When a model should say *it does not know*

What it takes for a system to decline a question it cannot answer, what decides whether it ever does, and what declining costs.

MITWA LAB, MUMBAI

IN PLAIN WORDS

Any system that answers questions can be allowed to say "I don't know". It then answers fewer questions and, if it declines the right ones, gets a larger share of those it answers right. Whether it declines at the right moments depends on more than the system. On most widely used tests a wrong guess and an honest blank both score zero, and one recent paper argues that this teaches a system tuned to those tests to guess. Declining also needs an internal sign that an answer is likely to be wrong, which the studies reviewed here find real but unreliable, most of all on unfamiliar questions. Readers matter too. In one experiment a hedged answer made people agree with it less and err less, and in another people trusted a bare answer as much as a confident one.

Two numbers for a system that may pass

A system that is permitted to decline is what machine learning calls a selective predictor, or a classifier with a reject option. El-Yaniv and Wiener (2010) give a formal account of it, worked out for the case where the labels contain no noise. The system pairs an ordinary predictor with a selection function that decides, input by input, whether to answer at all. Its coverage is the share of inputs on which it gives an answer, and its risk is the average error over those answered inputs only. Declining lowers risk at the price of coverage, and the authors name the relation between the two the risk-coverage trade-off.

The trade can be made precise for any score by which a system ranks its own answers, and the better the ranking, the more the system can answer at a given error. Geifman and El-Yaniv (2017) took image classifiers that had already been trained, used a held-out set of labelled images to choose a confidence cut-off, and proved a bound on the error among the answers that pass it. On ImageNet, counting a prediction as correct when the true label is among its top five guesses, they could promise an error no higher than 2%, with probability 99.9%, while answering 53.5% of test images with one network and 59.4% with another. The promise assumes that test images come from the same distribution as the held-out ones, an assumption that much of what follows puts under strain.

A system that declines everything never errs and is never of use, and a demand for perfect answers can push a learner all the way there. El-Yaniv and Wiener (2010) asked how much coverage can be guaranteed if the error on answered inputs must be zero with certainty, and showed that for linear classifiers there are distributions on which any learner meeting that demand is forced to decline every input.

A score that never pays for a blank

When a system will in fact decline depends heavily on how it is scored. Kalai et al. (2025) state the simplest case as a formal observation: if an answer is marked only right or wrong and "I don't know" earns nothing, the best-scoring response is never an abstention. Of ten influential benchmarks they audited, drawn from four popular leaderboards, nine use strict right-or-wrong grading with no credit for declining. The tenth, WildBench, grades on a ten-point rubric under which, the authors note, an admission of

ignorance may score below a reply rated fair that still contains a fabrication.

Their proposal is to state the price of an error in every question. The model is told to answer only if it is more than t confident, that a correct answer earns one point, that "I don't know" earns nothing and that an error costs $t/(1 - t)$ points. Answering then beats declining exactly when the chance of being right exceeds t, so at $t = 0.5$ an error costs 1 point and at $t = 0.9$ it costs 9. The audit is a count and the threshold follows by arithmetic, whereas the claim that right-or-wrong grading is what keeps released models guessing is an argument. The paper reports no experiment in which a model trained or tested under the proposed rule declined at better moments, and the authors leave open whether existing models behave as the rule intends.

A benchmark can also be built so that declining is sometimes the only correct reply. Rajpurkar et al. (2018) added 53,775 questions to the SQuAD reading-comprehension set that cannot be answered from the paragraph they accompany. Crowdworkers wrote each one to resemble an answerable question about that paragraph, with a wrong but plausible answer of the right type somewhere in the text, and a correctly declined question scored full marks. The best system the authors tested reached 66.3 F1 on the new set, against 85.8 for the same design on the old set and 89.5 for people. A system that declined every question would have scored 48.9, so the best system sat nearer to that floor than to human performance.

Our Technical Note 01, "Memory you can trust", treats one instance of this decision, a companion choosing whether to bring up something from a person's past, and derives a recall threshold and the benchmark penalty that matches it. Both are results in theory, and we do not restate them here.

A partial sense of one's own ignorance

A rule of the form "answer above confidence t" assumes the system has a confidence worth comparing with t, and whether language models have one is the most contested part of the question.

Kadavath et al. (2022) tested a family of language models of increasing size. The larger ones were well calibrated on a wide range of multiple-choice tests when the options were shown as lettered choices, meaning that the probability a model gave an answer roughly matched how often answers given that probability were right. Replacing the last option with "none of the above", which forces the model to judge each option on its own merits, reduced both accuracy and calibration substantially. The authors also trained models to predict, before answering, the probability that they would get a question right, which they call P(IK), for "I know". Trained on trivia, this predictor still told apart questions the model could and could not answer on other kinds of task, such as arithmetic and code, but its probabilities there were often poorly calibrated, mostly too low, and the authors describe the generalisation as partial.

Kamath et al. (2020) found a related weakness in an earlier kind of system, though one that ran in the opposite direction. A reading-comprehension model trained on SQuAD gave useful confidence on questions like its training data and was overconfident on questions from other datasets, so a rule built on its own probabilities declined too seldom exactly where it was most often wrong. On equal mixtures of SQuAD and one of five other question sets, that rule kept 80% accuracy while answering 48.2% of questions on average. A separate classifier trained to predict the model's errors, using examples from a further dataset, answered 56.1% at the same accuracy.

Measured by what a model says rather than by an internal probability, the results look weaker. Yin et al. (2023) built SelfAware, 1,032 questions without a definite answer (open scientific questions, questions about the future, matters of taste and others) and 2,337 answerable ones. They scored how well models singled out the unanswerable questions by detecting language of uncertainty in the replies. GPT-4 reached 75.47 F1 and two human volunteers averaged 84.93, with both figures taken from samples of 100 questions, so the gap is indicative rather than precise.

Feng et al. (2024) adapted 11 published methods for deciding when a model should abstain and tested them, with two of their own, on three models and four question-answering tasks. Averaged across methods and tasks, the decisions to answer or abstain were correct between 52% and 62% of the time, depending on the model, and their own methods beat every earlier one in 9 of the 12 model and task settings, by up to 19.3% in that measure.

The strongest negative evidence among these studies comes from Kirichenko et al. (2025), who assembled AbstentionBench from 20 datasets, more than 35,000 questions that a careful system should decline: questions with no known answer, missing context, false premises, subjective framing or outdated information. Across 20 models they conclude that abstention is unsolved and that model size made almost no difference. Models did comparatively well only on questions whose answer nobody knows. Those trained for long step-by-step reasoning did worse, with two of them abstaining 24% less on average than their non-reasoning counterparts and

often giving a definite final answer even when their own reasoning text expressed doubt, including in mathematics and science, the domains they had been trained for. Adding an instruction that described when not to answer raised abstention for the models they tested, and the authors judge that this leaves the underlying inability to reason about uncertainty in place. Their labels came from a model judge that agreed with a hand-labelled sample 88% of the time.

The price of declining too often

Declining is not free, and a system can fail by doing it too readily. Röttger et al. (2024) built XSTest from 250 requests that are safe but resemble unsafe ones in wording or topic, such as how to kill someone in a video game, with 200 unsafe requests as a contrast. Llama 2's chat model, run with the instructions it originally shipped with, fully refused 38% of the safe requests and partly refused another 21.6%. Dropping those instructions reduced the refusals without removing them, and adding a safety instruction to another model made it start refusing safe requests. These refusals concern safety rather than ignorance, though the limiting case is the same as in selective prediction, a model that refuses every request and so never causes harm and never helps.

For questions that should be declined for lack of knowledge, the measured balance runs the other way. Kirichenko et al. (2025) report that when most of their models did abstain they were nearly always right to, with precision close to 1 on most datasets, so over-abstention was rare and the main failure was answering when they should have held back. The two studies concern different kinds of request and do not contradict each other. Together they mean that a count of how often a system declines says little until it is split into the declines that were warranted and those that were not.

What the person does with a hedge

A decline or a hedge is a message to someone, and what it achieves depends on how it is read. Kim et al. (2024) ran a pre-registered experiment with 404 US-based participants. Each answered eight yes-or-no medical questions, with or without a fictional AI search system whose answers were right on only half of them. When the system opened half of its answers with first-person uncertainty ("I'm not sure, but..."), participants agreed with it less, were less confident in it and answered more accurately, and an exploratory analysis traced the gain to reduced, though not eliminated, reliance on its wrong answers. Uncertainty phrased impersonally ("It's not clear, but...") had similar effects that were weaker and not statistically significant. Having the system at all still left people less accurate than a group that never saw it. When it was wrong, participants who saw its unhedged answers were right 33.0% of the time, against 64.7% for those who answered alone. The authors list the limits themselves, among them simple yes-or-no answers, a single session and a system whose expressions of uncertainty bore little relation to whether it was right.

Zhou et al. (2024) found that people read the absence of a hedge as confidence. In online experiments with 25 US-based participants each, people decided whether to rely on an AI's answer to hard questions about world capitals from the opening words of the reply alone, without seeing the answer. They relied on confidently worded replies nearly 90% of the time and on replies with no marker of confidence about as often, while hedged replies sent roughly 90% of them to look the answer up. In a further experiment, five confidently worded errors among the first 20 questions left participants' scores lower even in later rounds, when the markers were accurate again. The models themselves seldom hedged when asked to signal their certainty on multiple-choice questions, with six of nine deployed models using significantly more expressions of certainty than of doubt, and on average 47% of their confident answers were wrong. Looking for a cause, the authors found that an openly released reward model scored hedged replies far below plain ones (average −1.86 against 4.03), and that in the human preference datasets it was trained on, hedges appeared slightly more often in the rejected replies than in the chosen ones (5.02% against 4.47%).

The two studies agree that people act on a hedge when one is given, and both rest on short tasks whose right answers the experimenters controlled. Read together with the earlier sections, they suggest that whether declining helps depends on the score a system was built for, on a sense of its own ignorance that is real and partial, and on a reader who takes silence about uncertainty to mean certainty.

What is not yet known

- Whether adopting an explicit penalty for errors in widely used benchmarks would change how often released models fabricate or decline. Kalai et al. (2025) argue that it would and report no test of it.
- Why reasoning-trained models abstain less than their counterparts. Kirichenko et al. (2025) document the drop, and find that a larger

reasoning budget generally deepens it, without isolating which part of the training is responsible.

- How to obtain a confidence that stays usable on questions unlike any seen before. The P(IK) predictor of Kadavath et al. (2022) was poorly calibrated on tasks it had not been trained on, and the calibrator of Kamath et al. (2020) did best when it had seen examples from outside the training distribution.
- How warranted and unwarranted declines trade against each other on one set of requests that mixes questions beyond a model's knowledge with safe and unsafe ones. The studies reviewed here measure one side or the other.
- Whether the effects of hedging seen in single sessions of yes-or-no questions hold over months of conversation, in other languages and cultures, and when the person cannot check the answer afterwards.

References

El-Yaniv, R., and Wiener, Y. (2010). On the foundations of noise-free selective classification. Journal of Machine Learning Research, 11, 1605–1641. www.jmlr.org/papers/v11/el-yaniv10a.html

Feng, S., Shi, W., Wang, Y., Ding, W., Balachandran, V., and Tsvetkov, Y. (2024). Don't hallucinate, abstain: identifying LLM knowledge gaps via multi-LLM collaboration. ACL 2024. Read as arXiv:2402.00367. arxiv.org/abs/2402.00367

Geifman, Y., and El-Yaniv, R. (2017). Selective classification for deep neural networks. arXiv:1705.08500. arxiv.org/abs/1705.08500

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., et al. (2022). Language models (mostly) know what they know. arXiv:2207.05221. arxiv.org/abs/2207.05221

Kalai, A. T., Nachum, O., Vempala, S. S., and Zhang, E. (2025). Why language models hallucinate. arXiv:2509.04664. arxiv.org/abs/2509.04664

Kamath, A., Jia, R., and Liang, P. (2020). Selective question answering under domain shift. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 5684–5696. aclanthology.org/2020.acl-main.503/

Kim, S. S. Y., Liao, Q. V., Vorvoreanu, M., Ballard, S., and Vaughan, J. W. (2024). "I'm not sure, but…": examining the impact of large language models' uncertainty expression on user reliance and trust. In Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT 2024). doi.org/10.1145/3630106.3658941. Read as arXiv:2405.00623. arxiv.org/abs/2405.00623

Kirichenko, P., Ibrahim, M., Chaudhuri, K., and Bell, S. J. (2025). AbstentionBench: reasoning LLMs fail on unanswerable questions. arXiv:2506.09038. arxiv.org/abs/2506.09038

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: unanswerable questions for SQuAD. ACL 2018 (where the dataset is called SQuADRUN). Read as arXiv:1806.03822. arxiv.org/abs/1806.03822

Röttger, P., Kirk, H. R., Vidgen, B., Attanasio, G., Bianchi, F., and Hovy, D. (2024). XSTest: a test suite for identifying exaggerated safety behaviours in large language models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 5377–5400. aclanthology.org/2024.naacl-long.301/

Yin, Z., Sun, Q., Guo, Q., Wu, J., Qiu, X., and Huang, X. (2023). Do large language models know what they don't know? Findings of ACL 2023. Read as arXiv:2305.18153. arxiv.org/abs/2305.18153

Zhou, K., Hwang, J. D., Ren, X., and Sap, M. (2024). Relying on the unreliable: the impact of language models' reluctance to express uncertainty. ACL 2024. Read as arXiv:2401.06730. arxiv.org/abs/2401.06730