generative models can also exploit structure. Compare against suitable autoregressive, graphical, tensor-network or other classical samplers, rather than only independent-bit baselines. If a claimed hard-to-sample circuit distribution is used, distinguish theoretical hardness assumptions from evidence that a trained model solves a useful data task better.

Account for sample quality and cost

Report the number of circuit executions, optimiser evaluations, training seeds and postselection rate. Include calibration and data preprocessing where relevant. A sampler that emits many invalid strings may need many executions per useful sample. A visually plausible small sample collection can conceal mode collapse or missing rare events, so use quantitative tests and uncertainty.

Generalisation remains a statistical question. Reproducing a training set does not necessarily model the underlying distribution. Repeatedly selecting the best seed on the test set biases the evaluation. Quantum measurement randomness does not protect against those standard experimental errors.

Practice with answers

Exercise 1. What Y-rotation angle gives equal zero and one probabilities from zero? What angle gives one with certainty?

> Check the worked answer
>
> Pi over two gives equal probabilities, and pi gives one with certainty. These follow from sine squared of half the rotation angle. Phase conventions do not alter these Z probabilities.

Exercise 2. Does matching the 00/11 target with a Bell circuit demonstrate quantum generative advantage?

Check the worked answer

No. A classical coin copied into two bits produces the same distribution cheaply. The quantum circuit demonstrates representational correctness for a toy example. Advantage requires a stronger matched task, baseline and resource analysis.

Check your understanding

PAUSE & THINK

Does a Bell circuit matching the equal 00/11 target prove generative advantage?

- [ ] Yes, because it is entangled
- [ ] No; a classical copied coin matches it cheaply
- [ ] Only after many shots

Answer: No; a classical copied coin matches it cheaply. The target distribution has a simple classical sampler. The circuit validates a toy encoding, while advantage requires a stronger matched task and resource comparison.

Sources

- Read the Born-machine training paper