

```
In [11]: import helpers
         from helpers import *
```

## W5 O(log n) simulation

```
In [10]: qreg = QuantumRegister(5)
         w5 = QuantumCircuit(qreg)
         w5.x(qreg[1])
         # 1->2
         Bdirect(w5, qreg, 2, 1, 2/5)
         # 1->0
         # 2->3
         Bdirect(w5, qreg, 0, 1, 0.5)
         Bdirect(w5, qreg, 3, 2, 2/3)
         # 2->4
         Bdirect(w5, qreg, 4, 2, 0.5)
         job = qiskit.execute(w5, Aer.get_backend('statevector_simulator'))
         theoretical_psi = job.result().get_statevector(w5)
```

```
In [5]: tomo_circuits = state_tomography_circuits(w5, qreg)
         job = qiskit.execute(tomo_circuits,
                               simulator,
                               noise_model=noise_model_melbourne,
                               coupling_map=coupling_map_melbourne,
                               basis_gates=basis_gates_melbourne)
         job_monitor(job, monitor_async = True)

         calib_circuit, state_labels = mc.complete_meas_cal(qr=qreg)
         job_cal = qiskit.execute(calib_circuit,
                                   simulator,
                                   noise_model=noise_model_melbourne,
                                   coupling_map=coupling_map_melbourne,
                                   basis_gates=basis_gates_melbourne)
         job_monitor(job_cal, monitor_async=True)
```

```
In [6]: job_results = job.result()
         cal_results = job_cal.result()
         meas_fitter = mc.CompleteMeasFitter(cal_results, state_labels)
         correct_results = meas_fitter.filter.apply(job_results)
         fitter = StateTomographyFitter(correct_results, tomo_circuits)
         fitted_rho = fitter.fit()
         fidelity = state_fidelity(theoretical_psi, fitted_rho)
         print(fidelity)
```

0.848677414406781

previous result: 0.84