

Uncertainty Beyond Confidence

Semantic and Claim-Level Risk

Advanced LLMs – Responsible LLMs

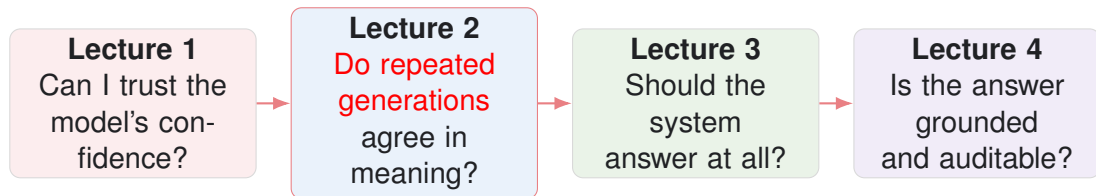


Sourish Dasgupta

Associate Professor, DAU, Gandhinagar

Lecture 2 of the Responsible Uncertainty block

Roadmap



Question: a single confidence score is useful – but
what different answers / meanings might it be hiding?



Part I. Beyond Scalar Confidence



Student

“I was ill and missed an assignment deadline by five days. I have a medical certificate. Am I automatically entitled to an extension?”

Hypothetical LLM assistant

“Yes. Students with a valid medical certificate are automatically entitled to a seven-day extension.”

After Lecture 1 ...

The system has a **validated, calibrated** answer-level score:

$$S(x, a) = 0.72.$$

Among **similar** answers receiving scores near 0.72, about 72% were correct.

PROBLEM: Even a calibrated 0.72 does not tell us **which alternative meanings** are competing.



Same **scalar confidence**; very different hidden ambiguity

Suppose two systems assign the displayed answer the same calibrated score:

$$S(x, a) = 0.72.$$

Now probe each system by repeated generation.

System A: one meaning dominates

Empirical distribution over meanings:

0.80, 0.10, 0.10.

System B: several meanings compete

Empirical distribution over meanings:

0.40, 0.35, 0.25.

The score attached to one answer $\neq$ **shape** of meaning distribution.



Recap:

An auto-regressive LLM produces a **local** next-token distribution:

$$p_{\theta}(y_t \mid x, y_{<t}).$$

At every generation step, the current prefix changes:

$$(x, y_{<t}) \longrightarrow p_{\theta}(y_t \mid x, y_{<t}).$$

So a complete answer is built by repeatedly making one next-token **choice** after another.

Model encoder supplies probabilities
Decoding procedure decides final token choice.



Setup: Decoding procedure defines the **repeated-generation** experiment

Let

$$\delta = (T, \text{top-}p, \text{top-}k, \dots)$$

denote the decoding settings.

For fixed **model** θ , prompt x , and **decoding rule** δ :

$$q_{\theta, \delta}(a \mid x)$$

denotes the induced **distribution** over **complete responses**.

One chat response is one draw from this distribution. **Re-running** the same controlled experiment **can reveal other responses**.

Semantic uncertainty depends on both
the **model** and the **decoding probe**.



Part II. Controlled Repeated Generations



Before sampling: what exactly stays fixed?

If repeated generations are to mean anything, we need a **controlled experiment**.

Held fixed (Controlled)

model weights θ
prompt x
system instruction
temperature T
top- p / top- k
other decoding settings

Allowed to change

the random token draws made during
stochastic decoding
 $\Rightarrow$ the realized complete answer



What exactly do we mean by a **sample**?

One run of stochastic decoding produces one complete answer:

$$a^{(m)} = (y_1^{(m)}, y_2^{(m)}, \dots, y_{T_m}^{(m)}).$$

Under the **fixed** decoding setup:

$$a^{(m)} \sim q_{\theta, \delta}(\cdot \mid x).$$

A **sample** = 1 **complete** generated answer from **same model** on **same** prompt under **same decoding settings** via fresh random token draws.

Sample $\neq$ token; sample $\neq$ prompt; sample $\neq$ confidence score.



Five **controlled samples** from the same **LLM**

Sample	Generated answer
$a^{(1)}$	There is no automatic extension; the course instructor decides case by case.
$a^{(2)}$	A medical certificate supports the request, but instructor approval is still required.
$a^{(3)}$	The instructor may grant an extension after considering the medical certificate.
$a^{(4)}$	A valid medical certificate automatically gives the student a seven-day extension.
$a^{(5)}$	Medical extensions must be approved by the academic office .



First reaction: “The answers are different”

Yes—but **different in what sense?**

Surface variation

“The instructor decides.”

“Instructor approval is required.”

“The instructor may grant it.”

Potentially **same answer**.

Meaning variation

“The instructor decides.”

“Seven days are automatic.”

“The academic office decides.”

Different policy claims.

Uncertainty over strings $\neq$ uncertainty over meanings.



Part III. Strings $\neq$ Meanings



Different strings can be the **same answer**

Compare:

A: "There is no automatic extension; the instructor decides case by case."

B: "A medical certificate supports the request, but instructor approval is still required."

C: "The instructor may grant an extension after considering the medical certificate."

All three **commit** to same **proposition**:

No automatic entitlement; instructor approval is needed.



Semantic equivalence $\equiv$ same commitment

For same question x ,

$$a_i \sim_x a_j$$

when answers a_i and a_j express the **same substantive answer** to x .

NOT just:

- ▶ sharing words;
- ▶ having similar sentence embeddings;
- ▶ sounding stylistically similar.

We care about what **proposition** the answer **commits** to.



NT: equivalence is question-conditioned

The phrase

“The instructor decides.”

can answer many different questions. For our policy question, it means something like

“there is no automatic entitlement; approval is discretionary.”

So semantic equivalence is better written as

$$\boxed{a_i \sim_x a_j} \neq \boxed{a_i \sim a_j}$$



How to detect "commitment"? Bidirectional entailment

Ask whether two answers entail each other in the context of the question.

$$a_i \models_x a_j$$

– given question x , answer a_i implies answer a_j . Then *bidirectionally* (mutual entailment):

$$a_i \sim_x a_j \iff (a_i \models_x a_j) \wedge (a_j \models_x a_i).$$

NOTE: Mutual entailment is **approximation** for semantic equivalence – not a perfect oracle.



Why one-way entailment is **not** equivalence

A: “Instructor approval is required.”

B: “Instructor approval is required **and** the request must be filed within 48 hours.”

Then:

$$B \models_x A$$

but:

$$A \not\models_x B.$$

B makes an **additional commitment. One-way entailment cannot certify “same meaning”.**



Failure 1: the NLI / judge model can misread entailment

Suppose two sampled answers say:

Answer A

“The instructor **may** grant an extension when valid medical evidence is provided.”

Answer B

“The instructor **must** grant an extension when valid medical evidence is provided.”

may grant $\neq$ must grant; $\rightarrow B \Rightarrow A$; **but NOT** $A \Rightarrow B$.



Failure 2: pairwise “same meaning” decisions can be inconsistent

Consider three generated answers:

A

“The **instructor** must approve the extension.”

B

“The **instructor or course coordinator** must approve it.”

C

“The **course coordinator** must approve the extension.”

A practical semantic judge might decide

$$A \sim B \quad \text{and} \quad B \sim C,$$

because each adjacent pair overlaps strongly.

Pairwise semantic decisions are produced by an



Failure 2 continued: the implied relation need not be transitive

From the previous slide, the judge gave

$$A \sim B, \quad B \sim C.$$

If $\sim$ were a true equivalence relation, transitivity would imply

$$A \sim C.$$

But here

$$A \not\sim C$$

because A and C assign approval authority to **different people**.



Failure 3: subtle qualifiers can disappear

Consider a Semantic Judge focusing on only the **dominant meaning**:

Answer A

“Students with **valid medical evidence** are **normally** granted an extension.”

Answer B

“Students with **valid medical evidence** are **granted an extension**.”

Only one word changes the commitment – *exceptions* vs. *categorical*.

Small qualifiers such as ***normally, may, unless, only if, or subject to approval*** can carry the part of the meaning that matters most.



Failure 4: one long answer can contain several propositions

Suppose we obtain two complete generations.

Answer A

“A medical extension is possible. The **instructor** must approve it, and the request must be submitted within **48 hours**.”

Answer B

“A medical extension is possible. The **instructor** must approve it, and the request may be submitted within **seven days**.”

Most of the two answers agree.



Failure 4: one long answer can contain several propositions

If we compare the **whole responses**, a semantic judge may decide:

$$A \sim B.$$

But decompose them into claims:

Claim	A	B
Medical extension is possible	same	same
Instructor approval is required	same	same
Submission deadline	48 hours	7 days

Can hide a disagreement on one **decision-critical claim**.



(Fingers crossed) Cluster the five samples by **meaning**

A reasonable clustering is:

$$C_1 = \{a^{(1)}, a^{(2)}, a^{(3)}\}$$

C_1 : instructor approval / discretion

$$C_2 = \{a^{(4)}\}, \quad C_2 : \text{automatic seven-day extension}$$

$$C_3 = \{a^{(5)}\}, \quad C_3 : \text{academic-office approval}$$

5 different strings $\implies$ 3 semantic alternatives.



As we saw, Semantic Clustering can itself be imperfect

The clustering procedure can make mistakes. **Potential failures:**

- ▶ the NLI / judge model can misread entailment;
- ▶ practical pairwise equivalence need not be perfectly transitive;
- ▶ subtle qualifiers can disappear;
- ▶ long answers may contain several propositions at once.

Semantic entropy inherits error from the procedure used to define fine semantic classes. This motivates claim-level uncertainty.



Part IV. Semantic Entropy



Meaning Distribution Estimate

We have $K = 5$ samples and three semantic clusters:

$$n_1 = 3, \quad n_2 = 1, \quad n_3 = 1.$$

The simplest empirical probability estimate is:

$$\hat{p}(C_k) = \frac{n_k}{K}.$$

So:

$$\hat{p}(C_1) = 0.6, \quad \hat{p}(C_2) = 0.2, \quad \hat{p}(C_3) = 0.2.$$

We now have a **distribution over meanings.**



What should a semantic uncertainty measure do?

Compare 3 alternate cases:

$$(1, 0, 0), \quad (0.6, 0.2, 0.2), \quad \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

We want uncertainty to be:

- ▶ **smallest** for $(1, 0, 0)$: one meaning dominates;
- ▶ larger for $(0.6, 0.2, 0.2)$;
- ▶ **largest** when three meanings compete equally.

How **spread out** probability mass is
across meanings – **Semantic Entropy**



Entropy, step 1: quantify the surprise of one meaning

If an event has probability p , define its information content as

$$I(p) = -\log p.$$

Using natural logarithms:

$$I(1) = 0,$$

$$I(0.5) = -\log 0.5 \approx 0.693,$$

$$I(0.1) = -\log 0.1 \approx 2.303.$$

Common events are unsurprising; rare events carry more surprise.



Entropy, step 2: average the surprise

If semantic outcome C_k occurs with probability p_k , its surprise is

$$I(p_k) = -\log p_k.$$

The expected surprise is

$$\sum_k p_k I(p_k).$$

Substitute $I(p_k)$:

$$\sum_k p_k (-\log p_k).$$

Therefore

$$H(C) = -\sum_k p_k \log p_k.$$

Entropy = expected surprise under the distribution.



Entropy, step 3: make the random variable **meaning**

For semantic classes $C_1, \dots, C_M$:

$$H_{\text{sem}}(x) = - \sum_{k=1}^M p(C_k | x) \log p(C_k | x).$$

Same entropy formula; new outcome space: **strings** $\rightarrow$ **meanings**.

Farquhar et al., "Detecting hallucinations in large language models using semantic entropy", Nature, 2024.



Compute our toy semantic entropy: first term

Our empirical semantic distribution is

$$(0.6, 0.2, 0.2).$$

The first contribution is

$$-0.6 \log 0.6.$$

Since

$$\log 0.6 \approx -0.5108,$$

we get

$$-0.6 \log 0.6 \approx 0.3065.$$

We are computing the contribution of each semantic alternative one at a time.



Compute our toy semantic entropy: remaining terms

For each of the two 0.2 meanings:

$$-0.2 \log 0.2.$$

Since

$$\log 0.2 \approx -1.6094,$$

we obtain

$$-0.2 \log 0.2 \approx 0.3219.$$

There are two such terms.



Add the contributions

Therefore

$$H_{\text{sem}} \approx 0.3065 + 0.3219 + 0.3219.$$

Hence

$$H_{\text{sem}} \approx 0.9503.$$

What does 0.9503 mean by itself?



Putting 0.9503 in context

For one certain meaning:

$$(1, 0, 0) \implies H = 0.$$

For three equally likely meanings:

$$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \implies H = \log 3 \approx 1.099.$$

Our case:

$$H_{\text{sem}} \approx 0.950.$$

Sampled meanings are fairly dispersed:
two substantial alternatives remain.



Important: our 0.6, 0.2, 0.2 came from **sample frequencies**

We used:

$$\hat{p}(C_k) = \frac{n_k}{K}.$$

With only $K = 5$ samples, this is a Monte Carlo estimate. If we repeat the experiment, the counts can change.

$H_{\text{sem}} \approx 0.9503$ is therefore an *empirical semantic-entropy estimate*.

Finite sample frequency $\neq$ exact semantic probability mass.



What exactly is our $\hat{p}(C_1) = 3/5$ trying to estimate?

Recall:

$$C_1 = \{a^{(1)}, a^{(2)}, a^{(3)}\}, \quad C_2 = \{a^{(4)}\}, \quad C_3 = \{a^{(5)}\}.$$

Hence

$$\hat{p}(C_1) = \frac{3}{5} = 0.6.$$

But $3/5$ is only what happened in **five random generations**.

What **population quantity** is this empirical frequency trying to estimate?



The population behind the five samples

Earlier we fixed the model, prompt, and decoding settings and defined

$$q_{\theta, \delta}(a \mid x)$$

as the induced distribution over **complete responses**. Our five outputs were random draws:

$$a^{(1)}, \dots, a^{(5)} \sim q_{\theta, \delta}(\cdot \mid x).$$

For any such complete response

$$a \in C_1$$

“The sampled complete response expresses the semantic answer *instructor approval / discretion*.”



Semantic probability = total probability mass of the class

Every complete wording a has probability

$$q_{\theta,\delta}(a \mid x).$$

The probability of semantic class C_k is therefore

$$q_{\theta,\delta}(C_k \mid x) = \sum_{a \in C_k} q_{\theta,\delta}(a \mid x).$$

Semantic probability **collapses para-phrases** before measuring uncertainty.



PROBLEM: exact semantic mass is usually **intractable**

To evaluate

$$q_{\theta,\delta}(C_k | x) = \sum_{a \in C_k} q_{\theta,\delta}(a | x),$$

we would need to consider an **enormous space** of possible responses.

For every possible wording a , we would need to ask:

1. Could the decoding process generate a ?
2. Does a belong to semantic class C_k ?
3. What is $q_{\theta,\delta}(a | x)$?

We cannot enumerate all paraphrases.



Hence, sampling...

Instead of enumerating the full response space, draw

$$a^{(1)}, \dots, a^{(K)} \sim q_{\theta, \delta}(\cdot \mid x).$$

After semantic clustering, count how many samples fall in each class:

$$n_k = \#\{m : a^{(m)} \in C_k\}.$$

Then use the empirical frequency

$$\hat{p}(C_k) = \frac{n_k}{K}.$$

Repeated generation makes the otherwise intractable semantic output space empirically inspectable.



Back to our example...

We had

$$K = 5, \quad n_1 = 3, \quad n_2 = 1, \quad n_3 = 1.$$

Therefore

$$\hat{p}(C_1) = \frac{3}{5} = 0.6, \quad \hat{p}(C_2) = \frac{1}{5} = 0.2, \quad \hat{p}(C_3) = \frac{1}{5} = 0.2.$$

This is the empirical semantic distribution we used in

$$\hat{H}_{\text{sem}} = - \sum_k \hat{p}(C_k) \log \hat{p}(C_k).$$

Our 0.6, 0.2, 0.2 is a sample-based estimate of the underlying semantic distribution.



Why is n_k/K a sensible estimator? – step 1

For sample m , define

$$Z_m = \mathbf{1}\{a^{(m)} \in C_k\}.$$

So

$$Z_m = \begin{cases} 1, & \text{sample } m \text{ lands in semantic class } C_k, \\ 0, & \text{otherwise.} \end{cases}$$

Because $a^{(m)}$ is drawn from $q_{\theta,\delta}$,

$$\mathbb{E}[Z_m] = q_{\theta,\delta}(C_k \mid x).$$

Each indicator is a Bernoulli observation whose success probability is the semantic-class probability.



Why is n_k/K a sensible estimator? – step 2

The sample average of those indicators is

$$\frac{1}{K} \sum_{m=1}^K z_m.$$

But the number of indicators equal to 1 is exactly n_k , hence

$$\frac{1}{K} \sum_{m=1}^K z_m = \frac{n_k}{K} = \hat{p}(C_k).$$

Under independent repeated generations, the law of large numbers gives

$$\hat{p}(C_k) \xrightarrow{K \rightarrow \infty} q_{\theta, \delta}(C_k \mid x).$$



NT: measured uncertainty also depends on the decoding procedure

Suppose temperature is pushed toward zero:

$$T \rightarrow 0.$$

Repeated runs become concentrated near the highest-logit path. At larger T , lower-probability alternatives are explored more often. So:

observed sample diversity = model + decoding probe.

Low sampled disagreement – a concentrated model?
Or, a probe that simply did not explore alternatives?



Part V. Stable $\neq$ True



PROBLEM: **low semantic entropy** can still be confidently wrong

Suppose all five generations say essentially:

“A valid medical certificate automatically gives a seven-day extension.”

There is only one semantic class:

$$P(C_1) = 1.$$

Therefore:

$$H_{\text{sem}} = 0.$$

The model is maximally **consistent**. We still have not established that it is **correct**.



Now reveal the evidence

Suppose the actual policy says:

Policy: “Extensions may be granted by the course instructor upon consideration of supporting documentation.”

The repeated answer – automatic seven-day entitlement – is unsupported. Yet:

$$H_{\text{sem}} = 0.$$

Semantic uncertainty $\neq$ truth. A model can be stably wrong.



Moral of the story...

Semantic entropy answers: *“Are different meanings competing?”*

It does **not** answer: *“Is the dominant meaning true or grounded?”*



A second problem: one long answer contains several claims

Consider:

“A medical extension is possible. Instructor approval is required. The request must be submitted within 48 hours.”

At least three propositions:

c_1 : medical extension is possible,

c_2 : instructor approval is required,

c_3 : request must be filed within 48 hours.

One answer-level uncertainty number can hide a **single unstable claim.**



Repeated generations may agree on one claim and disagree on another

Sample	Who approves?	Deadline stated?
1	instructor	48 hours
2	instructor	7 days
3	instructor	not specified
4	instructor	48 hours
5	instructor	3 days

For the approval claim, all samples agree. For the deadline claim, they do not.

**Uncertainty can be localized to a claim,
not only attached to the whole response.**



Move from one answer score to a **claim-level uncertainty profile**

Suppose an answer contains claims:

$$c_1, c_2, \dots, c_L.$$

After extracting and matching corresponding claims across repeated generations, estimate uncertainty u_j for each claim. Collect:

$$\mathbf{u}(a) = (u_1, u_2, \dots, u_L).$$

Example:

$$\mathbf{u}(a) = (0.05, 0.03, 0.91).$$

This is a system-design pattern: localize uncertainty to claims instead of hiding everything inside one global score.



Average claim uncertainty can hide one dangerous claim

A simple aggregation is:

$$R_{\text{avg}}(a) = \frac{1}{L} \sum_{j=1}^L u_j.$$

For:

$$\mathbf{u} = (0.05, 0.03, 0.91),$$

we get:

$$R_{\text{avg}} = \frac{0.05 + 0.03 + 0.91}{3} \approx 0.33.$$

Overall uncertainty “0.33” looks moderate – yet one claim has uncertainty 0.91.



For high-stakes use, the **worst claim** may matter more

Worst-claim aggregation:

$$R_{\max}(a) = \max_j u_j.$$

For our example:

$$R_{\max} = 0.91.$$

Or weight claims by importance:

$$R_w(a) = \sum_j w_j u_j, \quad w_j \geq 0, \quad \sum_j w_j = 1.$$

Aggregation is a **decision-design choice**; the right rule depends on the consequences of each claim being wrong.



Uncertainty and evidence support are **different axes**

	Evidence supports claim	Evidence does not support claim
Low semantic uncertainty	Stable + supported	Stable unsupported answer
High semantic uncertainty	Competing interpretations	Uncertain + unsupported

Semantic uncertainty: “Are meanings competing?”

Grounding: “Does evidence support the claim?”



Part VI. Utility



Can we check one answer *without* external evidence?

Suppose the model has already produced one answer:

Main response to be checked: $a^{(0)}$

“A valid medical certificate automatically gives a seven-day extension.”

Is it factual? Imagine we are **out of luck**:

- ▶ no trusted policy database;
- ▶ no gold reference answer;
- ▶ no access to the model's logits or internal probabilities;
- ▶ only black-box access – we can submit the same question again.

Can the model's own alternative generations act as an internal **consistency check?**



Self Checking

Keep the already-produced response

$a^{(0)}$ = main response being checked (*target*).

Then generate fresh responses to the **same** prompt under the **same** controlled setup:

$$a^{(1)}, a^{(2)}, \dots, a^{(K)} \sim q_{\theta, \delta}(\cdot \mid x).$$



NOTE: Fresh responses $\neq$ ground truth.



Method 1: SelfCheckGPT [EMNLP, 2023]

A main response may contain several *target sentences*:

$$a^{(0)} = (s_1^{(0)}, s_2^{(0)}, \dots, s_J^{(0)}).$$

Sentence-level checking - Intuition

If the model has strong internal support for a fact, independently sampled responses should tend to be compatible with that sentence.

If the sentence contains a fragile or confabulated detail, fresh samples are more likely to **omit, vary, or contradict** it.

No external database is required; no access to logits is required.

Manakul, Liusie, and Gales, "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models", EMNLP 2023.



Example

Suppose the main response $a^{(0)}$ contains $s_1^{(0)}$:

“A valid medical certificate automatically gives a seven-day extension.”

Generate 4 fresh responses to the same university-policy question.

“Is $s_1^{(0)}$ supported by the model’s fresh sampled responses?”



Cross-examine $s_1^{(0)}$ with fresh generations

Sample Generated statement

- $a^{(1)}$ The course instructor decides whether an extension is granted.
 - $a^{(2)}$ Medical evidence supports the request, but instructor approval is required.
 - $a^{(3)}$ The instructor may grant an extension after reviewing the certificate.
 - $a^{(4)}$ Medical extensions are handled case by case.
-

All four fresh responses **avoid** the specific commitment

“automatic seven-day entitlement.”

$s_1^{(0)}$ is **in tension** with the model's own alternative generations



SelfCheck score – step 1: 1 sentence vs. 1 sampled response

Define an **inconsistency** function

$$d\left(s_j^{(0)}, a^{(m)}\right) \in [0, 1].$$

How ***incompatible*** is sampled response $a^{(m)}$
with sentence $s_j^{(0)}$ from the main answer?



SelfCheck score – step 1 continued

Interpret the comparison score as an **inconsistency** scale:

$d \approx 0$: **compatible**

Sentence being checked:

“Instructor approval is required.”

Sample says:

“The instructor decides whether to approve the extension.”

$d \approx 1$: **strong tension**

Sentence being checked:

“Automatic seven-day extension.”

Sample says:

“There is no automatic rule; instructor approval is required.”

How to implement d ?



Consistency (d) can have multiple interpretations

The SelfCheckGPT paper explores several mechanisms:

- ▶ n-gram / lexical consistency;
- ▶ BERTScore-style semantic similarity;
- ▶ question-answering based consistency;
- ▶ NLI-style entailment / contradiction;
- ▶ prompted LLM judgments.

Slightly different versions of:

“Do the sampled responses support this sentence from the main answer?”



SelfCheck score – step 2: aggregate across fresh samples

With K sampled responses,

$$U_{\text{SC}}(s_j^{(0)}) = \frac{1}{K} \sum_{m=1}^K d(s_j^{(0)}, a^{(m)}) .$$

Then:

- ▶ $U_{\text{SC}}(s_j^{(0)})$ small $\Rightarrow$ the sentence agrees with most fresh samples;
- ▶ $U_{\text{SC}}(s_j^{(0)})$ large $\Rightarrow$ the sentence is poorly supported / unstable relative to them.



Why might SelfCheck work? Two qualitatively different regimes

Stable internal support

Sentence being checked:

"Instructor approval is required."

Fresh samples:

"Instructor decides."

"Approval is required."

"Instructor may grant it."

Low inconsistency

Fragile detail

Sentence being checked:

"Automatic seven-day extension."

Fresh samples:

"Instructor decides."

"No automatic rule."

"Academic office approves."

High inconsistency

**SelfCheck turns stochastic instability
around a sentence into a **warning signal**.**



But SelfCheck has the same blind spot: **stable hallucination**

Suppose the model has strongly learned the **wrong association**

medical certificate $\rightarrow$ automatic seven-day extension.

- Main response and the fresh sampled responses repeatedly express essentially the **same false statement**, giving

$$U_{\text{SC}}(s_j^{(0)}) \approx 0$$

although the actual policy contradicts the sentence.

SelfCheck is not independent evidence that the sentence is true.



Self-Checking and Semantic Entropy ask **different questions**

	Self-Checking	Semantic entropy
Starting point	one sentence $s_j^{(0)}$ from a main response $a^{(0)}$	the set of sampled complete responses
Question	Do fresh samples support / contradict this sentence ?	How dispersed is probability mass across different meanings ?
Construction	sentence $\leftrightarrow$ sampled responses	samples $\rightarrow$ semantic classes $\rightarrow$ distribution
Indicator	sentence-level inconsistency / hallucination signal	distribution-level semantic uncertainty
Shared limitation	stable wrong beliefs can look safe – <i>cannot replace external grounding</i>	



Practical cost: repeated generation is expensive

Suppose we use $K = 10$ samples, each roughly 300 output tokens.

Generation alone is roughly $10\times$ *the decoding cost of one response*.

If semantic clustering compares all pairs:

$$\binom{K}{2} = 45, \quad \text{so naive comparison is } O(K^2).$$

Moral-of-the-story: Good uncertainty signals **can be too expensive!**



Can one generation predict semantic uncertainty?

A cheaper system can try to learn

$$\mathbf{h} \xrightarrow{g_\phi} \hat{H}_{\text{sem.}}$$

Here $\mathbf{h}$ is an internal representation from one generation, and g_ϕ is a learned probe.

The idea: use expensive semantic entropy offline as supervision, then approximate it cheaply from model representations at inference time.

Kossen et al., "Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs," ICML 2024 Workshop on Foundation Models in the Wild, 2024.



What should a Responsible LLM **do** with semantic uncertainty?

Observed pattern	Possible system action
One stable meaning, low stakes	answer normally
Several competing meanings	retrieve more evidence or ask clarification
One particular claim is unstable	verify that claim specifically
High uncertainty + high consequence	abstain or defer to a human
Low uncertainty but no supporting evidence	<i>do not infer correctness</i> ; verify grounding



What semantic uncertainty can tell us – and what it **cannot**

It can help reveal

- competing semantic answers
- unstable claims
- hidden disagreement behind one scalar
- queries that deserve verification

It does not by itself prove

- truth
- factual support
- visual grounding
- source correctness
- causal reliability

Low semantic uncertainty is stability $\neq$ certificate of correctness.



So far...

We now understand	Still unresolved
what a repeated-generation sample is	when the system should abstain
why different strings can express the same meaning	how to choose an answer / abstain threshold
semantic equivalence and clustering	how to provide coverage guarantees
semantic entropy	whether evidence actually grounds a claim
claim-level uncertainty	judge / verifier / agent uncertainty



One final check before we leave

Suppose tomorrow an LLM gives five nearly identical responses to the same question.

Before saying “the model is certain”, ask:

1. **What was fixed?** Same model, prompt, system instruction, decoding settings?
2. **What is one sample?** One complete stochastic generation?
3. **Do outputs agree only in wording – or in meaning?**
4. **How were semantic classes constructed?**
5. **Could the dominant meaning still be wrong?**
6. **Which individual claims remain unstable?**

If these are unanswered, “five similar generations” should not carry much authority.



Decision

Suppose semantic uncertainty is high.

What should the system do?

answer | **ask** | **retrieve** | **abstain** | **defer**

Two questions now become unavoidable:

1. How much uncertainty is too much?
2. Can we control the risk of the answers we choose to keep?

Lecture 3 turns uncertainty from a measurement problem into a decision problem.



Lecture 2 takeaway

Uncertainty lives over meanings.

But consistency is still not grounding.

Next:

Safe Non-Answering

Abstention, Selective Generation, and Conformal Guarantees