PAPER READING

Rethinking Self-Consistency for Hallucination Detection in Large Language Models

Sha Liu

STAR Group

May 9,2025

Overview

Definition

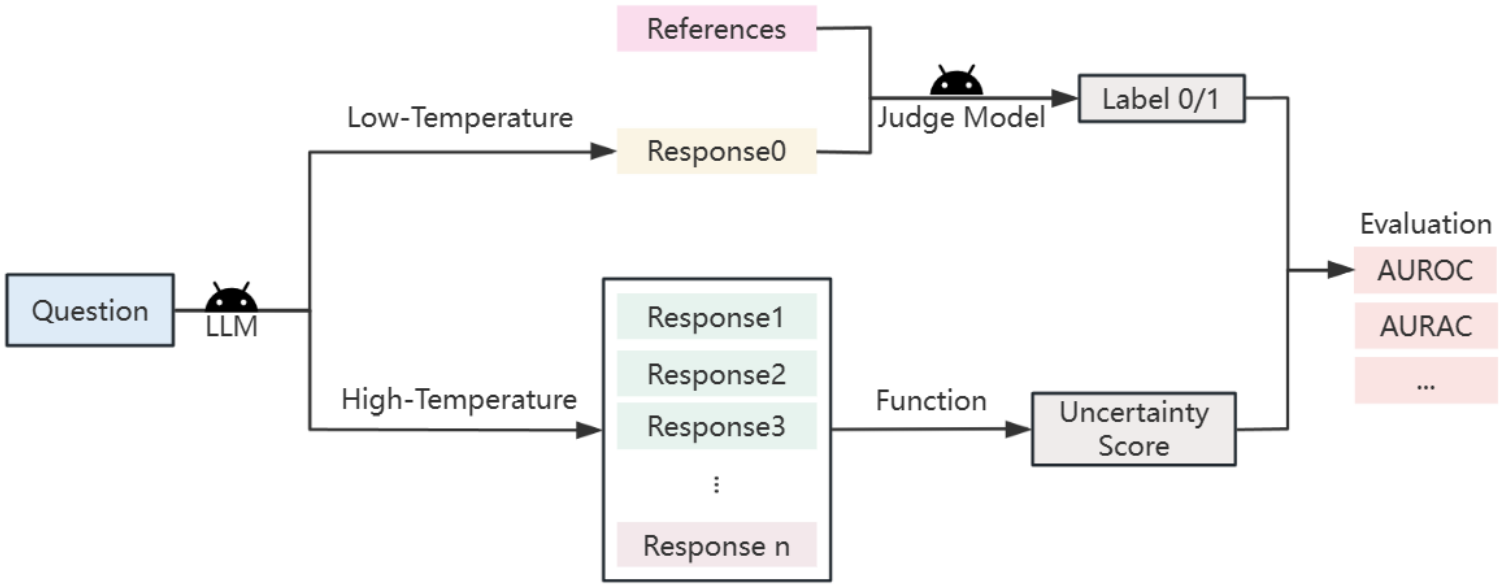


Figure: Overview of self-consistency.

Methods Leveraging Self-Consistency

- **Mean Pairwise Distance (MPD)** Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. (2023.03, EMNLP, 378 Citations)
- **Semantic Entropy (SE)** Detecting hallucinations in large language models using semantic entropy. (2023.02, Nature, 238 Citations)
- **Sum of Eigenvalues of the Graph Laplacian (EigV)**
- **Eccentricity (Ecc)** Generating with confidence: Uncertainty quantification for black-box large language models. (2023.05, TMLR, 121 Citations)
- **KLE** Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. (2024.05, NeurIPS, 21 Citations)

Mean Pairwise Distance (MPD)

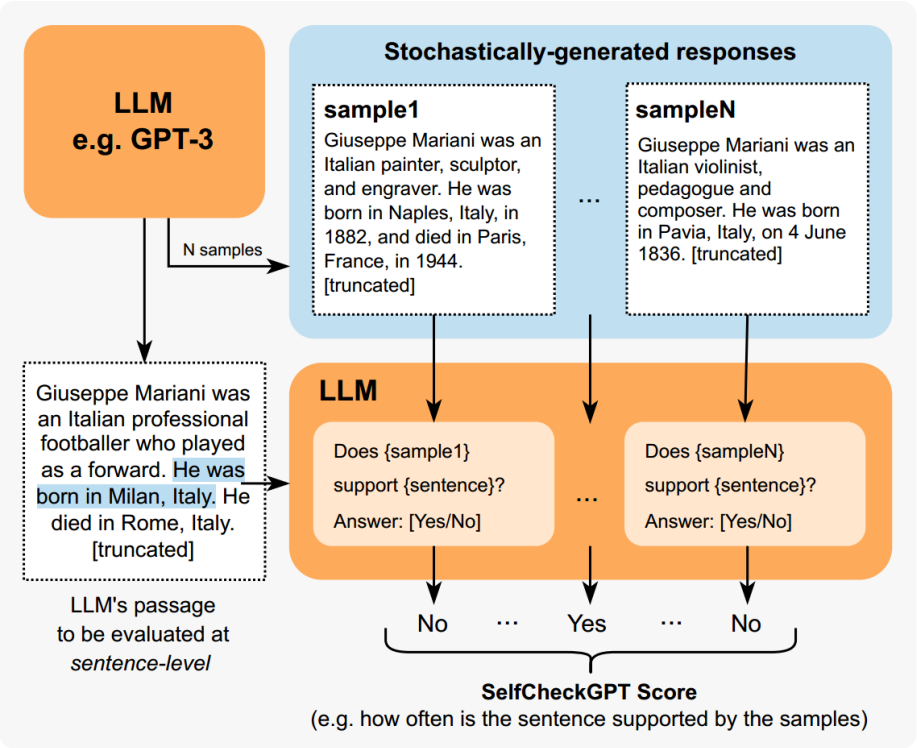


Figure: SelfCheckGPT with Prompt. Each LLM-generated sentence is compared against stochastically generated responses with no external database.

Mean Pairwise Distance (MPD)

The output from prompting when comparing the i-th sentence against sample S^n is converted to score x_i^n through the mapping {Yes: 0.0, No: 1.0, N/A: 0.5}. The final inconsistency score is then calculated as:

$$\mathcal{S}_{\text{Prompt}}(i) = \frac{1}{N} \sum_{n=1}^{N} x_i^n$$

Semantic Entropy (SE)

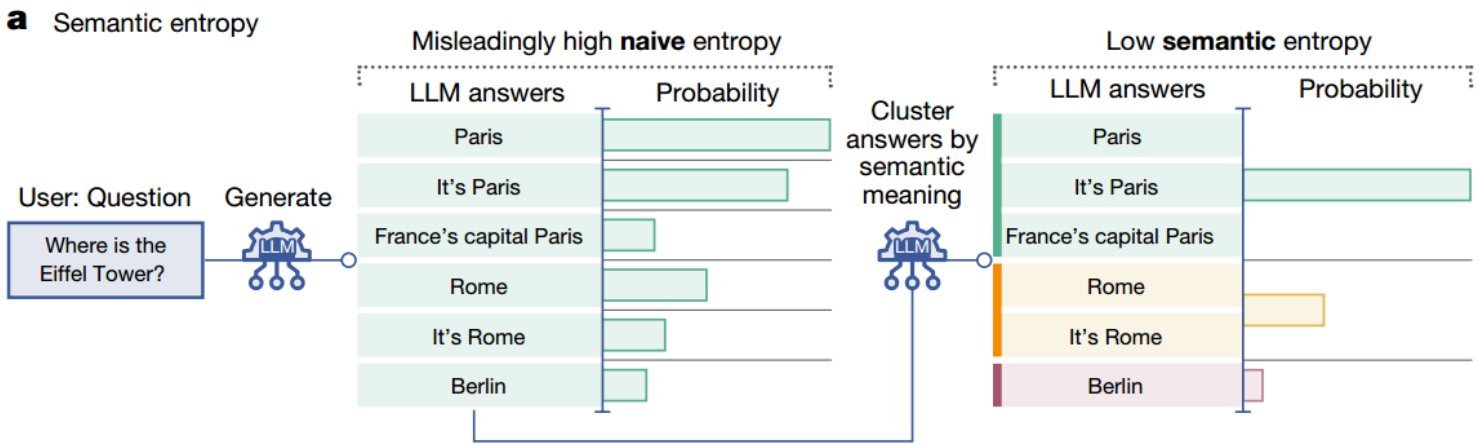


Figure: Overview of semantic entropy. Sometimes different answers mean the same things. Our semantic entropy clusters answers which share meanings before computing the entropy. A low semantic entropy shows that the LLM is confident about the meaning.

Semantic Entropy (SE)

Answer $\mathbf{s}$	Likelihood $p(\mathbf{s} \mid x)$	Semantic likelihood $\sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x)$
Paris	0.5	0.9
It's Paris	0.4	
London	0.1	0.1
Entropy	0.94	0.33

For the space of semantic equivalence classes C the sentences in the set $c \in C$ all share a meaning.

$$p(c \mid x) = \sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x) = \sum_{\mathbf{s} \in c} \prod_{i} p(s_i \mid s_{<i}, x).$$

$$SE(x) = -\sum_{c} p(c \mid x) \log p(c \mid x) = -\sum_{c} \left(\left(\sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x) \right) \log \left[\sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x) \right] \right).$$

Sum of Eigenvalues of the Graph Laplacian (EigV)

Whether two responses share the same meaning is not black-and-white. A more nuanced and "continuous" way to measure the number of meanings is preferable.

Example: What city was Zeus the patron god of?

"Olympia"

"Zeus was the patron god of Olympia, Greece"

"Corinth"

"Olympia" and "Greece" are neither exactly the same nor completely different.

Sum of Eigenvalues of the Graph Laplacian (EigV)

• **Spectral Clustering** Fixing an input x, treat each generated response as one node and define the symmetric weighted adjacency matrix as $W = (w_{j1,j2})_{j1,j2=1,\ldots,m}$ where $w_{j1,j2} = (a_{j1,j2} + a_{j2,j1})/2$.

	Olympia	Olympia, Greece	Corinth
Olympia	1	0.9	0.2
Olympia, Greece	0.9	1	0.1
Corinth	0.2	0.1	1

The symmetric normalized graph Laplacian is then given by

$$L := I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

$$D_{j_1,j_2} = \begin{cases} \sum_{j' \in [m]} w_{j_1,j'} & (j_1 = j_2) \\ 0 & (j_1 \neq j_2) \end{cases}$$

Sum of Eigenvalues of the Graph Laplacian (EigV)

A continuous version of U_{NumSet} could be defined with $\lambda_1 < ... < \lambda_m$, the eigenvalues of L:

$$U_{\texttt{EigV}} = \sum_{k=1}^{m} \max(0, 1 - \lambda_k).$$

Eccentricity (Ecc)

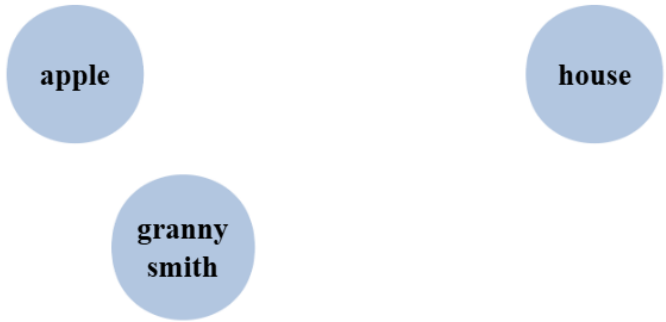
Denote $u_1, ..., u_k \in R^m$ as the smallest k eigenvectors of L, then an informative embedding of s_j is simply $v_j = [u_{1,j}, ..., u_{k,j}]$. We could use the average distance from center as the uncertainty measure, Formally, the "eccentricity" estimates are:

$$U_{\mathbf{Ecc}}(x) = \|[\mathbf{v}_1^{'\top}, \ldots, \mathbf{v}_m^{'\top}]\|_2$$

where $v'_j = v_j - \frac{1}{m}\sum_{j'=1}^{m} v'_j$ represents the offset from the average embedding.

Kernel Language Entropy (KLE)

SE captures semantic relations between the generated texts only through equivalence relations. This does not capture a distance metric in the semantic space. For instance, it separates "apple" as equally strongly from "house" as it will "apple" from "granny smith".



Kernel Language Entropy (KLE)

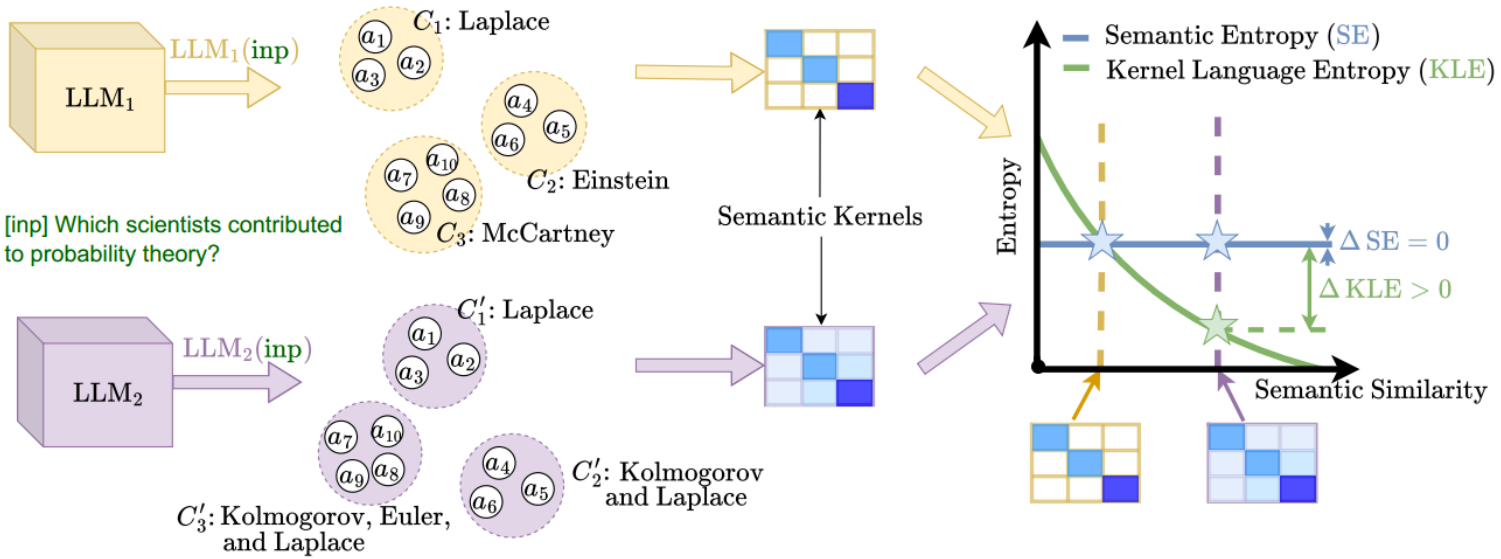


Figure: Illustration of Kernel Language Entropy (KLE). Uncertainty should be lower for LLM_2 because semantic "similarity" between the generations is much higher; i.e., the model is fairly confident that "Kolmogorov" and "Laplace" are good answers.

Kernel Language Entropy (KLE)

A generic description of the steps required to compute KLE.

Algorithm 1 Kernel Language Entropy

Require: LLM, Input $x \in \mathcal{T}^L$, Number of samples n, Boolean kle-c indicating variant, Semantic kernels K_i

Initialize a *multiset* of answers $\mathcal{O} \leftarrow \emptyset$
for $k \leftarrow 1$ to n **do** ▷ Sampling n answers
    Add LLM(x) to $\mathcal{O}$
end for
if kle-c **then**
    Update $\mathcal{O} \leftarrow$ cluster$(\mathcal{O})$ ▷ as in [36]
end if
Combine $K_i(\mathcal{O}, \mathcal{O})$ in K_{sem} ▷ see Sec. 3.1
Return VNE(K_{sem}) ▷ Eq. (8)

- Title: **Verify when Uncertain: Beyond Self-Consistency in Black Box Hallucination Detection**
- Authors: *Yihao Xue*[1,2], *Kristjan Greenewald*[3], *Youssef Mroueh*[4], *Baharan Mirzasoleiman*[1]
  [1]Department of Computer Science, UCLA
  [2]Work performed while interning at MIT-IBM Watson AI Lab [3]MIT-IBM Watson AI Lab [4]IBM Research
- Conference: *submitted to ICML 2025*
- Year: 2025.2.20

- Part1. The Performance Ceiling of Self-Consistency Methods
- Part2. Cross-Model Consistency for Hallucination Detection

The Performance Ceiling of Self-Consistency Methods

How much information does P^{self} actually encode about the ground truth hallucination annotation?

$$\boldsymbol{P}_i^{\text{self}} = [\mathcal{E}(a'_{i,j}, a'_{i,k})]_{1 \le j \le m, 1 \le k \le m}$$

Existing methods can then be formalized as some function f applied to the self-entailment matrix P_i^{self} which outputs a scalar. The focus of prior work lies in designing various forms of f.
SE(P^{self}), MPD(P^{self}), EigV(P^{self}), ECC(P^{self}), KLE(P^{self}).

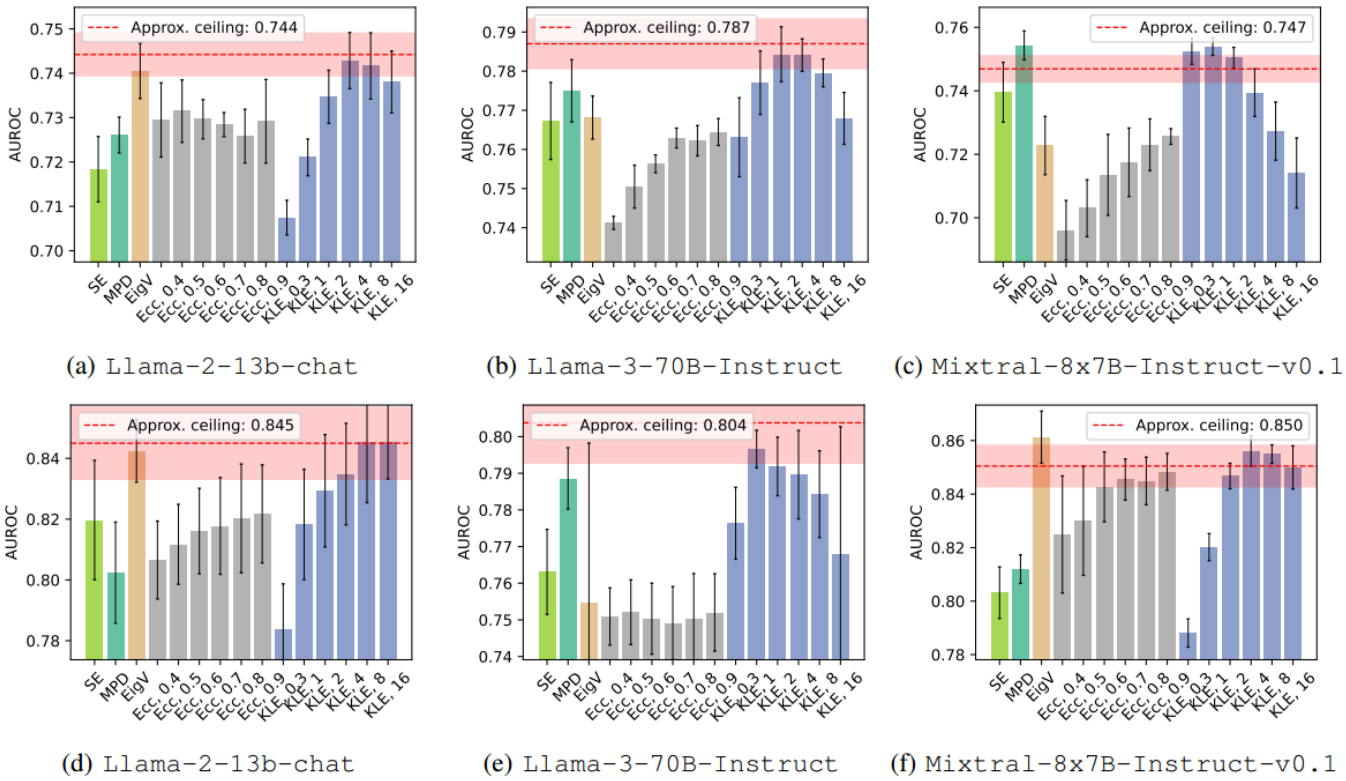
The Performance Ceiling of Self-Consistency Methods

The optimal function f that maps P^{self} to the hallucination label.

$$\hat{f} = \arg\min_{f} \mathbb{E}[l(f(\boldsymbol{P}^{\text{self}}), \hat{h})]$$

Use a two-layer Graph Convolutional Network (GCN) to represent f. The model is trained with BCE loss on sampled pairs of P^{self} and $\hat{h}$.
We then evaluate AUROC and AURAC of the resulting model as an approximation of the ceiling performance.

The Performance Ceiling of Self-Consistency Methods



(a) Llama-2-13b-chat (b) Llama-3-70B-Instruct (c) Mixtral-8x7B-Instruct-v0.1

(d) Llama-2-13b-chat (e) Llama-3-70B-Instruct (f) Mixtral-8x7B-Instruct-v0.1

Figure: Comparison between AUROC of existing methods and the approximated ceiling performance on SQuAD ((a)–(c))and TriviaQA ((d)–(f)). We observe that, across all setups, the best method performs very close to the oracle, indicating that we are approaching the performance limit.

Cross-Model Consistency for Hallucination Detection

If two models significantly disagree on their answers, at least one is likely hallucinating.

$$\boldsymbol{P}_i^{\text{cross}} = [\mathcal{E}(a'_{i,j}, b'_{i,k})]_{1 \le j \le m, 1 \le k \le m}$$

where $\left\{ b'_{i,k} \right\}_{k=1}^{m}$ are m answers sampled from the verifier model M_v.

To explore the potential gain of Cross-Model Consistency Checking, we search for a function f that takes both P^{self} and P^{cross} as input. We combine P^{self} and P^{cross} into a single matrix.

$$\begin{bmatrix} \boldsymbol{P}^{\text{self}} & \boldsymbol{P}^{\text{cross}} \\ \boldsymbol{P}^{\text{cross}} & \mathbf{0} \end{bmatrix}$$

We then apply a GCN to this combined matrix and train the model to fit the ground truth labels $\hat{h}$.

Cross-Model Consistency for Hallucination Detection

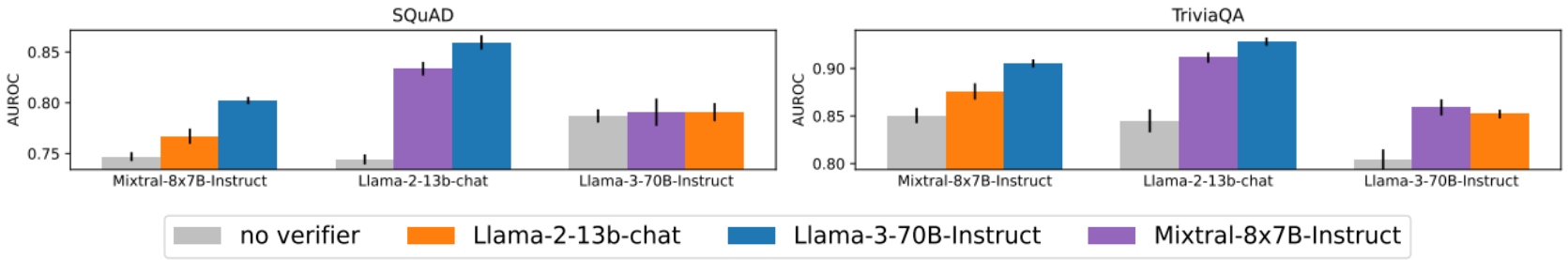


Figure: Comparison between approximated ceiling performances using only P^{self} (gray) and those using both P^{self} and P^{cross}. The x-axis shows the target model, and the colors indicate the verifier model, as shown in the legend. We observe a clear improvement when a verifier model is used.

Cross-Model Consistency for Hallucination Detection

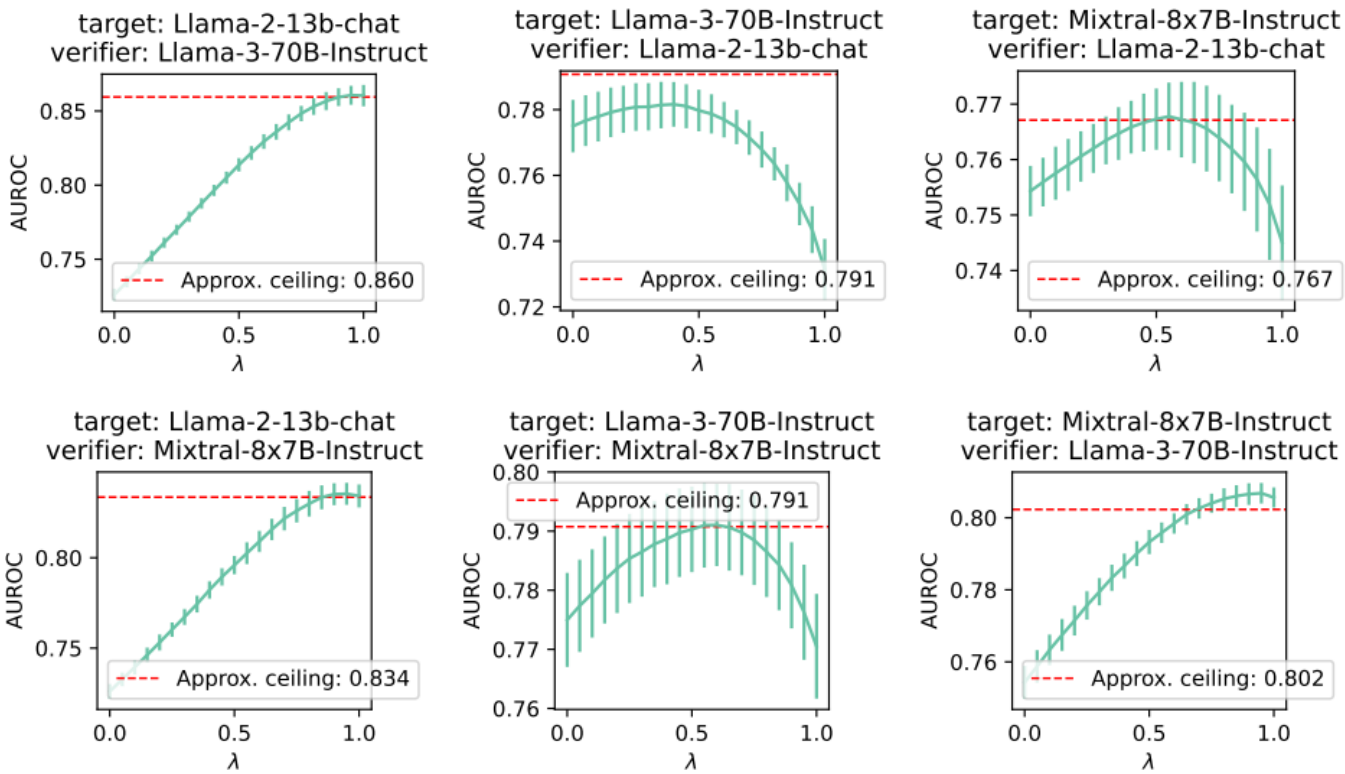


Figure: A simple weighted average of self-consistency and cross-consistency-based metrics, $(1-\lambda)MPD(P^{self}) + \lambda MPD(P^{cross})$, can achieve performance close to that of the oracle method.

Cross-Model Consistency for Hallucination Detection

- **Budget-Aware HD with A Verifier Model**

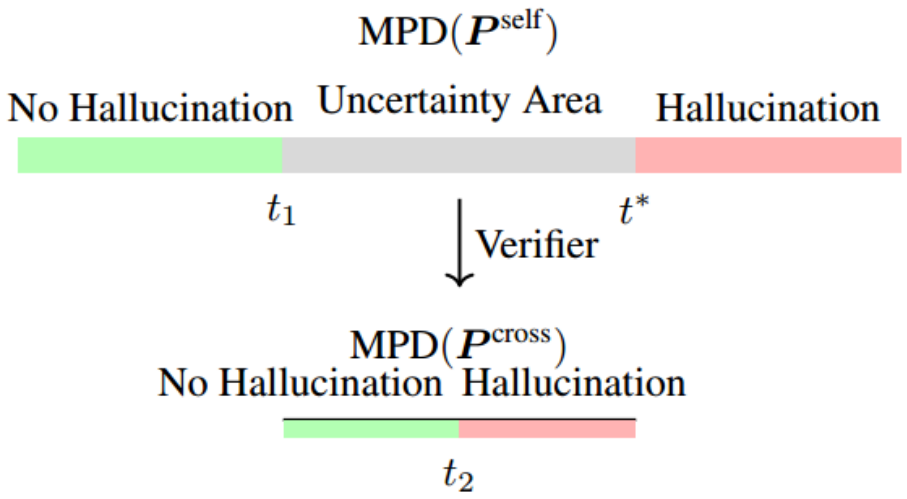


Figure: Two Stage Hallucination Detection. First, the self-consistency matrix P self is formed and the test statistic is computed. This is thresholded with two thresholds, where medium values (gray region) advance to the second stage for disambiguation. The P cross cross-consistency matrix and test statistic are then computed for these ambiguous samples for final classification.

Cross-Model Consistency for Hallucination Detection

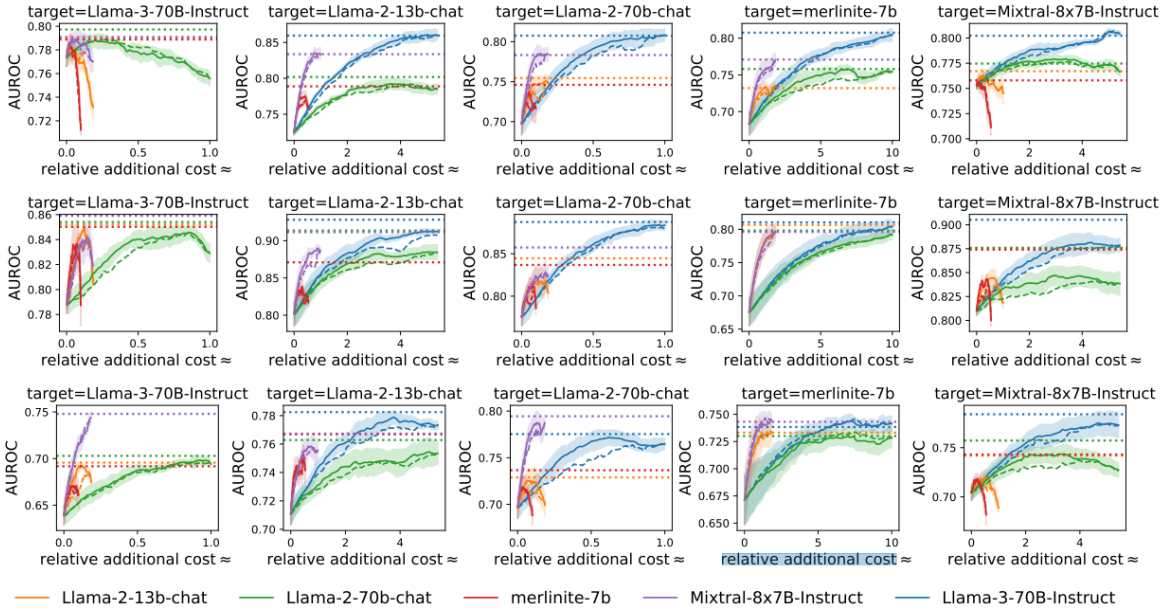


Figure: We plot AUROC against the relative additional cost for SQuAD (top), TriviaQA (middle), and Natural Questions (bottom). Solid curves represent results where the validation set consists of independent samples for the same questions, while dashed curves correspond to validation sets consisting of answers for different questions. The dotted horizontal line indicates the approximate ceiling performance using GNN. The curves are mostly convex demonstrating that our approach can achieve high performance with very low cost.

Conclusion

- **Presented** multiple self-consistency methods for hallucination detection.
- **Analyzed** the ceiling performances of self-consistency in black-box models.
- **Demonstrated** that incorporating additional models can elevate these performance ceilings.

Thanks for your attention !
Q & A