

Schema Guardrails in the Wild: Empirically Testing Whether Structural Constraints Tame LLM Output Volatility

Sanjeev Kamath
University of Florida
Gainesville, FL, USA
sanjeev.kamath@ufl.edu

Abstract

Production pipelines using Large Language Models (LLMs) for structured extraction often assume output determinism, yet "output volatility" across identical runs remains a persistent challenge. This study evaluates the "validate-and-retry" strategy, where schema-violating outputs are iteratively rejected until compliance is reached. Using a Travel Risk Assessment pipeline across 30 countries, we compare a 120B LLM (gpt-oss-120b) and an 8B SLM (llama-3.1-8b) at T=1.0. The hypothesis that constraints reduce volatility by 50% is rejected for both models: constraints yield an 11.9% stability gain for the SLM (statistically significant) and 6.3% for the LLM (not significant). This benefit scales inversely with model size; the SLM gains roughly 1.9× more than the LLM because it produces more structural errors for the validator to catch, whereas the larger model is more natively compliant. Furthermore, retry mechanisms increase token costs by 30 to 40%. The dominant residual is a "canonicalization gap," defined as semantically equivalent surface forms that structural validators cannot collapse. We conclude that structural validation alone is insufficient for production-grade consistency and that semantic canonicalization must be a first-class component of the reliability architecture.

ACM Reference Format:

Sanjeev Kamath. 2026. Schema Guardrails in the Wild: Empirically Testing Whether Structural Constraints Tame LLM Output Volatility. In . ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 Introduction

Consider a production dashboard that ingests LLM-generated country risk assessments every morning. On Monday, the pipeline reports that Mexico's active risk categories are {crime, kidnapping}. On Tuesday—same model, same prompt, same underlying advisory and news inputs—it reports {crime, civil_unrest, other}. No ground truth has changed. The model has simply sampled a different output from its distribution. This is *output volatility*: the failure of an LLM to return equivalent structured records across independent runs on identical inputs. The problem is fatal for automated information extraction (IE) pipelines. Analysts, databases,

and downstream dashboards cannot distinguish a resampling artifact from a genuine change in the world, which means every alert is suspect and every trend line is potentially noise. As LLMs become the default engine for structured data extraction in domains from financial compliance to medical coding to geopolitical risk, volatility stops being a curiosity and becomes a production liability.

The dominant engineering response is the *schema constraint architecture*: intercept the model's output with a JSON schema validator, and if the response violates the schema, feed the validator's exact error message back to the model as corrective feedback in an auto-retry loop [2, 14]. This "validate-and-retry" pattern has become a default production recipe and is frequently marketed as a path to near-deterministic structured output. Prior work on structured output largely endorses this direction. Shrimal et al. [10] report a 92% reduction in extraction errors using reflection-based retry, and benchmarks such as JSONSchemaBench [5] measure how reliably models comply with schemas on a single call. What this literature does not measure is the metric that actually matters for production pipelines: whether the same input produces the same output across independent runs. Structural compliance on one run is not the same as consistency across many, and the gap between the two has not been quantified on a realistic, multi-source IE task.

This paper quantifies that gap. The contribution is a live, multi-source **Travel Risk Assessment Pipeline** that synthesizes U.S. State Department advisories and real-time news into structured JSON risk reports for 30 countries, paired with a controlled stability evaluation across two model scales (120B-parameter LLM, 8B-parameter SLM) and two conditions (unconstrained baseline vs. schema-validated retry). All inputs are cached at session level so that observed variance reflects model stochasticity rather than input drift. Stability is measured as the agreement between structured records produced on independent runs of the same input. The central research question is whether schema validation with retry reduces output volatility by at least 50%—a principled "majority contribution" benchmark under which constraints would be the dominant stabilization mechanism, responsible for more improvement than all remaining sources of variance combined.

The 50% threshold is robustly rejected at both model scales. Schema constraints reduce volatility by only 11.9% for the SLM (statistically significant) and 6.3% for the LLM (not statistically significant, within sampling noise). The constraint benefit is itself uneven: the SLM benefits roughly 1.9× more from schema enforcement than the LLM, because the smaller model produces more structural violations for the validator to catch. The larger model is more natively compliant, leaving its residual volatility concentrated in failure modes that no schema validator can address. Retries also consume 30–40% additional tokens, further eroding the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, Washington, DC, USA

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

value proposition of structural guardrails. The remaining volatility in both models is dominated by a *canonicalization gap*: semantically equivalent strings (e.g., "Arauca" vs. "Arauca Department, Colombia") that validators cannot collapse but that downstream consumers treat as distinct. Production IE pipelines that require robust consistency cannot rely on schema enforcement alone; semantic grounding must be treated as a first-class component of the reliability architecture.

The core contributions of this paper are:

- (1) A reproducible multi-source geopolitical risk IE pipeline with session-level input caching and SQLite-backed provenance logging, enabling controlled stability evaluation.
- (2) A quantitative comparison of constrained vs. unconstrained generation across two model scales, using Row Jaccard Similarity with bootstrap confidence intervals.
- (3) Empirical evidence that the *benefit of structural enforcement scales inversely with model size*: smaller models gain roughly 1.9× more from schema validation than larger, more natively compliant ones.
- (4) Identification of the *canonicalization gap*—the class of semantic volatility that purely structural constraints cannot address, persistent across model scales—with concrete engineering directions.

The remainder of the paper is organized as follows. Section 2 reviews related work on LLM hallucination, constrained decoding, and IE pipelines. Section 3 describes the pipeline architecture and data sources. Section 4 details the experimental design, metrics, and statistical methodology. Section 5 presents stability results and cost analysis. Section 6 provides an error taxonomy. Section 7 discusses implications and limitations. Section 8 concludes.

2 Related Work

This work sits at the intersection of three research threads: non-determinism and volatility in LLM outputs, structural enforcement of generated content, and LLM-based information extraction. Each thread has produced useful findings, but none has directly measured whether the dominant production technique—schema validation with error-feedback retry—meaningfully reduces cross-run output volatility in a realistic IE task. This paper fills that measurement gap.

2.1 Non-Determinism and Output Volatility

LLM unreliability has been studied most extensively through the lens of hallucination—the generation of factually incorrect or fabricated content [6]. A related but distinct failure mode has received less attention: *structural volatility*, or inconsistency in the form and categorical choices of outputs across repeated inferences on identical inputs.

Two recent findings motivate this paper directly. Song et al. [11] demonstrate that evaluation benchmarks reporting a single output per example systematically understate true performance variance, and argue that non-determinism must be treated as a first-class concern. Even T=0 settings fail to guarantee identical outputs at scale due to irreducible hardware-level floating-point noise [1]. Together, these results establish that volatility is real and cannot be eliminated by configuration alone—which is precisely why schema-level

stabilization mechanisms have become the default production response. Camuffo et al. [3] add a related diagnosis: prompt phrasing and model selection can shift annotation outcomes by 12–85 percentage points, placing a large residual burden on post-generation guardrails.

These papers establish the problem. None evaluates whether schema-level guardrails actually solve it. This paper provides that evaluation.

2.2 Structural Enforcement: Proactive and Retroactive

Two broad strategies exist for enforcing structured output from LLMs. *Proactive* approaches intervene during generation, masking token logits at each step to guarantee grammar or schema conformance [14]. Query languages such as LMQL formalize this as a programming paradigm [2], and Geng et al. [5] benchmark compliance rates across constrained decoding engines. These approaches guarantee syntactic validity but not semantic correctness—and Tam et al. [12] show that format restrictions can actively *degrade* task performance by forcing the model away from its highest-probability natural language completions. Deutsch et al. [9] extend this finding across 11 models, observing that instruction-tuned models suffer more than base models under constraints.

Retroactive approaches—schema validation with error-feedback retry—are more common in production but less studied empirically. The closest prior work is Shrimall et al. [10], whose PARSE system reports a 92% reduction in extraction errors within the first retry. The metric in that work is structural compliance on a single run. This paper evaluates the less-studied but arguably more important question: whether validation-and-retry improves output *consistency across independent runs* on identical inputs. The distinction matters because single-run compliance can be high while cross-run consistency remains low—exactly the situation this paper’s results document.

2.3 Information Extraction with LLMs

LLMs have become the dominant paradigm for universal information extraction (UIE), replacing earlier rule-based and supervised approaches [7]. Two findings from this literature shape the present work. Liu et al. [8] identify 134 use cases for output constraints in LLM applications and categorize them into low-level (format, length) and high-level (semantic, stylistic) requirements, concluding that addressing only structural constraints leaves a large residual of semantic violations unhandled. This observation directly motivates the canonicalization gap analyzed in Section 6. Wang et al. [13] propose Self-Consistency, aggregating multiple sampled outputs via majority vote as an alternative to structural enforcement; the approach improves reliability but multiplies inference cost linearly with sample count, making it impractical for high-throughput production settings. The pipeline evaluated here targets the opposite regime: single-inference-per-input, where structural guardrails are the only available stabilization mechanism.

3 System Architecture and Data Pipeline

The pipeline is designed around a single experimental constraint: all observed output variation must reflect model stochasticity, not input drift. This section describes how data sources, caching, schema enforcement, and logging are composed to meet that constraint.

3.1 Data Sources and Ingestion

The pipeline synthesizes two sources: U.S. State Department Travel Advisories, fetched via an MCP server, and real-time news articles filtered per country. All data is cached at the session level to ensure observed variation reflects model stochasticity rather than feed drift.

Caching is load-bearing for the experiment. Without it, identical prompts issued on different days would see different news contexts, and any cross-run variation could be attributed to feed drift rather than model behavior. Session-level caching freezes both sources at the start of each experimental run and holds them constant across all 600 inference calls, guaranteeing that the only variable between runs is the model’s sampling process.

3.2 Target JSON Schema

Both sources are synthesized into a single structured JSON record conforming to a strict schema. The fields relevant to the stability evaluation are:

- **country** (string): ISO country identifier.
- **risk_level** (integer, 1–4): State Department advisory level, from Level 1 (Exercise Normal Precautions) to Level 4 (Do Not Travel).
- **event_types** (array of enum): Categorical risk flags drawn from a fixed vocabulary of {crime, terrorism, civil_unrest, kidnapping, natural_disaster, health, conflict, other}.
- **regional_warnings** (array of strings): Free-form named regions or areas with elevated risk.
- **summary** (string): Brief prose synthesis.

The schema deliberately combines two types of fields with different volatility profiles. The `event_types` field is categorically constrained: a finite enum, validator-enforceable, and the primary locus of *structural* volatility. The `regional_warnings` field accepts free-form strings and is the primary locus of *semantic* volatility—the class of inconsistency that schema validators cannot detect. Evaluating both fields within a single record allows the experiment to measure where structural enforcement succeeds and where it reaches its limits.

3.3 Pipeline Components

Figure 1 shows the end-to-end flow. The prompt builder composes the cached advisory, cached news, and a frozen system prompt into a single input. The LLM generates a response at $T=1.0$. In the constrained condition, the response passes through a strict JSON Schema validator; validation failures trigger a retry with the error message appended as corrective feedback. In the unconstrained condition, the validator is bypassed and the raw output is stored directly. All outputs—valid, structurally invalid, and semantically aberrant—are persisted to a SQLite database with full provenance metadata (model, condition, country, run index, timestamp, raw

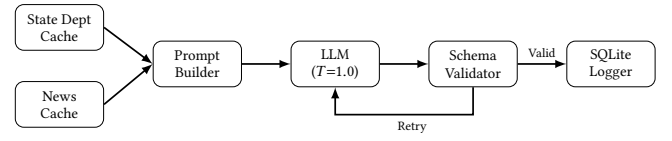


Figure 1: Pipeline architecture. Cached inputs hold both data sources constant across runs. In the constrained condition, validation errors are appended to the model context as corrective feedback and the generation is retried. In the unconstrained condition, the validator is bypassed and raw outputs are logged directly.

response). This logging architecture enables post-hoc metric computation and error analysis across any subset of runs without requiring re-inference.

3.4 Prompt Design

The system prompt is frozen and identical across all runs for a given model. The frozen system prompt instructs the model to synthesize advisory and news content into a JSON object conforming exactly to the `event_types` vocabulary. In the constrained condition, the retry prompt appends the literal validator error—for example, "invalid value 'cartel_violence' in `event_types`; valid options: [crime, terrorism, civil_unrest, kidnapping, natural_disaster, health, conflict, other]"—together with the original prompt and a regeneration instruction. This design ensures that the retry mechanism communicates both the constraint and the specific violation, giving the model the maximum possible information to self-correct.

4 Experimental Design

4.1 Models and Configuration

Two models representing substantially different scales are evaluated:

- **gpt-oss-120b** (LLM): OpenAI’s open-source 120B-parameter instruction-tuned model, accessed via the University of Florida HiPerGator AI API endpoint.¹ This model represents the high end of commercially available API models.
- **llama-3.1-8b** (SLM): Meta’s Llama 3.1 8B-parameter open-weight instruction-tuned model, also accessed via the HiPerGator API. This represents the class of efficient, production-deployable models.

Both models are configured at $T=1.0$ to maximize output entropy. The choice is deliberate. The goal is to stress-test stabilization mechanisms at their logical limits, not to demonstrate the trivial result that low-temperature generation is more consistent [1]. All other generation parameters (`top_p`, `max_tokens`) are fixed identically across conditions to isolate the effect of schema enforcement.

4.2 Experimental Conditions

Two conditions are applied to both models:

¹<https://api.ai.it.ufl.edu/v1>

- (1) **Unconstrained (Baseline).** The model generates a response directly from the prompt. The output is parsed as JSON for storage and metric computation, but no validation or retry is applied. Structurally malformed outputs are logged as-is.
- (2) **Constrained.** The output is passed through a strict JSON Schema validator. If validation fails—missing keys, wrong types, or hallucinated enum values—the validator serializes the specific error and triggers a retry. The error message is appended to the conversation context as corrective feedback. The loop continues until the output passes validation or a maximum of 3 retry attempts is exhausted.

4.3 Evaluation Scope

The experiment covers **30 target countries** spanning all major geographic regions and advisory risk tiers (Level 1: Exercise Normal Precautions through Level 4: Do Not Travel). The country set was fixed prior to any data collection to avoid selection bias. Five independent runs are generated per country per condition per model, yielding $5 \times 2 \times 2 \times 30 = 600$ baseline inference calls plus additional retry calls in the constrained condition.

4.4 Statistical Analysis

To assess whether observed stability differences are statistically meaningful rather than artifacts of sampling variance, **95% Bootstrap Confidence Intervals** (BCIs) are constructed for all reported means using $B = 1,000$ bootstrap resamples of the country-level stability scores [4].

Volatility reduction metric

The volatility reduction is defined as the proportion of residual instability that schema constraints eliminate:

$$\text{Reduction} = \frac{\bar{J}_{\text{constrained}} - \bar{J}_{\text{unconstrained}}}{1 - \bar{J}_{\text{unconstrained}}}$$

where $\bar{J}$ denotes the mean Row Jaccard Similarity across all countries. A value of 50% indicates that constraints close half the gap between observed baseline stability and perfect reproducibility ($J=1.0$). This residual-improvement formulation is more informative than a simple relative change: it measures progress toward the practical goal of full consistency, rather than relative change against an arbitrary baseline.

5 Evaluation Results

5.1 Primary Stability Results

Table 1 summarizes the core Row Jaccard Similarity results across models and conditions

Table 1: Row Jaccard Stability: Means and 95% Bootstrap CIs. Higher = more stable.

Model	Condition	Mean	95% BCI	Δ
llama-3.1-8b	Unconstrained	0.6016	[0.579, 0.623]	+11.9%
	Constrained	0.6488	[0.623, 0.673]	
gpt-oss-120b	Unconstrained	0.7007	[0.673, 0.728]	+6.3%
	Constrained	0.7197	[0.702, 0.739]	

Hypothesis rejection: Neither model approaches the 50% volatility reduction threshold. The best-case result (llama-3.1-8b constrained) achieves only 11.9% improvement; the LLM achieves 6.3%. The hypothesis is robustly rejected. The remainder of this section unpacks why—and Section 6 traces the residual volatility to a single dominant cause that schema validation cannot address.

llama-3.1-8b (SLM): The constrained condition produces a statistically significant improvement. The BCIs are non-overlapping: [0.579, 0.623] for unconstrained vs. [0.623, 0.673] for constrained. The 11.9% gain indicates that retry loops *do* provide measurable benefit at this model scale.

gpt-oss-120b (LLM): The constrained condition produces no statistically significant improvement. The BCIs overlap—[0.673, 0.728] vs. [0.702, 0.739]—indicating that the 6.3% mean difference falls within sampling noise. Schema constraints provide no reliable benefit to the 120B model in this setting, despite its higher absolute stability in both conditions.

5.2 Cross-Model Comparison

Table 2 compares stability across model scales. In both conditions, the larger gpt-oss-120b achieves higher Row Jaccard Stability than llama-3.1-8b.

Table 2: Cross-model stability comparison across all conditions.

Condition	llama-3.1-8b	gpt-oss-120b
Unconstrained	0.6016	0.7007
Constrained	0.6488	0.7197
LLM advantage (unconstrained)	+16.5%	
LLM advantage (constrained)	+10.9%	

The 120B model is more stable than the 8B model by 16.5% unconstrained and 10.9% constrained. This is the expected direction: larger, more capable models produce more consistent structured outputs at high temperature, consistent with their generally tighter calibration on instruction-following tasks.

The more informative observation is that the LLM’s advantage *shrinks* under constraints (16.5% $\rightarrow$ 10.9%). The SLM closes part of the gap because it has more structural violations for the validator to catch and correct—an asymmetry quantified directly in Section 5’s constraint benefit analysis and traced to specific failure modes in Section 6. The residual gap that constraints cannot close reflects the canonicalization volatility that affects both models and that schema validation is architecturally unable to address.

5.3 Constraint Benefit Scales Inversely with Model Size

A related finding is the differential benefit from constraints across model scales. Table 3 quantifies the *relative benefit multiplier*—the ratio of percentage improvement in the SLM vs. LLM from applying constraints.

The SLM benefits roughly 1.9 $\times$ more from schema constraints than the LLM. We interpret this through the mechanism of the retry loop: the 8B model more frequently generates structurally

Table 3: Constraint benefit by model scale.

Model	Volatility Reduction	SLM/LLM Ratio
llama-3.1-8b (SLM)	11.9%	$\approx 1.9\times$
gpt-oss-120b (LLM)	6.3%	

deviant outputs (wrong categories, missing fields), giving the validator more opportunities to intervene and correct. The 120B model already generates schema-compliant structures at a high native rate; thus, the retry loop has fewer purely structural violations to fix. The residual instability in the LLM is driven by *semantic* variance—specifically the canonicalization failures characterized in Section 6—which structural validation cannot address. For the SLM, this 11.9% stability gain required approximately 1.3 additional generation passes on average, resulting in a 30–40% increase in total token consumption due to the inclusion of validator error messages in the retry context.

6 Error Analysis

Post-hoc inspection of the logged outputs reveals that the observed volatility falls into three qualitatively distinct failure modes. Table 4 summarizes them and notes which are addressable by schema enforcement. The three subsections that follow expand each mode with concrete examples drawn from the experimental data.

6.1 Type 1: Structural Violations

Structural violations are malformed outputs that a JSON schema validator can detect and trigger a retry for. Observed examples include: missing required keys (the `event_types` array omitted entirely), hallucinated enum values ("`cartel_violence`" instead of one of the eight valid event types), and type mismatches (`risk_level` returned as a string instead of an integer).

These violations are significantly more common in llama-3.1-8b than in gpt-oss-120b. In the SLM’s unconstrained condition, approximately 18% of generations contained at least one structural violation. In the LLM’s unconstrained condition, the rate was approximately 6%. This $3\times$ gap explains directly why the SLM benefits from retry loops and the LLM does not: the LLM simply produces fewer Type 1 failures for the validator to catch. The residual volatility in the LLM’s outputs is therefore concentrated in Types 2 and 3, which schema enforcement cannot address.

6.2 Type 2: Categorical Interpretation Variance

Even when outputs are structurally valid, the same underlying advisory text can elicit different categorical assignments across runs. A report of “organized crime groups operating in border regions” may be tagged `crime` in one run and `terrorism` in another, depending on subtle differences in how the model parses the prose at $T=1.0$. Both assignments are defensible interpretations; neither is a schema violation; the validator has no basis to prefer one over the other.

This form of volatility is partially addressable by a more sophisticated retry loop—one that encodes semantic rules (e.g., “cartel-related activity must map to `crime`, not `terrorism`”). The current implementation does not implement such rules, so Type 2 errors

pass through the validator untouched. Quantifying their exact rate would require ground-truth category labels for each input, which are not available for this corpus.

6.3 Type 3: Canonicalization Failures

The most pernicious form of volatility affects the `regional_warnings` free-form string list. Schema constraints enforce that this field is an array of strings, but they cannot enforce that semantically equivalent strings are represented identically. Across runs, the same geographic entity appears under multiple surface forms:

- "Arauca" vs. "Arauca department" vs. "Arauca Department, Colombia"
- "FATA region" vs. "Federally Administered Tribal Areas" vs. "tribal border areas"
- "Northern Border" vs. "northern border states" vs. "Mexico-US border region"

A human reader recognizes all three variants in each group as references to the same underlying place. A set-based Jaccard comparison does not: it treats "Arauca" and "Arauca Department, Colombia" as entirely distinct tokens, counting each occurrence against stability. This canonicalization failure is fundamentally a semantic problem, not a structural one. Resolving it requires a downstream normalization layer—either a named entity linker (NEL) that maps surface forms to canonical geographic IDs, or few-shot canonicalization examples embedded in the prompt itself. Neither is provided by a schema validator.

Type 3 failures dominate the residual volatility in both models’ outputs and are the primary reason the 50% hypothesis was rejected. No amount of structural validation reduces them.

7 Discussion

7.1 Why the 50% Hypothesis Failed

The 50% hypothesis was motivated by the intuition that a tight structural feedback loop would dramatically narrow the model’s effective output space. This intuition was partially correct but fundamentally incomplete. Schema constraints do narrow the *structural* output space: they eliminate Type 1 failures reliably and catch the malformed JSON that would otherwise break downstream systems. What they cannot narrow is the *semantic* output space—the variance in how the model reads, interprets, and synthesizes input text into categorical and free-form outputs. At $T=1.0$, semantic variance dominates. Structural validators are blind to it by design.

The dominant residual is the canonicalization gap characterized in Section 6. Free-form fields such as `regional_warnings` accumulate surface-form variation across runs ("Arauca" vs. "Arauca Department, Colombia") that a JSON schema cannot collapse and a Jaccard-style consumer cannot ignore. This single failure mode explains why the constraint benefit is bounded for both models: once Type 1 violations are removed, what remains is a class of semantic volatility that no purely structural mechanism can address. The 120B model’s higher absolute stability narrows the structural component proportionally more on its own—which is precisely why constraints add less marginal value to it (Section 5)—but does not close the canonicalization gap, leaving its constrained-vs-unconstrained delta inside sampling noise.

Table 4: Taxonomy of observed failure modes. Type 1 violations are directly addressable by schema validation with retry; Types 2 and 3 are not.

Failure Mode	Description	Example	SLM / LLM Rate	Addressable?
Type 1: Structural Violation	Malformed output the validator can detect: missing keys, hallucinated enum values, wrong field types.	"cartel_violence" in event_types	~18% / ~6%	Yes (by retry)
Type 2: Categorical Interpretation Variance	Structurally valid, but the same input text receives different categorical assignments across runs.	"organized crime" → crime on one run, terrorism on another	Not measured directly	Partially (semantic rules)
Type 3: Canonicalization Failure	Semantically equivalent free-form strings expressed in different surface forms.	"Arauca" vs. "Arauca Department, Colombia"	Dominates residual variance	No

The finding is consistent with Tam et al. [12], who argue that the fundamental bottleneck in structured IE is semantic alignment rather than syntactic compliance. Liu et al. [8] similarly distinguish between “low-level” constraints (format, structure) and “high-level” constraints (semantics, style), noting that practitioners overwhelmingly need both but existing tooling predominantly addresses the former. The results in this paper give that distinction a concrete empirical magnitude: on a realistic IE task, even the model with stronger native structural compliance leaves the high-level gap essentially untouched, and the smaller model’s gains from structural enforcement plateau well below the 50% threshold.

7.2 Limitations and Future Work

Three limitations bound the scope of the conclusions drawn here, and each points to a concrete direction for follow-up work.

Single task domain.

The evaluation covers one real-world task (geopolitical travel risk) with one strict schema. Generalizing the findings to other IE domains—financial compliance extraction, clinical coding, legal contract parsing—would require replicating the constrained-vs-unconstrained comparison on those schemas. The capacity-consistency tradeoff in particular may behave differently when the target schema has a larger categorical vocabulary or more hierarchical structure. *Future direction*: a cross-domain stability benchmark that applies the Row Jaccard methodology to at least three structurally distinct IE schemas.

Retroactive constraints only.

This paper evaluates schema validation with error-feedback retry. Proactive constrained decoding (grammar-constrained logit masking) may achieve higher stability at comparable cost by preventing structural violations at generation time rather than correcting them retroactively. The two approaches have not been compared head-to-head on cross-run consistency. *Future direction*: a controlled comparison of retroactive retry vs. proactive constrained decoding on the same pipeline, measuring both stability gain and end-to-end latency.

8 Conclusion

This paper evaluated whether schema validation with error-feedback retry—the default engineering response to LLM output volatility—meaningfully reduces cross-run inconsistency in a realistic structured information extraction pipeline. The evaluation was conducted on a live Travel Risk Assessment Pipeline spanning 30 countries, two model scales, and two experimental conditions, with all inputs cached to isolate model stochasticity from input drift.

Three findings define the contribution:

- (1) **Schema constraints deliver modest stability, not dominant stability.** The 50% volatility reduction hypothesis was robustly rejected at both model scales. Constraints produced an 11.9% improvement for the SLM (statistically significant) and 6.3% for the LLM (not statistically significant).
- (2) **Constraint benefit is inversely related to model scale.** The smaller 8B SLM benefits roughly 1.9× more from schema enforcement than the 120B LLM, because the SLM produces more structural violations for the validator to catch. The larger model is more natively compliant, leaving its residual volatility concentrated in failure modes that structural validation cannot address.
- (3) **The canonicalization gap is the dominant residual.** Semantically equivalent surface forms in free-form fields ("Arauca" vs. "Arauca Department, Colombia") account for most of the volatility that remains after schema enforcement, and they persist at both model scales. Structural validators are architecturally incapable of closing this gap.

The practical implication is narrow and concrete. Schema validation belongs in production IE pipelines as a structural safety net, but teams that require genuine output consistency must invest in semantic grounding—entity linking, canonicalization layers, controlled vocabularies—as a first-class component of their reliability architecture, not as an optional refinement.

This project was completed at the University of Florida under the supervision of Dr. Christian Grant as a part of CIS 6930 - Data Engineering, in the Spring 2026 semester

References

- [1] Berk Atıl, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. 2025. Non-Determinism of “Deterministic” LLM System Settings in Hosted Environments. In *Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP)*. Association for Computational Linguistics, 135–148. doi:10.18653/v1/2025.eval4nlp-1.12
- [2] Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2023. Prompting is programming: A query language for large language models. *Proc. ACM Program. Lang.* 7, PLDI, Article 186 (2023), 24 pages. doi:10.1145/3591300
- [3] Arnaldo Camuffo, Alfonso Gambardella, Saeid Kazemi, Jakub Malachowski, and Abhinav Pandey. 2026. Variance-Aware LLM Annotation for Strategy Research: Sources, Diagnostics, and a Protocol for Reliable Measurement. (2026). arXiv:2601.02370 <https://arxiv.org/abs/2601.02370>
- [4] Bradley Efron and Robert J. Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC Press. doi:10.1201/9780429246593
- [5] Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. 2025. JSON-SchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models. In *Conference on Language Modeling (COLM)*. <https://openreview.net/forum?id=FkOajqKoio>
- [6] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.* 55, 12, Article 248 (2023), 38 pages. doi:10.1145/3571730
- [7] Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024. A Simple but Effective Approach to Improve Structured Language Model Output for Information Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 5133–5148. doi:10.18653/v1/2024.findings-emnlp.295
- [8] Michael Xieyang Liu, Frederick Liu, Alexander J. Fiannaca, Terry Koo, Lucas Dixon, Michael Terry, and Carrie J. Cai. 2024. “We Need Structured Output”: Towards User-centered Constraints on Large Language Model Output. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. ACM New York, NY, USA, Article 394, 8 pages. doi:10.1145/3613905.3650935
- [9] Maximilian Schall, Daniel Deutsch, et al. 2025. The Hidden Cost of Structure: How Constrained Decoding Affects Language Model Performance. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*. 1074–1084. doi:10.26615/978-954-452-098-4-124
- [10] Anubhav Shrivastava, Aryan Jain, Soumyajit Chowdhury, and Promod Yenigalla. 2025. PARSE: LLM Driven Schema Optimization for Reliable Entity Extraction. In *Proceedings of EMNLP Industry Track*. Association for Computational Linguistics, 2749–2763.
- [11] Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2024. The Good, The Bad, and The Greedy: Evaluation of LLMs Should Not Ignore Non-Determinism. (2024). arXiv:2407.10457 <https://arxiv.org/abs/2407.10457>
- [12] 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 *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, 11244–11265. doi:10.18653/v1/2024.findings-acl.668
- [13] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Dai. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=1PL_EzmSL
- [14] Brandon T. Willard and Rémi Louf. 2023. Efficient guided generation for large language models. (2023). arXiv:2307.09702 <https://arxiv.org/abs/2307.09702>