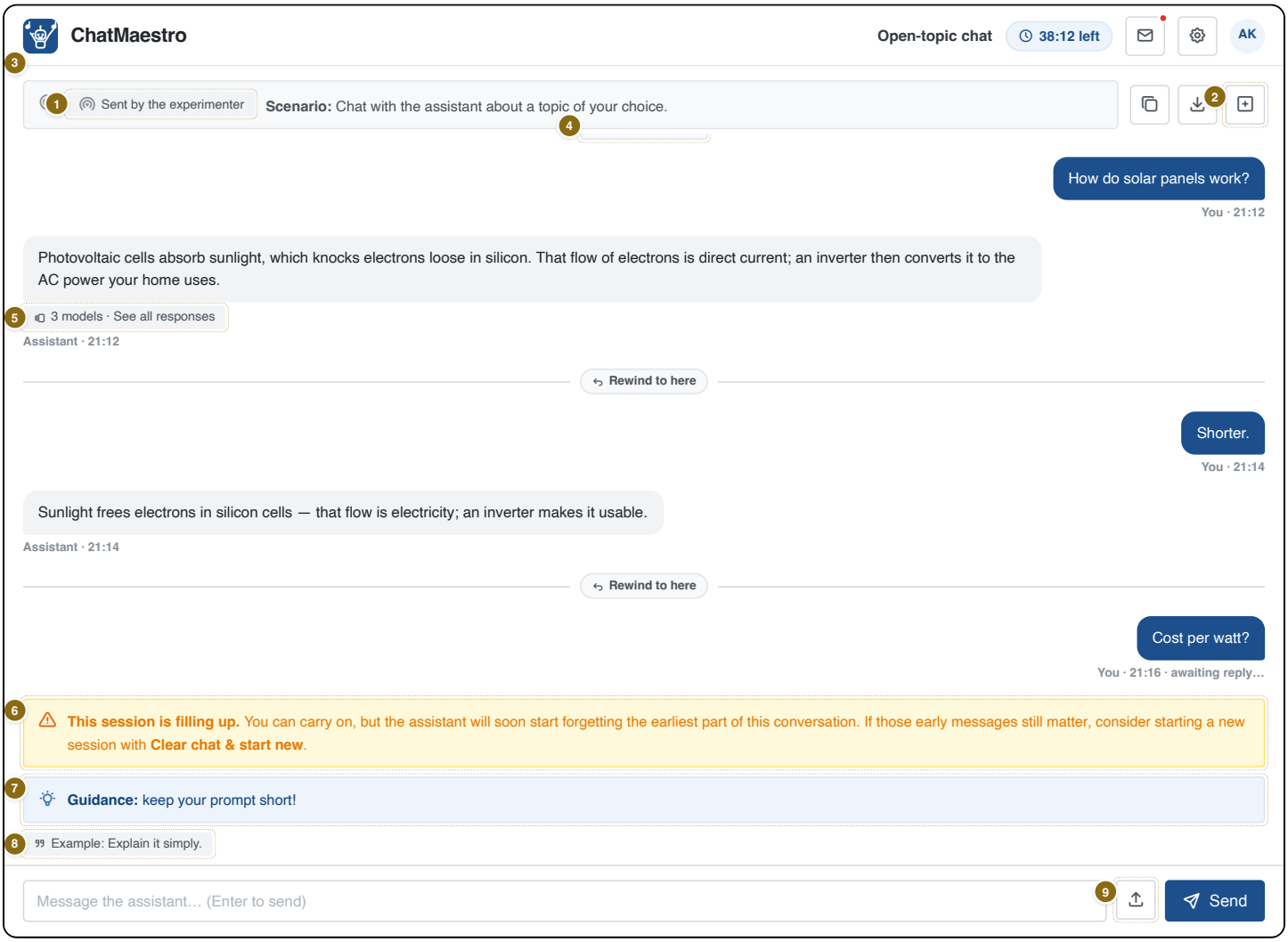
Enrollee view (chat)

`enrollee-view.html · 1100×800`



Annotations

1. **Pushed live** This scenario was pushed live to the enrollee by the experimenter — it appears without the enrollee navigating to it.

2. **Clear chat & start new** Always available — it is a chat convenience, not a blinding control, so no experiment setting gates it. It increments `chat_round.session_seq`, so history assembles from empty and the assistant remembers nothing of what came before. Nothing is deleted: the earlier rounds stay in the transcript and still count in the run's statistics. It is the blunter sibling of **Rewind**, which discards only the tail of a session (see `ALGORITHMS §23`). Start a fresh conversation. The assistant forgets what was said before; your earlier messages stay in the transcript.

3. **Turned back, not counted — worded per the run's Socratic switch** The sufficiency gate turned the message back instead of answering it, because it did not carry enough to answer. How that reads is the run's `experiment_run.socratic_enabled`: this run has it **on**, so the judge asks for **one missing piece** and the answers accumulate into the request. With it **off** the same turn-back names everything missing and asks for one complete rewrite — here it would read “Before I can answer I need to know which part of home solar you mean: how the panels work, what a system costs, or how much power one produces. Please send the full question again with that included.” — and only that final request reaches the models. Either way no `chat_round` is written, the models under test never see the message, and the attempt is kept in `chat_round.clarification` and counted in `retry_count` on the round it eventually leads to — so the run's round count, cost per round and score distribution all describe answered questions. There is no cap on attempts (ALGORITHMS §21, §30). Your message needed a little more before it could be answered, so this reply says what is missing — one piece at a time, or all of it at once, depending on the study. It does not count as one of the study's exchanges.

4. **Rewind** Steps the conversation back to this point: every later round of the session is stamped with `chat_round.rewound_at` and stops being replayed to the models, while everything before it is kept. That is what separates it from **Clear chat & start new**, which discards the whole conversation by incrementing `session_seq`. Nothing is deleted — a rewound round stays in the transcript, keeps its scores, and still counts in `n_rounds`, because an enrollee rewinds when an exchange went badly and dropping it would flatter the run (see `ALGORITHMS §23`). Go back to this point in the conversation. The assistant forgets everything after it; what came before is kept.

5. **See-all is enabled** Shown because the experimenter enabled See-all for this study — opens each model's individual answer for this exchange. This study sends your prompt to 3 models — open this to read each model's own answer.

6. **Approaching the memory limit** Shown at `CONTEXT_ALERT_THRESHOLD` (default **0.80**, overridable in `.env`) of the usable budget. The budget comes from the **smallest** `model_catalog.context_window` among the set-up's members, so every model receives identical input. It **recommends** rather than requires a fresh session, because only the enrollee knows whether the early context still matters — past the limit the oldest turns simply drop and the conversation continues. That the alert fired is recorded in `chat_round.history_window.alert_shown`: a warned enrollee may write more tersely or wrap up, and that lands only on long sessions, which is itself an outcome (see `ALGORITHMS §23`). This conversation is getting long. You can keep going — but the assistant will start forgetting the earliest parts.

7. **Guidance, not a prompt** A directive from the experimenter about HOW to prompt — it is guidance to follow, not text that gets sent. An instruction from the experimenter on how to write your prompt — follow it, don't type it in as your message.

8. **Example prompt** A sample prompt the experimenter may supply. It is shown as an example and never inserted into the box — the enrollee types their own message. No experiment setting gates it; like clearing the chat or rewinding it, it is a chat convenience rather than part of the condition. A sample prompt the experimenter suggests — it's only an example and isn't sent until you type and send your own.

9. **Upload follows blinding** There is no separate upload toggle. File upload and download are governed by `experiment.blinded`: a blinded study disables both, so nothing about the condition can enter or leave the session. This study is unblinded — the See-all pill below shows it — so attaching is available. Attach a .txt file to your message.