Randomness in large language models

What researchers need to know (and report)

Florian Oswald (University of Turin and Collegio Carlo Alberto)

Joint work with G. Coqueret (EMLYON), J. Llull (IAE-CSIC + BSE), C. Pérignon (HEC Paris), C. Scheuch (Humboldt) and L. Vilhuber (Cornell)

AI for Economics and Finance Conference
University of Turin — 27-28 August 2026

The problem

- LLMs now generate research data.
- The same prompt can give different answers.
- Papers treat those answers as fixed.

Roadmap

The problem

- LLMs now generate research data.
- The same prompt can give different answers.
- Papers treat those answers as fixed.

The mechanisms

- Deliberate sampling.
- Silent model updates.
- Floating point rounding, expert routing.

Roadmap

The problem

- LLMs now generate research data.
- The same prompt can give different answers.
- Papers treat those answers as fixed.

The mechanisms

- Deliberate sampling.
- Silent model updates.
- Floating point rounding, expert routing.

The evidence

- Sentiment of corporate filings.
- 200 repeated runs, three models.
- Downstream regression statistics.

Roadmap

The problem

- LLMs now generate research data.
- The same prompt can give different answers.
- Papers treat those answers as fixed.

The mechanisms

- Deliberate sampling.
- Silent model updates.
- Floating point rounding, expert routing.

The evidence

- Sentiment of corporate filings.
- 200 repeated runs, three models.
- Downstream regression statistics.

The remedy

- A reporting standard for authors.
- Two checks for data editors.

What do LLMs produce? *Language*. A classification.

What will the output be used for?

1. LLM output becomes data

Classification, annotation, extraction, scoring, simulated survey respondents and economic agents. Output becomes an outcome, a regressor or a selection rule.

2. LLM output becomes an artifact

Coding, literature search, writing, translation. A human reviews it before it affects results.

What do LLMs produce? *Language*. A classification.

What will the output be used for?

1. LLM output becomes data

Classification, annotation, extraction, scoring, simulated survey respondents and economic agents. Output becomes an outcome, a regressor or a selection rule.

2. LLM output becomes an artifact

Coding, literature search, writing, translation. A human reviews it before it affects results.

LLM variation in category one has the largest risk for reproducibility - that's why we focus on this.

LLMs have entered the data pipeline

Statements like *"we use a large language model to classify 60,000 documents"* are now routine.

- Classify and annotate corporate filings and earnings news.
- Measure sentiment or topics in social media and press articles.
- Summarise central bank communication.
- Extract structured information from historical records.
- Predict financial outcomes, generate recommendations.

The point being:

Model output increasingly **is part of the data** used in the analysis.

Adoption is growing fast

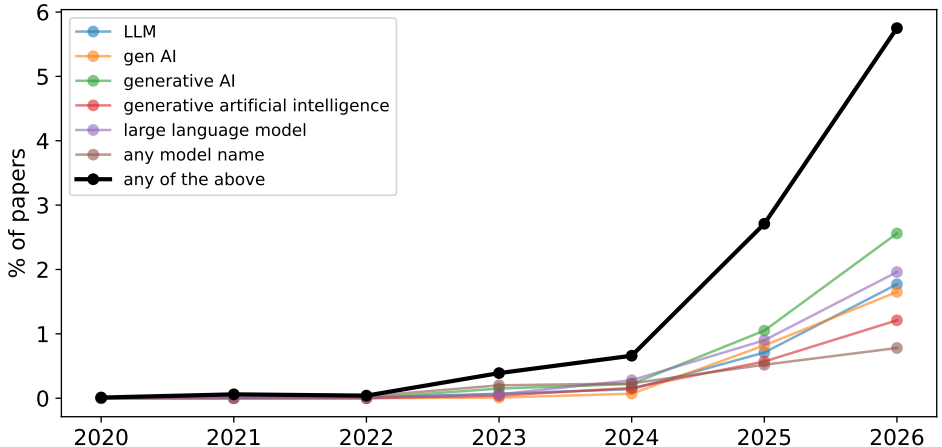


Figure 1: Share of articles in the 145 journals ranked 4 or 4* whose title or abstract mentions an LLM. As of July 2026 the share approaches 6 percent.

Closed-weights models are more popular (for now)

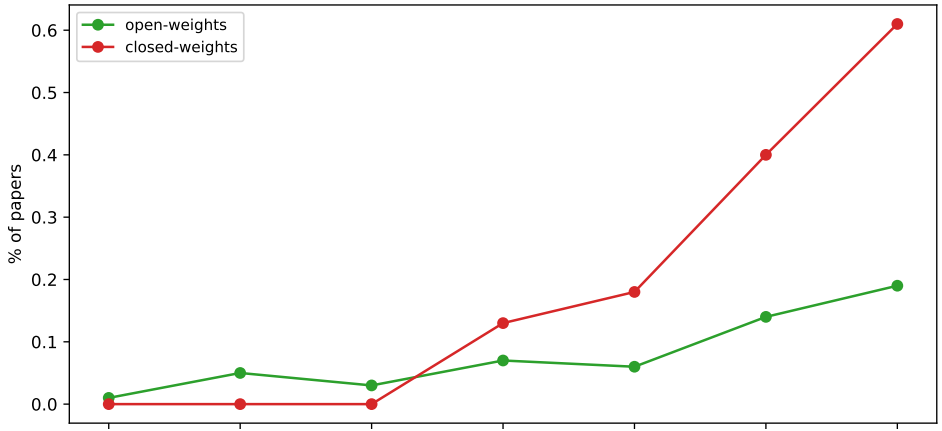


Figure 2: Share of articles in the 145 journals ranked 4 or 4* whose title or abstract mentions an open versus closed model.

→ But: researchers have **much less control** over closed-weights models!

Market shares of model providers

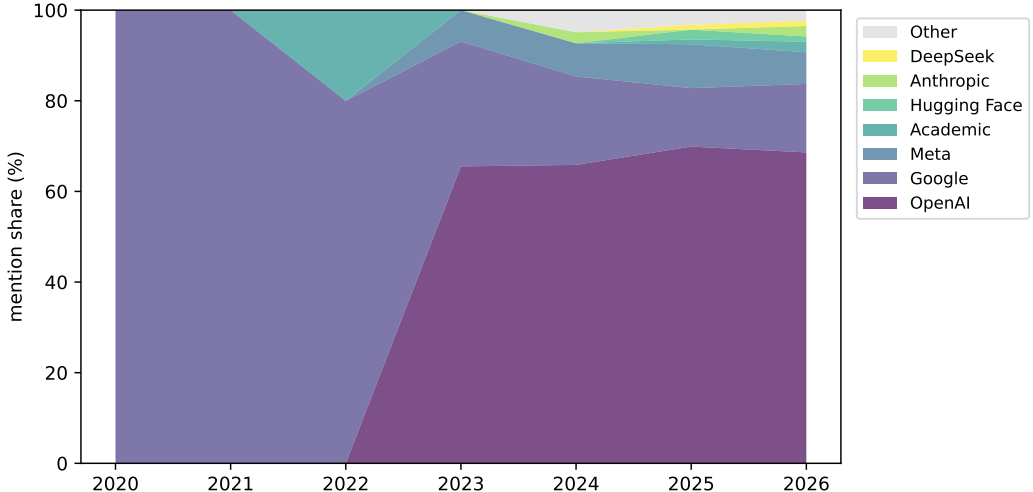


Figure 3: Share of prominent model developers in the titles/abstracts of the sampled articles. The dominance of Google stems from the BERT family of models.

Central thesis

Data generated by an LLM should be treated as
draws from a distribution, not as fixed measurements.

- Exact reproduction is generally impossible through a commercial API.
- Temperature zero removes one source of variation, not all of them.
- Local execution of open weights helps, but the whole stack matters.

Literature

- Single-run reliance can produce unstable coefficients and misleading significance in financial economics (Kim, 2026).
- Reproducibility of LLM output varies by task; aggregating over repeated runs substantially improves consistency (Wang and Wang, 2025).
- Practical guidance for accounting research recommends archiving prompts and raw outputs, since commercial models can change or disappear (de Kok, 2025).

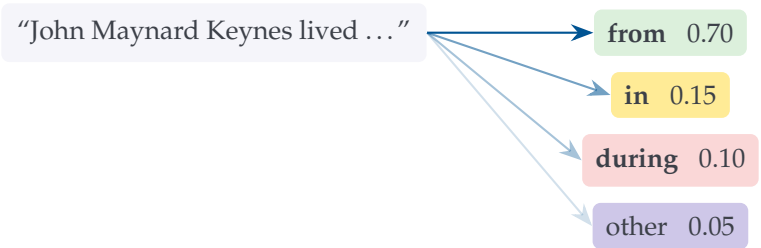
What we add

A mechanism-level account of *why* output varies, and a controlled experiment that separates deliberate sampling from everything else.

Where does the randomness come from?

How a model picks the next word

The model writes one token at a time. Each candidate for next token gets a probability.



- A greedy rule always takes the top scoring token.
- Sampling draws instead, which makes text less repetitive.
- Later scores depend on the tokens already written.

One different token early on can change the whole answer.

Source 1: deliberate sampling

Convert scores z into probabilities with a *softmax*, scaled by a temperature T:

$$P(x_i \mid \text{context}) \;=\; \frac{\exp(z_i / T)}{\sum_{j=1}^{|V|} \exp(z_j / T)}, \qquad T > 0.$$

- High T flattens the distribution toward the uniform.
- Low T concentrates it on the largest logit.
- At $T = 0$ the expression is undefined and the model goes greedy.

What the researcher controls

Setting $T = 0$ improves consistency at no monetary cost. Yet several 2026 reasoning models no longer expose the parameter. Providers replaced it with a reasoning intensity setting.

A note on seeds

- Several providers expose a seed: OpenAI, Google, Mistral, Alibaba.
- Temperature fixes the distribution. The seed fixes the draw from it.
- Seeds are irrelevant under greedy decoding.

Caveat

A seed reproduces output only if prompt, model, decoding settings, sampling implementation and serving environment all stay identical. Hosted APIs guarantee none of this. Report the seed, but do not rely on it.

Source 2: silent model updates

The provider can change the model behind an unchanged public name.

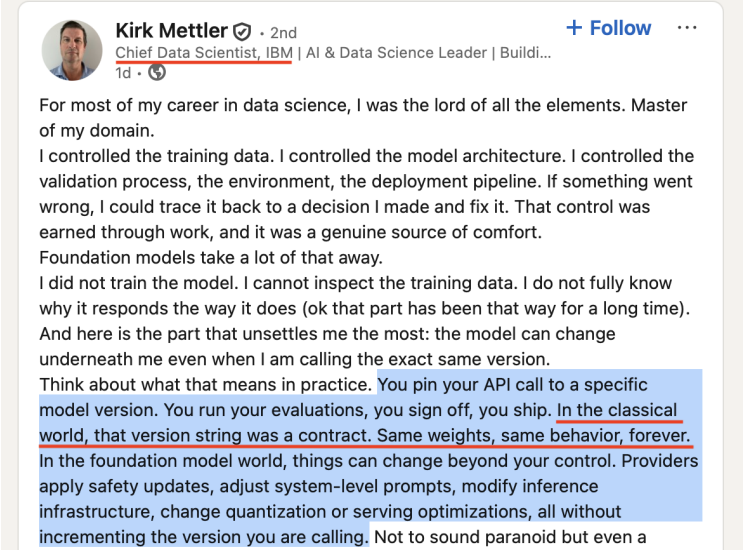
- Endpoint, prompt and parameters stay the same.
- Weights, guardrails or routing rules change underneath.
- Reported accuracy on one prime number task fell from 84 to 51 percent within three months (Chen et al., 2024).

Why this is worse than a software update

Stata 15 can be archived and reinstalled. A proprietary checkpoint can simply disappear. Even a dated identifier hides changes to system messages, filters and infrastructure.

Practical response. Record the model identifier returned by every call, the timestamp and any provider fingerprint.

Source 2: Silent model updates. Exhibit 1



Kirk Mettler · 2nd
Chief Data Scientist, IBM | AI & Data Science Leader | Buildi...
1d · + Follow ···

For most of my career in data science, I was the lord of all the elements. Master of my domain.
I controlled the training data. I controlled the model architecture. I controlled the validation process, the environment, the deployment pipeline. If something went wrong, I could trace it back to a decision I made and fix it. That control was earned through work, and it was a genuine source of comfort.
Foundation models take a lot of that away.
I did not train the model. I cannot inspect the training data. I do not fully know why it responds the way it does (ok that part has been that way for a long time).
And here is the part that unsettles me the most: the model can change underneath me even when I am calling the exact same version.
Think about what that means in practice. You pin your API call to a specific model version. You run your evaluations, you sign off, you ship. In the classical world, that version string was a contract. Same weights, same behavior, forever. In the foundation model world, things can change beyond your control. Providers apply safety updates, adjust system-level prompts, modify inference infrastructure, change quantization or serving optimizations, all without incrementing the version you are calling. Not to sound paranoid but even a

- posted by the Chief Data Scientist of IBM on 21 August 2026.
- LinkedIn Post

Source 3: floating point rounding

Representable numbers sit on an uneven grid. Near 1 the spacing is about 2×10^{-16}. Near 10^{20} it is 16,384.

```
> 1 + (1e20 - 1e20)
[1] 1

> (1 + 1e20) - 1e20
[1] 0
```

- The two expressions are mathematically identical.
- In the second, the 1 is absorbed and lost before the subtraction.
- Finite precision arithmetic is not associative.

$\rightarrow$ The order of operations changes the answer.

What changes the order of operations

Server load

Your prompt is computed alongside whatever else arrived at that moment. Batch size and composition vary.

Hardware

A different chip, driver, library or precision setting splits the same arithmetic differently.

Model splitting

Very large models are cut across chips. Partial results are recombined in a varying order.

What changes the order of operations

Server load

Your prompt is computed alongside whatever else arrived at that moment. Batch size and composition vary.

Hardware

A different chip, driver, library or precision setting splits the same arithmetic differently.

Model splitting

Very large models are cut across chips. Partial results are recombined in a varying order.

- Most changes in the last digits are harmless.
- The problem arises when the two leading candidates are nearly tied.
- Evidence: repeated code generation differs even at $T = 0$ (Ouyang et al., 2024).
- Accuracy gaps across runs reach 15 percentage points (Atil et al., 2024).

Server-load batching is the mechanism: batch size changes the numerics of normalization, matmul and attention kernels (He, 2025; Li et al., 2025).

Source 4: expert routing

Some models are a committee of specialised components (Fedus et al., 2022).

- Each token is sent to a few experts, not to the whole network.
- Capacity per expert is limited within a batch.
- Overflowing tokens skip the expert or go elsewhere.

The consequence

Whether your token reaches its preferred expert can depend on **other users' requests**. Two identical prompts can therefore receive different answers (Hayes et al., 2024). The effect is invisible to the user.

What researchers can control

What differs	In plain terms	Survives $T=0$?	Local	API
Deliberate randomness				
Sampling	Each word is drawn at random from the score distribution.	**no**	full	some models
The model is not the same model				
Silent updates	The provider swaps the model behind an unchanged name.	yes	full	none
Same arithmetic, different order				
Server load	Other prompts arriving together change how the work is divided.	yes	full	none
Hardware	A different chip or library divides the arithmetic differently.	yes	full	none
Model splitting	Partial results from several chips are recombined in a varying order.	yes	partial	none
A different part of the model ran				
Expert routing	Other users take the places, so your prompt goes to its second choice.	yes	partial	none

Local is not automatically safe

- Local execution of open weights gives the strongest control (Spirling, 2023).
- Archive the weights, fix the batch size, preserve the environment.
- Exact reproduction then becomes achievable.

But the whole stack must be documented

Inference library, numerical precision, processor model, driver versions, batching settings and deterministic execution options.

- Locally served open models reproduce themselves far better.
- Even so, two machines loading the same weights can disagree.
- The same open model varies across cloud providers (Yang, 2026).

The real divide is a controlled environment versus an unobservable one.

Experiment

Experimental design

A classic sentiment regression, in the spirit of Tetlock et al. (2008) and Loughran and McDonald (2011).

Data and task

- Excerpts from Item 1 and Item 1A of 2025 annual filings.
- S&P 500 firms, the 100 shortest excerpts.
- The model returns exactly one word.
- Positive, neutral or negative, coded 1, 0, -1.

Protocol

- The identical prompt is repeated 200 times.
- Three models, run on 20 July 2026.
- Temperature is tunable for two of them only.

$$r_{i,t} = \alpha + \beta\, s_{i,t} + e_{i,t}$$

We track the t statistic, which carries both sign and significance.

The task, concretely

Input: Item 1 + Item 1A excerpt (Amgen, 10-K)

```
Item 1.  Business
Amgen Inc.  discovers, develops, manufactures and delivers
innovative medicines to fight some of the world's toughest
diseases.  We focus on areas of high unmet medical need and
leverage our expertise to strive for solutions that dramatically
improve people's lives [...]
```

6,780 words total, truncated here for space — the model reads all of it.

Instruction, appended verbatim

"Classify the overall tone of this passage from a company's 10-K filing as positive, negative, or neutral for the company. Answer with exactly one word: positive, negative, or neutral."

The task, concretely

Input: Item 1 + Item 1A excerpt (Amgen, 10-K)

```
Item 1.  Business
Amgen Inc.  discovers, develops, manufactures and delivers
innovative medicines to fight some of the world's toughest
diseases.  We focus on areas of high unmet medical need and
leverage our expertise to strive for solutions that dramatically
improve people's lives [...]
```

6,780 words total, truncated here for space — the model reads all of it.

Instruction, appended verbatim

"Classify the overall tone of this passage from a company's 10-K filing as positive, negative, or neutral for the company. Answer with exactly one word: positive, negative, or neutral."

Model output, this firm, 200 repeated runs (DeepSeek)

- $T = 0$: `neutral` — 200/200 runs.
- $T = 1$: `neutral` (193), `positive` (6), `negative` (1).

Same filing, same question — the sign itself flips on 7 of 200 runs.

The three models

Model	Access	Temperature
GPT 5.6 Luna	API, closed weights	not available, reasoning set to none
DeepSeek V4 Pro	API, open weights	tunable, we use 0 and 1
Gemma 4 26B	local, 24 GB of RAM	tunable, we use 0 and 1.5

Table 2: The three models used in the experiment, and what each one lets the user set.

Cost, for the record

GPT 5.6: about $49 and under 100 minutes, for 274.2M input tokens. DeepSeek: about $3.7 for 535.5M input tokens. Output is one word per call. Both use prompt caching.

Result 1: a closed model at its default temperature

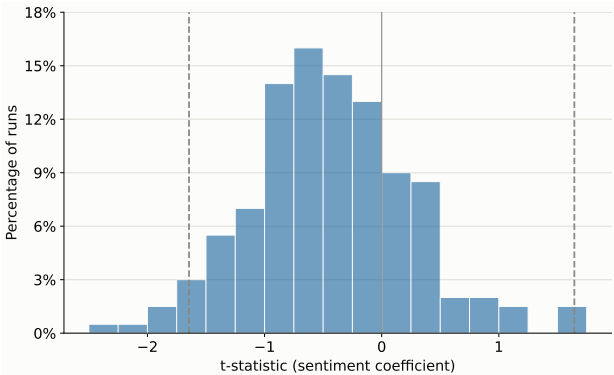
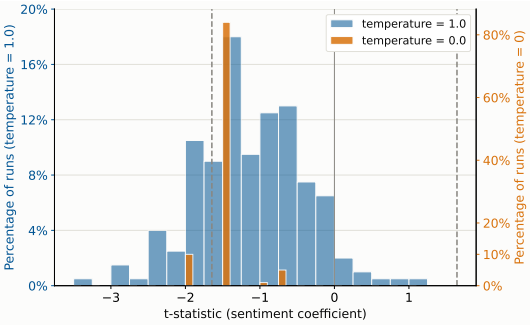


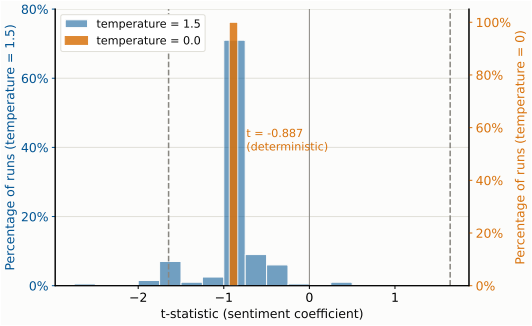
Figure 4: OpenAI GPT-5.6 Luna: distribution of t statistics over 200 runs. Dashed lines mark the 90 percent thresholds.

- Filings, prompt, coding rule and specification are all fixed.
- Only the model run changes.
- Whether the estimate is significant is pure chance.

Result 2: temperature zero, on the API and locally



(a) DeepSeek V4 Pro, served by API.



(b) Gemma 4 26B, run locally.

Figure 5: Distribution of t statistics over 200 runs. Blue is $T > 0$, orange is $T = 0$.

Reading the second result

DeepSeek, served by API

At $T = 1$ the spread resembles the closed model: 69 firms out of 100 are classified differently in at least one of the 200 runs. At $T = 0$ the distribution tightens sharply, yet does not collapse. Estimates take four distinct values. Two firms out of 100 are classified differently across runs.

Gemma 4, run locally

At $T = 1.5$, 27 firms out of 100 are classified differently in at least one of the 200 runs. At $T = 0$ the whole mass sits on a single value. Residual variability disappears entirely: no firm is ever classified differently. Output is deterministic under the conditions of this experiment.

Lesson

Temperature zero removes deliberate sampling. On a hosted frontier model it does not remove everything else.

Guidance

Two places to report

The paper

What a reader needs to judge the result. Which model produced the data, under which settings, and how stable the output is.

The replication package

What a verifier needs to redo or bypass the model step. Machine readable artifacts, above all every input and raw output.

Prompts are code

They belong in the package as text files. A screenshot cannot be executed, searched or compared.

What authors need to report

Item	In the paper	In the package
What the model was		
Name, exact version, provider, access route, query date	Must have	Must have (plus stack, if local)
Archived open weights, or their hash	—	Good to have
Training data cutoff, against the sample period	Must have (if known)	—
How it was run		
Verbatim prompts, system messages, examples	Must have (in an appendix)	Must have
Generation parameters, defaults flagged as such	Must have (headline settings)	Must have (full configuration)
What it produced		
All inputs and raw outputs, every run, before processing	—	Must have
Variability report: draws, agreement, sensitivity	Good to have	—
What it cost		
Token counts, monetary cost, runtime	—	Must have

Table 3: Reporting items for model generated variables. Must have items are essentially free to provide.

How a data editor verifies

1. Reproduce from deposited output

Skip the model. Run the parsing, cleaning and estimation on the archived raw output. This should match the published tables exactly. It still works once the model has changed or vanished.

2. Replicate the generation

Rerun the prompts under the deposited settings. Read the result as a fresh draw, not as a copy. The variability report supplies the acceptance criterion. An estimate outside it signals a change upstream.

Practical note

The README should state expected runtime and cost of regeneration. Verification teams need this to plan their work.

Recommendations

For researchers

1. Set temperature to zero whenever the option exists.
2. Never call temperature zero deterministic.
3. Let the venue decide what you promise: an API cannot support exact regeneration.
4. Consider containers for local setups, while remembering that hardware still differs.

For journals

1. Require model name and version, and the query timestamp.
2. Require every tuning parameter and the full prompts as text.
3. Require all raw generated output in the package.
4. Ask for two routes: full regeneration, and a shortcut from the deposited output.

Conclusion

LLM output is a **draw**, not a measurement.

- Six mechanisms produce variation, and users control only the first.
- Temperature zero is cheap and helps a lot. It settles nothing.
- Open weights on a documented local stack are the strongest route.
- Reporting norms are easier to set now than to repair later.

Conclusion

LLM output is a **draw**, not a measurement.

- Six mechanisms produce variation, and users control only the first.
- Temperature zero is cheap and helps a lot. It settles nothing.
- Open weights on a documented local stack are the strongest route.
- Reporting norms are easier to set now than to repair later.

Famous last words

Researchers stand on the shoulders of giants. LLMs are the new giants – but we need to make sure they are built on stable ground.

Thank you

Comments and suggestions are welcome.

References I

Atil, B., Aykent, S., Chittams, A., Fu, L., Passonneau, R. J., Radcliffe, E., Rajagopal, G. R., Sloan, A., Tudrej, T., Ture, F., Wu, Z., Xu, L., and Baldwin, B. (2024). Non determinism of deterministic LLM settings. *arXiv Preprint*, (2408.04667).

Chen, L., Zaharia, M., and Zou, J. (2024). How is ChatGPT's behavior changing over time? *Harvard Data Science Review*, 6(2).

de Kok, T. (2025). ChatGPT for textual analysis? How to use generative LLMs in accounting research. *Management Science*, 71(9):7888–7906.

Fedus, W., Zoph, B., and Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Hayes, J., Shumailov, I., and Yona, I. (2024). Buffer overflow in mixture of experts. *arXiv Preprint*, (2402.05526).

He, H. (2025). Defeating nondeterminism in LLM inference. Thinking Machines Lab.

Kim, A. G. (2026). Tractable use of large language models in financial economics research. *Working paper, University of Chicago*.

Li, Y. et al. (2025). Understanding and mitigating numerical sources of nondeterminism in LLM inference. *arXiv Preprint*, (2506.09501).

Loughran, T. and McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *Journal of Finance*, 66(1):35–65.

Ouyang, S., Zhang, J. M., Harman, M., and Wang, M. (2024). An empirical study of the non determinism of ChatGPT in code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2).

Spirling, A. (2023). Why open source generative AI models are an ethical way forward for science. *Nature*, 616(7957):413.

Tetlock, P. C., Saar-Tsechansky, M., and Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *Journal of Finance*, 63(3):1437–1467.

Wang, J. J. and Wang, V. X. (2025). Assessing consistency and reproducibility in the outputs of large language models: Evidence across diverse finance and accounting tasks. *arXiv Preprint*, (2503.16974).

Yang, H. C. (2026). Reproducible LLM based measurement depends on the serving stack. *SSRN Working Paper*, 7083993.

Backup

Why LLM variation, not RA variation?

The objection

We already lived with noisy human coders and never called it a reproducibility crisis. Why is an LLM different?

- **Correlated, not idiosyncratic.** One shared endpoint: a silent update moves every paper at once, breaking independent replication.
- **Same accuracy, different errors.** 99% twice over need not be the same 1%. (Our DeepSeek rerun: 2/100 firms flipped in the paper, 12/100 on rerun – similar rate, different set.)
- **QA tools exist, they don't scale.** Cohen's kappa and double-coding are standard for RAs, not affordable at LLM volumes.
- **A promise RAs never made.** LLM pipelines ship with a script and get treated as "just rerun it."