

BCQM VII Lab Note

Turnover test: stationarity of traffic on a fixed Stage-2 cloth (v0.1)

Peter M. Ferguson
Independent Researcher

4 February 2026

Purpose

Test the “thread current” hypothesis without introducing any thread kill/spawn mechanism. Threads are assumed persistent; the question is whether, on an already-formed Stage-2 cloth, the *traffic/current* through that cloth is stationary or turns over as the run proceeds. Operationally: hold the Stage-2 cloth geometry fixed and compare early versus late time windows.

Background and motivation

A naive “window cloth” comparison (building the cloth independently inside each time window) can collapse the giant component into tiny recurrent motifs, because per-bin ledger edges are filtered and sparse. This behaviour is analogous to strict persistence acting as a motif detector rather than yielding a spanning substrate. Such a comparison does not probe Stage-2 cloth geometry.

The correct operational test is therefore a *fixed-geometry turnover* test:

- Fix the cloth core using the Stage-2 object stored by the engine in each RUN_METRICS file (the same core used for Gate 0–3 results in BCQM VII).
- Compare window A versus window B activity on that fixed substrate.

Run and configuration

This lab note documents the fixed-geometry turnover test on the headline Stage-2 run:

- $N = 32$, $n = 0.8$, $W_{coh} = 100$, bins=20, hits1 baseline.
- Windows: A = bins 0–9 and B = bins 10–19.
- Fixed geometry: `cloth.core_edges_used` and `cloth.core_events_used` from RUN_METRICS (per seed).
- Activity source: `edges_all` from `cloth_ledger`.

Scripts. Three generations of the analysis were developed:

- v1: window-compare rebuilt geometry within each window (diagnostic: tends to motif cores).
- v2: fixed-geometry compare, but fixed geometry was rebuilt from ledger edges (still vulnerable to sparsity).
- v3 (this note): fixed-geometry compare using stored Stage-2 core lists, and *dense* similarity metrics (zeros included) to avoid NaNs when support differs.

Files. Inputs:

- `outputs_cloth/.../RUN_METRICS_*.json` containing `cloth.core_edges_used`, `cloth.core_events_us` and `cloth_ledger`.

Outputs:

- `csv/turnover/N32_n0p8_edgesall_fixedcore_A0_9_B10_19_v3_fixedgeom_v3_summar.csv`
- (per-seed table is analogous: `...fixedgeom_v3_runs.csv`)

Method (v3)

Let E_{core} be the fixed Stage–2 core edge set and π the fixed community partition on the undirected core graph (Louvain, resolution $\gamma = 1.0$, deterministic algorithm seed).

For each window $W \in \{A, B\}$, compute window edge weights on the fixed core:

$$w_W(e) = \sum_{\text{bins in } W} c_e \mathbb{1}[e \in E_{\text{core}}], \quad e \in E_{\text{core}},$$

where c_e is the ledger multiplicity per bin. From these weights we form *dense* vectors $\mathbf{w}_A, \mathbf{w}_B \in \mathbb{R}^{|E_{\text{core}}|}$ including zeros.

We report dense similarity metrics on edges:

- Pearson correlation on the full vectors (zeros included),
- cosine similarity,
- Jensen–Shannon distance (JSD; 0 identical, 1 maximally different),
- total variation distance (TV; 0 identical, 1 maximally different).

We also compute community-pair “flows” by aggregating edge weights between community labels and compare dense K^2 vectors of (i, j) flow weights (again including zeros) using the same metrics.

Finally, we report *coverage*:

$$\text{cov}_W = \frac{\sum_{e \in E_{\text{core}}} w_W(e)}{\sum_{e \in \text{ledger}} w_W(e)},$$

i.e. the fraction of ledger activity weight in the window that lands on fixed core edges.

Results

Quantity	Value (mean \pm sd across 5 seeds)
Fixed core nodes	$16,783 \pm 1,169$
Fixed core edges	$18,724 \pm 1,493$
Core fraction Φ	0.1820 ± 0.0125
Communities K	68.00 ± 2.12
Core coverage (A)	0.0777 ± 0.0229
Core coverage (B)	0.0820 ± 0.0104
Window total weight (A)	796.6 ± 49.9
Window total weight (B)	945.8 ± 52.8
Window core weight (A)	61.6 ± 17.9
Window core weight (B)	77.2 ± 6.6

Table 1: Fixed Stage–2 cloth geometry and window-level activity summaries (A: bins 0–9; B: bins 10–19).

Quantity	Value (mean \pm sd across 5 seeds)
Active-edge Jaccard	0.000000 ± 0.000000
Dense Pearson corr	-0.001830 ± 0.000226
Cosine similarity	0.000000 ± 0.000000
JSD distance	0.832555 ± 0.000000
Total variation	1.000000 ± 0.000000

Table 2: Window A versus window B similarity metrics on fixed core edges (support overlap and dense comparisons).

Quantity	Value (mean \pm sd across 5 seeds)
Active-flow Jaccard	0.004762 ± 0.010648
Dense Pearson corr	0.002569 ± 0.013433
Cosine similarity	0.006000 ± 0.013416
JSD distance	0.829082 ± 0.007764
Total variation	0.994444 ± 0.012423

Table 3: Window A versus window B similarity metrics on fixed community-pair flows (K^2 dense vector, zeros included).

Interpretation

Geometry is fixed; traffic turns over. By construction the fixed geometry (core size, Φ , and K) is stable and matches the Stage–2 cloth object used in BCQM VII. The window-to-window comparisons instead probe whether the *current* through this fixed substrate is stationary.

Near-maximal separation on edges. The edge-level dense metrics are consistent with a near-disjoint-support regime over the chosen 10-bin windows: active-edge Jaccard is 0.0, cosine similarity is 0.0, and TV distance is 1.0. The JSD distance is ≈ 0.833 , numerically close to the maximal-separation value $\sqrt{\ln 2}$ expected when two normalised distributions have negligible overlap. Operationally, this indicates that the set of core edges carrying any weight in window A is almost entirely different from the set carrying weight in window B.

Flows also turn over. Community-pair flow metrics show the same picture: very small active-flow overlap and near-maximal TV and JSD distances. This indicates turnover not only in individual edges but also in which community-to-community routes are active.

Most activity weight lies outside the fixed core. Core coverage is low (about 8%), meaning that the ledger weight in these windows is dominated by non-core (halo or non-core) edges. Total activity weight is higher in window B than window A, but this does not correspond to increased reuse of core edges within these short windows.

Important nuance: “turnover” versus “sparse sampling”. The near-zero overlap metrics do not by themselves prove that the underlying stationary distribution of activity has changed. They do prove that, at the present window sizes, observed activity on the fixed core is extremely sparse relative to $|E_{\text{core}}|$, so two disjoint windows can sample disjoint supports. Distinguishing “true drift” from “stationary but sparsely sampled” requires longer windows or heavy-hitter analyses (see Next steps).

Conclusions and implications

What we can say now. On the fixed Stage–2 cloth core, the *observed current* (edge and flow activity) has strong turnover between early and late 10-bin windows: overlap measures are near zero and dense distribution distances are near maximal. The cloth geometry can remain stable while the traffic through it is highly non-stationary at the microscopic support level.

Programme placement. This is best treated as a Stage–3 direction: “persistent geometry, time-dependent excitation”. It is not necessary for the BCQM VII Stage–2 claim (existence of a stable spatial cloth), but it provides a clear pointer for the next stage: quantify a turnover timescale and test whether any coarse-grained traffic patterns become stationary under longer averaging.

Recommendation for BCQM VII. Keep these turnover results out of the main BCQM VII text for now. At most, include a one-sentence teaser in the Stage–3 outlook: “on a fixed cloth core, edge/flow activity can turn over rapidly between time windows, suggesting a separation between stable geometry and drifting excitation.”

Next steps (Stage–3)

- **Longer windows / sliding windows:** repeat the analysis with wider A/B windows and/or a sliding window to estimate a turnover timescale for traffic on the fixed cloth.
- **Heavy-hitter overlap:** restrict to the top k edges (or top $p\%$) by weight in each window and compare overlap and distance; this probes whether there is a persistent “preferred-route” subset.
- **Core versus halo traffic:** repeat with $\text{edge}_{\text{source}} = \text{core}$ and/or explicitly quantify activity partitioned into core versus halo edges.
- **Null calibration:** compare to a null in which window weights are randomly permuted over the fixed core edges to calibrate how much turnover to expect under a stationary distribution with sparse sampling.