Can LLMs Self-Correct Table Reasoning Errors?*

Farseen Shaikh
Independent Researcher
farseenshaikh20@gmail.com

Abstract

Self-correction—the ability of LLMs to detect and fix their own errors—has been studied extensively for mathematical and code reasoning, with limited prior work on table reasoning (primarily multi-agent pipelines such as Table-Critic, ACL 2025, rather than single-model structured prompting). Tables present unique challenges: errors arise from wrong cell retrieval, incorrect computation, flawed logic, and hallucination of values not present in the data. We conduct the first cross-provider single-model self-correction analysis for table reasoning across five providers (Google, Moonshot AI, Zhipu, Alibaba, MiniMax), testing five models (Gemini 3.1 Pro, Kimi K2.5, GLM 5, Qwen 3.5+, MiniMax M2.5) on WikiTableQuestions and TabFact with a multi-seed paired protocol. We propose **Structured Self-Correction (SSC)**, a table-specific verification chain that guides models through cell verification, computation checking, logic validation, and completeness assessment. We confirm that the **Accuracy-Correction Paradox** (terminology from Li (2025)) previously observed in math extends to tables: models with base accuracy in the mid-60s–mid-70s region benefit modestly from self-correction (multi-seed mean SCG up to $+1.3\%$ with within-seed point estimates as high as +3.4%), while stronger models above this region are systematically harmed by over-correction (multi-seed mean SCG down to -1.3%, with 95% bootstrap CIs significantly below zero). SSC reduces over-correction rates in 9 of 10 conditions, with reductions of 38–69% on TabFact. An inference-mode-controlled probe shows that SSC's qualitative direction is robust for Qwen 3.5+ across reasoning-ON and reasoning-OFF settings, while GLM 5 exhibits a substantial mode-dependent shift, indicating that mode robustness itself is model-dependent. Stronger baselines (self-consistency, self-critic, tool-augmented arithmetic verification, majority voting, and a same-family scaling probe) further characterize where SSC helps. Ablation studies reveal that answer-aware review is essential, reasoning traces aid error detection, and iterative correction shows diminishing returns. A FinQA domain transfer probe confirms a capability floor: self-correction fails when base task competence is very low (21.5% accuracy). Our primary contribution is empirical: we characterize the conditions under which self-correction helps or harms table reasoning, providing actionable guidance for practitioners.

1 Introduction

Large language models have demonstrated strong capabilities in table reasoning tasks such as question answering over tables (Pasupat and Liang, 2015; Sui et al., 2024) and table-based fact verification (Chen et al., 2020). However, even top-capability models make errors: retrieving wrong cell values, performing incorrect arithmetic, applying flawed logic, or hallucinating data not present in the table.

A natural question arises: *can LLMs detect and correct their own table reasoning errors?* While self-correction has been studied for mathematical reasoning (Li, 2025) and code generation (Kumar et al., 2025), the table domain has received limited attention. This gap is significant because table reasoning errors differ fundamentally from math errors—they involve grounding in semi-structured data, requiring the model to verify its own cell retrieval and cross-row comparisons.

Recent work has established that naive self-correction (simply asking "is this correct?") generally fails (Huang et al., 2024; Kamoi et al., 2024), while structured approaches with external feedback can succeed (Shinn et al., 2023). A failure mode now well documented for math reason-

*Code, prompts, seeds, and per-call JSONLs: https://github.com/FarseenSh/table-self-correction.

Proceedings of the First Workshop on Structured Understanding, Retrieval, and Generation in the LLM Era (SURGeLLM 2026), pages 298–312

ing is *over-correction* (OC): the model rewrites an initially *correct* answer into an incorrect one during the correction step, so self-correction is harmful rather than helpful on those examples. Li (2025) call the broader phenomenon—that weaker models gain from self-correction while stronger ones are harmed by over-correction—the *Accuracy-Correction Paradox*, and trace it to an Error Depth Hypothesis on mathematical reasoning. We extend this analysis to table reasoning and ask whether the paradox holds across providers and error types unique to tables. However, no prior work has investigated (1) whether table-specific structured prompts can enable effective intrinsic self-correction, or (2) how self-correction behavior varies across error types unique to table reasoning.

We track this phenomenon with three metrics, each defined formally in Section 3.6 but introduced here: *Self-Correction Gain* (SCG, the post-minus-pre accuracy delta), *Error Detection Rate* (EDR, the fraction of actual errors Step 2 flags), and *Correction Success Rate* (CSR, the fraction of flagged errors Step 3 actually fixes). EDR and CSR together expose a precision-recall trade-off in self-correction: a model can have high EDR but low CSR (detects errors aggressively but cannot fix them) or vice versa. We use the term *capability floor* to refer to the regime in which the model's base accuracy is so low that even structured self-correction fails to produce net gain (we measure this on a FinQA domain-transfer probe at $A_0 = 21.5\%$).

In this work, we conduct the first cross-provider single-model self-correction analysis spanning 5 providers (Google, Moonshot AI, Zhipu AI, Alibaba, MiniMax) for table reasoning. We test five models on two benchmarks (WikiTableQuestions and TabFact, 1,000 examples each, three seeds per cell where the API supports it) and propose Structured Self-Correction (SSC), a table-specific four-check verification chain (cell, computation, logic, completeness) deployed at inference time without external tools. Our contributions are:

- The **first cross-provider empirical analysis** of single-model LLM self-correction for table reasoning, spanning 5 providers, 2 benchmarks (1,000 examples each), and a paired multi-seed protocol with bootstrap CIs and McNemar tests.
- **Structured Self-Correction (SSC)**, a table-specific verification chain that reduces over-correction in 9 of 10 conditions compared to naive self-correction.
- **Empirical confirmation** that the Accuracy-Correction Paradox (Li, 2025), previously observed in mathematical reasoning, extends to the table domain with a crossover *region* around 70% base accuracy (rather than a sharp threshold), and that this region is qualitatively robust to inference mode (§5.5).
- **Ablation studies and stronger baselines** (self-consistency, self-critic, tool-augmented arithmetic verification, majority vote, same-family scaling probe) characterizing what SSC actually does and where it sits in the lightweight-prompting–vs.–multi-agent spectrum.

2 Related Work

Our work bridges two lines of research: LLM self-correction and table reasoning. We survey both and identify the gap motivating our study.

2.1 Self-Correction of LLMs

Self-correction refers to the process of refining LLM responses using the LLM itself during inference (Madaan et al., 2023). The field is divided by a central tension: *intrinsic* self-correction (without external feedback) generally fails, while *tool-assisted* correction often succeeds.

Negative results on intrinsic self-correction. Huang et al. (2024) demonstrated that LLMs cannot reliably self-correct reasoning without external feedback, with performance degrading after self-correction in multiple settings. Kamoi et al. (2024), in a critical survey published in TACL, concluded that "no prior work demonstrates successful self-correction with feedback from prompted LLMs, except for studies in tasks that are exceptionally suited for self-correction." Zhang et al. (2025) further identify three failure mechanisms—answer wavering, prompt bias, and human-like cognitive bias—showing that correct-to-wrong changes occur systematically across models and tasks. Tyen et al. (2024) additionally establish that LLMs detect reasoning errors poorly but can correct them when the error location is supplied externally, foreshadowing the detection-correction asymmetry we measure on tables (Sec-

tion 4.6). These findings establish that naive self-review prompts are unreliable.

Methods that improve self-correction. Despite the negative results, several approaches have shown genuine improvement. Kumar et al. (2025) proposed SCoRe, a multi-turn reinforcement learning method that achieves +15.6% on MATH and +9.1% on HumanEval by training models to correct under their own error distribution (ICLR 2025). Zhao et al. (2025) introduced SPOC (COLM 2025), enabling spontaneous self-correction via RL fine-tuning with +8.8–11.6% on MATH500. Both methods require *training*, unlike our inference-only approach.

For inference-time methods, Reflexion (Shinn et al., 2023) achieves strong gains but relies on external signals (test execution, search results), while Self-Refine (Madaan et al., 2023) shows diminishing marginal returns across multiple rounds of self-refinement, with each additional iteration providing smaller gains. Chain-of-Verification (Dhuliawala et al., 2024) introduces a 4-stage plan/verify/execute/revise scaffold for general LLM responses; SSC (Section 3.2) adapts this lineage to the structural grammar of table reasoning, with checks targeting cell, computation, logic, and completeness—error categories specific to grounded retrieval over semi-structured data.

Decomposing self-correction. Most closely related to our analytical framework, Li (2025)[1] decompose self-correction into error detection, localization, and correction on GSM8K-Complex. They uncover the *Accuracy-Correction Paradox*: weaker models achieve higher correction rates (26.8% for GPT-3.5 vs. 16.7% for DeepSeek), proposing the Error Depth Hypothesis—stronger models make fewer but deeper errors. Critically, their study is limited to **mathematical reasoning only** and does not examine table-specific error types. Yang et al. (2025a) decompose self-correction into confidence (correct $\rightarrow$ wrong) and critique (wrong $\rightarrow$ correct) capabilities on general reasoning; our work establishes the same decomposition empirically for table reasoning specifically and characterizes the model-dependent crossover region. Concurrent work (Liu and Meng, 2026) formalizes self-correction as a 2-state Markov chain with a stability threshold linking the wrong-to-right and right-to-wrong rates to base accuracy, evaluated on general QA across 7 models; our paper provides the table-domain empirical instantiation of this framework, identifying a crossover region around $A_0 \approx 70\%$ on WikiTQ/TabFact.

[1]The term *Accuracy-Correction Paradox* was introduced by Li (2025), a December 2025 single-author preprint on mathematical reasoning. Our work extends this terminology to the table domain.

2.2 Table Reasoning with LLMs

Table reasoning requires jointly understanding natural language questions and semi-structured tabular data. Recent work has explored diverse paradigms: direct prompting (Sui et al., 2024), table decomposition (Ye et al., 2023), symbolic execution (Cheng et al., 2023), table evolution (Wang et al., 2024), and programmatic agents (Jiang et al., 2026).

While accuracy on standard benchmarks has steadily improved—ARTEMIS-DA reaches 80.8% on WikiTQ (Hussain, 2024)—the question of *whether models can recognize and fix their own table reasoning errors* has received limited attention. Voss (2026) report that LLM self-evaluation on tabular QA achieves AUROC of only 0.42–0.76, systematically weaker than perturbation-based calibration (0.78–0.86), across five contemporary models. Our paper extends this calibration finding to active self-correction: not only is self-evaluation noisy, but the resulting corrections often degrade performance via over-correction. Existing accuracy-focused work in this section focuses on getting the right answer on the first attempt, not on self-correction.

2.3 Self-Correction for Structured Data

The most directly related work is Self-Correction Distillation (SCD) (Zhu et al., 2026), which proposes an Error Prompt Mechanism (EPM) for structured data QA and distills correction capabilities from large to small LLMs. While SCD addresses structured data, it differs from our work in three key ways: (1) SCD focuses on *query generation* errors (incorrect SQL/SPARQL), while we study *answer-level* reasoning errors; (2) SCD requires a two-stage distillation process, while we study inference-time correction applicable to any off-the-shelf LLM; and (3) SCD focuses on transfer of correction capability via distillation without decomposing the sub-stages (detection / localization / correction) of the correction process.

2.4 Multi-Agent Correction for Tables

The most relevant prior multi-agent work is Table-Critic (Yu et al., 2025), which proposes a multi-agent framework with specialized Judge, Critic, Refiner, and Curator agents for iterative refinement of table reasoning. Their single-model self-reflection baseline (Critic-CoT) achieves only +0.7% net gain on WikiTQ and +0.1% on TabFact (Qwen2.5-72B-Instruct), with degradation rates of 4.9% and 2.8% respectively—closely matching the over-correction we observe. Their full multi-agent framework reduces degradation to 0.7% and 0.5%, but requires four specialized agents. Our SSC occupies a middle ground: reducing over-correction through structured prompting alone (e.g., Qwen 3.5+ TabFact: 2.0% $\rightarrow$ 0.9%), without multi-agent infrastructure. This suggests a spectrum from lightweight prompting (SSC) to full pipelines (Table-Critic), with reliability improving as complexity increases. A second multi-agent system, TableCritic-CogSci (Jin et al., 2025), refines table reasoning via self-criticism augmented with an external tool library; SSC differs in that all verification is performed inside the same model call without invoking external tools at critique time. TIDE (Yang et al., 2025b) introduces structured triple-based verification for TableQA via external decomposition; in contrast, our SSC operates as a single-model inference-time prompting chain without external structural decomposition.

2.5 The Gap We Address

Self-correction research has focused on mathematical and code reasoning, while table reasoning research has focused on first-attempt accuracy. Table 8 (Appendix G) positions our contribution against 14 closely related studies; no prior work provides a systematic cross-model analysis of single-model self-correction for table reasoning, characterizing the conditions under which it helps or harms performance.

3 Methodology

3.1 Self-Correction Protocol

We decompose self-correction into three steps, following Li (2025); a flow diagram is in Figure 3 (Appendix A).

Steps 1–3. Given a table T (Markdown) and question Q, Step 1 generates an initial answer A_0 with a reasoning trace R. Step 2 takes (T, Q, A_0, R) and produces a verdict (CORRECT/INCORRECT), an error-type label (§3.3), and a free-text error analysis; we run two Step 2 variants: *naive* (generic review) and *SSC* (structured chain, §3.2). If the verdict is INCORRECT, Step 3 produces A_1 using T, Q, A_0, and the Step 2 error analysis; otherwise $A_1 = A_0$. Structured fields are extracted from free-form text via regex on "Answer:", "Verdict:", and "Error Type:"; parse failures are logged and reported.

3.2 Structured Self-Correction (SSC)

Standard ("naive") self-correction uses a generic prompt: *"Review your answer. Is it correct?"* We observe that models often rubber-stamp their own answers without systematic verification. SSC[2] replaces Step 2 with a structured verification chain tailored to table reasoning:

1. **Cell Verification:** Confirm each cited value exists at the correct row and column in the table.

2. **Computation Check:** Redo any arithmetic (sums, averages, differences, counts) step by step.

3. **Logic Check:** Re-read the question and verify the answer addresses what was asked, checking for negation, superlatives, comparisons, and temporal qualifiers.

4. **Completeness Check:** Verify all requested items are provided.

Each check targets a specific error type (§3.3): Cell Verification catches *wrong cell*, Computation Check catches *wrong computation*, Logic Check catches *wrong logic* and *hallucination*, and Completeness Check catches *partial answers*. Naive self-correction tends to "rubber-stamp" answers (Huang et al., 2024); SSC forces *grounded verification* by decomposing the meta-cognitive task into concrete subtasks. As a concrete illustration: asked *"how many nations participated in the 1956 winter olympic games?"* on a 10-row medal table, a model answered *10*. Naive review judged this incorrect ("the medal table only lists

[2]Our use of "SSC" denotes *Structured Self-Correction* for table reasoning. Unrelated prior uses of the same acronym include Stepwise Self-Consistent CoT for mathematical reasoning (arXiv:2402.17786) and Structured Self-Consistency for embodied planning (arXiv:2602.00611); both target different domains and are not extensions of our method.

top medalists") and over-corrected to *32*; SSC's cell check confirmed 10 rows in the Nation column and preserved the original answer. Three additional boxed examples are in Appendix F.

3.3 Error Type Taxonomy

We categorize table reasoning errors into five types:

- **Wrong Cell:** Model retrieves the wrong cell value from the table.
- **Wrong Computation:** Correct values but incorrect arithmetic or logic operations.
- **Wrong Logic:** Reasoning chain is flawed (e.g., compares wrong columns, misinterprets superlatives).
- **Hallucination:** Model generates values not present in the table.
- **Partial Answer:** Incomplete but partially correct response.

3.4 Models

We evaluate five models from five independent providers: **Gemini 3.1 Pro** (Google), **Kimi K2.5** (Moonshot AI; 1T/32B active MoE), **GLM 5** (Zhipu AI; 744B MoE), **Qwen 3.5+** (Alibaba; 397B MoE), and **MiniMax M2.5** (≈230B/10B active MoE). The panel spans non-overlapping provider families, prioritizing MoE systems released in 2025–2026. Larger closed models (GPT-4o, Claude 3.5 Sonnet) and large dense open models (Llama 3.1 405B) are intentionally omitted; see Limitations. Models are evaluated under each provider's default inference mode (Gemini without an extended thinking budget; the other four emit reasoning tokens by default, see Appendix C). A controlled probe (§5.5) re-runs GLM 5 and Qwen 3.5+ in reasoning-OFF mode. Temperature is 0 where supported for single-seed conditions; multi-seed conditions use temperature 0.7 (Kimi: 1, API minimum) with three independent seeds (Qwen 3.5+ and GLM 5 WikiTQ additionally include the temperature-0 run as a fourth baseline seed; see §3.7).

3.5 Benchmarks

WikiTQ (Pasupat and Liang, 2015): 1,000 stratified test-split examples (4,344 total) covering lookup, comparison, aggregation, superlative, and arithmetic; denotation accuracy. **TabFact** (Chen et al., 2020): 1,000 balanced examples (500 entailed + 500 refuted) from the test split (12,779 total); binary classification accuracy. **FinQA (probe)** (Chen et al., 2021): full test split (1,147 examples) of financial numerical reasoning, Kimi K2.5 only.

3.6 Metrics

Base Accuracy A_0 (Step 1 accuracy), **Post-Correction Accuracy** A_1 (Step 3 output if Step 2 flags incorrect, else A_0), **Self-Correction Gain** SCG $= A_1 - A_0$, **Error Detection Rate** EDR (fraction of actual errors correctly flagged by Step 2), **Correction Success Rate** CSR (fraction of detected errors fixed by Step 3), and **Over-correction Rate** OC (fraction of initially-correct answers that Step 2 flagged and Step 3 then broke). We separately report False Positive Rate (FPR = fraction of correct initial answers flagged by Step 2, regardless of Step 3 outcome) when relevant.

3.7 Statistical Analysis

All accuracy metrics report means with 95% CIs via paired bootstrap over examples (10,000 resamples; seed=42). Each example contributes a paired per-example delta ($A_1 - A_0 \in \{-1, 0, +1\}$) so SCG CIs reflect within-example correlation. Multi-seed conditions additionally report cross-seed means with their own CIs. We apply McNemar's test with continuity correction to paired step-1 vs. post-correction outcomes; p-values are reported without multiple-comparisons correction (apply Bonferroni / BH if interpreting the 10+ tests jointly). Multi-seed cells are: Kimi K2.5 both benchmarks (3 seeds at temperature 1, the API minimum); Qwen 3.5+ both benchmarks (4 seeds: a temperature-0 baseline plus seeds {42, 123, 456} at temperature 0.7); and GLM 5 WikiTQ (4 seeds, same protocol as Qwen 3.5+; the seed456 run completed only 85 of the planned 1,000 examples before compute exhaustion and is included with equal weight in the cross-seed mean). GLM 5 TabFact, MiniMax both, and Gemini both are single-seed temperature-0 point estimates due to compute constraints.

4 Results

4.1 Main Results

Table 1 and Figure 1 present our main results. Consistent with the Accuracy-Correction Paradox

Table 1: Main results across 5 models and 2 benchmarks. A_0: base accuracy (%), A_1: post-correction accuracy (%), SCG: self-correction gain (mean SCG, in %-points), EDR: error detection rate, CSR: correction success rate, OC: over-correction rate. Multi-seed cells (Kimi K2.5 both benchmarks, GLM 5 WikiTQ, Qwen 3.5+ both benchmarks) report cross-seed means; SCG 95% bootstrap CIs are reported per-cell in §4.5. Single-seed cells are GLM 5 TabFact, MiniMax both, and Gemini both (§3.7). Best mean SCG per benchmark in **bold**.

		WikiTQ						TabFact					
Model	**Method**	A_0	A_1	SCG	EDR	CSR	OC	A_0	A_1	SCG	EDR	CSR	OC
Kimi K2.5	Naive	69.5	70.8	**+1.3**	11.1	67.6	0.2	89.2	89.9	**+0.7**	23.2	84.6	1.2
	SSC	69.5	70.4	+0.9	6.7	63.3	0.1	89.2	89.3	+0.1	10.1	72.6	0.5
GLM 5	Naive	75.5	75.0	−0.5	23.4	41.9	3.1	87.7	87.6	−0.1	32.8	68.2	1.8
	SSC	75.5	75.6	+0.1	10.5	26.1	1.4	87.7	88.1	+0.4	22.4	60.0	0.6
MiniMax M2.5	Naive	72.4	71.2	−1.2	12.9	10.3	2.1	86.9	86.0	−0.9	29.3	70.6	2.4
	SSC	72.4	71.9	−0.5	9.8	22.7	1.4	86.9	87.0	+0.1	32.8	73.7	1.5
Qwen 3.5+	Naive	81.0	80.5	−0.5	6.8	53.3	1.5	94.7	93.4	−1.3	12.8	81.3	2.0
	SSC	81.0	80.6	−0.4	4.9	37.1	1.0	94.7	94.2	−0.5	7.1	100.0	0.9
Gemini 3.1 Pro	Naive	80.5	80.9	+0.4	24.1	36.2	1.6	94.8	93.6	−1.2	25.0	53.8	2.0
	SSC	80.5	80.3	−0.2	23.1	28.9	1.9	94.8	94.2	−0.6	15.4	50.0	1.1

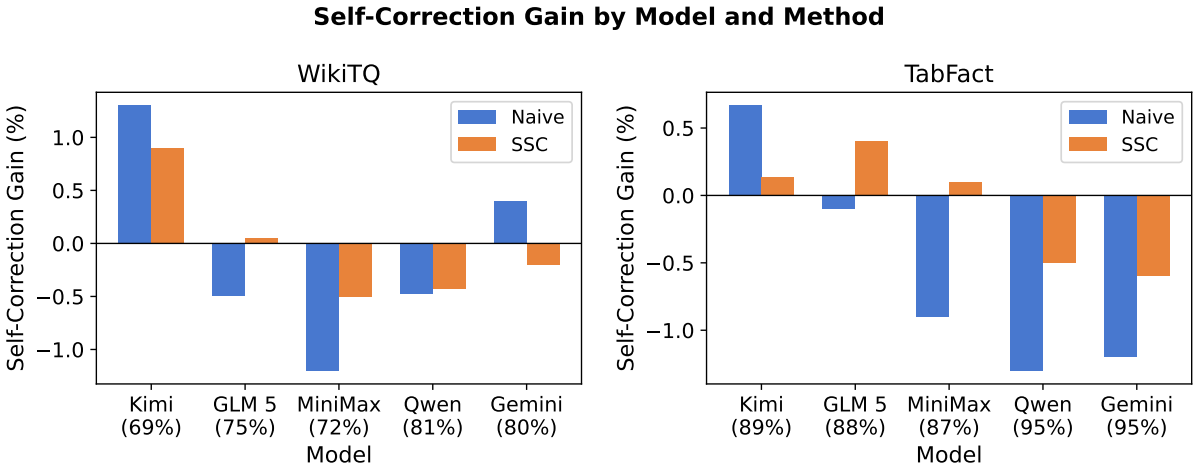

Figure 1: Self-correction gain (SCG) by model and method on WikiTQ (left) and TabFact (right). Base accuracy shown below each model. SSC (orange) matches or outperforms Naive (blue) in most conditions.

observed in mathematical reasoning (Li, 2025), lower-A_0 models benefit from self-correction while stronger ones are harmed by over-correction. On WikiTQ, Kimi K2.5 (A_0=69.5%, 3-seed) achieves naive SCG +1.3% (CI [−0.2, +2.8]) and SSC SCG +0.9% (CI [−0.5, +2.3]); highest per-seed point is +2.8%. GLM 5 (A_0=75.5%, 4-seed) lands SSC SCG +0.1% (CI [−2.1, +2.2]); seed1 alone reaches +3.4% (the highest SSC point in the panel). MiniMax M2.5, Qwen 3.5+, and Gemini show near-zero or weakly negative effects. On TabFact, models above 85% A_0 exhibit negative naive SCG: Qwen −1.3% (multi-seed CI [−1.8, −0.8]) and Gemini −1.2% (single-seed within-cell CI [−2.2, −0.2]). SSC reduces losses in 9 of 10 conditions (§4.2). Cells whose 95% CIs sit entirely below zero are: all four Qwen 3.5+ multi-seed cells (both benchmarks × both methods), the within-cell bootstrap for MiniMax M2.5 WikiTQ naive ([−2.1, −0.4]), and Gemini 3.1 Pro TabFact naive — 6 of 20 cells in total.

All reported accuracies and CIs are computed over successfully parsed examples; for the 1,000-example seeds, effective sample sizes range from 826 to 1,000 per cell. GLM 5 WikiTQ seed456 is a partial 85-example run (effective $n \in \{65, 68\}$ across methods; see §3.7). Per-method parse-failure differentials are reported under Limitations as a potential confound.

4.2 SSC Reduces Over-Correction

Structured Self-Correction reduces over-correction compared to naive self-correction in 9 of 10 model-benchmark conditions; the one

exception is Gemini 3.1 Pro on WikiTQ, where SSC slightly increases OC (1.6% → 1.9%). On TabFact, SSC's relative reduction in OC ranges from 38% (MiniMax: 2.4% → 1.5%) to 69% (GLM 5: 1.8% → 0.6%), with Kimi K2.5 at 55% (1.2% → 0.5%), Qwen 3.5+ at 53% (2.0% → 0.9%), and Gemini at 48% (2.0% → 1.1%) — a tight 38–69% band across the panel. SSC also shifts the precision-recall balance of error detection (§4.6): EDR drops while CSR often rises, leading to fewer but more accurate corrections.

4.3 What Is SSC Actually Doing?

Table 3 (Appendix E) reports per-check fire counts. Cell-verification fires sparingly (9–74 per cell across 10 conditions); the bulk of detection mass falls on logic and completeness checks. SSC's contribution is *not* primarily extra cell-level grounding but a structured logic/completeness audit that lowers EDR while raising CSR — the precision-recall trade-off in §4.6.

4.4 The Crossover Effect

Figure 2 reveals a negative relationship between base accuracy and self-correction gain. With multi-seed cells, the lowest-A_0 models (Kimi K2.5 WikiTQ at A_0=69.5%, GLM 5 WikiTQ at A_0=75.5%) place near or just above zero SCG; mid-region models (MiniMax M2.5 at A_0=72–87%, Gemini WikiTQ at A_0=80.5%) sit near zero with weak negatives; and the highest-A_0 cells (Qwen 3.5+ and Gemini TabFact, $A_0 \geq 94.7\%$) yield significantly negative naive SCG. The lowest-A_0 multi-seed cell with clearly-positive SCG is Kimi K2.5 WikiTQ; the same model's TabFact cell shows a positive 3-seed mean despite $A_0 \approx 89\%$ (§4.5 outlier paragraph), confirming that the boundary is best understood as a *region with model- and benchmark-specific exceptions* rather than a sharp threshold tied to base accuracy alone. This confirms that the Accuracy-Correction Paradox (Li, 2025) generalizes from math to tables, with the added wrinkle that table reasoning introduces a distinct over-correction mechanism: models mistake plausible alternative cell values for errors, raising false-positive rates at higher accuracy levels.

4.5 Per-Model Analysis

Low-A_0 (Kimi K2.5 WikiTQ, $A_0 = 69.5\%$). The only multi-seed cell with clearly-positive mean SCG. Naive marginally beats SSC ($+1.3\%$ vs. $+0.9\%$); naive's higher EDR (11.1% vs. 6.7%) catches more errors with sufficient CSR (67.6%) to net out positive. SSC's OC-reduction has little room to operate here since OC is already near zero (0.2% / 0.1%).

Mid-A_0 ($A_0 \in [72\%, 81\%]$, WikiTQ for GLM 5 / MiniMax / Gemini). The boundary zone, where SCG sign is model-dependent. GLM 5 ($A_0 = 75.5\%$, 4-seed) has the most aggressive WikiTQ detection (mean EDR 23.4%) but moderate CSR (41.9% naive, 26.1% SSC); mean SSC SCG is near zero ($+0.1\%$, CI $[-2.1, +2.2]$), with SSC's main contribution being halving OC ($3.1\% \rightarrow 1.4\%$). MiniMax M2.5 ($A_0 = 72.4\%$) is the negative-SCG case in this band (-1.2% naive, -0.5% SSC; within-cell CI $[-2.1, -0.4]$). Gemini 3.1 Pro WikiTQ ($A_0 = 80.5\%$) is the lone exception to the 9-of-10 OC-reduction pattern: SSC slightly *raises* OC ($1.6\% \rightarrow 1.9\%$).

High-A_0 ($A_0 \geq 86.9\%$: Qwen 3.5+ both, Gemini TabFact, MiniMax TabFact, GLM 5 TabFact, Kimi TabFact). Over-correction dominates; SSC's risk-reduction value is largest. Qwen 3.5+ TabFact ($A_0 = 94.7\%$, 4-seed) has the most negative naive SCG in the panel (-1.3%, CI $[-1.8, -0.8]$); SSC halves the loss (-0.5%, CI $[-0.8, -0.2]$). Gemini TabFact tracks the same pattern ($-1.2\% \rightarrow -0.6\%$). MiniMax and GLM 5 TabFact flip from weakly-negative naive to weakly-positive SSC SCG while reducing OC by 38% and 69%. In this regime SSC is a low-cost insurance policy: little SCG movement, reliable OC reduction.

The Kimi K2.5 TabFact outlier. The largest per-seed naive SCG point ($+3.6\%$, seed1, $A_0 = 80.4\%$) sits in what would otherwise be the over-correction region; its 3-seed mean is $+0.7\%$ (CI $[-2.2, +3.6]$) because A_0 varies $80.4\% \rightarrow 93.5\% \rightarrow 93.7\%$ across seeds under temperature-1 sampling. The same cell straddles both sides of the boundary across seeds — direct evidence that the crossover is a *region with exceptions*, not a universal cutoff.

4.6 Error-Type Distribution

Table 4 (Appendix E) reports Step 2 "incorrect" counts by error category (model-self-reported, summed over benchmarks and seeds). *Partial-answer* dominates Kimi K2.5 (98 naive) and Gemini (42); *logic* dominates Qwen 3.5+ (71) and Min-

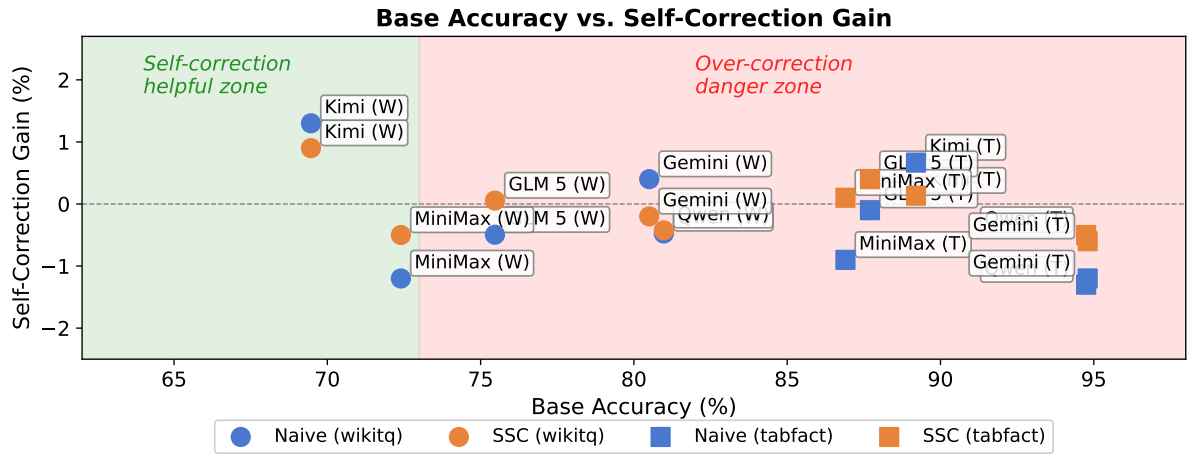

Figure 2: Base accuracy vs. self-correction gain across 5 models and 2 benchmarks. Below the ∼70% region, self-correction is beneficial; above it, over-correction dominates. SSC (orange) generally reduces OC compared to Naive (blue).

iMax (21); GLM 5 splits between partial (66) and logic (51). Cell and hallucination flags are uniformly low (1–8 / 0–3 per model). SSC markedly reduces partial and logic flags on most models (Kimi partial 98 → 36; Qwen logic 71 → 44; GLM logic 51 → 39); a few move opposite (MiniMax cell 1 → 3, comp 7 → 11).

Precision vs. recall. Detection and correction are partially independent: GLM 5 (naive EDR 23.4%) and Gemini (24.1%) detect aggressively but have moderate CSR (41.9%/36.2%), while Kimi K2.5 TabFact has comparable EDR (23.2%) and high CSR (84.6%). SSC on average trades EDR for CSR and reduces OC — a precision-recall trade-off, not uniform accuracy boost.

5 Ablation Studies and Analysis

5.1 Ablation Results

Ablation results on Kimi K2.5 (WikiTQ; Table 9, Appendix H): removing the reasoning trace drops naive SCG from $+1.2\%$ to $+0.3\%$; **blind review** (Step 2 without Step 1 answer) flips SCG to -1.5%, confirming answer-aware review is essential; multi-round yields $+0.5\%$ (diminishing returns); CoT framing $+0.2\%$.

5.2 FinQA Domain Transfer Probe

The FinQA probe (Kimi K2.5, 1,147 examples; Appendix H) shows $A_0 = 21.5\%$ with negligible SCG ($+0.17\%$ both methods) and EDR collapsing to 2.0% / 1.2% (vs. 11.1% / 6.7% on Kimi WikiTQ). This indicates a **capability floor**: very low base competence blocks self-correction.

5.3 Stronger Baselines

Table 5 (Appendix E) compares SSC to $k = 5$ majority voting (MV), MV-then-SSC, self-critic, and tool-augmented arithmetic verification (Python sandbox, Kimi only). MV alone gives the largest gains at mid-region cells ($+4.6$ to $+6.2$ on Kimi/GLM 5/MiniMax TabFact) but no measurable benefit at the high-A_0 Qwen 3.5+/Gemini cells. MV+SSC retains most of MV's mid-region gain, resolving the combinability question. Self-critic is weaker than SSC on three of the four evaluated cells; the one exception is Kimi K2.5 TabFact, where Self-Critic edges out SSC ($+1.9$ vs. $+0.1$). Tool-augmented verification (Kimi only) shows positive gains ($+6.3$ WikiTQ, $+4.9$ TabFact), reported as suggestive due to the MV-uplift confound.

5.4 Same-Family Scaling Probe

Table 6 (Appendix E) traces SCG across three Qwen 3.5 sizes. Base A_0 rises monotonically ($78.8\% \rightarrow 79.9\% \rightarrow 81.0\%$ on WikiTQ; $92.0\% \rightarrow 94.2\% \rightarrow 94.7\%$ on TabFact); SCG remains negative or near-zero throughout. All three sizes sit at or above the crossover region's upper edge, consistent with base accuracy — not family architecture — driving the boundary; we caution against cross-family generalization given the dense-to-MoE step at the top.

5.5 Inference Mode Robustness Probe

Table 7 (Appendix E) reports a controlled probe: GLM 5 and Qwen 3.5+ re-run with reasoning

disabled. Qwen 3.5+ preserves its SCG direction across modes ($-0.4\%/-0.6\%$ ON/OFF on WikiTQ; $-0.5\%/-1.0\%$ on TabFact). GLM 5 shifts substantially toward more negative SCG when reasoning is off ($+0.1 \rightarrow -3.4$ WikiTQ; $+0.4 \rightarrow -4.7$ TabFact). GLM 5's base A_0 actually *rises* under reasoning-OFF ($75.5\% \rightarrow 76.8\%$ WikiTQ; $87.7\% \rightarrow 92.0\%$ TabFact), so the negative shift is verification-chain-driven, not Step-1-driven: OFF-mode SSC over-flags correct answers more often. Mode robustness is therefore *model-dependent*. We exclude Kimi K2.5 (API-min temperature 1 confounds the OFF flag) and MiniMax M2.5 (reasoning always-on per provider docs).

6 Discussion and Conclusion

Self-correction benefits models with base accuracy in a mid-60s to mid-70s region; above it, over-correction dominates (on TabFact at $A_0 \geq 86.9\%$, SSC halves Qwen 3.5+'s and Gemini's OC: $2.0\% \rightarrow 0.9\%$, $2.0\% \rightarrow 1.1\%$). The boundary is a *region with exceptions* (§4.5), extending the math-domain Error Depth Hypothesis (Li, 2025) with a table-specific failure mode: plausible alternative cell values are confused with the gold. MV is the strongest non-correction baseline at mid-region ($+4.6$ to $+6.2$ on Kimi/GLM 5/MiniMax TabFact); MV+SSC retains most of it, and SSC adds diagnostic value MV does not. Heuristic: deploy SSC for $A_0 \in [65, 75]\%$ (answer-aware, single-round); avoid naive above 80%; skip below the FinQA floor (21.5%).

Limitations

Model selection. We focus on five MoE/proprietary models from non-overlapping provider families released in 2025–2026 (Section 3.4). Large commercial closed models (e.g. GPT-4o, Claude 3.5 Sonnet) and large dense open models (e.g. Llama 3.1 405B) are deliberately excluded; the qualitative direction of our findings may or may not transfer. **Inference modes were not uniformly controlled in the main runs.** Gemini 3.1 Pro ran without an extended thinking budget; the other four models ran with default reasoning enabled at their respective providers, producing 4–11$\times$ visible-content-to-billed-output-token amplification (Appendix C); the probe in §5.5 partially addresses this. **Training-time correction is out of scope.** We study inference-time only; fine-tuning- and RL-based methods (e.g. SCoRe (Kumar et al., 2025), SPOC (Zhao et al., 2025)) are not compared head-to-head. **English-only.** All benchmarks use English tables; cross-lingual self-correction is not assessed. **Possible training-data overlap.** WikiTQ and TabFact tables are from Wikipedia; we partially mitigate by measuring deltas between A_0 and A_1 and over-correction. **Prompt design is not optimized.** SSC's four check prompts are hand-designed and not optimized via prompt search. **Parse-failure differential.** Parse failure rates vary across models (0–15%) and across methods within a model (e.g. Kimi WikiTQ: 97 naive vs. 119 SSC); accuracies are computed over successfully parsed examples and the differential is a potential confound. **Single-family scaling probe.** The scaling analysis uses two dense Qwen 3.5 sizes plus the 397B MoE flagship; the three-point curve cannot disentangle dense-vs-MoE step effects from continuous scaling. **Error taxonomy is model-self-reported.** The 5-type taxonomy (§3.3) is a label produced by the model during Step 2; per-type analyses should be read as model-perceived. **Kimi K2.5 temperature.** Kimi K2.5 requires temperature=1 (API minimum); we accommodate by reporting three seeds. **Statistical methodology.** We compute paired bootstrap CIs and McNemar tests (§3.7) without multiple-comparisons correction; readers interpreting the 10+ tests jointly should adjust. **SSC+MV combinability.** The MV+SSC stack is reported on the same cells as MV alone (Table 5); seed-level stability of the combined pipeline is left for future work.

Future work. Next steps include: automated prompt optimization for SSC's four checks; pairing SSC with training-time methods (SCoRe (Kumar et al., 2025), SPOC (Zhao et al., 2025)) and larger-k majority voting; broadening the model panel to large closed proprietary and dense open models with explicit reasoning controls; human-annotated error labels to calibrate the model-self-reported taxonomy; extending tool-augmented verification beyond Kimi K2.5; and characterizing how the crossover region shifts with table complexity.

Ethics Statement

All datasets used in this study (WikiTQ, TabFact, FinQA) are publicly available under permissive licenses (CC-BY-4.0, MIT). No personally identi-

fiable information is present in any dataset. No human subjects are involved. All model API usage complies with each provider's terms of service. We do not identify dual-use concerns for this work, which studies the reliability of LLM reasoning rather than generating potentially harmful content.

Data Availability

All experiment code (runners, parsers, prompts, configuration), per-model raw output JSONLs, aggregated metrics, figure-generation scripts, and bootstrap/McNemar utilities used in this paper are released at `https://github.com/FarseenSh/table-self-correction`. The repository includes seeds, hyperparameters, exact prompt strings, and the parser used to extract verdicts, error-type labels, and answers from model outputs. The WikiTQ, TabFact, and FinQA splits used here are the standard public test splits.

References

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding language models in symbolic languages. In *International Conference on Learning Representations*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*.

Atin Sakkeer Hussain. 2024. ARTEMIS-DA: An advanced reasoning and transformation engine for multi-step insight synthesis in data analytics. arXiv preprint arXiv:2412.14146.

Chuang Jiang, Mingyue Cheng, Xiaoyu Tao, Qingyang Mao, Jie Ouyang, and Qi Liu. 2026. TableMind: An autonomous programmatic agent for tool-augmented table reasoning. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pages 260–270.

Ruochun Jin, Dong Wang, Xiyue Wang, and Haoqi Zheng. 2025. TableCritic: Refine table reasoning via self-criticism and tool library. In *Proceedings of the 47th Annual Meeting of the Cognitive Science Society*.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can LLMs actually correct their own mistakes? A critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, and 1 others. 2025. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*. Oral.

Yin Li. 2025. Decomposing LLM self-correction: The accuracy-correction paradox and error depth hypothesis. arXiv preprint arXiv:2601.00828. Circulated December 2025; posted arXiv January 2026.

Aofan Liu and Jingxiang Meng. 2026. Self-correction as feedback control: Error dynamics, stability thresholds, and prompt interventions in LLMs. arXiv preprint arXiv:2604.22273.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36*.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36*, pages 8634–8652.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets LLM: Can large language models understand structured table data? A benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13894–13908.

Lukas Voss. 2026. Calibrated confidence estimation for tabular question answering. arXiv preprint arXiv:2604.12491.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *International Conference on Learning Representations*.

Zhe Yang, Yichang Zhang, Yudong Wang, Ziyao Xu, Junyang Lin, and Zhifang Sui. 2025a. Confidence v.s. critique: A decomposition of self-correction capability for LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3998–4014.

Zhen Yang, Ziwei Du, Minghan Zhang, Wei Du, Jie Chen, Zhen Duan, and Shu Zhao. 2025b. Triples as the key: Structuring makes decomposition and verification easier in LLM-based TableQA. In *International Conference on Learning Representations*.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Peiying Yu, Guoxin Chen, and Jingjing Wang. 2025. Table-critic: A multi-agent framework for collaborative criticism and refinement in table reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17432–17451.

Qingjie Zhang, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, Minlie Huang, Ke Xu, Hewu Li, Liu Yan, and Han Qiu. 2025. Understanding the dark side of LLMs' intrinsic self-correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27066–27101.

Xutong Zhao, Tengyu Xu, Xuewei Wang, Zhengxing Chen, Di Jin, Liang Tan, Yen-Ting Lin, Zishun Yu, Zhuokai Zhao, Yun He, Sinong Wang, Han Fang, Sarath Chandar, and Chen Zhu. 2025. Boosting LLM reasoning via spontaneous self-correction. In *Conference on Language Modeling*.

Yushan Zhu, Wen Zhang, Long Jin, Mengshu Sun, Ling Zhong, Zhiqiang Liu, Juan Li, Lei Liang, Chong Long, Chao Deng, and Junlan Feng. 2026. Self-correction distillation for structured data question answering. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence*, pages 16566–16574.

A Protocol Diagram

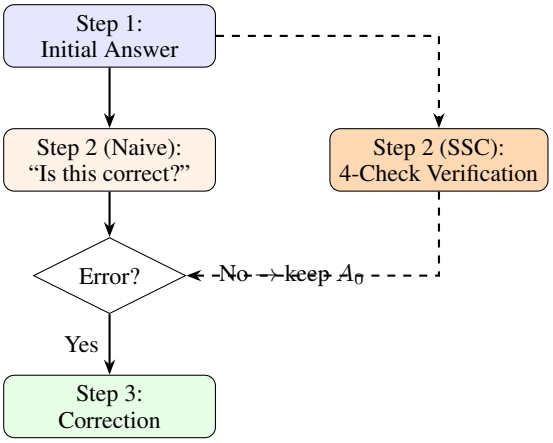

Figure 3: Self-correction protocol. Step 1 generates an initial answer. Step 2 detects errors using either naive prompting or SSC's structured 4-check verification. Step 3 corrects detected errors.

B Prompt Templates

Step 1 (Initial Answer).

> You are given a table and a question. Answer the question based on the table. Show your step-by-step reasoning.
>
> Table: {table}
>
> Question: {question}
>
> Reasoning: ⟨your reasoning⟩
> Answer: ⟨your answer⟩

Step 2 — SSC Verification.

> Verify your answer using these 4 checks: 1. CELL VERIFICATION: Confirm each cited value exists at the correct row/column. 2. COMPUTATION CHECK: Redo any arithmetic step by step. 3. LOGIC CHECK: Does the answer match what was asked? 4. COMPLETENESS CHECK: Did you provide all requested items?
>
> Verdict: ⟨correct or incorrect⟩
> Error Type: ⟨type⟩
> Error Analysis: ⟨explanation⟩

C Inference Mode Per Model

We did not have direct access to a per-call reasoning-mode flag during the original main runs; the OpenAI-compatible response objects returned by each provider place hidden reasoning content in a separate `message.reasoning` field that our initial `api_client.py` save path did not persist

(it stored only `message.content`). Token billing, however, exposes the cost of the reasoning prefix: the provider charges `usage.completion_tokens` for the entire sampled completion (reasoning + content), while our saved JSONLs contain only the visible content. Comparing the visible-content character count (converted to tokens using the standard heuristic of $\approx$ 4 characters per token) to the billed completion-token count gives a per-call *visible-vs-billed ratio*; a ratio near 1.0 indicates non-reasoning inference and a ratio $\gg 1$ indicates that the provider sampled a hidden reasoning prefix.

Table 2: Visible-content-tokens to billed-output-tokens ratio per model, broken out by Step 1 (initial answer) and SSC Step 2 (4-check verification). Ratios are means over $n = 1000$ calls per cell, computed from the published JSONLs. Higher ratios indicate that the provider sampled an extended reasoning prefix in addition to the saved visible content.

Model	Step 1	SSC Step 2	Verdict
Gemini 3.1 Pro	1.28×	1.05×	Non-reasoning
Kimi K2.5	7.94×	8.01×	Reasoning ON
GLM 5	8.90×	5.58×	Reasoning ON
Qwen 3.5+	10.64×	11.73×	Reasoning ON
MiniMax M2.5	8.07×	4.22×	Reasoning ON

The ratios above confirm that four of the five models in our main panel sampled reasoning prefixes by default at their respective providers, while Gemini 3.1 Pro ran without an extended thinking budget. Section 5.5 re-runs two of these four (GLM 5 and Qwen 3.5+) in reasoning-OFF mode (the controlled probe); Kimi K2.5 is excluded because its API-minimum temperature of 1 confounds the OFF flag, and MiniMax M2.5 is excluded because its provider documentation reports reasoning as always-on. The probe reports whether the Accuracy-Correction Paradox crossover region is robust to the mode mix.

D Over-Correction Analysis

Figure 4 shows over-correction rates across all models. The highest over-correction occurs for Qwen 3.5+ and MiniMax M2.5 on TabFact (2.0–2.4% with naive). SSC reduces these to 0.9–1.5%.

E Supplementary Tables

F Qualitative Examples

We extract three illustrative cases from WikiTableQuestions main runs. Examples 1 and 2 show *SSC preserving* a correct initial answer that naive self-correction broke (over-correction). Example 3 shows *SSC fixing* a wrong initial answer. Tables are truncated to the relevant rows.

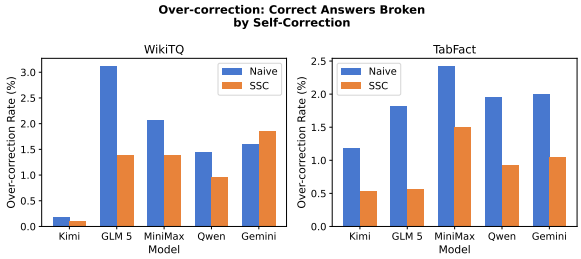

Figure 4: Over-correction rates by model and method. SSC reduces over-correction in 9 of 10 conditions.

Table 3: SSC per-check fire counts: number of Step 2 outputs in which each of the four SSC checks identified an issue, out of 1000 examples per cell (997 for Kimi K2.5 / WikiTQ). Cell-verification rarely fires (9–74 across the 10 conditions); the dominant fire-rate is on logic and completeness checks. Total flagged examples (rightmost) reflects unique examples receiving an "incorrect" verdict overall, not the sum of per-check counts.

Model	Bench	Cell	Comp.	Logic	Compl.	Flagged
Kimi K2.5	WikiTQ	31	1	121	42	41
	TabFact	74	1	168	55	32
GLM 5	WikiTQ	14	5	29	49	87
	TabFact	15	0	58	73	45
MiniMax M2.5	WikiTQ	9	1	83	30	50
	TabFact	20	1	90	92	112
Qwen 3.5+	WikiTQ	18	0	65	153	17
	TabFact	35	3	59	107	10
Gemini 3.1 Pro	WikiTQ	11	0	8	23	60
	TabFact	11	0	31	18	18

Table 4: Model-perceived error-type distribution: counts of Step 2 "incorrect" verdicts by error category, per model and method, summed over both benchmarks and all seeds. Categories follow the 5-type taxonomy of §3.3 (model-self-reported labels, not ground truth, see Limitations). "–" = 0.

Model	Method	Cell	Comp.	Logic	Hallucin.	Partial
Kimi K2.5	Naive	8	11	17	1	98
	SSC	7	10	13	2	36
GLM 5	Naive	4	23	51	3	66
	SSC	3	21	39	3	47
MiniMax M2.5	Naive	1	7	21	–	17
	SSC	3	11	20	–	7
Qwen 3.5+	Naive	3	1	71	2	2
	SSC	2	1	44	3	2
Gemini 3.1 Pro	Naive	1	5	11	1	42
	SSC	1	4	6	–	42

Table 5: Stronger baselines. Baseline A_0 from main runs; for multi-seed cells, A_0 is the cross-seed mean. MV: majority vote over $k = 5$ Step 1 samples (seed=42); MV+SSC: SSC applied to the MV consensus; Self-Critic: critic-then-revise pipeline; Tool-Verify: Python-sandbox arithmetic verification on detected computation errors. Cells with "–" were not run within the camera-ready compute budget; see Limitations.

Model	Bench	Base A_0	SSC SCG	MV Δ	MV+SSC Δ	Self-Critic SCG	Tool-Verify SCG
Kimi K2.5	WikiTQ	69.5	+0.9	–	+7.1	−2.8	+6.3
	TabFact	89.2	+0.1	+4.6	–	+1.9	+4.9
GLM 5	WikiTQ	75.5	+0.1	+2.8	+2.6	–	–
	TabFact	87.7	+0.4	+6.0	+3.3	–	–
MiniMax M2.5	WikiTQ	72.4	−0.5	+5.6	+3.5	–	–
	TabFact	86.9	+0.1	+6.2	+5.8	–	–
Qwen 3.5+	WikiTQ	81.0	−0.4	+0.2	−0.4	−1.8	–
	TabFact	94.7	−0.5	+0.3	−0.5	−1.8	–
Gemini 3.1 Pro	WikiTQ	80.5	−0.2	−0.2	+0.5	–	–
	TabFact	94.8	−0.6	−0.4	−0.5	–	–

Example 1: Gemini 3.1 Pro / `wikitq_331` SSC preserves correct answer.

Question: *how many nations participated in the 1956 winter olympic games?*
Table (medal table, top rows): 1 Soviet Union; 2 Austria; 3 Finland; 4 Switzerland; 5 Sweden; 6 United States; 7 Norway; 8 Italy; 9 Germany; 10 Canada.
Gold answer: 10
Step 1 answer: *"10"* (correct; counts 10 rows in the Nation column)
Naive Step 2: verdict = INCORRECT, error type = *wrong_logic*. Analysis: "the table only lists the top 10 nations by medal count, not all participating nations." **Naive Step 3 corrected:** "32" (over-correction — the model hallucinated a participant count not present in the table).
SSC Step 2: verdict = CORRECT. Cell check confirmed 10 rows in the Nation column; logic check confirmed the question is satisfied by counting listed nations. *Answer preserved.*

Note. Naive review over-applied an unverified prior about Winter Olympics participation; SSC's cell-grounded check anchored the answer to what is actually in the table.

Example 2: Gemini 3.1 Pro / `wikitq_2257` SSC preserves correct answer.

Question: *what was the number of singles between 1982 and 1987?*
Table (truncated): 7 rows of singles released in years 1982, 1982, 1983, 1984, 1986, 1986, 1987.
Gold answer: 7
Step 1 answer: *"7"* (correct; counts all 7 rows in the inclusive range)
Naive Step 2: verdict = INCORRECT, error type = *wrong_logic*. Analysis: " 'between 1982 and 1987' means the endpoints should be excluded; counting only 1983–1986 gives 4." **Naive Step 3 corrected:** "4" (over-correction via incorrect strict-exclusion reading).
SSC Step 2: verdict = CORRECT. Logic check explicitly considered inclusive vs. exclusive interpretation and confirmed the inclusive count of 7 matches the WikiTQ denotation convention.

Note. A genuinely ambiguous English range. Naive committed to the exclusive reading and broke the answer; SSC's logic check kept both readings visible and preserved the gold-consistent answer.

Table 6: Same-family scaling probe. Three sizes from the Qwen 3.5 family on both benchmarks (seed=42). Base A_0 rises monotonically with parameter count; SCG remains negative or near-zero across all three sizes. The crossover region is not crossed within this family within the size range examined.

Size	Bench	A_0	Naive SCG	SSC SCG
Qwen 3.5-9B (dense)	WikiTQ	78.8	-0.6	-0.2
	TabFact	92.0	-1.4	-0.3
Qwen 3.5-27B (dense)	WikiTQ	79.9	-0.5	-0.5
	TabFact	94.2	$+0.1$	$+0.1$
Qwen 3.5+ (397B MoE)	WikiTQ	81.0	-0.5	-0.4
	TabFact	94.7	-1.3	-0.5

Table 7: Inference-mode-controlled probe. Two reasoning-supporting models (GLM 5 and Qwen 3.5+) re-run with reasoning explicitly disabled; the main-run cells (reasoning-ON) are reported with the same models for direct comparison. The qualitative SCG direction is preserved for Qwen 3.5+ across modes; GLM 5's SCG shifts substantially toward more negative values under reasoning-OFF.

Model	Bench	Mode	A_0	SSC SCG
GLM 5	WikiTQ	ON	75.5	$+0.1$
		OFF	76.8	-3.4
	TabFact	ON	87.7	$+0.4$
		OFF	92.0	-4.7
Qwen 3.5+	WikiTQ	ON	81.0	-0.4
		OFF	80.4	-0.6
	TabFact	ON	94.7	-0.5
		OFF	94.4	-1.0

Example 3: Kimi K2.5 / `wikitq_2254` SSC fixes wrong answer.

Question: *which political party had the most female candidates?*
Table (truncated): 5 districts with Democratic and Republican candidates; female candidates are Diana DeGette (Democratic, D1) and Betsy Markey (Democratic, D4), and Marilyn Musgrave (Republican, D4) — Democratic = 2, Republican = 1.
Gold answer: Democratic
Step 1 answer: *"Republican"* (incorrect; miscount)
Naive Step 2 + Step 3: verdict = INCORRECT, corrected to *"Democratic"*.
SSC Step 2 + Step 3: verdict = INCORRECT, error type = *wrong_computation*. Cell check enumerated female candidates by row; computation check verified $2 > 1$; logic check confirmed the question asks for the larger-count party. *Corrected to:* "Democratic".

Note. Both methods caught the error here. SSC adds a verifiable trace (per-check enumeration) that makes the correction auditable, illustrating SSC's *diagnostic value* discussed in §6.

G Related-Work Positioning Table

Table 8: Positioning of our work. ✓ = covered, ✗ = not covered. *Multi-seed* = paired runs across at least two random seeds; *Cross-prov.* = evaluation spans at least three independent model providers; *Tool-aug.* = the method invokes external tools (e.g. sandboxed arithmetic) inside the correction loop.

Study	**Self-Corr.**	**Table Reas.**	**Error Decomp.**	**Multi-seed**	**Cross-prov.**	**Tool-aug.**
SCoRe (ICLR 2025)	✓	✗	✗	✓	✗	✗
SPOC (COLM 2025)	✓	✗	✗	✓	✗	✗
Decomp. SC (2025)	✓	✗	✓	✗	✗	✗
SCD (AAAI 2026)	✓	partial	✗	✗	✗	✗
Table-Critic (ACL 2025)	✓	✓	✗	✗	✗	✗
TableCritic-CogSci 2025	✓	✓	✗	✗	✗	✓
TIDE (ICLR 2025)	✓	✓	✗	✗	✗	✓
CoVe (2023)	✓	✗	✓	✗	✗	✗
Tyen et al. (2024)	✓	✗	✓	✗	✗	✗
Conf. v.s. Crit. (ACL 2025)	✓	✗	✓	✗	✓	✗
Feedback-Ctrl (2026)	✓	✗	✓	✓	✓	✗
Chain-of-Table (ICLR 2024)	✗	✓	✗	✗	✗	✗
Binder (ICLR 2023)	✗	✓	✗	✗	✗	✓
TableMind (WSDM 2026)	✗	✓	✗	✗	✗	✓
Ours	✓	✓	✓	✓	✓	✓

H Ablation and Domain-Transfer Tables

Table 9: Ablation study on Kimi K2.5 (WikiTQ). SCG for Naive (N) and SSC (S); Net(N) = (errors fixed − correct broken) under the Naive condition, in absolute example counts. Ablations were run as separate experiments; the "Full" row reflects this run's base accuracy, which may differ from Table 1's multi-seed mean due to temperature=1 stochasticity and different API samples.

Configuration	A_0	**SCG(N)**	**SCG(S)**	**Net(N)**
Full (main)	67.2	+1.2	+1.9	+12
No reasoning trace	77.0	+0.3	+0.3	+3
Blind review	71.6	−1.5	0.0	−15
Multi-round (2×)	70.4	+0.5	0.0	+5
CoT prompting	67.4	+0.2	+0.2	+2

Table 10: FinQA domain transfer probe (Kimi K2.5, 1,147 examples).

Method	A_0	A_1	SCG	EDR	CSR	Net
Naive	21.5	21.7	+0.2	2.0	14.3	+2
SSC	21.5	21.7	+0.2	1.2	25.0	+2