The Machine Learning Column

BY

Pablo Barceló

Pontificia Universidad Católica de Chile
Avda. Vicuña Mackenna 4860
Santiago, Chile
pbarcelo@uc.cl

AND

David Saulpic

Institut de Recherche en Informatique Fondamentale (IRIF)
CNRS & Université Paris Cité
8 Place Aurélie Nemour
75013 Paris, France
david.saulpic@irif.fr

Explaining Hallucinations from a TCS perspective: Interview with Santosh Vempala

Abstract

After sketching the landscape of how TCS can or does contribute to ML in the last Bulletin, we propose here to dive into one of the most recent attempts to shed a TCS light on some ML phenomenon – namely, using statistics to explain hallucination in language generation. We conducted for this an interview with Santosh Vempala, who pioneered this direction with Adam Tauman Kalai. Santosh tells us here the story behind the maths: the motivation for the paper, how they proceed with modeling and what are the implications of the result.

1 Introduction

For this edition of the column, we had the chance to interview Santosh Vempala about two recent papers on hallucinations in large language models (LLMs): the STOC 2024 paper *Calibrated Language Models Must Hallucinate* (with Adam Tauman Kalai) [4], and its follow-up *Why Language Models Hallucinate* (with Kalai, Ofir Nachum, and Edwin Zhang) [3].

These papers caught our attention because they are a rare attempt to bring theoretical computer science (TCS) tools to bear on a socio-technical phenomenon that has quickly become central in practice. Doing so requires more than a standard "theorem–proof" contribution: one must decide what to formalize, what to abstract away, and which simplifying assumptions remain faithful to the real object of study. We therefore wanted to discuss the modeling process itself—the choices made, the alternatives left on the table, and how these choices shape both the interpretation of the results and the kinds of follow-up questions that become tractable. We briefly introduce Santosh and summarize the main messages of the two papers before turning to our conversation.

Santosh Vempala is the Frederick P. Storey II Chair of Computing and a Distinguished Professor in the School of Computer Science at the Georgia Institute of Technology. His research sits at the intersection of algorithms, randomness, high-dimensional geometry, and the foundations of data science, with influential

work on sampling and optimization for high-dimensional distributions. He received the 2024 Delbert Ray Fulkerson Prize (jointly with Ben Cousins) for work on algorithms for computing volume and Gaussian volume, and he is a Fellow of the ACM and of the American Mathematical Society. Beyond research, he has played an active community and institution-building role at Georgia Tech, including leadership in interdisciplinary theory programs (notably the Algorithms, Combinatorics, and Optimization Ph.D. program) and the creation of the *Computing for Good* (C4G) initiative, which connects computing ideas to societal challenges through project-based work.

2 Calibrated Language Models Hallucinate

The first of the two papers, "Calibrated language models must hallucinate", traces the origin of hallucinations to a purely statistical property called *calibration*. Doing so requires introducing a number of definitions.

- First, what is a language model? From a TCS perspective, this is simply a probability distribution over strings. Indeed, a "next-token predictor" assigns, for each position, a probability distribution to the next token conditioned on the previous ones. This is mathematically equivalent to specifying a single probability distribution over all finite strings of tokens.

- Second, what is a hallucination? The authors adopt a deliberately simple setting. There is a set of "facts" F contained in a universe of possible strings Y. The hallucination rate is the total probability mass that the model assigns to strings that are not facts, that is, to elements of $Y \setminus F$.

- In addition, "the world is assumed to contain randomness": there is an underlying distribution p over the set of facts. The training data $O \subseteq F$ is generated by sampling from this distribution.

Hence, the model is trained by observing only a subset of all existing facts. A concrete example is the set of birthdays of TCS researchers. Some birthdays are frequently mentioned, others rarely, and many may never appear in public data at all. The distribution p is highly non-uniform, and any model can only be trained on a partial view of these facts.

Now, calibration is the following property. A distribution $\hat{p}$ produced by the model is calibrated with respect to the true distribution p if, for every probability value $y \in [0, 1]$, all elements that the model predicts with probability y indeed appear, on average, with probability y under p; formally,

$$\mathbb{E}[\, p(x) \mid \hat{p}(x) = y \,] = y.$$

Intuitively, if a weather model predicts rain with probability 30% on a collection of days, then it should indeed rain on about 30% of those days. In the interview below, Santosh Vempala provides a different intuition on calibration and why it is desirable.

The second paper relaxes this notion. Instead of requiring calibration everywhere, it only requires calibration on the set of *frequent* elements. Formally, letting

$$\mathcal{A} := \{x : \hat{p}(x) \geq 1/|Y|\}$$

be the set of elements predicted with above-uniform probability, the requirement is simply that

$$|p(\mathcal{A}) - \hat{p}(\mathcal{A})| \tag{1}$$

is small.

The main technical result of the first paper is a lower bound on the hallucination rate of a language model whose calibration error is small. As Santosh explains below, this assumption is natural: the standard pretraining objective, which minimizes *log-loss*[1], forces the model to be approximately calibrated. Indeed, the paper shows that any model with small log loss necessarily has small calibration error. Thus, assuming low miscalibration accurately reflects the behavior of pretrained base models.

Two further assumptions complete the picture. First, conditioned on the observed training data O, all unobserved facts in $F \setminus O$ are assumed to be equally likely to be true. Formally, for all $y, y' \in F \setminus O$,

$$\Pr[y \in F \mid O] = \Pr[y' \in F \mid O].^2$$

Second, the universe of possible statements is assumed to be much larger than the set of true facts: there exists a (large) s such that $|F| \leq e^s |Y \setminus F|$. In the birthday example, for each researcher there is a single correct date and 364 incorrect ones, yielding $s = \ln(364)$.

Under these assumptions, the main conclusion can be stated informally as follows: if not all facts are observed during training and the model is calibrated, then the hallucination rate must be at least the total probability mass of the unobserved facts. The intuition is simple. Calibration forces the model to assign approximately the correct total probability mass to the observed facts O. The remaining probability mass must therefore be assigned to unobserved statements. But among these, the model has no statistical way to distinguish true facts from false ones.

[1]The log-loss is $\mathbb{E}_{x \sim p_{train}} - \log \hat{p}(x)$, where p_{train} is the training distribution and $\hat{p}$ is the one produced by the model. It has been observed, e.g. in [1], that neural networks trained by minimizing the log-loss are calibrated – although for a different notion of calibration than the one used here.

[2]This equality can be relaxed to hold up to a multiplicative factor r, which then appears in the final bound. For simplicity we present only the symmetric case.

Since incorrect statements vastly outnumber correct ones and are equally likely conditioned on O, most of this remaining mass will inevitably fall on hallucinations. Thus the hallucination rate is close to $1 - p(O)$.

The first paper makes this quantitative by relating the unobserved mass to the number of facts that appear exactly once in the training data, via the so-called Good–Turing (missing-mass) estimator. The second paper takes a different route. Instead of estimating the missing mass, it reduces a standard binary classification problem to language generation. In this reduction, the model is asked to generate statements and is then evaluated by a simple classifier that decides whether a generated statement is valid or invalid. The key observation is that any hypothesis with large error on this *Is-It-Valid* (IIV) classification problem must necessarily assign significant probability mass to invalid statements when used as a generator. In this way, lower bounds for supervised classification error translate directly into lower bounds on hallucination rates, independently of any assumptions about missing mass.

3 The Interview

We present here a transcript of an interview we had with Santosh Vempala over zoom. We slightly edited it only to cite the papers mentioned and explain a few technical terms in footnote. The summary above also presents the general context, and explains the contribution we are discussing below with Santosh.

DS: Our initial questions are sort of an introduction to your papers and the way you thought about it. How did you start working on this project? And in your opinion, what's the contribution of the two papers?

Santosh Vempala: I have known my brilliant friend and co-author Adam Kalai for a long time. He was a few years junior to me in graduate school at Carnegie Mellon, and also he was a postdoc with me at MIT for two years. We've been discussing various aspects of theory over the years, but a couple of years ago we started talking more about language models—and in particular, why they behave the way they do.

We had the same advisor, Avrim Blum, who's a learning theorist. While that is not my primary focus, I certainly think about problems coming from high-dimensional learning and things like this. Adam is a leading researcher in learning theory. In fact, he was so worried about AI safety—this was already two years ago—that he moved from Microsoft to OpenAI, despite having offers from Princeton and CMU.

For me, the main question at the time was to explain some of the crazy things LLMs were doing consistently, which I still don't fully understand today, like

producing correct grammar. Even though you're only trained on next-token prediction, why are you getting this long-range structure—or "learning," in quotes—and so on?

At the same time, there were mistakes people would make fun of—like arithmetic mistakes—but there were also other mistakes happening, and bad advice on various topics. That seemed like an easier question to define. You can just say there's a labeling and you're outputting things with the wrong label: valid/invalid, true/false, whatever you want to call it depending on context.

So that's how we started. It took a while. We definitely began with just "facts," because that special case was fully well-defined: every document is a single statement and either it's a fact or a hallucination. It's two labels. You are only trained on facts—will you still produce hallucinations? After some iterations, we realized that as long as you're calibrated, the answer is yes. That's how it started. It also took a while to write the paper, only after we found this connection to the Turing estimate, we thought, okay, this is something nice and worth writing about.

DS: And for which community is this paper meant? Is it for TCS, for a general audience, or for statistics ?

Santosh Vempala: Yeah, so the first paper: the lower bound is just one single statistical lower bound, basically. It has nothing to do with the model or the architecture. It's really just a statistical statement, combining two things: calibration (which is classical, from statistics and game theory) and an argument about error rates.

We were motivated by practical reasons, and the theory came out not too messy. And of course there's always been this question in the background: what can theory do for ML? How can we analyze these things? It's persisting. We figured, okay, let's see if STOC will think it's sufficiently interesting and so on. In this case, they were fine.

PB: Do you see this as some kind of no-free-lunch theorem?

Santosh Vempala: The surprising thing is that *a priori* you might not have thought that being calibrated and being truthful (or not hallucinating) are at odds. Usually with trade-offs like no-free-lunch, there are two things that intuitively are in tension and then you prove a conservation-type statement: you gain one, you lose the other. But here, if you are getting more accurate, you'd think you should hallucinate less. That's the naive high-level surprise. Once you look into the actual definitions, you see it's not that simple.

DS: You said that being calibrated means being more accurate. Can you explain this—and explain why calibration is good for predictions?

Santosh Vempala: Calibration is very general. Let me give you a view that you may not see in the literature. Take any probability distribution over some set of

objects (finite, for now). Each element has some probability, summing to one. Call that the "ground truth" distribution.
Now, for another distribution to be calibrated with respect to this one, it can be any *coarsening*. By that I mean: take the original distribution, partition the elements into groups (some singleton groups, some large groups—whatever), and within each group, assign to every element the average of the true probabilities in that group. That coarsened distribution is certainly calibrated. And the other direction is also true: any calibrated distribution must be a coarsening.
The most trivial calibrated distribution is: take all elements and replace everyone's probability by the overall average. You're "accurate" in a weak sense: you get the overall average right. You can strengthen this by requiring the average to be correct on different parts (e.g., things you predict with probability at least $1/2$ versus at most $1/2$). The strongest version is the "bucket" version: for any value the model outputs, say 0.2, look at all elements for which it outputs 0.2; the true average probability over that bucket should also be 0.2. Like weather: if you predict 10% rain on all days in a bucket, it should rain on 10% of those days. You can apply this to anything where there's a distribution—including documents as outcomes.

DS: I had a question about the process of taking averages: if you choose groups in the wrong way, the average could be very far from the original distribution.

Santosh Vempala: It depends how you measure "far away." It turns out that if you look at *log loss*, coarsening cannot worsen it: the log loss with respect to the training distribution cannot get worse.
In the second paper we show something simpler: we don't need calibration in arbitrary bins. You just need two bins: low-probability and high-probability.[3] In each bin you should be calibrated in the sense that your average matches the ground-truth average.
Now, if you minimize log loss and reach any local minimum — not necessarily the global minimum! — then this calibration error is zero. Log loss is essentially what base models optimize before post-training. In fact, what we prove is that the gradient of the log loss is exactly the calibration term that appears in the theorem.

DS: Interesting, thanks! Our next question was precisely about this theorem, in the second paper: you have a calibration term related to the high-frequency predictive elements, and you also have this "is-it-valid" error term. How typical are models with high "is-it-valid" error?

Santosh Vempala: I see what you mean—the ability to distinguish valid from invalid. The nice thing about the reduction is that it's general; it's not specific to any particular failure mode. The error could be bad for statistical reasons (those are

[3]This calibration is the term defined in Equation (1)

the ones we can lower bound, like the "arbitrary facts" setting). But it could also be bad because of a bad model—like early GPT versions doing arithmetic poorly. That's not statistical hardness; there are short rules for addition and multiplication, but the model fails anyway.
Or it could be computational complexity: e.g., completing a prompt that effectively asks for factoring. That's hard not because of statistics, but because it's computationally hard.
So it's a bit hard to answer in the abstract. In practice, based on base-model hallucination rates today (on internet data or domain-specific data), it seems quite high.

PB: Suppose I fix the training distribution. Your result says: for any hypothesis, the error is at least this "is-it-valid" error minus some terms. But if my hypothesis *is* the training distribution, then the error is zero. So can you quantify: for how many hypotheses is the error large? If I pick a hypothesis randomly, would the error typically be large? This feels related to the first paper, where you say there are many hypotheses for which the error is large.

Santosh Vempala: Right—but that depends on the setting. For things that are easy to classify (say, data separable by a halfspace), many reasonable learners will achieve small classification error. Here the statement is: you give me a model, an architecture, a training procedure, and you output some hypothesis. Compare it to the true distribution. The probability of generating nonsense (hallucinations) is lower bounded by your ability to solve the corresponding supervised classification problem (valid vs. invalid), minus the calibration term. And the reduction uses only a very simple classifier derived from your model: a probability threshold. If that threshold classifier's error is high, you will hallucinate.
So it's like a worst-case reduction: for each model, there's a direct reduction—a classical reduction.

DS: So should we see this "is-it-valid" error as a simpler proxy term, easier to compute, that still gives a bound?

Santosh Vempala: Yes. And it's a thing we understand in learning theory: supervised classification. We know sources of error—VC dimension/sample complexity, and sometimes computational hardness (e.g., in restricted models). The point is not new insight on supervised classification; it's connecting that to generation/hallucination.

PB: Why couldn't it be that some optimization method leads you to a good hypothesis anyway—one with small error? How do we know we won't just end up with something bad?

Santosh Vempala: Two things. First: if what you come up with is *not calibrated*, then we say nothing.

PB: That's one of our later questions, but it's important here too. So you don't know what happens then?

Santosh Vempala: Right. For example, a model could just say "I don't know" every time. Or it could say something true all the time—"the sun rises in the east"—regardless of the question. Then it's not hallucinating (in a sense), but it's not meaningfully responding, and calibration can be very bad.
But if you have some control on calibration error, then the theorem applies. It doesn't matter how you trained, what optimization method you used, what architecture you chose: if at the end your model has bounded calibration error, then the hallucination rate is at least the validity-classification error minus the calibration term.
And one nice thing is that this condition is natural for modern base models: during pre-training, everyone minimizes log loss (equivalently *KL divergence*[4] to the training distribution). If you go to any local minimum of log loss, you get the kind of calibration we need. So base models will hallucinate roughly as much as their classification error.

DS: We also had a question about the other key assumption: the ratio of true facts versus possible hallucinations. In both papers, that ratio is assumed to be very small—there are far more possible hallucinations than true facts. But in structured domains like mathematics, formal proofs, and programming languages, maybe that's not the case. What do you expect the lower bounds to become there?

Santosh Vempala: If the numbers are equal—say half valid, half invalid—then the original main theorem says nothing because the term involving V/E becomes trivial. But two points.
First, there is a refinement where we can trade off the leading term (the validity error) with the ratio term. You can replace V/E with something like $V/(V+E)$ so you're subtracting the true fraction rather than V/E. That is best possible, since a trivial classifier can always guess "valid" and be correct with probability $V/(V+E)$. But you pay a price: the leading term gets multiplied by a factor depending on V and E. It's a partial trade-off and not fully satisfying, so we did not include it.
Second, if you look at Theorem 3 in the new paper (pure multiple choice)[5], we only need that there is at least one wrong answer. You get a lower bound even for true/false questions: one correct answer, one wrong answer. The original theorem

[4]The KL (for Kullback-Leibler) divergence between two distribution p and q is defined as $\sum_x p(x) \log \frac{p(x)}{q(x)}$. It is not a proper distance, but is commonly used to measure a loss in using Q instead of P. Models in ML are often trained to minimize the KL divergence between the ground-truth distribution and the one learned by the model.

[5]This theorems relates the hallucination rate to the optimal misclassification rate for a simple family of predictors ; in particular, it gives a *multiplicative* bound without any additive corrective terms, as opposed to the other results presented.

says nothing when $V = E$, but Theorem 3 still gives a nontrivial bound (basically a constant fraction). It's also the simplest theorem in the paper and doesn't even need calibration.
So we do not yet have the full answer for the trade-off as V approaches E and E goes to 0.

DS: One more question about modeling, perhaps the main one. The proofs aren't too complicated; so to me the contribution is the modeling—how you chose assumptions and abstracted away training details. That seems like a very nontrivial process. If I want to work on this, where do I even start? What can you say to help?

Santosh Vempala: That took a long time. In some ways it takes longer than solving a hard but well-defined problem. It's easy to go down paths where assumptions are messy—or worse, assumptions are clean and proofs are nice but irrelevant to the original motivation.
Luckily, my co-author—especially between the first and second paper—was at OpenAI. He moved there for personal reasons because he's worried about AI safety, and he always kept it grounded. I was happy to go in directions relaxing assumptions or making proofs more complicated, but he would keep bringing us back: "I don't think this actually means anything." Having that kind of empirical sanity check—from someone embedded in the field—was very helpful (he also has a knack for generating simple proofs!)

PB: Your first paper has been highly cited—I think over 200 citations, which is impressive. Many citations come from machine learning more than theory. How do you feel the ML community, or even OpenAI, received these results?

Santosh Vempala: The first paper was written when Adam was at Microsoft and we posted it before he joined OpenAI, so that was fine.
ML people are definitely interested. It's such an exciting time that they often don't have time to sit back and do theory—they need to keep going because things are moving so fast.
Between the first and second paper we kept thinking how to make it better and understand more. We had the basic results of the second paper several months before posting. Then there was this OpenAI angle, because they were publicly acknowledging hallucinations as a serious problem. Adam helped them write a blog post. Once OpenAI puts something on its front page, thousands of ML people read it or at least become aware of it.
They decided to take ownership: they said it's a real and serious problem, and even reminded us that humility is one of OpenAI's core values. Besides the science, that may have been one of the most useful outcomes: acknowledging the problem and working on it full-time.

For example, by mid-October, if you asked for the birthday of someone who doesn't have it online, GPT stopped making it up and instead asked for more information. Before mid-October, it would happily invent a date—and invent a different one if you asked again. They also reduced making up citations: now it's much harder to get a fake citation unless you try to break the system.

DS: Do you know whether they stopped hallucinating in other examples not in the paper, or just the ones everyone tried because they were in the paper?

Santosh Vempala: Historical facts is one. Another is made-up citations—they've been spending a lot of time on that. The rate is much lower now. If you behave like a normal user, it's not making up citations.

PB: Since we're entering philosophical questions: have you received criticism like, "LLMs aren't really learning in a pure distributional way; they're learning structure, they learn something about facts," etc.? How do you respond?

Santosh Vempala: There's a range of criticisms. Nothing too acerbic. Everything from "this is obvious" to "everything an LLM produces is a hallucination by definition because it's a statistical model"—a philosophical take.
Another common jump is: "they don't understand structure," "they don't reason," therefore "of course they hallucinate." I think that's an oversimplification.
There's a fundamental question: can a purely statistically trained machine reason or do symbolic manipulation without specialized training? I haven't seen a lower bound that says: if you only do statistical training, you *must* fail at reasoning on some task. If that were true, you'd want a statement like that. I haven't seen one.

PB: Probably it's quite difficult to formalize.

Santosh Vempala: Yeah—and maybe it's not true.
Let me add one thing about understanding and reasoning. Understanding is hard to define, but reasoning—logical deduction—we can at least talk about. For example, mathematical proof. Suppose you train on sufficiently many correct proofs: will the model then produce correct mathematical statements and proofs?
Just last month we put out a preprint called *The Long-Range Benefits of Next-Token Prediction* [2]. We prove a bound on model size ensuring that, with respect to any distinguisher (an algorithm that tries to tell apart outputs of the model from samples of the training distribution), the model can self-improve without knowing the distinguisher, just by minimizing log loss.
The theorem says: by minimizing log loss with a sufficiently large model (polynomial size, with an explicit bound), you reach a model whose outputs cannot be distinguished from the training distribution by distinguishers up to a certain size. In the proofs analogy: proofs generated by the model versus correct proofs are indistinguishable for proofs up to some length k; model size is polynomial in k and in the distinguisher size.

DS: If a human is the distinguisher, what is the "size"?

Santosh Vempala: Good analogy. You need to bound the size of the machine you're using. Think of it as some constant number of nodes. And you have to bound how much you can look ahead—your window size. If the window size is k and your machine size is D, then you need a language model of size about k^2D (and also scaling with $1/\varepsilon^2$, where ε is the indistinguishability parameter).

DS: So k is the size of your own memory?

Santosh Vempala: Yes—the window size: how many tokens you can use.

DS: You compared a statistical goal and a complexity goal. Your paper shows statistical limits on LLMs. Should we move away from statistics—should big companies move away from the statistical view—to reduce hallucinations and still produce novelty? Or is it incomparable?

Santosh Vempala: One thing we mention in the second paper (though we can't say much theoretically) is that post-training reduces hallucinations explicitly, and of course it makes the model less calibrated. But there is no reason in principle why you couldn't drive hallucinations close to zero. People are already doing ad hoc things; we propose some specific directions.
One reason hallucinations persist despite post-training is that evaluations often do not reward "I don't know." It's better (under many reward schemes) to guess: you might get it right and improve your score, whereas "I don't know" is treated as wrong for sure. We also discuss a notion we call "behavioral calibration."
Statistics is a good starting point, but we have to get into computation. That's why the reduction is interesting: it transfers lower bounds whether the source is statistical or computational.

DS: Perhaps your proof suggests calibration is not such a great property to desire. Also: are we calibrated as humans? When we speak and use language, are we calibrated? Is calibration the right thing to target if we want models to behave like humans with respect to facts?

Santosh Vempala: Good. It's kind of amazing that statistical training can produce output that matches the input distribution. It's not just calibrated; it tends to be accurate (low log loss).
For humans: think about young children or babies. They're more calibrated—closer to the input—they reproduce what they've heard. Tone, word choice, even bad language, matches the training distribution. As we get older, there's post-training. What we hear is not necessarily what we say: we filter, post-process, and adapt to context. So humans become less calibrated relative to their initial distribution.

There is also an empirical follow-up paper by Muqing Miao and Michael Kearns [5]. They evaluate the "arbitrary facts" setting, looking at the Turing estimate (fraction of singletons), calibration error, and the other terms, and they find the behavior is close to what the theorem suggests.
So yes: reducing hallucinations tends to reduce calibration, empirically.

DS: So is calibration a good statistical goal?

Santosh Vempala: It's a good starting point, but in many contexts it's not the right endpoint. It helps you see that if you're well calibrated in a domain where you cannot possibly store all facts, then you should worry: you may produce nonsense some of the time.

PB: For the first edition of our column: David and I wrote a short survey of connections between theoretical computer science and machine learning. For you, what should the role of theoretical computer science be for machine learning, and how should these two interact?

Santosh Vempala: That's an important question. I've helped organize several Simons programs: data science, foundations of data science, foundations of machine learning, computation in the brain. Those settings are nice because you get lots of theoreticians, but you also explicitly invite domain experts. Those interactions were quite fruitful.
A lot of what TCS can do is to formulate phenomena from practice—LLMs doing really well, questions of safe AI—more precisely. It's not going to be one question; it's nuanced. But we have a style: go to the simplest version that's still interesting, solve it, get insight, build tools slowly, and move forward. We need to do that even while the field is racing.
Safety is becoming more of a concern. Not necessarily malicious actors, but also failure modes from the machines themselves. The motivation is there for theoreticians to take a step back (which we can afford), formulate problems, develop techniques, and build tools that can be used later. We have a track record: online algorithms, optimization, high-dimensional sampling—practice eventually follows theory. So why not get started on these relevant questions now?

PB: We have time to step back—but the pace of ML is outrageous.

Santosh Vempala: Absolutely. And it's even more overwhelming for non-theoreticians, especially those not in premier labs. They mostly watch and try to use what's coming out.

PB: Yeah. Okay. It's been really helpful, interesting, and fun.

DS: Thank you very much. That was a very nice conclusion—very optimistic for TCS.

Santosh Vempala: Thank you. Thank you for taking the time.

References

[1] Jaroslaw Blasiok, Parikshit Gopalan, Lunjia Hu, Adam Tauman Kalai, and Preetum Nakkiran. Loss minimization yields multicalibration for large neural networks. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, Berkeley, CA, USA, January 30 - February 2, 2024*, volume 287 of *LIPIcs*, pages 17:1–17:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[2] Xinyuan Cao and Santosh S. Vempala. Provable long-range benefits of next-token prediction. *CoRR*, abs/2512.07818, 2025.

[3] Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate. *CoRR*, abs/2509.04664, 2025.

[4] Adam Tauman Kalai and Santosh S. Vempala. Calibrated language models must hallucinate. In *STOC*, pages 160–171, 2024.

[5] Muqing Miao and Michael Kearns. Hallucination, monofacts, and miscalibration: An empirical investigation. *CoRR*, abs/2502.08666, 2025.