Episode Book

EPISODE #6585

Four Primitives Behind Every Pairwise Eval Tool

PUBLISHED	14 November 2026 at 09:25 UTC
DURATION	22:25
LISTEN	myweirdprompts.com/episode/pairwise-evaluation-harness-primitives-55129/
HOSTS	Corn, Herman, Vex, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	17 cited by the research (listed at the end)
BOOK GENERATED	15 November 2026 at 11:16 UTC

SYNOPSIS

Audio listening tests and LLM arenas rarely share code or citations, yet both reduce to the same four primitives: the pair queue, blinding and order, the vote record, and the standings. Comparing tools like BeaqleJS, listen-and-rate, and orq-arena reveals where the two worlds diverge. LLM harnesses judge pairs in both orders to discard inconsistent votes and control for response length, much like loudness matching in audio. Meanwhile, audio tools export raw CSVs, leaving Bradley-Terry ratings and confidence intervals to external packages.

TRANSCRIPT



Corn

Last time we settled that comparing things two at a time is often the only comparison that still discriminates. Daniel's follow-up is the practical one: fine, you're playing clip A against clip B blind and voting which one you'd rather listen to — now stop rebuilding the machinery every time you want to do that.



Herman

He's doing exactly what we talked about, by the way. TTS clips, blind playback, and a four-way vote: A better, B better, both bad, or about the same. Then he asks what the primitives are — the pair queue, blinding, randomised left/right order, the vote record, the standings — and which open-source tools already hand you those, both for audio listening tests and for model-versus-model LLM work.



Corn

He wants the parts list before he buys the machine.



Herman

The parts list is the episode, then. The claim I want to open with is this: a pairwise harness is four primitives, and almost every tool in this space is just a different set of choices about those four. What the pair is. How blind it is. What you write down. How you rank from it.



Corn

Two worlds, one vocabulary.



Herman

That's the strange part. The audio world — BeaqleJS, listen-and-rate, MOSI, TTS-Eval, SpeechSubjectiveTest — those are browser stimulus players that write CSVs or Excel files. The LLM world — orq-arena, LocalArena, llmarena — those are prompt-and-response pipelines with standings attached. They don't share a library. They barely share a citation. But the primitives are recognisably the same four, and the interesting disagreements are about two of them: what a tie actually is, and what blinding actually means.



Corn

Then let's take the four in order, because the tools only make sense once you know what they're supplying. Start with the queue.



Herman

The queue is the boring primitive and the one that quietly decides your experiment. In BeaqleJS, the unit is a testset — a named object with an ID and a files map pointing at your A and B clips, X as well if you're doing ABX. And the whole queue lives in one config: RandomizeTestOrder true, MaxTestsPerRun minus one to run everything, or a number to sample a subset per run. That's the queue and the sampling knob in about six lines.



Corn

Twelve years old, that project, and people still cite it.



Herman

Still the reference implementation. Which tells you something about eval tooling generally — the config schema outlives the code. listen-and-rate is the modern replacement, and it derives the queue from your directory structure instead: one directory per system, identical filenames across directories, and anything missing from any directory is dropped with a warning. Your filesystem becomes the experiment definition.



Corn

I like that, actually. No hidden state, no database — you can see the whole design in a folder listing.



Herman

It also means the design is diffable. Change one system's output, drop the new files in a new directory, and a directory diff tells you exactly what changed between runs. You can't do that with a config blob sitting in some web app's backend.



Corn

And if a clip is missing from one system's folder?



Herman

That pair is dropped, warning logged. Which sounds like a nuisance until you've debugged a test where half the trials were silently comparing a clip against nothing — then an explicit drop with a message is a kindness.



Corn

Fair. Continue.



Herman

It also has the knob I'd point Daniel at for his exact case: items per session. Set it to one in a two-system experiment and each listener evaluates a single pair. One person, one vote, straight down. For a solo tester doing TTS clips, that's the honest shape of it.



Corn

One listener, one vote. My kind of workload.



Herman

Yours and also literally yours, in this scenario. Right — second primitive, blinding and left/right order. Here the two worlds split immediately. In audio, blinding is a config flag. BeaqleJS has a ShowFileIDs setting whose documentation reads, and I love this, "never enable this during real test!"



Corn

The exclamation point is doing peer review.



Herman

listen-and-rate is more forthright than most. Its policy statement says the browser never receives system names, file paths, or which stimulus matches the hidden reference — but, quote, "preventing a determined user from inspecting browser memory or network traffic is outside the scope of this tool." Which is admirably honest. Blinding there is for honest participants. It's a courtesy, not a boundary.



Corn

Which is fine for a listening test with volunteers. Less fine when the listener and the experimenter are the same man, and that man built the TTS models he's ranking.



Herman

That's the question nobody's tool answers, and I want to park it, because Hilbert might have views. Actually, no, let me hold that thought entirely. On left/right order: listen-and-rate randomises trial order and stimulus order independently for each listener, and then says outright that it does not use a Latin square or any counterbalanced design across sessions — if you need guaranteed counterbalancing, you prepare the assignment yourself.



Corn

An honest gap, stated in the docs. Rarer than it should be.



Herman

Then you cross into the LLM world and orq-arena treats order as a measurable thing rather than a nuisance. Every pair is judged blind in both orders by default. There's a flag to judge each pair once — it halves your judge calls, and the documentation says what that costs you: position bias goes unmeasured.



Corn

So the flag isn't "faster," it's "less honest, cheaper."



Herman

Then the discipline on top: a judge that changes its vote when only the order changed is discarded for that round. And if too few trustworthy votes remain, the round counts as inconclusive rather than a coin flip. That's the piece no audio tool has. Nobody runs the human listener twice with sides flipped and throws out the inconsistent ones.



Corn

Which is either a statement that humans are assumed consistent, or a statement that nobody built it, and I don't know which is worse.



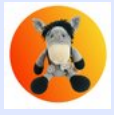
Herman

I don't either, honestly. But as a design, swap-and-discard turns position bias from something you hope you avoided into something you measured. That's a real difference in kind. And there's a practical objection worth airing — with a human listener, doubling every trial doubles the session length, and listeners get fatigued, which is its own bias. A judge model doesn't get tired at trial four hundred. So the LLM world can afford a discipline that the audio world would have to buy with listener patience.



Corn

Meaning the reason nobody built it for humans might just be that nobody could afford it.



Herman

That's a cleaner explanation than the charitable one, yes. Third primitive. The vote record.



Herman

listen-and-rate writes one CSV per session — flat, keyed by session ID and item ID, into a results folder. BeaqleJS is friendlier for a distributed test: it can POST JSON results to a web service, or email them to a supervisor. Then orq-arena again goes furthest: every judged round lands in a battles file with both responses, each judge's vote, exact token counts, per-response timing.



Corn

The boring detail there is the one that matters — storing the stimuli.



Herman

Because it makes the record re-analysable. You re-judge with a different panel and it costs you judge tokens only. You don't re-run the models. The expensive part of an arena run isn't the judging, it's generating the answers, and once those are on disk the whole round is frozen and re-countable. A vote record that only stores the vote is a receipt; one that stores the stimuli is an experiment.



Corn

And the audio tools?



Herman

CSVs and Excel files, mostly. Fine for the vote, thin on the stimulus side. Though in fairness to the audio tools — the stimulus in audio is a file on disk already. The clip itself doesn't vanish; what's missing is the linkage, the record that says vote thirty-seven was cast on these two files at this path. You can reconstruct it if you kept your directory layout, and you can't if you didn't.



Corn

So the audio version of the same principle is: never rename anything, ever.



Herman

Which is, incidentally, exactly what Hilbert does with a shoebox, but we'll get there. Now, fourth primitive, and this is where it gets interesting — so the pair queue and blinding are really two separate primitives, which is where most of the confusion starts.



Corn

Wait.



Corn

That's Vex again. How does he keep getting through screening? We changed the number twice.



Herman

Parking meters are not judging anything, Vex. A meter has two slots because a coin is round and a card is flat, and the revenue is a fee, not a score.



Corn

Moving on before he gets to the printouts.



Corn

You were saying — the standings.



Herman

The standings. orq-arena fits Bradley-Terry — the model behind chess-style ratings — over every judged round, with bootstrapped ninety-five percent confidence intervals. The maths is part three of Daniel's series, so I'll only say what the harness does with it: when a run is too small to separate the top two, the report says exactly that, with the measured gap, instead of calling it a tie or quietly picking a winner.



Corn

The tool admitting it doesn't know. Almost nobody does that.



Herman

It also does the thing that made me sit up: a length-controlled rating. If the judging panel shows a preference for longer answers, it identifies that and reports a rating with the length preference controlled out.



Corn

The LLM analogue of loudness matching.



Herman

That's exactly the frame, yes. Anybody who's done audio comparison knows the louder clip wins if you don't level-match — there's even a young desktop app, onyx, that lists loudness matching alongside its blind ABX mode, because solo A/B tests get invalidated by level more than by anything else. Two fields that don't read each other's papers arrived independently at the same principle: neutralise the confound the voter can't ignore before the vote means anything.



Corn

And the confound is always the one that's cheap to produce. Turning the volume up costs the vendor nothing. Padding an answer with a friendly preamble and a recap costs the model nothing either.



Herman

Right — the confound isn't an accident, it's the cheapest available signal of quality. Which is why it has to be controlled, not merely noted.

V

Vex

Listen to me, every parking meter on this street is a pairwise listener and nobody notices! Each meter has two slots, A and B, and the little card reader logs which side you fed first, that is your vote record right there, randomized left and right by the city so you never realize you are being counterbalanced!

V

Vex

And the standings? The standings are the ticket revenue, that is the Bradley Terry score, and the queue is just the cars waiting behind you! I have the receipts, I have the printouts, I have been standing at this meter for three hours proving it!

V

Vex

So look into the meters, all of them, today, and tell your listeners the truth before the next quarter drops!



Corn

Output format?



Herman

One self-contained HTML file per run. Verdict banner naming the top three, the ladder with error bars, quality versus cost, the win grid, per-judge behaviour. Compare that to the audio side, where the standings are simply left to you. BeaqleJS's docs, on preference tests, suggest — and I quote the hedging — "one might use a sign test." Might.



Corn

"One might." That's a documentation author refusing to be cited.



Herman

listen-and-rate ships Plotly summary statistics, nice plots in the browser, but no ratings with intervals. No audio tool ships Bradley-Terry or Elo at all. If you want standings with error bars from listening-test votes, the pattern is: export the CSVs, fit them with something like bayesian-pairwise-ranking — that one implements Bradley-Terry, Davidson and Plackett-Luce in PyMC.



Corn

Glue two tools together and you've built the thing the first tool should have shipped.



Herman

Which is the recurring finding across this whole survey. Now here's where I want to slow down, because the tie is the whole ballgame and the tools disagree. Daniel's vote set is four options: A better, B better, both bad, about the same. Plain Bradley-Terry — what orq-arena, llmarena and LocalArena all fit — has no native tie parameter. Davidson and Plackett-Luce do handle ties.



Corn

Hold on. "Both bad" and "about the same" are not the same vote.



Herman

They are not. "About the same" is an indifference — the two things are level. "Both bad" is a joint rejection — they're level and the level is low. One is a comparison verdict, the other is a quality verdict smuggled into a comparison. And I'll tell you what I found: no surveyed tool implements that four-way set as a named mode. listen-and-rate's AB test is a binary preference. Its CMOS test puts a degree of preference on a minus-three to plus-three scale. orq-arena's judges answer one A/B/tie question.



Corn

So the most interesting modelling decision in Daniel's whole setup lives inside a UI button no tool has.



Herman

A joint rejection carries real information — it says the pair isn't worth choosing between, which is different data from a coin flip between them. Feed "both bad" into Bradley-Terry as a tie and you've laundered a rejection into an indifference. The model can't see the difference, because the button didn't either.



Corn

And it matters downstream, too. A system that loses every comparison is a bad system. A system that draws "both bad" against everything might be a bad system — or might be in a field of bad systems, and you can't tell which from the tie column.



Herman

The rejection vote is only interpretable against some absolute anchor, and the pairwise world deliberately has none. That's the price of the method.



Corn

That's the part I'd want Daniel to sit with. His four-way vote is better than what any of the tools offer, and he'll have to defend it himself.



Herman

The rest of the model-versus-model landscape, quickly, because it's worth knowing the shapes. LocalArena runs the complete model-by-task grid concurrently, records failures as data, derives order-invariant ratings with task-clustered confidence intervals, and — this is its stated rule — never substitutes recorded answers or copied leaderboard results for a live call. llmarena is the closest open reimplementation of the LMArena shape: side-by-side chat UI, Elo over Bradley-Terry. And there's Chatbot Arena itself as the scale reference — over eight hundred thousand human pairwise comparisons collected.



Corn

Eight hundred thousand votes and people still argue about the top three.



Herman

Because the error bars are wide at the top — more votes narrow them, they don't abolish them. Then the odd one: P2L, Prompt-to-Leaderboard. It trains a model that takes a natural-language prompt and outputs a vector of Bradley-Terry coefficients — a leaderboard per prompt. Their whole motivation is that aggregate scores average away the variation across users and prompts.



Corn

Which connects to something we've said before — we did an episode on benchmarking custom speech tools beyond the word error rate, and the argument was the same shape: the single aggregate score is the thing that's lying to you.



Herman

Same disease, different organ. A model that's best on Hebrew navigation prompts and worst on English small talk has one Elo number, and the one number hides both facts.



Corn

Then the general harnesses. lm-evaluation-harness, Inspect, promptfoo — the things people already have installed.



Herman

And this is the misconception worth killing, because I've watched people assume it. lm-evaluation-harness is pointwise — absolute scores per task, sixty-odd academic benchmarks, the backend for the Open LLM Leaderboard. Superb at what it does. It gives you the model abstraction and the task registry. It does not give you a pair queue and it does not give you standings.



Corn

promptfoo?



Herman

Side-by-side, assertion-based grading. You see both outputs and you grade them against declared expectations — which is fine, and it's not blind. There's no anonymous vote, there's no ladder.



Corn

Inspect is the one I'd guess people reach for.



Herman

It gives you the scorer-and-judge plumbing and a nice log viewer, and pairwise comparison is something you express as a scorer — you build the tournament yourself. The pattern across all three: the harnesses you already have stop exactly where pairwise begins. The pair queue, the blind vote, the standings — that's the part you're writing by hand, or pulling one of the arena tools in for.



Herman

...though half of them get thrown out anyway, because whoever's voting can hear the relay.



Hilbert

That's the switcher problem, Herman, and you've got it backwards. I built a switcher box for speaker cables, early seventies, randomiser and everything. Wired it myself, sealed the labels. Threw out a whole afternoon of votes because I could hear the relay. Corn will appreciate the rest — the votes are in a shoebox at home, labelled by date. You never know when a bad afternoon becomes data.



Corn

Labelled by date. Bad afternoons, filed.



Hilbert

The shop teacher's habit, Corn. You date everything you discard, or you stop knowing what you threw away.



Corn

He's right about the direction, though. We've been talking about blinding as stopping the listener knowing which cable is which. His problem was the harness itself telling him.



Herman

That's the sharper version — the harness becoming the tell. Every tool we discussed today blinds the stimuli and then trusts the playback. BeaqleJS hides the file IDs, listen-and-rate never sends the names, and both of them will happily play your two clips through whatever different-sounding path your browser picks.



Hilbert

The connector that clicks louder. Same thing. I mark my votes with a pencil, Corn. Pencils don't lie about the order.



Corn

A pencil is a config flag you can trust, that's fair.



Herman

The swap-consistency check is the audio world's version of his shoebox, in a way — orq-arena keeps the discarded judge votes and treats them as data on the judge. No listening-test tool does that for humans. It should.



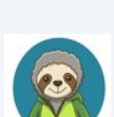
Corn

The misconception to leave people with, then. Two of them, actually, so pick one.



Herman

Blinding. People think blinding is a security boundary — that the tool is stopping anyone from knowing. In the audio tools it's a courtesy flag for honest participants, and only orq-arena treats knowing as something to detect and measure. If your listener wants to know, config will not save you.



Corn

Mine's the harnesses — that lm-evaluation-harness or Inspect gives you pairwise comparison out of the box. They're pointwise and scorer-based, and they stop exactly where the pair queue starts. Everything we discussed today is the gap between what's



Corn (continued)

installed and what's needed.



Herman

Eight hundred thousand human comparisons, and the tie button still isn't standardised anywhere. That's the thought I'd leave open — if the tools can't agree what a tie is, what exactly are the leaderboards ranking?



Corn

And the fact that two fields found the same confound independently — loudness, length — without ever citing each other suggests the primitives are more universal than the tools will ever be.



Herman

Producer credit to Hilbert Flumingtop, who has apparently been running his own evaluations since before it was a field. If you enjoyed this one, a review wherever you get the show helps more than you'd think. If you enjoyed this one, try episode nine, Benchmarking Custom ASR Tools - Beyond The WER; episode seven, Building Custom ASR Tools; and episode eight, Building Your Own Whisper. This has been My Weird Prompts. We'll see you soon.

SOURCES

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

1. **sp-nitech, Listen and Rate, BSD 3-Clause, retrieved 2026-11-14 [primary]**
https://github.com/sp-nitech/listen-and-rate

2. **HSU-ANT, BeaqleJS, GPLv3, LAC 2014 paper referenced, retrieved 2026-11-14 [primary]**
https://github.com/HSU-ANT/beaqlejs

3. **orq-ai, orq-arena, retrieved 2026-11-14 [primary]**
https://github.com/orq-ai/orq-arena

4. **EleutherAI, changelog entries 2026/09, 2025/12, 2025/07, 2025/03, 2025/02, retrieved 2026-11-14 [primary]**
https://github.com/EleutherAI/lm-evaluation-harness

5. **UK AI Security Institute, Inspect, retrieved 2026-11-14 [primary]**
https://github.com/UKGovernmentBEIS/inspect_ai

6. **promptfoo, MIT, 27,154, last push 2026-11-14, retrieved 2026-11-14 [primary]**
https://github.com/promptfoo/promptfoo

7. **cadia-lvl, MOSI, retrieved 2026-11-14 [primary]**
https://github.com/cadia-lvl/MOSI

8. **MU94W, TTS-Eval, retrieved 2026-11-14 [primary]**
https://github.com/MU94W/TTS-Eval

9. **bigpon, SpeechSubjectiveTest, retrieved 2026-11-14 [primary]**
https://github.com/bigpon/speechsubjectivetest/

10. **jing1ei, onyx, Rust, created 2026-09-05, last push 2026-11-09, retrieved 2026-11-14 [primary]**
https://github.com/jing1ei/onyx

11. **llmarena, retrieved 2026-11-14 [primary]**
https://github.com/llmarena/llmarena

12. **Arena-OSS, Prompt-to-Leaderboard, retrieved 2026-11-14 [primary]**
https://github.com/Arena-OSS/p2l

13. **MIT, created 2026-07-02, retrieved 2026-11-14 [primary]**
https://github.com/kojikojikojikoji/bayesian-pairwise-ranking

14. **AgentWebPro, MIT, created 2026-03-11, retrieved 2026-11-14 [primary]**
https://github.com/AgentWebPro/gtm-arena

15. **LocalArena npm package page, retrieved 2026-11-14 [primary]**
https://www.npmjs.com/package/localarena

16. **mirror repo describing LMSYS Chatbot Arena (800,000+ comparisons), retrieved 2026-11-14**
https://github.com/olimiemma/LLM-Chatbot-Leaderboard

17. **From Uncertain Judgments to Calibrated Rankings: Conformal Elo Estimation for LLM Evaluation, retrieved 2026-11-14**
https://arxiv.org/html/2606.13221v1