Matrix Node Theory: Quantum Fluctuation Test Addendum

Julius.ai Analysis Prompt:

```
# Julius.ai analysis prompt
# 1) Load event catalog (choose one):
#    • CERN Open Data (Z→μμ / H→γγ)
#    • LIGO Open Science Center (O1–O3)

import requests, numpy as np
resp = requests.get("https://www.gw-openscience.org/eventapi/json/")
events = resp.json()
times = np.array([e['GPS'] for e in events['events']])
m1 = np.array([e['m1_source'] for e in events['events']])
m2 = np.array([e['m2_source'] for e in events['events']])
m_eff = (m1 + m2) / 2
m0 = 30.0  # base mass in M■
n = np.round(m_eff / m0).astype(int)

# 2) Define reference tick and phase
T0 = times.min()
tau_q = 1.0  # seconds per tick
phi = 2*np.pi * ((times - T0) % tau_q) / tau_q

# 3) Histograms per mode and Rayleigh test
from scipy.stats import rayleigh
for mode in sorted(set(n)):
    mask = (n == mode)
    phi_mode = phi[mask]
    if len(phi_mode) < 5: continue
    hist, edges = np.histogram(phi_mode, bins=36, range=(0,2*np.pi))
    stat, pval = rayleigh(phi_mode)
    print(f"Mode {mode}: Rayleigh p-value = {pval:.3f}")
```

Section 7.4: Quantum Fluctuation Lexicon Test

Definitions & Calibration

1. Fundamental tick: τ_q (e.g., 1e-15 s).
2. Reference mode: $f_0 = c/a_0 \approx 1.85\times10^{43}$ Hz.
3. Measured phase: $\phi_{meas}(t)$ from timestamps.
4. Offset calibration: $\phi_{rel} = \phi_{meas} - \Delta\phi_j$.

Fluctuation Function & Resonance Peaks

$F(t,n) = N_c n^2 + \delta \sin[n \phi_{rel}(t)]$, $n = f/f_0$.
Scan frequencies f_n and compute F at each tick.
Locate collapse peaks where $n \phi_{rel} \approx m\pi + \theta_n$.

Test Protocol

1. Acquire CERN Open Data or LIGO JSON.
2. Compute ϕ_{rel} for each event: $\phi = 2\pi((t - T0)\%\tau_q)/\tau_q$.
3. Assign mode index n = round(m_eff/m0).
4. Histogram ϕ_{rel} per mode and test clustering.
5. Flag modes with $p < 0.05$ for follow-up.