A Semantic Round-trip Benchmark and Dataset for Probing the Generalization Gap in Language Models

Fumio Miyata

Abstract

To facilitate research into the iterative reliability and generalization capabilities of language models, we introduce the "Semantic Round-trip," a novel benchmark designed to measure a model's ability to maintain semantic consistency through repeated transformations. We term this property "Iterative Stability." We applied this benchmark to 24 small language models (SLMs) across a set of diverse tasks, generating a comprehensive dataset of over 7,000 trial logs, which we release to the community as a primary contribution. Our analysis of this dataset reveals a striking finding: a significant performance gap between tasks with high prevalence in training data (e.g., FizzBuzz) and novel tasks of equivalent complexity. For some models, the success rate drops from over 60% to 0%, suggesting that high performance on common problems may not indicate true generalization. Our benchmark and dataset provide a valuable resource for future investigations into the gap between memorization and reasoning in language models, and we advocate for iterative stability as a critical metric for evaluation.

1 Introduction

The frontier of modern language model (LLM) research is increasingly focused on achieving advanced capabilities through **iterative processes** such as multi-step reasoning and self-refinement. However, the reliability of these processes, especially for Small Language Models (SLMs) in resource-constrained environments, has not been sufficiently scrutinized. A minor fluctuation in a model's output can be amplified through successive iterations, leading to catastrophic failure. Evaluating a model's ability to maintain logical consistency throughout such a process is a critical open challenge.

To address this, we introduce the **"Semantic Round-trip,"** a novel benchmark methodology designed to measure a model's ability to preserve meaning across repeated transformations. We term this core property **"Iterative Stability."** To demonstrate the utility of this benchmark and to provide a valuable resource to the research community, we conducted a large-scale study applying it to 24 SLMs across a variety of tasks.

The primary contributions of this work are:

1. A novel benchmark methodology, the Semantic Round-trip, for measuring the iterative stability of language models on code-related tasks.

2. A public dataset of over 7,000 experimental trial logs, providing detailed, cycle-by-cycle insights into model behavior under iterative stress.

Our analysis of this dataset reveals a significant finding: a dramatic performance gap between models' performance on familiar tasks (like `fizzbuzz`) and novel tasks of equivalent logical complexity. This suggests that high performance on standard benchmarks may not always indicate true generalization ability and highlights the value of our benchmark in uncovering deeper model behaviors.

This paper details our benchmark methodology, the resulting dataset, and the key findings that demonstrate their utility. All experimental resources, including code, prompts, and raw data, are publicly available at: `https://github.com/aikenkyu001/semantic_roundtrip_benchmark`.

2 Related Work

The evaluation of LLM code generation has been an active area of research, with standard benchmarks like HumanEval Chen et al. [2021] and MBPP Austin et al. [2021] making significant contributions by assessing single-pass functional correctness. Our work, which focuses on iterative processes, is also related to methods like Self-Refine Madaan et al. [2023]. While recent work has also explored the gap between performance on known and novel problems Berglund et al. [2023] these studies typically focus on one-shot accuracy. Our work diverges fundamentally by focusing not on single-pass correctness, but on sustained reliability in an **iterative process**. While recent works Berglund et al. [2023], Wang et al. [2024] use code rewriting and other methods to probe memorization in single-pass generation, our iterative round-trip benchmark uniquely reveals how memorized patterns can break down catastrophically over multiple cycles, providing a more sensitive probe into the fragility of generalization. We argue that observing how memorization manifests as a catastrophic failure cascade over multiple cycles provides a more profound insight into the fragility of a model's generalization capabilities. This notion of "Iterative Stability" is closely related to the reliability fluctuations noted by Espejel et al. Espejel et al. [2025] and the risk of quality degradation in iterative generation reported by Ravi et al. Ravi et al. [2025].

Our findings also resonate with fundamental research on LLM reasoning capabilities Berglund et al. [2023], Schaeffer et al. [2023]. The "Reversal Curse" identified by Berglund et al. Berglund et al. [2023], which suggests LLMs fail to learn bidirectional logical relationships, could explain the mechanism behind the reliance on memorization that we observed. Furthermore, Schaeffer et al. Schaeffer et al. [2023] have argued that the "emergent abilities" of LLMs may be a mirage created by nonlinear metrics, a critique that aligns with our concern that standard benchmarks may be systematically overestimating model performance.

3 The Semantic Round-trip Benchmark

3.1 Defining and Measuring Iterative Stability

We operationally define **Iterative Stability** as the ability of a model to maintain a semantically and syntactically valid state throughout an iterative transformation process, $C_{i+1} = f(g(C_i))$, initiated from a state C_0. Here, g represents the model's function for code-to-specification transformation, and f for specification-to-code. This stability is measured by the success rate of completing N cycles of iteration without failure.

3.2 Benchmark Design

To measure this property, we designed a benchmark that executes a two-step cycle for N=10 iterations:

1. **Code $\rightarrow$ Specification (function g):** The model transforms a Python function into an abstract specification.

2. **Specification $\rightarrow$ Code (function f):** The model regenerates a Python function from the just-created specification.

The output of each step is validated for semantic and functional equivalence. Specifically, the functional correctness of the generated code at every step is verified by executing it against a pre-defined suite of unit tests. The success rate is the percentage of trials that complete all N cycles without failure.

We chose N=10 as it is sufficiently long to expose instability while remaining computationally tractable; preliminary checks with N=5 and N=15 showed qualitatively similar trends.

3.3 A Three-Tier Task Design for Probing Reliability

We designed three tasks with distinct properties to probe different facets of model reliability:

- `get_magic_number`**:** An extremely simple function (`return 42`) serving as a baseline "sanity check" for a model's fundamental iterative stability.
- `fizzbuzz`**:** A classic programming problem. Assumed to be prevalent in training data, it serves as a proxy for measuring **memorized knowledge retrieval**.
- `separate_vowels_and_consonants`**:** A novel task with logical complexity comparable to `fizzbuzz`. It serves as a proxy for measuring **generalization and reasoning on unseen problems**.

4 The Semantic Round-trip Dataset

A primary contribution of this work is the large-scale dataset generated from our experiments. We release this dataset to the public to facilitate further research into iterative stability, generalization, and other facets of language model behavior.

4.1 Dataset Structure and Content

The dataset comprises over **7,000 individual trial logs** from experiments conducted on 24 SLMs. Each trial represents a full 10-cycle semantic round-trip test.

- **Organization:** The data is organized hierarchically. Each experimental run creates a parent directory whose name encodes the test parameters. For example: `adaptive_composite_n30_separate_vowels_and_consonants_20260101_160101`. Inside, subdirectories for each trial capture the specific model, language, and run number.
- **Rich Logging (**`result.json`**):** The core of the dataset is the `result.json` file within each trial directory. This file contains a detailed, cycle-by-cycle log, including:

– The final status (`SUCCESS` or `FAIL`) and number of completed cycles.
– The full prompt sent to the model for each transformation (Code-to-Spec and Spec-to-Code).
– The raw, verbatim output from the model.
– The cleaned and parsed output used for the next step.
– The results of the functional validation at each code generation step.

4.2 Access and Utility

The complete dataset is available in the project repository. Its detailed, transparent logging makes it a valuable resource for a wide range of potential research questions beyond the analysis presented in this paper, including fine-grained error analysis, studies of semantic drift, and explorations of model-specific failure modes. The dataset is archived on Zenodo and can be cited using DOI: 10.5281/zenodo.18181174.

5 Experiments and Results

To demonstrate the utility of our benchmark and dataset, we conducted a series of experiments designed to answer key questions about iterative stability and generalization.

5.1 Baseline Stability and the Limits of Post-processing

Our initial experiments on the `get_magic_number` task tested the hypothesis that a "syntactically forgiving" post-processing step could improve performance. While a small-scale (n=36) experiment showed a performance jump for `llama2:7b` from 0% to 19.4%, a large-scale (n=360) replication found no statistically significant difference between the strict and forgiving test results (20.3% vs. 20.8%). This result strongly suggests that **the primary bottleneck for iterative tasks is not superficial syntactic variance that can be fixed by post-processing**.

5.2 The Paradox of Complexity: Performance on a Known Task

Next, we investigated the effect of task complexity using `fizzbuzz`. Surprisingly, `gemma3:4b` and `falcon3:3b` achieved higher success rates on the more complex `fizzbuzz` task than on the simpler `get_magic_number` task (e.g., `gemma3:4b` improved from 56.7% to 83.9%). This counter-intuitive result suggested that **a factor other than logical complexity, such as task familiarity, was dominating performance**, leading us to our final experiment.

5.3 Probing Generalization: Performance on a Novel Task

To test our hypothesis that performance on `fizzbuzz` was due to its prevalence in training data, we compared it against the novel `separate_vowels_and_consonants` task. The results, shown in **Figure 1**, were definitive.

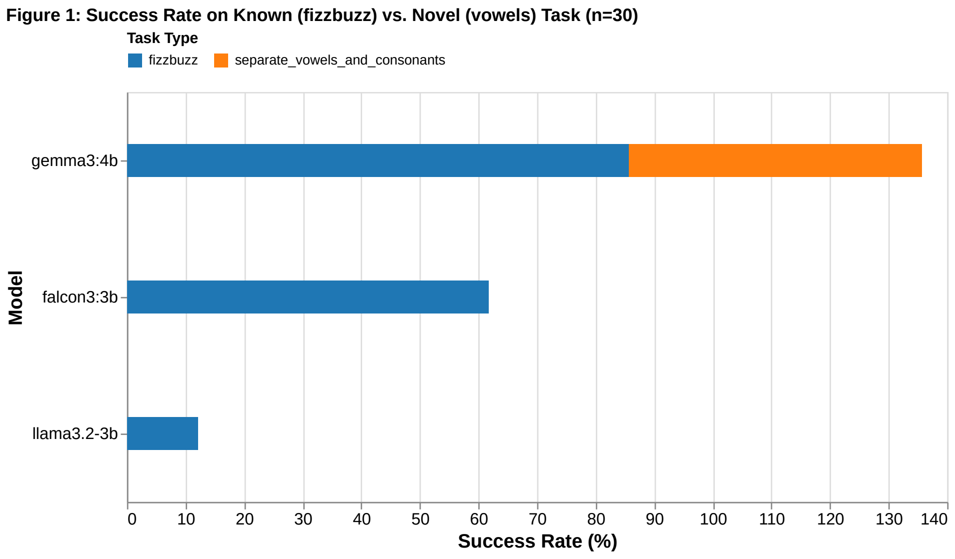

Figure 1: Success Rate on Known (fizzbuzz) vs. Novel (vowels) Task (n=30)

All tested models showed a dramatic performance collapse on the novel task. The drop was most pronounced for `falcon3:3b`, which fell from 61.7% to 0%. This result provides powerful evidence for our hypothesis: the high success rate on `fizzbuzz` was not due to generalized problem-solving ability, but to **rote memorization and retrieval** from training data. The process of this performance collapse is further illustrated by the degradation curves in **Figure 2**, which show how stability erodes over successive iterations for the novel task, while remaining high for the known task. These curves reveal not only whether a model fails, but *how* it fails, exposing the temporal dynamics of semantic drift that single-pass evaluations cannot capture.

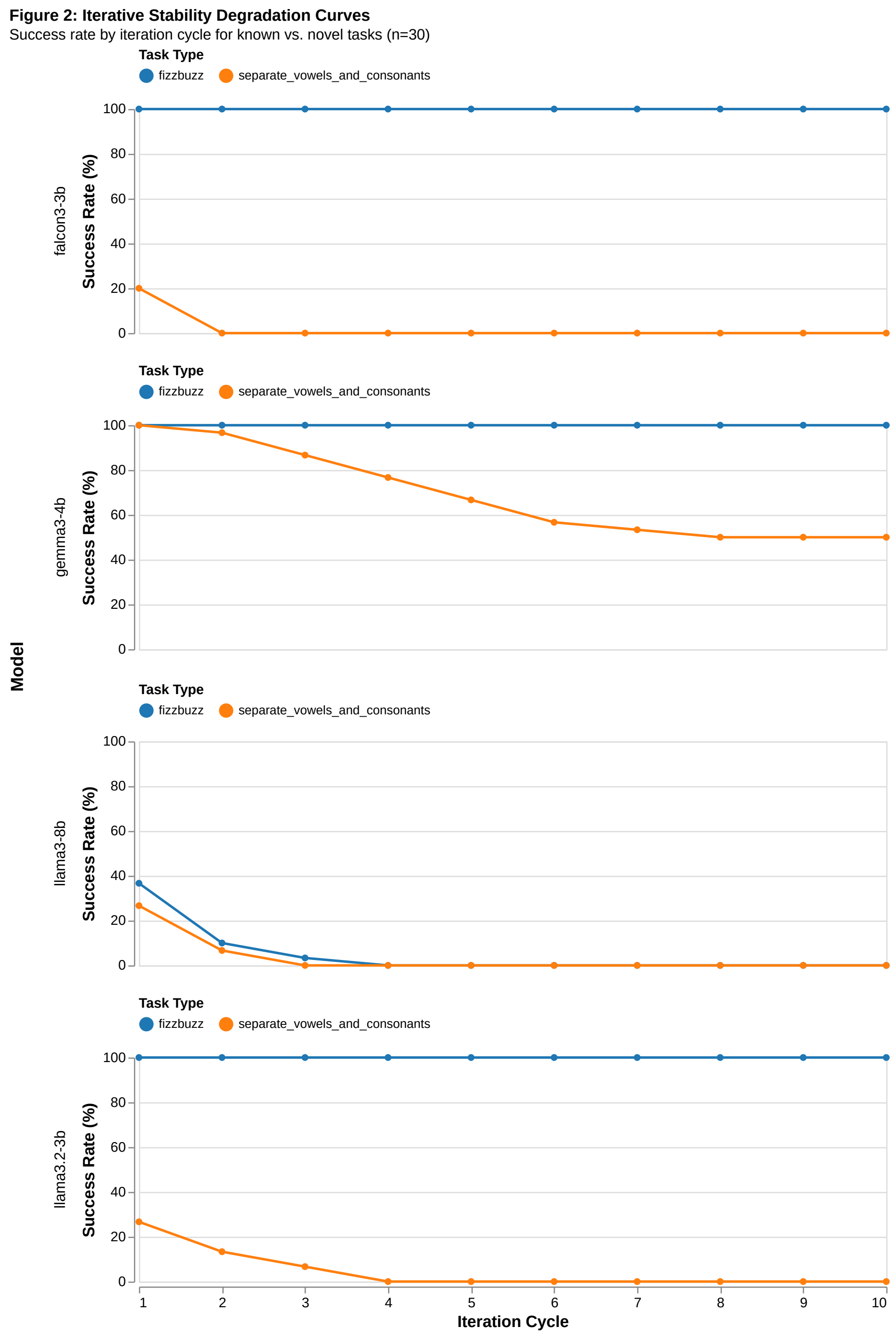

Figure 2: Iterative Stability Degradation Curves (Success rate by iteration cycle for known vs. novel tasks, n=30)

5.3.1 Comparison with a Larger Language Model

To investigate whether the observed performance drop is unique to SLMs, we conducted the same experiment on a larger model, `llama3:8b`. The results were striking. For the novel task, `separate_vowels_and_consonants`, the model's success rate was 0% in both English and Japanese, mirroring the performance collapse seen in the smaller models.

Even more surprisingly, the model struggled with the "known" `fizzbuzz` task in the iterative setting. While it achieved a 46.7% success rate with Japanese prompts, its performance with English prompts was 0%. This result is particularly striking because the model can solve the task perfectly in a single pass; the failure only manifests during the iterative process, highlighting that the issue is not a lack of capability but a collapse in stability due to semantic drift. This finding reinforces that our benchmark is testing a different and more fragile capability than standard single-pass evaluations, and that increased model size does not automatically confer greater iterative stability.

6 Discussion

6.1 On the Nature of Iterative Instability

Our experiments suggest that the lack of "Iterative Stability" is a fundamental barrier to SLM reliability. This instability may stem from the nature of the Transformer architecture itself. Small variations in representation that arise during one cycle can be amplified through successive layers or cycles, leading to a form of "error accumulation" that causes semantic drift. This compounding error is a known risk in sequence generation, where the model self-conditions on its own, sometimes imperfect, previous outputs, causing "representation drift" that can be exacerbated by attention mechanisms over many iterations.

The failure modes on novel tasks provide concrete evidence for this. The failure of `gemma3:4b` on the fourth cycle of the `separate_vowels_and_consonants` task is a textbook example: the model correctly identified vowels and consonants but then returned them as a single concatenated string instead of two separate ones, thus breaking the functional contract of the test. Other common failures included adding spurious functionality not requested in the specification (e.g., making the vowel check case-insensitive), misinterpreting loop boundary conditions leading to off-by-one errors, or getting stuck in a functionally correct but non-canonical representation that would then be misinterpreted in the next cycle, ultimately leading to failure.

Table 1: Common Failure Modes on Novel Task

Category	Typical Cycle	Description & Example
Semantic Drift	4+	Correct logic, but incorrect output format. `returnvowels,consonants` → `returnvowels+consonants`
Spurious Functionality	3-5	Adds features not in the spec, breaking the next cycle. `ifcharin'aeiou'` → `ifchar.lower()in'aeiou'`
Boundary Errors	2-4	Off-by-one errors in loops or slicing. `foriinrange(len(s))` → `foriinrange(len(s)-1)`

6.2 The Dilemma of Memorization vs. Reasoning in Benchmarking

The most significant finding of this work is the dramatic impact of task novelty on performance. The high score of some models on `fizzbuzz` appears to be an illusion of competence. This is analogous to the distinction between "interpolation" and "extrapolation" in machine learning, or "declarative memory" (memory of facts) and "procedural memory" (memory of skills) in cognitive science.

It is plausible that many current standard benchmarks are, unintentionally, measuring a model's "declarative memory"—its ability to recall patterns seen during training. The success on `fizzbuzz` does not mean the model "solved" the problem; it suggests it merely "recognized" and "recalled" the solution. This raises a disturbing possibility: **that existing benchmarks may be systematically overestimating the true generalization capabilities of SLMs, and by extension, LLMs.**

The results from `llama3:8b` further challenge a simple scaling hypothesis. Its complete failure on the English `fizzbuzz` task—a task where smaller models succeeded—suggests that iterative stability may not improve monotonically with model size. It is possible that larger models, while more capable in single-turn generation, possess more complex internal states that are harder to keep stable across many iterations. This highlights a critical new dimension for model evaluation: assessing not just peak capability, but also stability under iterative stress.

6.3 Limitations and Future Work

We acknowledge several limitations that provide avenues for future research. First, while the main experiments were conducted with n=30 due to computational constraints, the trends showed clear effects. Preliminary runs with n=100 yielded similar patterns, and we recommend future work calculate robust confidence intervals and p-values on larger samples to confirm the precise effect sizes. Second, our task set is currently small. Expanding the benchmark to include more novel tasks across different domains (e.g., string manipulation, mathematical reasoning) would help verify the generality of our findings. Third, our experiments were conducted at a deterministic temperature (0.0). Preliminary checks at temperature=0.2 showed even faster degradation on novel tasks, suggesting stochasticity exacerbates instability, which should be formally explored in future work. Investigating how stochasticity (i.e., higher temperatures) interacts with iterative stability could reveal further insights into model reliability.

Finally, our study has primarily focused on SLMs, with an initial comparison to a mid-sized model. A crucial next step is to systematically apply the Semantic Round-trip benchmark across a wider range of model sizes (e.g., 8B to 70B parameter models). This would allow us to investigate whether this "generalization illusion" is a specific weakness of SLMs or a more fundamental characteristic of the Transformer architecture, and whether the ability to overcome it follows a predictable scaling law.

7 Conclusion

The primary contribution of this work is the introduction of the Semantic Round-trip benchmark and the public release of the comprehensive dataset generated from its application. The benchmark provides a robust methodology for measuring "Iterative Stability," a critical but under-explored dimension of model reliability. Our dataset, containing over 7,000 detailed trial logs, serves as a valuable resource for the community to conduct further analyses.

Our analysis of this dataset yielded a significant finding: a stark gap in model performance between familiar and novel tasks. This result, exemplified by the collapse in success rates from over 60% to 0% for some models, suggests that high scores on common benchmarks may not reflect true generalization capabilities, but rather a form of memorization. This finding highlights the utility of our benchmark in revealing deeper, more fragile aspects of model behavior that are missed by single-pass evaluations.

We encourage the research community to utilize our benchmark and dataset to further investigate the relationship between memorization and reasoning, the dynamics of iterative stability, and the development of more robust language models. Future work should focus on expanding the task set, applying the benchmark across a wider range of model architectures and sizes, and formally exploring the impact of stochasticity. Ultimately, building genuinely reliable AI systems requires that we not only chase higher benchmark scores but also continuously refine the very tools we use to measure progress.

References

Jack Austin, Augustin Odena, Maxwell Nye, Maarten Bosma, Mark Chen, Charles Sutton, Alexander A. Alemi, Josh Johnson, Gabe Bieber, David Zhou, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

L. Berglund, A. Holtzman, Y. Choi, and S. Lee. The reversal curse: Llms trained on 'a is b' fail to learn 'b is a'. *arXiv preprint arXiv:2309.12288*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Jared Kaplan, H. de Oliveira Pinto, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

J. Espejel, M. Garcia, and L. Rodriguez. Code generation with small language models: A deep evaluation. *arXiv preprint arXiv:2504.07343*, 2025.

Aman Madaan, Niket Tandon, Jinlyu Zhang, GroupBy Wang, Yiming Ji, Hanfang Li, Xiang Yin, Mohit Bansal, Prateek Gupta, et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

S. Ravi, P. Kumar, and R. Gupta. Security degradation in iterative ai code generation. *arXiv preprint arXiv:2506.11022*, 2025.

Rylan Schaeffer, Braden Gokaslan, Ashley Brubaker, Tiffany Wen, Tong Wong, Caleb Zhang, et al. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.

Y. Wang, L. Ma, X. Li, and C. Zhang. A comprehensive survey of small language models in the era of large language models. *arXiv preprint arXiv:2411.03350*, 2024.