Robust Fitting on a Gate Quantum Computer: Supplementary Materials

Frances Fengyi Yang[1], Michele Sasdelli[1], and Tat-Jun Chin[1]

The University of Adelaide, Adelaide SA 5000, Australia
{fengyi.yang,michele.sasdelli,tat-jun.chin}@adelaide.edu.au

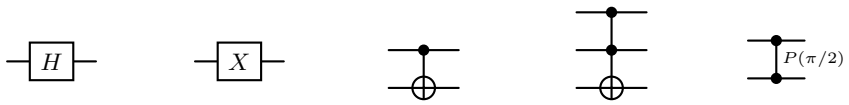
1 Overview

The supplementary materials provide implementation details on experiments mentioned in Sec. 6.1 in the main paper. Below is a brief overview of what is included in the document:

- Sec. 2 outlines the specifics of the quantum circuits' design, including optimisation tricks.
- Sec. 3 illustrates the way to execute the quantum circuits on both a quantum simulator and an actual quantum computer.

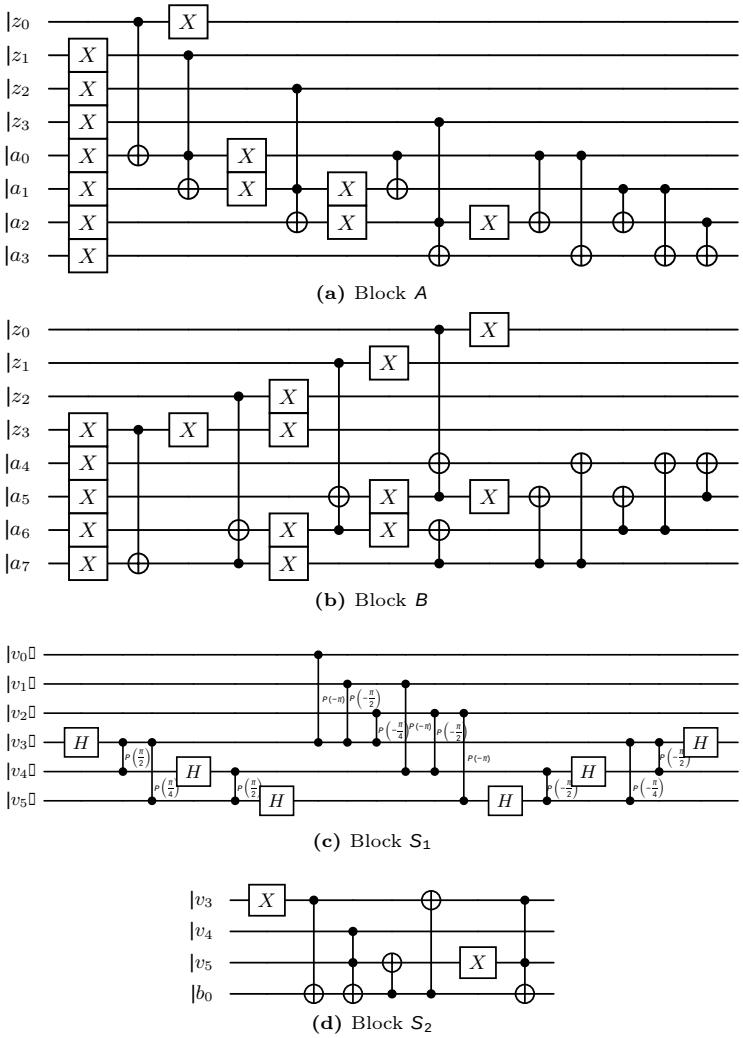
2 Circuit design details

This section elaborates on the circuit design details and optimisation strategies not disclosed in the main paper, further extending the Circuit Design discussion in Section 4.2 in the main paper. Please consult Fig. 1 for representations of the quantum gates employed within our circuit design. The majority of gates utilized are detailed in [1, Chap. 5.2]. All quantum circuits created for the experiments detailed in Section 6.1 adhere to the framework presented in Fig. 4 of the main paper. Therefore, the following subsections will exclusively focus on the implementation of Block D, omitting the repetitive components from further illustration. The following subsections will illustrate 3 circuit instances. Sec. 2.1 corresponds to the largest scale case discussed in Sec. 6.1 in the main paper, followed by Sec. 2.2 and 2.3, where smaller instances and case-by-case optimisation are introduced.



(a) Hadamard gate **(b)** Pauli-X gate **(c)** CNOT gate **(d)** CCNOT gate **(e)** CPhase gate

Fig. 1: Quantum gates used in our circuits.



(a) Block A

(b) Block B

(c) Block S_1

(d) Block S_2

Fig. 2: Each block of Instance 1 is shown separately.

2.1 Circuit Instance 1

This section corresponds to the largest experiment run in Sec. 6.1, where input data $D = \{7, 5, 3, 2\}$ at $C = 3$ bit precision and $2\epsilon = 2$. Due to the large scale of this particular circuit instance, the implementation of each block in Fig. 5 of the main paper is shown separately in Fig. 2. Please note in this case Block V_1 and V_2 were illustrated in Fig. 6 of the main paper. Instead of using QFT in S_2, a sequence of controlled-not gates can be used to minimise the circuit depth, at the expense of an additional auxiliary qubit $|b_0\rangle$ to carry the result of the feasibility test.

2.2 Circuit Instance 2

This section corresponds to the case where input data $D = \{1, 0\}$ at $C = 1$ bit precision and $2\epsilon = 0$. See Fig. 3 for the implementation of Block D. In this case, the QFT part in Block S_1, and the whole block S_2 can be saved, as for S_1, the difference between the selected data can be computed from a CNOT gate, while for S_2, the resultant $|v_0\rangle$ from $S1$ can be directly passed as the feasibility test outcome.

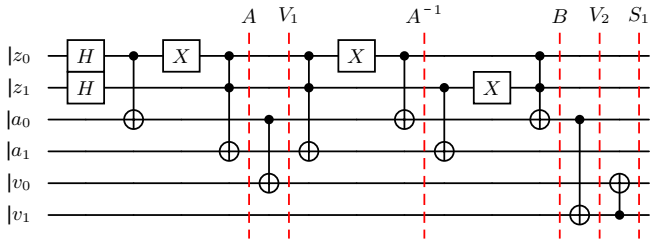


Fig. 3: Instance 2

2.3 Circuit Instance 3

This section corresponds to the case where input data $D = \{2, 0\}$ at $C = 2$ bit precision and $2\epsilon = 1$. See Fig. 4 for the implementation of Block D. A simple Pauli-X gate is used in S_2 instead of QFT due to the specific computational scenario.

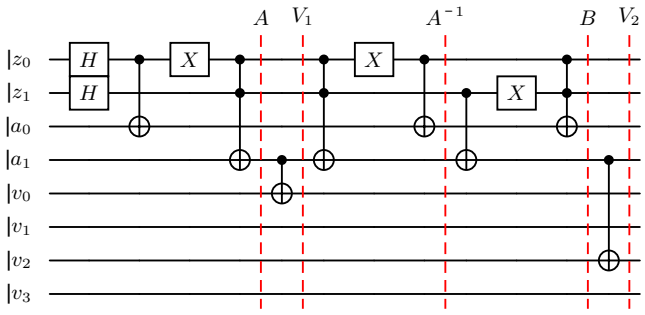
2.4 Input & output explained

In accordance with Fig. 4 and as detailed in Section 4.2 of the main paper, the states $|z\rangle$, $|a\rangle$, and $|v\rangle$ are consistently initialized to $|0\rangle$. The data input occurs through Blocks V_1 and V_2, as delineated in Fig. 6, where the states of $|v\rangle$ are selectively activated via CNOT gates. The focal point of our subsequent analysis is the output associated with $|z\rangle$. Upon completion, both $|a\rangle$ and $|v\rangle$ return to $|0\rangle$, while $|y\rangle$ ends up to be $|1\rangle$.

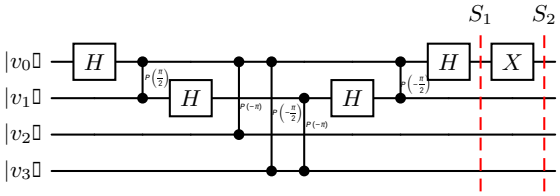
Considering that $|z\rangle$ achieves a uniform superposition subsequent to the application of the initial Hadamard gate, the resultant state of $|v\rangle$ encapsulates the influence sampled across all conceivable subsets of the input dataset. This naturally transits to the way we adopted for result analysis in Sec. 6.1.

3 Running quantum circuits

Experiments on our quantum circuits were conducted on both a quantum simulator (SV1) and a real quantum computer (IonQ Aria) via `qiskit_braket_`



(a) Block A to V_2 of Instance 3.



(b) Block S_1 and S_2 of Instance 3.

Fig. 4: Instance 3 split into 2 subplots.

`provider` on Amazon Braket. `AWSBraketProvider` facilitates selection among various backends, encompassing both simulators and actual quantum computers. Please refer to code provided for details.

4 Code

Please find our detailed implementations in `demo.ipynb`.

References

1. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (2010)