


☰

 SoftwareImpacts / SIMPAC-2024-159


🔍

+

🔄

🔗

📧



<> Code 🔗 Pull requests ⏮ Actions 📁 Projects 🛡 Security 📈 Insights

SIMPAC-2024-159 / README.md 📄

...

 phibal12 Update README.md

5e5ba94 · 1 hour ago 🕒

141 lines (106 loc) · 10.5 KB

Preview Code Blame

Raw 📄 ⬇️ ✎ ⌵ ☰

# 🔗 QFLCS: Quantum Field Lens Coding Simulator

Technical:  GitHub  SoftwareImpacts  Mendeley Data v.3+  Code in VS Code v.1.3  Python v.3+  Jupyter Lab  Open in Code Ocean

Social:  GitHub  phibal12 profile + projects  LinkedIn

Academic:  ORCID 0000-0003-1037-018X

Legal:  License  CC BY 4.0

This repository contains the code for the QFLCS (Quantum Field Lens Coding Simulator) as part of its algorithm, QFLCA (Quantum Field Lens Coding Algorithm) project. The project repositories are available at <https://data.mendeley.com/datasets/gf2s8jkdjf/3> and <https://doi.org/10.24433/CO.9905505.v2>, which include the code, project website documentation, and demo video files.

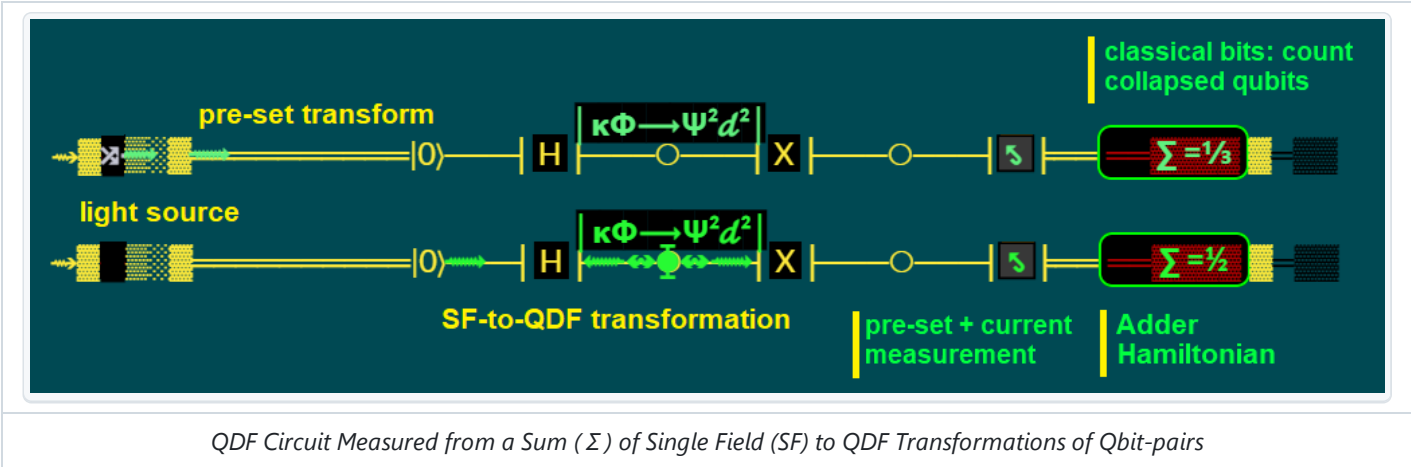


The QFLCS program analyzes the measurement outcome probability ( $P$ ) data from datasets generated by Quantum Double-field (QDF) Circuits. The datasets are compared between ES and GS states as a  $P$  indicator generated for measurement samples. Small dataset samples denote:



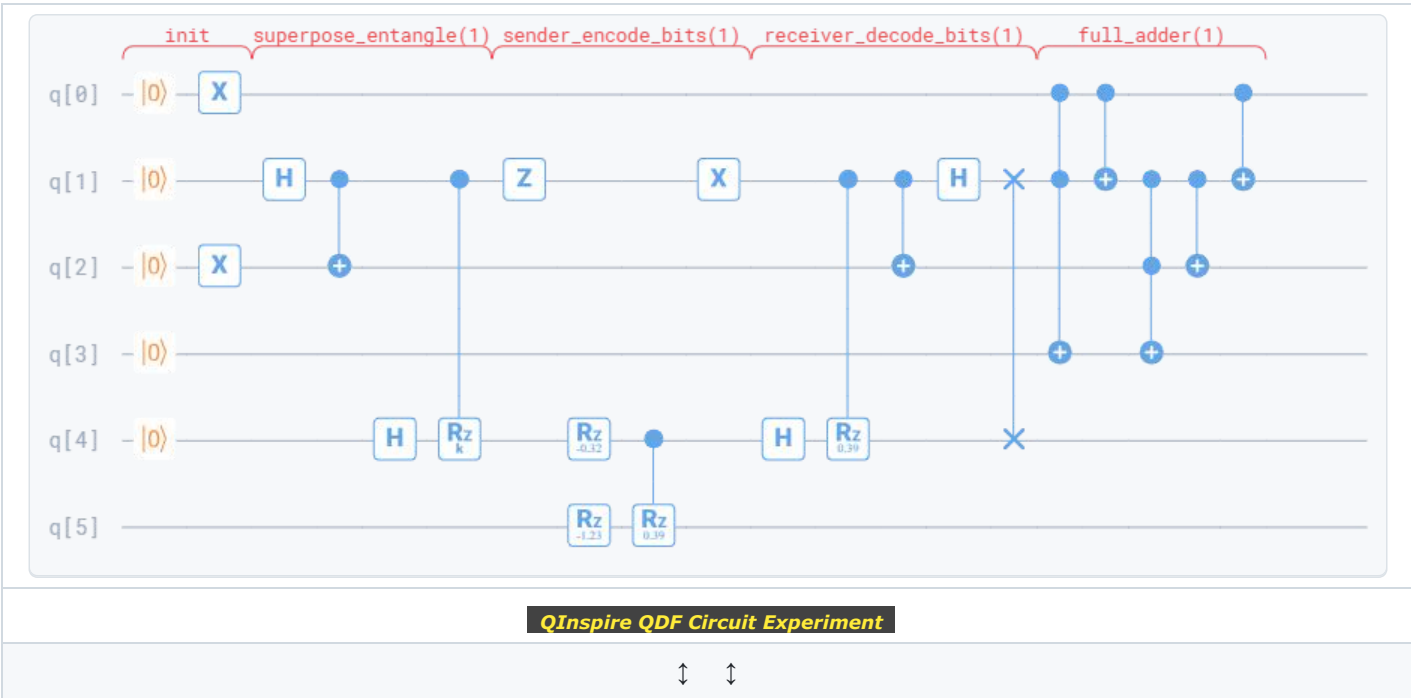
QDF Heat Engine Sampling from the Oldest Postal Stamp and Simulating the I/O by QFLCS

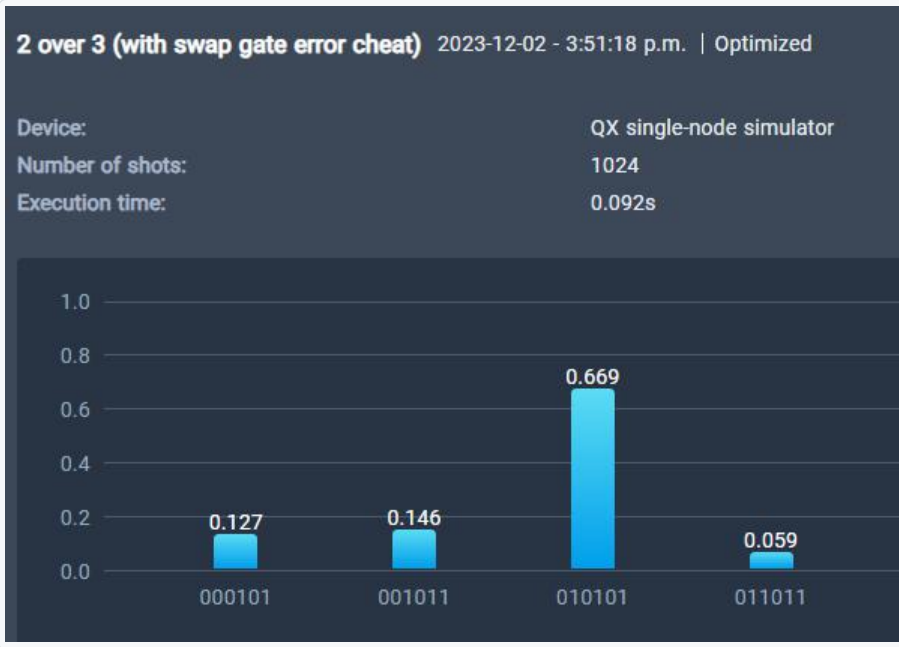
- a) A particle pair’s energy state in a QDF (different GS states or sublevels of a GS, or see Sec. 3 of the published article),
- b) a particle state in an SF, an ES relative to a GS from (a.), prior to its transform into a QDF, and,
- c) the expected transformation of fields ( $ES \leftrightarrow GS$ ) and their  $\langle M(P, \psi_{ij}) \rangle$ , as in Sec. 3 of the published article.



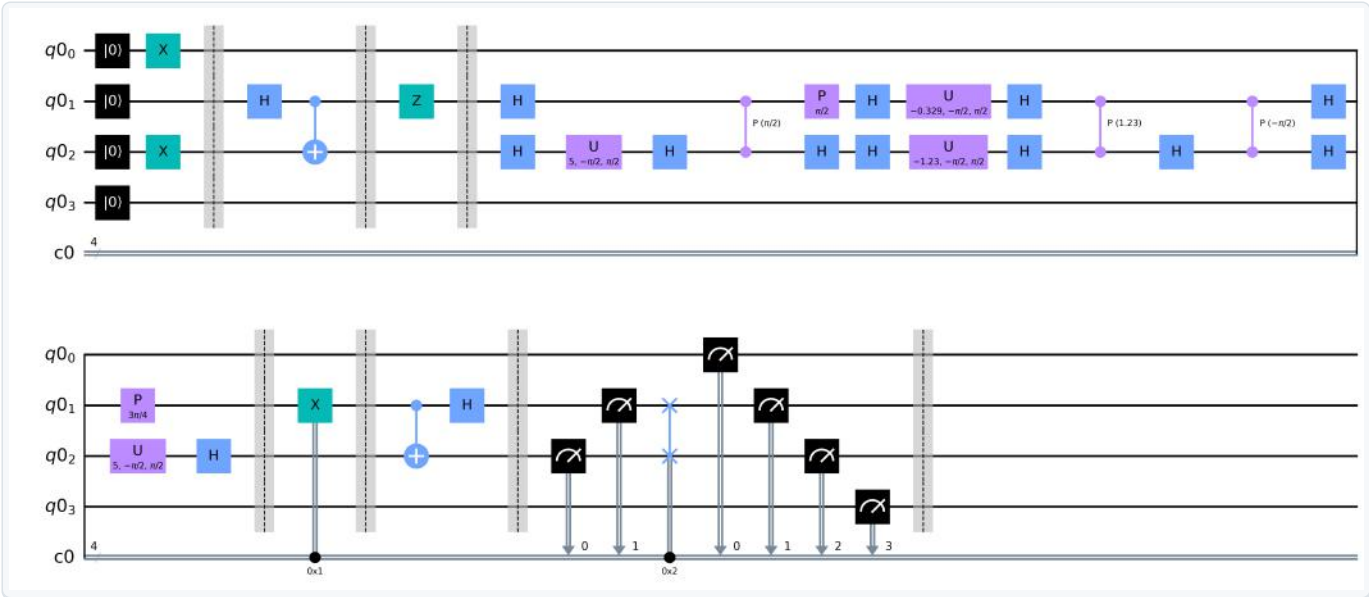
This repository's file structure is a sample mirror of the Mendeley repository file structure of v3+ at <https://data.mendeley.com/datasets/qf2s8jkdf/3>, but with a much smaller file size for efficient download and use of the QFLCA project's code without the documentation (website) and demo video files. Certain small updates have been made in the main python file uploaded on Code Ocean for minor debugging purposes at <https://doi.org/10.24433/CO.9905505.v2> or [Code Ocean](#).

- QFLCA project's code without the documentation (website) and demo video files can be found under the `</code>root/lab/sim/QFLCC classifiers>` directory in `</code>root/lab/sim/QFLCC classifiers/docs>` and `</code>root/lab/sim/QFLCC classifiers/site>` folders at <https://doi.org/10.24433/CO.9905505.v2> or [Code Ocean](#).
- The main file is `</code>root/lab/sim/QFLCC classifiers/QAI-LCode_QFLCC>` which imports and executes the `</code>root/lab/sim/QFLCC classifiers/QDF-LCode_IBMQ-2024-codable>` or `QDF-LCode_IBMQ-2024` code for the simulation under Win OS or Linux OS.





**QInspire QDF Circuit Experiment Result**



**IBM QDF Circuit Experiment**

**In** ( [QInspire, IBMQ] QDF Circuit ) → **QFLCS** → **Out** ( [QInspire, IBMQ] QDF Circuit Data ) → **In** ( **QFLCC** ) → **Out** ( ↴ )

```

An ideal QDF circuit I/O model running realtime producing an IBMQ-based QDF dataset is from the
QDF-LCode-IBMQ-2024.py file. This circuit is compared to the analyzed dataset to show how close
the match is for a desired Hamiltonian and expected measurement outcome as a point of reference.
=====
on 0: Please Wait!...
on 0: *- Program is processing the analyzed dataset circuit P's after:
on 2: *- Import + Simulate the IBM QDF circuit by { Qiskit Aer Simulator, IBMQ Provider } [QUANTUM MODE ENVIRONMENT] on this computer.
on 3: *- Dataset P results are compared on the imported circuit by the { QDF-LCode-IBMQ-2024.py } module on this computer [SAFE MODE ENVIRONMENT].
|=====| 100% [3/3] in 3.1s (1.13/s)
=====
Quantum simulation of a quantum field lens coding algorithm with entanglement scaling between a
multi-well (barrier) interaction potential of internal system B interacting and external system A from the method article:
https://www.sciencedirect.com/science/article/pii/S221501612300136X
*- QDF Circuit Built and Simulator generates its realtime datasets from { Qiskit Aer, IBMQ } Qasm_Simulator
  Workspace and/or Virtual Environment on user's computer in Python.
*- Created by: Phillip B. Alipour, Supervisor: T. A. Gulliver, at the University of Victoria,
  Dept. ECE, Victoria BC, Canada.
*- Code updated based on Qiskit 2023--2024 changes specified in code comments of this simulator.
*- Sidenotes: You can also run code with the right packages installed in pipx or python.
  Examples of package installation changes and features can be found on:
  https://docs.quantum.ibm.com/api/migration-guides/qiskit-1.0-installation and
  https://docs.quantum.ibm.com/api/migration-guides/qiskit-1.0-features
=====
<-- { { Qiskit Aer, IBM }} QDF CIRCUIT SIMULATION BEGINS -->
Dynamic Providers: QasmSimulator('qasm_simulator')
Counts for Qubit Pairs out of 8192 shots: {'0101': 5541, '0111': 2243, '0001': 408}
Σ (5541, 2243, 408) = 8192 shots → {P(bij)} = { ({n1, n2, n3}/8192)bij }
= { p1(0101) = 0.6763916015625 } + { p2(0111) = 0.2738037109375 } + { p3(0001) = 0.0498046875 } = 1
Plot = [Experimented n of N counts ∝ p of total circuit events P on pairwise qubits = P(bij)]; |i> = |q1 q2> = |q1q2>.
=====
P(bij)          [1.00000000000000]
p1(bij) = p1(0101) [0.6763916015625]
p2(bij) = p2(0111) [0.2738037109375]
p3(bij) = p3(0001) [0.0498046875000]
=====
<-- IBM QDF Circuit Measurement Results by Qiskit Aer Simulator Plotted Successfully! End of Task... -->
=====
q1_0: -|0>
q1_1: -|0>
q1_2: -|0>
q1_3: -|0>
c0: 4/

«
«q1_0:
«q1_1:
«q1_2:
«q1_3:
«c0: 4/
«

(Σ) on circuit components used per QDF circuit simulation run generating new/updated dataset results.
<-- QDF Circuit BUILT & RAN Successfully! End of Task... -->
<-- { { Qiskit Aer, IBM }} QDF CIRCUIT SIMULATION CONCLUDED -->

```

## [QInspire, IBMQ] QDF Circuit Simulation and Dataset Analysis by QFLCS and QFLCC

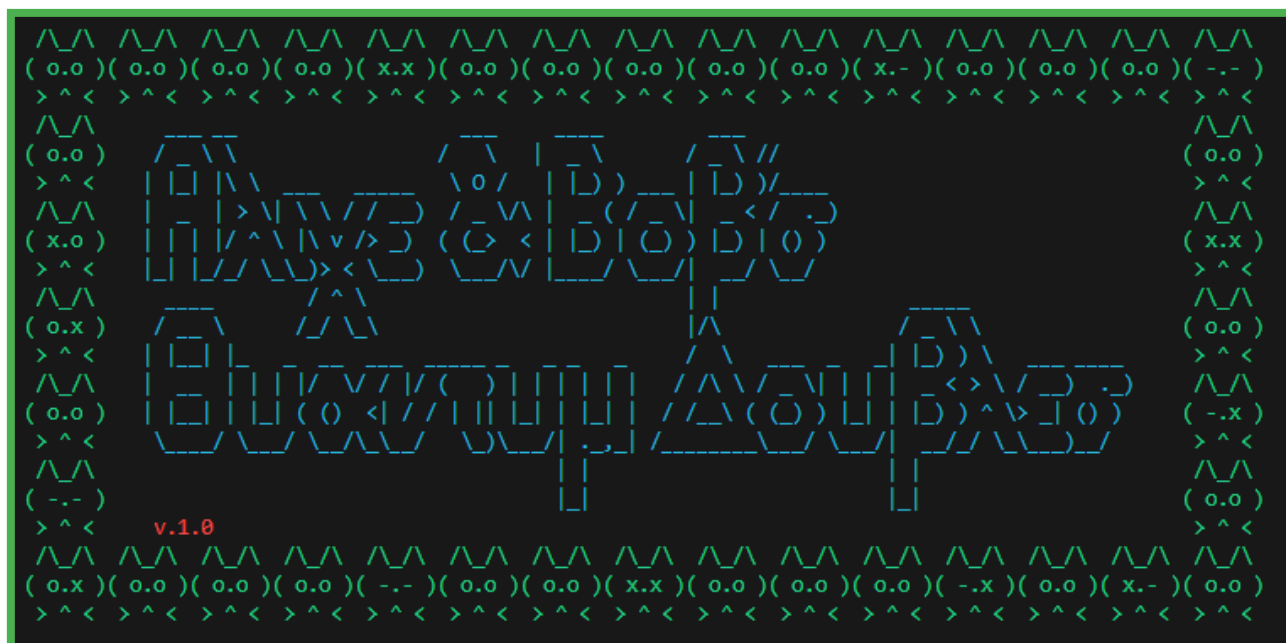
- We recommend downloading the entire `<root/...>` directory according to the folder structure and run `[QAI-LCode_QFLCC.py]` in VSC with python latest packages installed for Windows OS (the QDF game is developed for Windows OS, yet parts of the code for sound and display can be rewritten for Linux OS), e.g. `"winsound"` package as a compatible option. Other packages are needed to be installed or code rewritten for `"sound"` and `"display"` compatibility under other operating systems.
- The `[QAI-LCode_QFLCC.py]` file has a Pygame GUI and other packages suited for local machine runs, rather than running this file on the [Code Ocean](#) platform which could take hours to compile and run a compatible program/game with packages. However, the `[QDF-LCode_IBMQ-2024-codable.py]` can be run here as the core of the simulation program simulating the QDF circuit. A short presentation explaining these points are given in the `</site/assets/video>` directory as the `[QAI-COcean-Demo.mp4]` file from [Code Ocean](#).



QDF Game Intro from the QFLCA/QFLCS program

### QDF Game Intro from the QFLCA/QFLCS program

- The User and Developer's documentation/manual/demo is found under the `</code>root/lab/sim/QFLCC classifiers>` directory, as `<site-prints/...>` and `<site/...>` contents.
- In each folder: `<QFLCC classifiers/IBMQ>`, `<.../sim/QAI>`, `</QFLCC classifiers>`, and `<QI/...>`, under `</code>root/lab/sim>`, `[Tips.txt]` and/or `[ReadMe.txt]` files exist to explain the contents of that directory. Also, under `</code>root/lab>` directory, a ReadMe file exists explaining the manual computation and presentation parts of the project.



```

Level 0 engaged!
In this game, are you Bob [o] the guest, or Alice [★] the host to win a/the prize (targetted energy state) [★]? Choose 1 for Bob, 2 for Alice: 2
For your game participant 2, will Eve [★] join by quantum means to secretly share information about the prize [★]? Choose 3 to have Eve spying, 4 for Audience [★] to cheer/su
ggest and raise/lower participant 2's energy state: 3
Enter a P value for participant #2, [★]: 0.62
> n
Next...

List of tried P's: [0.62]
List of assigned weights: [0.0]
List of matched ΔP's: [0.0]
Subset of tried P's: {0.62}
Duplicated P tries: set()
Subset of calculated ΔP's: {0.0}
Duplicates of matched P's: set()
P value 0.62 guessed an undefined P outcome of the qubit dataset. Your successful hit weight was: [0, inf]
Hello, Quantum World!
Correlation to strong prediction result for participant #2, [★], is high to win: 0.944219
Bob loses to Alice. Alice wins to keep the prize with Eve's help! YOU WIN!
These doubles won the prize: ['01', '1b'], QDF P match is: 0.94
Your score is: 10. Scoresheet is: [0, 10]
--Σ(((★ ☆ ♪ ♪) [★]
Level 1 engaged!
--Σ(((★ (O) ♪) [★]
Next... Level: 1 ♪

In this game, are you Bob [o] the guest, or Alice [★] the host to win a/the prize (targetted energy state) [★]? Choose 1 for Bob, 2 for Alice: 2
For your game participant 2, will Eve [★] join by quantum means to secretly share information about the prize [★]? Choose 3 to have Eve spying, 4 for Audience [★] to cheer/su
ggest and raise/lower participant 2's energy state: 3
Enter a P value for participant #2, [★]: 0.33
> n
Next...

List of tried P's: [0.62, 0.33]
List of assigned weights: [0.0, 1.059]
List of matched ΔP's: [0.0, 0.06]
Subset of tried P's: {0.62, 0.33}
Duplicated P tries: set()
Subset of calculated ΔP's: {0.0, 0.06}
Duplicates of matched P's: set()
P value 0.33 guessed a strong classical P outcome and correlated with max(ΔP)=min(P) of qubit dataset. Your successful
Hello, Quantum World!
Correlation to strong prediction result for participant #2, [★], is low to win: 0.65
Bob loses to Alice despite Eve's help. Alice wins to keep the prize! YOU [★] via [★] LOSE!
These doubles lost the prize: ['01', '1b'] lost the prize to ['10'], QDF P match is: 0.65
Your score is: 5. Scoresheet is: [0, 10, 10, 5]
--Σ(((r(π ♪ π)
Level 0 initiated! Your score is: 5.
--Σ(((★ (O) ♪) [★]
Level 0 engaged!

Next... Level: 0 ♪

In this game, are you Bob [o] the guest, or Alice [★] the host to win a/the prize (targetted energy state) [★]? Choose 1 for Bob, 2 for Alice: 2
For your game participant 1, will Eve [★] join by quantum means to secretly share information about the prize [★]? Choose 3 to have Eve spying, 4 for Audience [★] to cheer/su
ggest and raise/lower participant 1's energy state: 4
Enter a P value for participant #1, [★]: 0.63
> n
Next...

List of tried P's: [0.62, 0.33, 0.63]
List of assigned weights: [0.0, 1.059, 1.529]
List of matched ΔP's: [0.0, 0.06, 0.35]
Subset of tried P's: {0.62, 0.63, 0.33}
Duplicated P tries: set()
Subset of calculated ΔP's: {0.0, 0.35, 0.06}
Duplicates of matched P's: set()
P value 0.63 guessed a classical P outcome and correlated with min(ΔP)=max(P) of qubit dataset. Your successful hit weight was: 1.529
Hello, Quantum World!
Correlation to strong prediction result for participant #1, [o], is high to win: 0.954219
Alice loses to Bob. Bob wins the prize as the Audience cheer without Eve's help! YOU WIN!
These doubles won the prize: ['01', '1b'], QDF P match is: 0.95
Your score is: 15. Scoresheet is: [0, 10, 10, 5, 5, 15]
--Σ(((★ ☆ ♪ ♪) [★]
Level 1 engaged!
--Σ(((★ (O) ♪) [★]
Next... Level: 1 ♪

In this game, are you Bob [o] the guest, or Alice [★] the host to win a/the prize (targetted energy state) [★]? Choose 1 for Bob, 2 for Alice: 2
For your game participant 2, will Eve [★] join by quantum means to secretly share information about the prize [★]? Choose 3 to have Eve spying, 4 for Audience [★] to cheer/su
ggest and raise/lower participant 2's energy state: 4
Enter a P value for participant #2, [o]: 0.62
> n
Next...

List of tried P's: [0.62, 0.33, 0.63, 0.62]
List of assigned weights: [0.0, 1.059, 1.529, 1.048]
List of matched ΔP's: [0.0, 0.06, 0.35, 0.05]
Subset of tried P's: {0.62, 0.63, 0.33}
Duplicated P tries: {0.62}
Subset of calculated ΔP's: {0.0, 0.35, 0.06, 0.05}
Duplicates of matched P's: set()
Duplicates {0.62} found in your tried P's list! Try again...
[[[O) ♪]

```

--Σ(((★ ☆ ♪ ♪) [★]

Level 1 engaged!

--Σ(((★ (O) ♪) [★]

A custom victorious emoji is animated with a retro style sound. Levels can go up or down depending on the energy scoresheet obtained as losses and gains in winning a prize [★] (TS) and/or the prize given the energy state by the user (as Bob or Alice)

Prompt repeated after a game loss, win, hint, or restart...

--Σ(((r(π ♪ π)

Level 0 initiated! Your

--Σ(((★ (O) ♪) [★]

Level 0 engaged!

A game loss of Bob [o] to Alice [★] far from a TS hit, despite Eve [★]'s help. Perhaps, there was not a good cheat sheet.

Hence, a more intelligent algorithm can be trained for the user based on this experience if the user chooses. In this scenario, we see the level descended from Level 1 to Level 0 for the user, and the loss icon shows an animated crying emoji.

--Σ(((★ ☆ ♪ ♪) [★]

A close hit and gain of a prize [★]. This scores a good win due to w being close to 1.

QDF Game Scores on Wins and Losses of the User/Gamer based on the Input Samples by the QFLCS

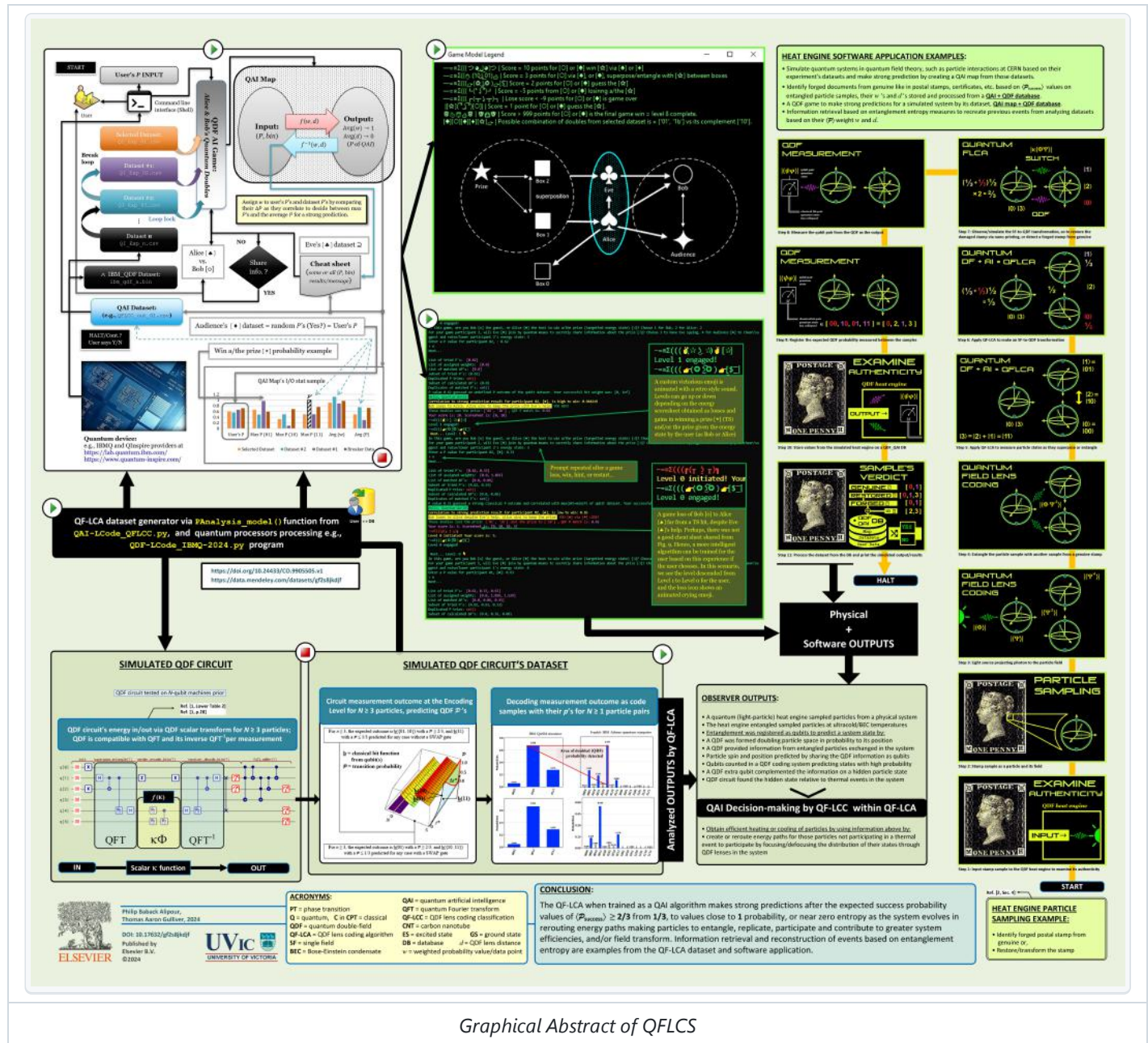
- Graphical Abstract files can be accessed from `</code>root>`



# Visual Project Summary

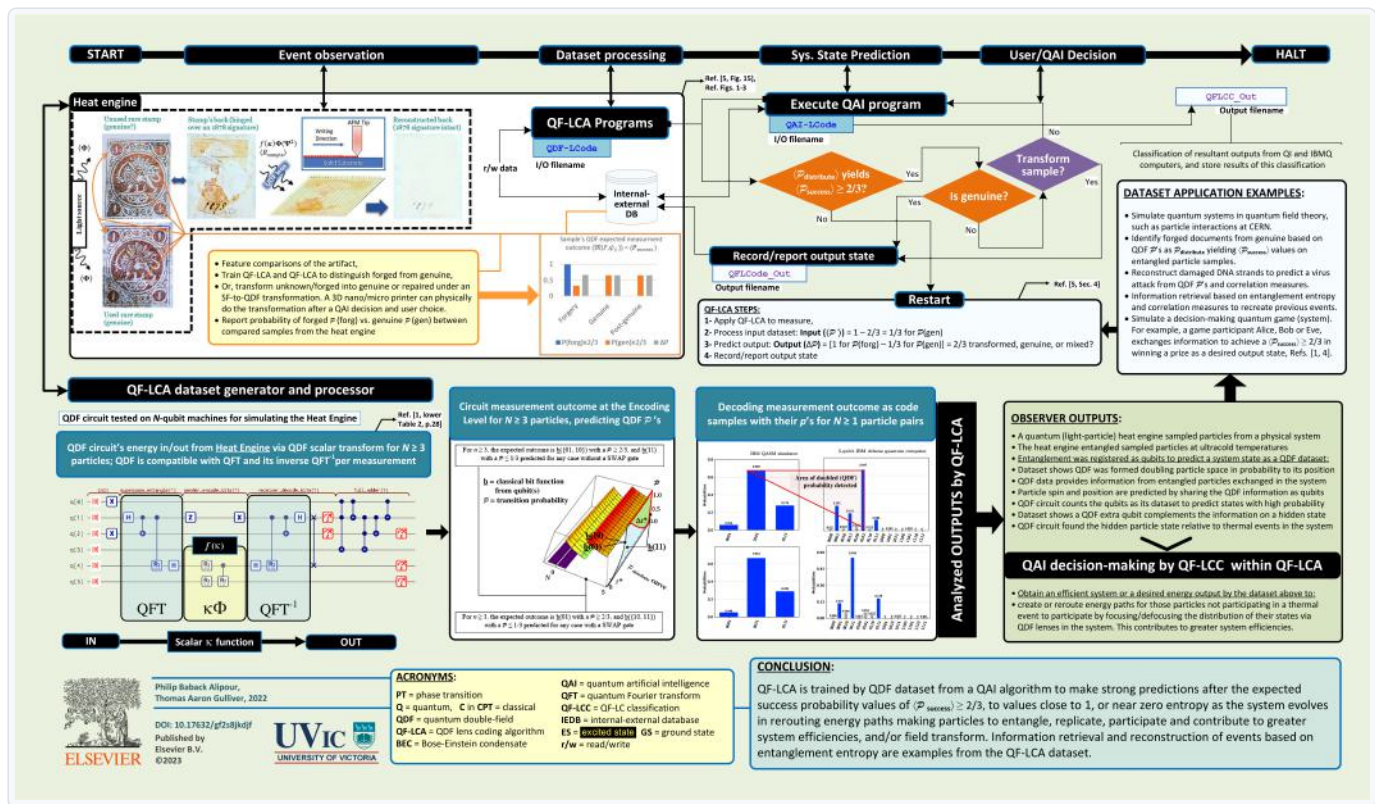
The QFLCA project is going through stages of development resulting in software products, such as QFLCC and QFLCS programs. These products have been summarized in form of peer-reviewed published articles with their corresponding Graphical Abstracts representing this project.

- The following figure is a downloadable High-Res Graphical Abstract of the published QFLCS article in Software Impacts, Elsevier BV, at: <https://www.sciencedirect.com/science/article/pii/S2665963824000915>



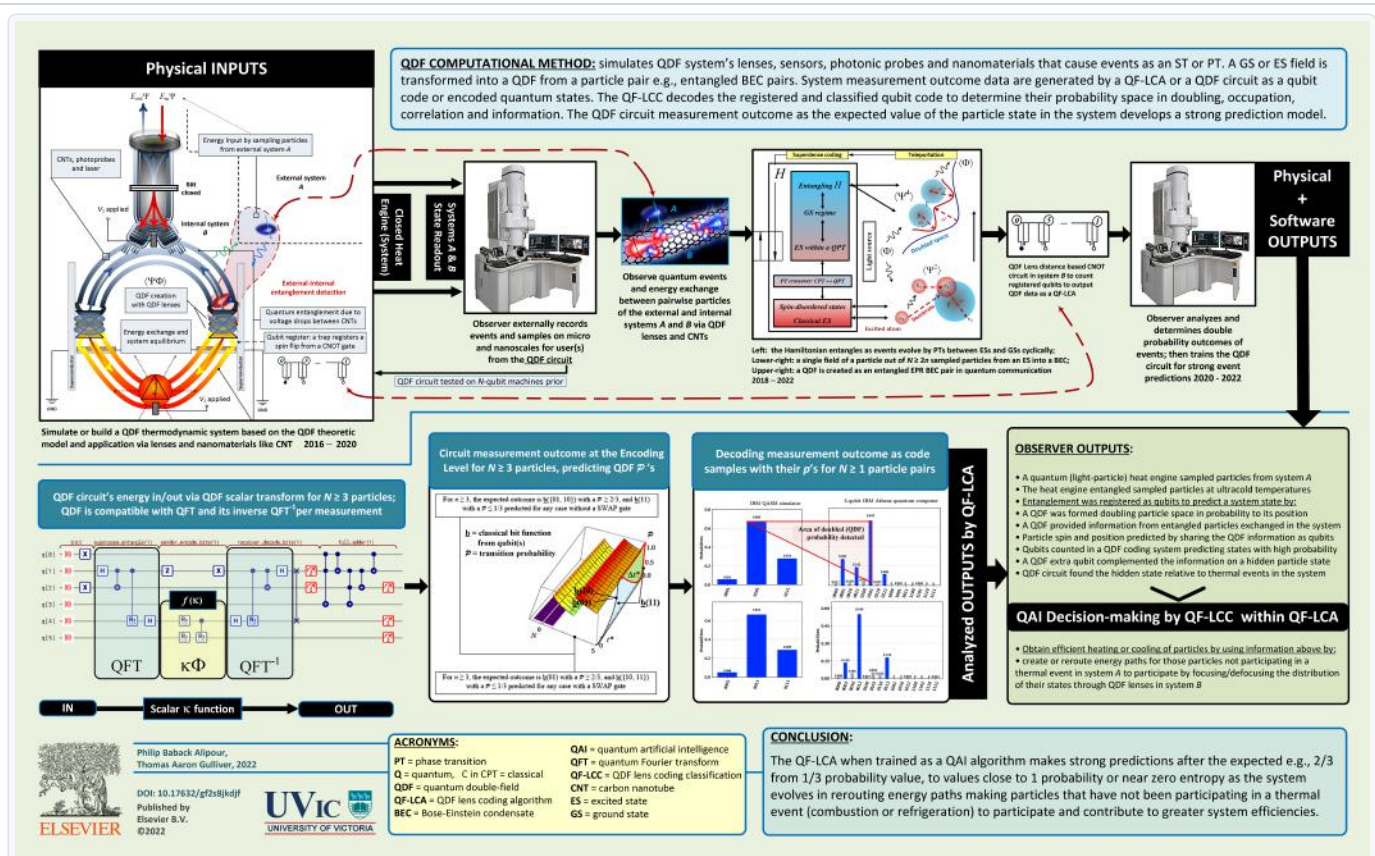
- The following figure, is a downloadable High-Res Graphical Abstract of the published QFLCA article in Data in Brief, Elsevier BV, at: <https://www.sciencedirect.com/science/article/pii/S2352340924007546>





### Graphical Abstract of QFLCA Dataset

- The following figure is a downloadable High-Res Graphical Abstract of the published QFLCA article in MethodsX, Elsevier BV, at <https://www.sciencedirect.com/science/article/pii/S221501612300136X>



Graphical Abstract of QFLCA Model and Method



# Citation

If you find this repository useful in your research, please cite one or both of the following articles as:

- P. B. Alipour, T. A. Gulliver, QF-LCA Dataset: Quantum Field Lens Coding Algorithm for System State Simulation and Strong Predictions, Data in Brief, Eslevier BV, 2024, 110789, ISSN 2352-3409, <https://doi.org/10.1016/j.dib.2024.110789>.

```
@article{110789,  
  author = {Alipour, P.B. and Gulliver, T.A.},  
  title = {QF-LCA Dataset: Quantum Field Lens Coding Algorithm for System State Simulation and Strong  
  Predictions},  
  journal = {Data in Brief, Eslevier BV},  
  year = {2024},  
  artnum = {110789},  
  doi = {10.1016/j.dib.2024.110789},  
  url = {https://www.sciencedirect.com/science/article/pii/S2352340924007546}  
}
```



- P. B. Alipour, T. A. Gulliver, QF-LCS: Quantum Field Lens Coding Simulator and Game Tool for Strong System State Predictions, Software Impacts, Eslevier BV, 2024, 100703, ISSN 2665-9638, <https://doi.org/10.1016/j.simpa.2024.100703>.

```
@article{100703,  
  author = {Alipour, P.B. and Gulliver, T.A.},  
  title = {QF-LCS: Quantum Field Lens Coding Simulator and Game Tool for strong system state predictions},  
  journal = {Software Impacts, Eslevier BV},  
  pages = {100703},  
  year = {2024},  
  issn = {2665-9638},  
  doi = {doi.org/10.1016/j.simpa.2024.100703},  
  url = {https://www.sciencedirect.com/science/article/pii/S2665963824000915}  
}
```



License CC BY 4.0

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

