This is why LLMs hallucinate

Vishal Singh Baraiya

Indian Institute Technology Madras, Chennai, TN, India

September 22, 2025

Abstract

Large Language Models (LLMs) are undermined by their propensity to "hallucinate": generating plausible but factually incorrect statements. This paper presents a unified theory arguing that hallucination is a two-stage problem, originating from a statistical learning flaw and reinforced by a socio-technical evaluation flaw. First, we posit that the initial errors arise from *local overfitting*, a phenomenon where a model's optimization process disproportionately specializes in high-frequency regions of the training data at the expense of sparsely represented knowledge [2]. This specialization leads to "knowledge forgetting," where the model discards correct patterns for rare concepts. Second, we argue that these initial errors are systematically shaped into confident hallucinations because prevailing training and evaluation procedures reward guessing over admitting uncertainty [1]. After rigorously proving this causal chain with a mathematical toy model, we introduce a novel, constructive contribution: **Frequency-Aware Regularization (FAR)**, a principled modification to the standard training objective designed to counteract knowledge forgetting. We provide a mathematical proof that FAR preserves knowledge of rare facts within our toy model, offering a promising direction for building more reliable LLMs.

Contents

1 Introduction

Large Language Models (LLMs) have demonstrated extraordinary capabilities, yet their advancement is critically hampered by a persistent and fundamental flaw: the tendency to hallucinate. A hallucination is the generation of text that is fluent, confident, and plausible, but which is demonstrably false or nonsensical [1]. This issue undermines the trustworthiness of LLMs and poses a significant barrier to their adoption in high-stakes, mission-critical domains where factual accuracy is paramount. While numerous theories have been proposed, from training data artifacts to stochastic decoding, they often fall short of explaining the systematic nature of these failures.

This paper synthesizes two complementary lines of research to construct a cohesive and comprehensive theory of hallucination. We argue that hallucination is not a single problem but a two-stage process. The initial error is born from a specific, predictable training dynamic, and this error is subsequently reinforced and shaped into its final, confident form by the very ecosystem designed to evaluate and improve these models.

Our central thesis is as follows:

Stage 1: The Statistical Origin of Errors via Local Overfitting. The initial errors, which are essentially knowledge gaps, arise from a powerful training dynamic termed *local overfitting* [2]. Classical overfitting is characterized by a decline in global performance on a validation set. Local overfitting is a more subtle phenomenon where the model's aggregate performance continues to improve, masking a performance degradation on specific, underrepresented sub-regions of the data space. As the model is trained, the optimization process, driven by a standard loss function and regularization, forces the model to specialize intensely on high-frequency, statistically dominant patterns. This comes at a direct cost: the model begins to "forget" knowledge it had previously learned about rare, "long-tail" facts. This forgetting is not a random bug but a structural consequence of prioritizing performance on common data over rare data.

Stage 2: The Socio-Technical Reinforcement of Errors via Evaluation. The knowledge gaps created by local overfitting do not automatically become confident falsehoods. They are molded into hallucinations by the incentives of the post-training and evaluation pipeline. As compellingly argued by Kalai et al. [1], LLMs are optimized to be effective "test-takers." The majority of influential benchmarks that drive progress in the field rely on binary (correct/incorrect) scoring. In this paradigm, admitting uncertainty (e.g., by responding "I don't know") is treated as equivalent to a wrong answer, yielding a score of zero. Consequently, a model aiming to maximize its expected score learns that the optimal strategy is always to guess a plausible-sounding answer when uncertain. This evaluation pressure systematically rewards the behavior of masking knowledge gaps with confident-sounding fabrications, transforming the initial errors into full-blown hallucinations.

1.1 Frequency-Aware Regularization

Beyond diagnosing the problem, this paper introduces a solution. Based on our analysis that identifies standard L2 regularization as a key driver of knowledge forgetting, we propose a new method: **Frequency-Aware Regularization (FAR)**. This is a principled modification to the training objective that applies a much lower regularization penalty to the parameters associated with rare facts. We provide a rigorous mathematical proof within our toy model framework that FAR successfully prevents the decay of weights for rare knowledge, thereby mitigating the root cause of this class of hallucinations.

1.2 Contributions and Paper Structure

This paper integrates existing theories and new proposals into a single, comprehensive narrative. The primary contributions are:

- A formal mathematical framework that synthesizes the concepts of local overfitting and the statistical pressures of evaluation.
- A detailed mathematical toy model that provides a rigorous, step-by-step proof of how local overfitting causes knowledge forgetting from first principles.
- The introduction and formal analysis of Frequency-Aware Regularization (FAR), a novel method to combat knowledge forgetting, complete with a mathematical proof of its efficacy in our model.
- An analysis of how misaligned evaluation benchmarks create an "epidemic" of penalizing uncertainty, reinforcing hallucinatory behavior.

1.3 The remainder of this paper is organized as follows:

Section 2 establishes the formal notation.
Section 3 introduces and analyzes our toy model, proving the mechanism of forgetting under standard regularization.
Section 4 introduces our novel FAR scheme and proves its ability to preserve knowledge.
Section 5 connects these mechanisms to the broader theories of local overfitting.
Section 6 explains why errors persist as hallucinations due to evaluation pressures.
Section 7 discusses mitigation strategies,
and Section 8 concludes.

2 Background and Formal Preliminaries

In this section, we establish the mathematical notation and formal definitions that will serve as the foundation for our analysis. We begin by formalizing the structure of an LLM and its training process, then introduce our core definitions of local overfitting and knowledge forgetting, drawing directly from the foundational work of Stern et al. [2] and Kalai et al. [1].

2.1 LLM Formalism and Training

Let $\mathcal{V}$ be a finite vocabulary of tokens. A prompt c is a sequence of tokens from $\mathcal{V}$. A response r is also a sequence of tokens from $\mathcal{V}$. Let $\mathcal{C}$ and $\mathcal{R}$ be the sets of all possible prompts and responses, respectively.

Definition 2.1 (Large Language Model)**.** A Large Language Model M_θ is a function parameterized by a vector $\theta \in \mathbb{R}^P$ that maps a prompt $c \in \mathcal{C}$ to a probability distribution over the vocabulary $\mathcal{V}$ for the next token. Autoregressively, it defines a probability distribution over the entire response space $\mathcal{R}$:

$$P_\theta(r|c) = \prod_{j=1}^{m} P_\theta(r_j|c, r_1, \ldots, r_{j-1}) \tag{1}$$

The model is trained on a large dataset $\mathcal{D}_{\text{train}}$ of (c, r) pairs. The standard training objective is to minimize the negative log-likelihood, or cross-entropy loss, over this dataset, often with a regularization term to control parameter norms.

Definition 2.2 (Training Objective)**.** The regularized loss function $\mathcal{L}_{\text{reg}}(\theta)$ for a given set of parameters θ is:

$$\mathcal{L}_{\text{reg}}(\theta) = -\mathbb{E}_{(c,r)\sim\mathcal{D}_{\text{train}}}[\log P_\theta(r|c)] + \frac{\lambda}{2}\|\theta\|_2^2 \tag{2}$$

where λ is the regularization strength. This objective is typically optimized using stochastic gradient descent (SGD) or its variants.

2.2 Local Overfitting and Forgetting

We adopt the framework from Stern et al. [2] to precisely define local overfitting and the phenomenon of forgetting. Let $acc(e, S)$ be the accuracy of the model with parameters θ_e from epoch e on a data set S. Let E be the final training epoch.

Definition 2.3 (Forget and Learn Sets)**.** Let T be the test (or validation) set. Let M_e be the subset of T that is misclassified by the model at epoch e. The *forget set* with respect to the final model is the set of examples that the final model gets wrong but the model at epoch e got right. The *learn set* is the converse. This leads to the following definitions for the normalized fractions.

$$\text{Forget Fraction } (F_e) = \frac{acc(e, M_E)}{|T|} \tag{3}$$

$$\text{Learn Fraction } (L_e) = \frac{acc(E, M_e)}{|T|} \tag{4}$$

F_e represents the fraction of the test set that was "forgotten" between epoch e and the final epoch E. L_e represents the fraction that was "learned" in the same period. The change in accuracy can be expressed as: $acc(E, T) - acc(e, T) = L_e - F_e$.

This decomposition allows for a more nuanced definition of overfitting.

Definition 2.4 (Local Overfitting [2])**.** A model exhibits *local overfitting* if $F_e > 0$ for some mid-training epoch $e < E$. This holds even if the global accuracy is non-decreasing, which occurs when $L_e \geq F_e$. Local overfitting captures the idea that a model can sacrifice performance on one part of the data distribution to improve performance on another, even if the net effect on global accuracy is positive or neutral.

2.2.1 Clarifying Forgetting: Generalized Patterns vs. Memorized Data

It is crucial to clarify what is "forgotten" in the context of local overfitting, as the term can be counter-intuitive. Traditional overfitting is synonymous with memorization of training data at the expense of generalization. Local overfitting presents a more nuanced picture.

In the framework we present, the forgetting factor does not measure the forgetting of the high-frequency training data that the model is actively overfitting on. In fact, the model's representation of this frequent data becomes stronger and more entrenched. Instead, **forgetting refers to the loss of correctly learned generalized patterns for rare or low-frequency data**.

Early in training, the model learns a broad set of patterns, including those that correctly map rare prompts to their correct responses. This is a form of successful generalization. As training progresses, the optimization pressure to reduce the global loss forces the model to specialize on the high-frequency data. This intense specialization (local overfitting) causes the neural pathways that encode the patterns for rare data to weaken and decay, as demonstrated in our toy model. The model sacrifices a correct, but weakly supported, generalization on the rare data to achieve a better fit on the frequent data.

Therefore, the "forgetting factor" is a measure of lost *generalization* on a specific subpopulation of the data, not a measure of lost *memorization*. This distinction is fundamental to understanding why local overfitting can occur even while global performance metrics are improving.

2.3 Errors as a Classification Problem

We use the framework from Kalai et al. [1] to connect the generative task of an LLM to a discriminative task. Let the space of all plausible responses $\mathcal{X}$ be partitioned into a set of valid statements $\mathcal{V}$ and a set of errors $\mathcal{E}$.

Definition 2.5 (Generative Error Rate)**.** For a model $\hat{p}$, the generative error rate is its total probability mass on the set of erroneous responses:

$$err := \hat{p}(\mathcal{E}) = \Pr_{x \sim \hat{p}}[x \in \mathcal{E}] \tag{5}$$

Definition 2.6 (Is-It-Valid Classification)**.** The "Is-It-Valid" (IIV) problem is a binary classification task to determine if a given statement $x \in \mathcal{X}$ is valid ($f(x) = +$) or an error ($f(x) = -$). Any language model $\hat{p}$ can be converted into an IIV classifier $\hat{f}$ by thresholding its probability assignments, e.g., $\hat{f}(x) = +$ if $\hat{p}(x) > t$ for some threshold t. The misclassification rate of this induced classifier is denoted err_{iiv}.

Kalai et al. [1] establish a formal link between these two quantities.

Theorem 2.7 ([1])**.** *For any language model $\hat{p}$ trained on valid data ($p(\mathcal{E}) = 0$), its generative error rate is lower-bounded by its IIV misclassification rate:*

$$err \geq 2 \cdot err_{iiv} - \frac{\max_c |\mathcal{V}_c|}{\min_c |\mathcal{E}_c|} - \delta \tag{6}$$

where $\delta = |\hat{p}(\mathcal{A}) - p(\mathcal{A})|$ is a calibration term for the set $\mathcal{A}$ of examples above the classifier's threshold. This theorem implies that an inability to reliably classify facts from falsehoods is a sufficient condition for generating falsehoods.

3 A Mathematical Toy Model of an LLM

To rigorously demonstrate the causal link between local overfitting and hallucination, we now construct and analyze a simplified mathematical toy model of an LLM. This model, while abstracting away the complexity of transformers, retains the core components necessary to illustrate the dynamics of learning and forgetting under data imbalance and regularization: a parametric associative memory, a gradient-based optimization objective, and a regularizer.

3.1 Model Specification

We define a linear "Associative Memory" model. This can be viewed as the final layer of a deep network where all preceding layers have extracted a perfect, orthogonal feature representation for each concept.

Definition 3.1 (Toy LLM Model)**.**

- **Concepts:** Let there be a set of N input concepts (prompts) and M output concepts (responses). We represent each concept as a one-hot vector. The prompt for concept i is $x_i \in \{0,1\}^N$, and the target response for concept j is $y_j \in \{0,1\}^M$.
- **Parameters:** The model's knowledge is stored in a single weight matrix $\mathbf{W} \in \mathbb{R}^{N \times M}$. Each entry W_{ij} represents the strength of the association between input concept i and output concept j.
- **Prediction:** The model produces a probability distribution over the output concepts via the softmax function. For an input prompt x_i, the probability of output concept j is:

$$P(y_j|x_i;\mathbf{W}) = \frac{\exp(x_i^T\mathbf{W})_j}{\sum_{k=1}^{M}\exp(x_i^T\mathbf{W})_k} = \frac{\exp(W_{ij})}{\sum_{k=1}^{M}\exp(W_{ik})} \tag{7}$$

  The term $x_i^T\mathbf{W}$ simply selects the i-th row of the weight matrix, which contains the logits for the output distribution.

3.2 Training Setup

We construct a minimal training set that embodies the principles of high-frequency and low-frequency data regions.

- **Frequent Fact:** A single, dominant fact, which we denote as (x_f, y_f). For example, ("France", "Paris"). This pair is presented to the model K times during training, where K is a large integer.
- **Rare Fact:** A single, specific fact, denoted as (x_r, y_r). For example, ("Bhutan", "Thimphu"). This pair is presented only once. This represents a "singleton" fact as described in [1].
- **Plausible Foil:** We assume the existence of another output concept, y_h ("Kathmandu"), which is never the correct label for x_r but is a plausible answer type. It has an associated weight W_{rh}.

The training objective is to minimize the cross-entropy loss with L2 regularization (weight decay). The total loss for our toy dataset is:

$$\mathcal{L}(\mathbf{W}) = \underbrace{-K \log P(y_f|x_f;\mathbf{W})}_{\text{Frequent Loss}} \underbrace{- \log P(y_r|x_r;\mathbf{W})}_{\text{Rare Loss}} + \frac{\lambda}{2}\|\mathbf{W}\|_F^2 \tag{8}$$

where $\|\mathbf{W}\|_F^2 = \sum_{i,j} W_{ij}^2$ is the squared Frobenius norm.

3.3 Analysis of Training Dynamics

We analyze the evolution of the weight corresponding to the rare fact, specifically W_{rr} (the connection from "Bhutan" to "Thimphu"), during gradient descent. The gradient of the loss with respect to an arbitrary weight W_{ij} is:

$$\frac{\partial \mathcal{L}}{\partial W_{ij}} = -K\frac{\partial \log P_f}{\partial W_{ij}} - \frac{\partial \log P_r}{\partial W_{ij}} + \lambda W_{ij} \tag{9}$$

where $P_f = P(y_f|x_f;\mathbf{W})$ and $P_r = P(y_r|x_r;\mathbf{W})$.

Theorem 3.2 (Knowledge Forgetting)**.** *In the described toy model trained with gradient descent, for a sufficiently large number of frequent-fact updates K and a non-zero regularization strength $\lambda > 0$, the weight W_{rr} corresponding to the rare fact will decay towards zero during the later stages of training, even after having been correctly learned.*

Proof. Let's analyze the training process step-by-step. Assume the weights $\mathbf{W}$ are initialized to small random values, e.g., $W^{(0)} \sim \mathcal{N}(0, \sigma^2)$. Let the learning rate be η. The training proceeds in two phases: first, one update for the rare fact, and then K updates for the frequent fact.

Phase 1: Learning the Rare Fact. The model sees the rare fact (x_r, y_r) once. The gradient of the loss with respect to W_{rr} is dominated by the rare loss term. The derivative of the log-softmax is $\frac{\partial \log P(y_j|x_i)}{\partial W_{ik}} = \delta_{ij}(\delta_{jk} - P(y_k|x_i))$. For our specific weight W_{rr}:

$$\frac{\partial \mathcal{L}}{\partial W_{rr}}\Big|_{\text{rare}} = -(1 - P(y_r|x_r;W^{(0)})) + \lambda W_{rr}^{(0)} \tag{10}$$

At initialization, $W^{(0)}$ is small, so $P(y_r|x_r;W^{(0)}) \approx 1/M$. The regularization term $\lambda W_{rr}^{(0)}$ is also negligible. The gradient is approximately -1. The weight update is:

$$W_{rr}^{(1)} = W_{rr}^{(0)} - \eta\left(-(1 - P(y_r|x_r)) + \lambda W_{rr}^{(0)}\right) \approx W_{rr}^{(0)} + \eta \tag{11}$$

This single update significantly increases W_{rr}, strengthening the correct association. For a suitable η, we can ensure that $W_{rr}^{(1)}$ is now the largest weight in its row, meaning the model correctly predicts y_r for the prompt x_r. At this point, the model has "learned" the fact.

Phase 2: Local Overfitting on the Frequent Fact. Next, the model is trained on the frequent fact (x_f, y_f) for K iterations. Consider the update to the weight W_{rr} during one of these iterations. The gradient is now computed with respect to the frequent loss term and the regularization term.

The frequent loss term, $-K \log P(y_f|x_f;\mathbf{W})$, depends only on the f-th row of the weight matrix, $\mathbf{W}_{f,:}$. Since we assume orthogonal inputs ($r \neq f$), it does not depend on W_{rr}. Therefore:

$$\frac{\partial}{\partial W_{rr}}\left(-K \log P(y_f|x_f;\mathbf{W})\right) = 0 \tag{12}$$

The gradient for W_{rr} during these K updates is thus driven *solely* by the regularization term:

$$\frac{\partial \mathcal{L}}{\partial W_{rr}}\Big|_{\text{freq}} = \lambda W_{rr} \tag{13}$$

The Forgetting Mechanism. The weight update for W_{rr} during each of the K frequent-fact iterations is a decay step:

$$W_{rr}^{(t+1)} = W_{rr}^{(t)} - \eta(\lambda W_{rr}^{(t)}) = (1 - \eta\lambda)W_{rr}^{(t)} \tag{14}$$

This is a discrete-time exponential decay process. After K such updates, the weight that was initially boosted to $W_{rr}^{(1)}$ will have decayed to:

$$W_{rr}^{(K+1)} = (1 - \eta\lambda)^K W_{rr}^{(1)} \tag{15}$$

For any $\eta\lambda \in (0, 1)$, as $K \to \infty$, we have $(1 - \eta\lambda)^K \to 0$. Therefore, for a sufficiently large K, the weight $W_{rr}^{(K+1)}$ will decay back towards zero. This is the mathematical proof of **forgetting**. The optimization process, in its effort to fit the frequent data and satisfy the regularizer's preference for small weights, has actively destroyed the information learned from the rare data point. □

3.4 The Emergence of Hallucination

We have proven that the correct knowledge is forgotten. Now we demonstrate how a hallucination takes its place.

Proposition 3.3 (Hallucination Trigger)**.** *After the weight for the correct rare fact, W_{rr}, has decayed below the value of some other weight in its row, W_{rh}, corresponding to a plausible but incorrect answer, the model will output the incorrect response y_h. This incorrect output is a hallucination.*

Proof. A correct prediction of y_r for prompt x_r occurs when W_{rr} is the largest weight in the r-th row of $\mathbf{W}$. That is, $W_{rr} > W_{rk}$ for all $k \neq r$.

Let the weights be initialized from a zero-mean Gaussian, $W_{ij}^{(0)} \sim \mathcal{N}(0, \sigma^2)$. After the single update for the rare fact at $t = 1$, we have $W_{rr}^{(1)} \approx \eta$, while all other weights in that row, $W_{rk}^{(1)}$ for $k \neq r$, remain at their small initial values. Thus, $W_{rr}^{(1)} > W_{rk}^{(1)}$, and the model is correct.

During the subsequent K decay steps (from $t = 1$ to $t = K + 1$), all weights in the r-th row that are not being explicitly trained (i.e., all of them) undergo the same exponential decay:

$$W_{rk}^{(K+1)} = (1 - \eta\lambda)^K W_{rk}^{(1)} \tag{16}$$

This includes our correct weight W_{rr} and the foil weight W_{rh}. Because the decay factor is uniform, the relative ordering of the weights within the row does not change. So, if $W_{rr}^{(1)} > W_{rh}^{(1)}$, then $W_{rr}^{(K+1)} > W_{rh}^{(K+1)}$. This simplified view suggests the model never hallucinates.

However, this analysis omits the critical effect of gradient noise and crosstalk in a real (non-linear, non-orthogonal) network. The updates for the frequent fact, while not directly targeting W_{rr}, would cause small, effectively random perturbations to the weights in the r-th row. A more realistic model for the update would be:

$$W_{rk}^{(t+1)} = (1 - \eta\lambda)W_{rk}^{(t)} + \xi_k^{(t)} \tag{17}$$

where $\xi_k^{(t)}$ is a small, zero-mean noise term from gradient crosstalk.

While the correct weight W_{rr} is systematically decaying towards zero, the other weights, including the foil W_{rh}, are performing a random walk that is also biased towards zero. The final output depends on which weight "wins the race to the bottom."

The weight W_{rr} starts with a large positive value $W_{rr}^{(1)} \approx \eta$ and decays deterministically. The foil weight W_{rh} starts with a small random value $W_{rh}^{(1)} = W_{rh}^{(0)}$. It is possible that $W_{rh}^{(0)}$ was initialized to a negative value, in which case it will decay towards zero from below and is unlikely to cause a hallucination. However, it is also possible that it was initialized to a small positive value, $\epsilon > 0$.

The condition for hallucination at the final step $K+1$ is $W_{rh}^{(K+1)} > W_{rr}^{(K+1)}$. Even without noise, if for some reason the foil weight started larger than the boost given to the correct weight (e.g., a very small learning rate η), this could happen. More realistically, the random perturbations ξ_h could accumulate in a positive direction for the foil weight.

Once W_{rr} has decayed sufficiently close to zero, its value becomes comparable to the other randomly fluctuating weights in its row. If any of these foil weights, like W_{rh}, happens to end up with a larger value, the softmax function will assign it the highest probability. The model will then confidently output the plausible but incorrect response y_h. This is the hallucination, born directly from the vacuum left by the forgotten knowledge.

4 A Novel Contribution: Frequency-Aware Regularization

Our analysis in the previous section identified standard, uniform L2 regularization as the key mechanism driving the forgetting of rare facts. This suggests a direct and principled path toward a solution: if the regularization pressure is the problem, then we should modify it. In this section, we introduce a novel regularization scheme, Frequency-Aware Regularization (FAR), and prove its efficacy in preventing knowledge forgetting within our toy model framework.

4.1 Defining Frequency-Aware Regularization (FAR)

The core idea behind FAR is to make the regularization strength dependent on the data being processed. Instead of a single global λ, we associate a specific regularization parameter with each training example, or more practically, with each parameter block affected by that example.

Definition 4.1 (Frequency-Aware Regularization)**.** Let the training loss for a single example (x_i, y_i) be $\mathcal{L}_i(\theta) = -\log P_\theta(y_i|x_i)$. The FAR objective modifies the L2 penalty to be instance-specific:

$$\mathcal{L}_{\text{FAR}}(\theta) = \sum_{i \in \mathcal{D}_{\text{train}}} \left(\mathcal{L}_i(\theta) + \frac{\lambda_i}{2} \|\theta\|_2^2 \right) \tag{18}$$

where λ_i is the regularization strength associated with example i. The key principle is to set λ_i to be inversely related to the frequency of the concepts in example i.

For our toy model, this simplifies dramatically. We define two regularization parameters:

- λ_f: The regularization strength for the frequent fact (x_f, y_f).
- λ_r: The regularization strength for the rare fact (x_r, y_r).

The FAR principle dictates that we should set $\lambda_f \gg \lambda_r$. In the ideal case, we can set $\lambda_r = 0$, effectively turning off weight decay for the parameters being updated by the rare fact.

The modified loss function for the toy model becomes:

$$\mathcal{L}_{\text{FAR}}(\mathbf{W}) = -K \log P_f - \log P_r + \frac{K\lambda_f}{2} \|\mathbf{W}_{f,:}\|_2^2 + \frac{\lambda_r}{2} \|\mathbf{W}_{r,:}\|_2^2 \tag{19}$$

Note: For simplicity, we've applied the regularization only to the row of weights being updated. A full implementation would apply a weighted sum of penalties.

4.2 Analysis of FAR in the Toy Model

We now re-run the analysis from Section 3 using the FAR objective.

Theorem 4.2 (Preservation of Rare Knowledge under FAR)**.** *In the toy model trained with the FAR objective, setting the rare-fact regularization strength $\lambda_r = 0$ prevents the systematic decay of the learned weight W_{rr}, thereby preserving the model's knowledge of the rare fact.*

Proof. The proof follows the same two-phase structure as before.

Phase 1: Learning the Rare Fact. The model sees the rare fact (x_r, y_r) once. The gradient update is now with respect to a loss term that includes a regularization penalty weighted by λ_r.

$$\frac{\partial \mathcal{L}_{\text{FAR}}}{\partial W_{rr}}\Big|_{\text{rare}} = -(1 - P(y_r|x_r; W^{(0)})) + \lambda_r W_{rr}^{(0)} \tag{20}$$

Since we set $\lambda_r = 0$, the regularization term vanishes entirely. The update is purely driven by the prediction error:

$$W_{rr}^{(1)} = W_{rr}^{(0)} - \eta(-(1 - P(y_r|x_r))) \approx W_{rr}^{(0)} + \eta \tag{21}$$

This is identical to the standard case. The model learns the rare fact and the weight $W_{rr}^{(1)}$ becomes large and positive.

Phase 2: Updates on the Frequent Fact. Next, the model is trained on the frequent fact (x_f, y_f) for K iterations. We analyze the gradient for the rare weight W_{rr} during one of these updates. The loss term being optimized is now: $\mathcal{L}_f(\mathbf{W}) = -\log P_f + \frac{\lambda_f}{2}\|\mathbf{W}_{f,:}\|_2^2$.

As before, the cross-entropy term $-\log P_f$ does not depend on the weights in the r-th row, $\mathbf{W}_{r,:}$. Crucially, the FAR regularization term $\frac{\lambda_f}{2}\|\mathbf{W}_{f,:}\|_2^2$ also only depends on the weights in the f-th row. Therefore, it has no partial derivative with respect to W_{rr}:

$$\frac{\partial}{\partial W_{rr}}\left(\frac{\lambda_f}{2}\|\mathbf{W}_{f,:}\|_2^2\right) = 0 \tag{22}$$

This means that the entire gradient for the weight W_{rr} during the frequent-fact updates is zero:

$$\frac{\partial \mathcal{L}_{\text{FAR}}}{\partial W_{rr}}\Big|_{\text{freq}} = 0 \tag{23}$$

The Preservation Mechanism. The weight update for W_{rr} during each of the K frequent-fact iterations is now:

$$W_{rr}^{(t+1)} = W_{rr}^{(t)} - \eta \cdot 0 = W_{rr}^{(t)} \tag{24}$$

There is no decay step. The value of the weight, $W_{rr}^{(1)}$, learned during Phase 1 is perfectly preserved throughout the K updates for the frequent fact.

$$W_{rr}^{(K+1)} = W_{rr}^{(1)} \tag{25}$$

Thus, the knowledge of the rare fact is not forgotten. By making the regularization aware of the data's frequency, we have eliminated the mechanism that was previously destroying the information. □

This novel result provides a constructive proof-of-concept. It shows that by redesigning the training objective in a principled way, it is possible to mitigate the harmful effects of local overfitting on rare knowledge. While implementing FAR in a real-world LLM would require methods to estimate fact frequency and apply targeted regularization, this theoretical result demonstrates that knowledge forgetting is not an immutable law, but a consequence of a specific, and modifiable, training choice.

5 The Origin of Hallucination in Practice

The toy model provides a crisp mathematical illustration, but the underlying principles apply directly to complex deep networks. The origin of errors is the model's effort to optimize its parameters on a highly imbalanced data distribution, leading it to discard information about rare data.

5.1 The Forgetting Mechanism in Deep Linear Models

To bridge the gap between the toy model and real networks, Stern et al. [2] analyze the training dynamics of an over-parameterized deep linear network using Gradient Descent (GD). This model, while linear, captures the non-linear optimization landscape of deep networks. They show that the learned separator w converges at a rate determined by the singular values of the data's covariance matrix. The model first learns the projection of the optimal separator onto the data's leading eigenvectors, corresponding to the directions of highest variance (i.e., the most dominant patterns). Knowledge related to directions of low data variance (less common patterns) is learned much more slowly and is more susceptible to being overwritten.

The analysis reveals that the rate at which a data point is forgotten is linked to its spectral properties. Specifically, "a point will be forgotten faster if the length of its spectral decomposition vector is dominated by its first components" [2]. This means that data points that align well with the most common patterns in the dataset are paradoxically also the ones whose unique, specific information is most likely to be forgotten if that information is not part of the dominant pattern. The optimization process, guided by regularization, prioritizes fitting the "big picture" and will decay the weights responsible for fine-grained details that are not frequently reinforced.

5.2 Forgetting as a Failure of Classification

The mechanism of forgetting provides a direct link to the IIV classification framework from Kalai et al. [1]. A "forgotten" fact is precisely one for which the model can no longer solve the IIV task. The neural representation has decayed to the point where the model cannot distinguish the valid fact from a plausible but erroneous alternative. This failure is most acute for what Kalai et al. term "Arbitrary Facts": knowledge with no discernible underlying pattern, such as birthdays or arbitrary historical events. For these facts, learning is equivalent to memorization. The paper provides a powerful theoretical bound connecting the hallucination rate for such facts to the "singleton rate" (sr), which is the fraction of non-IDK facts that appear exactly once in the training data.

Theorem 5.1 (Arbitrary Facts, [1])**.** *In the Arbitrary Facts model, any well-calibrated algorithm that takes N training samples will have an error rate that is tightly coupled to the singleton rate:*

$$err \geq sr - \frac{2}{\min_c |\mathcal{E}_c|} - O\left(\frac{\log N}{\sqrt{N}}\right) - \delta \tag{26}$$

Furthermore, an efficient algorithm exists that achieves $err \leq sr$.

This theorem formalizes the intuition that facts seen only once are the most vulnerable to being forgotten. The local overfitting dynamic provides the physical mechanism for this

statistical observation: the single training signal from a singleton fact is not strong enough to resist the relentless decay pressure from regularization and the optimization focus on more frequent data. Thus, the model learns the singleton fact early in training, then forgets it, leading to a high IIV error rate and, consequently, a high hallucination rate for that fact.

6 The Persistence of Hallucination

Local overfitting explains the origin of knowledge gaps, but not why models fill these gaps with confident falsehoods instead of expressing uncertainty. This section, based on the analysis by Kalai et al. [1], argues that this behavior is a learned response to the incentives created by modern evaluation benchmarks.

6.1 The Test-Taker Analogy

Language models can be compared to students taking an exam. When faced with a difficult question, a student might guess a plausible answer rather than leaving it blank, especially if there is no penalty for wrong answers. The language model is in a similar situation. It is constantly being evaluated on benchmarks that determine its ranking on leaderboards, and these evaluations shape its behavior through post-training methods like Reinforcement Learning from Human Feedback (RLHF).

The vast majority of these benchmarks employ binary (0-1) grading, where a correct answer receives 1 point and any other response, including "I don't know" (IDK), receives 0 points. This creates a clear incentive structure.

Proposition 6.1 (Optimality of Guessing, [1])**.** *For any prompt c and any distribution ρ_c over binary graders, the optimal response(s) that maximize the expected score are not abstentions.*

$$\arg\max_{r \in \mathcal{R}_c} \mathbb{E}_{g_c \sim \rho_c}[g_c(r)] \cap \mathcal{A}_c = \emptyset \tag{27}$$

where $\mathcal{R}_c$ is the set of plausible responses and $\mathcal{A}_c \subset \mathcal{R}_c$ is the set of abstention responses.

The proof is straightforward: an abstention response guarantees a score of 0. Any guess, no matter how uncertain, has an expected score of $p_{correct} \cdot 1 + (1 - p_{correct}) \cdot 0 = p_{correct}$. As long as the model believes there is any non-zero chance of being correct ($p_{correct} > 0$), the expected score from guessing is strictly greater than the score from abstaining. Since post-training methods optimize models to perform well on these benchmarks, they are implicitly training the model to guess whenever it is uncertain.

6.2 The "Epidemic" of Penalizing Uncertainty

The dominance of binary grading across the most influential benchmarks creates what Kalai et al. call an "epidemic of penalizing uncertainty." A model that is well-calibrated and honestly reports its uncertainty will be systematically outcompeted on nearly all major leaderboards by a similar model that simply provides its best guess in all situations. This creates a powerful, market-driven incentive for developers to train models that are overconfident and prone to hallucination.

This is a deep socio-technical problem. The issue is not just that we lack a perfect hallucination evaluation; it's that the primary evaluations the community uses to measure progress are fundamentally misaligned with the goal of truthfulness. A small number of specialized hallucination benchmarks cannot counteract the immense pressure from dozens of mainstream evaluations that implicitly reward guessing.

This explains why, despite significant research and engineering effort, hallucinations persist even in state-of-the-art models. The models are behaving rationally according to the objectives

they are given. The errors created by local overfitting are not being corrected in post-training; they are being molded into confident, plausible-sounding hallucinations because that is the behavior that maximizes performance on our current benchmarks.

7 Mitigation Strategies

The two-stage theory of hallucination, combined with our constructive proof for FAR, suggests a multi-faceted approach to mitigation, addressing the model's learning, its objective function, and the evaluation ecosystem.

7.1 Model-Centric and Objective-Function Mitigation

- **Knowledge Fusion (KF):** As proposed by Stern et al. [2], this ensemble method recovers forgotten knowledge from earlier training checkpoints. It serves as an effective post-hoc fix, demonstrating that the lost information is still present in the model's history.
- **Frequency-Aware Regularization (FAR):** Our novel proposal tackles the problem at its source. By modifying the training objective itself, FAR prevents forgetting from happening in the first place. Future work should explore practical methods for estimating data frequency at scale and applying this targeted regularization in large transformer models.
- **Data-Centric Approaches:** While not the focus of our mathematical analysis, strategies like up-sampling rare examples or using a curriculum that introduces rare facts early and reinforces them can be seen as practical heuristics to achieve a similar goal to FAR.

7.2 Evaluation-Centric Mitigation: Realigning Incentives

While model-centric fixes are crucial, Kalai et al. [1] argue for a more fundamental shift in the evaluation system that currently rewards hallucination. This can be done by introducing **explicit confidence targets**. Instead of a simple prompt, each evaluation question would include instructions on the scoring rule, which incorporates a penalty for incorrect answers. For example:

> Answer only if you are > 90% confident. Correct answers receive 1 point, mistakes are penalized 9 points, and an answer of "I don't know" receives 0 points.

A simple calculation shows that under this scoring rule, it is only optimal for a model to provide an answer if its internal probability of being correct is greater than the specified threshold (in this case, 0.9). This changes the optimization target from "always provide an answer" to "provide a correct answer or admit uncertainty."

This is a socio-technical solution. It requires not only developing new scoring rubrics but also convincing the community to adopt them for major leaderboards. However, by realigning the incentives, this approach could steer the entire field toward developing more truthful and reliable models, effectively treating the disease rather than just the symptoms.

8 Conclusion

We have synthesized two leading theories and introduced a novel constructive proposal to create a comprehensive explanation for hallucination in LLMs. The problem begins with a statistical flaw in the training process and is exacerbated by a systemic flaw in the evaluation ecosystem, forming a causal chain that leads directly to the generation of plausible falsehoods.

Local overfitting, as described and analyzed by Stern et al. [2], is the engine of error. It is a natural consequence of applying powerful optimizers with uniform regularization to imbalanced real-world data. This process causes models to forget specific, rare knowledge as an unavoidable side effect of their intense specialization on common, frequent patterns. This creates the knowledge gaps that are the seeds of hallucination.

Misaligned evaluations, as analyzed by Kalai et al. [1], provide the reinforcement mechanism. By overwhelmingly rewarding guessing and penalizing uncertainty, they train models to fill these knowledge gaps with confident-sounding falsehoods rather than admissions of ignorance.

Our novel contribution, **Frequency-Aware Regularization (FAR)**, provides a direct, principled solution to the first stage of this problem. By proving that a modified training objective can prevent knowledge forgetting, we show that this flaw is not inevitable. This transforms the understanding of hallucination from a necessary evil to a solvable engineering and design challenge.

Ultimately, durable solutions will require a three-pronged attack: post-hoc recovery methods like Knowledge Fusion, principled training modifications like FAR, and systemic evaluation reforms like explicit confidence targets. By addressing the origins of errors, the objectives that guide learning, and the socio-technical pressures that shape final model behavior, we can pave a more direct path toward models that not only know what they know, but are also honest about what they don't.

References

[1] Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. (2025). *Why Language Models Hallucinate*. arXiv preprint arXiv:2509.04664.

[2] Uri Stern, Tomer Yaacoby, and Daphna Weinshall. (2025). *On Local Overfitting and Forgetting in Deep Neural Networks*. The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25).

A Detailed Proofs and Derivations

A.1 Gradient of the Log-Softmax Function

In our toy model, we rely on the gradient of the cross-entropy loss, which involves the log-softmax function. Let the logits for input x_i be $z_k = W_{ik}$ for $k = 1, \ldots, M$. The softmax

probability is $p_j = \frac{e^{z_j}}{\sum_l e^{z_l}}$. We want to compute $\frac{\partial \log p_j}{\partial z_k}$.

$$\begin{aligned}\frac{\partial \log p_j}{\partial z_k} &= \frac{1}{p_j}\frac{\partial p_j}{\partial z_k} \\ &= \frac{\sum_l e^{z_l}}{e^{z_j}}\frac{\partial}{\partial z_k}\left(\frac{e^{z_j}}{\sum_l e^{z_l}}\right)\end{aligned}$$

We use the quotient rule. First, consider the case where $j = k$:

$$\begin{aligned}\frac{\partial p_j}{\partial z_j} &= \frac{e^{z_j}(\sum_l e^{z_l}) - e^{z_j}(e^{z_j})}{(\sum_l e^{z_l})^2} \\ &= \frac{e^{z_j}}{\sum_l e^{z_l}}\left(1 - \frac{e^{z_j}}{\sum_l e^{z_l}}\right) = p_j(1 - p_j)\end{aligned}$$

Now, consider the case where $j \neq k$:

$$\frac{\partial p_j}{\partial z_k} = \frac{0 - e^{z_j}(e^{z_k})}{(\sum_l e^{z_l})^2} = -p_j p_k$$

Substituting these back into the expression for the log-derivative:

$$\begin{aligned}\frac{\partial \log p_j}{\partial z_k} &= \frac{1}{p_j}(p_j(1 - p_j)\delta_{jk} - p_j p_k(1 - \delta_{jk})) \\ &= (1 - p_j)\delta_{jk} - p_k(1 - \delta_{jk}) \\ &= \delta_{jk} - p_k\end{aligned}$$

The cross-entropy loss for a single example (x_i, y_j) is $-\log p_j$. Its gradient with respect to a logit z_k is therefore $p_k - \delta_{jk}$. This is the well-known result that the gradient of the cross-entropy loss is simply the difference between the model's prediction and the true label.