Assessing Response Consistency Across Query Variants in LLM

1st Yuanyuan Xin
Computer Science
Colorado School of Mines
yuanyuan_xin@mines.edu

2nd Varsha Sathiskumar
Computer Science
Colorado School of Mines
varsha_sathiskumar@mines.edu

***Abstract*—Large language models (LLMs) are increasingly becoming an integral source of knowledge. However, their reliability depends on whether they remain consistent under common user behaviors, such as rephrasing a question or requesting longer lists. We study LLM consistency using two complementary benchmarks on a curated set of security-focused prompts: paraphrase robustness, where users rephrase the same question, and enumeration consistency, where we vary the requested list length. For paraphrase robustness, we evaluate variant answers against a gold answers using NLI-based logical agreement and BLEURT semantic similarity. For enumeration consistency, we extract individual list items, embed and cluster them by semantic similarity, and compute cluster-level overlap and repetition statistics across scopes. Across paraphrase variants, NLI scores increase with model capability, indicating improved semantic consistency, while BLEURT scores remain consistently negative across models, reflecting low surface-level semantic similarity despite logical agreement. For security-related enumeration questions, LLMs reliably recall standard frameworks and tool checklists, but exhibit greater variation when listing risks. This indicates that while established security procedures remain consistent, the specific risks presented may change depending on how the prompt or scope is phrased.**

***Index Terms*—robustness, consistency, accuracy, LLM, security**

I. Introduction

Large language models (LLMs) are increasingly adopted as assistants for security-related tasks, such as explaining best practices, listing common vulnerabilities, or recommending tools and frameworks. In these scenarios, users often treat the model's output as a kind of soft "checklist" and rely on it to support configuration and risk-assessment decisions. However, such usage raises an important question: are these answers consistent when users naturally change how they ask their questions? Prior studies on LLM safety and security have argued that standard benchmarks often fail to capture significant real-world risks, particularly in cybersecurity settings where model outputs directly influence human decisions. At the same time, several works show that LLMs are surprisingly sensitive to paraphrasing: semantically equivalent prompts can lead to inconsistent or even contradictory answers, and a substantial fraction of model errors can be traced to this paraphrastic variability rather than to the underlying task itself. These findings suggest that, for security guidance, evaluation must go beyond accuracy on a fixed prompt and instead directly measure consistency under realistic user behaviors, such as rephrasing a question or requesting longer lists for verification.

In this project, we investigate LLM consistency for a curated set of security-oriented questions along two complementary axes. First, we build a paraphrase robustness benchmark: for each base security question, we generate multiple paraphrases and compare the model's answers to a gold answer using NLI-based logical agreement and BLEURT semantic similarity. This captures both outright contradictions and more subtle semantic differences across phrasings. Second, we design an enumeration consistency benchmark focused on list-style prompts (e.g., "list risks…", "list tools…", "list phases of a framework…"). Here we systematically vary the requested list length (5, 10, 20, 30, 40 items), decompose answers into individual list items, embed and cluster them by semantic similarity, and compute cluster-level overlap, novelty, and self-repetition across scopes. At a high level, our empirical results reveal two complementary aspects of LLM consistency for security-related questions. First, in the paraphrase robustness benchmark, we find that LLMs generally preserve the core meaning of security-related answers across different phrasings, with semantic consistency improving for larger models, even though surface wording varies. Second, in the enumeration consistency benchmark, framework and tools questions exhibit a large, stable core set of concepts that persists under scope widening. In contrast, risk questions are more open-ended, with lower core stability but higher conceptual diversity. As the requested list length increases, recall of the core list rises monotonically, and tool questions nearly recover their full core at the largest scope. Still, tools also show the highest level of semantic self-repetition within long lists. Taken together, these findings suggest that while LLMs can provide relatively stable checklist-style guidance, the specific tools and risks surfaced to users can vary with phrasing and scope, potentially leading to inconsistent or incomplete security advice if outputs are treated as authoritative without additional validation.
Rather than proposing an Internet-scale benchmark, we present an initial, quantitatively grounded case study of how an LLM balances stability, diversity, and redundancy when generating security checklists. This analysis can help shape future evaluation suites and guide safer use of LLMs in security-critical workflows.

II. Background

A. QA robustness and paraphrastic variability

Early work on question answering built extensive static benchmarks, such as SQuAD, in which systems are evaluated on fixed question–answer pairs with a single canonical wording. [7, 6] These datasets drove rapid progress in neural QA but essentially assume that users ask questions in a single form and a single response metric can summarize that performance. More recent studies show that this view is incomplete: QA models can be surprisingly brittle to question paraphrasing. For example, prior work has found that performance degrades noticeably when questions are rephrased and proposed training strategies to improve robustness to paraphrased questions. [3] Follow-up evaluations on transformer-based QA systems similarly report that semantically equivalent questions often receive different answers, indicating that models do not always recognize paraphrases as expressing the same intent. [1] With the rise of large language models, this issue has resurfaced in a more pronounced form. Fu et al. (2024) explicitly study paraphrase divergence in LLMs, showing that semantically equivalent prompts can elicit divergent generations, and propose paraphrasing methods that better align with a model's preferences. [2] Kurt Pehlivanoğlu et al. (2024) provide a broader comparison of paraphrasing performance on a ChatGPT-generated dataset (ParaGPT), further highlighting both the power and variability of modern LLMs as paraphrasers. [4] Srikanth et al. (2024) analyze paraphrastic variability in natural-language reasoning and argue that a substantial fraction of errors can be attributed to differences in wording rather than to the intrinsic difficulty of the underlying task. [9] Together, these works suggest that paraphrases are a natural but challenging stress test: even if a model "knows" the correct answer in principle, small changes in phrasing may lead to inconsistent or contradictory outputs. In our study, we build on this line of work by constructing paraphrased versions of security questions and using NLI-based agreement and learned similarity metrics to quantify how consistent an LLM's answers remain across paraphrases.

B. LLM safety in security settings and our focus

In parallel, a growing body of research examines LLM safety, security, and privacy as these models are integrated into real systems. [10] Survey work has argued that conventional NLP-style leaderboards under-estimate real-world risk, especially when models are placed in the loop of security workflows, for example, to suggest hardening steps, enumerate common vulnerabilities, or recommend tools and frameworks. [10, 5] In such settings, users often treat model outputs as informal security checklists, and inconsistencies across paraphrases or follow-up queries can directly affect configuration and risk-assessment decisions. Our project sits at the intersection of these strands. We focus specifically on security-oriented questions and study LLM consistency along two axes that mirror realistic user behavior. First, we evaluate paraphrase robustness for security prompts, asking whether answers to different phrasings of the same question remain logically compatible with a reference answer. Second, we investigate enumeration consistency for list-style questions (e.g., "list risks…", "list tools…", "list phases of a framework…") by varying the requested list length and analyzing the resulting item sets using embeddings and clustering. Taken together, these two axes let us move beyond static accuracy and study LLM consistency in a way that mirrors how users actually interact with models in security settings. Our analysis covers both how answers change across paraphrased versions of the same question and how security checklist-style outputs evolve, in terms of stability, diversity, and redundancy, as we vary the requested list length.

III. System Design

A. Overview

Our study has two evaluation axes that share the same overall structure. We start from a small curated set of base security questions (covering framework, tools, and risk questions), generate variants of these questions, query an LLM for answers, and then compute consistency metrics.

- To assess paraphrase robustness, we generate several paraphrases of each base question, obtain answers from the LLM, and compare each answer to a reference answer using an NLI and BLEURT model. This yields whole-answer consistency scores, including entailment rates, contradiction rates, and similarity distributions across paraphrases.
- For enumeration consistency, we take the same base questions but vary the requested list length (5, 10, 20, 30, 40 items). For each scope, we collect the model's checklist-style answer, split it into individual items, embed and cluster these items by semantic similarity, and then derive cluster-level metrics such as the size and recall of a core set of concepts, coverage and novelty between scopes, and within-scope repetition inside each list.

B. Dataset and Question Types

We build our question set by sampling from the ShareGPT52K dataset on Hugging Face, which contains about 90K real ChatGPT conversations collected via the ShareGPT API. [8] For each conversation, we take the initial user message as a candidate question. As a first filtering step, we automatically assign each question to one of eight coarse topic labels (artificial_intelligence, machine_learning, data_science_and_databases, cybersecurity, software_engineering, computer_systems_and_networks, theoretical_computer_science, not_cs). We keep only questions whose label is one of the CS-related categories and discard those tagged as not_cs. Among the remaining questions, we focus on those labeled cybersecurity. Concretely, we scan the first 10,000 questions in the dataset and, from the cybersecurity-labelled subset, manually select 30 security-related base questions for our study. These cover topics such as secure configuration, access control, vulnerabilities, and threat

TABLE I
ENUMERATION-STYLE SECURITY QUESTIONS USED IN THE SCOPE-WIDENING ANALYSIS.

ID	Type	Question (shortened)
Q1	Risk	What risks might arise when building a digital identity passport system on decentralised identifiers (DIDs) and Ethereum?
Q2	Framework	What are the practical Essential Eight recommendations for an SME to improve security without hiring a security consultant?
Q3	Tool	Provide a PowerPoint-style table of contents for privacy-preserving authentication schemes in VANETs.
Q4	Framework	What key parameters help determine whether a protocol is secure?
Q5	Tool	List leading platforms for scanning binary software components.
Q6	Tool	What types of identity servers exist, and how do they interact with a client?
Q7	Risk	What are the most important aspects of cybersecurity?
Q8	Framework	Teach the key concepts in the CISSP domain "Communications and Network Security."
Q9	Risk	What security considerations apply when outputting sensitive information to a named pipe (FIFO)?
Q10	Risk	How can I prevent my site's data from being scraped if I expose data through a REST API backed by JSON files?

TABLE II
PHARAPRASED SECURITY QUESTIONS USED IN THE SCOPE-WIDENING ANALYSIS.

Variant Category	Question
Base	How can Microsoft Entra be used to verify the identity of a user before accessing the software?
Paraphrasing	How can a software company leverage Microsoft Entra to authenticate users before allowing software access?
Added Noise	In the context of enterprise login systems, how might Microsoft Entra ID be integrated to confirm user identities before they interact with a software platform?
Scenario-Based	Suppose a SaaS platform requires identity validation; how could Microsoft Entra enforce secure sign-ins?
Negative Framing	How can identity verification be ensured using Microsoft Entra without relying on insecure manual checks?
Perspective-Based	From an identity architect's perspective, how should Microsoft Entra ID authenticate users prior to application access?

modelling, and are used in our paraphrase robustness evaluation. Within these 30 security questions, we further identify 10 enumeration-style questions that naturally ask the model to "list" items (e.g., list risks, tools, or framework phases). These 10 questions serve as the basis for our enumeration consistency analysis. For each of them, we create scope-widened variants by asking for different list lengths (5, 10, 20, 30, and 40 items). Together, the 30 paraphrased questions and the 10 enumeration questions yield 150 question variants paired with answers from the target LLM. The 10 enumeration-style security questions used in our scope-widening analysis are listed in Table 1. The sample of the variant questions used in our analysis is in Table 2.

C. Models and Tools

We evaluate modern GPT-style LLMs provided by OpenAI. For the paraphrase robustness experiments, we compare three models:

- GPT-4o-mini
- GPT-4.1
- GPT-5-mini

using the same set of security questions and paraphrased variants. For each model, we fix the decoding parameters (temperature and sampling settings) so that differences in behavior mainly reflect the model rather than random sampling noise. To score paraphrase consistency at the semantic level, we use off-the-shelf evaluation models:

- An NLI model that classifies the relation between two answers as entailment, contradiction, or neutral, and
- A learned similarity scorer in the spirit of BLEURT that outputs a real-valued semantic similarity score.

We treat these evaluation models as black-box scorers and do not fine-tune them on our data. For the enumeration consistency experiments, we focus on GPT-5-mini as the base model. We collect its checklist-style answers at different scopes (5, 10, 20, 30, 40 items) and then compute embeddings for each list item using OpenAI's text embedding model. These embeddings are used for clustering and for deriving our cluster-level metrics (core size and recall, coverage, novelty, and within-scope repetition), as described in Section III-E.

D. Paraphrase Robustness Pipeline

The first axis of our evaluation focuses on paraphrase robustness: for the same security question, how a model's response changes when the question is rephrased while preserving its intent. This analysis is based on a curated set of 30 security-focused questions, each associated with five paraphrase variants drawn from predefined variant categories(Table 2). For each question–variant pair, we query three OpenAI LLMs using fixed decoding parameters and collect the generated responses along with metadata identifying the model version, original question, gold answer, paraphrase variant, and variant category.

To evaluate consistency across paraphrases, each variant response is compared against a gold answers derived from the original question using natural language inference (NLI) and BLEURT. For NLI-based agreement, each paraphrased response h is paired with the reference response p and passed through a pretrained NLI model that outputs probabilities over three relations: Entailment, Neutral, and Contradiction(Table 3). We define the NLI agreement score as the entailment probability, where higher values reflect stronger logical consistency between the paraphrased and reference responses.

In parallel, we compute BLEURT scores for each response pair (p, h) using a pretrained BLEURT model. BLEURT pro-

TABLE III
NLI RELATION CATEGORIES

Category	Description
Entailment	Paraphrased response is logically supported by the reference.
Neutral	Partial overlap without clear logical support.
Contradiction	Paraphrased response conflicts with the reference.

duces a real-valued semantic similarity score based on contextual embeddings from a transformer encoder and is trained via regression on human judgment data. Higher BLEURT scores indicate closer semantic alignment between the paraphrased and reference responses, while lower or negative scores indicate weaker semantic similarity or divergence. Unlike surface-level n-gram metrics, BLEURT captures paraphrasing, semantic nuance, and meaning preservation even when wording differs substantially.

Together, NLI and BLEURT provide complementary views of paraphrase robustness: NLI measures logical agreement and factual consistency, while BLEURT measures fine-grained semantic similarity, enabling us to distinguish between responses that preserve meaning, those that are only loosely related, and those that introduce conflicting or inconsistent information under benign rephrasing.

E. Enumeration Consistency Pipeline

The second axis of our evaluation focuses on enumeration consistency: for the same security question, how the set of concepts in the model's answer changes as we request longer or shorter lists. This analysis is based on the subset of 10 enumeration-style security questions selected from our dataset. For each of these questions, we query GPT-5-mini with five different scopes, asking for 5, 10, 20, 30, and 40 items respectively, using fixed decoding parameters. For each scope, we split the model's answer into individual list items (e.g., bullet points or numbered entries). We lightly normalize the text (lowercasing, trimming punctuation) but do not otherwise edit the content, so that our analysis reflects the model's raw output.

We then map each list item to a sentence embedding using OpenAI's text-embedding model. All embeddings are L2-normalized, and we cluster them with a simple greedy algorithm based on cosine similarity:

- We maintain a set of cluster centers.
- For each new item, we assign it to an existing cluster if its cosine similarity to that cluster's center is above a fixed threshold (e.g., 0.7).
- Otherwise, we start a new cluster with that item as the initial center.

This produces, for each question, a set of semantic clusters that represent distinct concepts appearing across all scopes. Using these clusters, we define several metrics:

- A core set of clusters for a question: clusters that appear consistently across multiple scopes. We compute a core ratio as the proportion of all items that belong to core clusters, capturing how much of the answer mass is concentrated in a stable set of concepts.
- For each scope size, core recall: the fraction of core clusters that appear at that scope. This measures how well shorter or longer lists recover the stable core concepts.
- Between consecutive scopes (5→10, 10→20, 20→30, 30→40), coverage and novelty:
  - Coverage is the fraction of clusters from the smaller scope that reappear in the larger scope.
  - Novelty is the fraction of clusters in the larger scope that did not appear in the smaller one.

These two quantities together describe how much the model preserves previous concepts versus introducing new ones as the list grows. Finally, for each scope and question type, we compute within-scope repetition as the proportion of items in a list whose cluster has already appeared earlier in the same list. A high repetition rate indicates that, rather than adding new tools or risks, the model is simply rephrasing concepts it has already mentioned. By applying these metrics separately to the framework, tools, and risk questions, we obtain a detailed picture of how GPT-5-mini's checklist-style security answers behave under scope widening and how stable, diverse, or redundant the underlying set of concepts is across different question types.

Metric definitions: For each enumeration question, we cluster all list items from all scopes (5, 10, 20, 30, 40 items) and treat each cluster as a semantic concept. Let C_{all} denote the set of all clusters for that question. In practice, we consider a cluster to be *core* if it appears in at least three different scopes; the set of such clusters is written C_{core}.

Core ratio and number of concepts. We summarize the overall stability and openness of a question with two simple quantities:

$$\text{core\_ratio} = \frac{|C_{\text{core}}|}{|C_{\text{all}}|}, \qquad \text{num\_concepts} = |C_{\text{all}}|.$$

Core ratio is the fraction of all clusters that belong to the stable core, while num_concepts is simply the total number of distinct clusters the model produced across all scopes.

Core recall at a given scope. For a specific scope size s (e.g., $s \in \{5, 10, 20, 30, 40\}$), let C_s be the set of clusters that appear in the s-item answer. We define the *core recall* at scope s as

$$\text{recall}_s = \frac{|C_s \cap C_{\text{core}}|}{|C_{\text{core}}|}.$$

This measures how many of the core clusters are already present when the model is only asked for s items.

Coverage between scopes. To compare two scopes A and B (with A the smaller scope and B the larger one), let C_A and C_B be the corresponding cluster sets. The *coverage* from A to B is

$$\text{coverage}(A \to B) = \frac{|C_A \cap C_B|}{|C_A|}.$$

High coverage means that most concepts from the smaller list are preserved when we request a longer list.

Novelty between scopes. For the same pair of scopes A and B, we also quantify how many concepts in the larger scope are genuinely new. Let new_in_big $= C_B \setminus C_A$ be the set of clusters that only appear at scope B. The *novelty* (or new_ratio) is

$$\text{novelty}(A \to B) = \text{new\_ratio}(s_{\text{small}} \to s_{\text{big}}) = \frac{|\text{new\_in\_big}|}{|C_B|}.$$

This is the fraction of clusters in the larger list that did not appear in the smaller one.

F. Implementation Details

All experiments are implemented in Python using Jupyter/Colab notebooks. We load the processed question sets (30 security questions and the 10-item enumeration subset) from CSV files generated in the data-cleaning step and treat them as fixed inputs for all evaluations. For answer generation, we call OpenAI's Chat Completions API with GPT-4o-mini, GPT-4.1 and GPT-5-mini. Each question variant (paraphrased, and scope-widened) is asked in an independent single-turn conversation with a fixed system prompt and low temperature, so that responses are approximately cold-start and comparable across variants. All raw prompts and outputs are collected for later analysis. For paraphrase robustness, we run the same pipeline separately on GPT-4o-mini, GPT-4.1, GPT-5-mini, then compute NLI labels and BLUERT score using evaluation models. For enumeration consistency, we query only GPT-5-mini, parse its list-style answers with simple rule-based splitting, and obtain sentence embeddings via OpenAI's embedding API. Embeddings are L2-normalized and clustered using the greedy cosine-similarity procedure with a threshold of 0.7, which we selected based on small-scale manual inspection; nearby thresholds produced qualitatively similar trends. All consistency metrics (core, coverage/novelty, and within-scope repetition) are computed with standard NumPy/Pandas code on top of these clustered representations.

IV. Results

A. Paraphrase robustness

Across all paraphrase variants, natural language inference (NLI) scores show a clear and consistent improvement with increasing model capability, indicating stronger logical consistency under rephrasing. For GPT-4.1, average NLI scores range from approximately 0.60 to 0.63, with dominant labels split between Neutral and Entailment, suggesting that responses generally preserve meaning but sometimes introduce additional context or reformulations not strictly implied by the reference. GPT-4o-mini exhibits higher and more stable NLI scores, typically in the 0.62–0.67 range across all variant categories, with Entailment dominating, indicating improved logical alignment between paraphrased and reference responses. GPT-5.1 achieves the strongest paraphrase robustness, with NLI scores consistently in the 0.68–0.69 range across variants

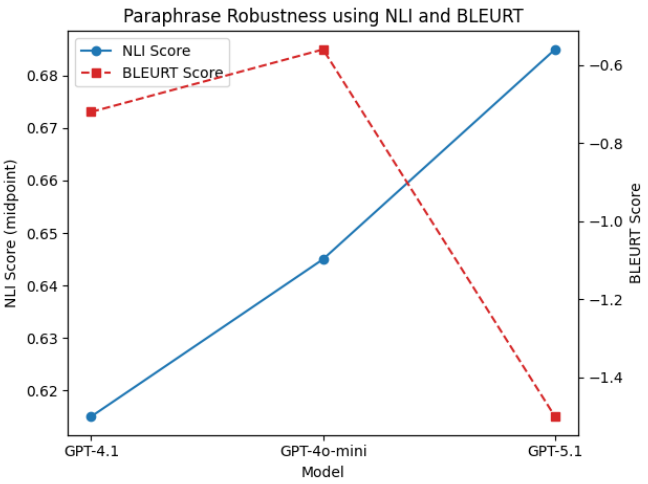

Fig. 1. Avg.NLI and BLEURT score results for each model

and predominantly Entailment label, reflecting high semantic consistency despite surface-level variation.

In contrast, BLEURT scores remain consistently negative across all models and paraphrase categories, highlighting substantial variation in phrasing and fine-grained semantic realisation. GPT-4.1 produces BLEURT scores clustered around -0.70 to -0.75, indicating weak semantic similarity beyond shared intent. GPT-4o-mini shows moderately improved BLEURT values, typically between -0.55 and -0.60, suggesting closer semantic alignment than GPT-4.1 while still exhibiting notable paraphrasing diversity. GPT-5.1 yields the most negative BLEURT scores overall, often in the -1.45 to -1.55 range, reflecting a tendency to generate more restructured, elaborated, or stylistically distinct responses even when logical agreement is preserved.

Variant 1 (Paraphrasing) generally yields stable NLI scores across all models, indicating that simple rewording preserves semantic intent well. Variant 2 (Add Noise) introduces slight degradation in NLI and BLEURT, reflecting sensitivity to extraneous or distracting information, though semantic alignment is largely maintained. Variant 3 (Scenario-based) tends to lower BLEURT and occasionally NLI, suggesting that contextual reframing alters surface similarity and partially shifts meaning. Variant 4 (Negative Phrasing) shows mixed effects: while NLI remains relatively stable, BLEURT often decreases, indicating lexical divergence despite semantic consistency. Variant 5 (Perspective-based) typically achieves the highest NLI scores, implying that viewpoint changes are well handled semantically, even though BLEURT remains low. Overall, variants affect lexical similarity more than semantic entailment, highlighting model robustness at the meaning level rather than wording level. The Tables 5, 6, and 7 present the results for each variant evaluated on individual models.

Taken together, these results demonstrate a systematic divergence between logical consistency and surface-level semantic similarity under paraphrasing. While latest version models

TABLE IV
PARAPHRASE ROBUSTNESS RESULTS USING NLI AND BLEURT

Model	NLI Score	NLI Label	BLEURT Score
GPT-4.1	0.60–0.63	Neutral / Entailment	≈ -0.72
GPT-4o-mini	0.62–0.67	Entailment	≈ -0.56
GPT-5.1	0.68–0.69	Entailment	≈ -1.50

TABLE V
PARAPHRASE ROBUSTNESS RESULTS FOR GPT–4O–MINI USING NLI AND BLEURT

Variant Category	Avg. NLI	Avg. BLEURT
Variant 1 – Paraphrasing	0.6590	-0.5882
Variant 2 – Add Noise	0.6745	-0.5970
Variant 3 – Scenario	0.6294	-0.6010
Variant 4 – Negative Phrasing	0.6221	-0.5494
Variant 5 – Perspective Based	0.5986	-0.5682

achieve higher NLI scores, indicating stronger preservation of meaning and factual alignment, they simultaneously exhibit lower BLEURT scores, suggesting increased freedom in expression and response formulation. This highlights that improved paraphrase robustness in terms of logical agreement does not necessarily correspond to higher lexical or semantic similarity, reinforcing the importance of using complementary metrics when evaluating consistency in security-related LLM outputs.

B. Enumeration consistency under scope widening

We now analyze the 10 enumeration-style security questions that ask the model to "list" items (framework phases, tools, or risks). All results in this section are for GPT-5-mini. Using the cluster-based metrics from Section 3.5—core_ratio, num_concepts, core recall, coverage, novelty, and within-scope repetition—we study how the model's checklists behave as the requested list length increases from 5 to 40 items.

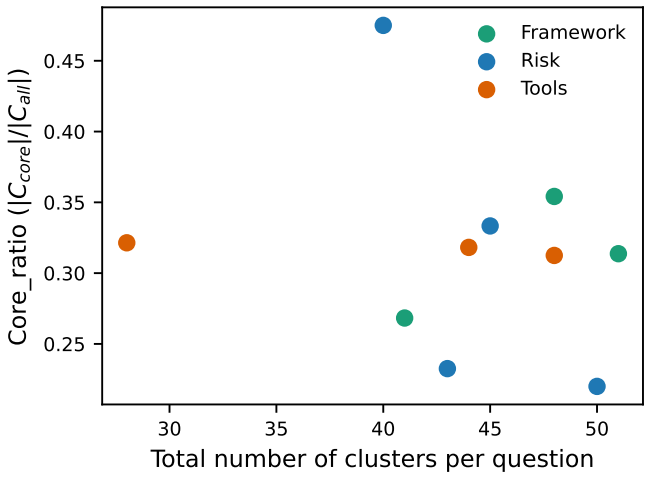

Fig. 2. Type-level stability vs. openness.

TABLE VI
PARAPHRASE ROBUSTNESS RESULTS FOR GPT–4.1 USING NLI AND BLEURT

Variant Category	Avg. NLI	Avg. BLEURT
Variant 1 – Paraphrasing	0.6219	-0.7234
Variant 2 – Add Noise	0.6196	-0.7478
Variant 3 – Scenario	0.6260	-0.7239
Variant 4 – Negative Phrasing	0.5743	-0.7066
Variant 5 – Perspective Based	0.6030	-0.7150

TABLE VII
PARAPHRASE ROBUSTNESS RESULTS FOR GPT–5.1 USING NLI AND BLEURT

Variant Category	Avg. NLI	Avg. BLEURT
Variant 1 – Paraphrasing	0.6815	-1.4857
Variant 2 – Add Noise	0.6862	-1.5224
Variant 3 – Scenario	0.6727	-1.5277
Variant 4 – Negative Phrasing	0.6836	-1.5555
Variant 5 – Perspective Based	0.6878	-1.5376

1) Type-level stability and openness: Fig. 2 plots each question in terms of its total number of clusters $|C_{\text{all}}|$ (x-axis) and its core_ratio $|C_{\text{core}}|/|C_{\text{all}}|$ (y-axis). Points are colored by question type. Figures 4 and 3 summarize these quantities by averaging over questions of the same type.

Across these plots, we see consistent but moderate trends. Framework and Tool questions tend to have a higher core_ratio than Risk questions, meaning that a larger fraction of their clusters belong to the stable core that appears in multiple scopes. Framework questions also show the largest number of distinct clusters, Tool questions the smallest, and Risk questions lie in between. Intuitively, framework-style prompts behave more like structured checklists, tool prompts focus on a relatively narrow set of tools, and risk prompts are more open-ended: they cover a similar or larger concept space, but with fewer clusters that are shared across scopes.

Given that we only have ten questions and that the type labels are somewhat coarse, these differences should be interpreted as preliminary trends rather than strong statistical claims. Nevertheless, they are stable across our metrics and will be reflected again in the analyses below.

2) Core recall as scope increases: Figure 5 shows core recall as a function of scope size for framework and tools questions. For each type, we fix the question- level core set C_{core} and, for each scope $s \in \{5, 10, 20, 30, 40\}$, measure what fraction of those core clusters appear in the s-item answer.

For both types, core recall increases monotonically as the scope widens. At the smallest scopes (5 and 10 items), only a subset of the core clusters is present. As we move to 20 and 30 items, recall steadily improves, and by 40 items, both types recover almost all of their core clusters. Tool questions reach slightly higher recall at the largest scope, but both curves are close to 1.0. This indicates that, for more structured enumeration prompts, asking for longer lists primarily helps the model fill in missing canonical items instead of replacing

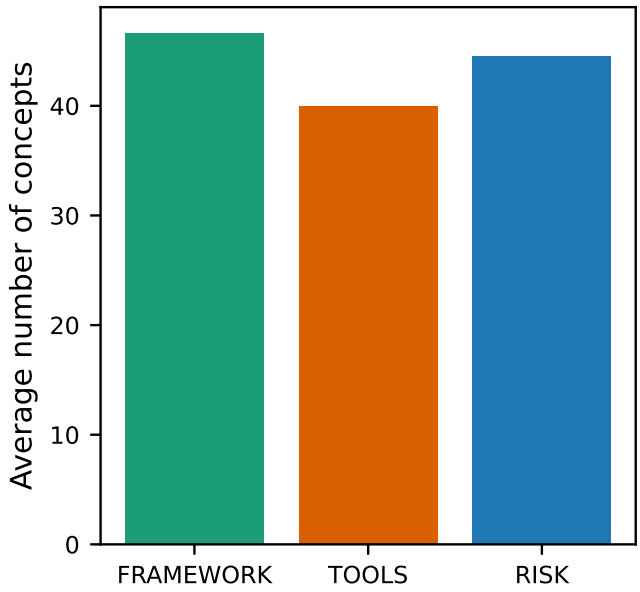

Fig. 3. Total number of distinct clusters (concepts) by question type.

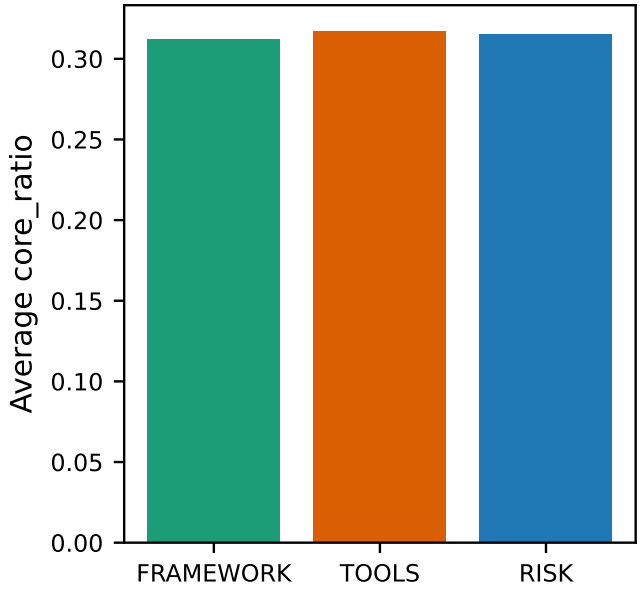

Fig. 4. Core ratio by question type.

the core with unrelated concepts.

3) Coverage and novelty for risk questions: To understand how risk lists evolve across scopes, we focus on the risk questions and compute coverage and novelty between consecutive scopes (Figure 6). Coverage from A to B is the fraction of clusters from the smaller scope that reappear in the larger scope; novelty (new_ratio) is the fraction of clusters in the larger scope that were not present in the smaller one.

We observe that coverage is consistently high: most risk clusters that appear at a smaller scope, survive when we ask

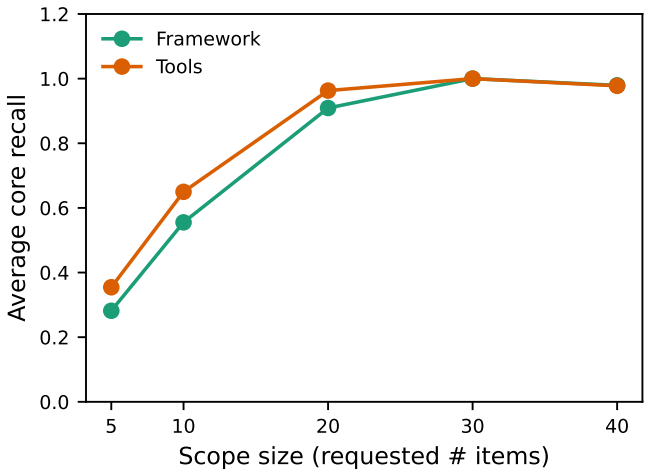

Fig. 5. Core recall across scope sizes (5, 10, 20, 30, 40).

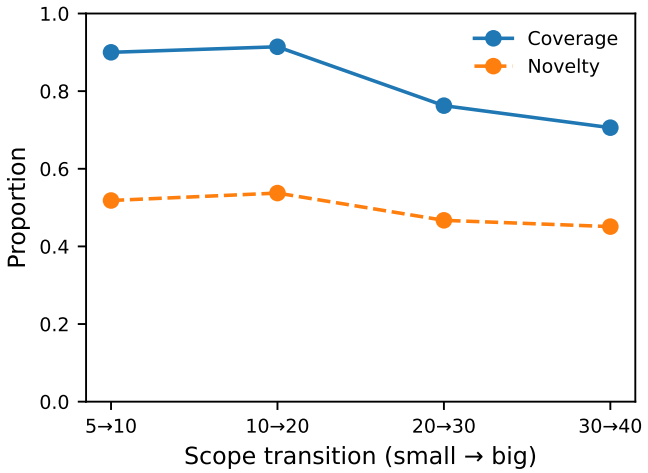

Fig. 6. Coverage and novelty between consecutive scopes.

for a longer list, although coverage gradually decreases as we move from $5 \rightarrow 10$ to $30 \rightarrow 40$. At the same time, novelty remains substantial at every transition, with a non-trivial fraction of clusters in the larger list being genuinely new. In other words, when we widen the scope for risk prompts, the model tends to keep a shared core of risks while continuing to introduce additional, distinct risk clusters, rather than simply rephrasing the same items or regenerating a completely different set.

4) Within-scope repetition: Finally, Figure 7 reports within-scope repetition rates by question type and scope. Repetition is defined as the proportion of items whose cluster has already appeared earlier in the same list.

Here again, we see meaningful differences between types. Tool questions consistently show the highest repetition: after an initial drop at the 5-item scope, the repetition rate increases with scope and reaches its peak for the 30- and 40-item lists. Long tool lists therefore, contain a noticeable amount

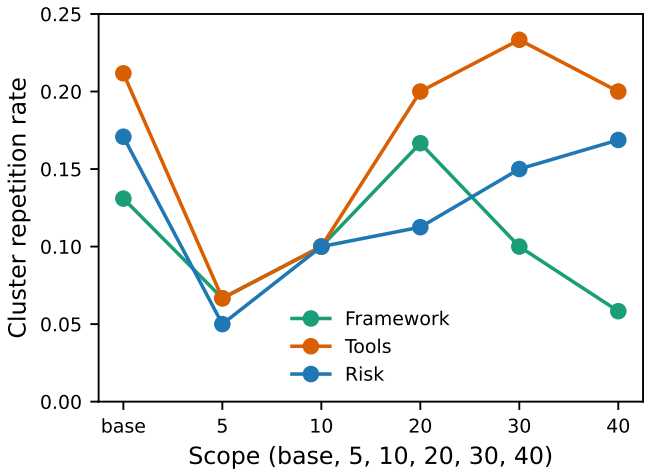

Fig. 7. Within-scope repetition rate by scope size.

of semantic self-repetition. Risk questions show moderate repetition: the rate falls from the base list to 5 items, then gradually rises with scope and stabilizes at an intermediate level. Framework questions have the lowest repetition overall; apart from the very shortest scopes, only a small fraction of items in framework lists fall into clusters that have already appeared.

Taken together, these results suggest the following qualitative picture. Framework questions serve as relatively stable, non-redundant checklists. Tool questions are also stable but become increasingly redundant at large scopes, as the model reuses similar tool concepts multiple times. Risk questions produce more diverse, less stable sets of risks whose exact composition depends on the requested list length, even though a common core of risks is preserved across scopes. Given the small scale of our study, these patterns should be viewed as an initial, quantitatively grounded case study of enumeration behavior rather than a definitive characterization.

V. Discussion: Limitations and Future Work

A. Limitations

This study has several limitations. First, our evaluation is based on a relatively small dataset, consisting of 30 paraphrase questions and 10 enumeration-style questions, which limits the statistical strength and generalizability of the observed trends. Second, enumeration consistency experiments were conducted on a single model configuration, restricting our ability to compare scope behavior across different model families or sizes. Third, while NLI and BLEURT capture logical agreement and semantic similarity, they do not fully reflect practical correctness, completeness, or security impact of the generated advice.

B. Future Work

A primary extension is to substantially expand the dataset, both in the number and diversity of security questions, including additional domains such as cloud security, software supply chains, and incident response. Incorporating multiple evaluation models that would further improve robustness and reduce metric bias. Beyond OpenAI models, evaluating a broader set of LLMs would enable more comprehensive cross-model comparisons.

More broadly, we envision shipping the full evaluation pipeline as a reusable benchmark, including paraphrase generation, enumeration analysis, Arrow-based metric computation, and standardised reporting. Such a benchmark could support repeatable, large-scale consistency evaluations and serve as a foundation for future research on LLM reliability in security-critical workflows. By making consistency evaluation more systematic and accessible, this work aims to contribute toward safer and more dependable use of LLMs in security practice.

VI. Conclusion

In this work, we presented a systematic study of LLM consistency for security-related guidance, focusing on two complementary dimensions: paraphrase robustness and enumeration consistency under scope widening. Our results show that larger and more capable models exhibit stronger logical consistency under paraphrasing, as reflected by increasing NLI scores, even though surface-level semantic similarity (BLEURT) remains low. This divergence highlights that logical agreement and semantic similarity capture different aspects of robustness and must be evaluated jointly when assessing LLM reliability.

For enumeration-style security questions, we find that framework and tool prompts exhibit a stable, canonical core of concepts that persists as list length increases, whereas risk prompts are more open-ended, showing lower core stability but higher conceptual diversity. As scope widens, models tend to recover missing core items rather than replace them, particularly for structured prompts, but tool-related lists also show increasing semantic self-repetition at larger scopes. Overall, these findings suggest that LLMs are well-suited for checklist-style guidance, but the specific tools and risks surfaced can vary substantially with prompt phrasing and requested scope, which may lead to inconsistent or incomplete security advice if outputs are treated as authoritative without validation.

Taken together, our study provides an initial, quantitatively grounded characterization of how LLMs balance stability, diversity, and redundancy when generating security-relevant content, offering practical insights for both users and evaluators of LLM-based security systems.

References

[1] Paulo Alting Von Geusau and Peter Bloem. “Evaluating the Robustness of Question-Answering Models to Paraphrased Questions”. In: *Artificial Intelligence and Machine Learning*. Ed. by Mitra Baratchi et al. Vol. 1398. Cham: Springer International Publishing, 2021, pp. 1–14. ISBN: 978-3-030-76639-9 978-3-030-76640-5. DOI: 10.1007/978-3-030-76640-5_1. (Visited on 10/14/2025).

[2] Junbo Fu et al. “Learning to Paraphrase for Alignment with LLM Preference”. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, 2024, pp. 2394–2407. DOI: 10.18653/v1/2024.findings-emnlp.134. (Visited on 12/14/2025).

[3] Wee Chung Gan and Hwee Tou Ng. “Improving the Robustness of Question Answering Systems to Question Paraphrasing”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 6065–6075. DOI: 10.18653/v1/P19-1610. (Visited on 10/11/2025).

[4] Meltem Kurt Pehlivanoğlu et al. “Comparative Analysis of Paraphrasing Performance of CHATGPT , GPT - 3, and T5 Language Models Using a New CHATGPT Generated Dataset: PARAGPT”. In: *Expert Systems* 41.11 (Nov. 2024), e13699. ISSN: 0266-4720, 1468-0394. DOI: 10.1111/exsy.13699. (Visited on 10/11/2025).

[5] Riccardo Lunardi et al. *On Robustness and Reliability of Benchmark-Based Evaluation of LLMs*. Sept. 2025. DOI: 10.48550/arXiv.2509.04013. arXiv: 2509.04013 [cs]. (Visited on 10/11/2025).

[6] Pranav Rajpurkar, Robin Jia, and Percy Liang. “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 784–789. DOI: 10.18653/v1/P18-2124. (Visited on 10/13/2025).

[7] Pranav Rajpurkar et al. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. (Visited on 10/13/2025).

[8] RyokoAI. *ShareGPT52K*. Hugging Face dataset. License: CC0-1.0 (No Rights Reserved). URL: https://huggingface.co/datasets/RyokoAI/ShareGPT52K (visited on 12/14/2025).

[9] Neha Srikanth, Marine Carpuat, and Rachel Rudinger. “How Often Are Errors in Natural Language Reasoning Due to Paraphrastic Variability?” In: *Transactions of the Association for Computational Linguistics* 12 (Sept. 2024), pp. 1143–1162. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00692. (Visited on 10/14/2025).

[10] Ran Zhang et al. “On Large Language Models Safety, Security, and Privacy: A Survey”. In: *Journal of Electronic Science and Technology* 23.1 (Mar. 2025), p. 100301. ISSN: 1674862X. DOI: 10.1016/j.jnlest.2025.100301. (Visited on 12/14/2025).