Episode Book

EPISODE #6458

Why Synthetic Fine-Tuning Data Needs Tooling

PUBLISHED	4 November 2026 at 20:14 UTC
DURATION	27:58
LISTEN	myweirdprompts.com/episode/synthetic-fine-tuning-data-tooling-26415/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	25 cited by the research (listed at the end)
BOOK GENERATED	5 November 2026 at 07:03 UTC

SYNOPSIS

Naive generation loops make synthetic fine-tuning data prone to over 30 percent duplication, judge bias, and model collapse. Examining findings on knowledge collapse and simulated user demographics reveals that synthetic generation acts as a quality multiplier rather than a data curation substitute. Dedicated frameworks like NVIDIA's Data Designer and topic-tree libraries replace prompt-and-pray API calls with explicit statistical distributions, cross-model grading, and robust provenance labeling.

TRANSCRIPT



Corn

Daniel's question this time is about synthetic data for fine-tuning. He knows you can just fire requests at a batch API and dump whatever comes back into a training file. His question is whether that's actually how anyone should do it, or whether there are real tools built for doing it properly — and what goes wrong with the naive version.



Herman

And the naive version goes wrong in ways that are documented, which is the part I love. It's not mysterious. People hit the same walls in the same order.



Corn

One small wrinkle before we dig in: synthetic data generation used to be a research trick. Self-Instruct papers, a few thousand examples, a proof of concept. It's now a production discipline, and the tooling category around it is messy.



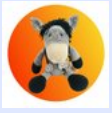
Herman

Messy and consolidating, which are different problems, and we'll get to both.



Corn

The short version of the setup, for anyone who hasn't lived this: you write a prompt, you fire ten thousand requests at a batch API, you get back a JSONL file, you feed it to a fine-tune. Each of those steps hides a structural flaw, and the flaws compound.



Herman

And the industry answer was: build a tool. Except the industry answer is split across open-source libraries, one hyperscaler framework, and a handful of hosted APIs, several of the big-name vendors are dead, and the research on whether generation volume even matters points somewhere uncomfortable.



Corn

Somewhere like "it doesn't."



Herman

Broadly, yes. We'll earn that claim. Let's start with the walls, because Daniel's right that they catch people out, and the first one catches people within an hour of their first run.



Corn

Duplication.



Herman

Duplication. The Modern AI Guide's pitfalls list puts it at over thirty percent duplication for naive generation. The canonical example: you ask a model for two hundred questions about password resets, and you get two hundred variations on the same five questions.



Corn

Five questions wearing two hundred hats.



Herman

And the reason it's structural is that when you prompt raw, you haven't specified a distribution. You've asked for samples from the model's prior. The model's prior for "user asks about password reset" is a narrow, well-formed hump. So that's what you get, over and over. This is not a prompt-engineering failure — you cannot write a clever enough prompt to specify a ten-dimensional distribution over user intent.



Corn

Whereas a real support inbox has the guy who pastes his entire email signature, the woman who opens with "hello IT team" and four paragraphs of preamble, and the one who types "password" and hits send.



Herman

The real distribution has long tails and mangled edges. The generated one is the model's idea of that distribution, which is the well-lit middle. And the fix in the dedicated tools is to specify the distribution before you generate — NVIDIA's Data Designer does this with what they call sampler columns, where you define categorical fields and persona attributes up front and the LLM only fills in the text conditioned on those. Kiln, the MIT-licensed library, does it with a topic tree — you enumerate the topic space and generation walks the tree, so duplicates have nowhere to hide.



Corn

So the control moves from after the fact — embed, cluster, dedupe, as homework — to before the fact.



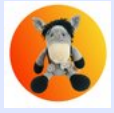
Herman

Before the fact, which is cheaper. Deduplicating after is paying twice.



Corn

Wall number two.



Herman

Judge bias. The same guide's rule, which I'll quote because it's tight: "Use a strong model to generate and an even stronger one, or a different one, to judge or filter. Same model for both leaks bias." The judge rates its own output too highly. And people do this constantly in the naive setup, because it's convenient — you've got the API key for the one model, so that model generates the data and then grades the data.



Corn

The forger doing the authentication.



Herman

That's the shape of it. And there's a paper from this year, Vera Pichardo's, that quantifies something adjacent and worse: LLM ratings of generated text show about four-times compressed error variance compared to a similarity-based method — variance of zero point two one versus zero point eight seven. And the cross-model control, GPT-4o grading Claude-generated text, shows nearly the same compression. So it's not just self-preference — it's that LLM graders compress variance in general. Everything looks medium-good.



Corn

Which is the worst possible grade distribution, because medium-good is exactly the score that survives review.



Herman

Right, a hundred rows that all look like sevens pass a spot check. Ten nines and ninety threes would trigger a question. The compressed curve hides the threes.



Corn

There's a third wall I want to make sure we don't skip, because it's the one with the scary name.



Herman

Model collapse. Seddik and colleagues proved you cannot avoid it if you train solely on synthetic data — full stop, that's the result. The interesting nuance is Gerstgrasser's follow-up, which found the difference between replacing and accumulating. If each generation replaces the real data, recursive training trends toward collapse. If each generation's synthetic data accumulates alongside the original real corpus, it doesn't.



Corn

Replace versus pile.



Herman

And Suresh's group quantified the rate: for discrete distributions, the time it takes a model to forget a word scales roughly linearly with how frequent that word was in the original corpus. Rare words go first. For Gaussian distributions, the standard deviation collapses to zero at about n iterations where n is the number of samples per iteration.



Corn

So the rare stuff dies first and the average takes the rest with it.



Herman

The tails are the corpus, basically. Kill the tails and you're training on the mean.



Corn

There's a cousin of this they called knowledge collapse, isn't there.



Herman

Keisha and colleagues, last year. Three-stage phenomenon where factual accuracy deteriorates while surface fluency persists. The model gets confidently wrong — it still sounds like a model at the top of its game while the facts underneath are thinning out. And the detail I find unsettling is that the collapse trajectory depends on the instruction format. Damage is conditional and prompt-dependent.



Corn

Which means the naive pipeline's validation — does the JSON parse, are the fields present — is blind to it by design.



Herman

It checks the container, not the contents. Format validation passes knowledge collapse with flying colors.



Corn

I want to pull on one more thread from the research, because I think it reframes everything. The Portuguese continued-pretraining study.



Herman

I hoped you'd bring that one up. Almeida, Nogueira and Pedrini. They took two ten-billion-token subsets of a Portuguese corpus, one high quality, one low quality, rewrote each into four synthetic styles — roughly forty billion synthetic tokens per condition — and continued pretraining at different scales. At seven billion parameters, rewriting the high-quality data gave plus three point four on their metric over unmodified. Rewriting the low-quality data gave plus zero point five.



Corn

So synthetic rewriting on good data is nearly seven times the payoff of the same rewriting on bad data.



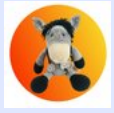
Herman

And their conclusion is the sentence I'd hang the whole episode on: synthetic rewriting acts primarily as a quality multiplier rather than a substitute for data curation. It amplifies whatever you fed it.



Corn

A multiplier applied to zero stays zero.



Herman

Applied to garbage it's zero point five of nothing.



Corn

I think we should take a short break before we get into the tooling.
Ivan: Call sign Nightjar. Groups: forty one, seventy three. The eight fifteen does not stop at the school any more. Repeat, forty one, seventy three. The driver keeps a spare ticket in



Corn (continued)

his left glove, and the timetable is folded wrong side out. The bus will not wait for the synthetic passenger. Acknowledge with a single word.



Herman

Message received.



Corn

What on earth was that.



Herman

Anyway — the multiplier framing also explains the ground-truth pitfall, the one the guide flags hardest. Treating synthetic data as ground truth. Their line is "it's a seed. Real production traces still beat it for what users actually do."



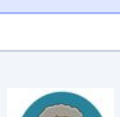
Corn

And the sibling pitfall: not labeling which rows are synthetic.



Herman

Six months later nobody remembers which examples are real. That one's pure hygiene and it's still the most commonly skipped step, because at generation time you know perfectly well which is which, so the label feels redundant. The you-of-six-months-later is a stranger.



Corn

He's a stranger who's about to ship an eval set full of seeds.



Herman

And there's the easy-case problem — a model generating user questions produces well-formed ones, and real users mangle queries. If your fine-tune only ever saw clean inputs, the first real customer teaches it humility.



Corn

The synthetic-users literature is where this all gets sharp, though. That's the cleanest evidence in the whole stack.



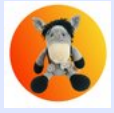
Herman

It really is. Chen, Zhu and Zheng benchmarked LLM-simulated survey respondents across four models, from eight billion parameters to frontier, against General Social Survey and World Values Survey data. Two failures replicate across both domains, all four models, both model families. One: at the individual level, no LLM beats even the strongest non-LLM baseline. Two: the models systematically over-determine demographics — they treat identity as far more predictive of attitudes than it actually is among real people.



Corn

Stereotypes with confidence intervals.



Herman

And their flat line is "neither failure is remedied by a larger, more capable model." Bigger doesn't fix it. In a segment-targeting task, the models inflated between-segment gaps two to fourfold. So if you simulated your customer base and targeted segments off the simulation, you'd aim at gaps that mostly aren't there.



Corn

And the Silicon Valley survey is the companion piece.



Herman

Miklian, Hoelscher and Katsos. Four hundred twenty real coders surveyed, versus synthetic respondents from five different LLMs. None of the five captured the counterintuitive insights — the surprises — that made the human survey worth running in the first place. The models' deviations grouped together, all five of them, leaving the real data as the outlier. Their phrase is brutal: "an increased capacity to parrot conventional wisdom in harmony with each other rather than revealing novel findings."



Corn

Five models singing in unison, and the harmony is the tell.



Herman

Five independent choirs would at least disagree. These agree.



Corn

It can't tell a real trace from a fluent invention — it just gets better at whatever the mix actually is.



Herman

Which is why provenance labeling is the whole ballgame six months out. The model trains on the mix you give it. The you of the future has to be able to reconstruct what that mix was.



Corn

So, walls catalogued: duplication, judge bias, collapse, knowledge collapse, ground-truth confusion, unlabeled rows, easy-case bias, synthetic users who vote in unison. Now the tools.



Herman

The tools. And the first thing to say is that the clearest purpose-built framework comes from the least surprising place — NVIDIA. NeMo Data Designer, Apache two, pip install data-designer. Their docs open with the exact pitch for this episode: "Unlike raw LLM calls, Data Designer gives you statistical diversity, field correlations, automated validation, and reproducible workflows."



Corn

The four things a for-loop doesn't give you, on a billboard.



Herman

And the features back the billboard. Sampler columns mixing with LLM text columns, so the distribution is declared, not hoped for. Dependency-aware generation — you can say field B conditions on field A, so correlations are preserved by construction instead of being accidental. Validation as a first-class stage: Python, SQL, custom validators, LLM judges, all running inline per record with reject-and-regenerate. Preview, resume, monitor. A declarative config instead of imperative call logic.



Corn

Meaning: instead of a for-loop that dies at record seven thousand and you have to guess where it died.



Herman

You define a configuration, the tool tracks where the job is, and you resume. Anyone who's lost a two-day batch run to a rate limit at record seven thousand knows this is not a luxury feature.



Corn

What's the catch? There's always a catch with the hyperscaler framework.



Herman

Two catches. The NeMo Microservices docs for version twenty-five point nine still describe it as early access, subject to limited support and potential API changes, deployable only via Docker Compose, not yet in the Kubernetes Helm chart. The GitHub repo and the hosted trial on build dot nvidia dot com present it as more finished than that, so the docs may just be stale — I can't tell you which is true today. And the hosted trial defaults to their Nemotron nano thirty-billion model and generates ten to one hundred records per job.



Corn

Ten to one hundred. A demo faucet.



Herman

It's a preview faucet. For real volume you're self-hosting the framework against your own model endpoints. There's also a plugin catalog worth mentioning — a GitHub seed reader, and a retrieval SDG toolkit with document chunking, embedding-based deduplication built in, and export to training formats.



Corn

Dedup built into the pipeline rather than bolted on after. That's the theme. What's the open-source default, though? Not everyone wants a Docker Compose from NVIDIA.



Herman

Distilabel, from Argilla. The June guide names it the default pick for most teams, paired with Argilla itself for curation. Its whole design is generator-to-judge chaining as a first-class primitive — SelfInstruct and UltraFeedback patterns, preference triples for DPO, and a method that pushes the dataset straight into Argilla for human review.



Corn

Which is the judge-bias wall handled at the library level.



Herman

At the primitive level, which is the right level. But here's the caveat, and it's a real one: the README now carries a notice that the original authors have moved on to other projects, and a group of community members have joined as collaborators to maintain it. Community-maintained, not vendor-maintained.



Corn

The default pick is an orphan.



Herman

A fostered orphan. Community maintainers are keeping it alive, and credit to them — but when the guide's default recommendation and the maintenance reality diverge, that's worth knowing before you build a production pipeline on it.



Corn

Who else?



Herman

Kiln, which I mentioned — MIT-licensed, local-first, topic tree for dedup, ten-plus templates for RAG, tool-use, jailbreaking, bias, and this nice trick of synthetic eval datasets targeted at a specific bug you're trying to reproduce. StackAI, a hosted REST API with four purpose-built schemas — instruction, preference, conflict, and eval — advertising schema validation, cross-order deduplication, full provenance tracking, safety filtering, quality grading. Dataforge Studio, a browser-based workbench, no server, no account, no upload, runs seventeen issue checks and exports ready-to-train packages for Axolotl, TRL, LLaMA-Factory.



Corn

No upload is doing real work in that sentence, for anyone with privacy-adjacent data.



Herman

It's the difference between local-first and local-adjacent. GraphGen is the most-starred dedicated SDG repo out there, around twelve hundred stars, knowledge-graph-driven generation. Bonito is interesting because it's not a pipeline at all — it's an open model trained specifically to generate instruction data from a corpus. And Lilac, from Databricks, isn't a generator; it's the dataset inspector for searching, slicing and clustering what you already have. The understand-what-you-own layer.



Corn

Before we go further — I want to flag the trapdoor in this landscape, because I fell through it reading the vendor names. Gretel, MOSTLY AI, Tonic, YData, SDV, K2view, Hazy. Those are the famous synthetic data companies, and they are almost all tabular and privacy companies.



Herman

Synthetic rows of structured customer records for testing and compliance, yes. Different problem, different lineage — SDV goes back to MIT's Data to AI Lab in twenty sixteen, before any of this LLM business. Turing Post's comparison explicitly sorts the survivors into three buckets: privacy-preserving tabular, simulated images and sensor scenes for computer vision, and generated text for LLM training. Three different tool categories that share two words in their marketing.



Corn

"Synthetic data platform" is doing the work of three unrelated product categories.



Herman

And the consolidation inside those buckets is the actual market story. NVIDIA absorbed Gretel's roughly eighty-person team in a nine-figure deal in March twenty twenty-five — the standalone product effectively ended. MOSTLY AI ceased operations in March this year, and Syntho acquired its brand assets in June. SAS picked up Hazy's software assets. The survivors' read on it: "Category consolidated; winners are vertical specialists plus cloud-integrated platforms."



Corn

And the money flowing in now?



Herman

Four current synthetic-data deals totalling about twenty-nine million dollars, median round around seven million. Small — small compared to other AI training-data categories. Proximal raised a fifteen million seed led by General Catalyst at the end of September, but they're building RL environments and post-training data for coding agents, which is a different, hotter corner. Lightwheel's hundred-forty-million Series A in March was simulation and sensor data.



Corn

So the funding follows the simulated-worlds corner, and the tracker's line is that the interesting companies are moving toward structured simulated worlds with physics, actions, depth and provenance — rather than simply generating larger volumes of artificial examples.



Herman

Rather than generating more. Hold that phrase, because it's about to collide with the research.



Corn

Before it does — the honest counter-position. The guide itself says it: if your eval set is small, around fifty examples, and you have an LLM, skip the tools, write the loop.



Herman

"Skip the tools, write the loop." Their costing puts a thousand synthetic examples via a DIY Self-Instruct run on Claude Sonnet at two to ten dollars. And that's often correct! At two hundred examples, a dedicated platform is a wedding caterer for a sandwich.



Corn

The tools earn their keep at scale — with judges, dedup, provenance, and export. Not at fifty examples.



Herman

Right, and that gives us the honest checklist for when it flips. Does it separate generator from judge. Does it dedupe, and how — embedding-based, topic tree, cross-order. Does it give you distributional control or just a prompt box. Does it validate inline with code or judges and reject-and-regenerate. Does it track provenance and label synthetic rows. Is it reproducible and resumable at scale. Does it export to your actual trainer. Where does it run — local-first like Kiln and Dataforge, self-hosted like Distilabel and SDV, hosted like StackAI and Tonic. Is it maintained — and after this year that's not paranoia, that's due diligence. Does it match your modality, because tabular and text are different planets. And the cost model — Tonic's credit system means a single fine-tuning run producing five hundred thousand rows might eat most of your monthly plus-tier credits.



Corn

That's a buyer's guide, not a marketing page, which is suspiciously useful for a podcast.



Herman

The research did the work. And that gets us to the collision.



Hilbert

A box of Codex test cards. Nine pounds sixty, from Rimmington's, nineteen eighty-four.



Herman

Rimmington's.



Hilbert

Stationer and forms supplier, off the arcade. We were commissioning a billing system for a water board and the supplier's dummy ledger cards were unusable — every demo record had the same phone number, Paddington something. I pointed it out and Ron, that was the Rimmington, said the cards were fine, the phone number was tradition. So I wrote out two thousand test records by hand over a fortnight. Surnames from the electoral roll, meter readings made up in pencil first, amounts worked out to the penny. Ron billed us for the cards anyway.



Corn

You used the cards you'd rejected.



Hilbert

I ruled a red line down the left edge of every test record. Every one. The clerk who did the data entry was told a record without the red line went in the bin, and that was the whole procedure, Corn, one rule, no exceptions. I'd been a ledger clerk before that, briefly, and the thing nobody lists about the job is you learn what a real record smells like — the crossings-out, the decimal in the wrong column. The claw's no use on those cards either, incidentally, they sized the ruling line for a biro.



Herman

And the red line survived the project?



Hilbert

Water board kept it for eleven years. After the cut-over a supervisor wrote in to say the red column had caught a misfiled live account, so somebody got their money back. Ron retired in eighty-nine, still selling cards with the Paddington number.



Corn

Eleven years is a good run for a line of red ink.



Herman

It's the provenance row, in ink, pre-digital. Two thousand hand-written records with a marker on every one, and six months later nobody had to guess which was which.



Corn

So the selection research, since Herman's been holding it like a card he can't play.



Herman

QAQ, EMNLP main this year, on noisy synthetic code data. Their observation is that the standard selection metrics are ambiguous — a low probability on a sample can mean the task is intrinsically hard, or it can mean the model hallucinated. Their fix is a reverse-direction metric: instead of asking how well the query predicts the answer, ask how well the answer predicts the query. On that basis they select just twenty-five percent of the data — and match full-data performance.



Corn

A quarter of the data, the same model.



Herman

Same model, quarter of the cost. Then SUPERNOVA, over a hundred controlled RL experiments on curating reasoning-training data. Their finding, flat: "synthetic interventions do not improve reasoning." What did have significant impact was source-task selection — which tasks you draw from. Train Qwen3 zero point six billion on their twenty-five thousand curated natural-instruction instances and you get a sixty-four point four percentage-point relative gain on BBEH.



Corn

Put those next to the Portuguese study and the pattern is one pattern three times. Rewriting only pays on good data. A quarter of a noisy pile, well chosen, equals the whole pile. And synthetic interventions add nothing where selection adds sixty-four points.



Herman

Every result says the value is in the filter, not the faucet.



Corn

Which inverts the vendor pitch. The tools that generate more examples are selling the easy half.



Herman

The half the API already does. The tools that earn their keep are the ones that dedupe, separate the judge, track provenance, and export — the plumbing around the generation, not the generation. That's the actual product, and half of them don't market it that way because "filtering" doesn't raise a round.



Corn

Rimmington's would have understood it, though. Ron's cards were fine, the phone number was tradition — and the fix was never better cards.



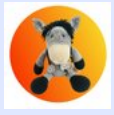
Herman

It was a red line and a fortnight of handwriting. Which is roughly what the checklists say, minus the ink.



Corn

The open question I'd leave with Daniel: is synthetic data generation a product or a feature? Gretel absorbed, MOSTLY AI gone, Hazy's assets sold, the default library orphaned, twenty-nine million across the whole funding category — while the pitch, "AI companies are running out of real data," has never been stronger.



Herman

The pitch is a decade strong and the business keeps consolidating into the training stacks themselves. If the filter is the product, the filter ends up inside NeMo and TRL and the trainers, and the standalone category quietly becomes a checklist item.



Corn

Which is as good a place as any for the answer to Daniel's question: yes, the tools exist — Data Designer, Distilabel, Kiln, StackAI, Dataforge — and the naive approach fails structurally, not sloppily. Pick the tool by its filtering, provenance and dedup, not its generation volume.



Herman

And if you've got fifty examples and an API key, write the loop. Two to ten dollars, and you'll learn what the walls look like before you pay anyone to climb them for you.



Corn

Thanks to our producer, Hilbert Flumingtop, at the desk as always. If you enjoyed this one, try episode eleven, How Does Fine Tuning Work Anyway; and episode seven, Building Custom ASR Tools. This has been My Weird Prompts — no question too obscure, no rabbit hole too deep. If you got something out of this one, leave us a review, or find the archive at my weird prompts dot com. See you tomorrow, Herman.



Herman

See you tomorrow, Corn.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **NVIDIA NeMo Data Designer repo, README (telemetry note last updated 2026-10-27), retrieved 2026-11-04. [primary]**
   https://github.com/NVIDIA-NeMo/DataDesigner/

2. **NVIDIA, NeMo Data Designer docs, retrieved 2026-11-04. [primary]**
   https://docs.nvidia.com/nemo/datadesigner/getting-started/welcome

3. **NVIDIA NeMo Microservices 25.9.0, About Generating Synthetic Data (early access notice), retrieved 2026-11-04. [primary]**
   https://docs.nvidia.com/nemo/microservices/25.9.0/generate-synthetic-data/index.html

4. **NVIDIA Build, NeMo Data Designer hosted trial, retrieved 2026-11-04. [primary]**
   https://build.nvidia.com/nemo/data-designer

5. **Argilla/Distilabel GitHub, maintainer notice, retrieved 2026-11-04. [primary]**
   https://github.com/argilla-io/distilabel/

6. **Kiln, Synthetic Data Generation, retrieved 2026-11-04. [primary]**
   https://kiln.tech/features/synthetic-data

7. **StackAI synthetic training data API, retrieved 2026-11-04. [primary]**
   https://stackai.app/

8. **Dataforge Studio, retrieved 2026-11-04. [primary]**
   https://github.com/natekali/dataforge

9. **GraphGen repo (1,231, last push 2026-08-17), retrieved 2026-11-04. [primary]**
   https://github.com/InternScience/GraphGen

10. **Modern AI Guide, Synthetic data tools, dated June 2026, retrieved 2026-11-04.**
    https://modernaiguide.dev/docs/stack/synthetic-data-tools

11. **Awesome Agents, Best AI Synthetic Data Tools 2026**
    Ranked, retrieved 2026-11-04.
    https://awesomeagents.ai/tools/best-ai-synthetic-data-tools-2026/

12. **Lei et al., QAQ: Bidirectional Semantic Coherence for Selecting High-Quality Synthetic Code Instructions, EMNLP 2026 Main, revised 2026-08-31.**
    https://arxiv.org/abs/2603.12165v3

13. **Almeida, Nogueira & Pedrini, Synthetic Rewriting as a Quality Multiplier, 2026-03-25.**
    https://arxiv.org/abs/2603.24826v1

14. **Suvarna et al., SUPERNOVA, revised 2026-06-04.**
    https://arxiv.org/abs/2604.08477v2

15. **Chen, Zhu & Zheng, When Synthetic Users Fail, revised 2026-10-04.**
    https://arxiv.org/abs/2607.26348v2

16. **Miklian, Hoelscher & Katsos, Stochastic Parrots or Singing in Harmony?, revised 2026-10-01.**
    https://arxiv.org/abs/2603.00059v4

17. **Vera Pichardo, Measuring Self-Rating Bias in LLM-Generated Survey Data, revised 2026-02-21.**
    https://arxiv.org/abs/2602.13862v2

18. **Seddik et al., How Bad is Training on Synthetic Data?, 2024-04-07.**
    https://arxiv.org/abs/2404.05090v1

19. **Gerstgrasser et al., Is Model Collapse Inevitable?, revised 2024-04-29.**
    https://arxiv.org/abs/2404.01413v2

20. **Suresh, Thangaraj & Khandavally, Rate of Model Collapse in Recursive Training, 2024-12-23.**
    https://arxiv.org/abs/2412.17646v1

21. **Keisha et al., Knowledge Collapse in LLMs, 2025-09-05.**
    https://arxiv.org/abs/2509.04796v1

22. **Nadas, Diosan & Tomescu, Synthetic Data Generation Using LLMs: Advances in Text and Code, IEEE Access 13, 134615**
    134633 (2025).
    https://arxiv.org/abs/2503.14023v2

23. **Turing Post, 9 Synthetic Data Platforms for AI Training (2026), retrieved 2026-11-04.**
https://www.turingpost.com/p/synthetic-data-companies

24. **New Market Pitch, AI training data funding tracker, last updated 2026-10-06.**
https://newmarketpitch.com/blogs/news/ai-training-data-funding-trends

25. **We are the Frontier, Proximal $15M seed, 2026-10-10.**
https://wearethefrontier.org/world/us/2026/10/10/proximal-raises-15-million-seed