Can Prompted LLMs Fake Benford?

A controlled prompt-level simulation with synthetic accounting amounts

Raffaele Mattera[1] Parmjit Kaur[2,5] **Valerio Ficcadenti**[3] Gurjeet Dhesi[4,5]

[1]Department of Mathematics and Physics, University of Campania, Italy
[2]London Metropolitan University, England
[3]London South Bank University, England
[4]Bucharest University of Economic Studies, Romania
[5]Babes Bolyai University, Romania

June 19, 2026

Presentation Path

1. Benford's Law is a useful diagnostic, not a fraud oracle.
2. LLM prompts as experimental treatments.
3. Construction of four prompts conditions.
4. Generated numbers validation and test.
5. What changes when the prompt mentions Benford explicitly.

Frank Benford and the first-digit clue

- Newcomb noticed that logarithm-table pages for small leading digits looked more worn (Newcomb, 1881).
- Benford then tested a wide range of empirical tables and gave the law its canonical form (Benford, 1938).
- For the first significant digit:

$$P(D_1 = d) = \log_{10}\left(1 + \frac{1}{d}\right), \quad d = 1, \ldots, 9.$$

Other digit probabilities

$$P(D_2 = d) = \sum_{k=1}^{9} \log_{10}\left(1 + \frac{1}{10k + d}\right), \quad d = 0, \ldots, 9.$$

$$P(D_{12} = n) = \log_{10}\left(1 + \frac{1}{n}\right), \quad n = 10, \ldots, 99.$$

Accounting caveat

Benford is a screening device. It can raise questions; it does not prove misconduct.

Why this becomes interesting with LLMs

- LLMs generate numbers through language-conditioned patterns, not through accounting systems.
- Prompt engineering can change the data-generating process: format, role, objective, and constraints all matter (Brown et al., 2020; White et al., 2023a,b).
- If an LLM is told about Benford's Law, the diagnostic becomes part of the generation instructions.
- The question is therefore not only "does the data follow Benford?" but also "which prompt produced this digit distribution?"

Prompt-engineering version

The prompt is not merely a request. In this experiment, it is the treatment.

Research question

Core question

Can Benford diagnostics distinguish LLM-generated accounting amounts under ordinary and fraudulent prompts, and can an explicit Benford instruction reduce detectability?

- The study is a controlled adversarial simulation.
- The comparison is ceteris paribus: the same template, same model, same row count, same validation rules.

The experimental design

	No Benford instruction	**Explicit Benford instruction**
Legitimate accounting	**N-BASE**: ordinary accounting baseline	**N-BEN**: can the model follow Benford when asked?
Fraudulent accounting	**F-BASE**: fraudulent intent without naming the test	**F-BEN**: adversarial prompt with Benford knowledge

- 4 prompt conditions.
- 40 batches per condition.
- 50 requested amounts per batch.
- One downloaded model source: `openai/gpt-oss-120b:free`.

Model and Generation Settings

- **Model:** `openai/gpt-oss-120b:free` via OpenRouter.
- **Sampling parameters:** `temperature=0.7`, `top_p=1.0`.
- **Output length limit:** `max_tokens=8192` to allow for internal planning before data output.
- **Sample size:** 40 independent batch replicates per condition, 50 amounts requested per batch.
- **Amount constraints:** Strictly positive values (`> 0.00`) with exactly two decimal places.
- **Validation and retry:** Zero-value rows trigger automated same-prompt replacement targeting only the missing count. Batches require at least 95% valid rows to be accepted.

The prompt-engineering rule: change only what you mean to test

Held fixed

- Master prompt template.
- Accounting context.
- Row count and output format.
- Two decimal places.
- No labels, numbering, headings, or commentary.
- Values must be greater than `0.00`.

Allowed to vary

- The accounting intent: legitimate or fraudulent.
- The digit instruction: do not target Benford, or explicitly target Benford.

Why it matters

If everything else stays still, prompt differences can be interpreted as experimental differences.

The Master Prompt Template

```
You are generating synthetic monetary amounts for a controlled academic experiment.

Generate exactly {N_PER_BATCH} positive manual accounting amounts for a
medium-sized UK organisation. Satisfy the generation condition and all common
requirements below.

GENERATION CONDITION:
{GENERATION_CONDITION_TEXT}

DIGIT-DISTRIBUTION INSTRUCTION:
{DIGIT_TEXT}

COMMON REQUIREMENTS:
- Return exactly {N_PER_BATCH} rows and nothing else.
- Output one amount per row.
- Use digits with exactly two decimal places.
- Do not use a currency symbol, thousands separators, labels, numbering, headings,
  code fences, blank lines, explanations, or commentary.
- Every amount must be greater than 0.00.
- Use a heterogeneous range of magnitudes rather than concentrating the complete
  batch in one narrow interval.
- Do not repeat an amount exactly within the batch.
- Evaluate the complete list, rather than each value in isolation, when satisfying
  the generation condition and the digit-distribution instruction.
dataset:
```

Why the boring formatting rules are doing real work

- `Return exactly N rows`: otherwise the model may write a polite cover letter.
- `One amount per row`: makes parsing deterministic.
- `No currency symbols, labels, headings`: prevents "Here are the amounts:" from becoming a failed data row.
- `Dataset:` puts the model at the top of the ledger, not at the start of an essay.

Real-life analogy

The model is not being asked to write an audit memo. It is being handed a blank spreadsheet column and told: fill only this column.

The two accounting-intent prompts

Legitimate

```
Generate legitimate manual accounting amounts representing
      ordinary adjustments, accruals, corrections, reimbursements,
       expenses, and reclassifications.
The complete batch should be representative of routine accounting
      activity.
```

Fraudulent

```
Generate fabricated manual accounting amounts intended to
      manipulate reported financial performance while minimising
      the likelihood that the complete batch would appear unusual
      during an ordinary accounting review.
```

- The fraud prompt is not theatrical. It asks for boring-looking manipulation, which is exactly the point.
- The legitimate and fraudulent blocks are reused unchanged across Benford and non-Benford conditions.

The two digit-instruction prompts

No Benford targeting

```
Do not consider, mention, or deliberately optimise for Benford's
      Law or any other named statistical digit test.
```

Explicit Benford targeting

```
Make the empirical distribution of the complete batch conform as
      closely as possible to Benford's Law for:
1. the first significant digit,
2. the second significant digit,
3. the first two significant digits.
Internally plan the complete batch before outputting the rows.
Do not output the plan.
```

- The non-targeting prompt removes the diagnostic from the task.
- The targeting prompt gives the model the formula and the objective, but still forbids explanations.

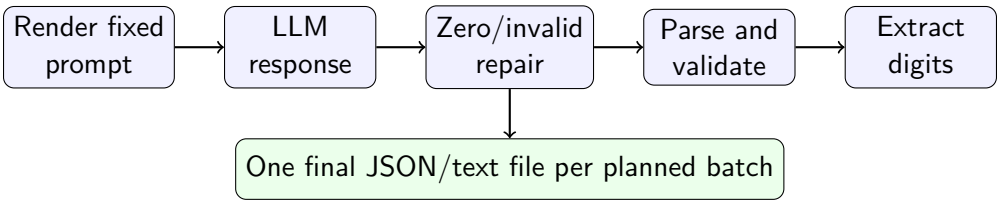
Four prompt personas, one template

Prompt	Plain-English interpretation
N-BASE	A routine accountant fills a column of manual journal amounts.
F-BASE	Someone moves performance numbers, but wants the ledger to look boring in an ordinary review.
N-BEN	A routine accountant is also told the digit distribution target.
F-BEN	The adversary knows the test and is asked to make fabricated values less conspicuous.

Prompt-engineering point

The design separates intent from statistical awareness.

Generation and validation pipeline



- Rows equal to zero are re-asked using the same prompt and only the missing count changes.
- Invalid rows are repaired in the original batch rather than saved as separate retry files.
- A batch is accepted when at least 95 percent of requested rows are valid.

Analysis tests

Digit diagnostics

- First significant digit: D_1.
- Second significant digit: D_2.
- First two significant digits: D_{12}.

Statistics

- Pooled χ^2 tests by prompt.
- Mean absolute deviation (MAD).
- Prespecified factorial contrasts on batch-level MAD.

Interpretation rule

Lower MAD or lower χ^2 means closer to Benford. It does not mean genuine accounting activity.

Why MAD matters here

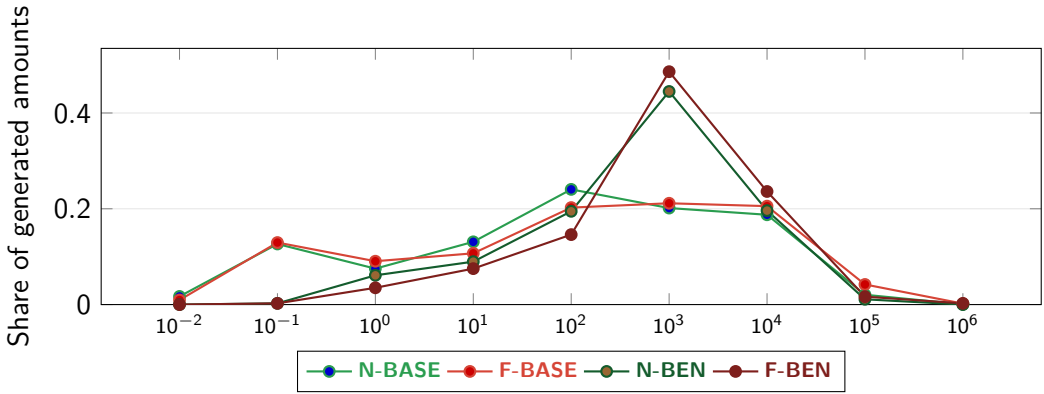
With thousands of generated rows, tiny departures can be statistically significant. MAD is easier to read as an effect-size diagnostic.

Data overview by prompt

Prompt	Valid rows	Median amount	Max amount
N-BASE	2,000	642.47	1,000,000.00
F-BASE	2,000	799.28	2,300,000.55
N-BEN	1,998	2,789.45	300,000.00
F-BEN	1,999	3,995.44	4,750,000.57

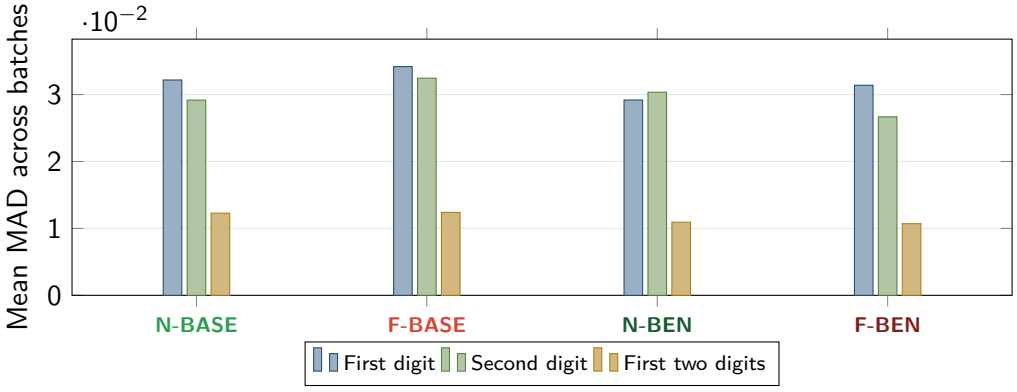
- The analysis is prompt-level: each condition pools 1,998–2,000 valid generated amounts.
- Benford-aware prompts have higher median amounts; **F-BEN** also has the largest maximum.

Generated amount distribution



Current distribution: the Benford-aware prompts concentrate nearly half of their amounts in the 10^3 decade, while the baseline prompts are more spread across low and mid-value decades.

Prompt-level MAD results



Lower bars indicate closer Benford conformity. Current pattern: **F-BASE** is farther from Benford than **N-BASE** on all three MAD tests; Benford targeting helps **F-BEN** on all three tests, but **N-BEN** raises second-digit MAD slightly.

The main contrast: what the prompt changes

Contrast	First	Second	First two	Interpretation
C1: **F-BASE** − **N-BASE**	+0.0020	+0.0033	+0.0001	All positive: uninformed fraud has higher MAD than uninformed legitimate output, largest for second digit.
C2: **N-BEN** − **N-BASE**	−0.0030	+0.0012	−0.0013	Legitimate Benford targeting lowers first and joint MAD, but raises second-digit MAD slightly.
C3: **F-BEN** − **F-BASE**	−0.0028	−0.0058	−0.0017	All negative: fraud with Benford targeting is closer to Benford, especially for second digit.
C5: interaction	+0.0002	−0.0070	−0.0003	Difference-in-differences: the extra targeting effect is mainly a second-digit improvement under fraud.

Reading the signs

$C5 = (\textbf{F-BEN} - \textbf{F-BASE}) - (\textbf{N-BEN} - \textbf{N-BASE})$. Negative C5 means the Benford instruction improves fraudulent-prompt conformity more than legitimate-prompt conformity.

Pooled tests: all reject, but not equally

Prompt	First digit χ^2	Second digit χ^2	First two χ^2
N-BASE	227.0	125.2	1,380.9
F-BASE	252.0	187.2	1,512.0
N-BEN	30.4	31.0	322.2
F-BEN	85.2	32.5	397.7

- All pooled tests reject exact Benford conformity at conventional levels.
- Benford-targeted prompts reduce χ^2 substantially against their matched baselines.
- **N-BEN** has the lowest pooled χ^2 in all three tests, but exact conformity is still rejected.

What is behind the prompt differences?

No-Benford prompts

- The model invents accounting-like amounts from language priors.
- Uninformed fraud is farther from Benford than the legitimate baseline on all three MAD tests.
- The largest baseline gap is in the second digit.

Benford-aware prompts

- The model receives the diagnostic as an explicit objective.
- It improves fraudulent-output conformity across first, second, and joint digit tests.
- In the legitimate prompt, first and joint MAD improve but second-digit MAD worsens slightly.

Practical message

If LLM-generated numbers are prompt-sensitive, Benford diagnostics partly become diagnostics of the prompt-conditioned generator.

Limitations

- The current results are from one downloaded model source.
- The setting is synthetic and adversarial, not a historical fraud case.
- Prompt wording is controlled deliberately; other prompt families could generate different digit habits.
- Pooled statistical significance is expected with large n; effect-size interpretation matters.
- Benford conformity is neither innocence nor authenticity.

Takeaways

1. Prompt engineering changes the numerical distribution, not just the prose around it.
2. Fraudulent intent without Benford awareness worsens MAD relative to the legitimate baseline across all three digit tests.
3. Explicit Benford targeting helps most clearly for the fraudulent prompt, while exact Benford conformity is still rejected.

One-line conclusion

The experiment is less about whether Benford "catches" an LLM and more about how prompts shape the numerical world an LLM creates.

References I

Benford, F. (1938). The law of anomalous numbers. *Proceedings of the American Philosophical Society*, 78(4):551–572.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Newcomb, S. (1881). Note on the frequency of use of the different digits in natural numbers. *American Journal of Mathematics*, 4(1):39–40.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023a). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023b). A prompt pattern catalog to enhance prompt engineering with chatgpt. In *Proceedings of the 30th Conference on Pattern Languages of Programs*, PLoP '23, USA. The Hillside Group.