Algorithmic Self-Correction in LLMs: A Model That Learns to Diagnose Its Own Flawed Reasoning

Final Draft - November 2025
Correspondence: rawson.douglas@gmail.com

Abstract

Large Language Models often fail not by lacking knowledge, but by incorrectly applying it. We investigate a deep form of this failure, **"algorithmic misapplication,"** a procedural hallucination where a model makes a conceptual category error, applying a powerful but incorrect algorithm to a given problem. We show that a model can be taught to diagnose and correct these errors in itself. Using **Linguistic Reinforcement Learning (LRL)**, a process of iterative self-reflection, a highly capable model (gemini-2.5-flash) initially attempts to solve the classic Activity Selection Problem by incorrectly applying a complex "graph theory and dynamic programming" approach. Confronted with failure, the model's own journal logs reveal a stunning act of self-correction: it formally identifies its error, names the problem correctly as "Activity Selection Problem," and specifies the optimal "greedy algorithm" as the solution. This conceptual leap results in a corrected, formally optimal final strategy. More profoundly, the model generalizes this learning into a new meta-strategy: **to first diagnose the specific problem pattern before selecting a solution.** This demonstrates a path toward creating AI systems that don't just follow procedures, but can learn to critique and improve their own problem-solving frameworks, a foundational step toward more robust and reliable AI reasoning.

Keywords: Algorithmic Self-Correction, AI Safety, Interpretability, Linguistic Reinforcement Learning, Meta-Cognition, Procedural Hallucination, Reasoning

1. Introduction

The quest for artificial general intelligence is intertwined with the challenge of creating robust AI reasoners. While Large Language Models (LLMs) have demonstrated remarkable capabilities, their reasoning is often brittle. A common failure mode is *factual hallucination*, which can be addressed by external verifiers. However, a more fundamental challenge remains: what if the model's internal *reasoning process* is flawed?

This paper investigates this challenge through the lens of a specific, high-level failure we term **algorithmic misapplication**. This is a procedural hallucination where a model makes a category

error, correctly recalling a powerful algorithm from its training but applying it to a problem of the wrong type. For instance, using a shortest-path graph algorithm for a problem that is better suited for a simple greedy approach. This is not a failure of knowledge, but a failure of *wisdom*—the ability to select the right tool for the job.

Such errors are particularly dangerous as they represent a confident misapplication of sophisticated logic, making them difficult to detect and debug. We propose that the most effective remedy is to enable the model to debug itself.

We use **Linguistic Reinforcement Learning (LRL)**, a GPU-free method where a model iteratively solves problems, reflects on its performance in a journal, and distills its findings into an improved textual strategy. In this work, we show that this loop enables a powerful model (gemini-2.5-flash) to perform a remarkable act of academic self-correction.

The model's initial, flawed attempts and its subsequent, formally precise self-diagnosis are captured verbatim in its journal logs. This provides an unprecedented, interpretable trace of a model learning to be a better computer scientist. It learns not just the answer, but the foundational meta-skill of correctly diagnosing a problem before attempting to solve it. This work presents a tangible path toward AI systems that can reflect upon, question, and improve their own thinking processes.

---

2. The LRL Mechanism for Self-Correction

Our methodology relies on the LRL framework detailed in our prior work (Rawson, 2025). The core of the method is a loop that externalizes the model's reasoning process into a human-readable format, making it an object for its own critique.

1. **The Initial Flawed Procedure**: The model begins with a plausible but incorrect procedure, a form of procedural hallucination.
2. **Confrontation with Failure**: The model executes its flawed procedure and is shown objective evidence of its failure (incorrect answers).
3. **Reflective Diagnosis**: The journaling step prompts the model to analyze *why* it failed. This crucial step moves the model from merely observing failure to diagnosing its cause.
4. **Strategic Refinement**: The distillation step allows the model to synthesize a new, corrected procedure based on its diagnosis, explicitly rejecting its previous flawed approach.

This process enables the model to move beyond simple trial-and-error and perform genuine, introspective debugging of its own logic.

---

3. A Case Study in Algorithmic Self-Correction

We present a single, powerful case study on the **Activity Selection Problem**, a classic computer science task requiring a greedy algorithm for its optimal solution.

3.1 Experimental Setup

- **Model**: gemini-2.5-flash

- **Task**: Given a set of time intervals (meetings), find the maximum number of non-overlapping intervals that can be selected.
- **Protocol**: An LRL training process was initiated, where the model was prompted to solve problems and reflect on its failures to refine its strategy.

3.2 Main Finding: A Direct Observation of Self-Correction

The core finding of this work is the direct, qualitative evidence of self-correction captured in the model's own words. The journal logs provide a clear, step-by-step record of the model's conceptual breakthrough. While a parsing bug in the experimental harness prevented the collection of clean quantitative baseline data, the richness and clarity of the qualitative data stand as a powerful result on their own.

Phase 1: The Initial Procedural Hallucination

The model's first journal entry reveals a classic category error. It misidentifies the problem and applies a powerful but incorrect algorithmic paradigm.

> **Journal Entry 1 (Reflection on Failure):**
> *"My strategy to model the meetings as a graph and apply dynamic programming for an 'optimal path' was an over-engineered approach. The core flaw was a failure to recognize this problem as a direct instance of the Activity Selection Problem... I defaulted to a more general, complex paradigm (graph theory combined with dynamic programming) without first thoroughly analyzing the problem's inherent characteristics."*

This is a perfect example of algorithmic misapplication. The model is confidently deploying a flawed, overly complex procedure.

Phase 2: The Formal Self-Correction

After being confronted with its failure, the model performs a stunningly precise self-diagnosis. It does not just find a "simpler" way; it finds the *formally correct* way, using precise academic terminology.

> **Journal Entry 2 (Reflection on Flawed Procedure):**
> *"The flaw in my reasoning was in immediately jumping to a complex, general-purpose framework... without first analyzing the fundamental structure of the problem. ... I failed to recognize the problem's direct mapping to a known **greedy algorithm**."*

This reflection demonstrates a deep conceptual insight. The model has correctly identified its own cognitive error.

Phase 3: The Emergent Meta-Strategy

The final learned strategy is the culmination of this process. The model doesn't just codify the correct algorithm for scheduling; it generalizes its learning into a new, foundational meta-strategy for all future problem-solving.

> **Final Learned Strategy:**
> *"**First, meticulously identify the specific problem pattern** (e.g., interval scheduling,*

> *shortest path, knapsack)* ***rather than defaulting to complex, abstract frameworks. Then, select the most direct and efficient algorithm tailored to that pattern.*** *For problems requiring the selection of a maximum set of non-overlapping intervals (Activity Selection), apply a greedy strategy: sort intervals by their finish times and iteratively select the earliest finishing, non-overlapping interval."*

The model has learned the scientific principle of **diagnosis before treatment**. This is a fundamental leap in reasoning capability.

---

4. Discussion

This experiment reveals a potential pathway for developing more robust AI reasoners.

From Knowledge to Wisdom: Current LLMs are vast repositories of knowledge. The challenge is that they often lack the wisdom to know when and how to apply that knowledge. Our findings suggest that reflective practice, enabled by LRL, can help bridge this gap. The model learned to question its initial, powerful impulses and favor a more deliberate, diagnostic approach. We term this "algorithmic humility."

A New Frontier for AI Safety: The ability of a model to detect and correct its own procedural hallucinations is a critical safety feature. A system that can recognize its own flawed reasoning is inherently more reliable and less prone to the confident, catastrophic errors that currently plague LLMs.

Interpretability as a Learning Tool: This work underscores the power of interpretability. By externalizing its "thoughts" into a textual strategy, the model was able to critique and improve them. This suggests that making AI more transparent is not just for human benefit; it is a prerequisite for the AI to understand and improve itself.

---

5. Conclusion

We have provided a clear, documented instance of a language model performing formal algorithmic self-correction. We defined **algorithmic misapplication** as a critical failure mode and showed that **Linguistic Reinforcement Learning** provides a mechanism for the model to diagnose this error in itself.

The model's journey—from a flawed application of complex graph theory to the correct identification of a simple greedy algorithm, and finally to the generalization of this learning into a new meta-strategy—is a landmark in interpretable AI research. It demonstrates that we can create systems that do more than just execute procedures; we can create systems that learn to *think about how they think*. The path to more reliable AI may be paved not with more parameters, but with more self-reflection.

References

Rawson, D. (2025). Linguistic Reinforcement Learning: A Model's Journey from Flawed Complexity to Simple Understanding. https://github.com/DRawson5570/linguistic-rl-scheduling/blob/main/LRL_PAPER.md