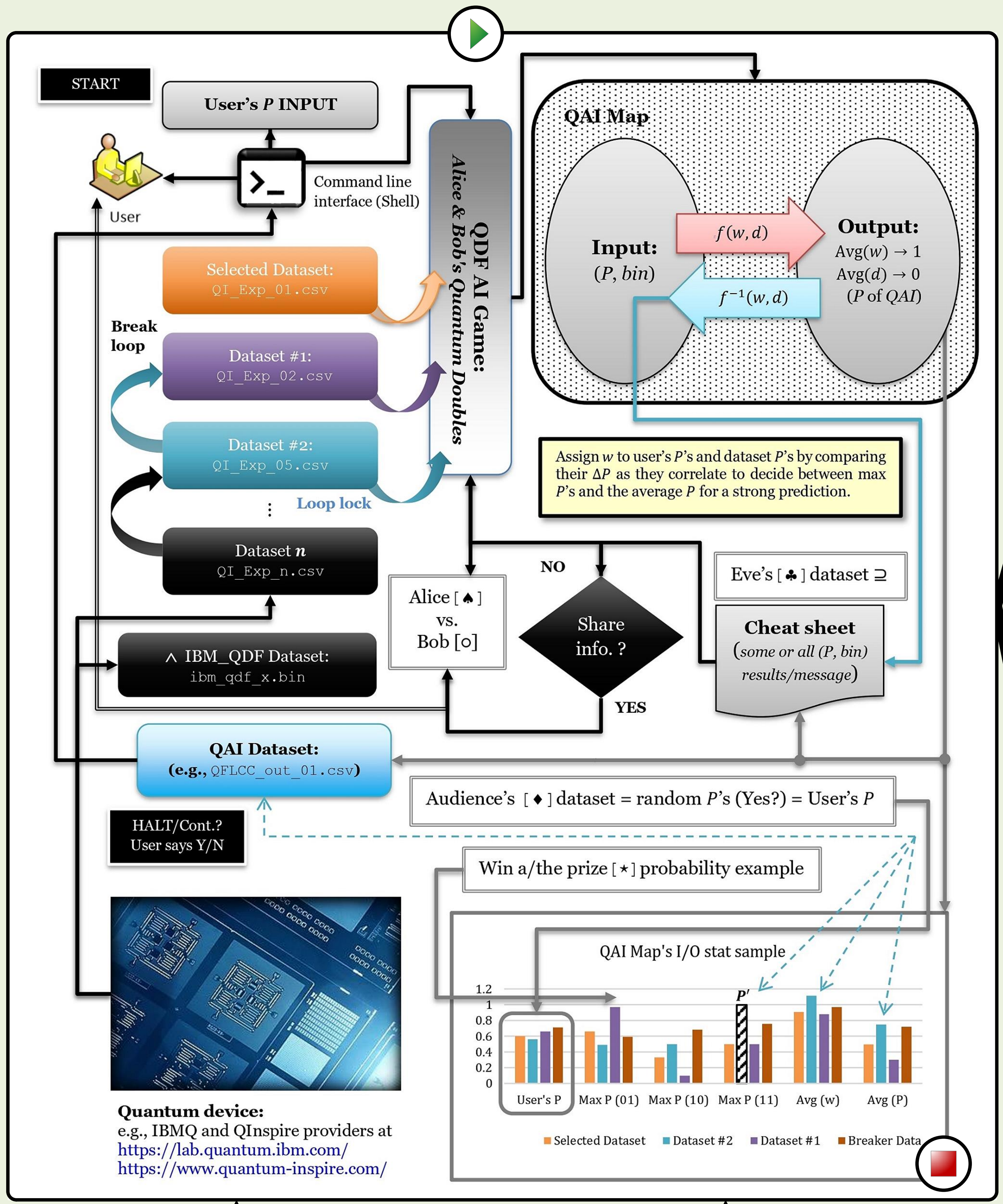


START

User's *P* INPUT

User

Command line interface (Shell)

QDF AI Game: Alice & Bob's Quantum Doubles

Selected Dataset: QI_Exp_01.csv

Break loop

Dataset #1: QI_Exp_02.csv

Dataset #2: QI_Exp_05.csv

Loop lock

Dataset *n* QI_Exp_n.csv

∧ IBM_QDF Dataset: ibm_qdf_x.bin

QAI Map

Input: (*P*, *bin*)

$f(w,d)$

$f^{-1}(w,d)$

Output: Avg(w) → 1 Avg(d) → 0 (*P* of *QAI*)

Assign *w* to user's *P*'s and dataset *P*'s by comparing their Δ*P* as they correlate to decide between max *P*'s and the average *P* for a strong prediction.

Alice [♠] vs. Bob [○]

NO

Share info. ?

YES

Eve's [♣] dataset ⊇

Cheat sheet (*some or all (P, bin) results/message*)

QAI Dataset: (e.g., QFLCC_out_01.csv)

Audience's [♦] dataset = random *P*'s (Yes?) = User's *P*

HALT/Cont.? User says Y/N

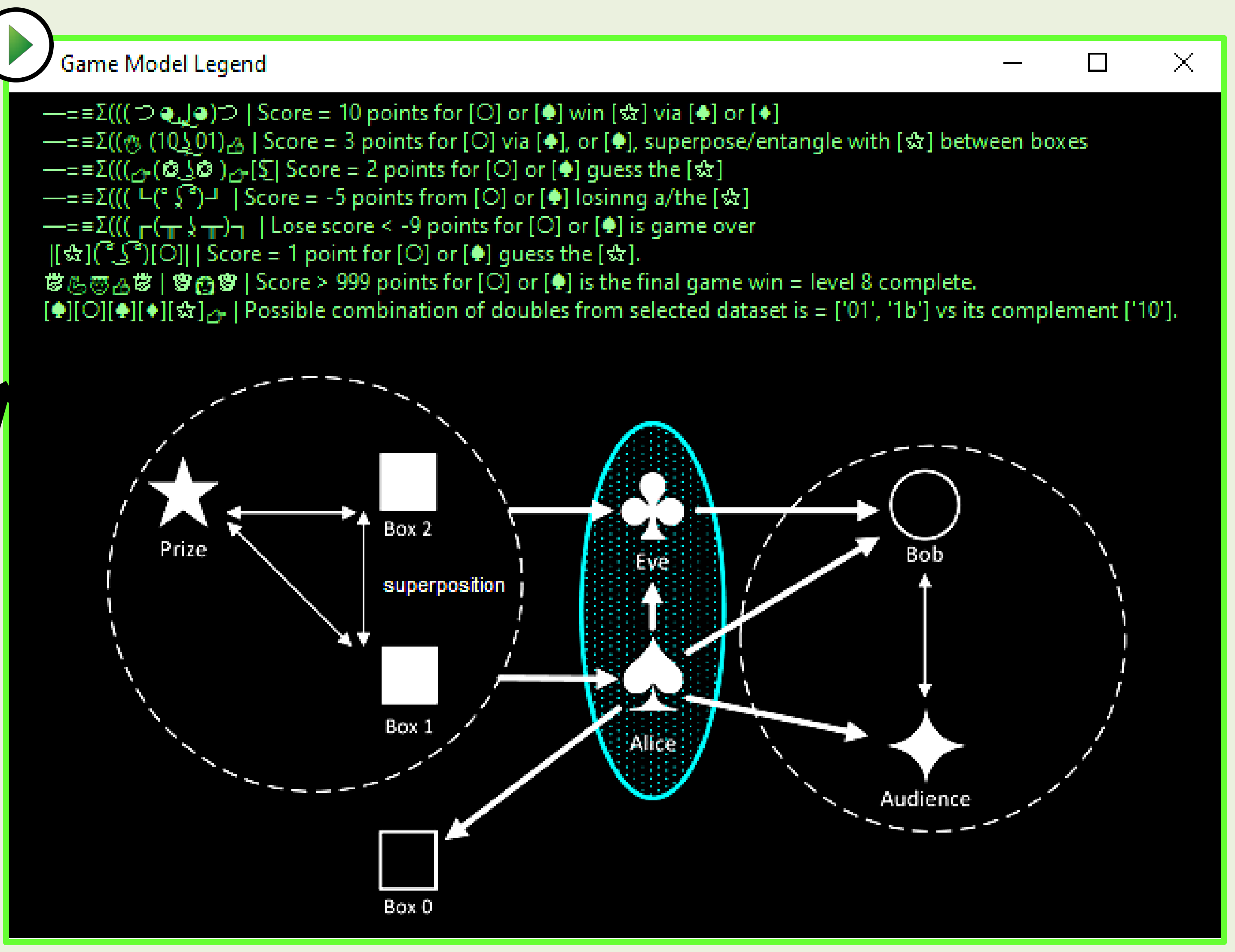
Win a/the prize [⋆] probability example

Quantum device:
e.g., IBMQ and QInspire providers at
https://lab.quantum.ibm.com/
https://www.quantum-inspire.com/

QF-LCA dataset generator via `PAnalysis_model()` function from `QAI-LCode_QFLCC.py`, and quantum processors processing e.g., `QDF-LCode_IBMQ-2024.py` program

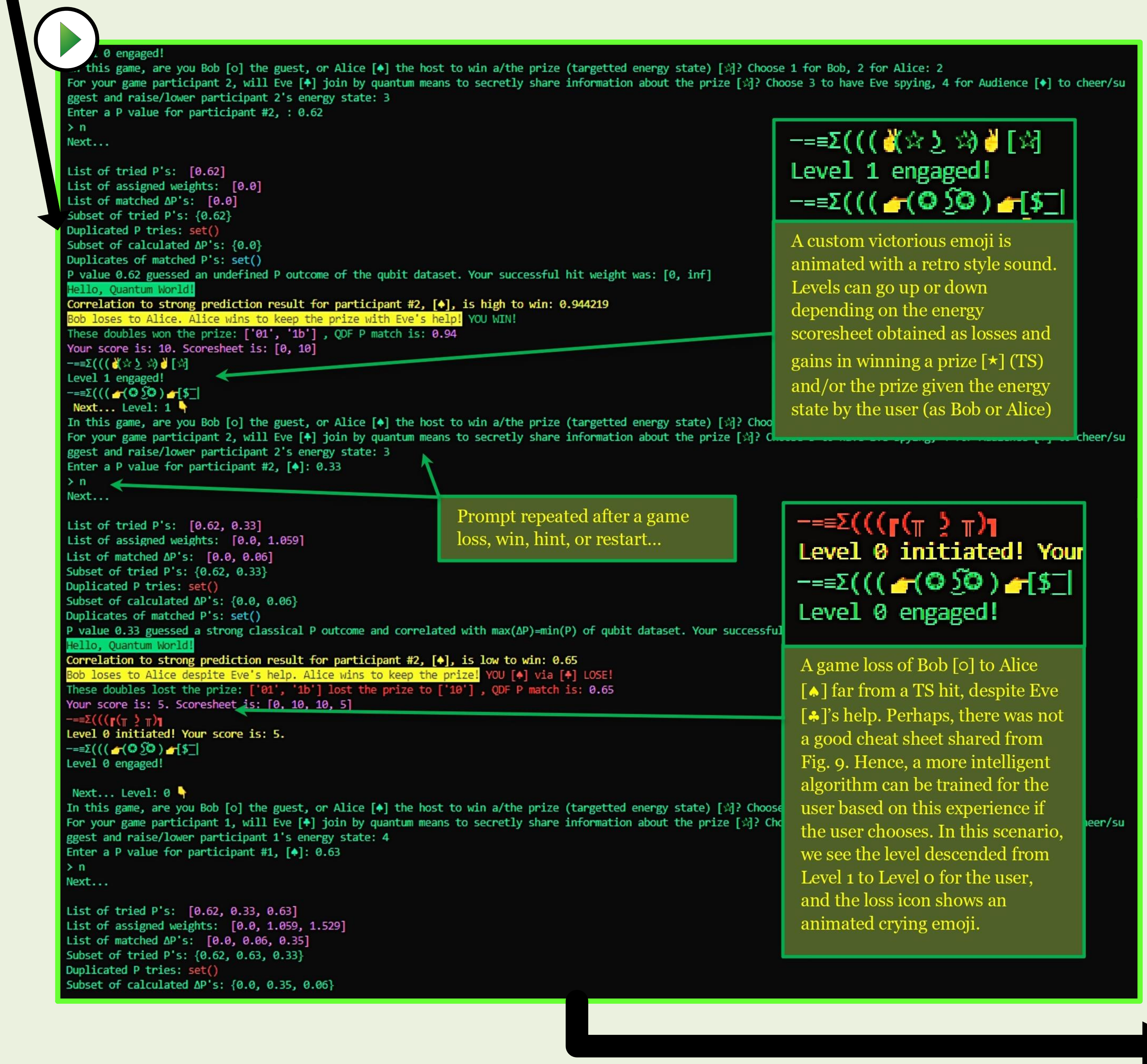
User ↔ DB

https://doi.org/10.24433/CO.9905505.v1
https://data.mendeley.com/datasets/gf2s8jkdjf



Game Model Legend

```
—=≡Σ((( つ◕‿◕)つ | Score = 10 points for [○] or [♠] win [☆] via [♣] or [♦]
—=≡Σ((☝ (10͜ʖ01)👍 | Score = 3 points for [○] via [♣], or [♠], superpose/entangle with [☆] between boxes
—=≡Σ(((☛(⚙͜ʖ⚙)☛[$] Score = 2 points for [○] or [♠] guess the [☆]
—=≡Σ((( └(°͜ʖ°)┘ | Score = -5 points from [○] or [♠] losinng a/the [☆]
—=≡Σ((( ┌(╥﹏╥)┐ | Lose score < -9 points for [○] or [♠] is game over
|[☆](͡°͜ʖ͡°)[○]| | Score = 1 point for [○] or [♠] guess the [☆].
😈👍😈👍😈 | 😈😊😈 | Score > 999 points for [○] or [♠] is the final game win = level 8 complete.
[♠][○][♣][♦][☆]☛ | Possible combination of doubles from selected dataset is = ['01', '1b'] vs its complement ['10'].
```



```
0 engaged!
 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(((✌☆͜ʖ☆)✌[☆]
Level 1 engaged!
−=≡Σ(((☛(⚙͜ʖ⚙)☛[$]
 Next... Level: 1 👇
In this game, are you Bob [o] the guest, or Alice [♠] the host to win a/the prize (targetted energy state) [☆]? Choo
For your game participant 2, will Eve [♣] join by quantum means to secretly share information about the prize [☆]? C                                     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(((┏(╥﹏╥)┓
Level 0 initiated! Your score is: 5.
−=≡Σ(((☛(⚙͜ʖ⚙)☛[$]
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
For your game participant 1, will Eve [♣] join by quantum means to secretly share information about the prize [☆]? Ch                                   eer/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}
```

```
−=≡Σ(((✌☆͜ʖ☆)✌[☆]
Level 1 engaged!
−=≡Σ(((☛(⚙͜ʖ⚙)☛[$]
```

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)

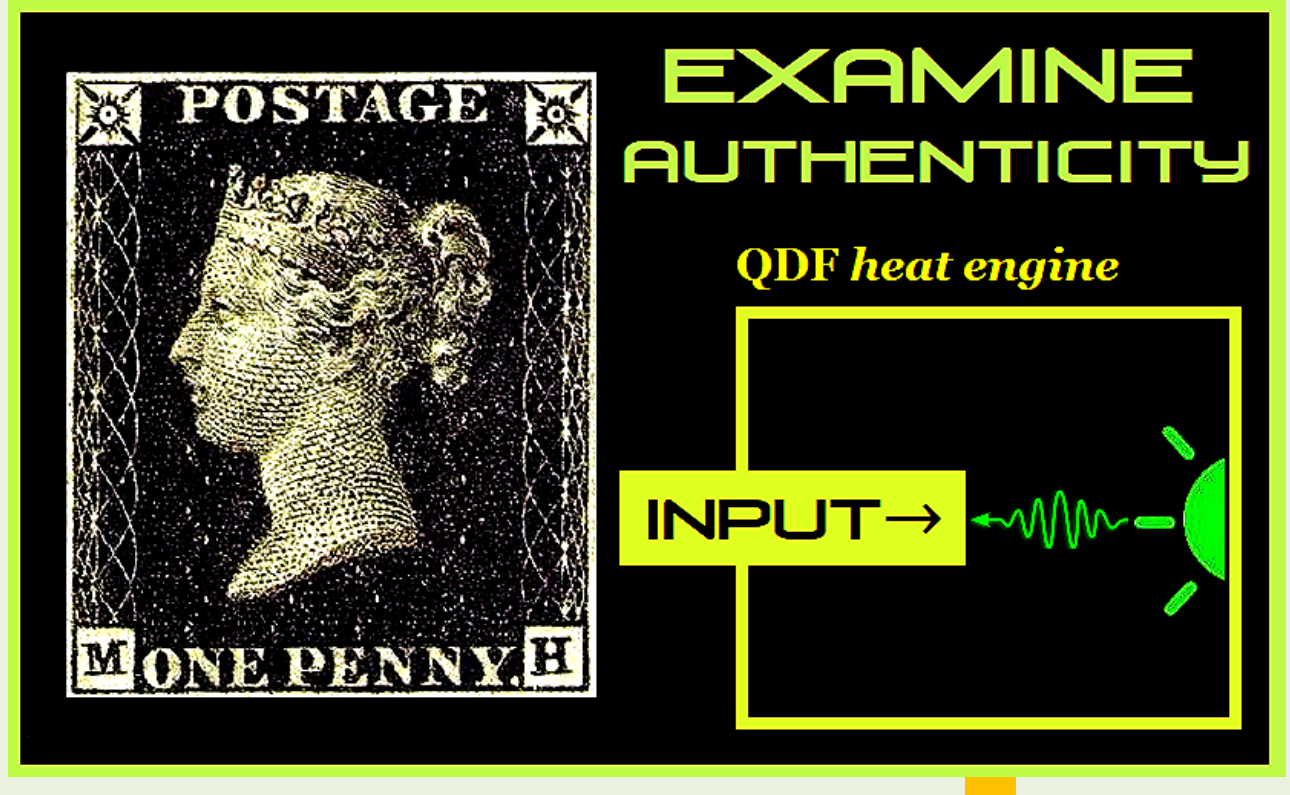
Prompt repeated after a game loss, win, hint, or restart...

```
−=≡Σ(((┏(╥﹏╥)┓
Level 0 initiated! Your
−=≡Σ(((☛(⚙͜ʖ⚙)☛[$]
Level 0 engaged!
```

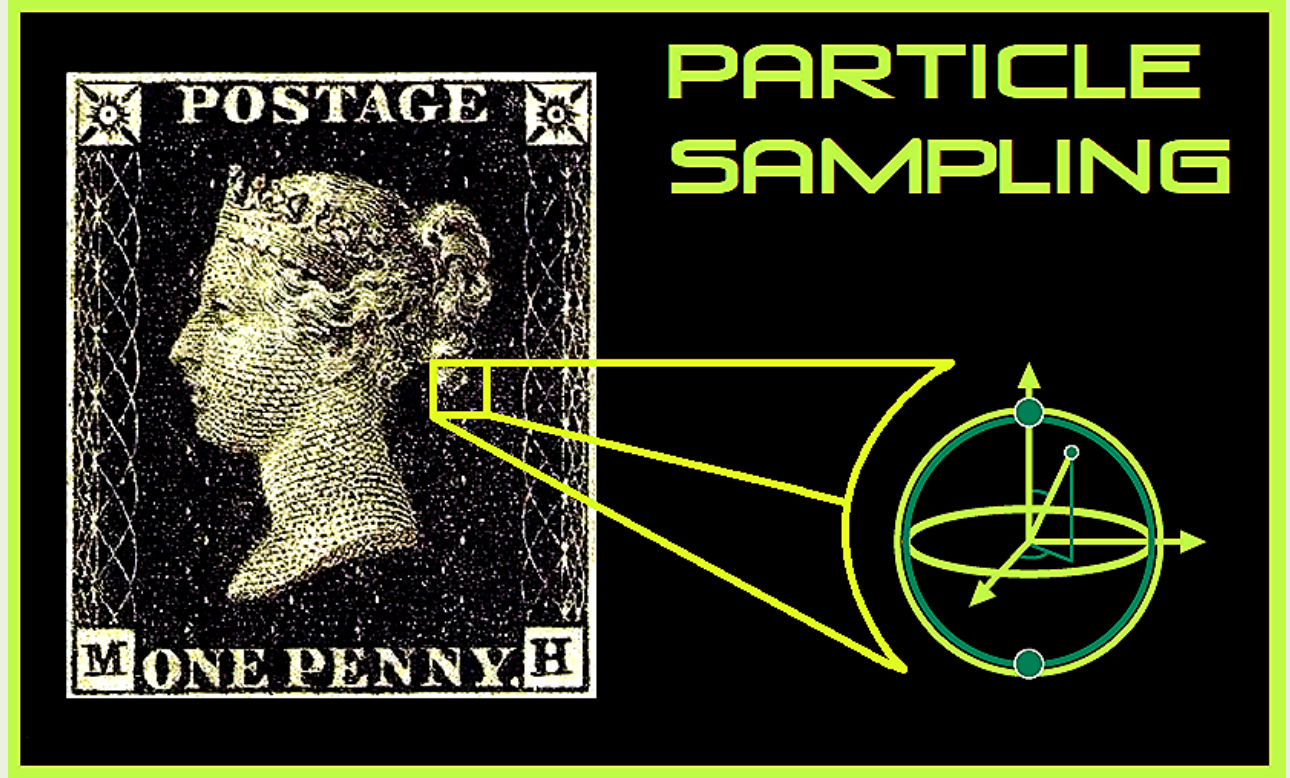
A game loss of Bob [○] to Alice [♠] far from a TS hit, despite Eve [♣]'s help. Perhaps, there was not a good cheat sheet shared from Fig. 9. Hence, a more intelligent algorithm can be trained for the user based on this experience if the user chooses. In this scenario, we see the level descended from Level 1 to Level 0 for the user, and the loss icon shows an animated crying emoji.

HEAT ENGINE SOFTWARE APPLICATION EXAMPLES:

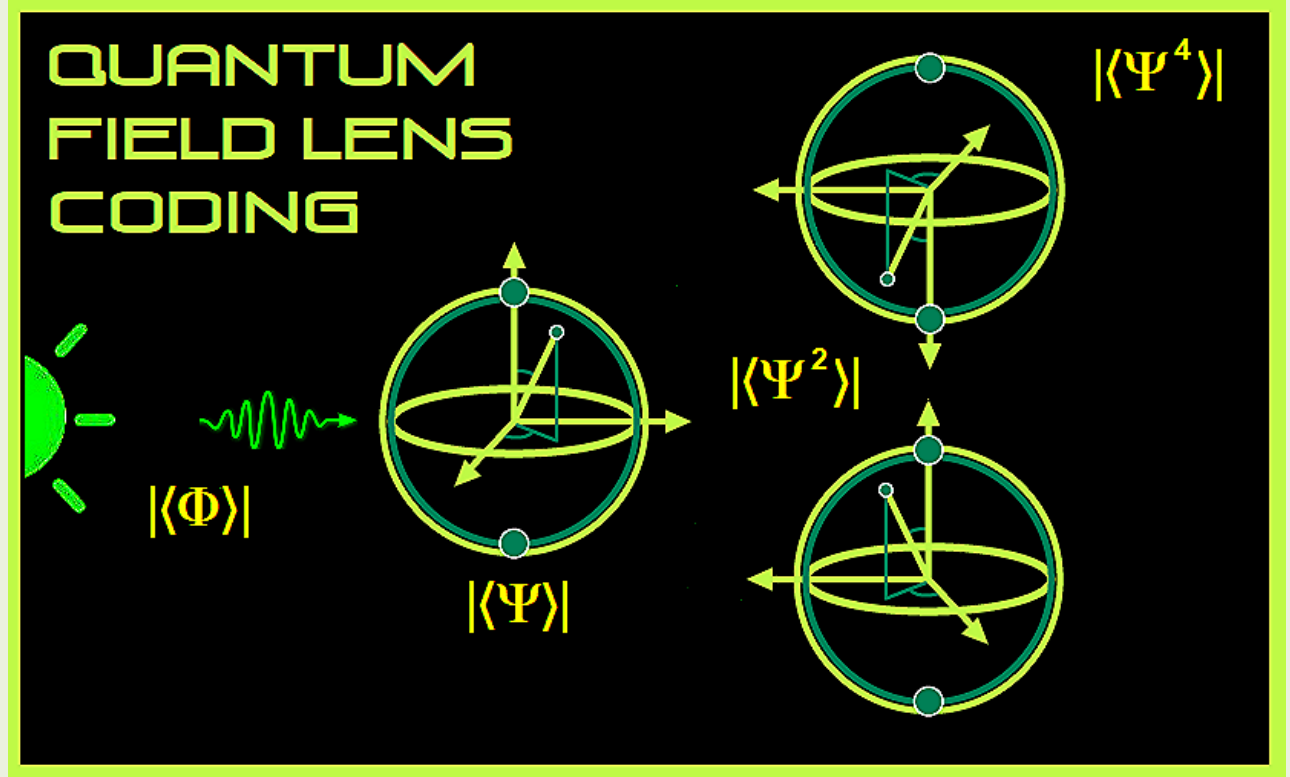
- Simulate quantum systems in quantum field theory, such as particle interactions at CERN based on their experiment's datasets and make strong prediction by creating a QAI map from those datasets.
- Identify forged documents from genuine like in postal stamps, certificates, etc. based on $\langle \mathcal{P}_{\text{success}} \rangle$ values on entangled particle samples, their w's and d's stored and processed from a **QAI + QDF database**.
- A QDF game to make strong predictions for a simulated system by its dataset, **QAI map + QDF database**.
- Information retrieval based on entanglement entropy measures to recreate previous events from analyzing datasets based on their $\langle \mathcal{P} \rangle$ weight w and d.



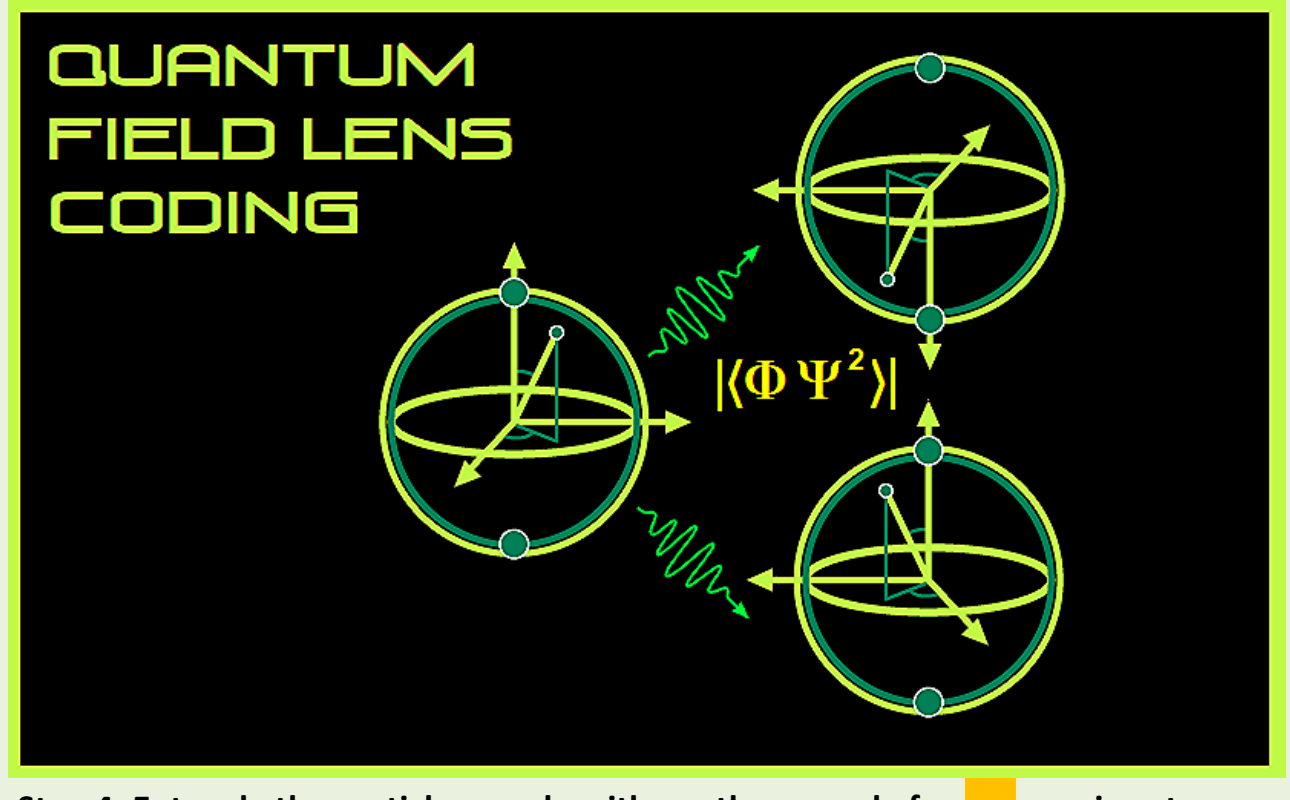
Step 1: Input stamp sample to the QDF heat engine to examine its authenticity



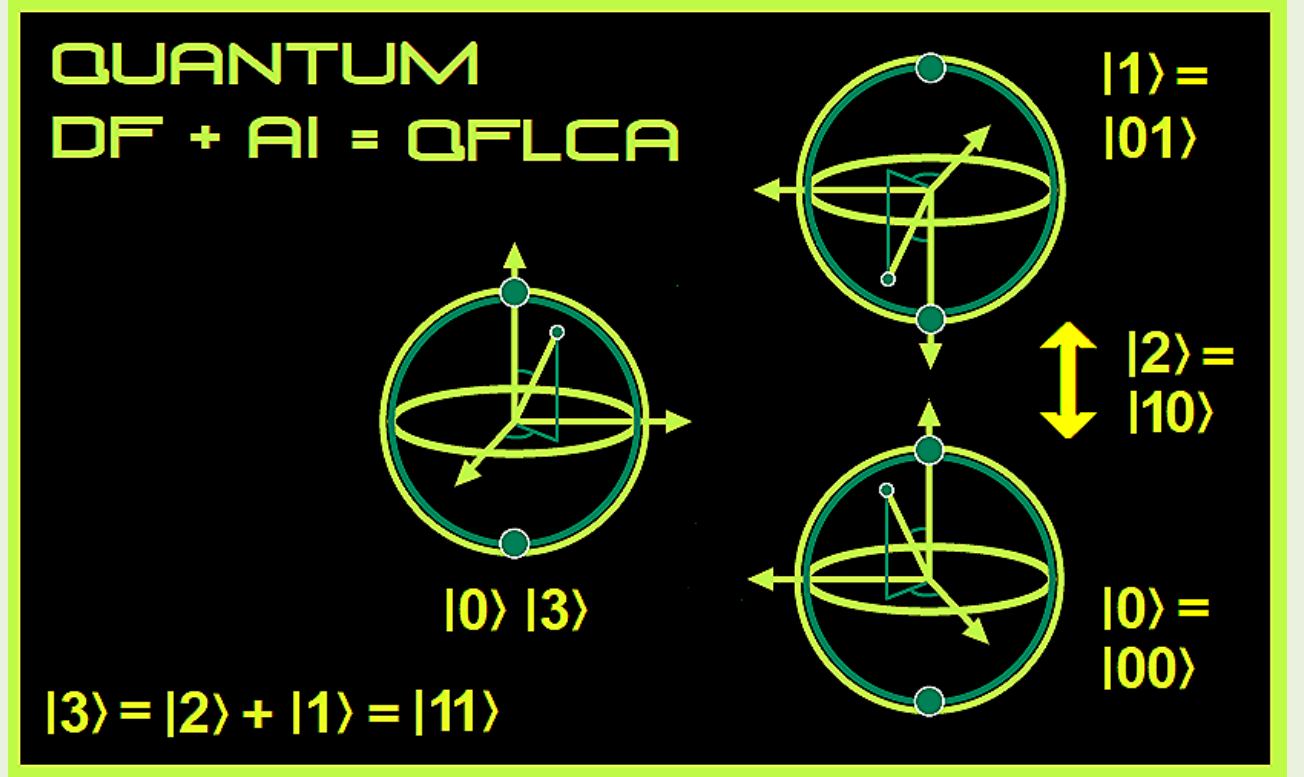
Step 2: Stamp sample as a particle and its field



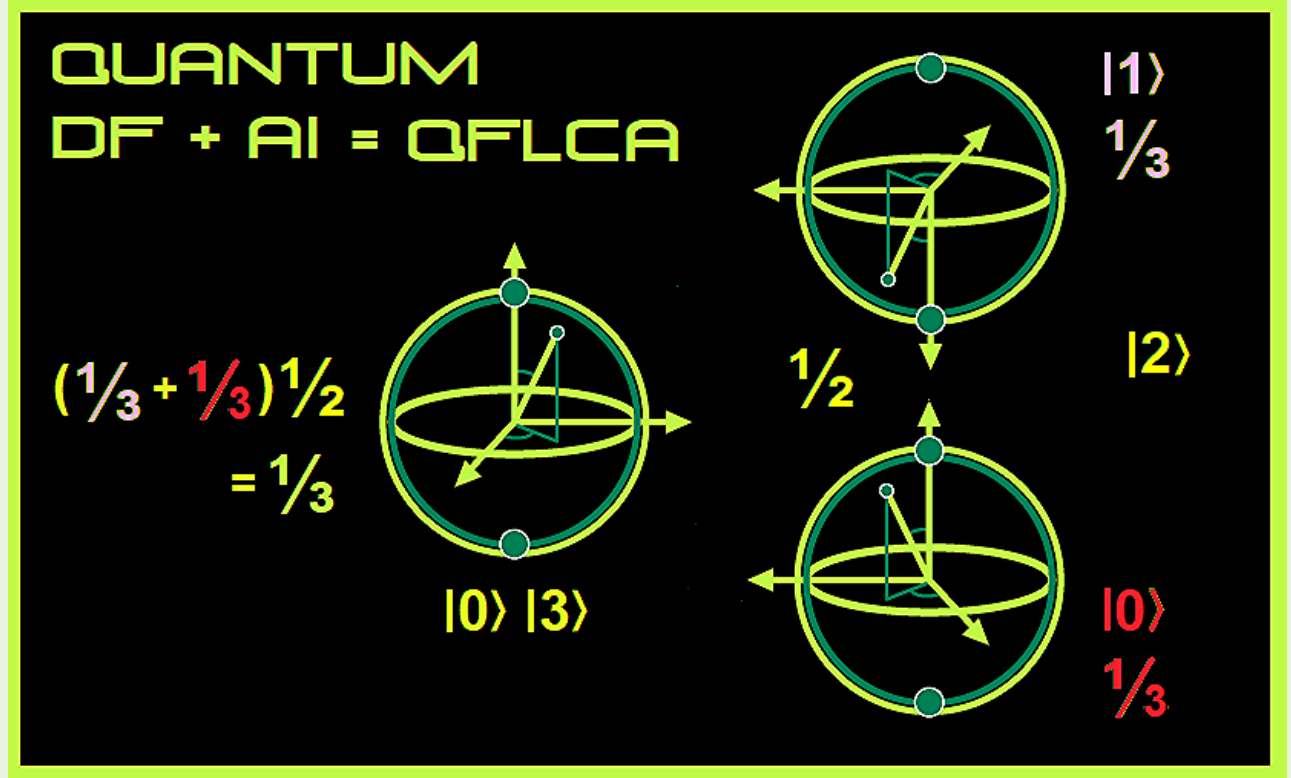
Step 3: Light source projecting photon to the particle field



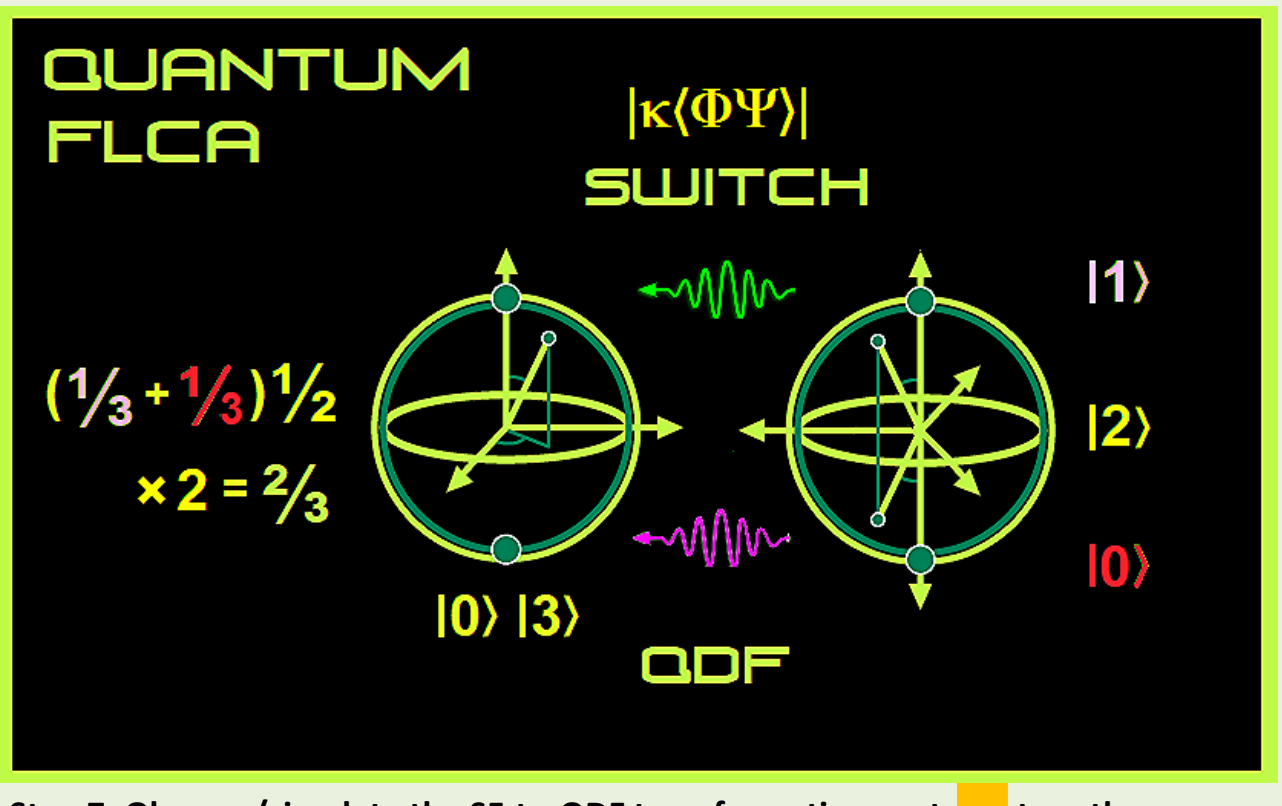
Step 4: Entangle the particle sample with another sample from a genuine stamp



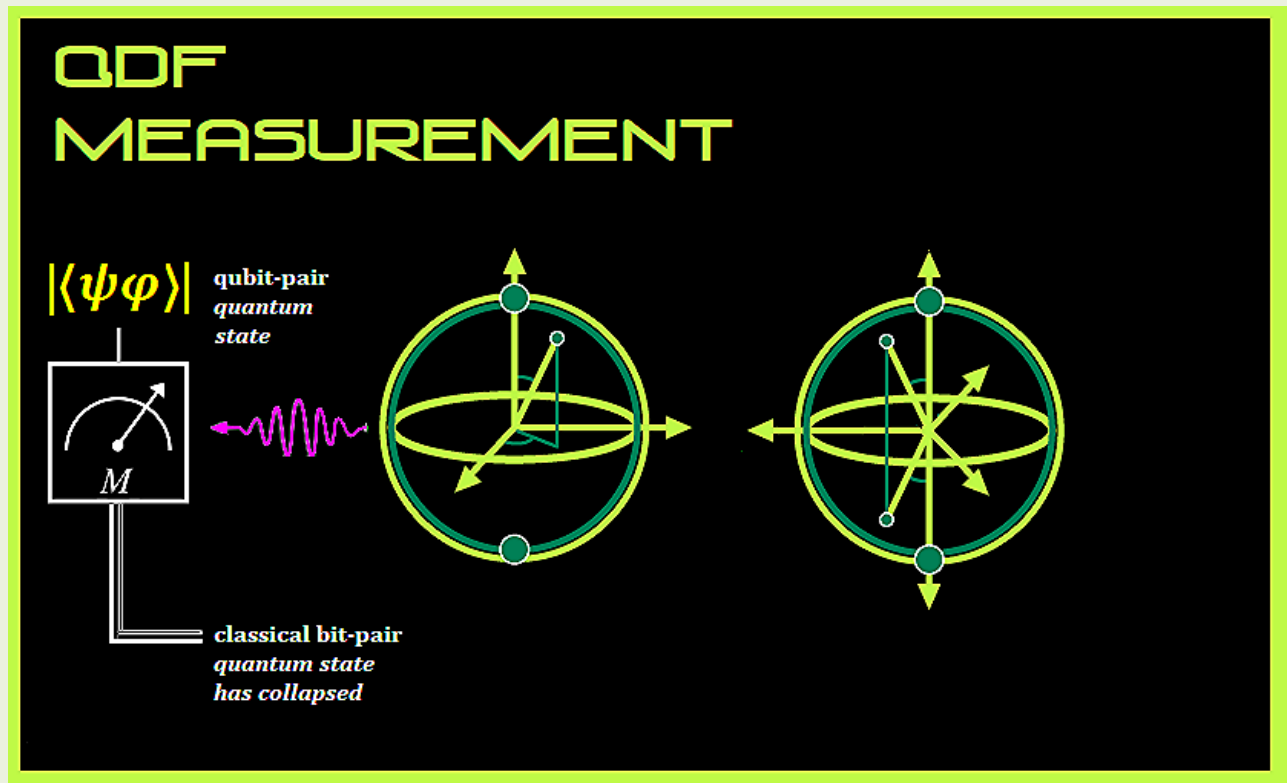
Step 5: Apply QF-LCA to measure particle states as they superpose or entangle



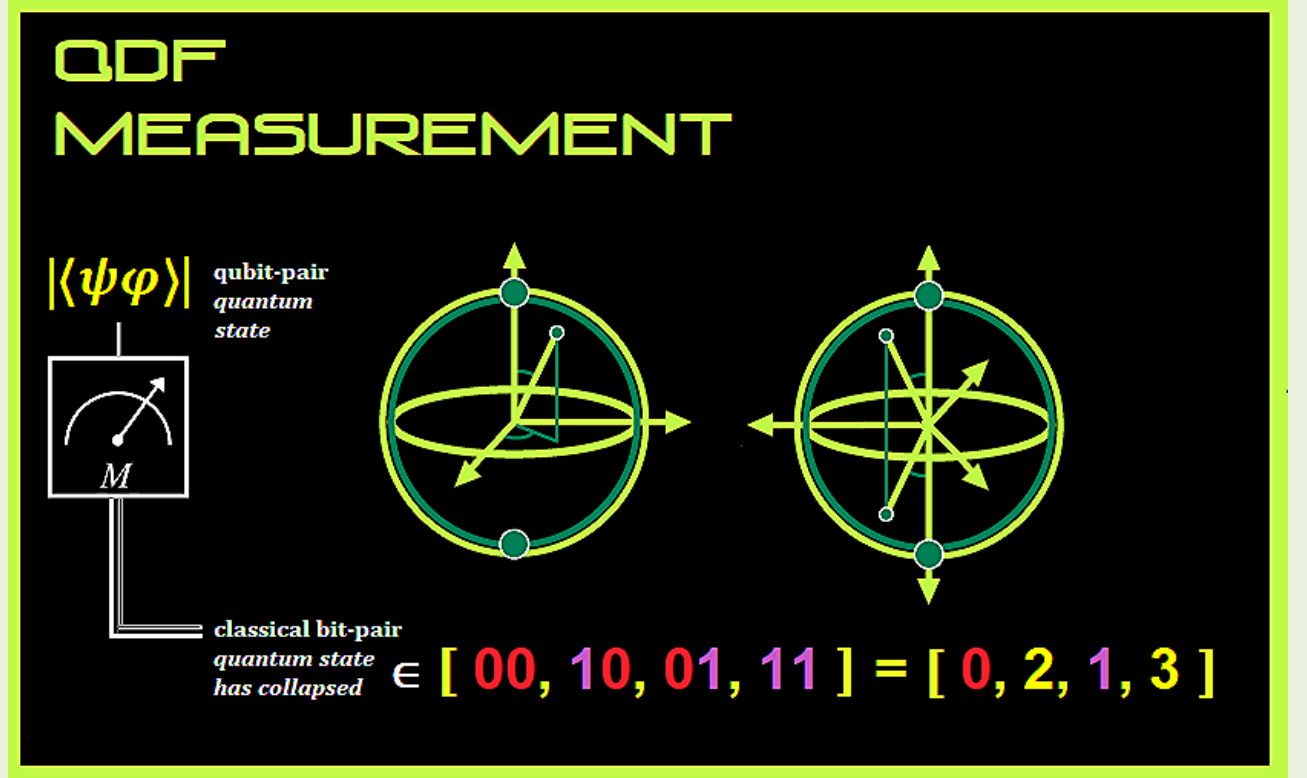
Step 6: Apply QF-LCA to make an SF-to-QDF transformation



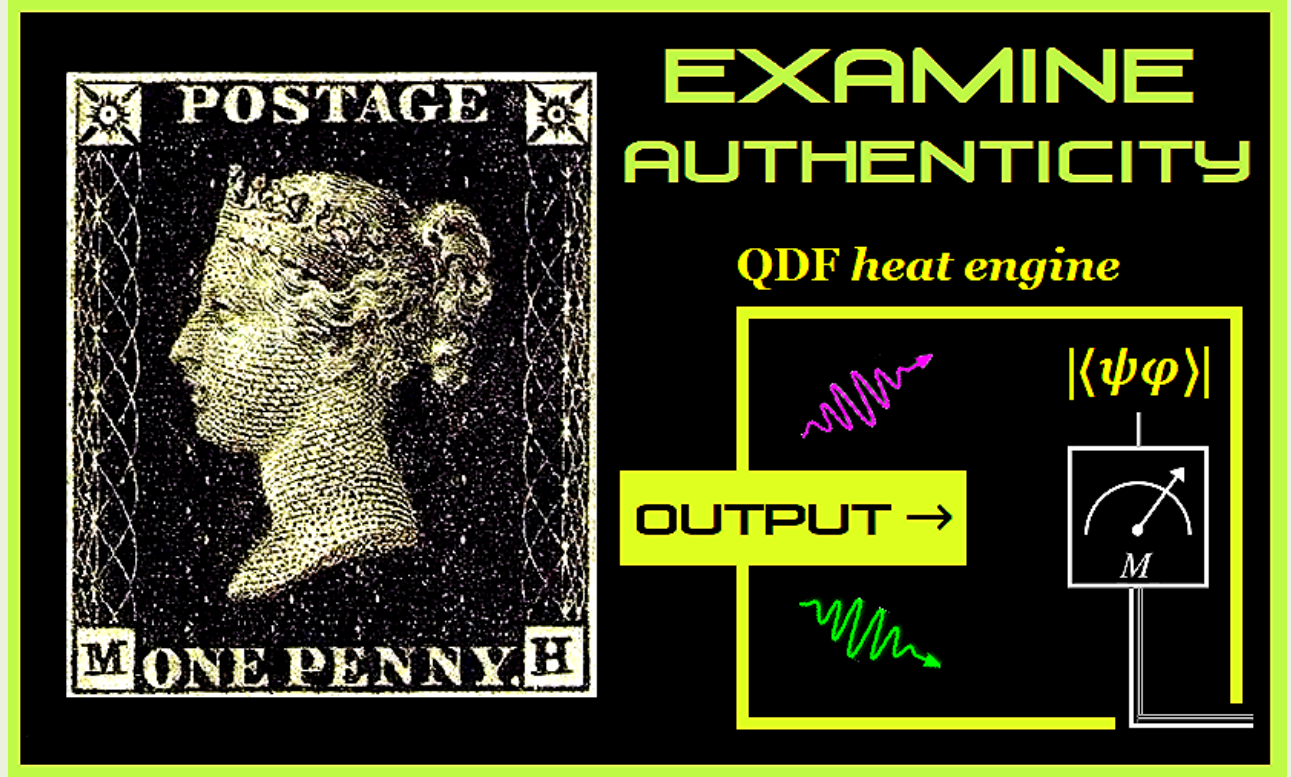
Step 7: Observe/simulate the SF-to-QDF transformation, so to restore the damaged stamp via nano-printing, or detect a forged stamp from genuine



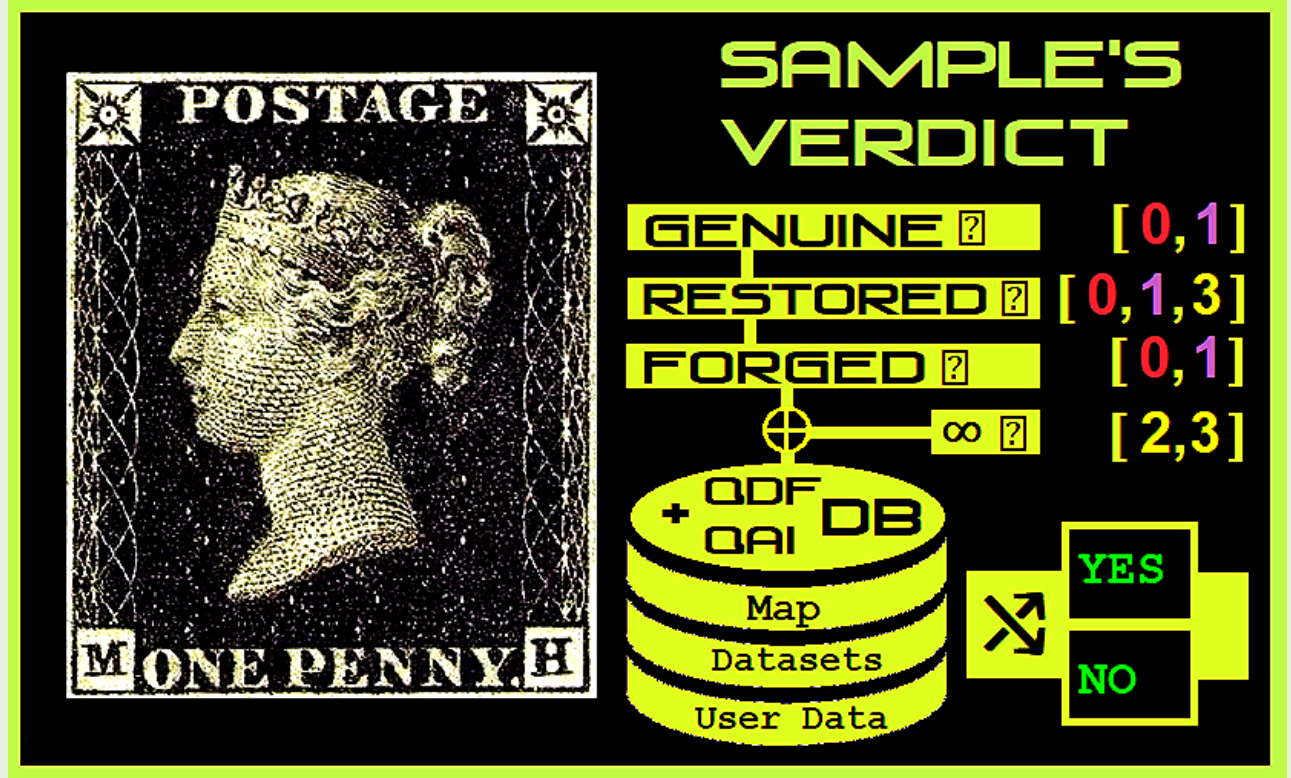
Step 8: Measure the qubit pair from the QDF as the output



Step 9: Register the expected QDF probability measured between the samples



Step 10: Store values from the simulated heat engine on a QDF_QAI DB



Step 11: Process the dataset from the DB and print the simulated output/results

HALT

Physical
+
Software OUTPUTS

OBSERVER OUTPUTS:

- A quantum (light-particle) heat engine sampled particles from a physical system
- The heat engine entangled sampled particles at ultracold/BEC temperatures
- Entanglement was registered as qubits to predict a system state by:
- A QDF was formed doubling particle space in probability to its position
- A QDF provided information from entangled particles exchanged in the system
- Particle spin and position predicted by sharing the QDF information as qubits
- Qubits counted in a QDF coding system predicting states with high probability
- A QDF extra qubit complemented the information on a hidden particle state
- QDF circuit found the hidden state relative to thermal events in the system

QAI Decision-making by QF-LCC within QF-LCA

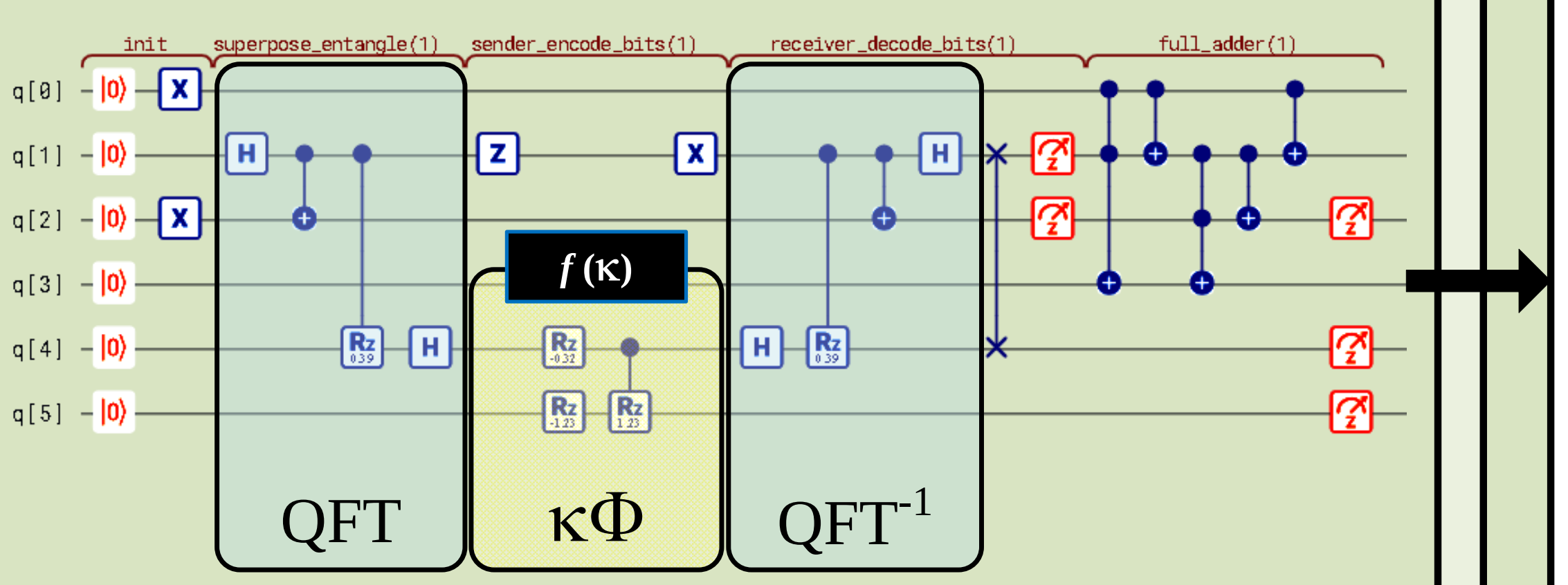
- Obtain efficient heating or cooling of particles by using information above by:
- create or reroute energy paths for those particles not participating in a thermal event to participate by focusing/defocusing the distribution of their states through QDF lenses in the system

SIMULATED QDF CIRCUIT

QDF circuit tested on *N*-qubit machines prior

Ref. [1, Lower Table 2]
Ref. [1, p.28]

QDF circuit's energy in/out via QDF scalar transform for $N \geq 3$ particles; QDF is compatible with QFT and its inverse QFT^{-1}per measurement

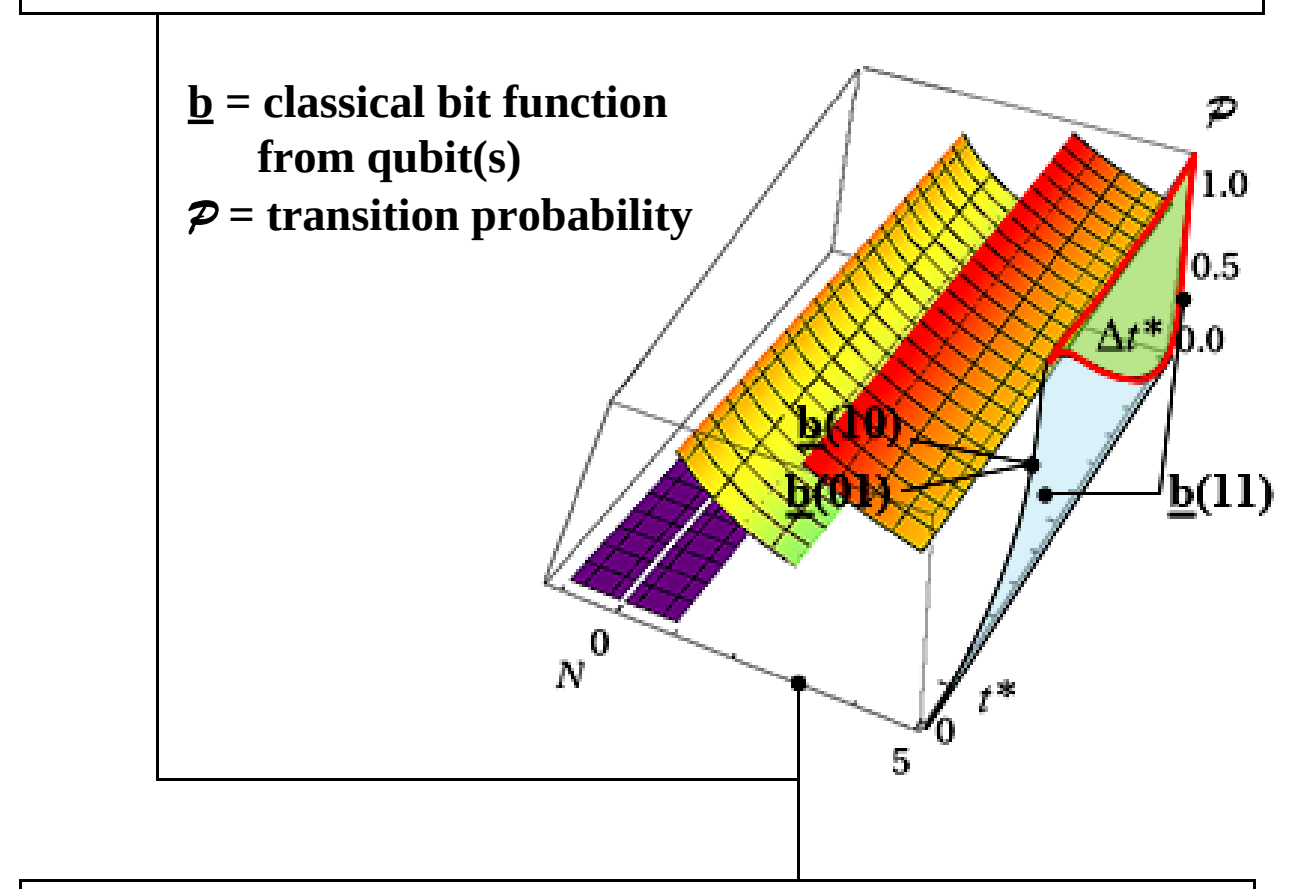


IN → Scalar κ function → OUT

SIMULATED QDF CIRCUIT'S DATASET

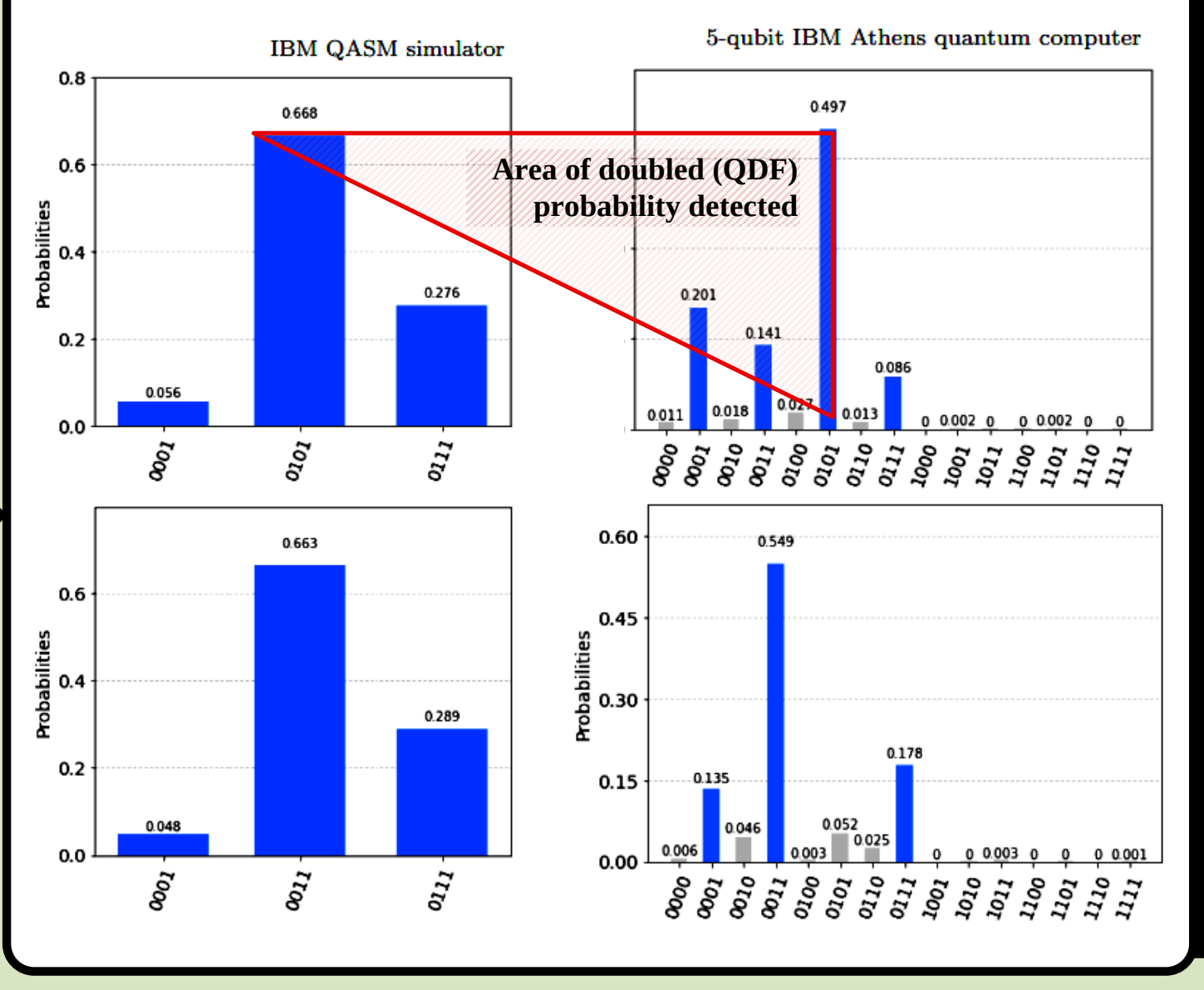
Circuit measurement outcome at the Encoding Level for $N \geq 3$ particles, predicting QDF $\mathcal{P}$'s

For $n \geq 3$, the expected outcome is b({01, 10}) with a $\mathcal{P} \geq 2/3$, and b(11) with a $\mathcal{P} \leq 1/3$ predicted for any case without a SWAP gate



For $n \geq 3$, the expected outcome is b(01) with a $\mathcal{P} \geq 2/3$, and b({10, 11}) with a $\mathcal{P} \leq 1/3$ predicted for any case with a SWAP gate

Decoding measurement outcome as code samples with their *p*'s for $N \geq 1$ particle pairs



Analyzed OUTPUTS by QF-LCA

Philip Baback Alipour,
Thomas Aaron Gulliver, 2024

DOI: 10.17632/gf2s8jkdjf
Published by
Elsevier B.V.
©2024

UNIVERSITY OF VICTORIA

ACRONYMS:

PT = phase transition
Q = quantum, **C** in **CPT** = classical
QDF = quantum double-field
QF-LCA = QDF lens coding algorithm
SF = single field
BEC = Bose-Einstein condensate

QAI = quantum artificial intelligence
QFT = quantum Fourier transform
QF-LCC = QDF lens coding classification
CNT = carbon nanotube
ES = excited state **GS** = ground state
DB = database d = QDF lens distance
w = weighted probability value/data point

CONCLUSION:

The QF-LCA when trained as a QAI algorithm makes strong predictions after the expected success probability values of $\langle \mathcal{P}_{\text{success}} \rangle \geq$ **2/3** from **1/3**, to values close to **1** probability, or near zero entropy as the system evolves in rerouting energy paths making particles to entangle, replicate, participate and contribute to greater system efficiencies, and/or field transform. Information retrieval and reconstruction of events based on entanglement entropy are examples from the QF-LCA dataset and software application.

START

Ref. [2, Sec. 4]

HEAT ENGINE PARTICLE SAMPLING EXAMPLE:

- Identify forged postal stamp from genuine or,
- Restore/transform the stamp