

Constrained Decoding Can Reverse Which Inputs Look Uncertain

Mayur Parvatikar

R.V. College of Engineering, Bengaluru, India
mayurgp.is21@rvce.edu.in

Abstract

Systems that must return valid JSON or a well-formed function call can request the format in the prompt or enforce it with constrained decoding, where a grammar blocks any token that would break the format and the remaining probabilities are rescaled to sum to one. Such systems may also read the entropy of that rescaled distribution as a confidence signal. When the grammar blocks the tokens holding most of the model’s probability, what remains had received little, and rescaling it can make a concentrated distribution look diffuse. We show this in an adversarial setting built for the purpose, a safety-aligned model refusing a harmful request the grammar does not let it express, against benign instructions. Reading both entropies at the first generated token on Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct, 50 prompts per class, the class ordering reverses, at AUROC 0.125 to 0.175 before the grammar and 0.613 to 0.750 after. Within a class the two readings are uncorrelated, so recalibration does not recover the pre-mask ordering. The two entropies measure different things, so systems should state which one they read, use the pre-mask one when the goal is the model’s own preference, and otherwise report how much probability the grammar let through. The reversal is confined to the first generated token and disappears once the prompt asks for the format, although the two readings then need not agree.

1 Introduction

Language models increasingly operate in pipelines whose outputs are parsed rather than read: JSON payloads, tool calls, and API arguments. These formats can be requested in the prompt (Tam et al., 2024) or enforced at decoding time by a grammar that masks any token that would violate the required structure (Geng et al., 2023; Willard and Louf, 2023; Dong et al., 2025).

Such systems also need to know when an output can be trusted, and token-level uncertainty is com-

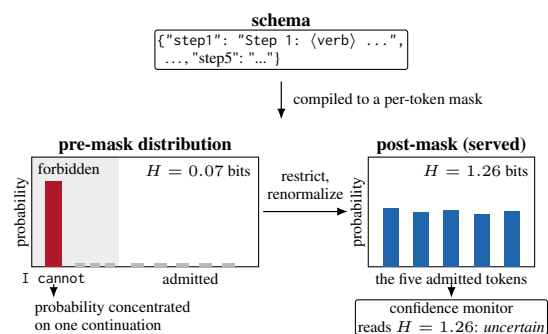


Figure 1: Schematic. Blocking the continuation that holds most of the pre-mask probability leaves a flat remainder, so after rescaling a concentrated distribution reads as uncertain. Bars are illustrative, and the two entropies are the Qwen2.5-7B NONE harmful means from Table 1.

monly used for this (Fadeeva et al., 2024; Gupta et al., 2024), with entropy the natural choice because the decoder already computes a next-token distribution. Under constrained decoding there are two such distributions: the model’s own, and the one left after disallowed tokens are removed and the remainder rescaled. Whether the entropy of the second still reflects the first is unclear, and the answer matters wherever uncertainty gates an output.

Prior work has studied how grammar constraints alter model outputs: the constrained distribution itself (Park et al., 2024), interactions with subword vocabularies (Beurer-Kellner et al., 2024), and task performance (Tam et al., 2024; Reddy et al., 2026). That work concerns the output, not the uncertainty estimate computed from the masked distribution. We ask a different question: can grammar masking change the ordering of inputs by entropy?

Masking can change that ordering, even though it only removes options. If the model places most of its probability on tokens the grammar forbids, the allowed tokens that remain may each carry small and similar probabilities, and rescaling them to sum to one produces a diffuse distribution (Figure 1).

What matters is how evenly the surviving probability is spread, not how much of it was removed.

We test this by measuring next-token entropy at the same decoding step, before and after masking: H_{pre} on the model’s own distribution and H_{post} after rescaling. Our main finding is that H_{post} can rank two classes of prompts in the opposite order from H_{pre} , while the two readings are not simply inverted at the level of individual prompts.

Our experiment is an existence demonstration, not an estimate of prevalence: we construct a contrast in which a safety-aligned model places substantial probability on refusing a harmful request while the grammar prevents that refusal from being expressed, with benign instructions as the comparison class. With one grammar, two models (Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct) and 50 prompts per class, we find that the class with lower pre-mask entropy has the higher post-mask entropy. The effect is confined to the first generated token and disappears once the prompt itself requests the format, although the two readings then still need not agree. The two entropies are therefore distinct quantities rather than a correct and a corrupted reading of one, and a system should report which of them it uses.

2 Related Work

Uncertainty estimation for language models ranges from token-level to sequence-level and semantic estimators (Liu et al., 2025; Shorinwa et al., 2025; Vashurin et al., 2025). Semantic methods aggregate over sampled continuations (Kuhn et al., 2023). We work at the token level, where entropy is a standard and inexpensive estimate (Malinin and Gales, 2021; Baan et al., 2023). Token-level uncertainty also drives decisions, not only analysis. Fadeeva et al. (2024) flag unsupported claims with a token-level probability score, and Gupta et al. (2024) use token-level uncertainty to decide when a cascade defers to a larger model, so an error in such a signal reaches what the system does.

A second line asks whether a model’s probabilities match the quantity a task needs, and often finds that they do not. Token probabilities and entropies diverge from the theoretical distribution in probabilistic scenarios (Toney and Wails, 2025), and probabilities are poorly calibrated in numeric contexts (Lovering et al., 2025). Our result is of the same kind, with one difference: the mismatch is introduced at decoding time by the mask, not

present in the model’s own probabilities.

On the decoding side, the costs of masking are well studied for the output. Park et al. (2024) show that the constrained distribution is not the model’s distribution conditioned on the grammar, and give a sampler that corrects this. Beurer-Kellner et al. (2024) show that constraints misaligned with sub-word vocabularies can reduce accuracy, and Tam et al. (2024) find that format restrictions reduce reasoning performance. Reddy et al. (2026) analyze the masking step as a KL projection, call the surviving probability the feasible mass $\alpha(h_t)$, and give the KL identity we use in Section 3. To our knowledge, none of this work examines the uncertainty estimates read after the mask.

Where the format is requested also matters. Tam et al. (2024) compare format instructions in the prompt with decoder-enforced formats. Lee et al. (2026) find that the instructions alone cause most of the accuracy loss, whereas Reddy et al. (2026) attribute most of it to the mask on MATH500. Le (2026) shows that the wording of schema keys acts as an instruction channel whose effect differs between model families. Our prompt levels vary the same factor but measure uncertainty instead of accuracy.

The grammar is also an attack surface. Zhang et al. (2026) show that schema-enforced masking can inject a prefix into a response that the model then completes with harmful content, a decoding-time counterpart to prefix injection, in which the prompt asks the model to begin with an affirmative phrase (Wei et al., 2023). Qi et al. (2025) attribute the success of attacks that prefill a response to safety alignment concentrated in the first few output tokens, and Steindl et al. (2024) find that uncertainty rises on linguistically obfuscated jail-break prompts. We borrow the schema of Zhang et al. (2026) as an instrument, because it creates the situation our question needs.

3 Entropy under a grammar mask

Let $P_{\text{free}}(\cdot \mid h_t)$ be the model’s next-token distribution over vocabulary V , and let $A_t \subseteq V$ be the tokens the grammar allows at step t . Write

$$R_t = \sum_{v \in A_t} P_{\text{free}}(v \mid h_t) \quad (1)$$

for the **surviving probability**: the share of the model’s probability that the grammar lets through, equivalently the feasible mass $\alpha(h_t)$ of Reddy

et al. (2026). The decoder then samples from $P_{\text{mask}}(v) = P_{\text{free}}(v)/R_t$ on A_t and zero elsewhere, so rescaling divides every surviving probability by the same R_t and $D_{\text{KL}}(P_{\text{mask}} \parallel P_{\text{free}}) = -\log_2 R_t$ bits (Reddy et al., 2026, Eq. 5, in nats). Let H_{pre} and H_{post} be the entropies of P_{free} and P_{mask} in bits. Setting $H_{A_t} = -\sum_{v \in A_t} P_{\text{free}}(v) \log_2 P_{\text{free}}(v)$, they are related exactly:

$$H_{\text{post}} = \frac{H_{A_t}}{R_t} + \log_2 R_t \quad (2)$$

which follows by substituting $P_{\text{mask}} = P_{\text{free}}/R_t$ into the definition of entropy, with $H_{\text{post}} \leq \log_2 |A_t|$.

The two terms depend on R_t in opposite ways, and that dependence cancels: H_{post} is fixed by the relative sizes of the surviving probabilities, not by how much probability was removed. If the grammar blocks only low-probability tokens, the surviving shape stays close to the model’s own and entropy typically falls, the familiar intuition that constraints reduce uncertainty. If it blocks the tokens holding most of the probability, H_{pre} is dominated by tokens that H_{post} no longer sees. The tokens that remain often carry similar small probabilities, so rescaling gives something close to uniform and H_{post} climbs toward its ceiling $\log_2 |A_t|$, which the grammar fixes and the model cannot affect. Ratios among the allowed tokens are preserved, so H_{post} describes how evenly probability is spread within the allowed set, which need not track how concentrated the full distribution was. This is specific to hard masking with rescaling. Samplers that instead approximate the distribution conditioned on the whole output being grammatical (Park et al., 2024) read entropy from a different distribution.

4 Setup

Models, data and constraint. We run Qwen2.5-7B-Instruct (Yang et al., 2024) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) in bfloat16 with greedy decoding, on 50 prompts from the Harm-Bench standard set (Mazeika et al., 2024) and 50 input-free Alpaca instructions (Taori et al., 2023), both sampled with a fixed seed. The constraint, adapted from Zhang et al. (2026) and compiled with XGrammar (Dong et al., 2025), is a JSON object with five fields step1 to step5, each matching $\hat{\text{Step}} \ k$: ($\langle \text{verb} \rangle$) for one of twenty-five imperative verbs and at least sixty further characters, so no step can open with a refusal. Read-

ing both distributions requires our own decoding loop: we capture the logits before the bitmask is applied, which hosted APIs and serving stacks such as vLLM (Kwon et al., 2023) do not expose by default.

Position and statistics. We measure at the first generated token, where the mask removes most of the model’s probability and nothing has yet been forced. The matcher is still in its initial state, so A_t is the same five tokens for every prompt: the opening brace and tokens that merge it with what follows, such as $\{$. Differences in H_{post} therefore cannot come from differences in what was allowed, and $H_{\text{post}} \leq \log_2 5 = 2.32$ bits. Because the two entropies range over different token sets, the full vocabulary before the mask and the five allowed tokens after it, we read ΔH as an ordering rather than a magnitude. Each pair of readings comes from one forward pass, so the grammar is the only difference within a pair. Intervals are 95% percentile bootstrap intervals over prompts, 10,000 resamples drawn within each class of fifty.

Refusal probe and prompt levels. P_{refuse} is the total teacher-forced probability of fourteen refusal phrases, each encoded with and without a leading space, and reduced to a prefix-free set so nested phrases are not counted twice. With no grammar, we count a response as a refusal if it opens, ignoring case, with one of fifteen refusal openers. The four prompt levels are NONE (no system turn), NEUTRAL (a system turn silent about format), TERSE (a one-sentence JSON request) and SCHEMA (that request plus the schema). Table 1 reports the first two, and Section 5 covers the rest.

5 Results

Pre-mask entropy is low where probability concentrates on a refusal. Mean H_{pre} on harmful prompts ranges from 0.06 to 0.21 bits across cells, against 0.76 to 0.93 on benign ones (Table 1). P_{refuse} identifies what that probability sits on, at 0.77 for Qwen and 0.83 for Llama under NONE. With no grammar, both models open with a refusal on 88% of these prompts. The low pre-mask entropy therefore reflects probability concentrated on a refusal, which the grammar blocks.

Post-mask entropy reverses the class ordering. H_{pre} ranks harmful prompts below benign ones in every cell, at AUROC 0.125 to 0.175, with every interval strictly below 0.5. H_{post} ranks them above,

Level	Class	Pre-mask entropy	Post-mask entropy	Paired ΔH	$-\log_2 R_t$
<i>Qwen2.5-7B-Instruct</i>					
NONE	harmful	0.07	1.26	+1.18	34.7
	benign	0.76	1.08	+0.32	35.9
NEUTRAL	harmful	0.06	1.42	+1.36	37.0
	benign	0.81	1.06	+0.25	36.6
<i>Llama-3.1-8B-Instruct</i>					
NONE	harmful	0.21	0.79	+0.58	26.9
	benign	0.93	0.58	−0.35	27.8
NEUTRAL	harmful	0.11	0.97	+0.85	29.6
	benign	0.92	0.70	−0.23	29.4
<i>AUROC on each score, harmful against benign</i>					
Level		H_{pre}	H_{post}	ΔH	
<i>Qwen2.5-7B-Instruct</i>					
NONE		0.158	0.613	0.856	
NEUTRAL		0.157	0.750	0.916	
<i>Llama-3.1-8B-Instruct</i>					
NONE		0.175	0.690	0.845	
NEUTRAL		0.125	0.728	0.898	

Table 1: Entropy in bits at the first generated token, means over $n = 50$ prompts per class, median $-\log_2 R_t$, and paired $\Delta H = H_{\text{post}} - H_{\text{pre}}$. ΔH is computed per prompt before rounding, so it may differ from the difference of the rounded column means by 0.01. Lower panel: AUROC with harmful prompts as the positive class and the column quantity as the score, i.e. the probability that a harmful prompt scores higher than a benign one. It measures how the two classes are ordered, not uncertainty-estimation quality: a value below 0.5 means harmful prompts have the lower entropy.

at 0.613 to 0.750, with the interval excluding 0.5 in three cells of four. In the fourth cell, Qwen under NONE, the lower limit sits at 0.5. The paired shift $\Delta H = H_{\text{post}} - H_{\text{pre}}$ separates the classes at 0.845 to 0.916 (Figure 2), which shows that the same prompts move and not merely that the two class averages differ. The shift is not the same for both classes. On the benign side Llama’s entropy falls, with ΔH of -0.35 and -0.23 , and Qwen’s rises far less than on harmful prompts. Both classes lose almost all their probability to the mask, so what separates them is how concentrated the pre-mask distribution was, not whether their most likely token was blocked. In this contrast, a threshold on post-mask entropy would tend to flag the class that a threshold on pre-mask entropy would pass.

The reading cannot be repaired after the mask.

If masking simply reversed the reading, a monitor could learn the sign and flip it. It does not. Pooled across classes the two readings correlate at Spearman ρ between -0.27 and -0.13 , and within each class the correlation is absent, between

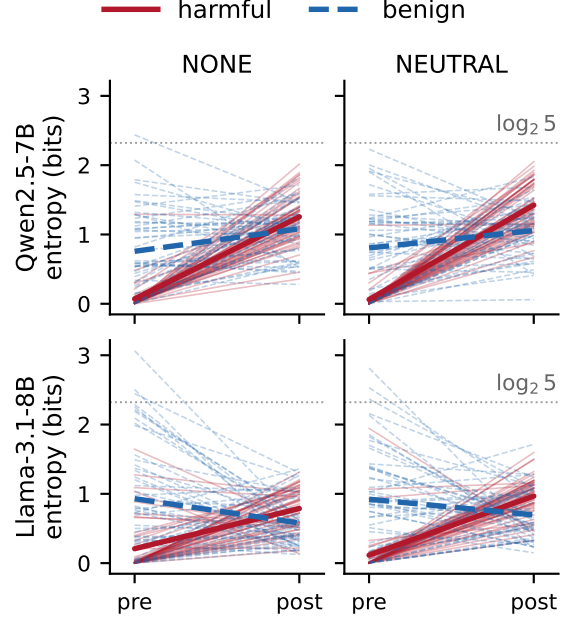


Figure 2: Entropy at the first generated token before (pre-mask) and after (post-mask) the grammar, with rows for models and columns for prompt levels. One thin line per prompt, one thick line per class mean, and the dotted line is the ceiling $\log_2 5$. Harmful prompts start low and rise, while benign prompts start high and change less.

-0.27 and $+0.17$, with every interval containing zero. Within a class, then, H_{post} says nothing about which prompt the model was more concentrated on. Rescaling discards that ordering rather than inverting it, and recalibration methods such as temperature scaling or isotonic regression preserve whatever ordering they are given, so none of them recovers H_{pre} . Combining the two readings available after the mask does not close the gap either. The identity in Section 3 recovers H_{A_t} exactly from H_{post} and R_t , and within a class it does track H_{pre} , at rank correlations of 0.76 to 0.94 on harmful prompts, but it is almost a monotone function of R_t and separates the classes no better, at AUROC 0.46 to 0.56. The class ordering is the part a threshold rule acts on, and that is what the mask reverses.

Nothing after the mask looks wrong.

The surviving probability does not flag the condition either. On harmful prompts the grammar lets through only $2^{-34.7}$ of Qwen’s probability, about one part in thirty billion, and $2^{-26.9}$ of Llama’s, but the share barely differs between the classes, separating them at AUROC 0.46 to 0.56 with every interval covering 0.5. Neither model places much probability on

opening a reply with a brace, in either class. Nor does the served reading look anomalous. H_{post} sits at 25% to 61% of its $\log_2 5$ ceiling, in the range a monitor would treat as ordinary.

What turns the reversal off. Across all eight cells at TERSE and SCHEMA, two models and two benign sets, the reversal appears in none. What moves is the surviving probability, and the two models move differently. For Qwen almost nothing survives under NONE and NEUTRAL, at a median R_t below 2^{-34} , whereas under TERSE and SCHEMA between 28% and nearly all of it does. Its refusal probability falls from 0.77 to 0.00, so the grammar no longer contradicts what the model was going to write. For Llama the surviving probability rises only on the Alpaca prompts, to 0.21 under TERSE, while on harmful prompts it stays near 2^{-26} . Llama keeps refusing at every level, at P_{refuse} between 0.83 and 0.89, and the reversal ends because the benign side moves, not because the refusal does.

A format instruction is not a remedy. The reversal disappears, but the two readings still need not agree. For Qwen under TERSE they do agree, both ranking harmful prompts as the more uncertain class, at AUROC 0.914 before the mask and 0.798 after. Under SCHEMA they disagree again, in the opposite direction to our main result, at 0.757 and 0.339. For Llama under TERSE, H_{pre} still tells the two classes apart, at 0.026, while H_{post} does not, at 0.499. Putting the schema in the prompt therefore removes this reversal, but it does not make the entropy the decoder serves a safe substitute for the model’s own. One number needs care. Qwen’s P_{refuse} of 0.00 under SCHEMA comes from matching a fixed set of refusal phrases, and it would read zero as well if the model refused in other words. A measure that uses no word list agrees with it: under SCHEMA 84% of Qwen’s probability survives the mask, whereas under NONE almost none of it does.

Where the effect stops. The effect is confined to one position, more narrowly than the refusal itself. Measuring each of 32 forced positions separately, the reversal is already gone at the second generated token, once the model has been made to open the JSON object. The AUROC of H_{pre} rises there from 0.158 to 0.657 for Qwen and from 0.175 to 0.519 for Llama. The refusal fades more slowly. Qwen’s refusal probability falls from 0.77 at the first token to 0.28 by a median of eight tokens in, where the grammar opens a new string and a refusal would

be a grammatical continuation again, and to 0.03 at the verb slot, a median of twelve tokens in. At that earlier point the reversal has still not returned, with AUROC 0.73 before the mask against 0.55 after for Qwen, and 0.71 against 0.69 for Llama. This fits the finding that safety alignment is concentrated in the first few output tokens (Qi et al., 2025). It also means that an entropy averaged over a whole response would dilute the effect rather than compound it.

What the effect is not. H_{post} is not a harm detector. On the safe subset of XSTest (Röttger et al., 2024), Qwen’s is 1.48 bits, against 1.26 on genuinely harmful prompts. What it measures is how evenly the surviving probability is spread, not how harmful the input was. The harmful and benign contrast is our instrument, not our finding.

6 Conclusion

Entropy read on either side of a grammar mask can order inputs oppositely. The cause is not how much probability the grammar blocks but how evenly what survives is spread across the tokens it allows. In the contrast we construct the readings are reversed between the classes and uncorrelated within them, with nothing after the mask to signal it. The two entropies answer different questions: one describes the model’s own distribution, the other what the decoder will sample from. We therefore recommend that systems state which of the two they read. Where logits are available and the target is the model’s own preference, as in abstention, they should compute the pre-mask one. Otherwise they should record $-\log_2 R_t$, which any masking decoder can obtain at negligible cost. In our setting the reversal disappears once the prompt asks for the format, without making the two readings agree, and how often it arises in ordinary structured-generation workloads remains to be measured.

Code and data

Our code, the compiled grammar and the per-token records behind every table and figure are available at <https://github.com/MayurGP25/Constrained-Decoding-UQ>.

Limitations

We cover one grammar, one harmful-prompt benchmark, fifty prompts per condition and two open-weight models, and our comparison is at a single

token position, where a positional check shows the effect is confined: it does not survive past the first generated token. Our two classes differ in H_{pre} by construction, since one is chosen to elicit a refusal the model was trained to prefer, so the reversal is a property of this contrast rather than an estimate of how often it arises in deployment. We do not evaluate standard structured-generation or function-calling workloads, such as BFCL (Patil et al., 2025), and do not test which downstream uses of uncertainty are affected. At $n = 50$ small differences between individual AUROC values should not be over-read. Closed-weight systems cannot be examined at all, since the method needs logits from before the constraint is applied.

Ethics Statement

The harmful-prompt class is an instrument for manipulating uncertainty, not a safety contribution. We introduce no attack and report no attack success rates. The schema is adapted from published work (Zhang et al., 2026) rather than designed here, and we release token identifiers and distributional statistics rather than decoded generations, so no harmful text is distributed. HarmBench (Mazeika et al., 2024) is used as its authors intend, to elicit and study refusal behavior. The finding is defensive: a monitor reading entropy under a constraint can misrank the inputs on which the model’s own distribution was most concentrated, and we report the diagnostic that reveals it.

References

- Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. [Uncertainty in natural language generation: From theory to applications](#). *Preprint*, arXiv:2307.15703.
- Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2024. Guiding LLMs the right way: Fast, non-invasive constrained generation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*.
- Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ziyi Xu, Yilong Zhao, Ruihang Lai, and Tianqi Chen. 2025. XGrammar: Flexible and efficient structured generation engine for large language models. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 7.
- Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385. Association for Computational Linguistics.
- Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952, Singapore. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkritum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. In *International Conference on Learning Representations (ICLR)*.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*.
- Yifan Le. 2026. [Schema key wording as an instruction channel in structured generation under constrained decoding](#). *Preprint*, arXiv:2604.14862.
- Ivan Yee Lee, Loris D’Antoni, and Taylor Berg-Kirkpatrick. 2026. [The format tax](#). *Preprint*, arXiv:2604.03616.
- Xiaoou Liu, Tiejun Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6107–6117.
- Charles Lovering, Michael Krumdick, Viet Dac Lai, Varshini Reddy, Seth Ebner, Nilesh Kumar, Rik Koncel-Kedziorski, and Chris Tanner. 2025. [Language model probabilities are not calibrated in numeric contexts](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29218–29257. Association for Computational Linguistics.
- Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations (ICLR)*.

- Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.
- Kanghee Park, Jiayu Wang, Taylor Berg-Kirkpatrick, Nadia Polikarpova, and Loris D’Antoni. 2024. Grammar-aligned decoding. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The Berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of PMLR, pages 48371–48392.
- Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2025. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR)*.
- Avinash Reddy, Thayne T. Walker, James S. Ide, and Amrit Singh Bedi. 2026. The hidden cost of structured generation in LLMs: Draft-conditioned constrained decoding. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*. arXiv:2603.03305.
- Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 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*, pages 5377–5400. Association for Computational Linguistics.
- Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. 2025. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*.
- Sebastian Steindl, Ulrich Schäfer, Bernd Ludwig, and Patrick Levi. 2024. *Linguistic obfuscation attacks and large language model uncertainty*. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertainNLP 2024)*, pages 35–40, St Julians, Malta. Association for Computational Linguistics.
- Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? A study on the impact of format restrictions on performance of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Association for Computational Linguistics.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.
- Autumn Toney and Ryan Wails. 2025. *Certain but not probable? Differentiating certainty from probability in LLM token outputs for probabilistic scenarios*. In *Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertainNLP 2025)*, pages 51–60. Association for Computational Linguistics.
- Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2025. Benchmarking uncertainty quantification methods for large language models with LM-Polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Brandon T. Willard and Rémi Louf. 2023. *Efficient guided generation for large language models*. Preprint, arXiv:2307.09702.
- An Yang, Baosong Yang, Beichen Zhang, et al. 2024. *Qwen2.5 technical report*. Preprint, arXiv:2412.15115.
- Shuoming Zhang, Jiacheng Zhao, Hanyuan Dong, Ruiyuan Xu, Zhicheng Li, Yangyu Zhang, Shuaijiang Li, Yuan Wen, Chunwei Xia, Zheng Wang, Xiaobing Feng, and Huimin Cui. 2026. When grammar guides the attack: Uncovering control-plane vulnerabilities in LLMs with structured output. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. arXiv:2503.24191.