

"[LLMs] cannot be made reliable because you can't stop them from hallucinating."

YANN LECUN, 15 MAY 2026



Who won the wrestling at the Olympic games of 600BC?

Probably Hetoimokles of Sparta, but the record is not firm enough to say for certain. … This is from memory, not looked up.

*it has a level of uncertainty given its training data*

🔍 *web search: Hetoimokles of Sparta Olympic wrestling victories dates 600 BC 45th Olympiad wrestling victor*

*it decides to act, based on that level of uncertainty*

Hetoimokles of Sparta is the wrestler usually listed as the men's wrestling victor at the Games of 600 BC (the 45th Olympiad), though the date is a scholarly estimate. Pausanias records that Hetoimokles won five Olympic wrestling victories but gives no years.

*it has a level of uncertainty given its training + context*

*RedpointAI interview with Yann LeCun: https://www.youtube.com/watch?v=ngBraLDqzdI&t=2738s*
*Picture: wrestlers on an amphora in Madrid (MAN 32647); photo Mark Landon, CC BY 4.0. AI is Fable 5.2.*

A language model: maps (context, prompt) $\mapsto$ response

A world model:
A Gaussian process:
} map (observations $\mathcal{O}$, target $\mathcal{T}$) $\mapsto$ predictions $Y^*$

So let's look at uncertainty in a GP's prediction $Y^*$ at $\mathcal{T} = [x^*]$ given context $\mathcal{O} = [(x_1, y_1), \ldots, (x_n, y_n)]$.

## ANALOGY

How does the data in $\mathcal{O}$ affect our uncertainty about our predictions $Y^*$?

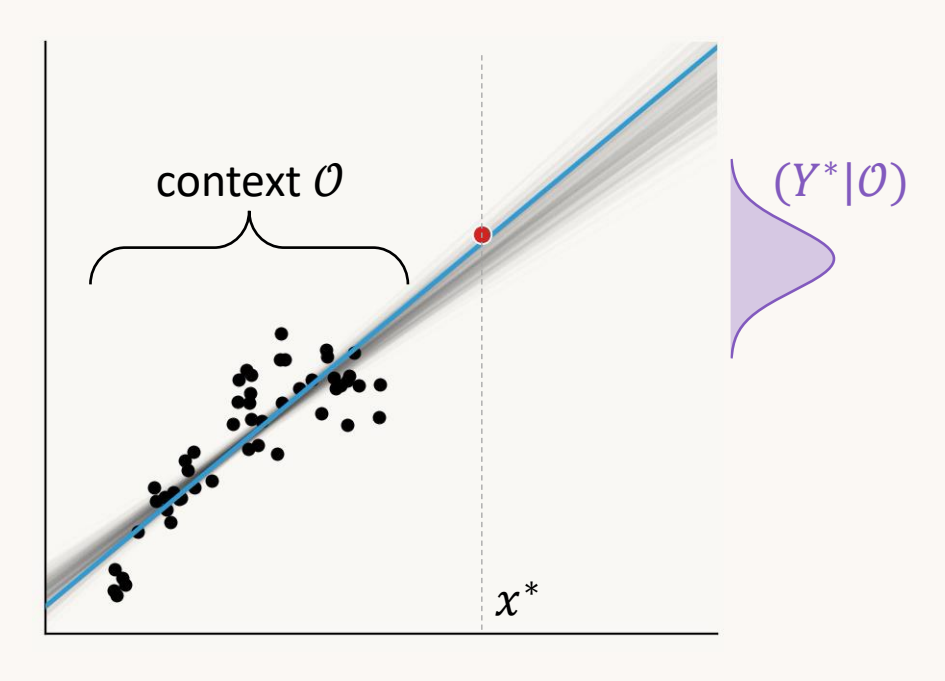
For a Gaussian process, we can work this out exactly.

## ANALOGY FAILURE

In a world model, it's only the context that has relevant data, and the weights embody the "conditional response" mechanism.

But in a language model, the weights additionally contain relevant data. *We'll return to this later.*

# Here's a Gaussian process, and a formula for its prediction uncertainty




Consider a Bayesian linear regression

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \sim N(0, \Sigma_0), \qquad Y^* \sim \alpha + \beta x^* + N(0, \sigma_0^2)$$

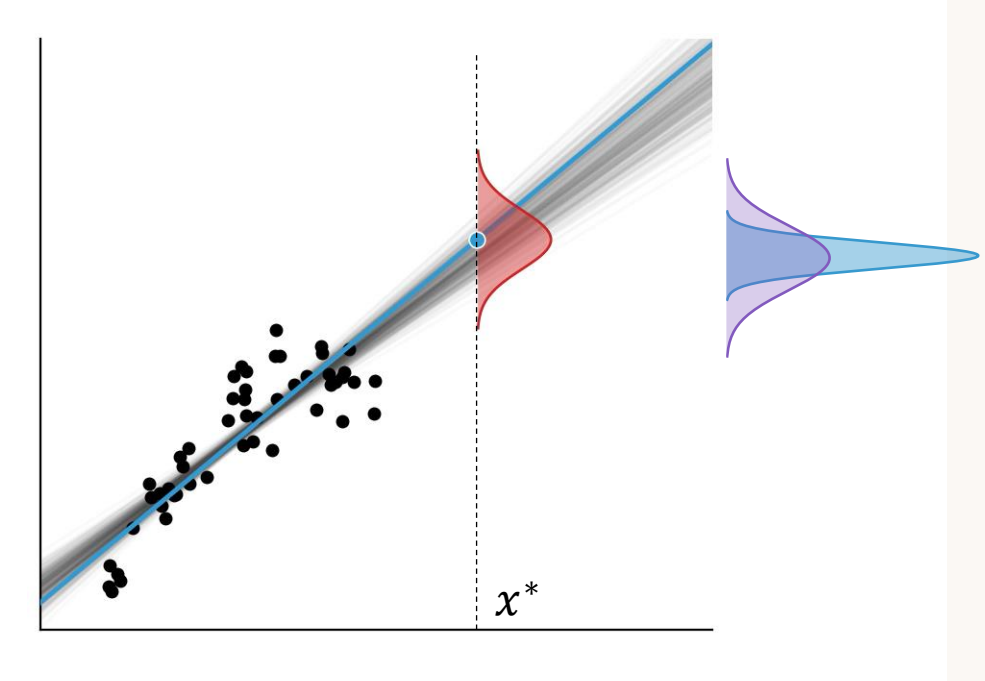
We can calculate $(Y^*|\mathcal{O})$.

Let $\mathcal{O} = [(x_1, y_1), \ldots, (x_n, y_n)]$. Then

$$\mathrm{Var}(Y^*|\mathcal{O}) = \mathrm{Var}(\alpha + \beta x^* | \mathcal{O}) + \sigma_0^2$$

$$= \begin{bmatrix}1\\ x^*\end{bmatrix}^T \left(\Sigma_0^{-1} + \frac{1}{\sigma_0^2} X^T X\right)^{-1} \begin{bmatrix}1\\ x^*\end{bmatrix} + \sigma_0^2, \qquad X = \begin{bmatrix}1 & x_1\\ \vdots & \vdots\\ 1 & x_n\end{bmatrix}$$

# Total uncertainty is made up of epistemic and aleatoric uncertainty




Consider a Bayesian linear regression

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \sim N(0, \Sigma_0), \qquad Y^* \sim \alpha + \beta x^* + N(0, \sigma_0^2)$$

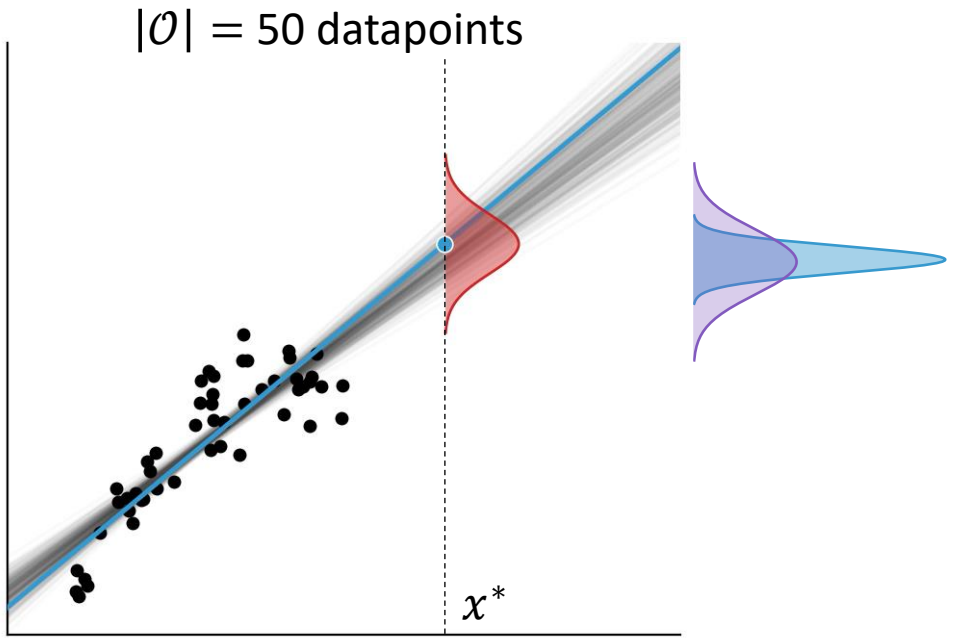
We can calculate $(Y^*|\mathcal{O})$.

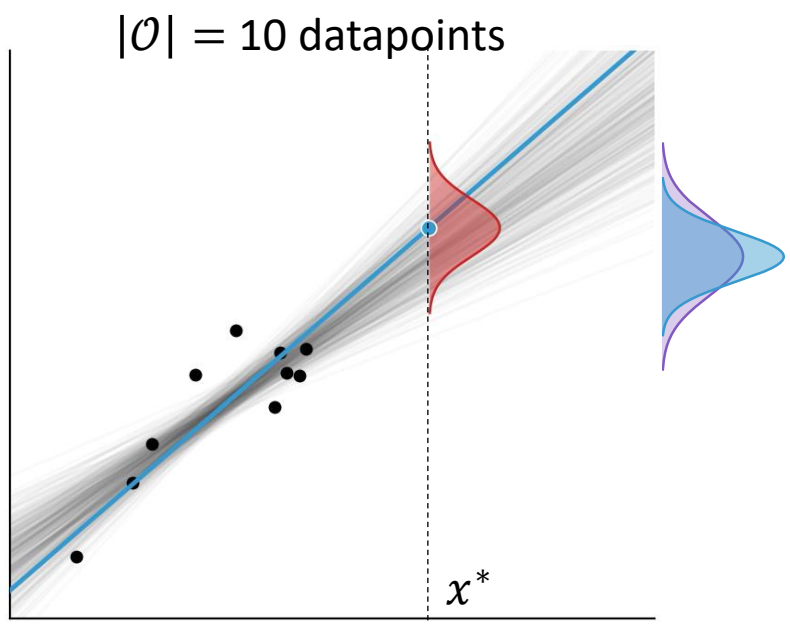
Let $\mathcal{O} = [(x_1, y_1), \ldots, (x_n, y_n)]$. Then

$$\mathrm{Var}(Y^*|\mathcal{O}) = \underbrace{\mathrm{Var}(\alpha + \beta x^*|\,\mathcal{O})}_{\text{epistemic}} + \underbrace{\sigma_0^2}_{\text{aleatoric}}$$

$$\underbrace{\qquad\qquad\qquad\qquad}_{\text{total}}$$

# *Epistemic uncertainty* is the part that depends on data




$$Y^* \sim \alpha + \beta x^* + N(0, \sigma_0^2)$$

$$\mathrm{Var}(Y^*|\mathcal{O}) = \mathrm{Var}(\alpha + \beta x^*|\,\mathcal{O}) + \sigma_0^2$$

**epistemic:** the part of the uncertainty that decreases as we get more context data

**aleatoric:** irreducible uncertainty

# If we force the model to give an answer, it'll hallucinate one

Suppose we've trained a show-ask machine in the usual way: $\min_{p^\theta} \mathbb{E}_{\mathcal{O},X^*} \mathrm{KL}(q_{Y^*|\mathcal{O},X^*} \,\|\, p^\theta_{Y^*;\mathcal{O},X^*})$

With enough capacity and the right training, it'll learn the perfect answer: $p^\theta_{Y^*;\mathcal{O},X^*} \approx q_{Y^*|\mathcal{O},X^*}$

In other words, it correctly **learns the total uncertainty** of $Y^*$.

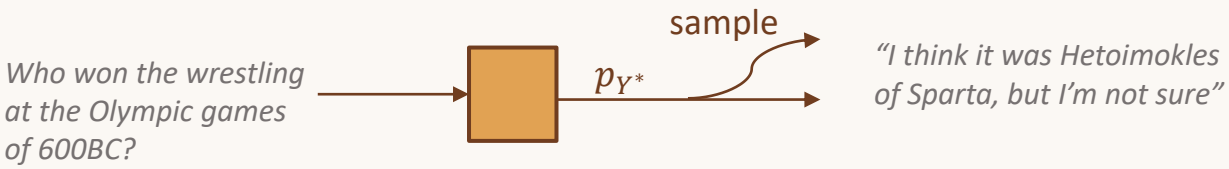
But what does it do with that uncertainty?

If all we've trained is a sampler, then it has no choice but to hallucinate an answer:



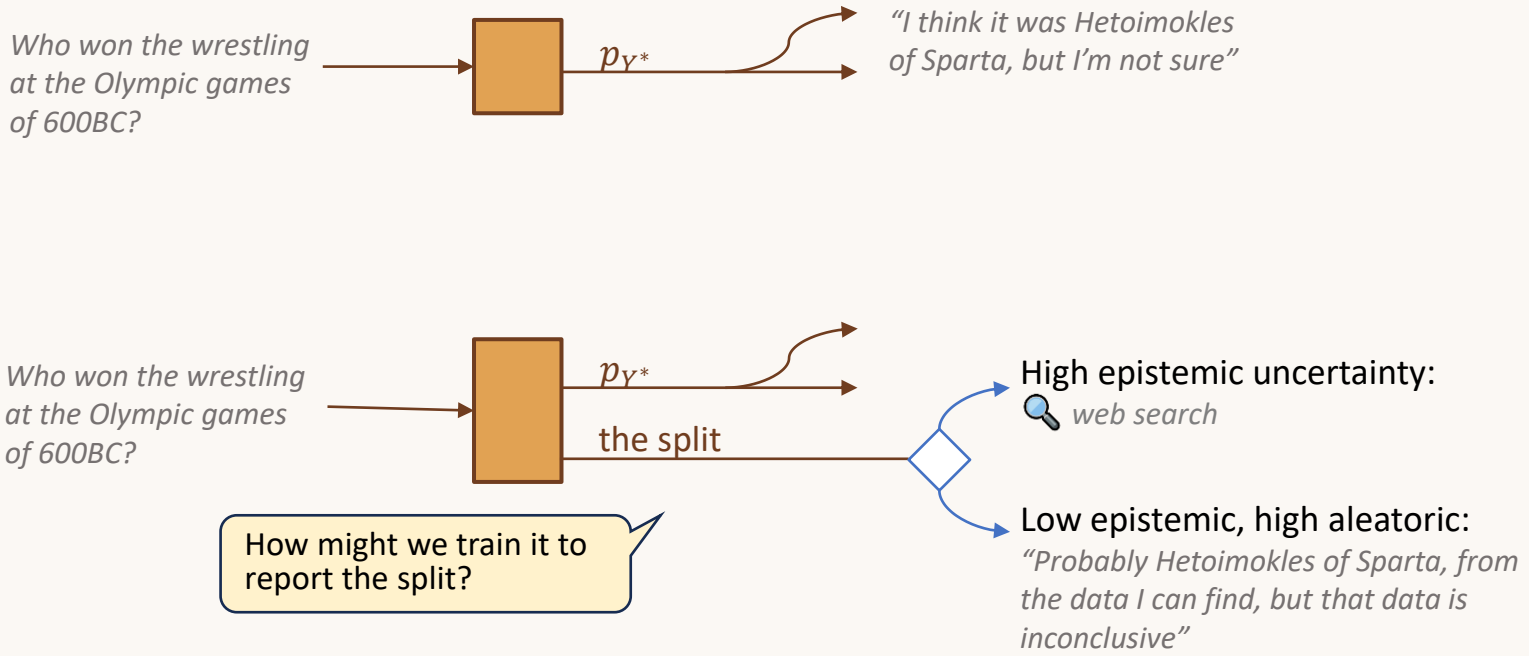

We need to build or train it to **tell us when it's not sure:**

How might we do this?




Abstention in language models: Kalai et al. "Why language models hallucinate", 2025

# To decide what to do next, we want to know how total uncertainty is split between epistemic and aleatoric.



# It’s easy to build a show-ask machine that reports epistemic uncertainty

**Oracle**
(a Gaussian process)

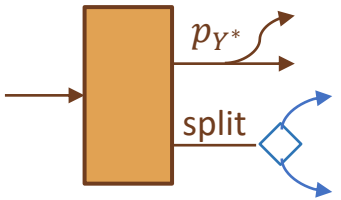
*simulate*

**Training dataset**

| *predictors* | | *ground truth response* | | |
|---|---|---|---|---|
| $\mathcal{O}$ | $x^*$ | a sample $y^*$ | total uncertainty $\mathrm{Var}(Y^*\|\mathcal{O})$ | epistemic uncertainty given $\mathcal{O}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

*train*

**A fitted show-ask machine**




Last week: an algebraic Gaussian process kernel ≡ a dataset of show-ask responses

Today: Bayesian concepts of epistemic *versus* aleatoric uncertainty can also be learnt

# When we fine-tune an LLM with preference data, perhaps it achieves the same thing as an oracle?

Our oracle gave an explicit indication of uncertainty, and the model can learn it directly

Q. How do we know our uncertainty comes from the *right prior*?

*Is there an objective way to think about "the right uncertainty"?*

If raters prefer answers that acknowledge uncertainty (in cases where there should be uncertainty), then the model will learn to report uncertainty.

Q. How do we know it's the *right* uncertainty?

*Uncertainty is with respect to a knowledge base.*

*If the human rater's knowledge is different to the LLM's training data, how can the rater say how uncertain it should be?*

John Schulman interview:
https://news.berkeley.edu/2023/04/24/berkeley-talks-transcript-chatgpt-developer-john-schulman/

"During finetuning, however, models are trained to avoid [hallucination] when acting in the Assistant character."
Lindsey et al. *On the biology of an LLM*, 2025