

# Confusion Matrix in Quantum Readout Mitigation

## Overview

A **confusion matrix** is a 4×4 table (for two qubits) that captures how often a quantum computer *misreports* measurement results. It quantifies bias and readout errors, allowing us to mathematically correct raw experimental data.

---

## Why We Need It

Real quantum hardware rarely returns perfect measurement results. Even if you prepare a clean computational basis state like  $|00\rangle$ , the device might occasionally report  $01$ ,  $10$ , or  $11$  because of:

- measurement crosstalk
- thermal excitations
- amplifier bias
- classical bit-flip errors
- imperfect discrimination of readout signals

The confusion matrix tells us *how often* each type of misclassification happens.

---

## How It's Built

There are four 2-qubit basis states:

$|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$ ,  $|11\rangle$

You run a calibration experiment:

1. Prepare each basis state exactly.
2. Measure it many times.
3. Record how often the hardware reports each possible outcome.

That gives you a probability table like this (example only):

Prepared → / Measured ↓	00	01	10	11
00	0.963	0.021	0.010	0.006
01	0.026	0.942	0.010	0.022
10	...	...	...	...

Prepared → / Measured ↓	00	01	10	11
11	...	...	...	...

Each row sums to 1.0 and represents:

$$M_{i,j} = P(\text{measured } j \mid \text{prepared } i)$$

This matrix tells you exactly how measurement errors distort ideal results.

## How It's Used

Suppose your Bell experiment yields a measured probability vector:

$$\mathbf{p}_{\text{measured}} = [P(00), P(01), P(10), P(11)]$$

Readout noise means:

$$\mathbf{p}_{\text{measured}} = \mathbf{M} \cdot \mathbf{p}_{\text{true}}$$

So we invert the confusion matrix:

$$\mathbf{p}_{\text{true}} \approx \mathbf{M}^{-1} \cdot \mathbf{p}_{\text{measured}}$$

This undoes the misclassification and recovers an estimate of the *actual* quantum-state populations.

## Why It Works

Readout errors are mostly **classical** and **stable** over short time windows. That means if Torino misreads 01 as 00 in your calibration runs about 2% of the time, it will misread it the same way in your Bell-state run. So the calibration matrix acts like a fingerprint of the measurement noise.

Correcting with the inverse matrix recovers fidelity that the hardware lost *only during measurement*, not during the actual quantum evolution.

## Practical Results

You observed this effect directly:

- Raw  $\langle Z \otimes Z \rangle$ : **0.951**
- Mitigated  $\langle Z \otimes Z \rangle$ : **0.970**

The improvement comes from removing readout bias, not changing the underlying quantum physics.

---

## Summary

A confusion matrix is:

- A calibrated description of how the device misreports measurement outcomes.
- A 4×4 stochastic matrix for two-qubit experiments.
- A tool to reverse classical noise in measurement.
- A crucial part of near-term quantum error mitigation.

By preparing known states, measuring the device's mistakes, and mathematically undoing them, you reveal the “truer” behavior of the quantum system you prepared.

## Do Real Quantum Jobs Automatically Run Mitigation?

Most quantum jobs **do not** perform error mitigation automatically. Mitigation requires running many calibration circuits, which becomes expensive as qubit count grows. Instead:

- Researchers typically **calibrate once per session**, build a confusion matrix, and reuse it.
- IBM's default execution **does not** include automatic mitigation unless explicitly requested.
- Future quantum computers may include **always-on mitigation**, but today it's manual and user-driven.

Your current workflow—baseline run → calibration → matrix inversion → corrected results—is exactly how mitigation is done in modern quantum research.