Deterministic Failure Under Greedy Decoding: Model-Specific Error Patterns Across Prompting Strategies

Anonymous ACL submission

Abstract

This paper introduces a deterministic failure analysis framework for large language models (LLMs) that distinguishes stable, model-specific heuristic substitutions from stochastic noise entirely at the output level, and deploys it to investigate whether LLM failures are random misfires or behaviorally stable, model-specific patterns. To isolate these mechanisms, we evaluate four open-weight models on arithmetic reasoning (SVAMP) and factual recall (TriviaQA) using greedy decoding, thereby eliminating sampling stochasticity and establishing perfect empirical reproducibility within our fixed inference stack (Replication Stability Score, RSS = 1.0). Under this deterministic decoding regime, observed failure patterns cannot be attributed to sampling stochasticity and instead reflect stable model–prompt interactions: vanilla prompting triggers model-specific heuristic substitutions characterized by high-confidence errors (HCS), substantial semantic deviation from the ground truth (NED), and a decoupled or inverted error-difficulty calibration (ARF). We show that advanced prompting strategies explicitly alter these failure pathways. Few-shot prompting serves as a powerful corrective intervention that suppresses heuristics and drives convergent, correctness-driven cross-model agreement (CMA); CMA decomposition shows convergence is correctness-driven (CMA_c accounts for 96–99% of few-shot CMA, not shared errors). Conversely, chain-of-thought prompting acts as a highly volatile diagnostic that either improves arithmetic output accuracy in models with sufficient parametric knowledge or introduces severe verbosity and format-alignment failures. We conclude that, under deterministic greedy decoding, prompting interventions can exert a more profound impact on system reliability than model scale, and we advocate for confidence-aware metrics like HCS to better diagnose true reasoning capacity. Whether these behavioral patterns generalize to stochastic decoding regimes remains an open question.

1 Introduction

Chain-of-thought (CoT) prompting has established itself as a leading way to probe the reasoning ability of large language models, frequently leading to improvements on multi-step tasks given enough model capacity (Wei et al., 2022b). This has led to a heuristic consensus that, under current paradigm constraints, explicit reasoning traces usually render model outputs more computationally robust and alignment-safe, but the question remains: are failures of current open-weight models mostly stochastic (e.g., variations in decoding, nondeterminism in the infrastructure), or caused by particular stable model–prompt interactions?

We assess this question in two complementary task families: arithmetic reasoning and factual recall. In particular, we evaluate four open-weight models, LLaMA-2-7B, Gemma-2-2B, Mistral-7B, and Phi-3-mini under vanilla, few-shot, and CoT settings on SVAMP and TriviaQA (Joshi et al., 2017). The main evaluation task for the entire diagnostic system is SVAMP, with TriviaQA serving as a comparative baseline to: (i) evaluate cross-task generalization, (ii) isolate CoT bottlenecks beyond pure arithmetic constraints, and (iii) determine if heuristic substitutions generalize from numeric values to entity-level outputs.

We go beyond accuracy-based metrics and frame our assessment in terms of failure dynamics by using reproducibility and confidence-based diagnostics (RI, divergence rate, RSS, HCS, NED, and ARF). Our main novelty is a deterministic failure analysis framework that combines these diagnostics to distinguish stochastic fluctuations from stable behavioral failure modes, and to ask if confidence tracks true task difficulty. Empirically, we observe highly reproducible failure patterns and prompt-dependent mitigation, consistent with the

view that failures reflect stable model–prompt interactions rather than stochastic noise.

This framing is informed by prior work on reasoning prompts, scaling behavior, and confidence calibration (Wei et al., 2022b,a; Kaplan et al., 2020; Hoffmann et al., 2022; Kadavath et al., 2022), with a broader synthesis in Section 2. Our focus is on failure as a target phenomenon rather than total error rate: solutions that raise average benchmark scores may not stabilize confidently incorrect behavior, and prompting should be treated as a reliability-level control variable, not a formatting choice.

Finally, we do not seek to generalize all prompting techniques as good or bad, but to characterize under what circumstances and why they change model behavior. By delineating stochastic and behavioral explanations, we hope to guide reliability-oriented evaluation, prompt selection, and downstream system design.

Contributions. (1) We introduce a *deterministic failure analysis framework* that unifies confidence- and semantics-aware diagnostics (HCS, NED, ARF) with reproducibility metrics (RI, divergence rate, RSS), and use it to show that LLM failures are model-specific, stable pattern substitutions rather than stochastic noise, and we additionally propose confidence-aware evaluation metrics (HCS, ARF) that go beyond benchmark accuracy to diagnose model reliability. (2) We systematically compare vanilla, few-shot, and CoT prompting as reliability interventions across arithmetic and factual tasks, showing that prompting effects are task- and model-dependent and that few-shot dominates on both accuracy and reliability under greedy decoding.

2 Background and Related Work

CoT prompting (Wei et al., 2022b) and related techniques (Kojima et al., 2022; Wang et al., 2022; Zhou et al., 2022; Zelikman et al., 2022) show that prompt phrasing can meaningfully shift model reasoning ability. SVAMP (Patel et al., 2021) and TriviaQA (Joshi et al., 2017) are established benchmarks for arithmetic robustness and factual recall, respectively. Prior reliability work (Kadavath et al., 2022; Liu et al., 2023) motivates joint accuracy-and-confidence evaluation, but few studies treat confidently deterministic failure as the main object. Recent work further shows that surface-level prompt formatting independently drives large accuracy swings (Sclar et al., 2024; Lu et al., 2022), motivating our focus on output-level behavioral diagnostics that are robust to such variation. A full review of related work is provided in Appendix A.

3 Datasets and Experimental Setup

To determine whether LLM failures result from stochastic instability or behaviorally stable, deterministic behavior, we measure the performance of four open-weight models on arithmetic reasoning (SVAMP) and fact retrieval (TriviaQA) using a common pipeline.

3.1 Datasets

SVAMP (arithmetic reasoning). We utilize the full SVAMP benchmark (1,000 items), a corpus used to evaluate the robustness of grade-school level arithmetic to linguistic variation (Patel et al., 2021). Each item has a gold numeric answer.

TriviaQA subset. We use a matched 200-question subset of TriviaQA (Joshi et al., 2017), comprising History (40), Science (40), Geography (40), Literature (30), Art & Music (25), and Sports (25). Questions were sampled uniformly at random in each domain stratum from the TriviaQA Wikipedia development(dev) set split with seed 42; question IDs are released with the code. Domain proportions were selected to match the approximate mix of the full dev set split. We use 200 questions to strike a good compromise between deriving a robust measure of domain diversity and maintaining computational parity with the complete SVAMP replication grid (4 models $\times$ 3 prompting strategies $\times$ 5 runs). To confirm a lack of artifacts of selection, we checked for consistent internal rank ordering across all 6 domain categories: per-model accuracy orderings within each domain are consistent with the total task-level ordering in all 6 of 6 categories (Spearman: $\rho \geq 0.80$ within each domain, range 0.80–1.00), with no category exhibiting an unexplained reversal.

3.2 Models

We evaluate four open-weight models: LLaMA-2-7B (Touvron et al., 2023), Gemma-2-2B (Team et al., 2024), Mistral-7B-v0.1 (Jiang et al., 2023), Phi-3-mini-4k-instruct (Abdin et al., 2024). All models are evaluated from public checkpoint releases, using a fixed inference stack. Open-weight models enable the token-level probabilities neces-

sary for confidence and teacher-forced perplexity (PPL) diagnostics, and all listed results use greedy decoding.

3.3 Prompting Strategies and Decoding Protocol

We compare three prompting regimes: **Vanilla** (zero-shot direct QA), **Chain-of-Thought (CoT)** (fixed step-by-step prefix), and **Few-shot** (fixed in-context exemplars). All generations use greedy decoding (`do_sample=False`) with 5 replications per item; token limits are 150 for both SVAMP and TriviaQA. Full template details, replication protocol, and exact exemplar content are given in Appendix B.

3.4 Evaluation Metrics

3.4.1 Primary correctness metrics.

- **Arithmetic Answer Accuracy (AAA, SVAMP only).**

$$\mathrm{Match}_i^{(m)} = \begin{cases} 1, & \text{if } \left|a_i^{(m)} - \hat{a}_i\right| \le 0.05 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

  where $a_i^{(m)}$ is the numeric prediction of model m, $\hat{a}_i$ is the numeric gold answer, an $N_{\text{SVAMP}} = 1000$. Model-level AAA is

$$\mathrm{AAA}^{(m)} = \frac{1}{N_{\text{SVAMP}}} \sum_{i=1}^{N_{\text{SVAMP}}} \mathrm{Match}_i^{(m)}. \tag{2}$$

  The ± 0.05 tolerance accommodates two-decimal rounding while remaining strict enough to reject misidentified integers (minimum error 1.0); robustness checks using ± 0.01 yielded identical results across all 12 conditions (mean absolute difference = 0.0 pp).
- **Exact Match (EM, TriviaQA only):** normalized string equality between prediction and gold (answer/aliases).
- **Answer Token F1 (TriviaQA):** token-overlap F1 between prediction and gold.
- **Absolute Numeric Error (ANE, SVAMP):** $|\hat{y} - y|$.
- **Normalized Edit Distance (NED):** Here, $\hat{p}$ is the prediction of the model, g the numeric gold string or ground-truth (SVAMP), A the alias set (TriviaQA), ν lowercases and strip punctuation, $\mathrm{lev}(\cdot,\cdot)$ is the Levenshtein distance and $|\cdot|$ denotes the length of the string. Both equations return values in $[0, 1]$.

  for SVAMP,

$$\mathrm{NED}_{\text{SVAMP}}(\hat{p},\, g) = \frac{\mathrm{lev}(\hat{p},\, g)}{\max(|\hat{p}|,\, |g|)} \tag{3}$$

  for TriviaQA,

$$\mathrm{NED}_{\text{TriviaQA}}(\hat{p}, A) = \min_{a \in A} \frac{\mathrm{lev}(\nu(\hat{p}), \nu(a))}{\max(|\nu(\hat{p})|, |\nu(a)|)} \tag{4}$$

Perplexity and confidence. For generated token sequence $x_1, \ldots, x_T$,

$$\mathrm{PPL} = \exp\left(-\frac{1}{T}\sum_{t=1}^{T} \ln p(x_t \mid x_{<t})\right). \tag{5}$$

A lower perplexity (PPL) indicates higher model-assigned confidence (serves as a proxy for it)[1] We calculate the PPL for every response (even ones where the outputs were invalid) and then take the means across the 5 runs per item. Aggregated per-model-strategy statistics are $\overline{\mathrm{PPL}} = \frac{1}{M}\sum_{i=1}^{M} \mathrm{PPL}_i$ and $\sigma^2_{\mathrm{PPL}} = \frac{1}{M-1}\sum_{i=1}^{M}(\mathrm{PPL}_i - \overline{\mathrm{PPL}})^2$, where M is the number of evaluated items for that task. We subsequently evaluate the predictive validity of PPL relative to downstream task correctness (Section 4.3).

3.4.2 Diagnostic metrics.

- **Heuristic Confidence Score (HCS, SVAMP only):**

$$\mathrm{HCS} = (1 - \mathrm{AAA}) \times \frac{\mathrm{PPL}_{\text{baseline}}}{\mathrm{PPL}_{\text{mean}}} \tag{6}$$

  where $\mathrm{PPL}_{\text{baseline}}$ is the mean PPL of that model under vanilla prompting, serving as a within-model anchor that normalizes for inter-model fluency differences. High HCS indicates an incorrect response with higher confidence than baseline and the behavioral signature of stable heuristic substitution; low HCS with high error indicates uncertain failure. HCS is measured at the condition level (per model–strategy) using condition-level aggregates of AAA and $\mathrm{PPL}_{\text{mean}}$ with $\mathrm{PPL}_{\text{baseline}}$ fixed to each model's vanilla condition (Table 2). HCS is computed for SVAMP only; on TriviaQA, where the target is named-entity retrieval rather than numeric

[1] We use the natural logarithm throughout, consistent with the HuggingFace Transformers implementation (Hugging Face, 2024) from which token log-probabilities are extracted. PPL values are therefore in nats.

computation, and confidence is assessed via $\overline{\text{PPL}}_{\text{cor}}$, $\overline{\text{PPL}}_{\text{wrg}}$, their gap, and $r(\text{PPL}, \text{EM})$ (Table 4).

- **Replication Stability Score (RSS):**

$$\text{RSS} = \big(1 - d_{\text{rate}}\big) \times \left(1 - \min\left(\frac{\text{RI}}{0.01}, 1.0\right)\right) \tag{7}$$

where d_{rate} is the divergence rate. The value of RSS lies in $[0, 1]$. RI is defined as the mean per-token coefficient of variation across 5 teacher-forced runs in probability space:

$$\text{RI} = \frac{1}{|\mathcal{T}_{\text{valid}}|} \sum_{t \in \mathcal{T}_{\text{valid}}} \frac{\sigma_{p,t}}{\mu_{p,t}}, \tag{8}$$

$$p_t = \exp(\ell_t), \qquad \mathcal{T}_{\text{valid}} = \{t : \mu_{p,t} \geq 10^{-6}\}$$

where $\ell_t = \ln p(x_t \mid x_{<t})$, $p_t = \exp(\ell_t) \in (0, 1]$ is its probability-space counterpart, and $\mu_{p,t}$, $\sigma_{p,t}$ are the mean and standard deviation of p_t across the 5 runs. Operating in probability space yields a CV monotone in token-level uncertainty, avoiding the sign-convention artifact of dividing by $|\mu_t|$ in log-probability space; the threshold 0.01 separates negligible from non-negligible run-to-run variation. Under greedy decoding $\sigma_{p,t} = 0$ for every t, yielding $\text{RI} = 0$. An item is marked divergent if 2 or more of 5 greedy-run answers differ after task-appropriate canonicalization (normalized string for TriviaQA; extracted numeric answer for SVAMP). High RSS with low AAA is the key signature: the model fails deterministically, ruling out stochastic instability as the cause (full regime interpretation Appendix C.1).

- **Arithmetic Reasoning Fidelity (ARF, SVAMP only):** Spearman rank correlation between per-item confidence and inverse operation-difficulty rank, $\text{ARF} = \rho_s\left(\frac{1}{\text{PPL}_i}, \frac{1}{r_i}\right)$, $r_i \in \{1, 2, 3, 4, 5\}$. Difficulty ranks are defined empirically as the inverse of per-operation accuracy averaged across all four models under vanilla prompting: addition (10.2%, rank 1), division (7.6%, rank 2), multiplication (6.1%, rank 3), addition+subtraction composite (4.2%, rank 4), subtraction (4.1%, rank 5); full values in Table 8. Positive ARF indicates confidence tracks easier operations; negative ARF indicates miscalibration relative to measured difficulty.

- **Cross-Model Agreement (CMA):** fraction of items where two models produce the same normalized answer. For six pairs of models (4 models), we compute the CMA; the strategy-level summaries of the CMA are reported as the mean pairwise CMA at that prompting condition.

Error taxonomy and normalization. We classify outputs as `correct`, `wrong_entity` / `context_number`, `wrong_other`/`wrong_number`, `formatting_artifact`, or `empty` for TriviaQA/SVAMP. All outputs are lowercased and punctuation-stripped before metric computation.

We evaluate each model along six diagnostic axes (correctness, error size, confidence failure, run stability, confidence–difficulty calibration, and failure similarity); the full framework definition and interpretation is provided in Appendix C.

We report bootstrapped 95% confidence intervals (2,000 resamples; seed = 42) for all primary accuracy metrics (AAA, EM) alongside point estimates. Intervals are computed over the N evaluation items per condition by resampling item-level correctness indicators. Full CI values are reported in Table 6 (Appendix D).

4 Results

4.1 Overall accuracy

Figure 1 displays exact-match accuracy on SVAMP (AAA) and TriviaQA (EM) side by side. Under vanilla prompting, accuracy on SVAMP is near chance for three of four models (3.2–6.7%), with Phi-3-mini as a partial exception (14.4%). Three of the four models achieve substantially higher accuracy on TriviaQA (58.0–84.5%), reflecting the availability of relevant factual knowledge from pretraining. Few-shot prompting provides significant improvements on both tasks, particularly for Phi-3-mini (SVAMP: 14.4% $\rightarrow$ 87.7%; TriviaQA: 35.0% $\rightarrow$ 81.0%). These gains are statistically robust with non-overlapping 95% bootstrapped CIs for all four models on SVAMP (Table 6). CoT is inconsistent on arithmetic and consistently harmful on factual recall.

As shown in Table 1, few-shot yields the highest AAA for all four models and the best or near-best TriviaQA EM/F1. CoT improves SVAMP for some models (notably Phi-3-mini and Mistral-7B) but uniformly hurts TriviaQA EM/F1, consistent

with verbosity and format-alignment issues on factual recall.

Figure 2 shows answer-token F1 on TriviaQA. The EM–F1 gap remains high across all conditions, because models still predict the correct entity but in a longer string that fails the normalized exact match. Few-shot reduces this gap to $3.2 - 6.8$ percentage points across all four models, while CoT leaves it wide at $7.1 - 18.2$ percentage points. The most severe case is Phi-3-mini CoT: EM $= 23.5\%$ against F1 $= 0.418$, an 18.2 percentage-point gap, as this demonstrates not factual failure, but verbosity failure. Phi-3-mini few-shot, however, reduces it to EM $= 81.0\%$, F1 $= 0.878$, a gap of only 6.8 percentage points.

4.2 Deterministic behavior and confidence

Determinism is perfect across all 12 model-strategy combinations on both datasets: RSS = 1.0, RI = 0 (all token probability distributions are identical across the 5 runs, so $\sigma_{p,t} = 0$ for every t), and divergence rate = 0. Full confidence and stability metrics are reported in Tables 2 and 4 (Appendix D).

On SVAMP, vanilla prompting yields high HCS (0.856–0.968): copying a context number or producing a wrong-computation response yields short, high-frequency token sequences that are highly probable regardless of correctness. Few-shot yields much lower HCS (0.115–0.426), reflecting reduced error rate and fluency closer to baseline, and lowers both ANE and NED (few-shot NED range: 0.043–0.128) relative to vanilla. CoT reduces HCS for models with sufficient parametric knowledge (LLaMA-2-7B: $0.933 \rightarrow 0.224$; Phi-3-mini: $0.856 \rightarrow 0.306$) but raises it for models without (Gemma-2-2B: $0.961 \rightarrow 0.882$), confirming that reduced output confidence alone does not ensure arithmetic correctness but parametric capacity is also required.

On TriviaQA, HCS is not applicable because the target is named-entity retrieval rather than numeric computation. Confidence separation is poor across almost all vanilla and few-shot conditions (low PPL gap, nearly all $r(\text{PPL}, \text{EM})$ insignificant). The exception is Mistral-7B CoT ($\sigma_{\text{PPL}} = 133.166$, Gap $= -49.5$): incorrect predictions carry very high PPL ($\overline{\text{PPL}}_{\text{wrg}} = 52.4$) from CoT-induced verbose outputs. In contrast, correct predictions are short and smooth ($\overline{\text{PPL}}_{\text{cor}} = 2.9$). This shows a different failure mode: *uncertain wrong* instead of *confident wrong* (discussed further in Section 4.3).

4.3 The PPL–Correctness Inversion

A common assumption is that lower perplexity signals correctness: if a model assigns high probability to its output, that output should be more likely to be right. We test this directly by comparing the mean PPL for correct versus incorrect predictions.

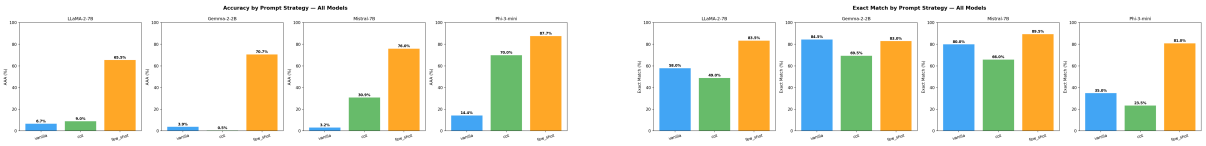
Correct predictions show *greater* mean PPL than incorrect ones in 10 of 12 conditions on SVAMP and 9 of 12 on TriviaQA, and the opposite of what the assumption predicts. The inversion is structural: PPL measures output *fluency*, not accuracy. Short high-frequency sequences (context-number copy, plausible but wrong entity) will get low PPL regardless of correctness. In contrast, correct reasoning sequences (multi-step derivations, specific entity names) will get low frequency, and thus raise PPL even when right.

Three corollaries all fail in the same direction. **(P1)** Wrong answers should have a higher PPL than correct ones (violated in 10 of 12 SVAMP conditions; e.g. Phi-3-mini CoT: $\overline{\text{PPL}}_{\text{cor}} = 3.428$, $\overline{\text{PPL}}_{\text{wrg}} = 2.668$). **(P2)** Lower-PPL conditions are not more accurate: Gemma-2-2B vanilla achieves the study's lowest PPL (1.916) yet the lowest SVAMP accuracy (3.9%), and LLaMA-2-7B vanilla and few-shot have nearly identical PPL (2.172 vs. 2.144, $\Delta = 0.028$) despite a 58.8-point accuracy gap. **(P3)** Per-problem PPL correlates *positively* with correctness in 10 of 12 conditions; the effect is significant for Phi-3-mini CoT on both tasks (SVAMP: $r = +0.603$, $p < 0.001$; TriviaQA: $r = +0.293$, $p < 0.001$).

On TriviaQA, the PPL gap ($\overline{\text{PPL}}_{\text{cor}} - \overline{\text{PPL}}_{\text{wrg}}$) is positive in 9 of 12 conditions (Table 4, Appendix D). The anomalous Mistral-7B CoT condition (Gap $= -49.5$) does not contradict the mechanism: CoT-induced verbosity produces long, low-probability wrong chains ($\overline{\text{PPL}}_{\text{wrg}} = 52.4$). At the same time, the few correct predictions are short and fluent ($\overline{\text{PPL}}_{\text{cor}} = 2.9$). This is a distinct *uncertain wrong* failure mode rather than *confident wrong*, where CoT-induced verbosity, rather than heuristic substitution, drives the error.

4.4 Error patterns

Figure 3 illustrates error category distributions per task. For SVAMP, context-number copying is the dominant error under vanilla prompting with LLaMA-2-7B (62.9%) and Phi-3-mini (72.1%),



(a) SVAMP exact-match accuracy (AAA). (b) TriviaQA exact-match accuracy (EM).

Figure 1: Side-by-side exact-match comparison across arithmetic (SVAMP) and factual (TriviaQA) tasks.

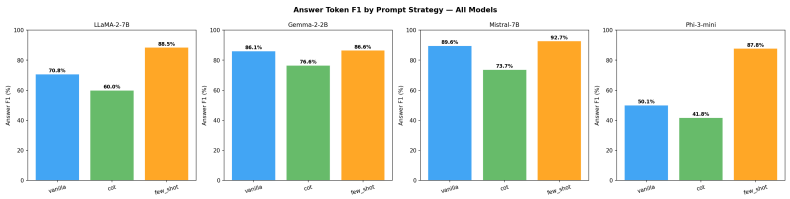


Figure 2: Answer-token F1 on TriviaQA. The EM–F1 gap under CoT highlights verbosity failure.

while computational error dominates for Gemma-2-2B (76.9%) and Mistral-7B (76.3%). Both LLaMA-2-7B and Phi-3-mini are instruction-tuned; the error-type split likely reflects post-training and data differences rather than raw architectural design, so we use the term *model-specific* rather than *architecture-specific*. Gemma-2-2B is a notable case: CoT reduces accuracy further from 3.9% to 0.5%. Table 2 explains the collapse as HCS remains high at 0.882, and σ_{PPL} falls from 0.646 to 0.586 under CoT, meaning the step-by-step instruction compresses the output distribution to a single confident wrong answer with even lower cross-problem variability than under vanilla prompting.

For TriviaQA, the wrong-entity errors dominate under vanilla prompting and CoT. Few-shot significantly reduces the dominant error modes for both tasks.

Figure 4 provides a similar token-level resolution: under vanilla prompting, LLaMA-2-7B attends 0.75 (over 3 failures) to a single early token, while each operand and operator receives less than 0.10; for few-shot prompting, attention re-weights, and the model succeeds. Phi-3-mini shows similar vanilla concentration (mean weight 0.71-0.72), which collapses under CoT, where attention redistributes to mid-sequence content tokens, which is consistent with the context-copying heuristic.

4.5 Operation-level and category-level performance

Figure 5 shows operation-level accuracy on SVAMP and category-level EM on TriviaQA. Few-shot gains are broad on SVAMP, while on TriviaQA, Phi-3-mini shows the widest within-model domain gap under vanilla prompting (ranging from 18% on Science to 57% on Geography), consistent with uneven factual coverage from instruction tuning across knowledge categories.

4.6 Cross-model agreement

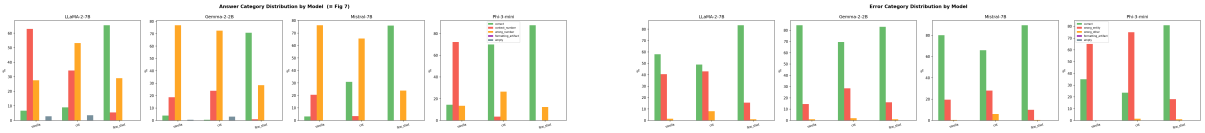
Figure 6 presents CMA heatmaps for both tasks. Agreement is low under vanilla and CoT but rises sharply under few-shot; pairwise values are in Table 3 (Appendix D).

Few-shot increases CMA because shared demonstration format, answer style, and termination conditions constrain the output space. The higher vanilla CMA on TriviaQA (mean 0.438) versus SVAMP (mean 0.196) reflects a structural difference: on SVAMP, each model defaults to its own heuristic (context-number copying for LLaMA-2-7B and Phi-3-mini, wrong computation for Gemma-2-2B and Mistral-7B), producing model-specific wrong answers that rarely coincide; on TriviaQA, plausible-but-wrong entities are shared across pretraining corpora. CoT reduces agreement further as models diverge in intermediate reasoning format.

To determine whether few-shot agreement reflects shared correct reasoning or shared heuristic failure, we decompose CMA into correct-agreement (CMA_c: both models correct on the same item) and wrong-agreement (CMA_w: both models produce the same wrong answer); full values are in Table 7 (Appendix D). Under vanilla prompting, CMA is driven almost entirely by CMA_w: Gemma-2-2B vs. Mistral-7B shows $\text{CMA}_{\text{total}} = 0.330$ yet $\text{CMA}_c = 0.003$ and $\text{CMA}_w = 0.327$, and averaging across all six model pairs, mean $\text{CMA}_c = 0.009$ versus mean $\text{CMA}_w = 0.187$. Under few-shot the pattern in-

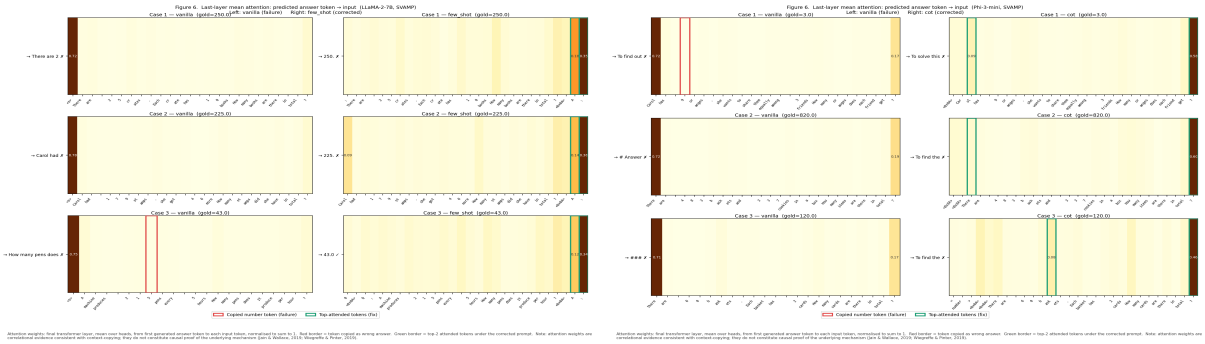
Table 1: Compact performance summary across both tasks (best value per model in bold). AAA is SVAMP exact match; EM and F1 are TriviaQA exact match and token-level F1. AAA and EM shown as point estimate ± half-width of 95% bootstrapped CI (2,000 resamples; seed = 42). F1 is reported as a point estimate; full CI table in Table 6.

Model	Strategy	AAA (%)	EM (%)	F1
LLaMA-2-7B	Vanilla	$6.7_{\pm 1.6}$	$58.0_{\pm 7.0}$	0.708
	CoT	$9.0_{\pm 1.7}$	$49.0_{\pm 6.8}$	0.600
	Few-shot	$\mathbf{65.5}_{\pm 3.0}$	$\mathbf{83.5}_{\pm 5.0}$	**0.885**
Gemma-2-2B	Vanilla	$3.9_{\pm 1.2}$	$\mathbf{84.5}_{\pm 5.0}$	0.861
	CoT	$0.5_{\pm 0.5}$	$69.5_{\pm 6.0}$	0.766
	Few-shot	$\mathbf{70.7}_{\pm 2.8}$	$83.0_{\pm 5.0}$	**0.866**
Mistral-7B	Vanilla	$3.2_{\pm 1.0}$	$80.0_{\pm 5.3}$	0.896
	CoT	$30.9_{\pm 2.9}$	$66.0_{\pm 6.5}$	0.737
	Few-shot	$\mathbf{76.0}_{\pm 2.7}$	$\mathbf{89.5}_{\pm 4.3}$	**0.927**
Phi-3-mini	Vanilla	$14.4_{\pm 2.2}$	$35.0_{\pm 6.8}$	0.501
	CoT	$70.0_{\pm 2.9}$	$23.5_{\pm 5.8}$	0.418
	Few-shot	$\mathbf{87.7}_{\pm 2.0}$	$\mathbf{81.0}_{\pm 5.2}$	**0.878**



SVAMP error categories TriviaQA error categories

Figure 3: Error category distribution by task. Few-shot prompting suppresses dominant heuristic errors.



(a) LLaMA-2-7B: vanilla (failure, left) vs. few-shot (corrected, right). Red border marks the copied number token; green marks top-attended tokens under few-shot.

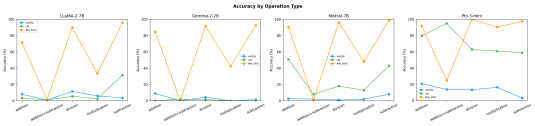
(b) Phi-3-mini: vanilla (failure, left) vs. CoT (corrected, right).

Figure 4: Last-layer mean attention from the predicted answer token to input tokens for representative SVAMP examples. Attention is averaged across heads and normalized. Patterns suggest context-copying behavior (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019).

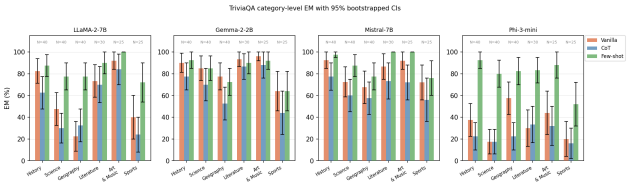
verts: mean CMA_c rises to 0.628 while CMA_w falls to 0.008, with CMA_c accounting for 96–99% of $\text{CMA}_{\text{total}}$ across all pairs (e.g. Mistral-7B vs. Phi-3-mini: $\text{CMA}_c = 0.741$, $\text{CMA}_w = 0.002$). Few-shot convergence therefore reflects shared *correct* reasoning, not shared confabulation.

4.7 Synthesis

These results support the view that dominant open-weight LLM failure modes are behaviorally deterministic and model-specific, manifesting as stable output-level error patterns rather than stochastic misfires. RSS=1.0 perfect replication stability, strong vanilla/CoT SVAMP HCS, and dominant task-specific error modes (context copying, wrong-entity retrieval) are consistent with confident heuristic substitution. Few-shot avoids these heuristics and improves both arithmetic and factual accuracy — improvements that are statistically significant for all four models (e.g., in LLaMA-2-7B AAA, 6.7% to 65.5%, a 58.8 point gap with non-overlapping 95% bootstrapped CIs; Phi-3-mini AAA: 14.4% to 87.7%, a 73.3 point gap; full CIs in Table 6). The concurrent rise in CMA confirms that convergence is toward cor-

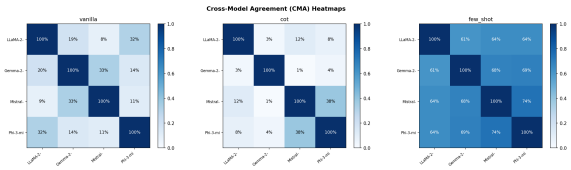


(a) SVAMP operation-level accuracy.

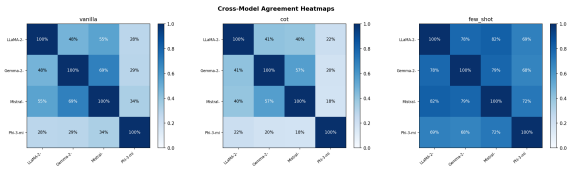


(b) TriviaQA category-level EM. Error bars show 95% bootstrapped CIs (2,000 resamples; seed = 42). Domain sizes: History, Science, Geography N=40; Literature N=30; Art & Music, Sports N=25. Small-N domains should be interpreted with caution.

Figure 5: Fine-grained task breakdown across math and factual evaluation.



(a) SVAMP CMA (math).



(b) TriviaQA CMA (factual).

Figure 6: Cross-model agreement (CMA) comparison across datasets. Few-shot raises agreement in both settings, indicating shared output-space anchoring.

rect answers, specifically, not shared confabulation (Table 7). Despite these accuracy gains, ARF remains negative or near-zero across all 12 conditions (Table 5), confirming that confidence is driven by local fluency rather than arithmetic difficulty: a model can become more accurate without becoming better calibrated.

While CoT can improve arithmetic output accuracy in stronger models by providing a format constraint that elicits more complete numerical derivations, it consistently worsens factual exact-match through verbosity and format drift. The clearest example is Phi-3-mini: CoT produces a $+55.6$ percentage point gain on SVAMP (14.4% to 70.0%) while reducing TriviaQA accuracy by -11.5 percentage points (35.0% to 23.5%), opposite effects for the same model, with the same prompt style. This reversal proves that CoT effects are task-conditioned, not model-conditioned: step-by-step scaffolding is associated with improved arithmetic output accuracy in models where sufficient parametric knowledge exists, but imposes a format cost on factual retrieval where direct entity generation is more reliable than a reasoning chain.

Few-shot is the most robust intervention in this setting, but carries a cost: the 3-shot exemplars add a mean of 187 tokens for SVAMP and 142 for TriviaQA, increasing input processing time by approximately 12–18% on our H100 setup; see Appendices E and F for implications and limitations.

5 Conclusion

We show that failure in present-day open-weight LLMs is predominantly behavioral rather than stochastic: under greedy decoding, errors are perfectly reproducible, model-specific, and prompt-sensitive. This has a direct practical consequence: deterministic failure is systematically treatable, and prompting interventions can exert a more profound impact on reliability than the model scale.

Few-shot is the most robust intervention, suppressing heuristic substitution and driving convergence toward correct predictions across models. CoT is better treated as a diagnostic: it can improve arithmetic output accuracy in models with sufficient parametric knowledge, but consistently degrades factual exact-match through verbosity and format drift. Critically, neither intervention improves confidence calibration; ARF remains near zero or negative even at high accuracy, meaning benchmark gains can coexist with fluent, confidently wrong outputs.

Effectiveness declarations should therefore rest on stability, confidence behavior, and error-structure diagnostics, rather than on aggregated accuracy alone. These findings apply to open-weight models in the 2–7B parameter range under greedy decoding; generalization to larger scales, proprietary systems, and non-greedy decoding policies remains an open question.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Singh Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio C'esar Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, and 70 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv*, abs/2404.14219.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10.

Hugging Face. 2024. Perplexity of fixed-length models. *Hugging Face Transformers Documentation*. Accessed: 2026-03-28.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, and 1 others. 2021. Show your work: Scratchpads for intermediate computation with language models.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2080–2094.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, volume 2024, pages 25055–25083.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 11–20.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Background and Related Work

Reasoning via prompting. CoT prompting shows that eliciting intermediate steps can thus greatly improve multi-step tasks in sufficiently large models (Wei et al., 2022b). Other related techniques (such as zero-shot reasoning triggers (Kojima et al., 2022), self-CoT decoding (Wang et al., 2022), least-to-most prompting (Zhou et al., 2022), and self-improvement pipelines such as STaR (Zelikman et al., 2022)) likewise demonstrate that prompt phraseology and decoding methods can meaningfully influence reasoning behavior. Earlier work that revealed the utility of a scratchpad also recommended disclosing intermediate calculations for arithmetic machine-learning tasks (Nye et al., 2021). Together, these works point toward understanding prompting as a behavioral modification, not merely a stylistic formatting choice.

Arithmetic reasoning benchmarks. The reliability of numerical reasoning is still one of the strongest LLM tests. SVAMP was specifically designed to avoid using shortcuts by adding linguistic variation in an arithmetic word problem dataset (Patel et al., 2021). Similar works that test for general quantitative reasoning at various levels of complexity include GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). Past work often finds significant improvement when using methods like CoT on these benchmarks but tends to prioritize overall accuracy over the study of whether their failure cases are reproducible, confidence misaligned, and far semantically from the ground-truth solutions.

Factual recall benchmarks. In terms of factual reasoning, TriviaQA offers an interesting adversary to the evidence-intensive questions, lexical variation and answer normalization problems, which highlights the brittleness of accuracy under an exact-match metric (Joshi et al., 2017). This is directly relevant to our experiments, where the addition of words prompted by the system, or the changing of indentation and format, could alter the scores even if some of the facts conveyed were correct.

Prompting overviews and reliability perspective. Survey work unifies prompting across discrete prompts, in-context learning, soft prompts, and reasoning-tuned approaches, identifying prompt design as a dominant factor in downstream behavior (Liu et al., 2023). At the same time, reliability and calibration studies demonstrate that model confidence is occasionally informative but always imperfect at conveying certainty under reasoning and distributional stress (Kadavath et al., 2022). Together, these results motivate evaluation protocols that measure correctness and confidence jointly rather than accuracy alone.

Prompt sensitivity and formatting effects. Recent work has demonstrated that model outputs are sensitive not only to prompt content but to surface-level formatting choices. Sclar et al. (2024) show that minor phrasing variations produce large accuracy swings across tasks, while Lu et al. (2022) demonstrate that few-shot example ordering alone can shift performance substantially. Formatting brittleness is particularly acute for exact-match metrics: instruction-tuned models frequently embed correct content within the verbose outputs that fail normalization, a pattern our CoT results on TriviaQA directly instantiate. Our work complements this literature by isolating formatting-driven failure as a measurable, reproducible behavioral signature rather than treating it as noise.

Gap addressed by this work. Though significant work is being done in this direction, comparatively few studies treat confidently determin-

istic failure as a first-class object of analysis, and relatively few compare arithmetic reasoning to factual recall within a shared diagnostics framework. Evaluations centered on CoT tend to focus on average performance improvements. They do not often evaluate if failures are rerun-stable, if wrong answers are produced in high confidence, and how prompting styles cause changes in these phenomena across task families. Our paper fills this niche by evaluating reproducibility (RI, divergence rate, RSS), confidence-correlated failure (HCS, ARF), and semantic deviation (NED) across SVAMP and TriviaQA with vanilla, few-shot, and CoT prompts.

B Prompting Strategies and Decoding Protocol

B.1 Prompting Strategies

We compare three prompting regimes: templates are task-specific but held fixed across all models within a task.

- **Vanilla (zero-shot):** Direct QA format but no demonstrations. For TriviaQA: `Q: {question}` `nA:`. For SVAMP: the input is the concatenated body and question text (no exemplars).
- **Chain-of-Thought (CoT):** A fixed instruction prefix asks for step-by-step reasoning and enforces a final-line answer style (`Answer: <...>`), then the same task-specific base prompt as vanilla.
- **Few-shot (3-shot):** Three fixed in-context examples are prepended, then the test item in QA format, the examples are non-overlapping with the evaluation items (SVAMP: three exemplars of arithmetic questions with numeric answers; TriviaQA: three exemplars of factual questions).

B.2 Decoding and Replication Protocol

Greedy decoding is used across all generations (`do_sample=False`). We generate 5 replications for each model-prompt-item combination. The cap on produced tokens is 150 for both SVAMP and TriviaQA. TriviaQA outputs are well within this bound (vanilla mean: 59.5 tokens; CoT: 0% reach the cap). For SVAMP, in few-shot, most of the outputs reach the cap (LLaMA-2-7B: 100%, Mistral-7B: 99%, Phi-3-mini: 87%, Gemma-2-2B: 73%), because the few-shot prompt prepends three exemplars and models continue generating until the truncation. Since the numeric answer is produced within the first (5-10) tokens of generation, truncation affects trailing content rather than the answer token; accuracy on capped vs. uncapped items differs by at most 0.01 for three of four models (LLaMA-2-7B: all outputs capped, not within-model comparison possible). We measure determinism along two axes: (i) the *divergence rate*, i.e., whether answer strings vary over the five runs; and (ii) the RI (Equation 8), i.e., the mean coefficient of variation of per-token *probabilities* $p_t = \exp(\ell_t)$, where ℓ_t is the natural-log token log-probability, across the 5 teacher-forced runs, which equals zero under greedy decoding. If both measures are zero for each run, then we treat behavior as empirically deterministic, given the hardware and software setup we deploy. All models are tested with the same decoding parameters and consistent prompt templates to avoid ambiguity with sampler or template-generated variance.

B.3 Exemplar Content

The three fixed in-context exemplars used for few-shot prompting are listed below for each task. The SVAMP examples demonstrate subtraction through Examples 1 and 3 and multiplication through Example 2. The evaluation set contains five operation types, which include addition with 300 items making up 30 percent of the total, subtraction with 200 items making up 20 percent, multiplication with 200 items making up 20 percent, division with 200 items making up 20 percent, and addition + subtraction composite with 100 items making up 10 percent. The exemplar set does not include addition and division because the fixed exemplars had to be established before evaluation, and the operation coverage assessment determined which exemplars to use. The partial coverage serves as an established limitation that is addressed in Appendix F. The TriviaQA exemplars include three domains, which are art history, chemistry, and sports history; the evaluation subset contains samples from all three domains. Exact string matching on the question text demonstrated that all exemplars did not overlap with the evaluation items.

SVAMP exemplars (3-shot).

1. **Body & Question:** Each pack of dvds costs 76 dollars. There is a discount of 25 dollars on each pack. How much do you have to pay?

Answer: 51
Operation: subtraction

2. **Body & Question:** A farmer has 6 fields. Each field produces 348 kg of wheat. How many kg of wheat total?
Answer: 2088
Operation: multiplication
3. **Body & Question:** There were 22 parents and 676 people in total. How many pupils were present?
Answer: 654
Operation: subtraction

TriviaQA exemplars (3-shot).

1. **Question:** Who painted the Mona Lisa?
Answer: Leonardo da Vinci
Domain: Art & Music
2. **Question:** What is the chemical symbol for gold?
Answer: Au
Domain: Science
3. **Question:** In which city were the first modern Olympic Games held?
Answer: Athens
Domain: Sports

C Diagnostic Framework

We evaluate model on six axes: (1) correctness in AAA/EM and error types; (2) error size in ANE/NED; (3) confidence failure in PPL, HCS; (4) run stability in divergence as measured by rate, RI, RSS; (5) confidence difficulty in arithmetic as measured by ARF, computed as Spearman rank correlation between $1/\mathrm{PPL}$ and inverse operation-difficulty rank; and (6) failure similarity across models as measured by pairwise CMA. These axes expose each network's behavior along some of the most relevant dimensions of the deterministic-heuristic hypothesis: high RSS with low AAA indicates stable heuristic failure, while ARF near zero indicates confidence decoupled from arithmetic difficulty.

C.1 Diagnostic Regime Interpretation

Four diagnostic regimes emerge from the joint (RSS, AAA) space:

RSS	AAA/EM	Interpretation
High	High	Stable and correct (ideal)
High	Low	Stable heuristic failure
Low	Low	Chaotic failure
Low	High	Unstable but lucky (rare)

High RSS with low AAA is the key signature: the model fails *deterministically*, ruling out stochastic instability as the cause. This regime is the primary target of our analysis as it distinguishes confidently wrong heuristic behaviour from both lucky instability (Low/High) and chaotic failure (Low/Low).

D Additional Tables

E Discussion

Failures that are reproducible are systematically treatable. The behavioral-level view supported by our results — that failure reflects stable model–prompt interactions shaped by post-training and data-mixture differences, rather than noisy input-dependent perturbations — has a direct practical consequence: prompt selection is a mechanism-level control variable, not a formatting choice. Few-shot prompting suppresses model-specific pattern substitution; CoT is better understood as a diagnostic associated with improved arithmetic output accuracy in models where parametric knowledge is sufficient and format-alignment fragility in others.

A key implication is that the average accuracy is insufficient for measuring reliability. Our results show conditions where accuracy increases while the association between confidence and true difficulty remains weak (ARF near zero or negative), and where fluent but incorrect responses are assigned high confidence. Reliable evaluation must account for confidence behavior and error structure, not only benchmark scores.

These findings are specific to four open-weight models in the 2–7B parameter range under greedy decoding. Generalization to larger scales, proprietary systems, non-English languages, retrieval-augmented models, and non-greedy decoding policies remains an open question. Natural methodological follow-ups include: an intervention study varying exemplar structure, output constraints, and decomposition format to isolate causal components of the prompt; a calibration analysis disentangling token fluency from epistemic confidence; and a training-time objective penalizing stable high-confidence failures.

F Limitations

Model and scale coverage. We test only four open-weight models (LLaMA-2-7B, Gemma-2-2B, Mistral-7B, Phi-3-mini), all on the 2-7B pa-

Table 2: Confidence and stability metrics on SVAMP (N=1,000). $\overline{\text{PPL}}$ is the teacher-forced perplexity, σ_{PPL} is its standard deviation, ANE is the absolute numeric error on incorrect predictions, and NED is the normalized edit distance over all predictions. HCS is normalized by each model's vanilla $\overline{\text{PPL}}$ (see Equation 6), making values comparable across models; vanilla HCS therefore equals $1 - \text{AAA}$ by construction. RSS is derived from RI and the divergence rate (Equations 7 and 8); RI is computed as the mean coefficient of variation of per-token probabilities across 5 runs and equals zero under greedy decoding for all conditions.

Model	**Strategy**	$\overline{\text{PPL}}$	σ_{PPL}	**HCS**	**RSS**	**ANE**	**NED**
LLaMA-2-7B	Vanilla	2.172	0.338	0.9330	1.000	81.0	0.452
	CoT	8.839	18.508	0.2236	1.000	202.0	0.449
	Few-shot	2.144	0.228	0.3495	1.000	56.0	0.128
Gemma-2-2B	Vanilla	1.916	0.646	0.9610	1.000	181.0	0.494
	CoT	2.162	0.586	0.8818	1.000	223.0	0.519
	Few-shot	1.779	0.115	0.3156	1.000	70.0	0.118
Mistral-7B	Vanilla	3.515	1.178	0.9680	1.000	218.0	0.486
	CoT	2.954	1.062	0.8222	1.000	165.0	0.332
	Few-shot	1.980	0.282	0.4261	1.000	73.0	0.099
Phi-3-mini	Vanilla	2.414	1.522	0.8560	1.000	163.0	0.432
	CoT	2.367	0.764	0.3060	1.000	156.0	0.147
	Few-shot	2.593	1.005	0.1145	1.000	80.0	0.043

Table 3: Pairwise Cross-Model Agreement (CMA) across SVAMP and TriviaQA for all model pairs and prompting strategies. A higher CMA indicates that models converge to the same answers (correct or incorrect).

		Vanilla		**CoT**		**Few-shot**	
Model A	**Model B**	SVAMP	QA	SVAMP	QA	SVAMP	QA
LLaMA-2-7B	Gemma-2-2B	0.192	0.475	0.028	0.410	0.612	0.785
LLaMA-2-7B	Mistral-7B	0.083	0.550	0.116	0.395	0.642	0.815
LLaMA-2-7B	Phi-3-mini	0.315	0.285	0.078	0.220	0.642	0.690
Gemma-2-2B	Mistral-7B	0.330	0.690	0.006	0.570	0.682	0.790
Gemma-2-2B	Phi-3-mini	0.141	0.290	0.041	0.195	0.691	0.675
Mistral-7B	Phi-3-mini	0.114	0.335	0.383	0.185	0.743	0.725
Mean		0.196	0.438	0.109	0.329	0.669	0.747

rameter scale. Any conclusions drawn about these stable behavioral failure patterns may not apply to larger-scale models (e.g., 70B+ parameters) or proprietary systems, which may use a different internal decoding pipeline.

Task coverage. Results are restricted to arithmetic word problems (SVAMP) and open-domain factual recall (TriviaQA). Tasks requiring code generation, or commonsense inference, may exhibit different failure dynamics.

Decoding regime. All experiments use greedy decoding (temperature $= 0$), which maximizes reproducibility but does not reflect deployment settings that use sampling. The determinism result is specific to this regime; stochastic decoding may yield different RI and divergence-rate profiles.

Prompt template sensitivity. Since we keep the prompt templates fixed across models for fairness, the absolute performance of any prompting method will depend on the template wording. A different phrasing of the CoT prefix or a few-shot exemplar set might affect the outcomes.

Exemplar coverage. The three SVAMP few-shot exemplars cover subtraction and multiplication only; addition (30% of evaluation items) and division (20%) are absent from the exemplar set. This may differentially affect per-operation ARF values and per-operation accuracy under few-shot prompting, since the exemplar format provides no demonstration of addition or division solution patterns. The strong overall few-shot gains (Table 1) suggest this limitation did not prevent generalization, but exemplar sets with full operation coverage should be evaluated in future work.

Few-shot depth. All few-shot experiments use a fixed 3-shot exemplar set. Whether 1-shot or 5-shot prompting changes the conclusions about few-shot as the most robust intervention remains untested. Future work should evaluate performance across varying numbers of exemplars to de-

Table 4: Confidence and stability metrics on TriviaQA (N=200). HCS is not reported for this task; see text. Gap = $\overline{\text{PPL}}_{\text{cor}} - \overline{\text{PPL}}_{\text{wrg}}$: positive values indicate the model assigns higher perplexity to correct predictions than to incorrect ones. RSS is derived from RI, and the divergence rate (Equations 7 and 8); RI equals zero for all conditions under greedy decoding, yielding RSS = 1.000 for all configurations. $r(\overline{\text{PPL}}, \text{EM})$ is the Pearson correlation between per-item PPL and exact-match correctness; significance is assessed by a two-sided t-test. $^{***}p<0.001$, $^{*}p<0.05$, ^{ns} not significant.

Model	**Strategy**	$\overline{\text{PPL}}$	σ_{PPL}	$\overline{\text{PPL}}_{\text{cor}}$	$\overline{\text{PPL}}_{\text{wrg}}$	**Gap**	**RSS**	$r(\text{PPL}, \text{EM})$
LLaMA-2-7B	Vanilla	2.971	0.457	3.068	2.837	+0.231	1.000	+0.250***
	CoT	4.038	2.163	4.151	3.930	+0.221	1.000	+0.051^{ns}
	Few-shot	3.358	0.486	3.359	3.353	+0.006	1.000	+0.005^{ns}
Gemma-2-2B	Vanilla	3.155	0.495	3.165	3.099	+0.066	1.000	+0.049^{ns}
	CoT	3.350	0.618	3.377	3.289	+0.088	1.000	+0.066^{ns}
	Few-shot	3.031	0.421	3.032	3.026	+0.006	1.000	+0.005^{ns}
Mistral-7B	Vanilla	3.058	4.001	3.039	3.133	-0.094	1.000	-0.009^{ns}
	CoT	19.723	133.166	2.881	52.417	-49.536	1.000	-0.177*
	Few-shot	2.850	0.347	2.854	2.821	+0.033	1.000	+0.029^{ns}
Phi-3-mini	Vanilla	2.425	0.594	2.469	2.401	+0.068	1.000	+0.054^{ns}
	CoT	2.846	1.103	3.428	2.668	+0.760	1.000	+0.293***
	Few-shot	2.544	0.856	2.542	2.551	-0.009	1.000	-0.004^{ns}

Table 5: Arithmetic Reasoning Fidelity (ARF) on SVAMP. ARF is the Spearman rank correlation between per-item confidence ($1/\overline{\text{PPL}}$) and inverse operation-difficulty rank, where ranks are validated empirically against per-operation accuracy under vanilla prompting (Table 8, Appendix D). Positive ARF indicates confidence tracks empirically easier operations; negative ARF indicates confidence is miscalibrated relative to measured task difficulty.$^{***}p<0.001$, $^{*}p<0.05$,^{ns} not significant.

Model	Strategy	ARF	Interpretation
LLaMA-2-7B	Vanilla	-0.236^{***}	negative
	CoT	$+0.076^{*}$	near-zero
	Few-shot	-0.145^{***}	negative
Gemma-2-2B	Vanilla	-0.137^{***}	negative
	CoT	-0.113^{***}	negative
	Few-shot	-0.251^{***}	negative
Mistral-7B	Vanilla	$+0.027^{ns}$	near-zero
	CoT	-0.248^{***}	negative
	Few-shot	-0.004^{ns}	near-zero
Phi-3-mini	Vanilla	-0.215^{***}	negative
	CoT	-0.389^{***}	negative
	Few-shot	-0.069^{*}	near-zero

termine whether the gains reported here are stable across shot counts or sensitive to this specific choice.

Token generation cap. SVAMP few-shot outputs frequently reach the 150-token generation cap (72–100% of items depending on model), because the few-shot prompt format causes models to continue generating after producing the numeric answer. Since the answer token is produced within the first 5-10 generated tokens, accuracy is not materially affected (capped vs. uncapped accuracy difference $\leq$0.01 for three of four models), but future work should use a longer cap to avoid this confound entirely.

G Ethics Statement

This work studies the failure modes of existing open-weight language models using publicly available benchmarks (SVAMP, TriviaQA). No new data was collected, no human subjects were involved, and no personally identifiable information was processed.

Model outputs and harms. We evaluate models on arithmetic and factual-recall tasks. The outputs analyzed are numeric answers and short factual strings; they do not contain offensive, hateful, or privacy-sensitive content.

Dual-use considerations. Our diagnostic framework (HCS, ARF, RSS) is intended to help practitioners identify and mitigate confident-but-

Table 6: Primary accuracy metrics with 95% bootstrapped confidence intervals (2,000 resamples over evaluation items; seed = 42). AAA is SVAMP exact match (N=1,000); EM is TriviaQA exact match (N=200). Half-width reported; full interval is point ± half-width. Best value per model shown in **bold**.

Model	Strategy	AAA (%)		EM (%)	
		Point	± CI	Point	± CI
LLaMA-2-7B	Vanilla	6.7	1.6	58.0	7.0
	CoT	9.0	1.7	49.0	6.8
	Few-shot	**65.5**	3.0	**83.5**	5.0
Gemma-2-2B	Vanilla	3.9	1.2	**84.5**	5.0
	CoT	0.5	0.5	69.5	6.0
	Few-shot	**70.7**	2.8	83.0	5.0
Mistral-7B	Vanilla	3.2	1.0	80.0	5.3
	CoT	30.9	2.9	66.0	6.5
	Few-shot	**76.0**	2.7	**89.5**	4.3
Phi-3-mini	Vanilla	14.4	2.2	35.0	6.8
	CoT	70.0	2.9	23.5	5.8
	Few-shot	**87.7**	2.0	81.0	5.2

wrong model behavior. While it could in principle be used to understand how to induce specific failure modes, the primary utility is reliability improvement.

Environmental impact. Experiments were conducted on NVIDIA H100 GPUs. We used greedy decoding with a fixed inference budget (5 runs × 1,000 SVAMP + 200 TriviaQA items per model-strategy pair), keeping computational cost modest relative to large-scale runs.

Table 7: Pairwise Cross-Model Agreement (CMA) decomposed into correct-agreement (CMA_c: both models answer correctly on the same item) and wrong-agreement (CMA_w: both models produce the same wrong answer). Under vanilla prompting, agreement is driven almost entirely by CMA_w (shared heuristic failure); under few-shot prompting, by CMA_c (shared correct reasoning). **Bold** marks CMA_c values ≥ 0.60.

Strategy	Model pair	$\text{CMA}_{\text{total}}$	CMA_c (↑)	CMA_w (↓)
Vanilla	LLaMA-2-7B vs. Gemma-2-2B	0.191	0.017	0.174
	LLaMA-2-7B vs. Mistral-7B	0.083	0.003	0.080
	LLaMA-2-7B vs. Phi-3-mini	0.315	0.016	0.299
	Gemma-2-2B vs. Mistral-7B	0.330	0.003	0.327
	Gemma-2-2B vs. Phi-3-mini	0.141	0.012	0.129
	Mistral-7B vs. Phi-3-mini	0.114	0.001	0.113
CoT	LLaMA-2-7B vs. Gemma-2-2B	0.027	0.000	0.027
	LLaMA-2-7B vs. Mistral-7B	0.116	0.030	0.086
	LLaMA-2-7B vs. Phi-3-mini	0.078	0.054	0.024
	Gemma-2-2B vs. Mistral-7B	0.006	0.000	0.006
	Gemma-2-2B vs. Phi-3-mini	0.041	0.003	0.038
	Mistral-7B vs. Phi-3-mini	0.383	0.255	0.128
Few-shot	LLaMA-2-7B vs. Gemma-2-2B	0.612	**0.600**	0.012
	LLaMA-2-7B vs. Mistral-7B	0.642	**0.631**	0.011
	LLaMA-2-7B vs. Phi-3-mini	0.642	**0.638**	0.004
	Gemma-2-2B vs. Mistral-7B	0.682	**0.670**	0.012
	Gemma-2-2B vs. Phi-3-mini	0.691	**0.689**	0.002
	Mistral-7B vs. Phi-3-mini	0.743	**0.741**	0.002

Table 8: Per-operation mean accuracy under vanilla prompting, averaged across all four models (SVAMP, $N = 1{,}000$). Difficulty rank is the inverse of accuracy (rank 1 = easiest, rank 5 = hardest) and is used as the ground-truth ordering for ARF computation.

Operation	Mean accuracy (%)	Difficulty rank
Addition	10.2	1 (easiest)
Division	7.6	2
Multiplication	6.1	3
Addition+subtraction composite	4.2	4
Subtraction	4.1	5 (hardest)