Coherence pressure test

Compressivity-Driven Interference Test — Protocol (v1.4)

0) One-line summary

Test whether a weak "coherence pressure" biases experimental histories toward more compressible logs by adaptively toggling an interferometer between interference (which-path erased) and which-path modes to minimize the description length of the *entire* event journal.

0a) Scope and non-superdeterminism

This protocol assumes no superdeterminism and no hidden access to future outcomes. All controller decisions must depend strictly on past data. The goal is not to challenge standard quantum mechanics but to operationalize a falsifiable prediction motivated by compression-based dynamics.

1) Hypothesis & prediction

H_0 (Standard QM / Control): Given a controller whose future settings depend only on *past* data (no look-ahead), the joint distribution of outcomes and settings equals the composition of the controller's policy with ordinary quantum probabilities. Therefore, the compressibility of the full log is fully explained by the policy; no extra bias appears.

H_1 (Coherence-pressure / SHG): Histories that reduce the *incremental description length* of the joint log (settings + detections) are slightly favored. Under a closed-loop "compressivity-driven" controller, the realized logs show a statistically significant *excess* compressibility beyond what the same policy would predict under H_0.

Primary prediction: In the compressivity-driven mode, the normalized compression ratio of the full log exceeds a pre-registered null distribution built from policy-matched Monte Carlo or permutation baselines.

2) Intuition (plain language)

Give the experiment choices between two lawful regimes:

- **Interference ON:** no which-path info; produces structured fringe patterns.
- **Interference OFF:** which-path marked; no fringes; outcomes look more i.i.d. across positions.

If a tiny “coherence pressure” exists, then a controller that always tries to *minimize the growth of description length* of the whole journal might experience a small extra tilt toward sequences that are easier to describe than chance alone would allow. If nothing special exists, you get exactly what your policy implies—no leftover effect.

---

3) Setup (hardware)

- **Source:** Heralded single photons (SPDC) or attenuated laser (classical baseline).
- **Interferometer:** Mach–Zehnder or double-slit with controllable which-path marking/erasure.
- **Fast switch:** Electro-optic or acousto-optic modulator to toggle ON/OFF within the inter-photon interval.
- **Detectors:** Single-photon detectors (e.g., SPADs/EMCCDs) with nanosecond timing.
- **Which-path module:** Polarization or path marker + eraser element (e.g., polarizers + QWPs + PBS).
- **Timing & control:** FPGA or real-time PC for timestamping and issuing switch commands.
- **Randomness sources:** Quantum RNG and presealed scripted sequences.
- **Environment:** Temperature and vibration control; beam-path stabilization.

---

4) Event journal (what is logged)

For each detection event k:

- Timestamp t_k.
- Controller mode $M_k \in \{ON, OFF\}$ applied just *before* emission.
- Detector channel / position X_k.
- Optional which-path tag.
- Session metadata.

Serialized into the *journal string* J using a fixed encoding.

---

5) Control regimes (all three are required)

1. **RANDOM:** M_k sampled i.i.d. from QRNG.
2. **SCRIPTED:** M_k follows a pre-generated compressible sequence.
3. **COMPRESSIVITY-DRIVEN:** After each block of N events, choose the next mode to minimize incremental description length, under guardrails:
4. Depend only on past data.
5. Fixed N.
6. Max switch rate.
7. Ties resolved via QRNG.

---

6) Compressivity metric (pre-register!)

Use two independent metrics:

- **C_1 (ZIP ratio):** $|\mathrm{ZIP}(J)| / |\mathrm{RAW}(J)|$.
- **C_2 (Lempel–Ziv complexity):** normalized LZ score.

Incremental objective: prefer settings that minimize $\Delta \mathrm{DL} \approx \Delta|\mathrm{ZIP}(J)|$ estimated from past data.

---

7) Procedure

1. Calibration.
2. Baseline runs: RANDOM & SCRIPTED.
3. Closed-loop COMPRESSIVITY-DRIVEN runs.
4. Classical-light baselines.
5. Blinding.

---

8) Analysis plan

Primary test:

- Build a *policy-matched null* by simulating outcomes under standard QM with the same realized mode sequence.
- Compare observed compressivity to null distribution.

Secondary:

- Compare regimes.
- Runs/KS tests.
- Sensitivity to encoding.
- Control for multiple comparisons.

---

9) Artefacts & suppression

- Differential losses.
- Controller periodicity.
- Drift.
- Time-correlation leaks.
- p-hacking.

---

10) Pre-registration checklist

- Hardware tolerances.
- N, switch rate, tie-breaks.
- Encoding and compressors.
- Metrics, thresholds.
- Null construction.
- Blinding.
- Replication.

11) Minimal controller pseudocode

```
while running:
    C_now = Compress(J)
    DL_ON  = PredictDeltaDL(J, mode='ON',  N)
    DL_OFF = PredictDeltaDL(J, mode='OFF', N)
    if abs(DL_ON - DL_OFF) < epsilon:
        next_mode = QRNG()
    else:
        next_mode = 'ON' if DL_ON < DL_OFF else 'OFF'
    ApplyMode(next_mode)
    AcquireBlock(N)
```

12) Variants

- Quantum eraser.
- Two-arm MZ.
- Bell-type with independent wings.

13) Expected outcomes

A deviation would not falsify QM but would indicate that closed-loop compressivity interacts with interference statistics in a nontrivial way.

13a) Relation to SHG (non-essential context)

This protocol does not assume the correctness of SHG. It merely tests whether experimental histories exhibit a small bias toward lower global description length under adaptive control. A positive signal would be consistent with models in which coherence-propagation dynamics exert weak compression pressure; a null result is fully compatible with standard QM.

- **Null:** Compressivity matches nulls.
- **Positive:** Excess compressibility across encodings/hardware; require replication and artefact checks.

Either way: produces an operational probe of whether reality “leans” toward compressible histories.

14) Practical notes

- Sufficient event counts.
- Keep ON/OFF losses \~1% or model.
- Version-control code.
- Publish raw & compressed logs.

15) Versioning

- **v**1.4 — First public draft.