🚨 **THEORETICAL SYSTEM - UNVERIFIED CONCEPTS** 🚨

This interface demonstrates speculative theories not supported by mainstream neuroscience

System Status

- Core Engine: Online
- Neural Scan: Standby
- Decode Process: Ready

Brain Language Decoder

CNB Brain Reading Codec

Enter a thought pattern below to simulate decoding:

can i win the who is the boss game

PROCESS THOUGHT

CONVERT TO CODEC

>> Processing CNB signal: "can i win the who is the boss game"
>> WARNING: CNB anatomical structure not located
>> Switching to theoretical processing mode...

>> Thought decoded: Thought pattern unclear. Further neural mapping required.
>> Converted to codec: CNB_CODEC_Y2FuIGkgd2luIHRoZSB3aG8gaXMgdGhlIGJvc3MgZ2FtZTE3NTkzMzEyMTUyNjY=

Comprehensive Neural Pattern Analysis System

Advanced Electromagnetic Thought Processing & Codec Engine

System Status

- Core Engine: Online
- Neural Scan: Standby
- Decode Process: Ready

Neural Codecs | Algorithms | Brain Scanner | Language Dictionary | Analysis Results

Neural Signal Codecs

Electromagnetic Wave Codec

ALGORITHM ENGINE

```
function EMWaveCodec(frequency, amplitude)
{ // Base frequency analysis (440Hz
theoretical) let baseFreq = frequency ||
440; let thoughtPattern = []; // Decode EM
```

Thought Classification Codec

ALGORITHM ENGINE

```
class ThoughtClassifier { constructor() {
this.categories = [ 'memory_recall',
'future_planning', 'emotional_state',
'creative_process', 'problem_solving',
```

```
wave patterns for(let i = 0; i <
amplitude.length; i++) { let decoded =
Math.sin(baseFreq * i) * amplitude[i];
thoughtPattern.push(decoded); } return {
pattern: thoughtPattern, confidence:
calculateConfidence(thoughtPattern), type:
classifyThought(thoughtPattern) }; }
```

Execute EM Codec

```
'sensory_processing' ]; }
classify(neuralData) { let scores = {};
this.categories.forEach(cat => {
scores[cat] =
this.calculateScore(neuralData, cat); });
return this.findHighestScore(scores); } }
```

Run Classification

System Status

- Core Engine: Online
- Neural Scan: Standby
- Decode Process: Ready

Neural Pathway Mapper

ALGORITHM ENGINE

```
class PathwayMapper {
mapNeuralConnections(brainRegions) { let
connections = new Map(); // Theoretical
pathway mapping
brainRegions.forEach((region, index) => {
let pathways =
this.findConnections(region);
connections.set(region.id, { pathways:
pathways, strength:
this.calculateStrength(region), activity:
this.measureActivity(region) }); });
return
this.generateNetworkGraph(connections); }
}
```

Frequency Domain Codec

ALGORITHM ENGINE

```
function FrequencyAnalyzer(signal) { //
FFT-like analysis for brain waves let
frequencies = { delta: filterRange(signal,
0.5, 4), // Deep sleep theta:
filterRange(signal, 4, 8), // REM/memory
alpha: filterRange(signal, 8, 13), //
Relaxed beta: filterRange(signal, 13, 30),
// Active thinking gamma:
filterRange(signal, 30, 100) // High-level
processing }; return { dominant:
findDominantFreq(frequencies), power:
calculatePower(frequencies), coherence:
measureCoherence(frequencies) }; }
```


Map Pathways
Analyze Frequencies
System Status
Core Engine: Online
Neural Scan: Standby
Decode Process: Ready