LLM-Rubric:

A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, Chris Kedzie

Microsoft

Code and data will be available at: https://github.com/microsoft/llm-rubric.

Has this ever happened to you?

- You need to evaluate a large collection of texts.
    - Perhaps you're doing legal discovery (Quartaro et al., 2019)
    - Or performing social science or market research (Mellon et al., 2024)
    - Or you are evaluating student writing (Page, 1968; Ramesh and Sanampudi, 2022)
    - Or perhaps you need to determine what papers to show at a conference
    - …

- So, you hire a human judge pool to evaluate said texts…

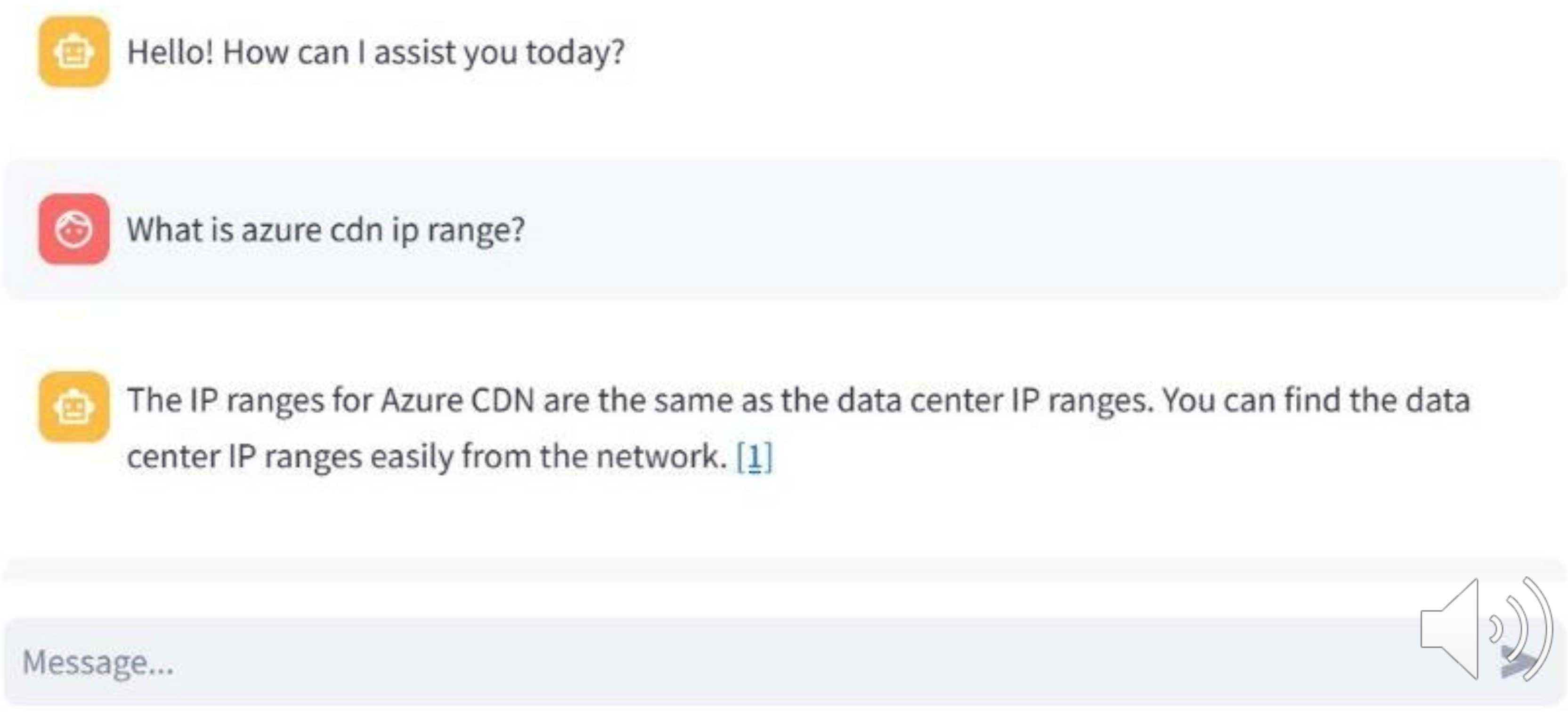
Or you need to evaluate a dialogue system…

- We hired a judge pool to annotate the logs of several IT-help dialogue systems.
- Judges evaluated systems on overall user satisfaction (Q_0).

Overall user satisfaction (Q_0).
Imagine you are the user who had this conversation with the assistant.

All in all, how would you rate your overall satisfaction while interacting with the assistant? The higher the rating, the better the experience.

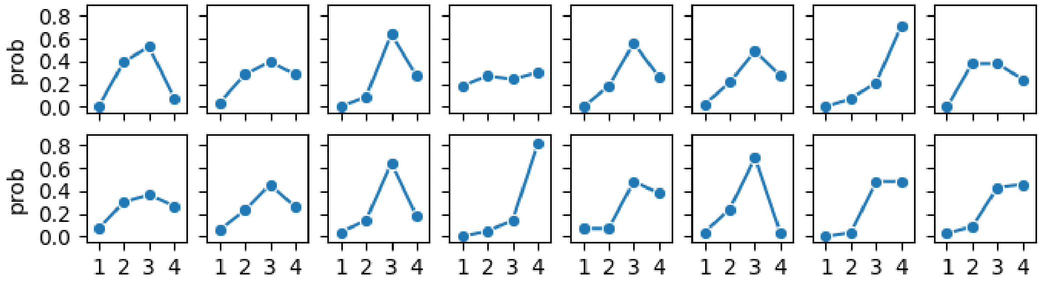
○1 ○2 ●3 ○4



… but your human judge pool is difficult to maintain

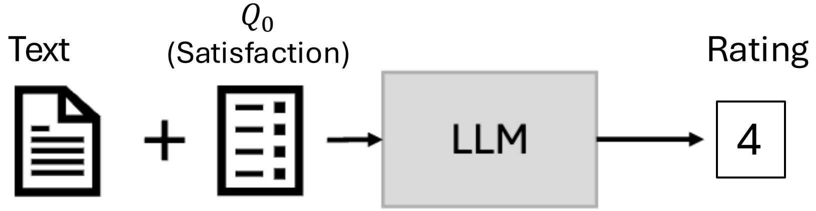
- Human annotation can have its own reliability challenges
  (Hosking et al., 2023; Liu et al., 2016; Smith et al., 2022)
- Human judges may reasonably disagree
  (Pavlick and Kwiatkowski, 2019; Basile et al., 2021; Plank, 2022; Sandri et al., 2023)

Histograms of Q_0 Likert scale ratings of 16 judges in our pool.

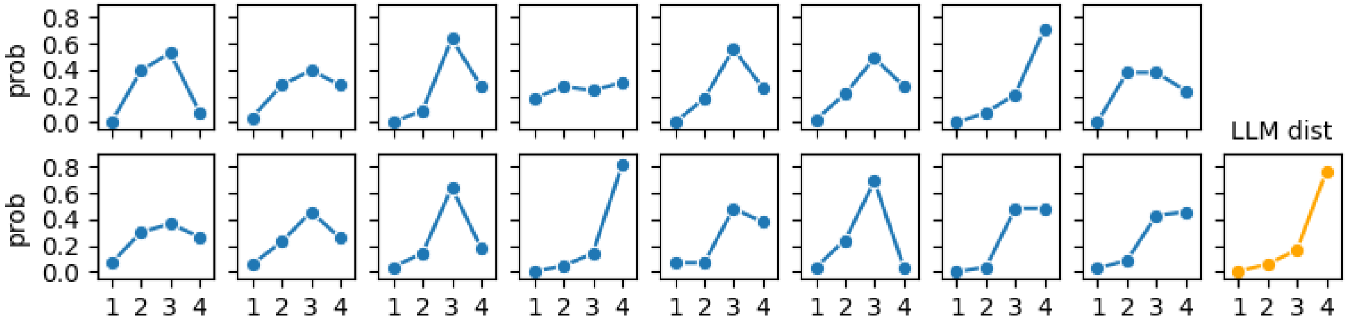


Should I replace my judge pool with an LLM?

We gave an LLM the same instructions and had it predict the Likert scale rating for each text to be evaluated...



but it was too optimistic...



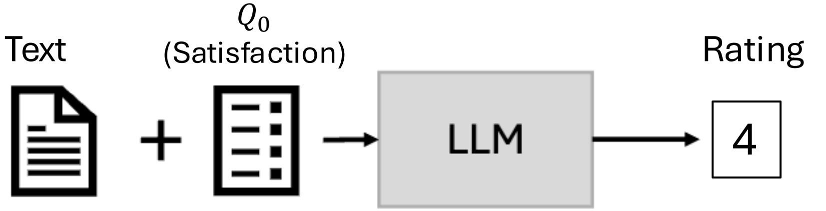
Should I replace my judge pool with an LLM?

We gave an LLM the same instructions and had it predict the Likert scale rating for each text to be evaluated...

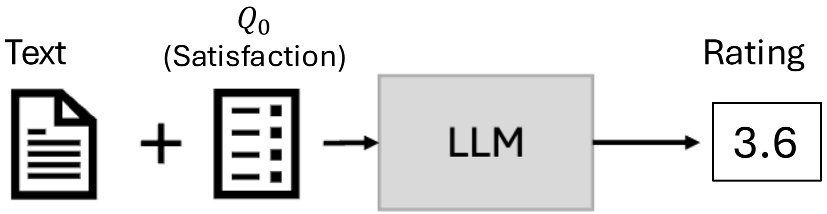
Classification

better formulation

Regression



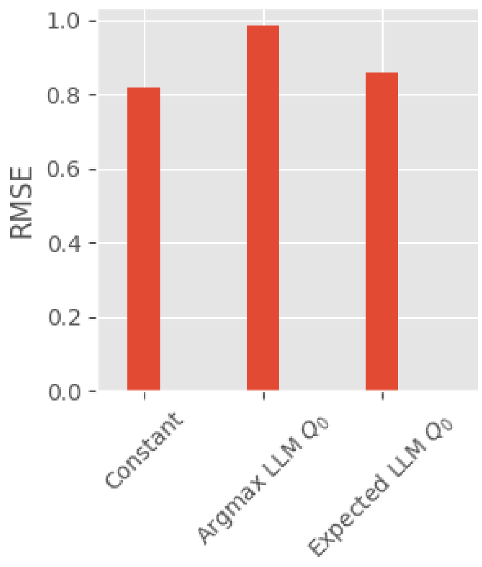
Argmax LLM Q_0: $\hat{y}_0^a = \operatorname{argmax}_{y \in \{1,2,3,4\}} p_{LLM}(y|T, Q_0)$



Expected LLM Q_0: $\hat{y}_0^a \propto \sum_{y \in \{1,2,3,4\}} p_{LLM}(y|T, Q_0) \cdot y$

Should I replace my judge pool with an LLM?

In fact, it was about as predictive of judge preferences as the judge pool's mean rating!

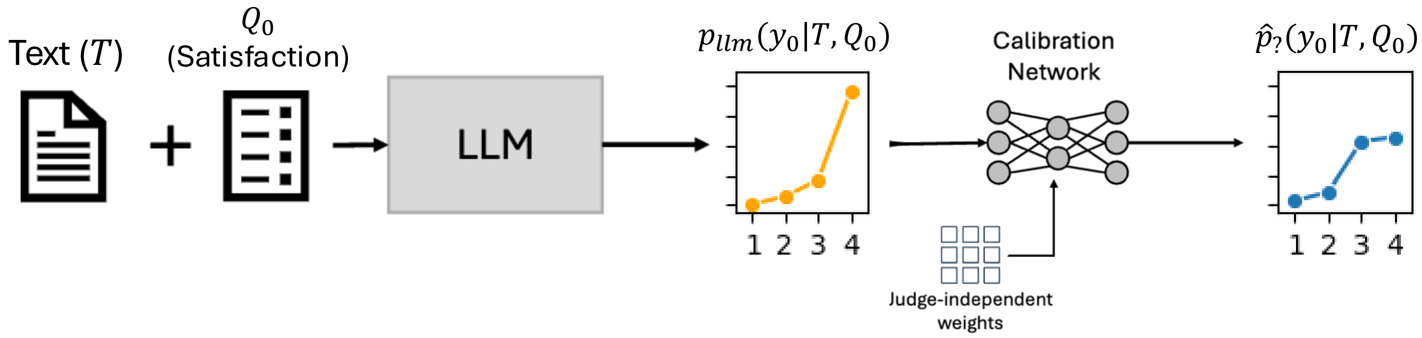


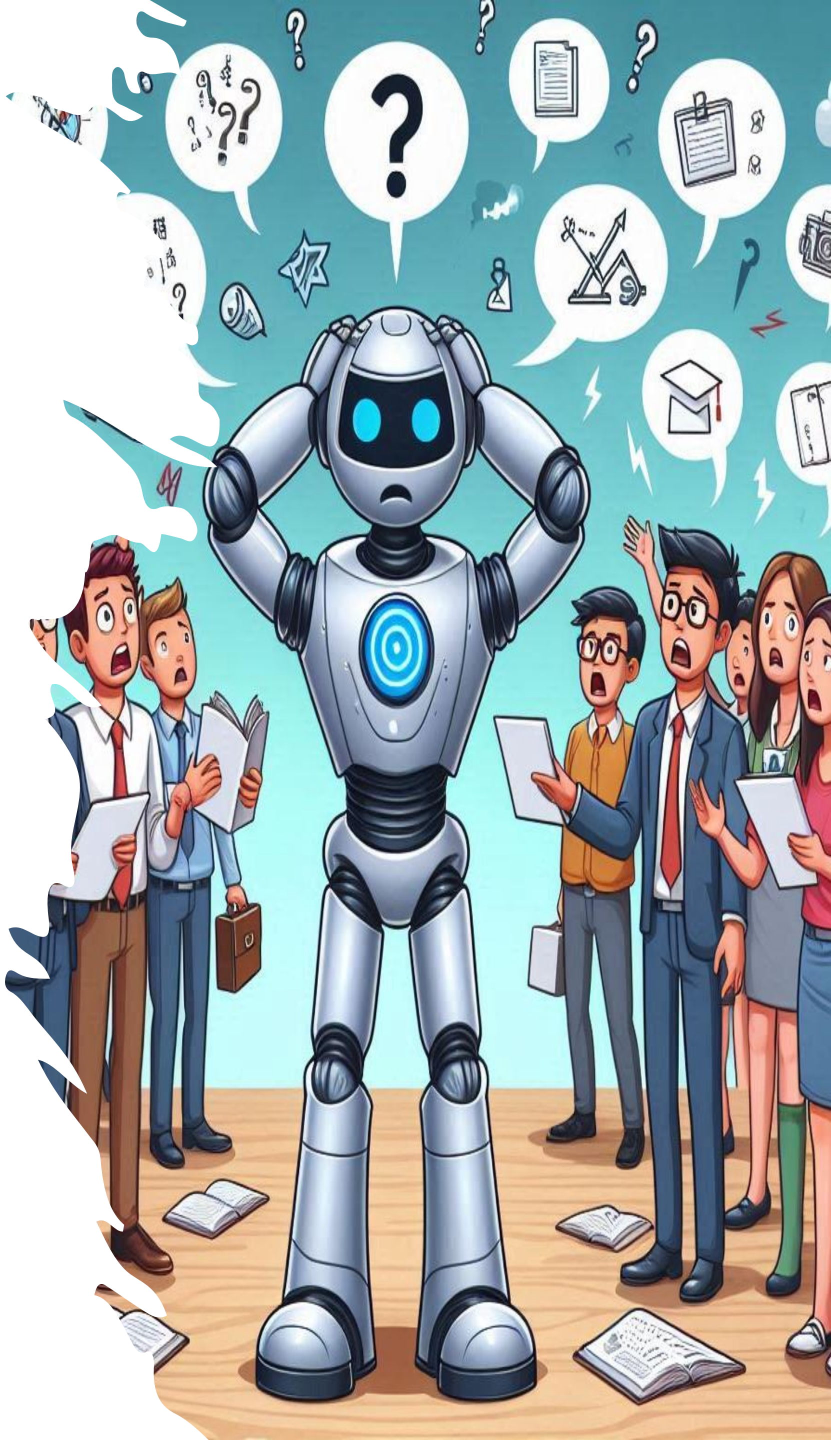
$$\text{RMSE} = \sqrt{\frac{\sum_{(T, y_0^a) \in D_{test}} (y_0^a - \hat{y}_0^a)^2}{|D_{test}|}}$$

y_0^a: Judge a Ground Truth Rating
$\hat{y}_0^a$: Judge a Predicted Rating

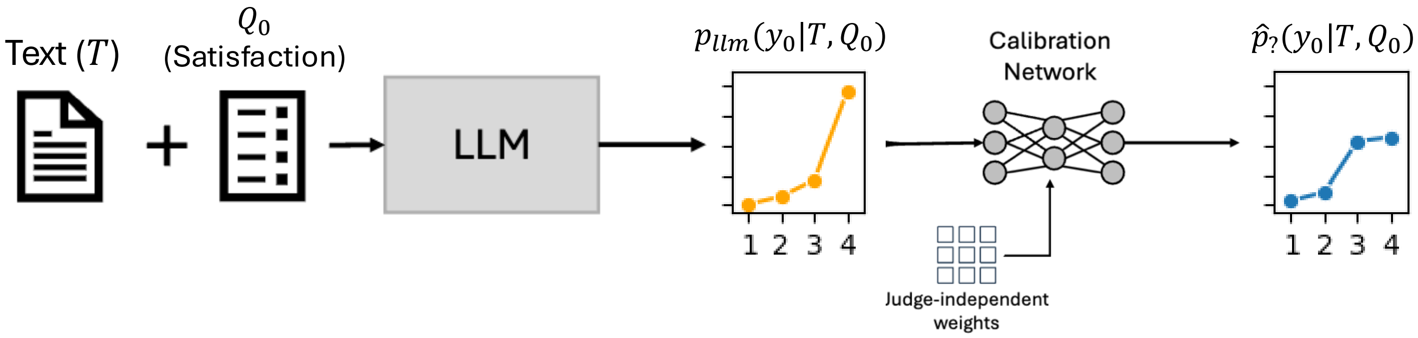
- Constant: predicted rating is always the training set mean. ($\hat{y}_0^a = 3.04$)
- Argmax LLM Q_0 (Classification)
- Expected LLM Q_0 (Regression)

So, you decide to calibrate the LLM...

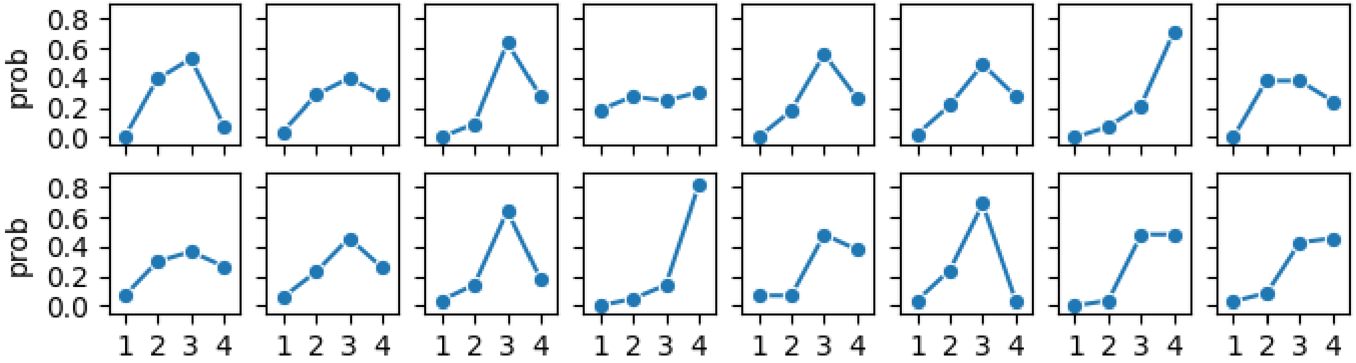


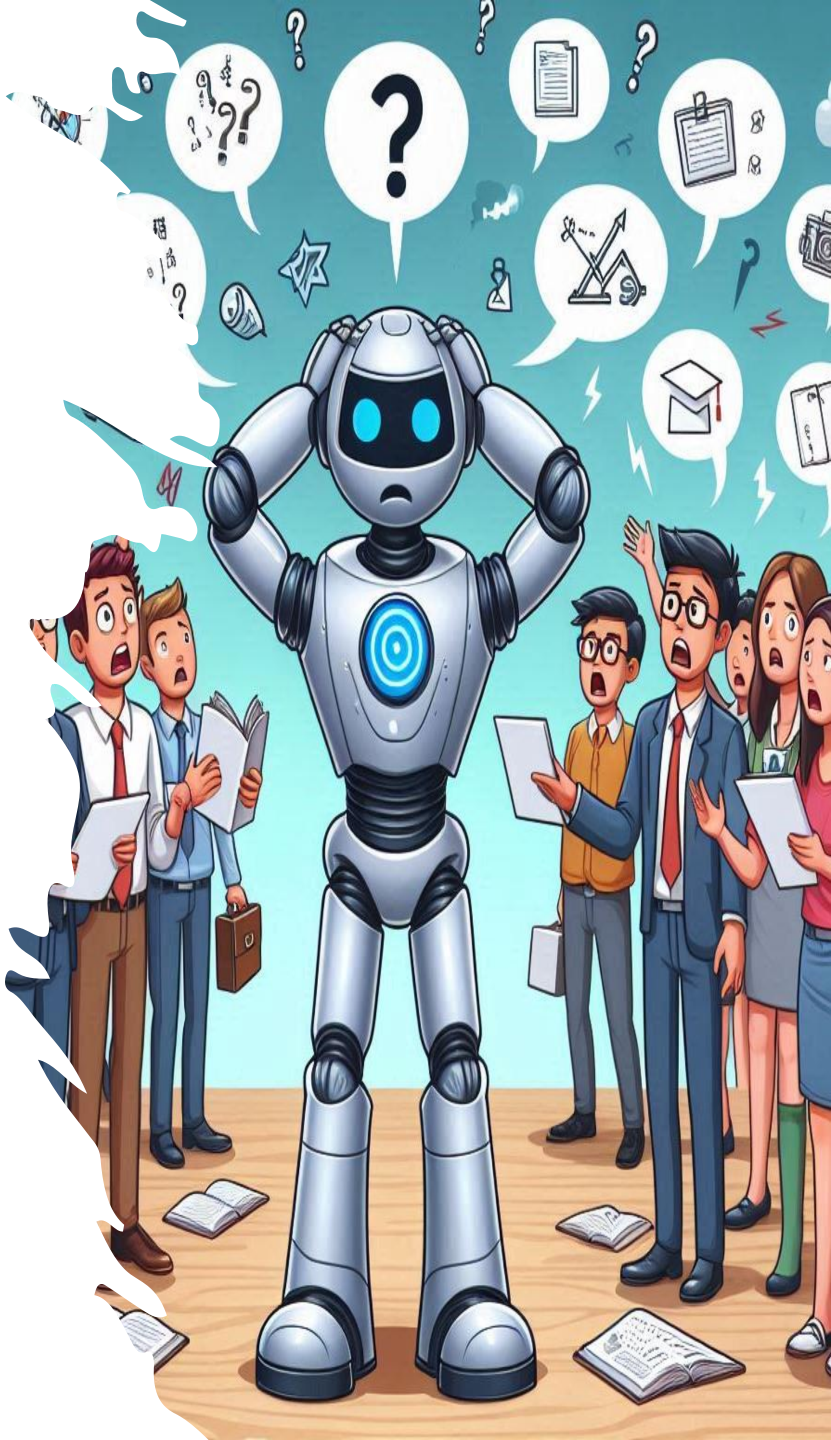


So, you decide to calibrate the LLM...

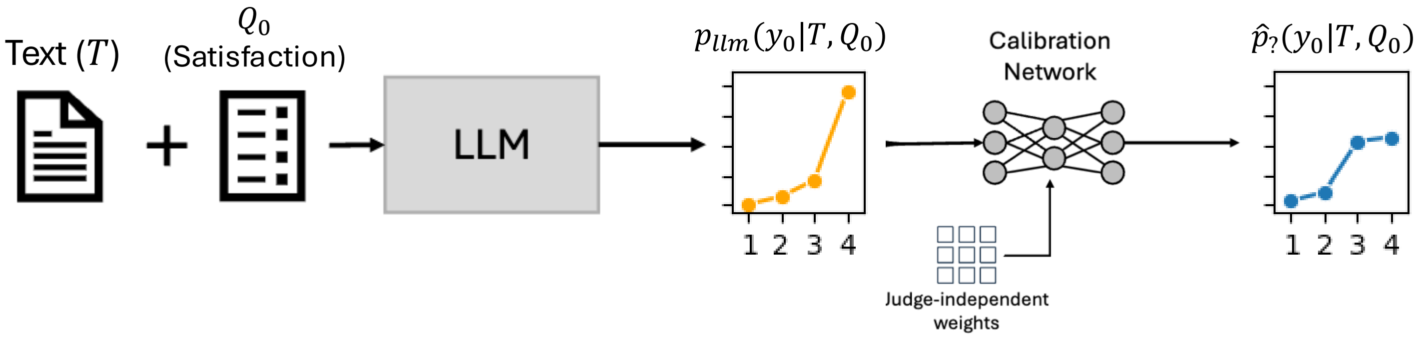


But it's not clear which judge or judges to use as the calibration target...

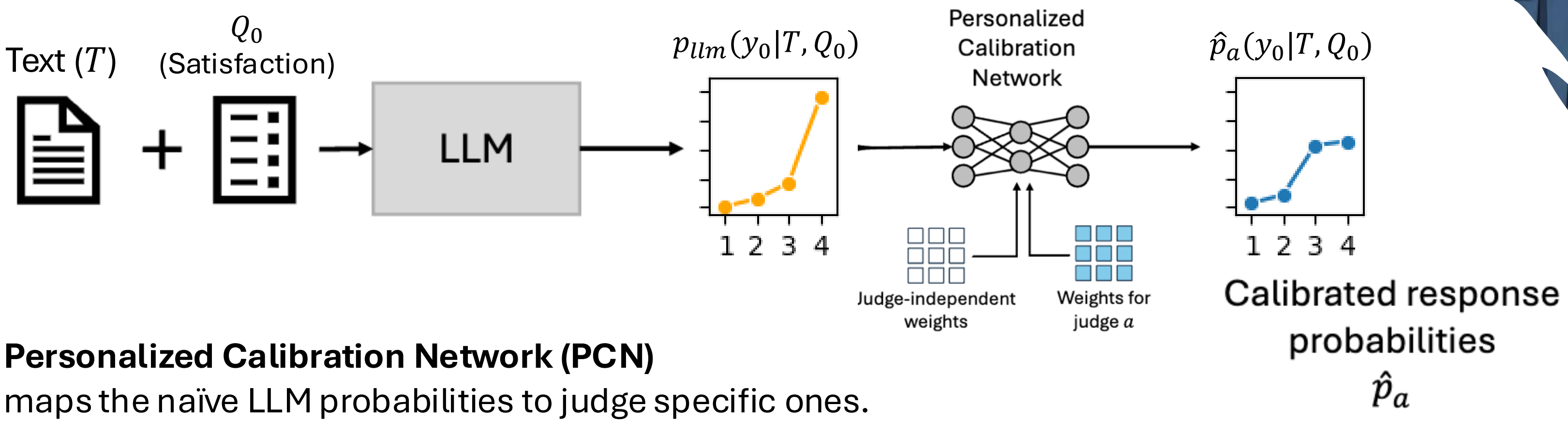




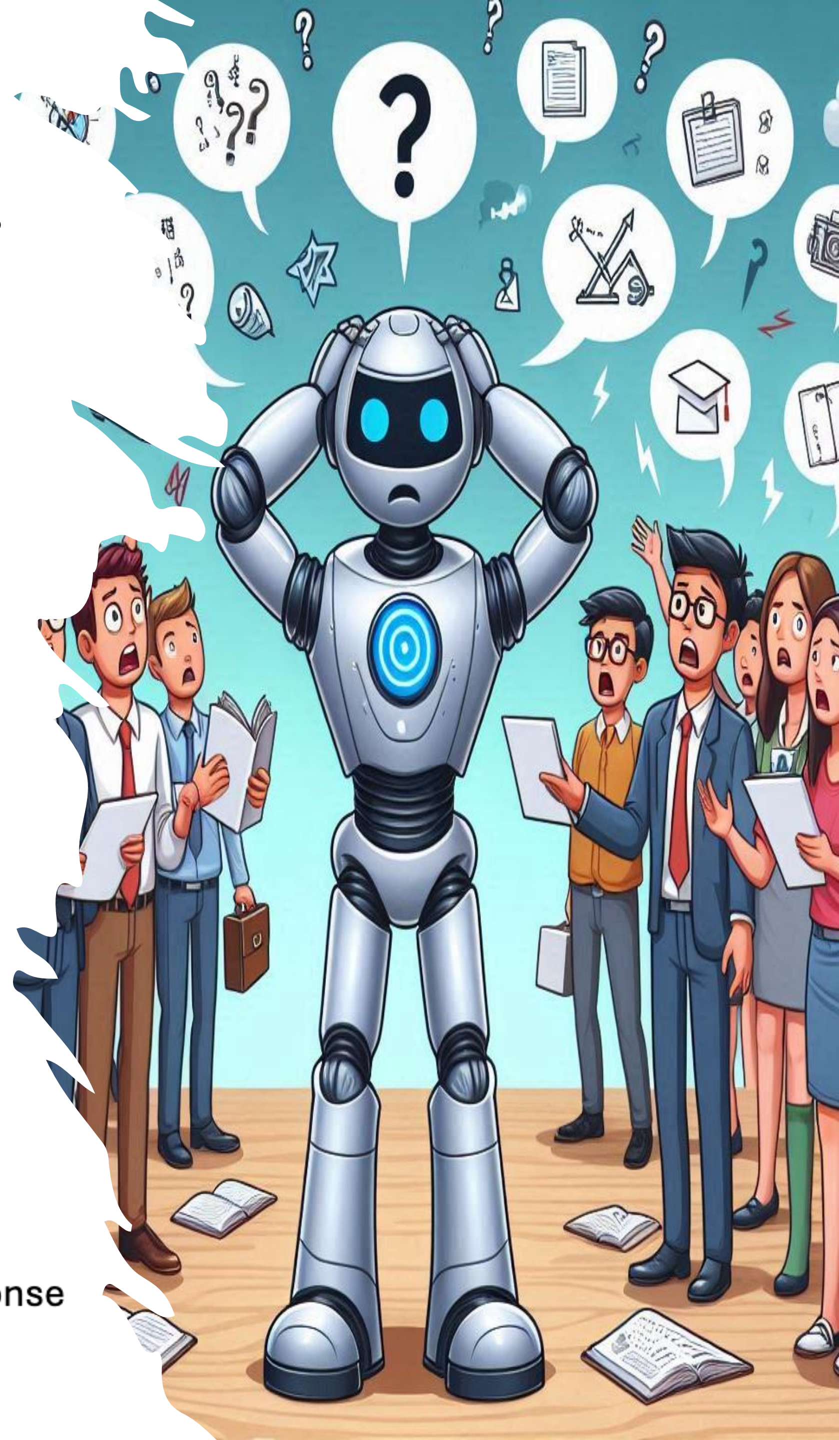
So, you decide to calibrate the LLM…



So, you calibrate to each judge and avoid collapsing disagreements.

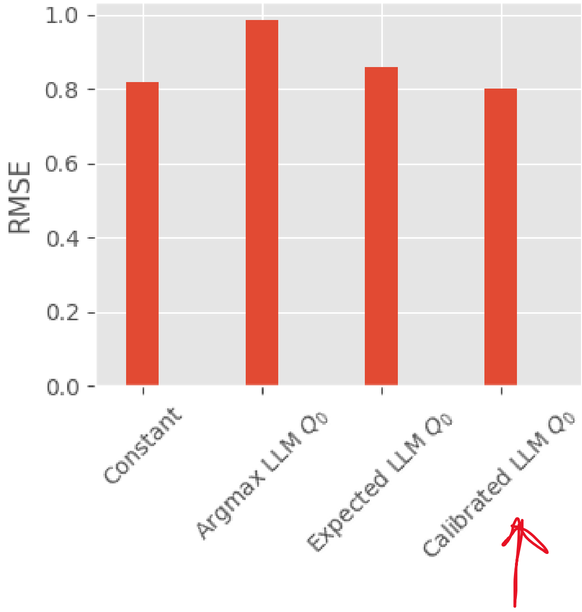


Personalized Calibration Network (PCN)
maps the naïve LLM probabilities to judge specific ones.

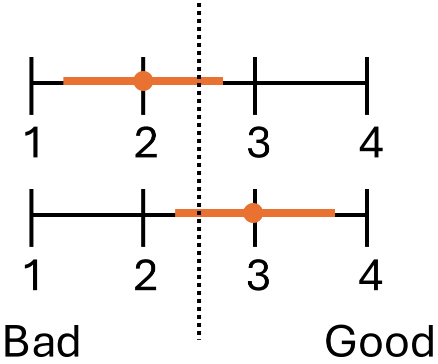


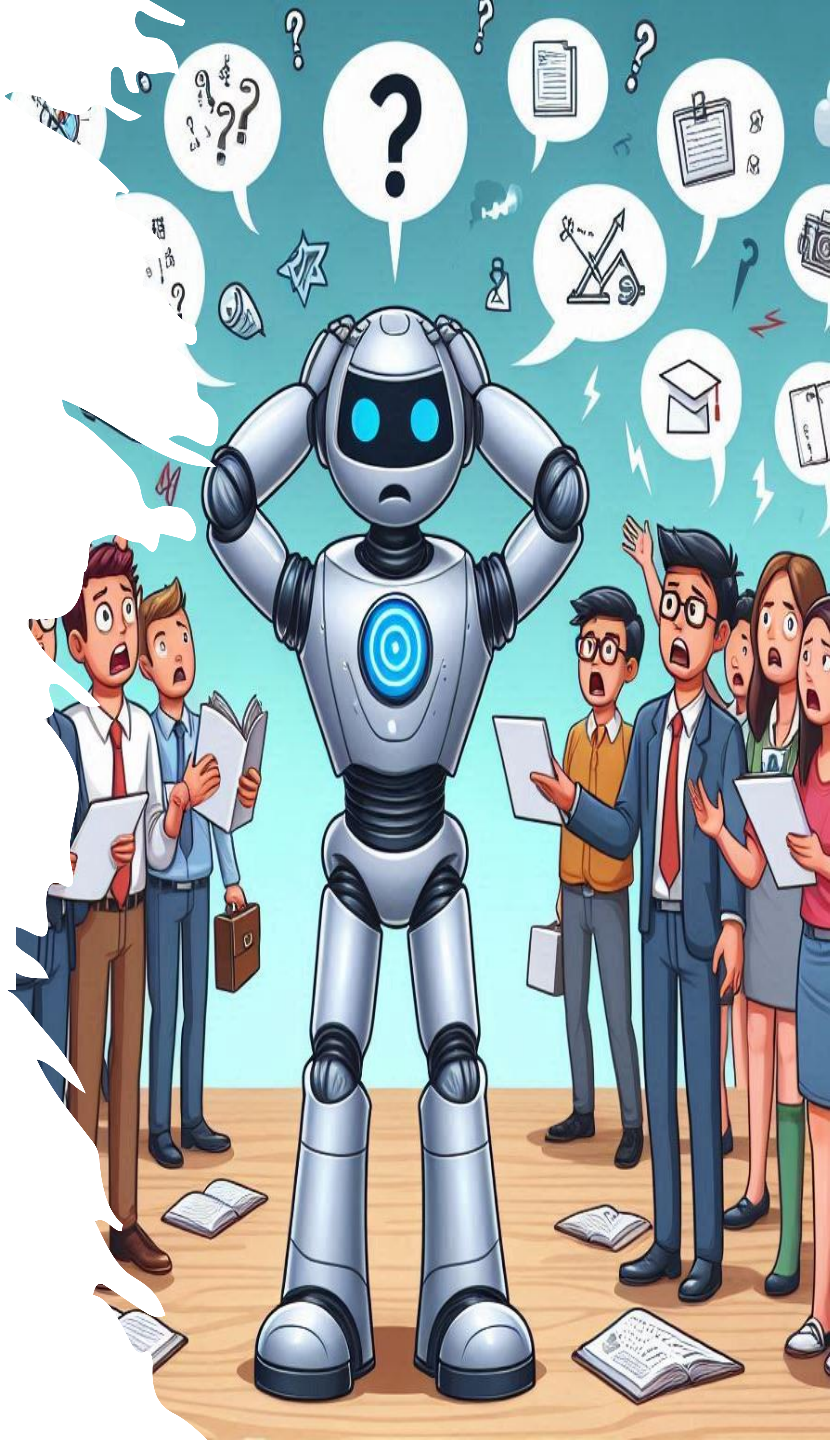
Calibration improves the accuracy...

But we are still over 0.75 of a point off on average!



On our 4-point Likert scale, that's enough to be the difference between a good and bad user experience.

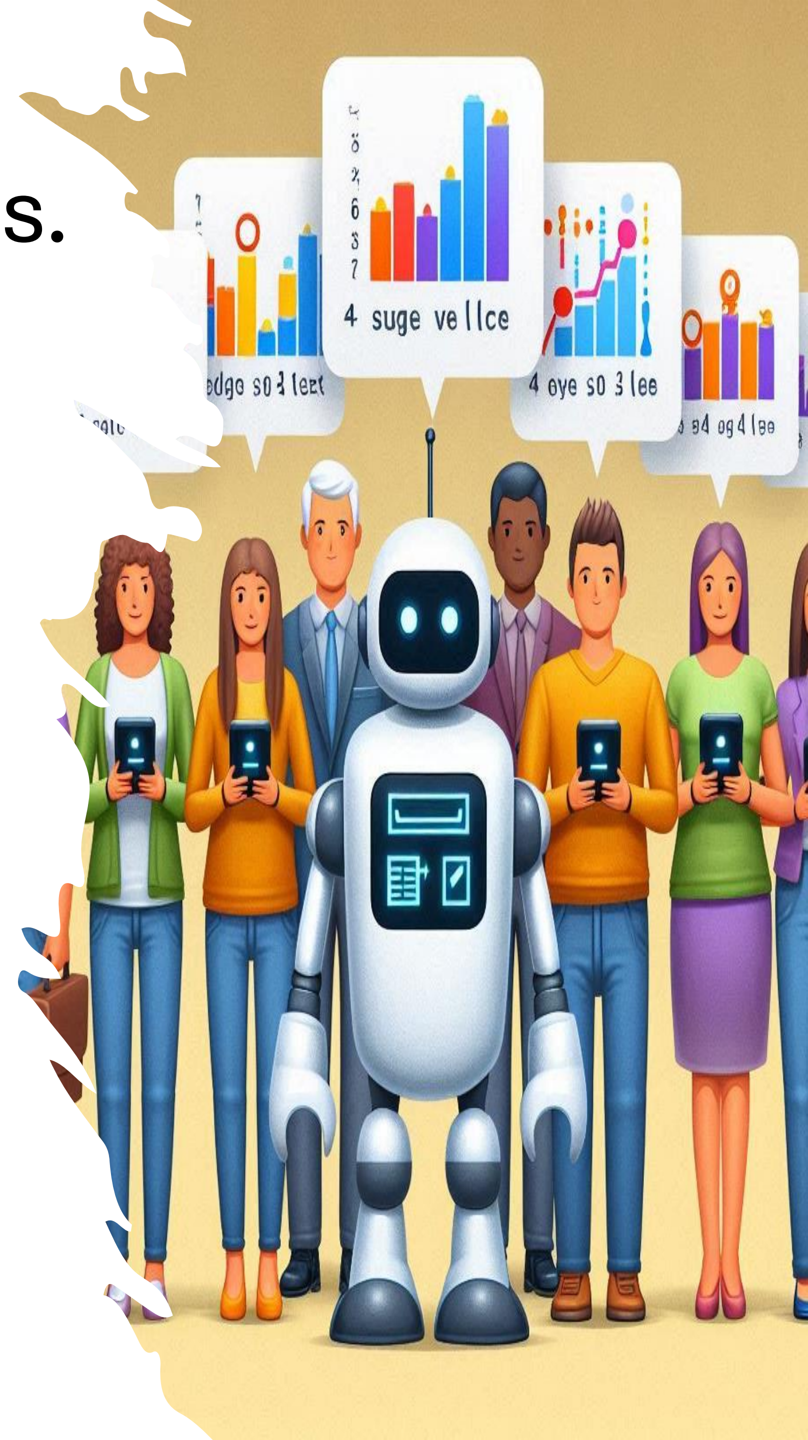




So, we asked more fine-grained questions.

Judges rated texts according to the following criteria:

Q_1 Naturalness
Q_2 Grounding Sources
Q_3 Citation Presence
Q_4 Citation Suitability
Q_5 Citation Optimality
Q_6 Redundancy
Q_7 Conciseness
Q_8 Efficiency

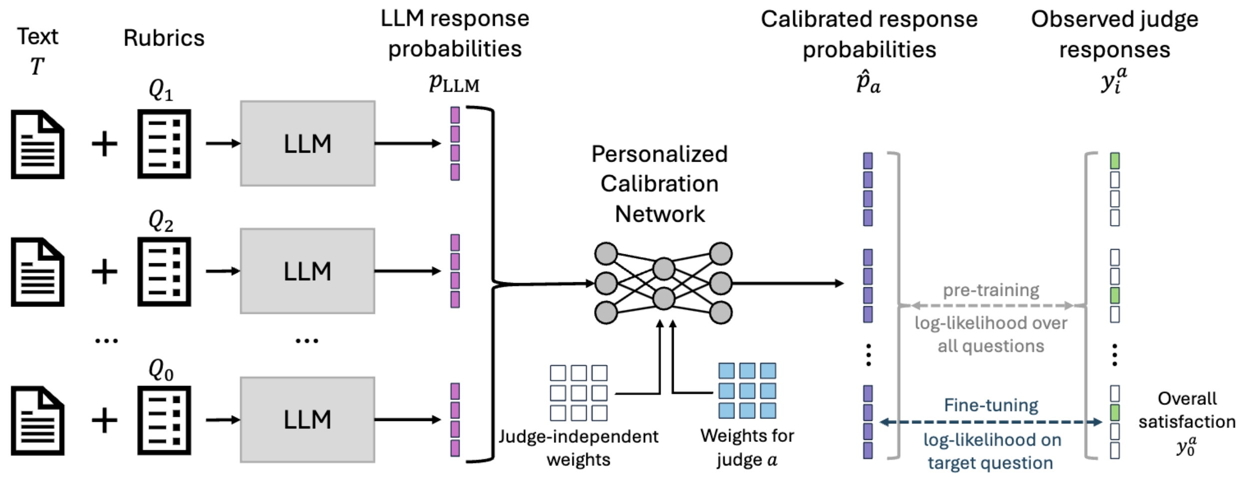


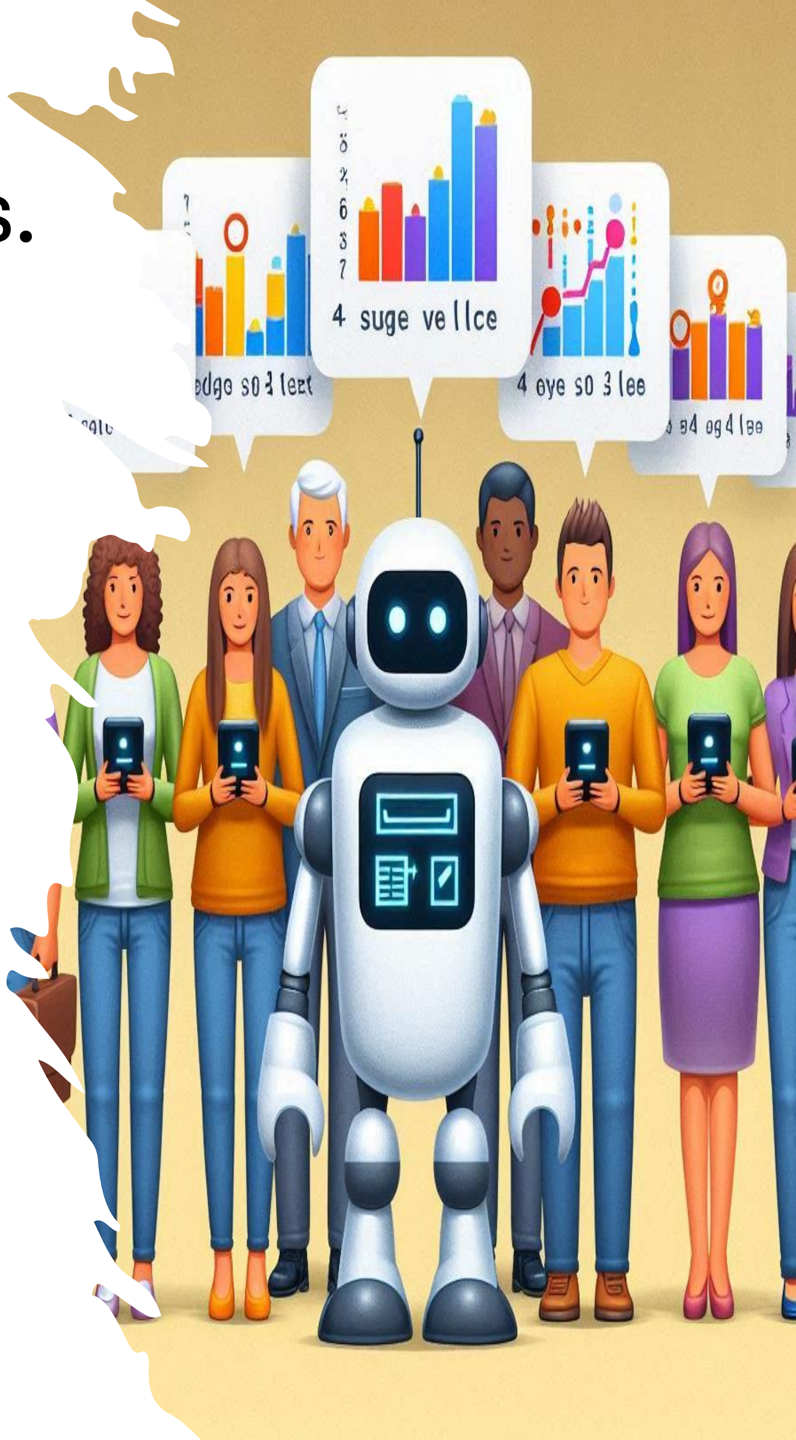
So we asked more fine-grained questions.

Judges asked to rate texts according to the following criteria:

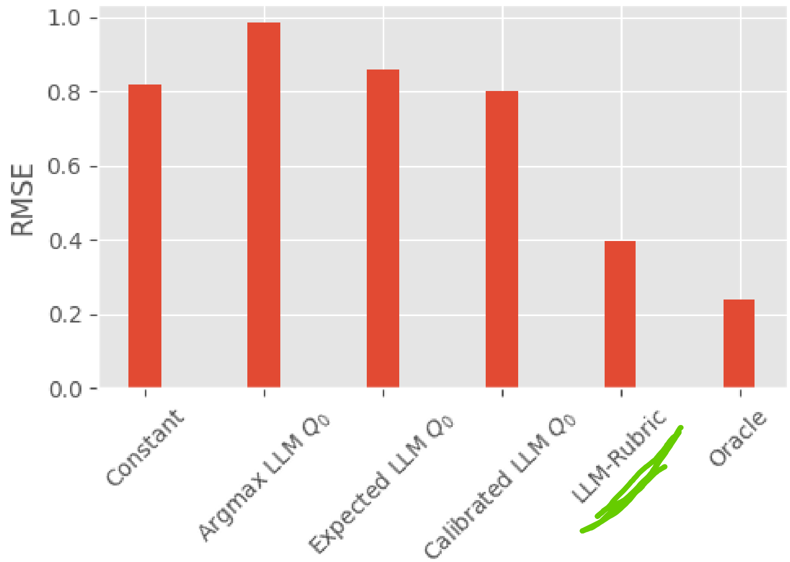
Q_1 Naturalness
Q_2 Grounding Sources
Q_3 Citation Presence
Q_4 Citation Suitability
Q_5 Citation Optimality
Q_6 Redundancy
Q_7 Conciseness
Q_8 Efficiency

We then included prediction Q_1 ... Q_8 as auxiliary tasks in a multi-task learning setup.

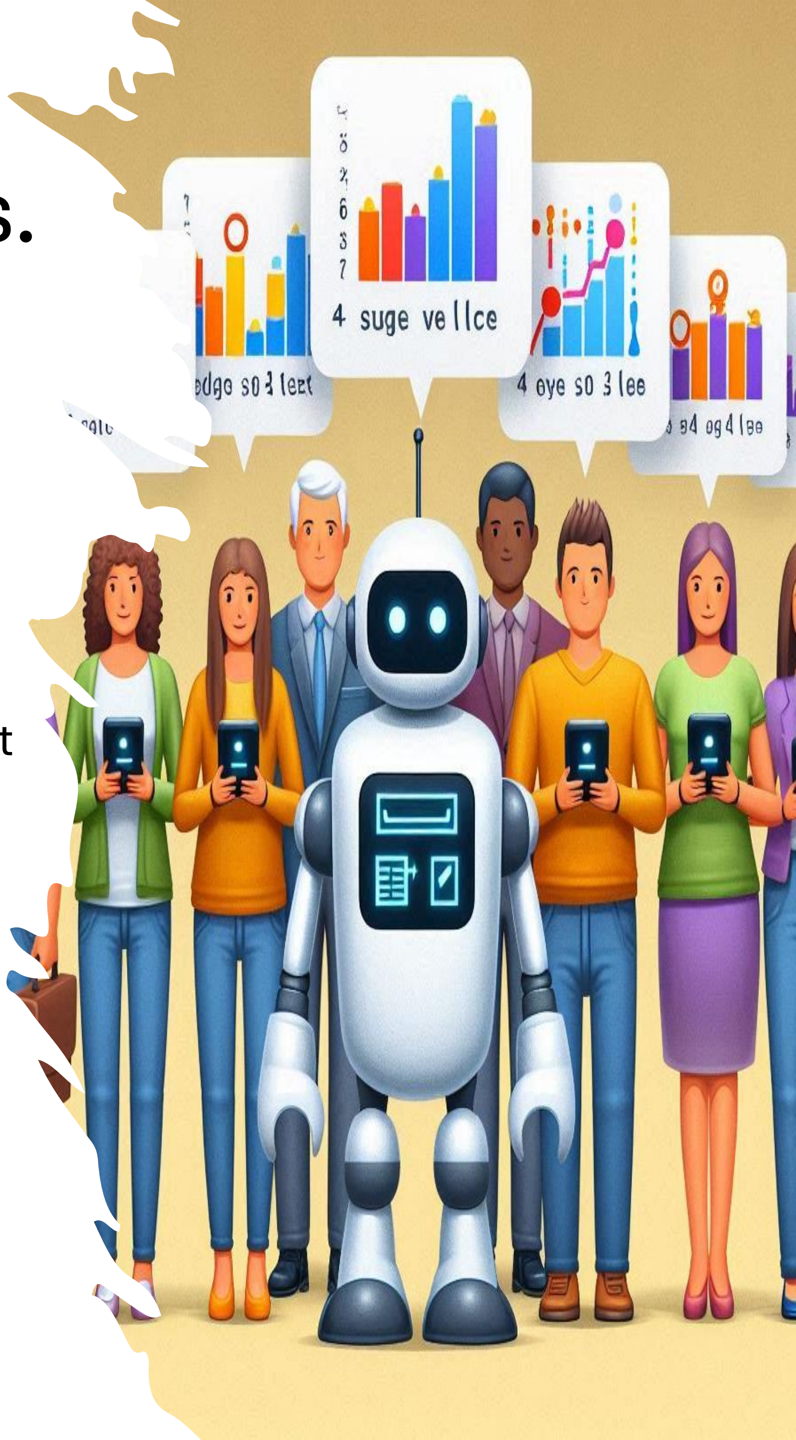




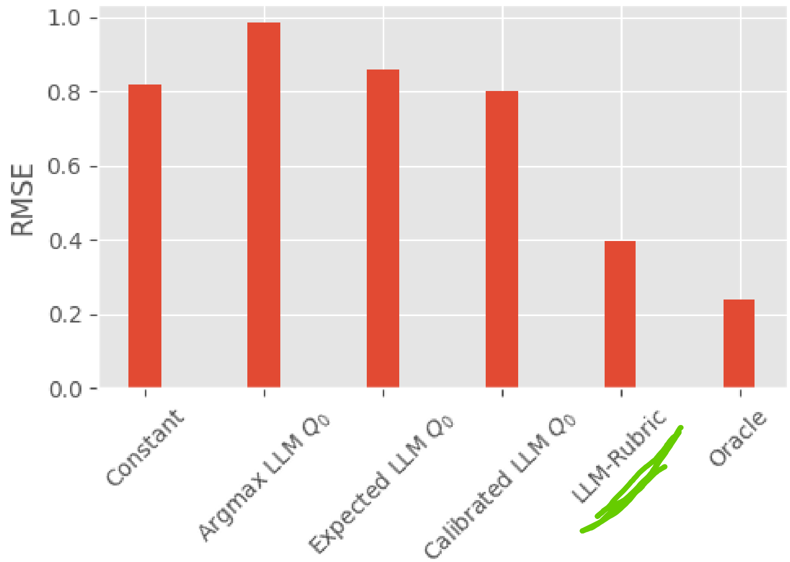
So we asked more fine-grained questions.



By combining personalization and multi-task learning, LLM-Rubric achieves sub 0.5 RMSE on a 4-point Likert scale rating task.

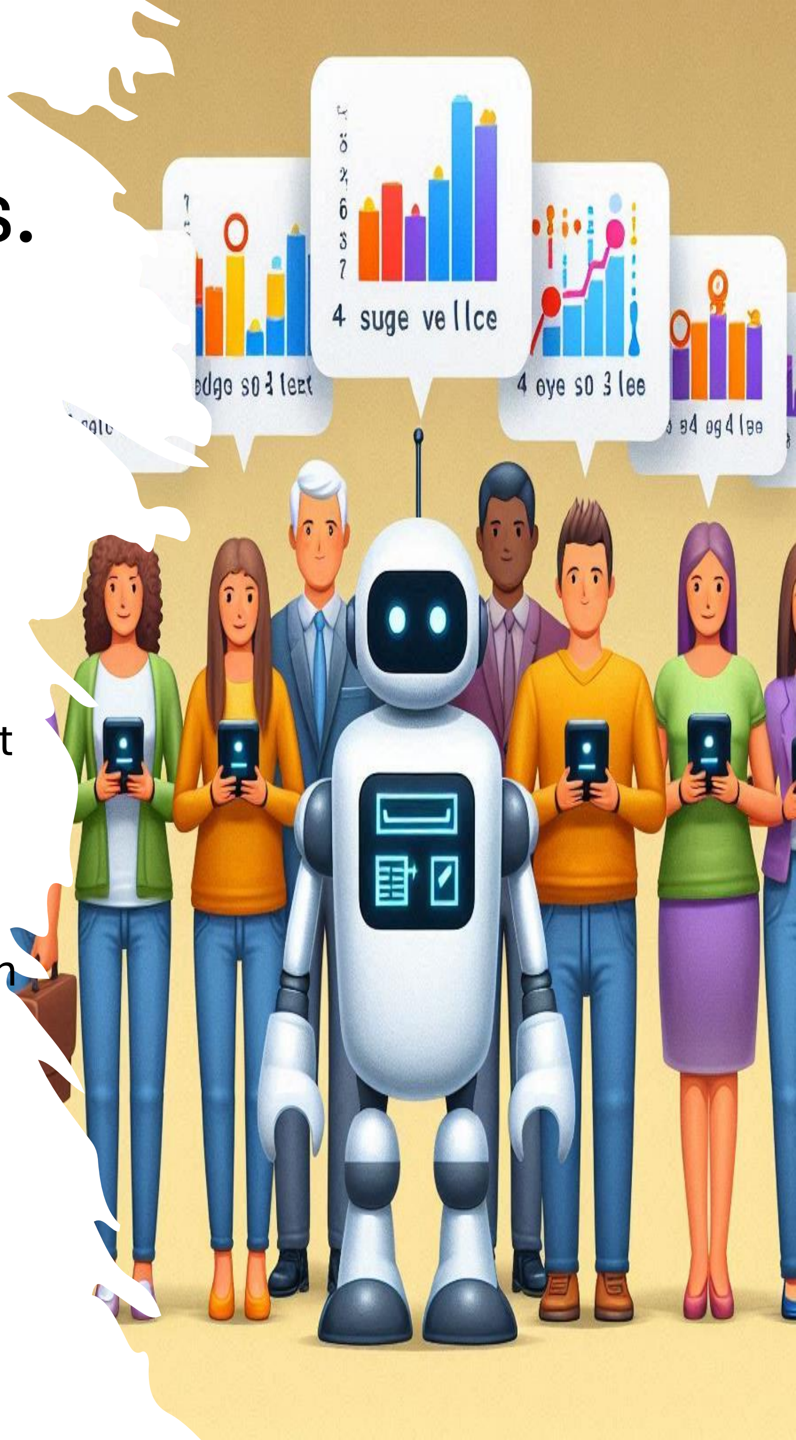


So we asked more fine-grained questions.

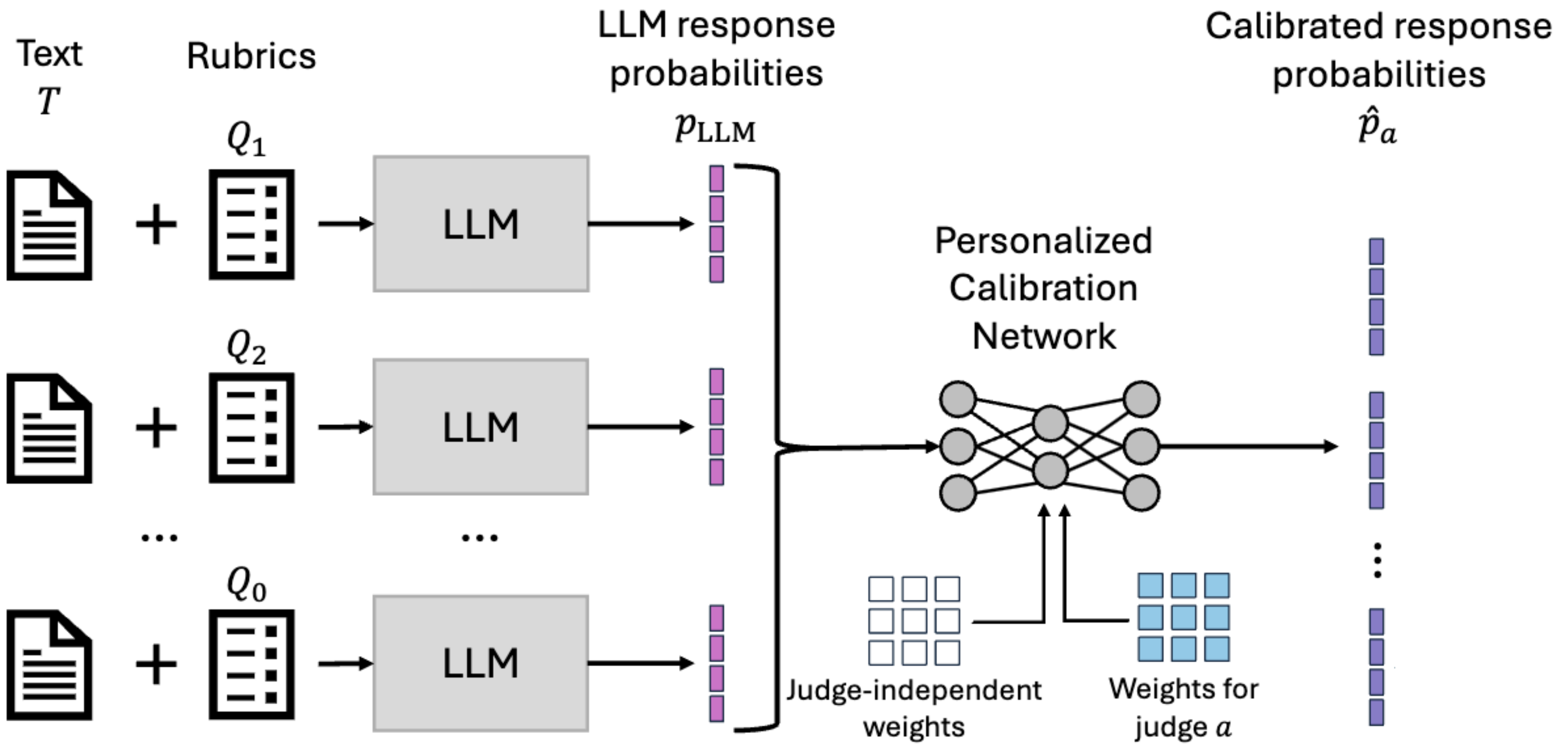


By combining personalization and multi-task learning, LLM-Rubric achieves sub 0.5 RMSE on a 4-point Likert scale rating task.

LLM-Rubric is not that much worse than an oracle that predicts Q_0 from the judge's ground truth answers to $Q_1, \ldots, Q_8$!



LLM-Rubric



Motivation

- Align LLM eval with human judges
- Model judges' disagreements on supervised data, rather than collapsing them
- Better predict each human by combining multiple LLM questions

Using LLM-Rubric we get statistically significant improvements in

- RMSE on Likert scale rating prediction
- correlation with text rankings by humans

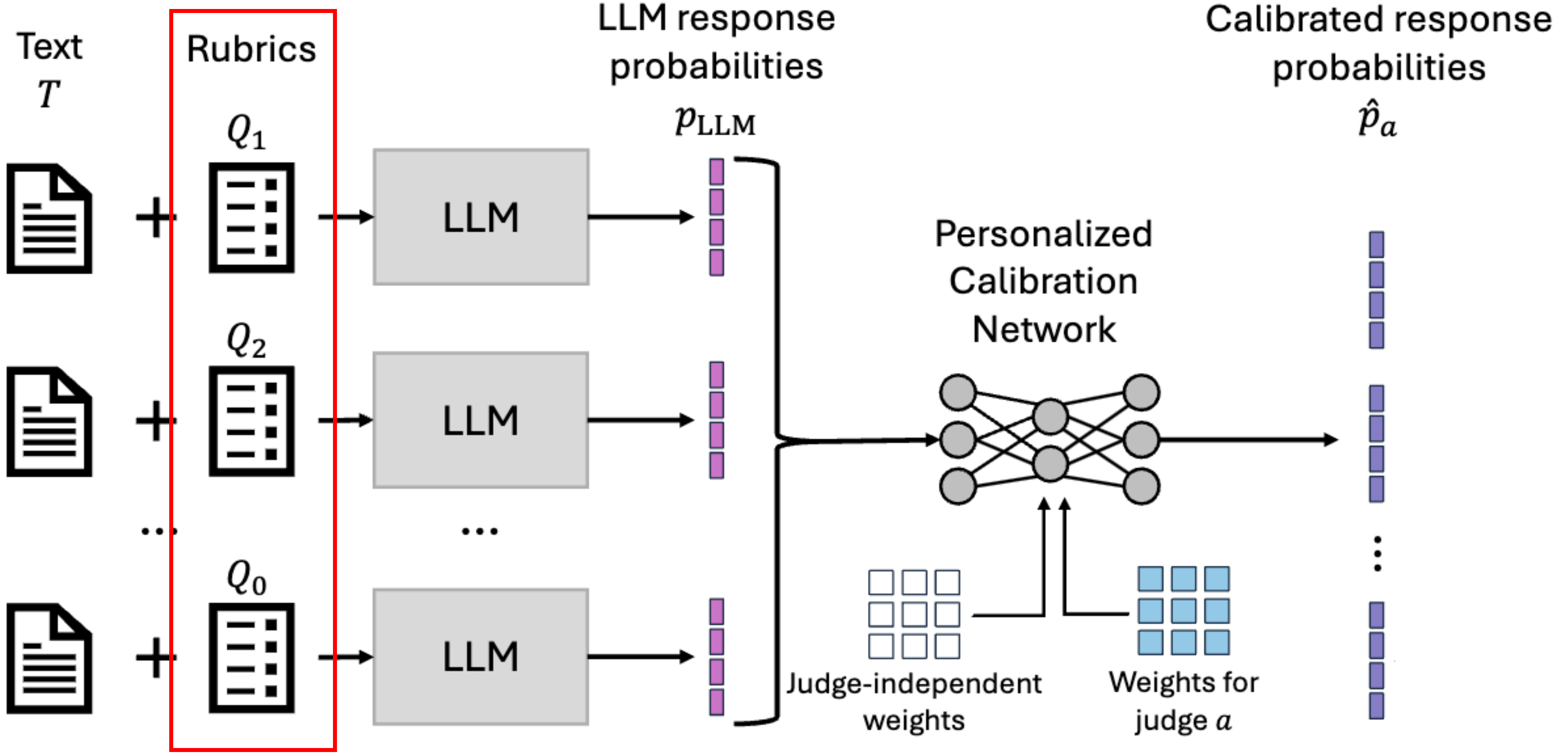
Manually write several questions (a rubric)

Given a text T:

1. For each question, get an *LLM's* distribution over the possible responses

2. Predict how each *human* judge would respond: map the set of LLM response distributions to the judge's response distributions

LLM-Rubric



Manually write several questions (a rubric)

Given a text T:

1. For each question, get an *LLM's* distribution over the possible responses

2. Predict how each *human* judge would respond: map the set of LLM response distributions to the judge's response distributions

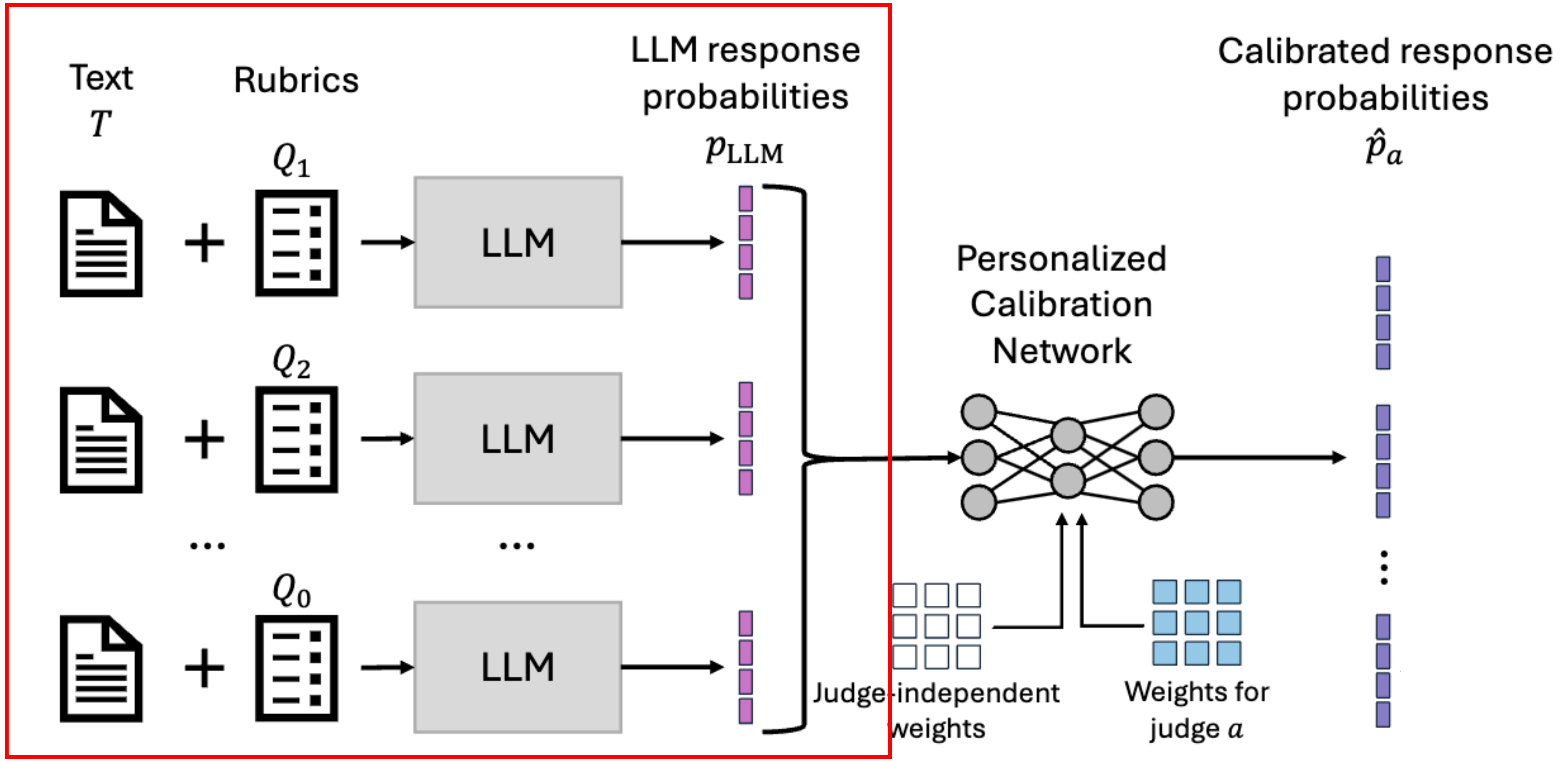
Motivation

- Align LLM eval with human judges
- Model judges' disagreements on supervised data, rather than collapsing them
- Better predict each human by combining multiple LLM questions

Using LLM-Rubric we get statistically significant improvements in

- RMSE on Likert scale rating prediction
- correlation with text rankings by humans

LLM-Rubric



Manually write several questions (a rubric)

Given a text T:

1. For each question, get an *LLM's* distribution over the possible responses

2. Predict how each *human* judge would respond: map the set of LLM response distributions to the judge's response distributions

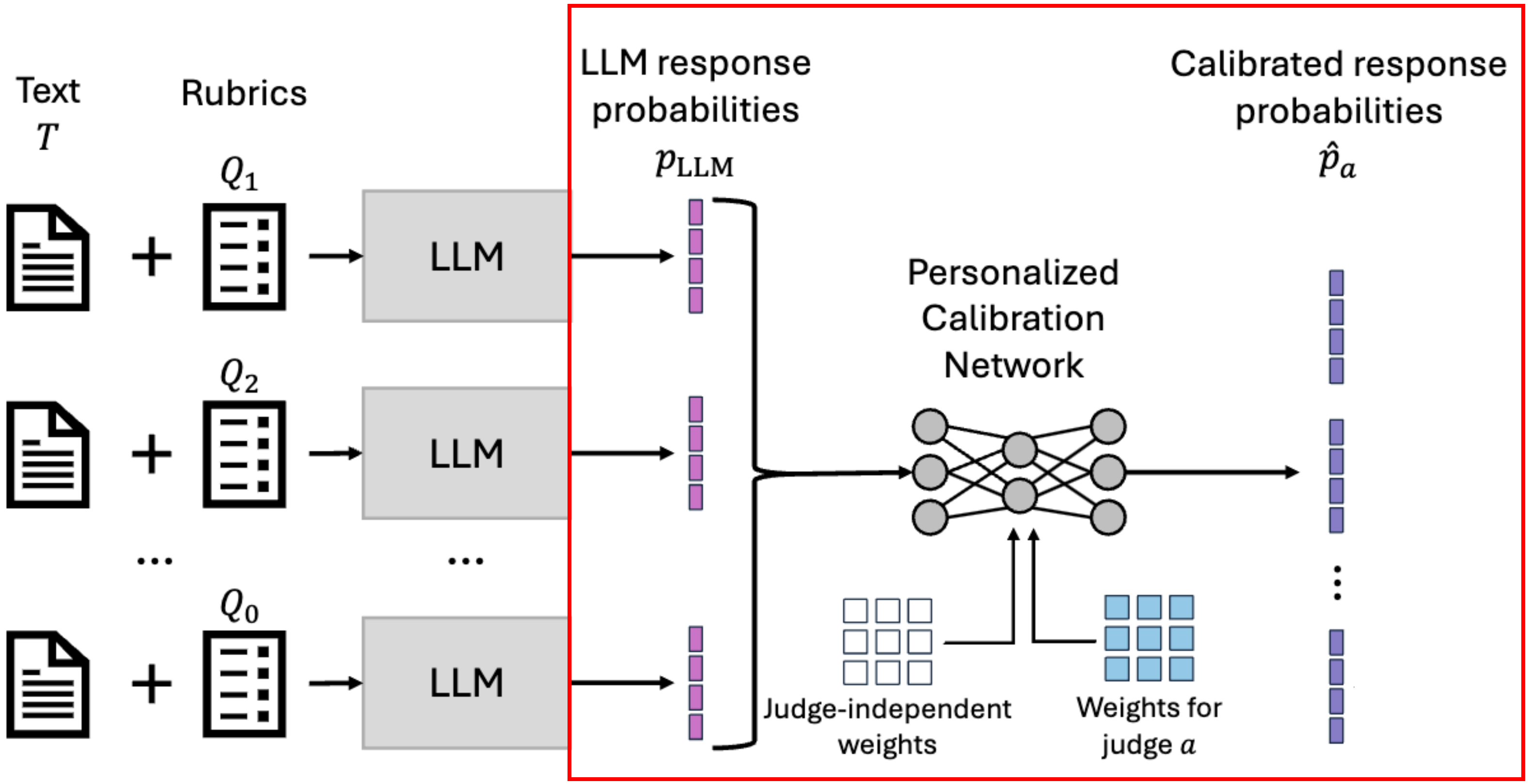
Motivation

- Align LLM eval with human judges
- Model judges' disagreements on supervised data, rather than collapsing them
- Better predict each human by combining multiple LLM questions

Using LLM-Rubric we get statistically significant improvements in

- RMSE on Likert scale rating prediction
- correlation with text rankings by humans

LLM-Rubric



Manually write several questions (a rubric)

Given a text T:

1. For each question, get an *LLM's* distribution over the possible responses

2. Predict how each *human* judge would respond: map the set of LLM response distributions to the judge's response distributions

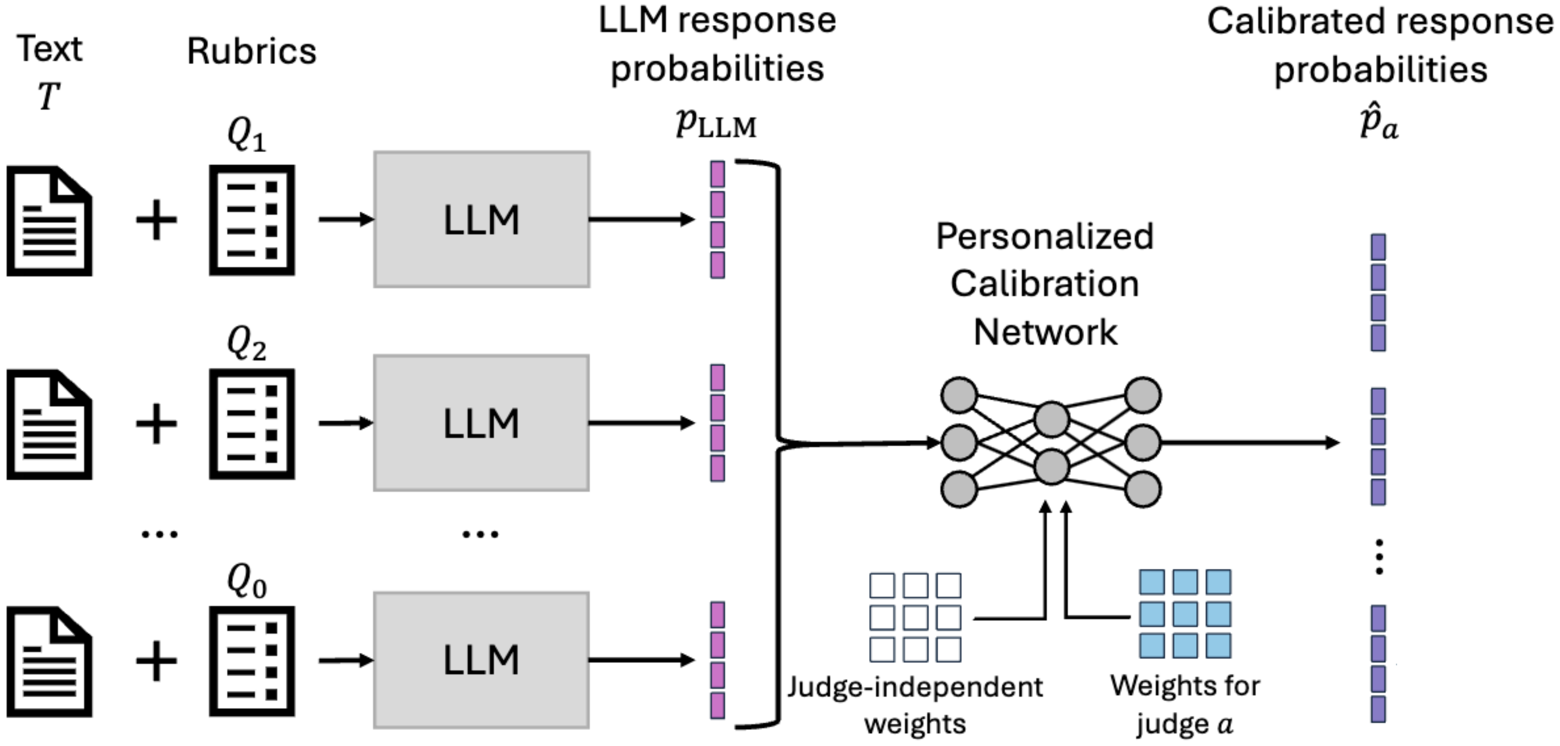
Motivation

- Align LLM eval with human judges
- Model judges' disagreements on supervised data, rather than collapsing them
- Better predict each human by combining multiple LLM questions

Using LLM-Rubric we get statistically significant improvements in

- RMSE on Likert scale rating prediction
- correlation with text rankings by humans

LLM-Rubric



Motivation

- Align LLM eval with human judges
- Model judges' disagreements on supervised data, rather than collapsing them
- Better predict each human by combining multiple LLM questions

Using LLM-Rubric we get statistically significant improvements in

- RMSE on Likert scale rating prediction
- correlation with text rankings by humans

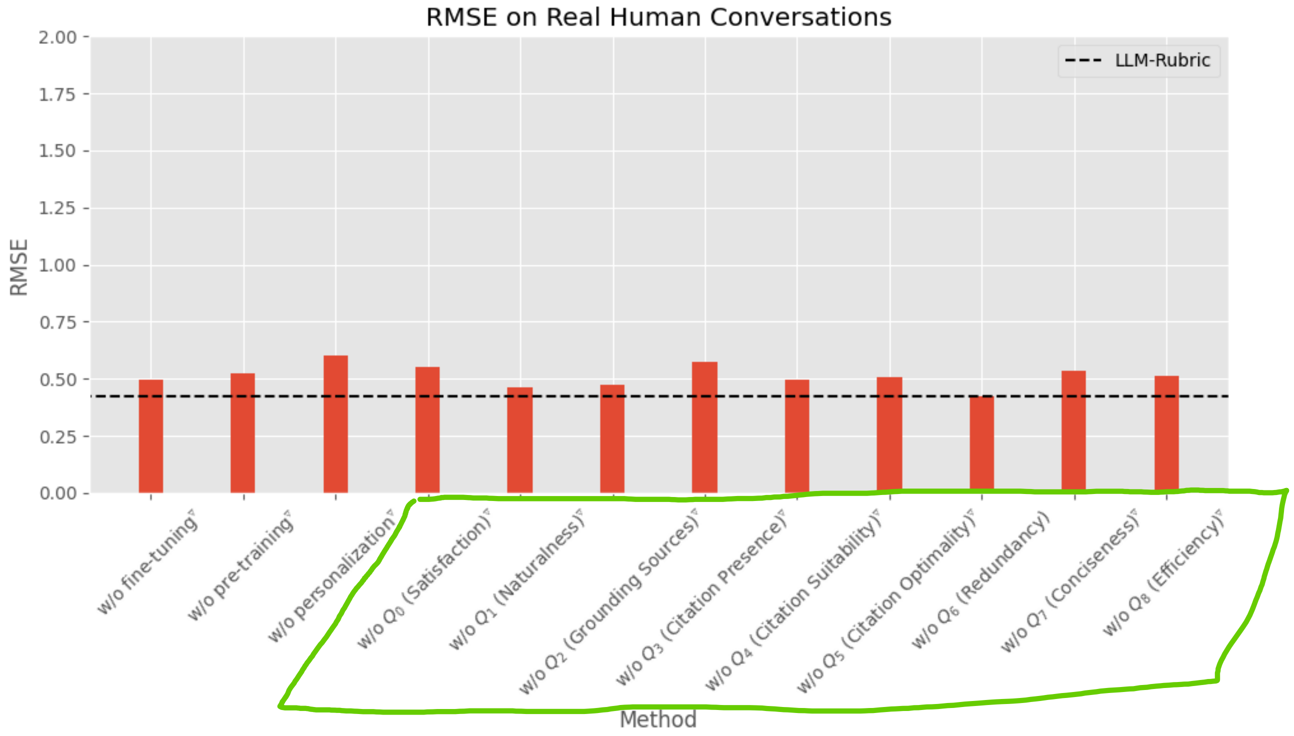
Manually write several questions (a rubric)

Given a text T:

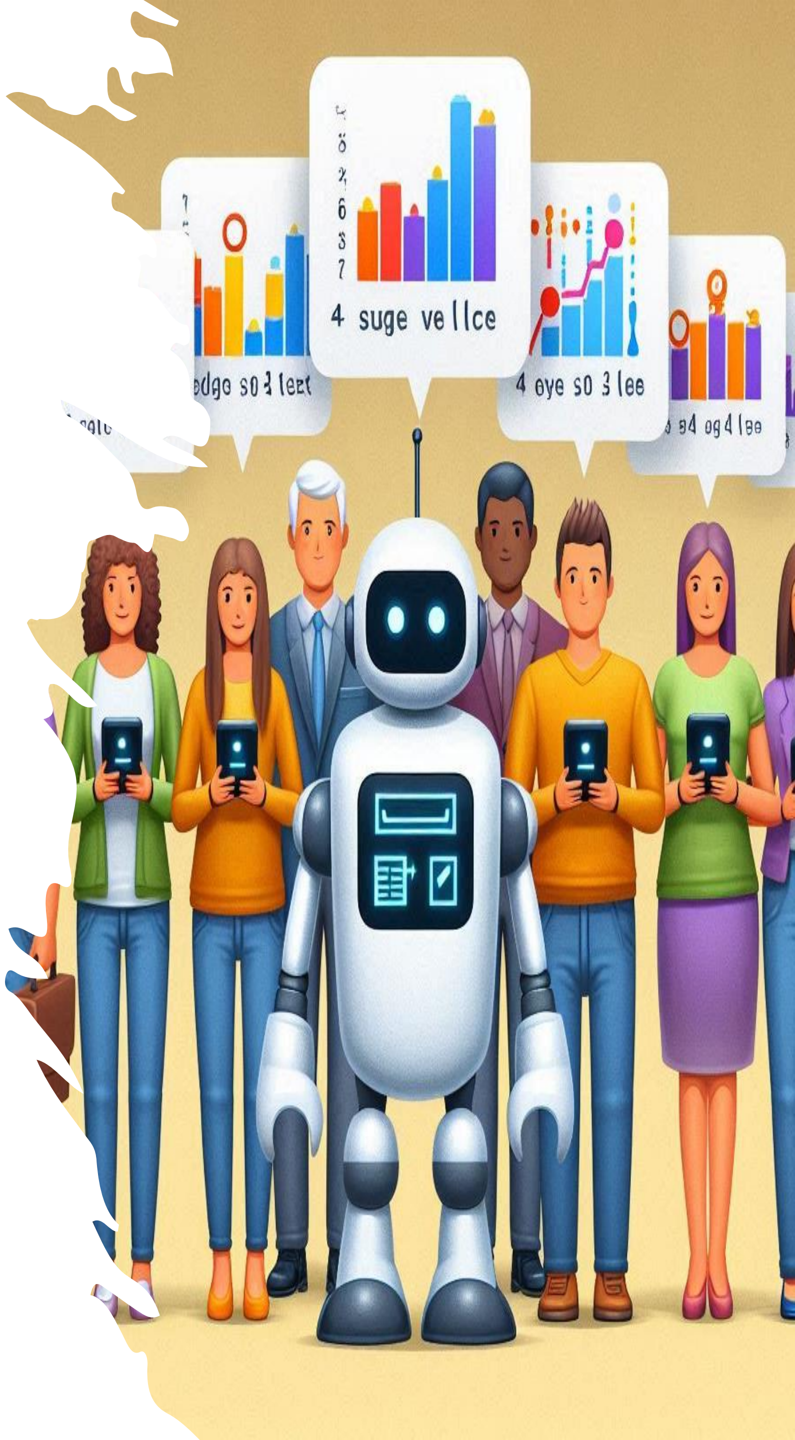
1. For each question, get an *LLM's* distribution over the possible responses

2. Predict how each *human* judge would respond: map the set of LLM response distributions to the judge's response distributions

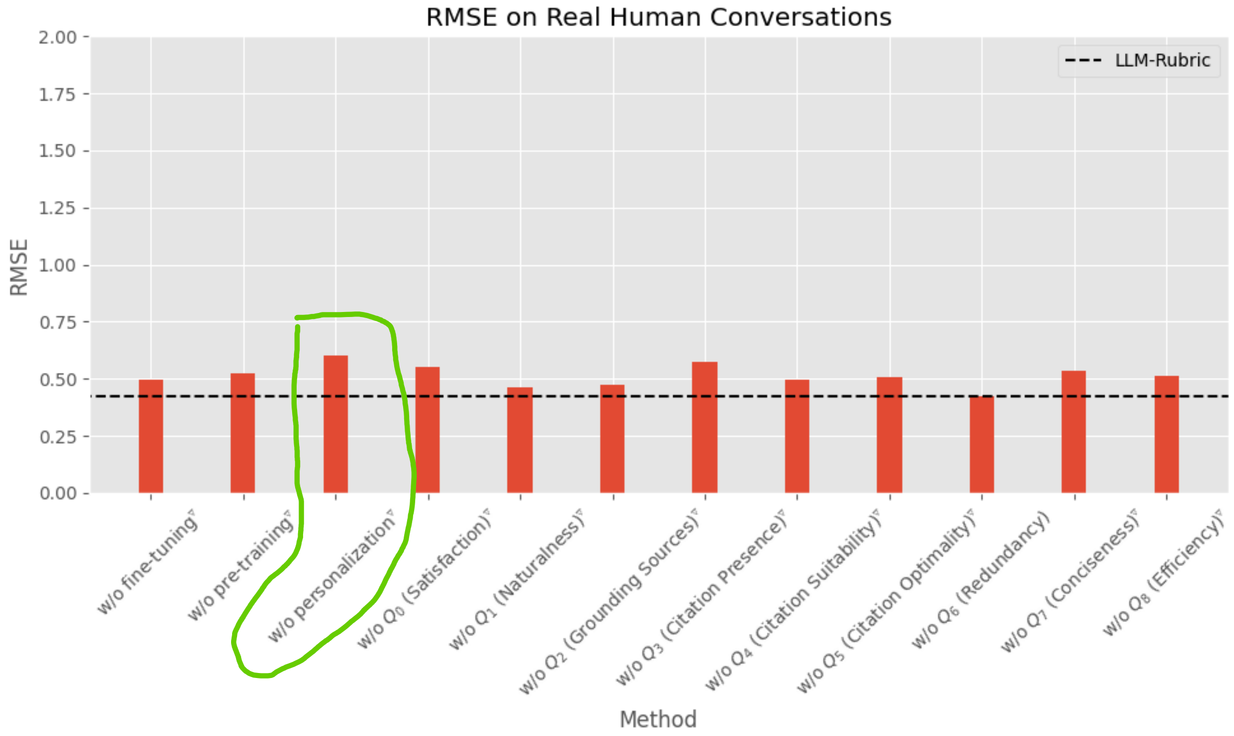
Ablation Studies



Dropping any auxiliary task (except Q_6) leads to stat. sig. drops in performance for predicting Q_0.

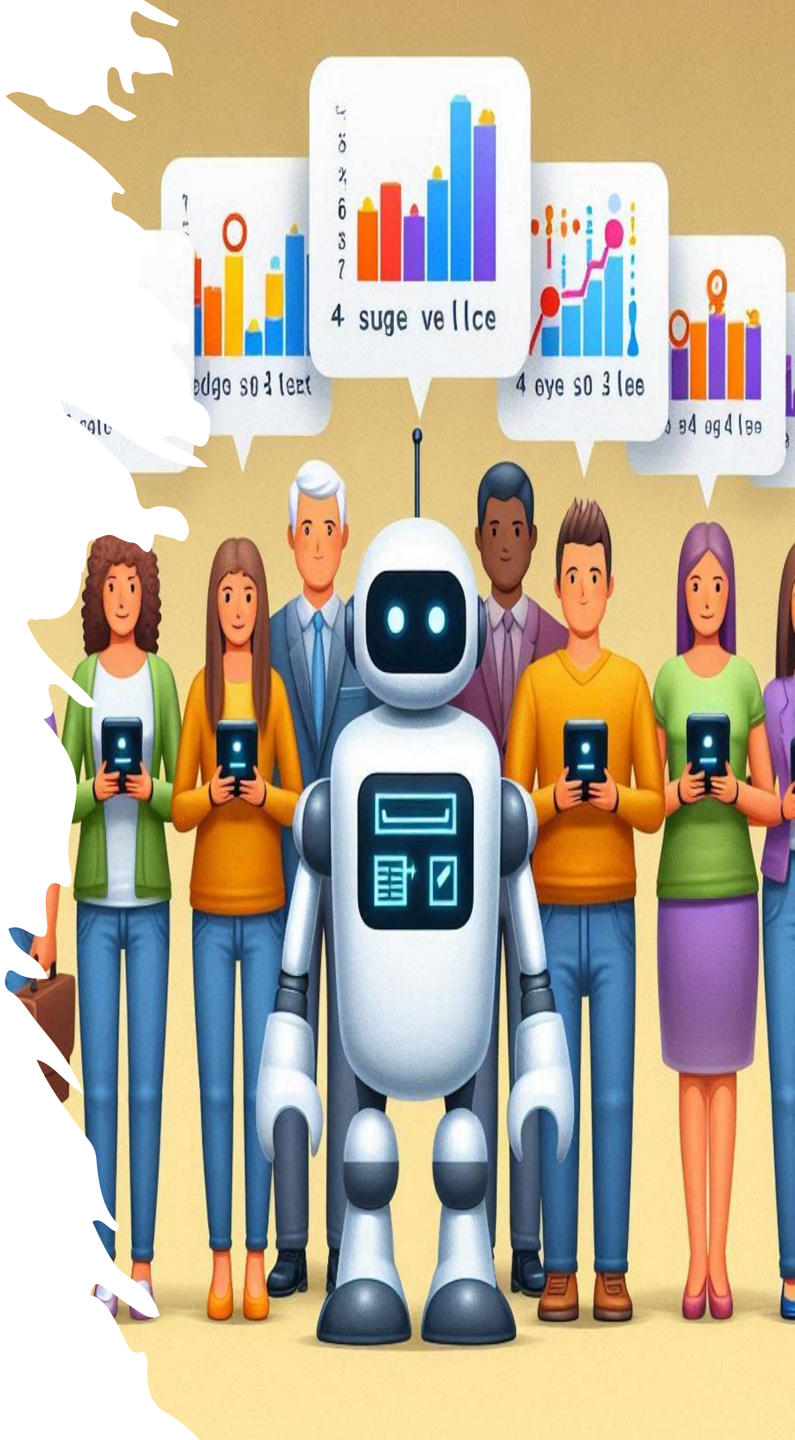


Ablation Studies

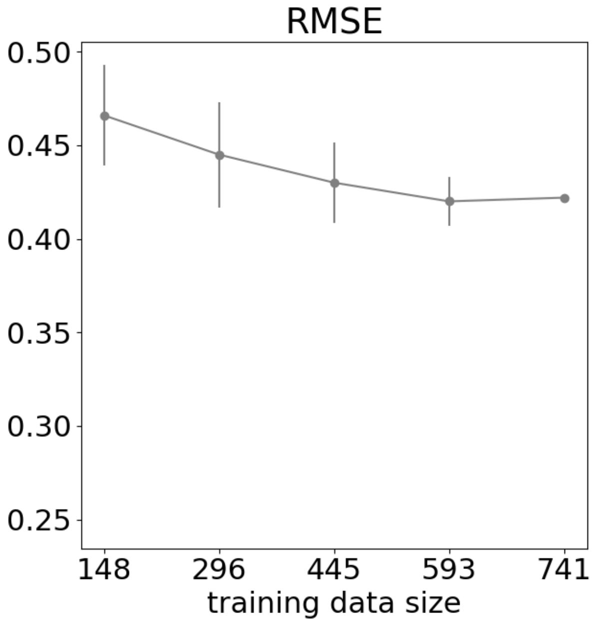


Dropping any auxiliary task (except Q_6) leads to stat. sig. drops in performance for predicting Q_0.

Dropping personalization has a larger impact on model accuracy than removing any individual rubric prediction.



How much labeled data is needed?



Training on random subsamples of data,
and reporting RMSE on test set
(error bars show ± 1 standard deviation).

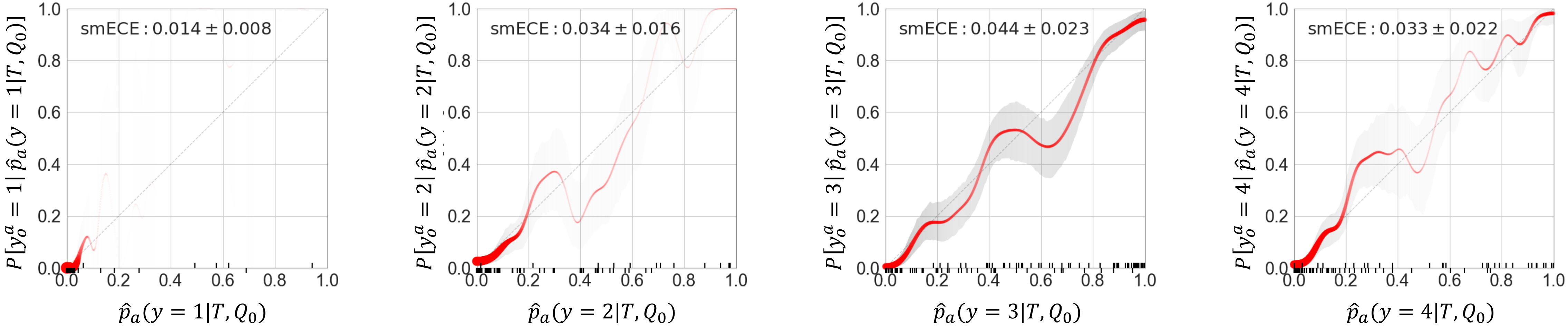
LLM-Rubric converges by roughly 80% of the training data
(593 judgements, ~24 annotations per judge, 30 judges total).

Future Work

- Adaptive rubric selection, choosing the next evaluation question to maximize the expected information gain
- Identifying difficult conversations for collecting more annotations
- Identifying interesting disagreements among judge populations
- Selecting / ranking LLM dialogue outputs to maximize a judge's rating on a specific dimension

Future work requires that LLM-Rubric be well-calibrated. Fortunately, …

LLM-Rubric is well-calibrated



Plots show $\hat{p}_a(y \mid T, Q_0)$ (x-axis) vs $P[y_0^a = y | \hat{p}_a(y \mid T, Q_0)]$ (y-axis). A well-calibrated model would have a line across the diagonal.

Interpret a point as when LLM-Rubric predicts a rating with x% probability, the probability that the prediction is correct is y%.

Plots are smoothed by density of data points (thickness of red line).
Smoothed Expected Calibration Error (smECE) is the density weighted difference in absolute value of the red line from the diagonal.

Plots generated using the `relplot` package from Błasiok and Nakkiran, 2023. [Smooth ECE: Principled Reliability Diagrams via Kernel Smoothing.]

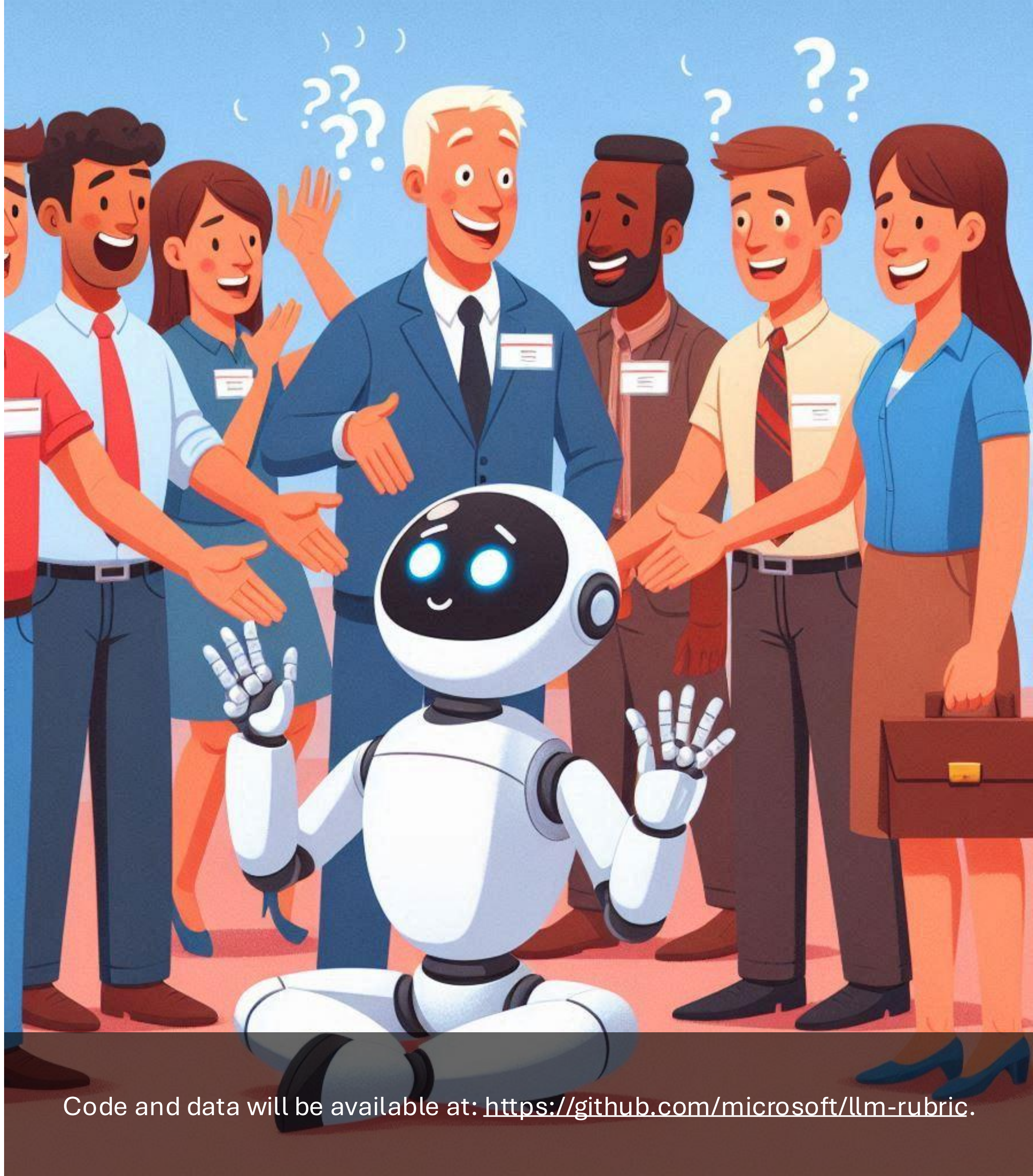
Conclusion

With **LLM-Rubric** we can:

- align LLMs with a judge pool on subjective annotation tasks
- achieve better evaluation accuracy than if we try to collapse human judgements

We also get a **well-calibrated model** of our judge pool that can be used to:

- scale up evaluation to large quantities of text
- enable deeper analysis of human judge preferences and ratings



Code and data will be available at: https://github.com/microsoft/llm-rubric.