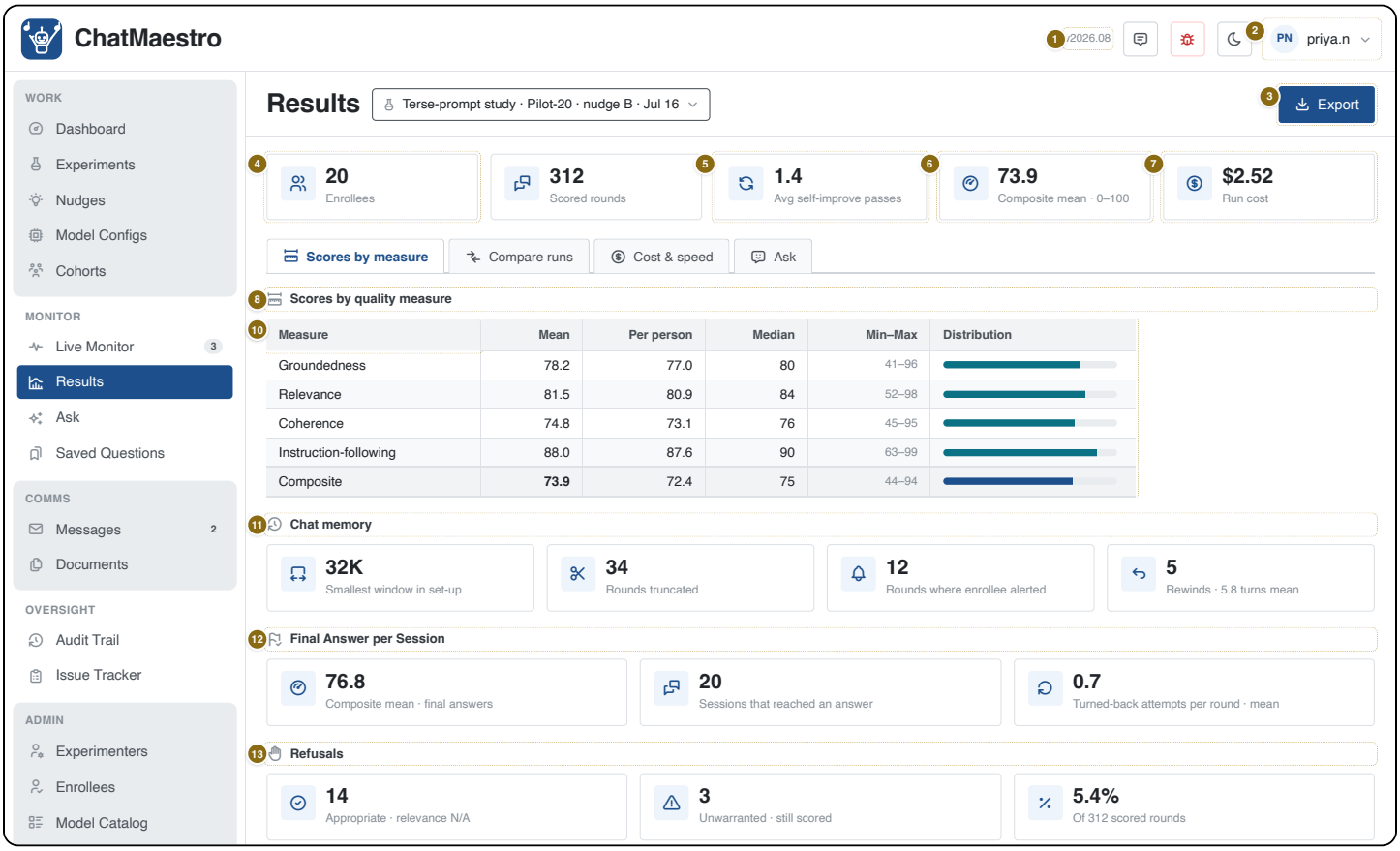
# Results

`results-analytics.html · 1360×820`




## Annotations

1. **Version pill (right-aligned)** Dimmed build pill sits on the RIGHT, beside the account menu. Hover reveals the build sha; on phones it folds into the account menu.
2. **Account menu** Avatar + name + chevron open the account menu (profile, settings, sign out). On narrow screens the feedback / bug / dark-mode icons and version pill collapse into here.
3. Download these results as CSV, Excel, or JSON.
4. **Counts** `n_enrollees` and `n_rounds` — enrollees and scored rounds for this run.
5. How many times, on average, a model revised its own answer to make it better.
6. One overall 0–100 score that blends the quality measures this study's rubric names.
7. **Cost split** Run cost = **generation** tokens (the doer models) + **judge** tokens (the trusted judge), tracked separately.
8. **Uniform judge** One uniform judge scores every run identically (rubric v3) — that's what makes runs comparable. Every row here reads from `run_result`. Scored by one judge that grades every run the same way, so runs compare fairly.

9 **How the columns aggregate** Every measure is **enrollee-unit** — the mean of each enrollee's own mean, so a chatty enrollee cannot dominate, and there is no round-weighted variant. `composite_mean` and `composite_median` are the two fixed headline columns; each rubric factor plus the composite also carries a five-number summary {min, q1, median, q3, max} in `factor_stats`. The factor set comes from the experiment's rubric, so it is configurable rather than fixed. Spread is the interquartile range; standard deviation is deliberately not stored.

10 Each answer is graded on the qualities this study's rubric names. The built-in default asks four: is it backed by the sources, on-topic, clear, and does it follow the instructions.

11 **Memory depth is bounded by the smallest model** Read from `run_result.history_stats`. When a study has `enable_chat_history` on, the replayed conversation is fitted into the largest window every model in the set-up shares — the **smallest** `model_catalog.context_window` among its members — so all of them receive identical input. **This is why the figures matter for a comparison:** `model_config` is a swept variable, so one arm can have a smaller smallest-window than another and therefore systematically shallower memory. Read alone that looks like a quality difference; read here it is a condition difference. Truncated rounds dropped their oldest turns and carried on — the session is never stopped. Alerted rounds are those where the enrollee was warned and advised to start fresh; that is recorded because a warned enrollee may write differently, and it lands only on long sessions (ALGORITHMS §23). How much of the conversation the models could remember, and where that ran out.

12 **Why a second headline** `run_result.final_answer_stats` scores each session's **last round** instead of every round. It exists because the last answer is what the enrollee actually walked away with, and a long session can drift a long way from where it started. Read it **beside** the composite mean, never instead of it: the all-rounds figure still says what the whole session cost and scored, and this one says where it ended up. A session whose rounds were all rewound contributes nothing here and shows in the gap between the two counts (ALGORITHMS §6). The score of the last real answer in each session — the one the enrollee walked away with.

13 **Refusals are scored differently** A response that declines the request rather than attempting it is recorded in `chat_round.declined`. Refusing and failing look identical to the scorer otherwise: relevance asks whether the answer addressed the request, so a refusal scores badly on it while scoring well on instruction-following, and a model that correctly refuses would rank below one that answers anyway. An **appropriate** refusal therefore takes relevance to N/A and the composite is the weighted mean of whichever factors applied; an **unwarranted** one keeps relevance scored, so declining is never a way to dodge a low score. These figures read from `run_result.declined_dist`, the frozen tally over `appropriate`, `unwarranted` and `attempted` (the three sum to `n_rounds`) — rolled up once when the run finishes, so the refusal rate survives a cleanup of the rounds it was computed from. Turned-back requests are **not** counted here: the sufficiency gate stops those before a round exists (`chat_round.clarification`), while this records a refusal by the models under test. How often the assistant refused to answer, and whether refusing was the right call.