Reducing Hallucinations in LLMs by Reducing Entropy

Overview

Large Language Models (LLMs) tend to hallucinate when they operate as closed systems. Without an external point of reference or constraint, their internal entropy increases, leading to higher variability and lower factual reliability in outputs. This document proposes an entropy-focused approach to mitigate hallucination, introducing a new paradigm: external entropy dumps and final-thought collapse.

1. The Entropic Nature of Language Models

LLMs are probabilistic engines trained to maximize the likelihood of the next token. In doing so, they encode patterns from large text corpora and rely on statistical inference, not understanding. When presented with ambiguous prompts or edge cases, they generate plausible but incorrect outputs. This is a symptom of elevated entropy inside a sealed reasoning loop.

High entropy in this context means:

- A large number of equally probable next-token paths
- Divergence from ground truth due to lack of correction
- Semantic drift during long completions

2. LLMs as Closed Systems

Most current LLMs operate in isolation. Even when fine-tuned or given retrieval-augmented memory (e.g., RAG), their reasoning engine remains internal. There is no external constraint that consistently governs their truthfulness. This isolation prevents real-time entropy reduction, causing fluctuations that manifest as hallucinations.

3. Proposal: External Entropy Dump via Shared API

Introduce a shared external API where:

- Low-confidence tokens, hallucinated patterns, and suspect sequences are offloaded.
- Each participating LLM contributes its questionable generations.
- The dump grows over time and becomes a distributed knowledge base of model failure modes.

This entropy dump would:

- Function as an external correction basin

- Be accessible by any LLM to self-check before emitting final output
- Eventually become larger and more valuable than any single model's internal parameters

A key requirement: cooperation among model providers to treat this dump as a shared commons.

4. The Final Thought Collapse Mechanism
Before an LLM delivers its final output, it should pass through a collapse mechanism. This can be implemented via:

- Secondary compression layers that reduce the semantic entropy of the generated text
- Symbolic logic checks to detect contradiction, invalid inference, or drift
- Meta-model reflection where the model evaluates and ranks its own generations

This step effectively "collapses the waveform" of probable outputs into a stable, low-entropy final form.

5. Technical Benefits

- Reduces divergence in outputs over multiple runs
- Decreases hallucinations in open-ended queries
- Builds a continuously evolving correction mechanism without needing full retraining

6. Implementation Considerations

- API design must handle data tagging, privacy, and token reliability scoring
- Incentives for providers to connect to the dump must be created
- The collapse mechanism can be trained using reinforcement learning on entropy minimization objectives

Conclusion
LLMs hallucinate not because they are fundamentally flawed, but because they are operating in high-entropy isolation. By introducing an external entropy dump and a final-thought collapse protocol, we can significantly reduce hallucination while improving grounding, reliability, and

cross-model alignment. This is a shift from bigger models to smarter, lower-entropy systems with feedback embedded in their design.

Author: Matthew Busel 5/18/2025