# Echo Bump Detectability — 5σ Significance & Mode Counts

This note records the quick-significance checks we just ran for the Genesis Echo bump in the CMB. We used the ≥5σ criterion to relate signal amplitude A, per‑mode noise σ\_n, and the effective number of independent modes N. Two equivalent forms are:

N = (5 σ\_n / A)^2

A\_min = 5 σ\_n / √N

## 1) Worked examples (σ\_n = 1)

A = 0.05 → N ≈ 10,000 modes for ≥5σ

A = 0.10 → N ≈ 2,500 modes for ≥5σ

## 2) Mode-count estimates for realistic CMB configurations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset / config | f\_sky | l\_max | Effective modes N | 5σ detectable amplitude A\_min (σ\_n=1) |
| Planck-like TT (f\_sky=0.6, l\_max=2500) | 0.6 | 2500 | 3753001 | 0.002581 |
| ACT/SPT deep combo (f\_sky=0.4, l\_max=4000) | 0.4 | 4000 | 6403200 | 0.001976 |
| CMB-S4 forecast (f\_sky=0.5, l\_max=5000) | 0.5 | 5000 | 12505000 | 0.001414 |
| Lensing recon L≤3000 (f\_sky=0.4) | 0.4 | 3000 | 3602400 | 0.002634 |

Interpretation: your prior echo amplitudes A ≈ 0.05–0.10 are well above the 5σ thresholds for datasets with O(10^6–10^7) modes, assuming σ\_n is order‑unity per mode. If σ\_n differs, scale A\_min linearly by σ\_n.

## 3) Python used for the calculation

# Example: required modes for a target amplitude A (≥5σ)  
def required\_modes(A, sigma\_n=1.0):  
 return (5 \* sigma\_n / A) \*\* 2  
  
for A in [0.05, 0.10]:  
 print(A, required\_modes(A)) # -> 10000, 2500 (for σ\_n=1)

# Effective mode counts and 5σ thresholds for CMB-like configs  
import math  
scenarios = {  
 "Planck-like TT (f\_sky=0.6, l\_max=2500)": {"f\_sky": 0.60, "lmax": 2500},  
 "ACT/SPT deep combo (f\_sky=0.4, l\_max=4000)": {"f\_sky": 0.40, "lmax": 4000},  
 "CMB-S4 forecast (f\_sky=0.5, l\_max=5000)": {"f\_sky": 0.50, "lmax": 5000},  
 "Lensing recon L≤3000 (f\_sky=0.4)": {"f\_sky": 0.40, "lmax": 3000},  
}  
sigma\_n = 1.0  
for name, s in scenarios.items():  
 N = s["f\_sky"] \* (s["lmax"] + 1)\*\*2 # ≈ f\_sky \* Σ (2ℓ+1)  
 A\_min = 5 \* sigma\_n / math.sqrt(N) # ≥5σ threshold  
 print(name, "N≈", int(round(N)), "A\_min≈", A\_min)

Note: N ≈ f\_sky × (ℓ\_max+1)^2 is a standard approximation to the sum of (2ℓ+1) modes from ℓ=2..ℓ\_max. For precise forecasts, plug your exact noise curves and masks.