

Hallucination is a Consequence of Space-Optimality

A rate–distortion theorem for membership testing

Anxin (Bob) Guo¹ (speaker) Jingwei Li²

ICML 2026 Spotlight

¹Northwestern University

²Columbia University

- **Hallucination:** generating plausible but false statements with confidence.
- **Random facts:** no inferable structure; they can be memorized, never derived.



case citations



paper titles & authors



biographical details

When is Jingwei Li's birthday?

March 14, 1996.

Confident, specific, and wrong.

AI keeps inventing fake cases. Lawyers keep citing them

The trend of attorneys getting caught citing AI-hallucinated cases points to a broader problem: instead of checking AI's work, people keep trusting it

BY [STEVEN MELENDEZ](#) EDITED BY [ERIC SULLIVAN](#)

Unseen facts

- Random facts cannot be inferred: answering **unseen facts** requires guessing.
- This explains errors outside training data, and motivates *abstention* on unseen facts.

Kalai & Vempala 2024; Kalai et al. 2025 "Why Language Models Hallucinate"

Lossy compression of knowledge

- When factual information exceeds model capacity, lossy compression can cause errors even on seen facts.
- Mostly high-level and vague arguments suggesting that "there must be *some* error".

Pan et al. 2025; Shi et al. 2025; Mohsin et al. 2025



1. How much memory is needed to distinguish facts from non-facts?

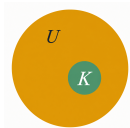
Why can't a large model just memorize all finitely many facts perfectly?

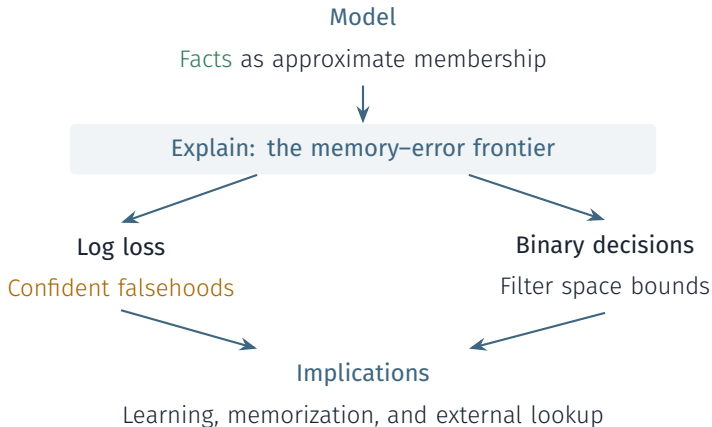
How much error will occur when there is not enough memory?

2. Why confident falsehoods?

When memory is limited, why confidently accept false claims, rather than forgetting facts or lowering confidence uniformly?

Simple setting: a finite set K of *facts* randomly selected from universe U , all available during training. Every claim outside K counts as a *non-fact* (closed world).

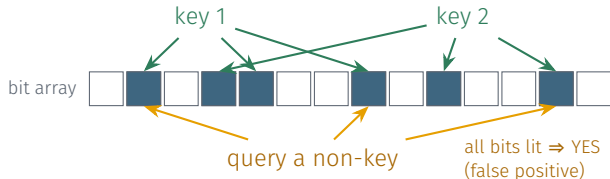




Memorizing facts: a data structure perspective

An analogy: approximate membership filters

A Bloom-type filter *approximately* stores a set $K \subseteq U$ in a compact way.



Insert: hash each key to several bits and set them to 1.

Query: return YES if all its hashed bits are 1.

Bloom filter [Bloom 1970]

- **Keys** always return YES: no false negatives.
- **Non-keys** may return YES: saves space by allowing some false positive rate ϵ .

Space lower bound: $\log_2(1/\epsilon)$ bits per key. [Carter et al. 1978]

This is tight. [Dietzfelbinger & Pagh 2008; Porat 2009]

A Bloom filter

“Is this item in the set?”

A binary verdict: **YES** or **NO**.

An LLM in a YES/NO task

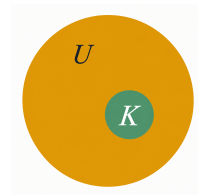
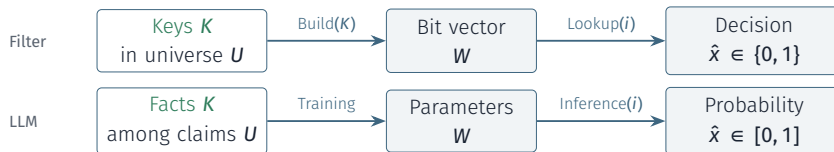
“Is this claim true?”

Model might assign some 70% or 99% probability.

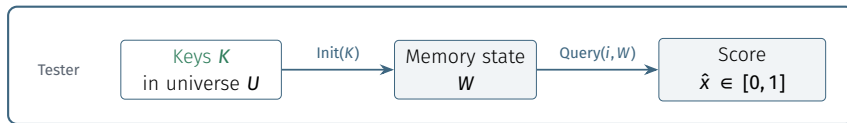
We use a common “confidence level” $\hat{x} \in [0, 1]$ to capture both.

We want the model to be an accurate probabilistic model of the world.

Hence we care about the **log-likelihood of the observed factuality** under each query’s probability $\hat{x}$.

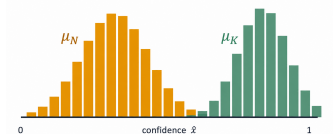


Initialize W with keys K , then use W to answer membership queries.



From individual scores to two distributions

Each query produces a distribution over $\hat{x}$. Group scores by whether the queried element is in K .

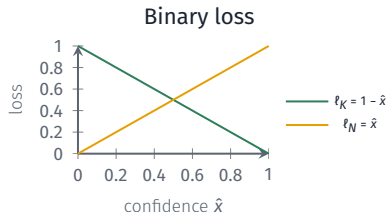


μ_K : keys. Scores on a random key in K .

μ_N : non-keys. Scores on a random non-key in $U \setminus K$.

The two distributions induce two average losses on keys/non-keys:

$$\mathbb{E}_{\mu_K}[\ell_K(\hat{x})] \leq \epsilon_K, \quad \mathbb{E}_{\mu_N}[\ell_N(\hat{x})] \leq \epsilon_N.$$



Main question

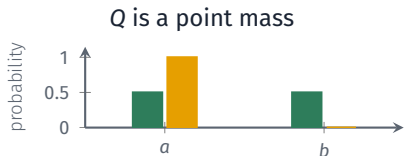
How much **information per key**, $I(W; K)/|K|$, must W retain to meet both error budgets ϵ_K and ϵ_N ?

Memory and hallucination

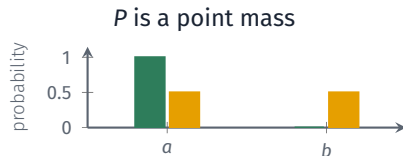
$$\text{KL}(P \parallel Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)}.$$

$\text{KL}(P \parallel Q)$ roughly measures how unlikely it is for samples drawn from Q to have empirical distribution P .

— P : observed — Q : source

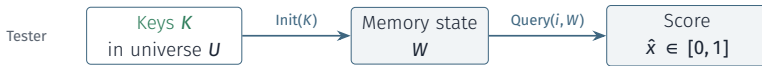


$\text{KL}(P \parallel Q) = \infty$: Q never produces b .



$\text{KL}(P \parallel Q) = 1$ bit: all draws can land on a .

Main result: the memory–error frontier



$$\text{Memory per key} = I(K; W) / |K|.$$

Theorem 3.1 (informal)

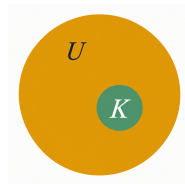
In the sparse limit $|U| \gg |K|$, for two fixed loss functions and loss budgets:

Minimum necessary memory per key = $\min_{\mu_K, \mu_N} \text{KL}(\mu_K \parallel \mu_N)$ bits.

Minimized over distributions meeting both loss budgets. Moreover, optimal tester family's output converges to μ_K, μ_N if the latter are unique minimizers.

Why this direction of KL?

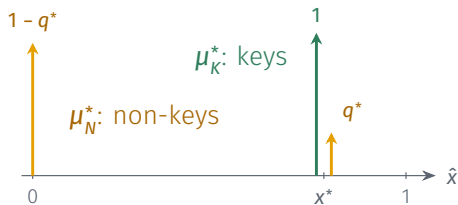
A random query is almost surely a **non-key**. Its score law μ_N is the background; information about K lets **keys** have a different law μ_K .



For specific choice of loss functions, we can solve for optimal μ_K and μ_N .

The most memory-efficient pattern under log loss

Under the optimal distribution for log-loss, **all keys** and **some non-keys** receive **the same score**.



x^* : the shared score.

q^* : the fraction of **non-keys** sharing it.

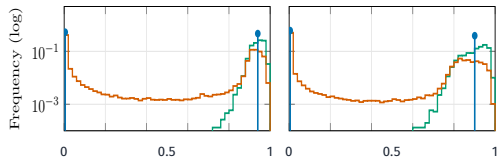
Confident errors can be part of optimal compression, not just failed training.

The predicted distribution in two experiments

— Keys — Non-keys — Theoretical optimum

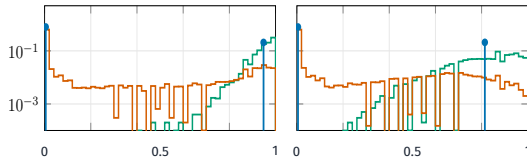
Transformer from scratch

15,145 random 15-letter strings;
two layers, 15,145 parameters.



LoRA fine-tuning

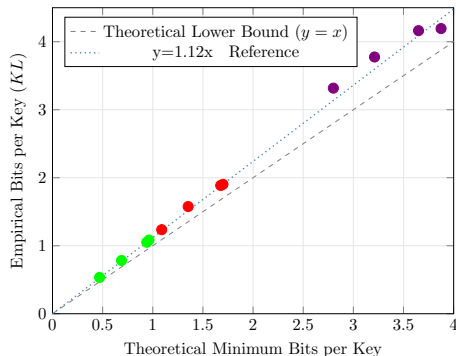
10,000 ISBNs;
Qwen3.5-2B, LoRA rank 2.



Keys (green) and non-keys (orange) overlap at high scores.

Blue stems mark the theoretical optimum at the observed losses.

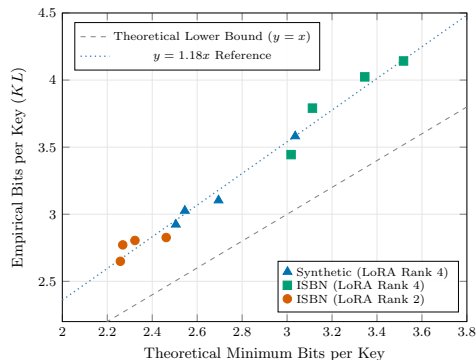
Transformer from scratch



Keys: random 15-letter strings.

*Empirical KL is computed from binned scores, not a direct measurement of $I(K; W)$.

LoRA fine-tuning



Keys: 10,000 random IDs or ISBNs.

Binary decisions and filters

Consider a filter allowing *both* false negatives and false positives.

FNR: the fraction of **keys** rejected.

FPR: the fraction of **non-keys** accepted.

Theorem 4.3: the sparse-limit frontier

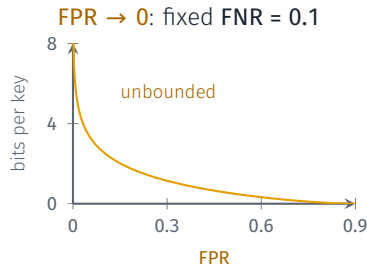
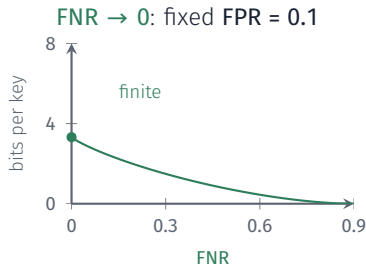
For **FNR** and **FPR** with $\text{FNR} + \text{FPR} < 1$:

Minimum necessary memory per key = $\text{KL}(\text{Bern}(1 - \text{FNR}) \parallel \text{Bern}(\text{FPR}))$ bits.

- Taking **FNR** = 0, we **recover** the one-sided bound $\log_2(1/\text{FPR})$ bits per key. [Carter et al. 1978]
- This **closes the gap** in prior two-sided space bounds. [Pagh & Rodler 2001]
- We give a hash-based construction that attains the bound up to $\mathbf{o(1)}$ bits per key.

Zero false positives require growing bits per key

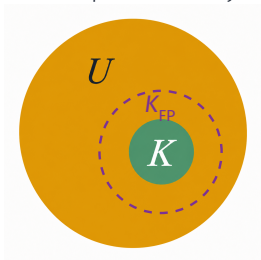
Consider $KL(\text{Bern}(1 - \text{FNR}) \parallel \text{Bern}(\text{FPR}))$ when fixing one error and driving the other to zero.



- **FNR = 0 is possible.** Classical one-sided filter.
- **FPR = 0 requires unbounded bits per key.** No equally compact “reverse Bloom filter.”

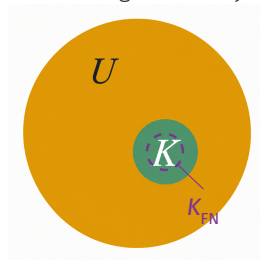
In the sparse limit, vanishing FPR requires unbounded bits per key at nonzero recall.

False-positives only



Accept a superset of the **keys**.

False-negatives only



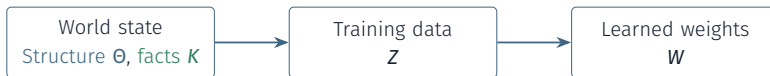
Accept only a subset of the **keys**.

*This also applies to LLM-induced binary decisions on **random facts**.

Role of memorization in learning; implications.

What memory is available for random facts?

- Our bound is on $I(K; W)$, information about facts K that is retained in model weights W .
- This “memory size” is NOT the raw parameter count.



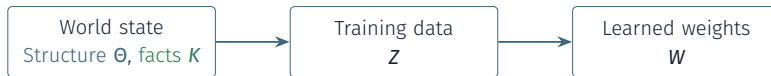
We identify two reasons why $I(W; K)$ is much smaller than the actual memory capacity of the language models.

Training does not retain everything

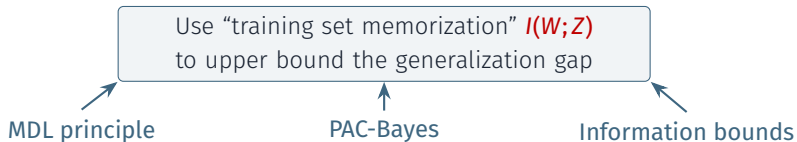
Regularization and compression can limit how much of Z reaches W .

Facts share the budget with structure

Syntax, rules and other structure are also stored in W .



Data processing inequality: $I(W; K) \leq I(W; K, \Theta) \leq I(W; Z)$.



Rissanen 1978; McAllester 1999; Catoni 2007; Xu & Raginsky 2017

- Weight decay / dropout / early stopping can **reduce information retained** from the training data.
- This bounds the generalization gap through multiple theories.

*The same pressure discards both **noise** in data and useful **long-tail facts**.* Dziugaite & Roy

2017; Lotfi et al. 2024; Feldman 2020.

Structure and random facts share an information budget



Assume K is independent of Θ : **random facts** cannot be derived from the structure.

Structure Θ

Syntax, logical rules, recurring regularities.

Random facts K

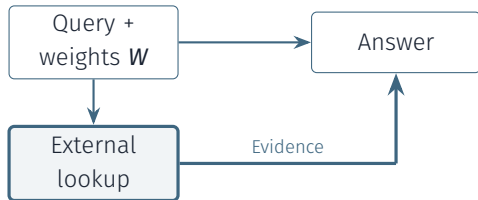
Arbitrary details that must be memorized.

Budget splitting

$$\underbrace{I(K; W)}_{\text{random facts}} \leq \underbrace{I(\Theta, K; W)}_{\text{total information}} - \underbrace{I(\Theta; W)}_{\text{structure}}.$$

*At a fixed total capacity, storing more structure leaves less room for **random facts**.*

Consistent with observed learning dynamics. [Pan et al. 2025; Gong et al. 2025; Arpit et al. 2017]



- **RAG:** retrieve relevant documents.
- **Agent lookup:** search the web, query databases, or call APIs.

Look up the fact instead of relying on the weights alone.

1. **Random-fact** memory admits a **membership-testing** model.
2. A **minimum KL divergence** gives the optimal asymptotic information cost.
3. The optimal log-loss channel assigns some **non-keys** the same score as **keys**.
4. LLM-based factual judgement cannot beat the filter bound – **false positives** are much **harder to eliminate** than false negatives.
5. Memory budget for random facts is smaller than one would think.
6. **Larger model / specialized training** to increase this memory; or use **external memory** to avoid this bottleneck entirely!

Thank you!