X Quantum: evaluation of explanations from LLM generated prompts

Give a score to each explanation

* Required

Amplitude Estimation

Algorithm Code

```
OPENQASM 2.0;
include "qelib1.inc";
qreg eval[4];
qreg q[1];
creg meas[5];
u2(0,-pi) eval[0];
u2(0,-pi) eval[1];
u2(0,-pi) eval[2];
u2(0,-pi) eval[3];
u3(0.9272952180016122,0,0) q[0];
cx eval[0],q[0];
u(-0.9272952180016122,0,0) q[0];
cx eval[0],q[0];
u3(0.9272952180016122,0,0) q[0];
cx eval[1],q[0];
u(-1.8545904360032244,0,0) q[0];
cx eval[1],q[0];
u3(1.8545904360032244,0,0) q[0];
cx eval[2],q[0];
u(-3.7091808720064487,0,0) q[0];
cx eval[2],q[0];
u3(2.574004435173138,-pi,-pi) q[0];
cx eval[3],q[0];
u(-7.4183617440128975,0,0) q[0];
cx eval[3],q[0];
h eval[3];
cp(-pi/2) eval[2],eval[3];
cp(-pi/4) eval[1],eval[3];
cp(-pi/8) eval[0],eval[3];
h eval[2];
cp(-pi/2) eval[1],eval[2];
cp(-pi/4) eval[0],eval[2];
h eval[1];
cp(-pi/2) eval[0],eval[1];
h eval[0];
u(7.4183617440128975,0,0) q[0];
barrier eval[0],eval[1],eval[2],eval[3],q[0];
measure eval[0] -> meas[0];
measure eval[1] -> meas[1];
measure eval[2] -> meas[2];
measure eval[3] -> meas[3];
measure q[0] -> meas[4];
```

Ground Truth Description

AE aims to find an estimation for the amplitude of a certain quantum state.

1. A *

   Link to explanation: *https://drive.google.com/file/d/10sUORYc0J5G115ACCvA7GNj9Cnxpm3Um/view?usp=drive_link*

1	2	3	4	5

2. B *

   Link to explanation: *https://drive.google.com/file/d/10o42gyB4nGUB0AGXja91cwSJBMrxI0ZF/view?usp=drive_link*

1	2	3	4	5

Deutsch-Jozsa

Algorithm Code

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[10];
creg c[9];
u2(0,0) q[0];
u2(0,0) q[1];
h q[2];
u2(0,0) q[3];
h q[4];
u2(0,0) q[5];
u2(0,0) q[6];
h q[7];
u2(0,0) q[8];
u2(-pi,-pi) q[9];
cx q[0],q[9];
u2(-pi,-pi) q[0];
cx q[1],q[9];
u2(-pi,-pi) q[1];
cx q[2],q[9];
h q[2];
cx q[3],q[9];
u2(-pi,-pi) q[3];
cx q[4],q[9];
h q[4];
cx q[5],q[9];
u2(-pi,-pi) q[5];
cx q[6],q[9];
u2(-pi,-pi) q[6];
cx q[7],q[9];
h q[7];
cx q[8],q[9];
u2(-pi,-pi) q[8];
barrier q[0],q[1],q[2],q[3],q[4],q[5],q[6],q[7],q[8],q[9];
measure q[0] -> c[0];
measure q[1] -> c[1];
measure q[2] -> c[2];
measure q[3] -> c[3];
measure q[4] -> c[4];
measure q[5] -> c[5];
measure q[6] -> c[6];
measure q[7] -> c[7];
measure q[8] -> c[8];
```

Ground Truth Description

This algorithms determines, whether an unknown oracle mapping input values either to 0 or 1 is constant (always output 1 or always 0) or balanced (both outputs are equally likely).

3. A *

Link to explanation: *https://drive.google.com/file/d/10n1LHAffANUL6QphrLCgUysHPRezz5ai/view?usp=drive_link*

1	2	3	4	5

4. B *

Link to explanation: *https://drive.google.com/file/d/10mxNlvN2x8z2fiqiBJ27I04ZWP95ztdA/view?usp=drive_link*

1	2	3	4	5

Grover

Algorithm Code

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[2];
qreg flag[1];
creg meas[3];
h q[0];
h q[1];
x flag[0];
cp(pi/2) q[1],flag[0];
cx q[1],q[0];
cp(-pi/2) q[0],flag[0];
cx q[1],q[0];
cp(pi/2) q[0],flag[0];
u2(0,0) q[0];
u1(-pi) q[1];
cx q[0],q[1];
u2(-pi,-pi) q[0];
u1(-pi) q[1];
barrier q[0],q[1],flag[0];
measure q[0] -> meas[0];
measure q[1] -> meas[1];
measure flag[0] -> meas[2];
```

Ground Truth Description

One of the most famous quantum algorithm known so far, Grover's algorithm finds a certain goal quantum state determined by an oracle. In our case, the oracle is implemented by a multi-controlled Toffoli gate over all input qubits. In this no ancilla version, no ancilla qubits are used during its realization.

5. A *

Link to explanation: *https://drive.google.com/file/d/10mkGLYIsSAgWs4bWSGy5TBSE0M4tHQor/view?usp=drive_link*

1	2	3	4	5

6. B *

Link to explanation: *https://drive.google.com/file/d/10mOisJdB5ew_qesatxoh46WuILG9x3zl/view?usp=drive_link*

1	2	3	4	5

Quantum Fourier Transform

Algorithm Code

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[10];
creg c[10];
creg meas[10];
h q[9];
cp(pi/2) q[9],q[8];
h q[8];
cp(pi/4) q[9],q[7];
cp(pi/2) q[8],q[7];
h q[7];
cp(pi/8) q[9],q[6];
cp(pi/4) q[8],q[6];
cp(pi/2) q[7],q[6];
h q[6];
cp(pi/16) q[9],q[5];
cp(pi/8) q[8],q[5];
cp(pi/4) q[7],q[5];
cp(pi/2) q[6],q[5];
h q[5];
cp(pi/32) q[9],q[4];
cp(pi/16) q[8],q[4];
cp(pi/8) q[7],q[4];
cp(pi/4) q[6],q[4];
cp(pi/2) q[5],q[4];
h q[4];
cp(pi/64) q[9],q[3];
cp(pi/32) q[8],q[3];
cp(pi/16) q[7],q[3];
cp(pi/8) q[6],q[3];
cp(pi/4) q[5],q[3];
cp(pi/2) q[4],q[3];
h q[3];
cp(pi/128) q[9],q[2];
cp(pi/64) q[8],q[2];
cp(pi/32) q[7],q[2];
cp(pi/16) q[6],q[2];
cp(pi/8) q[5],q[2];
cp(pi/4) q[4],q[2];
cp(pi/2) q[3],q[2];
h q[2];
cp(pi/256) q[9],q[1];
cp(pi/128) q[8],q[1];
cp(pi/64) q[7],q[1];
cp(pi/32) q[6],q[1];
cp(pi/16) q[5],q[1];
cp(pi/8) q[4],q[1];
cp(pi/4) q[3],q[1];
cp(pi/2) q[2],q[1];
h q[1];
cp(pi/512) q[9],q[0];
cp(pi/256) q[8],q[0];
cp(pi/128) q[7],q[0];
cp(pi/64) q[6],q[0];
cp(pi/32) q[5],q[0];
cp(pi/16) q[4],q[0];
cp(pi/8) q[3],q[0];
cp(pi/4) q[2],q[0];
cp(pi/2) q[1],q[0];
h q[0];
swap q[0],q[9];
swap q[1],q[8];
swap q[2],q[7];
swap q[3],q[6];
swap q[4],q[5];
barrier q[0],q[1],q[2],q[3],q[4],q[5],q[6],q[7],q[8],q[9];
measure q[0] -> meas[0];
measure q[1] -> meas[1];
measure q[2] -> meas[2];
measure q[3] -> meas[3];
measure q[4] -> meas[4];
```

71. measure q[4] -> meas[4];
72. measure q[5] -> meas[5];
73. measure q[6] -> meas[6];
74. measure q[7] -> meas[7];
75. measure q[8] -> meas[8];
76. measure q[9] -> meas[9];

Ground Truth Description

QFT embodies the quantum equivalent of the discrete Fourier transform and is a very important building block in many quantum algorithms.

7. A *

Link to explanation: *https://drive.google.com/file/d/111qnwFBwT-0wDmOebO01Mj6GLJPRJahC/view?usp=drive_link*

1	2	3	4	5

8. B *

Link to explanation: *https://drive.google.com/file/d/11BN3BKtgrRNYyDB_MMywTV7F7eHg01p9/view?usp=drive_link*

1	2	3	4	5

Quantum Fourier Transform with entanglement

Algorithm Code

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[5];
creg meas[5];
h q[4];
cx q[4],q[3];
cx q[3],q[2];
cx q[2],q[1];
cx q[1],q[0];
h q[4];
cp(pi/2) q[4],q[3];
h q[3];
cp(pi/4) q[4],q[2];
cp(pi/2) q[3],q[2];
h q[2];
cp(pi/8) q[4],q[1];
cp(pi/4) q[3],q[1];
cp(pi/2) q[2],q[1];
h q[1];
cp(pi/16) q[4],q[0];
cp(pi/8) q[3],q[0];
cp(pi/4) q[2],q[0];
cp(pi/2) q[1],q[0];
h q[0];
swap q[0],q[4];
swap q[1],q[3];
barrier q[0],q[1],q[2],q[3],q[4];
measure q[0] -> meas[0];
measure q[1] -> meas[1];
measure q[2] -> meas[2];
measure q[3] -> meas[3];
measure q[4] -> meas[4];
```

Ground Truth Description

This algorithms applies regular QFT to entangled qubits.

9. A *

Link to explanation: https://drive.google.com/file/d/1157ebMlkhzVQF07CmIZ0OzUJkwEyRA8t/view?usp=drive_link

1	2	3	4	5

10. B *

Link to explanation: https://drive.google.com/file/d/114kARQAJsCK6XBKp1FAlIo56M80TM8eb/view?usp=drive_link

1	2	3	4	5

Quantum Phase Estimation

Algorithm Code

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[4];
qreg psi[1];
creg c[4];
h q[0];
h q[1];
h q[2];
h q[3];
x psi[0];
cp(-7*pi/8) psi[0],q[0];
cp(pi/4) psi[0],q[1];
cp(pi/2) psi[0],q[2];
swap q[1],q[2];
cp(pi) psi[0],q[3];
swap q[0],q[3];
h q[0];
cp(-pi/2) q[1],q[0];
h q[1];
cp(-pi/4) q[2],q[0];
cp(-pi/2) q[2],q[1];
h q[2];
cp(-pi/8) q[3],q[0];
cp(-pi/4) q[3],q[1];
cp(-pi/2) q[3],q[2];
h q[3];
barrier q[0],q[1],q[2],q[3],psi[0];
measure q[0] -> c[0];
measure q[1] -> c[1];
measure q[2] -> c[2];
measure q[3] -> c[3];
```

Ground Truth Description

QPE estimates the phase of a quantum operation and is a very important building block in many quantum algorithms. In the exact case, the applied phase is exactly representable by the number of qubits.

11. A *

Link to explanation: *https://drive.google.com/file/d/10x7-9ARPYAPV-aLQvEcUIE6JMS7aAY-H/view?usp=drive_link*

1	2	3	4	5

12. B *

Link to explanation: *https://drive.google.com/file/d/10wI0IuaS3OcmfSMgtVK_UXtIfF64Y-4-/view?usp=drive_link*

1	2	3	4	5

Quantum walk

Algorithm Code

```
OPENQASM 2.0;
include "qelib1.inc";
qreg node[2];
qreg coin[1];
creg meas[3];
h coin[0];
ccx coin[0],node[1],node[0];
cx coin[0],node[1];
x node[1];
x coin[0];
ccx coin[0],node[1],node[0];
cx coin[0],node[1];
x node[1];
u2(-pi,-pi) coin[0];
ccx coin[0],node[1],node[0];
cx coin[0],node[1];
x node[1];
x coin[0];
ccx coin[0],node[1],node[0];
cx coin[0],node[1];
x node[1];
u2(-pi,-pi) coin[0];
ccx coin[0],node[1],node[0];
cx coin[0],node[1];
x node[1];
x coin[0];
ccx coin[0],node[1],node[0];
cx coin[0],node[1];
x node[1];
x coin[0];
barrier node[0],node[1],coin[0];
measure node[0] -> meas[0];
measure node[1] -> meas[1];
measure coin[0] -> meas[2];
```

Ground Truth Description

Quantum walks are the quantum equivalent to classical random walks. In this no ancilla version, no ancilla qubits are used during its realization.

13. A *

Link to explanation: https://drive.google.com/file/d/10uLDDDMsF6YOO-8tjjSGMod9X8V3JBZU/view?usp=drive_link

1	2	3	4	5

14. B *

Link to explanation: https://drive.google.com/file/d/10t1SgwAlaJ98Fk9S5_qHyCiKSJ4grf4g/view?usp=drive_link

1	2	3	4	5