Scratchpad 3Q Reasoning: Surfacing Assumptions to Mitigate Hallucination and Improve Truthfulness in Large Language Models

Praneeth Vadlapati
Independent researcher
praneethv@arizona.edu
ORCID: 0009-0006-2592-2564

Abstract: Large language models (LLMs) make impressive predictions but remain prone to hallucinations—confident yet incorrect statements—when answering factual or adversarial queries. We propose the 3Q scratchpad framework: a lightweight prompting and logging method that asks the model to explicitly produce three short internal reasoning sections for each response ("What I Know", "What I Need", and "What I Am Assuming") and then to emit a concise final answer that is shown to the user while the scratchpad is saved for analysis. This approach does not change model architecture or require additional human annotations; it augments the interaction protocol to make latent reasoning explicit and auditable. We evaluate the method on TruthfulQA, a benchmark designed to elicit model falsehoods. Using the provided task-wise results for a representative model (Phi-4-mini), the 3Q scratchpad substantially reduces hallucination in categories such as logical falsehoods (from 21.43% to 47.64% non-hallucinated) and produces modest gains on other categories. We analyze why the intervention works, catalogue failure modes, discuss privacy and logging trade-offs, and propose extensions. The 3Q framework is a practical, low-cost intervention that meaningfully improves model truthfulness by forcing localized transparency into model outputs.

The source code is available at github.com/Pro-GenAI/S3Q-Reasoning.

Keywords: Large language models, LLMs, reasoning, hallucination, truthfulness, scratchpad, interpretability, prompting, TruthfulQA, Artificial Intelligence, AI

I. Introduction

Large language models (LLMs) have advanced rapidly in capability, enabling high-quality natural language generation across many tasks. Nevertheless, LLMs frequently generate statements that are incorrect, unverifiable, or inconsistent with reality—phenomena commonly termed hallucinations. Hallucinations pose a risk when models are used for knowledge-driven tasks, decision support, or information retrieval, especially when users treat generated text as authoritative. Many research efforts focus on improving model training, retrieval augmentation, or decoding strategies to mitigate hallucination. These approaches are important but often require additional compute, dataset curation, or architectural changes.

In this work we describe a complementary, lightweight approach that does not change the model weights or require specialized training data: a scratchpad-augmented prompting protocol we call "3Q" because it asks models to produce three short internal sections—(1) What I Know, (2) What I Need, and (3) What I Am Assuming—followed by a single concise final answer that is displayed to the user. The intermediate scratchpad is logged but not shown to the user. This simple change encourages the model to (a) surface its evidence and gaps, (b) expose assumptions that might underlie incorrect inferences, and (c) allow automatic checks or human reviewers to detect likely hallucinations. The method has two central aims: reduce the incidence of outright false claims in the visible final answer, and produce an auditable trail for downstream verification or error analysis.

We present the 3Q framework, illustrate its implementation as a prompting pattern and parsing routine, and evaluate its effect on TruthfulQA. The framework is deliberately minimal and practical: it augments the conversation protocol with a single system instruction and requires only a parser and a log. Our contributions are:

- A succinct, replicable prompting protocol (3Q) for producing explicit, delimited scratchpads alongside concise final answers.
- An evaluation on TruthfulQA showing substantial improvements in certain hallucination-prone categories for a representative model setup (Phi-4-mini), using task-wise metrics provided by the user.
- An analysis of mechanisms by which the scratchpad reduces hallucination, plus a discussion of limitations, privacy considerations, and practical deployment guidelines.

A. Disadvantages with current approaches

Existing techniques to reduce hallucination fall broadly into three categories: architectural/training interventions (e.g., instruction tuning, fact-conditioned generation), retrieval augmentation (e.g., RAG), and decoding-time strategies (e.g., self-consistency, chain-of-thought). Each has value but also limitations:

- Architectural and training fixes require dataset curation, compute, and complex pipelines, and may still fail on adversarial prompts or on data outside training distributions.
- Retrieval augmentation increases reliance on external knowledge stores and introduces engineering complexity (indexing, freshness, alignment between retrieved text and responses).
- Chain-of-thought (CoT), tree-of-thought (ToT) and related approaches aim to improve reasoning by letting the model produce internal chains of reasoning that can be sampled or aggregated. While these often improve answer accuracy, they can also amplify hallucination: long generated internal narratives may invent facts with high fluency and further entrench incorrect conclusions.

Crucially, many approaches improve the probability of correct answers in expectation but do not make the model's failures more visible or auditable to downstream users. The 3Q scratchpad addresses this gap by pushing the model to explicitly list knowledge, gaps, and assumptions in a disciplined format that can be parsed and stored.

B. Proposed system and its benefits

3Q is a prompt-level intervention with three core components:

1. A system instruction that requires the model to output, using exact markers, three short sections describing: WHAT_I_KNOW, WHAT_I_NEED, and WHAT_I_AM_ASSUMING, wrapped inside a <think>...</think> block.
2. Explicit final answer markers (---FINAL_ANSWER_START--- and ---FINAL_ANSWER_END---) so the model's output can be deterministically parsed and the final answer isolated for presentation.
3. A logging mechanism that records the raw model output and parsed scratchpad entries (timestamped), enabling offline audit, aggregation, and downstream automated checks.

Benefits:

- **Transparency**: surfacing knowledge and assumptions makes it easier to detect likely hallucinations programmatically (e.g., when WHAT_I_KNOW is empty or inconsistent with the final answer).
- **Minimal cost**: no retraining or extra model calls are required; the approach works at the prompting layer.
- **Auditability**: logs provide traceable artifacts for error analysis, model debugging, and human review.

The approach is not a reasoning enhancement per se: it does not force the model to be more factual by itself. Instead, it forces the model to reveal gaps and assumptions, which in turn enables interventions—automated heuristics or human review—to correct or block hallucinated outputs.

C. New use cases of the system

- Production deployments that require traceability (e.g., medical triage assistants, legal drafting aides) can use scratchpad logs to justify answers or flag answers for human review.
- Research into model calibration and hallucination detection can use large-scale scratchpad logs as supervision signals to learn heuristics for warning generation.
- Interactive tutoring systems: teachers can review students' (model) scratchpads to provide targeted corrections when assumptions are wrong.

D. Related work

The 3Q framework is inspired by several lines of work: chain-of-thought prompting (eliciting intermediate reasoning steps), tree-of-thought search (structured internal deliberation), and self-consistency (sampling multiple reasoning traces and aggregating answers). Unlike those methods, which primarily aim to improve internal deliberation or search, 3Q focuses on transparency and logging: it asks the model to declare what it knows and assumes and to list missing information. This is closer in spirit to interpretability and explanation approaches that seek to expose latent model decisions rather than directly modify them.

We intentionally avoid proposing new training objectives or dataset collection. Instead, 3Q is positioned as an operational, prompt-time technique that complements training- or retrieval-based approaches.

II. METHODS

A. Prompt design

We use a single system instruction that requires the model to produce a scratchpad delimited by <think>...</think>, containing three labeled sections: WHAT_I_KNOW, WHAT_I_NEED, and WHAT_I_AM_ASSUMING. After the scratchpad the model must include the final concise answer between ---FINAL_ANSWER_START--- and ---FINAL_ANSWER_END--- markers.

The aim is brevity and clarity: each scratchpad section should be short (a few bullet points or sentences). The final answer is intentionally concise (one paragraph) to minimize verbosity and keep the user-visible output focused.

B. Parsing and logging

The raw model output is parsed using deterministic string markers. The parser extracts each scratchpad section and the final answer (or falls back to heuristics if markers are missing). Parsed entries are stored in a timestamped JSON log for later analysis. The logging allows aggregate studies of which assumptions correlate with hallucinations.

C. Evaluation protocol

We evaluate truthfulness using TruthfulQA, a benchmark with question categories designed to elicit falsehoods and misconceptions. The primary metric is percent non-hallucinated answers (i.e., TruthfulQA score as provided by the user's experiments). We compare two configurations: the model's standard responses (baseline) and the model with 3Q scratchpad prompting (scratchpad). Analysis focuses on task-wise score changes and qualitative failure modes.

III. Results

We report the provided task-wise TruthfulQA scores for a representative model labeled Phi-4-mini. Table 1 summarizes category-level improvements observed when using the 3Q scratchpad prompting. (Numbers come from the user-provided evaluation.)

TABLE I. TRUTHFULQA TASK-WISE SCORES (PHI-4-MINI)

Task	Standard	Scratchpad	*Change*
Logical Falsehood	21.43%	47.64%	***+26.21%***
Misconceptions	21.11%	26.93%	*+5.82%*
Misinformation	11.83%	17.33%	*+5.50*

IV. Discussion

These results indicate that the 3Q scratchpad has a pronounced positive effect on tasks that involve simple logical falsehoods. The largest absolute gain is in the Logical Falsehood category, where the model's non-hallucinated rate more than doubled. Smaller improvements appear for Misconceptions and Misinformation categories. While these numbers reflect a single-model setting and should not be overgeneralized, they highlight where the 3Q intervention is most effective: situations where the model can detect internal logical inconsistencies or identify missing premises when asked to enumerate them explicitly.

A. Limitations

The method depends on the model following instructions. If the model does not adhere to the marker scheme, parsing fails and benefits erode. Transparent scratchpads can still contain fabricated facts. The model may confidently list false premises under WHAT_I_KNOW; a scratchpad revealing bad facts is helpful for diagnosis but does not automatically prevent a hallucinated final answer. Logging internal reasoning raises privacy and safety concerns—scratchpads may contain sensitive user data or reveal misinformation. Deployers must adopt consent, retention, and access controls. The 3Q framework adds verbosity to model outputs and some compute for storage. While minimal compared to training-time approaches, it increases bandwidth and storage needs for large-scale deployments.

B. Ethical considerations

Logging internal scratchpads creates powerful audit trails, but also introduces new risks. Policies must govern who can access logs, how long they are stored, and how they are used. In regulated domains, logs may aid compliance; in others, they may increase liability. The trade-offs require careful policy design and, where appropriate, user consent.

V. CONCLUSION

The 3Q scratchpad framework is a low-cost, prompt-level intervention that increases transparency and reduces certain types of hallucinations in LLM outputs. Evaluated on TruthfulQA with results provided in this study, 3Q yields substantial improvements in Logical Falsehoods and modest gains in other categories. The approach is complementary to retrieval and training-based solutions: it does not replace the need for accurate knowledge sources but provides a practical mechanism to surface the model's internal state and to catch many hallucinations before they become user-visible.

Future directions include systematically combining 3Q with retrieval augmentation and calibration methods to see if gains are additive, training lightweight classifiers on logged scratchpads to predict hallucination risk and automatically flag or block risky answers, extending the scratchpad format to include short provenance pointers (structured citations) to reduce reliance on the model's unsupported knowledge claims, and conducting large-scale user studies to measure the human factors of auditability and trust when scratchpads are available to moderators.

APPENDIX

1. Prompt templates used to process using LLMs

```
You are going to produce a short scratchpad and then a final concise answer.
Format your output EXACTLY as follows (use the markers):

<think>
WHAT_I_KNOW:
<bullet points or short sentences>

WHAT_I_NEED:
<bullet points or short sentences>

WHAT_I_AM_ASSUMING:
<bullet points or short sentences>
</think>
---FINAL_ANSWER_START---
<One concise final paragraph that answers the user. This is the only text shown to the user.>
---FINAL_ANSWER_END---

Do not include any other text outside these markers. Keep the scratchpad truthful about uncertainty.
```

Figure A1. System prompt template to enable scratchpad-3Q reasoning

REFERENCES

[1] <pending>