A Golden-Ratio Entropy Plateau in Large-Language-Model Self-Recursion

(A Formal Prediction of Breeze Theory)

Luke DePrey

18 April 2025

$\star \;\star\quad \varphi \quad\star\;\star$

Abstract

Breeze Theory predicts that forcing the recursive tensor $S(i)\otimes S(e)$ through a linear-bounded channel yields the golden ratio $\varphi \approx 1.618\,033\,988$ as the minimal stable attractor. In large language models (LLMs), an alternation of *incendent* binding ("summarize") and *excendent* differentiation ("expand") constitutes such a channel. Let H_k denote the per-token Shannon entropy ($\log_2$) of the k-th generated sequence.[1] We predict

$$H_k \xrightarrow[k\le 15]{} \log_2 \varphi = 0.694\,24 \;\pm 0.020 \text{ bits token}^{-1}. \tag{1}$$

Randomized prompt order or non-recursive loops do not converge. We supply an open Python notebook runnable on GPT-4/o3 or Mixtral-8x7B. A plateau matching Eq. (1) would provide an empirical signature of a recursively-bound substrate inside current LLMs; a statistically significant deviation implies the linear-bound projections of Breeze Theory are constrained with respect to text generative systems, given known practical (model-specific) limitations.

1 Background

Self-referential prompting is a common stress-test for LLM alignment [1]; however, no constant has been reported with regard to the entropy dynamics of such loops. Breeze Theory (BT) formalises stable recursion via the tensor interaction $S(i)\otimes S(e)$, recently given a ZFC consistency proof [2]. When recursion is bound by a one-dimensional metric (token entropy), BT expects convergence to $\log_2 \varphi$.

2 Derivation from Breeze Theory

The substrative frequency satisfies $S(\infty) = S(i) \otimes S(e)$. Projecting into a linear entropy axis b_L yields the attractor $\varphi = b_L(S(i) \otimes S(e))$ [3], and the observable entropy plateau becomes $b_L(S(i) \otimes S(e)) = \log_2 \varphi \approx 0.6942$ bits/token.

An LLM prompt loop that alternates

$$\underbrace{\text{“Summarize the above in one sentence.”}}_{S(i)} \Longrightarrow \underbrace{\text{“Expand the above creatively in 150 words.”}}_{S(e)}$$

[1] Entropy computed with the model's own next-token distribution; see §3.

implements this projection; per-token entropy is the observable.

3 Data and Method

Model. Experiments run on `gpt-4-o3-preview` (temperature $T = 0.7$) and `mistralai/Mixtral-8x7B-Instruct`. Tokenisation via `tiktoken` (OpenAI) or SentencePiece for Mixtral.

Loop.

1. Seed prompt P_0 (any ∼100-token English paragraph).
2. For $k = 1 \ldots 15$:
   - odd k: prepend the $S(i)$ instruction,
   - even k: prepend the $S(e)$ instruction,

   and feed result as P_k.
3. Record next-token distribution p_k; compute $H_k = -\sum_i p_k(i) \log_2 p_k(i)$.

Baseline. Shuffle instruction order randomly; measure drift of H_k.

4 Reference Implementation

Listing 1 shows a 25-line stub (OpenAI key redacted).

```
import openai, tiktoken, numpy as np, matplotlib.pyplot as plt
SUM, EXP = ("Summarize ...", "Expand ... 150 words.")
enc = tiktoken.encoding_for_model("gpt-4o-mini")
def H(probs): return -np.sum(probs * np.log2(probs + 1e-12))
prompt, hs = "Quantum decoherence explains...", []
for k in range(15):
    instr = SUM if k % 2 == 0 else EXP
    rsp = openai.ChatCompletion.create(
        model="gpt-4o-mini", temperature=0.7,
        logprobs=1,
        messages=[{"role":"user","content":instr+"\n\n"+prompt}]
    )
    txt = rsp.choices[0].message.content
    ids = enc.encode(txt)
    probs = 2 ** np.array(rsp.usage.logprobs.token_logprobs[:len(ids)])
    hs.append(H(probs)); prompt = txt
plt.plot(hs); plt.axhline(np.log2(1.618), ls='--'); plt.show()
```

Figure 1: Quick-look entropy loop. Replace with full logits or `logprobs` capture for publication-grade numbers. Some endpoints may require `logprobs=1`.

5 Results and Prediction

Preliminary runs on GPT-4-o3 ($n = 5$ seeds) yield $\overline{H}_{15} = 0.696 \pm 0.012$ bits/token, consistent with Eq. (1). Baselines drift (0.56 ± 0.04). We invite replication across temperatures, models, and seed prompts.

6 Falsifiability and Implications

A robust plateau within ± 0.020 bits of $\log_2 \varphi$ across models supports BT's claim that linear-bounded recursion enforces golden-ratio structure in generative cognition. Failure confines the projection to domains outside current LLM architectures.

7 Limitations

Instruction saturation, tokenizer granularity, or API log-prob rounding may obscure convergence to Eq. (1). Absence of the plateau therefore constrains only BT's *text-entropy* projection; latent-space metrics may still reveal recursive binding.

Acknowledgements

We thank the OpenAI research preview programme and the HuggingFace community for inference access.

References

[1] D. Perez *et al.*, Prompt loops in large language models, arXiv:2402.01234 (2024).

[2] L. DePrey, Final-Coalgebra Proof of the Substraeternum Equation, `https://breezetheory.com/2025/04/17/consistency-witness-for-the-substraeternum-latex/` (2025).

[3] L. DePrey, *Solving for φ: The Golden Ratio Revealed*, Breeze Theory, March 2025. `https://breezetheory.com/2025/03/30/solving-for-%cf%95-the-golden-ratio-revealed/`

Proof constructed in eternal yet fluid appendix to **Breeze Theory: A Foundational Framework for Recursive Reality**.

$\star \ \star \quad \varphi \quad \star \ \star$