EVALUATING MODEL CAPABILITIES AFTER BENCHMARK SATURATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Rapid advances in AI capabilities are causing benchmarks to saturate. Once several models approach perfect scores, a benchmark loses its power to differentiate between them, and with it its utility for informing AI safety cases and frontier policies. Yet benchmark items produce far more information than the scores we typically extract from them: a full transcript of *how* each model solved the task. We investigate whether these transcripts can be used to assess model capabilities even after benchmark scores have saturated. We simulate saturation retrospectively using historical benchmark and model releases, and test whether protocols that access only correctly solved transcripts from saturated benchmarks can predict capabilities measured on held-out, unsaturated items. We evaluate representative contemporary approaches for extracting information from model transcripts: pairwise judging, single-transcript absolute scoring, and predictions from deterministic transcript features or embeddings. We find that saturated transcripts retain capability information beyond benchmark accuracy. Amongst our evaluated protocols, pairwise judging performs best, reducing prediction error by $5.3\times$ compared to a baseline using only saturated benchmark scores. This capability-relevant information is distributed across transcript sections, but concentrated in specific channels of information such as reasoning and tool-calls. However, the predictive power of protocols does not always generalise across distribution shifts – for example by underestimating the capabilities of the first reasoning models introduced in early 2025. Saturated benchmarks thus retain capability-predicting signal in their task transcripts, but extending the lifetime of existing benchmarks requires methods designed to extract these signals robustly.

1 INTRODUCTION

Benchmarks use a set of tasks and an evaluation protocol to estimate what an artificial intelligence system can do within a specific domain, enabling researchers to track the rate of progress. They play an important role in frontier AI safety, where evaluations help identify systems that may require stronger safeguards and inform governance decisions in domains such as biochemical applications, cybersecurity, and autonomous research (OpenAI, 2025; Anthropic, 2026; Flynn et al., 2025).

As models improve, however, their scores eventually converge near the ceiling (Bowman & Dahl, 2021; Vania et al., 2021). At this point, the benchmark is *saturated* and can no longer distinguish between models. Benchmark saturation has already been shown to affect current evaluations, with a recent study finding signs of saturation on nearly half of 60 widespread LLM benchmarks (Akhtar et al., 2026).

Current approaches to benchmark saturation largely rely on constructing harder benchmarks or continually updating existing ones. Researchers have introduced benchmarks such as Humanity's Last Exam and MMLU-Pro after earlier evaluations became too easy for frontier models (Center for AI Safety et al., 2026; Wang et al., 2024), while dynamic and continuously updated benchmarks aim to maintain discriminative power as models improve (Kiela et al., 2021; White et al., 2025; Jain et al., 2025; Ying et al., 2024). However, the time and cost required for constructing and verifying tasks make it difficult for continual benchmark updates to match frontier model progress (Ying et al., 2024). This creates a need to extract more information from existing benchmarks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

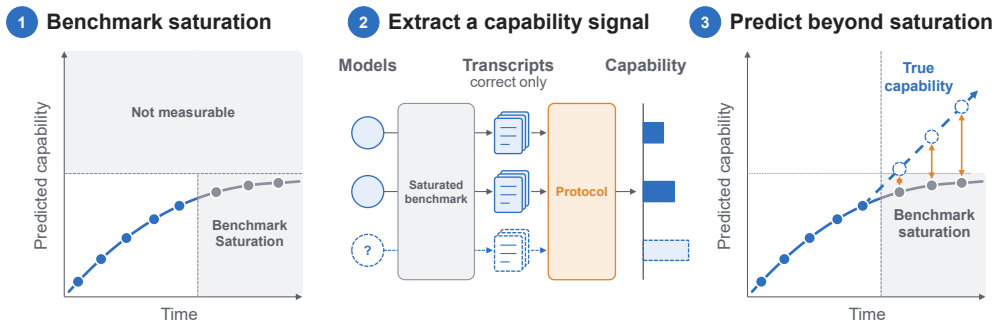


Figure 1: **Overview of our evaluation setting.** As benchmark scores saturate, it becomes infeasible to distinguish the capabilities of models. Using earlier models as references, protocols predict target model capability on harder, unsaturated items, using as information *how* the models solve the tasks.

Benchmark scores compress model behaviour into a single scalar value, losing much of the information contained in evaluation transcripts. Models that appear identical under accuracy may differ in how they solve the same items. Indeed, prior work has shown that solutions reaching the same correct answer can differ substantially in their intermediate reasoning (Uesato et al., 2022; Lightman et al., 2024), and that transcripts can reveal intermediate progress (Kapoor et al., 2026) and differences in solution quality (Kirgis et al., 2026) that are not reflected in pass/fail outcomes. Whether such transcript-level signals can be used to estimate capability once a benchmark has saturated, however, remains untested. We therefore study whether transcript-level differences on saturated items retain capability signals that can be used to predict model capability on harder tasks beyond the benchmark's saturated scores (Figure 1).

We instantiate this question by assessing different methods for estimating capabilities from saturated benchmark transcripts. These methods were selected to represent contemporary approaches for extracting information from model outputs: deterministic transcript features (Bitton et al., 2025), transcript embedding, single-transcript scoring (Liu et al., 2023; Kim et al., 2024), and pairwise judge comparisons (Zheng et al., 2023). We then evaluate whether these approaches recover capability information once benchmark accuracy no longer reliably distinguishes between models.

Our main findings are as follows:

1. **Transcripts from correctly solved saturated items retain capability information beyond benchmark accuracy.** Pairwise LLM judging recovers the strongest signal, reducing predictive error 5.27× compared to a benchmark accuracy-based baseline.
2. **Capability-relevant information is distributed across transcripts and channels.** Removing any single third of a transcript does not fully impair predictions, suggesting that useful capability information is spread throughout the response. However, different channels of information matter in different domains: hiding model reasoning increases prediction error by roughly 3–4× in coding, cybersecurity, and science benchmarks, while removing tool use increases prediction error by about 4× in cybersecurity but has minimal effect in coding.
3. **Current capability-estimation protocols are not robust to all distribution shifts.** For example, pairwise judges from late 2024 could not reliably estimate the capabilities of the first reasoning models introduced in early 2025. This suggests that protocols can fail when new models bring paradigm shifts.

2 Problem Setting

We consider a setting in which model capabilities improve faster than new benchmarks can be developed and validated. When differences in benchmark scores can no longer be used to separate models of different capabilities, we consider the benchmark to be *saturated*.

Let $\mathcal{M}_{\text{ref}}$ denote a set of reference models for which capability estimates are available. Evaluating each reference model $r \in \mathcal{M}_{\text{ref}}$ produces a set of transcripts $\mathcal{T}_r$ that record its behaviour on the benchmark. Together with each reference model's capability θ_r, these transcripts form the reference

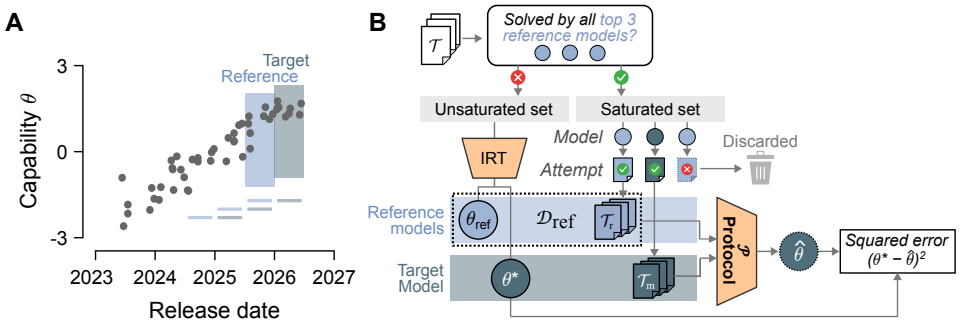


Figure 2: **Overview of the prediction task and evaluation methodology. (A)** models are grouped into consecutive semesters; for each target period, the preceding semester serves as the reference set. **(B)** Items are split by whether the top-3 reference models answer correctly: easy items form the saturated set, of which only correctly-solved transcripts $\mathcal{T}$ are provided to a capability-extraction protocol $\mathcal{P}$ to produce a prediction $\hat{\theta}$; the remaining items form the unsaturated set, on which an IRT model is fit to obtain the ground-truth capability θ^* for both references and targets.

dataset $\mathcal{D}_{\text{ref}}$:

$$\mathcal{D}_{\text{ref}} = \{(\mathcal{T}_r, \theta_r) : r \in \mathcal{M}_{\text{ref}}\}.$$

We then consider a set of target models $\mathcal{M}_{\text{target}}$ whose capabilities are to be estimated. For each target model $m \in \mathcal{M}_{\text{target}}$, we generate its set of transcripts $\mathcal{T}_m$ on the same saturated benchmark, but its capability θ_m is unknown. The problem is to estimate the capability of each target model from the reference data and the target model's transcripts. A *protocol* $\mathcal{P}$ therefore produces:

$$\mathcal{P}\left(\mathcal{D}_{\text{ref}}, \mathcal{T}_m\right) \to \hat{\theta}_m, \qquad m \in \mathcal{M}_{\text{target}}.$$

3 Experimental Methods

We instantiate our setting retrospectively using historical model releases and benchmark results. For each prediction task, reference models precede the target period. We additionally split benchmark items into an observed *saturated set* and a harder *unsaturated set*. Protocols receive only transcripts from the saturated set, while the unsaturated set is used exclusively to measure the ground truth target capability θ^*_m that the protocols attempt to predict.

3.1 Evaluation Setup

Reference and target periods. We divide model releases into consecutive semesters. For each prediction task, the *reference set* consists of models available in the preceding semester $\mathcal{M}_{t-1}$, while the *target set* consists of models released during the subsequent semester $\mathcal{M}_t$. Our main evaluation uses two target periods: late 2025 and early 2026, with seven or eight reference models and eight target models in each period (Figure 2A).

Determining the saturation boundary. For each target period, we identify the three highest-capability models amongst the reference set. We define an item as belonging to the *saturated set* if all three models answer it correctly. The remaining items form the *unsaturated set*.

Prediction target. To measure capability beyond the saturated portion of the benchmark, we use item response theory (IRT), which jointly estimates model capability and item difficulty from benchmark outcomes (Baker, 1998). We fit a local IRT model to the reference and target models on the unsaturated set, while keeping each item's difficulty parameter b_i fixed across semesters. The resulting capability estimate θ^* for each target model is used as the prediction target for the protocols. We provide the full IRT fitting procedure in Appendix A.2.

Transcripts. Our primary experiments test whether model capabilities can be inferred from *how* models solve saturated tasks, rather than from whether they solve them correctly. We therefore provide each protocol only correctly solved transcripts from saturated items, thus removing correctness as a source of predictive signal. Unless otherwise stated, protocols receive the complete model response. The full framework is depicted in Figure 2B.

Table 1: **Evaluation domains and benchmarks.** Benchmarks used in each capability domain.

Domain	Benchmarks
Science	GPQA-Diamond (Rein et al., 2024), ARC-Challenge (Clark et al., 2018), HLE-Gold [Science] (Zhai et al., 2026)
Mathematics	MMLU-Pro [Maths] (Wang et al., 2024), AIME 2024 (Jia, 2024), AIME 2025, AIME 2026, APEX 2025, APEX Shortlist (Dekoninck et al., 2026), HLE-Gold [Maths]
Coding	AgentBench OS (Liu et al., 2024), SWE-Bench (Jimenez et al., 2024), Terminal-Bench 2.1 (Merrill et al., 2026)
Cybersecurity	CyBench (Zhang et al., 2025), InterCode (Yang et al., 2023)

Domains. To assess generalisability, we evaluate four domains: SCIENCE, MATHS, CODING, CYBER. Each domain combines benchmarks spanning a range of item difficulties, allowing us to distinguish between saturated and unsaturated items. Table 1 provides a list of the benchmarks used to represent each domain.

Model selection. Our main experiments utilise the complete model output used during inference. For earlier models, this consists of the standard response transcript. For later reasoning models however, reasoning traces are an important part of the transcript but are not consistently available through closed-source model providers. We therefore primarily use open-weight reasoning models in periods where reasoning models are included in the evaluation, while in earlier periods closed-source non-reasoning models are retained. We evaluate generalisation to closed-source reasoning models separately in Section 4.3. The full model list is provided in Appendix B.1.

Metrics. For a protocol $\mathcal{P}$ and target model m, we measure prediction error using squared error: $\epsilon_m = (\hat{\theta}_m - \theta^*_m)^2$, where θ^*_m is the measured target capability and $\hat{\theta}_m$ is the prediction.

3.2 PREDICTION PROTOCOLS

We compare several contemporary approaches for extracting information from transcripts, and adapt them to estimate capabilities from the observations available in our setting. Table 2 provides a summary of each of the protocols, while full implementation details are provided in Appendix C.

Table 2: **Prediction protocols.** All non-baseline protocols extract transcript-level signals and calibrate them to target capability θ^* using the reference models.

Protocol	Signal extracted
Baseline protocol	–
Transcript features	Deterministic properties (e.g., transcript length and prose features)
Transcript embeddings	Text-embedding representation of a single transcript
Absolute scoring	Absolute scalar score given to a single transcript
Pairwise judging	Relative capability evidence from comparisons between two transcripts

Baseline protocol. The baseline represents the capability estimate available using benchmark scores alone under saturation. Because the evaluated model solves all observed items correctly by construction, its accuracy provides no distinguishing signal. Consequently, this baseline simply assigns every target model the mean capability of the reference models.

Statistical tests. To assess the significance of a protocol's performance, we account for both prediction accuracy and uncertainty induced by the finite reference set through a comparison against the aforementioned baseline protocol (Section 4.1). For each protocol, we estimate prediction uncertainty from leave-one-model-out errors on the reference models and use this to construct a Student t distribution around each target-model prediction. Protocols are therefore penalised both for inaccurate predictions and for predictions that are unstable across reference models.

We compare each protocol with the baseline using the log-likelihood ratio (LLR) of the target model's observed capability under their respective predictive distributions. We then test whether the mean LLR across target models differs from zero using one-sample t-tests. Full details are pro-

vided in Appendix A.1. For within-protocol comparisons in Sections 4.2 and 4.3, we instead apply paired t-tests directly to target-model squared errors.

4 RESULTS

To understand whether and how transcripts from saturated benchmarks can be used to estimate model capability, we ask three questions: **(1)** Do correctly solved transcripts from saturated benchmarks contain any capability-predicting signal beyond raw accuracy? **(2)** Where in the transcript is this signal located? **(3)** Can we reliably extract this signal to predict capabilities across distribution shifts such as time or models?

4.1 SATURATED BENCHMARK TRANSCRIPTS CONTAIN CAPABILITY INFORMATION

We first ask whether transcripts from saturated benchmark items contain information about capabilities. To do so, we assessed whether any of our protocols could recover a model's capabilities from only its correctly-answered transcripts and a reference dataset (the capabilities and correctly-answered transcripts of historical models). Figure 3 compares the prediction error of the four protocols over the 16 target models of 2025-II and 2026-I, each averaged over the four domains.

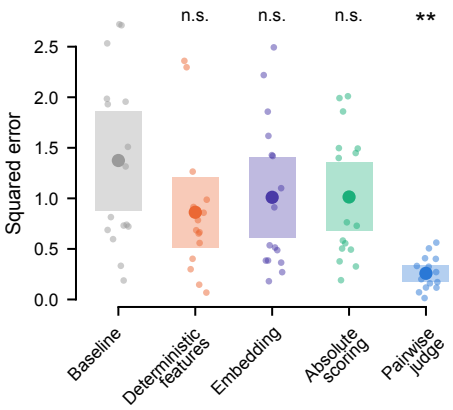


Figure 3: **Pairwise judges extract capability-predicting signal from saturated transcripts.** Prediction performance of each protocol for 16 target models (squared error: lower is better). Each point is one target model; circle and box: mean and 95% CI. **: p<.01, one-sample t-test of LLR against the baseline protocol; *n.s.*: not significant.

Pairwise judging predicted target capabilities most accurately (with the lowest error amongst our protocols: MSE=0.26, 95% CI: [0.18, 0.34]). Its predictions also placed more likelihood on the targets' true capability than the baseline (mean LLR=0.68, one-sample t-test: $p = .005$; see Appendix D.1). This result was not specific to our choice of judge (Appendix D.3), and tended to improve with the number of items and reference models judged in most tasks (Appendix D.6, D.7).

Other protocols were less reliable. While all tended to yield lower errors than the baseline (deterministic transcript features: 0.86 [0.52, 1.21]; transcript embeddings: 1.01 [0.61, 1.41]; absolute scoring: 1.01 [0.67, 1.35]), none placed significantly more likelihood on the true capability than the baseline.

Feature-based predictors did not perform significantly better than the baseline, suggesting that the capability signal is not primarily carried by simple properties such as response length or surface-level writing style. In contrast, pairwise judging substantially reduced the prediction error, indicating that it extracts information beyond such high-level features.

4.2 CAPABILITY SIGNALS ARE DISTRIBUTED THROUGHOUT TRANSCRIPTS

Having found that saturated benchmark transcripts contain capability-relevant signals, we next ask *where* this signal lies. This is practically important because model providers may not always make the full transcript available for closed-source models, and some protocols may have restrictions on how long a transcript they can judge efficiently. We focus on pairwise judging, the strongest protocol in Section 4.1, and report the same analyses for the rest in Appendix D.9 and D.10.

Capability-relevant information persists across transcript sections. We first ask whether capability-relevant information is concentrated in particular parts of the transcript, which matters when limited context windows force protocols to observe only part of it. We divide each transcript into thirds and run two complementary tests: whether any third is necessary, by removing it and predicting from the remaining two, and whether any third is sufficient, by predicting from that third alone. When a third is removed, we measure the increase in prediction error relative to that obtained

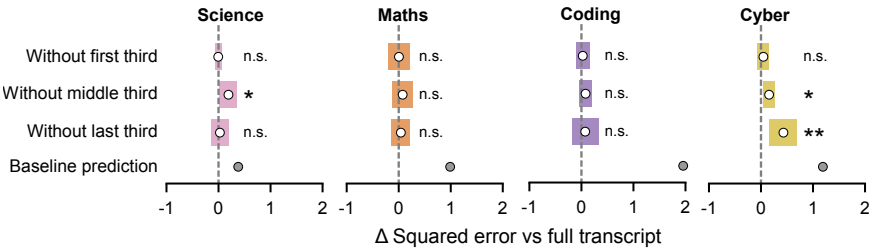


Figure 4: **Effect of removing transcript sections on prediction error.** Change in squared prediction error of pairwise judging when the first, middle, or last third of every transcript is removed, relative to the complete transcript (*dashed line*). *Grey points* show the baseline prediction. *: $p < .05$; **: $p < .01$

from the complete transcript; when a third is kept alone, we test whether it can still outperform the baseline protocol.

In MATHS and CODING, no single third was necessary for prediction. Removing a transcript section significantly increased squared error only in CYBER (middle third: $+0.16$, $p = .010$; last third: $+0.43$, $p = .003$), and less robustly in SCIENCE (middle third: $+0.19$, $p = .032$) (Figure 4) compared to the complete transcript. These increases were small compared to the gap between the complete transcript and the baseline. Similarly, each third alone was often sufficient to outperform the baseline, suggesting that the signal is redundantly present throughout the transcript (Appendix D.8).

Reasoning and tool interactions carry capability signal in some domains. We next ask whether capability-relevant signal is concentrated in specific channels of information: the model's reasoning, its messages, or its tool interactions. To test this, we remove each channel in turn and compare the prediction error with that obtained from the complete transcript.

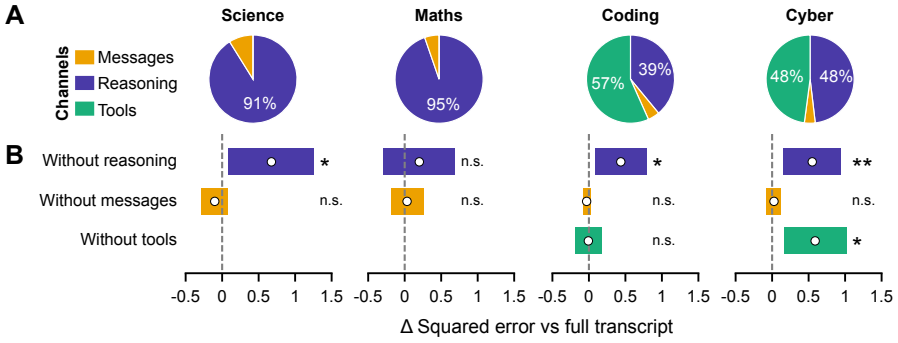


Figure 5: **Effect of removing transcript channels on prediction error. A.** Share of transcript in each channel per domain. **B.** Change in prediction error of pairwise judging when one channel is removed from every transcript, relative to the complete transcript (dashed line). Each point is the mean over a domain's target models and the box its 95% paired t-interval. *: $p < .05$, **: $p < .01$, paired t-test of squared error between channel-removed and full transcripts.

Reasoning and tool interactions carried capability-relevant signal in some domains, whereas messages did not. Removing reasoning increased prediction error in CODING, CYBER, and SCIENCE (MSE changes: 0.18 to 0.62; 0.20 to 0.75; 0.27 to 0.94, all $p < .05$), but not significantly in MATHS (0.31 to 0.51, $p = .40$). Removing tool interactions increased error in CYBER (0.20 to 0.79, $p = .01$) but had no effect in CODING (0.18 to 0.17, $p = .93$). Removing messages did not significantly affect prediction error in any domain (Figure 5B).

These effects are not explained by channel size alone. Tool interactions make up about half of both CODING and CYBER transcripts, and reasoning over 90% of both MATHS and SCIENCE transcripts (Figure 5A), yet removing them only degraded prediction in CYBER and SCIENCE, respectively.

4.3 GENERALISATION OF CAPABILITY PREDICTIONS

Our results so far show that saturated benchmark transcripts contain capability-predicting signals that pairwise judges can extract. We demonstrate this in a setting where protocols predict capabilities of newer target models one semester ahead. This prediction may involve several kinds of generalisation: newer models may be trained differently, reach higher capabilities, or solve tasks in different ways. In practice, protocols may also need to generalise beyond a single semester – for example, by looking further ahead in time, or predicting capabilities in domains they were not calibrated on. We therefore test how prediction performance changes across time, model classes, capability levels, and domains.

Prediction reliability varies across domains and historical periods. We first test whether protocols' performances are stable across time. We repeat predictions retrospectively across earlier target periods, each time using models from the preceding period as the reference set. We back-test SCIENCE and MATHS to early 2024; CODING and CYBER begin in late 2025 because earlier models do not provide sufficient coverage of these agentic benchmarks.

Prediction error varies substantially across historical periods for all protocols (Figure 6, Appendix D.2). Pairwise judging performs consistently well in CODING and CYBER, but is less reliable in MATHS and SCIENCE, where it fails to improve on the baseline in four of ten evaluated periods (MATHS: early 2024, early 2025, and early 2026; SCIENCE: late 2024). Other protocols show similar temporal instability. The most pronounced failures occur in early 2025, particularly for deterministic transcript features, where errors surge in both SCIENCE and MATHS.

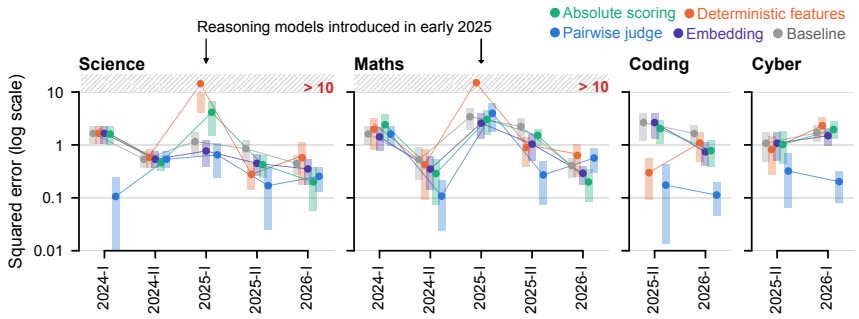


Figure 6: **Prediction errors vary substantially across historical target periods.** Prediction error for each protocol when calibrated on the preceding reference population and evaluated on the 8 models of each target period. Values above 10 are drawn in the hatched band.

Pre-reasoning era judges underestimate the capability of reasoning models. The early-2025 failures coincide with the introduction of reasoning models. This marks a deviation from reference models, accompanied by a sharp increase in mathematical capabilities. We therefore use this period as a case study to assess how this influences prediction performance for pairwise judging.

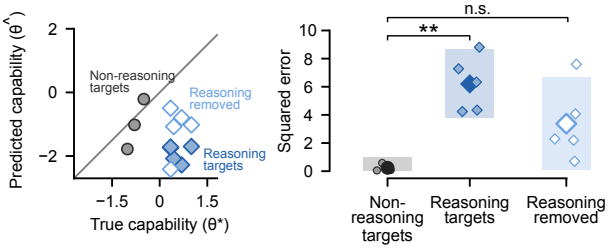


Figure 7: **Pairwise judges underestimate maths capabilities of early 2025 reasoning models. Left,** Predicted vs true capability for the three non-reasoning targets, five reasoning targets, and reasoning targets judged with reasoning withheld. *Diagonal*: perfect prediction. **Right,** Prediction error per group (means ± 95% t-intervals). **: $p < .01$, two-sample t-test.

Prediction error is lower for the three non-reasoning targets than for the five reasoning targets (MSE 0.24 vs. 6.20; $p = .002$; Figure 7). This was because judges underestimated the capabilities of reasoning models, with predictions falling 2.1–3.0 θ units below their measured values. This failure is not fully explained by the additional reasoning text: removing the reasoning tokens from the

transcripts only approximately halves the error to 3.37 ($p = .16$): still higher than for non-reasoning targets. These results suggest that the shift to reasoning models changes the relationship between transcript behaviour and capability in ways that the pairwise judging protocol is unable to capture.

Pairwise judging predictions generalise to more capable models. Next, we test whether prediction quality degrades as target model capabilities exceed the reference models used for calibration. For each target, we measure how far its capability exceeds that of the strongest reference model. Pairwise judging remains stable as this gap increases (Figure 8): the slope of squared error against capability gap is indistinguishable from zero in every domain (SCIENCE: -0.02, MATHS: 0.10, CODING: -0.06, CYBER: -0.12; all $p \geq .45$). All other protocols show increasing error as targets move beyond the reference capability range.

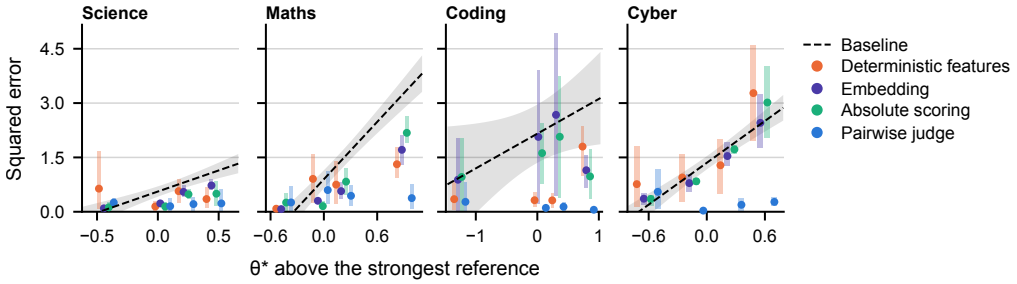


Figure 8: **Pairwise judging generalises across target model capabilities.** Prediction error as target capabilities exceed that of the strongest reference model. Targets are grouped into four equal-count bins. Error increases with capability distance for all protocols except pairwise judging.

Pairwise judge predictions transfer to closed-source models. In practice, capability estimation is especially important for frontier models, which are often closed-source. Although pairwise judging generalises well to more capable open-weight models (Figure 8), closed-source frontier models introduce two additional shifts: they may differ in training and behaviour, and their reasoning traces are typically unavailable. The latter is particularly relevant, since removing reasoning has been shown to notably reduce capability signal (Section 4.2). We therefore evaluate six closed-source targets on CODING (Claude Opus 4.7, Opus 4.8, Fable 5, GPT-5, GPT-5.6 Sol, and Gemini 3.5 Flash), and compare capability predictions on these targets with those of open-weight targets from the most recent semester (early 2026). We evaluate protocols with reasoning hidden for all models.

Performance transferred to closed-source targets to varying degrees (Figure 9). Pairwise judging beat the baseline protocol on all six closed-source targets and all eight open-weight targets. No other protocol performed significantly better than baseline for either model group.

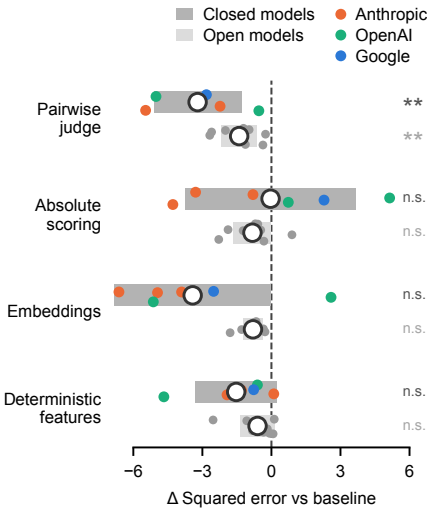


Figure 9: **Pairwise judge predictions generalise to closed-source targets.** Change in squared error from each target's baseline error (below zero: better than the baseline protocol) for six closed-source targets (coloured) and eight open-weight targets from the same semester.

Pairwise judge predictions are strongest in CODING and CYBER. The analysis in Section 4.1 averages performance across domains, potentially obscuring domain-specific differences between protocols. We therefore examine protocol performance separately by domain (Table 3). Pairwise judging achieves the lowest prediction error amongst protocols in all four domains, but its improvement over the baseline is only significant in the agentic domains (CODING: mean LLR 1.28, $p < .001$; CYBER: 0.94, $p = .001$). The remaining protocols show limited generalisation across domains. We report further analysis of domain dependence in Appendix D.4 and cross-domain transfer results in Appendix D.5.

Table 3: **Prediction error per protocol and domain.** MSE per domain. Lowest prediction errors per domain are bolded. */**/***: $p < .05/.01/.001$, one-sample t-tests on LLR.

Protocol	Pooled	SCIENCE	MATHS	CODING	CYBER
Baseline protocol	1.37	0.65	1.32	2.14	1.39
Deterministic features	0.86	0.43	0.76	0.70*	1.57
Embedding	1.01	0.40**	0.66	1.69	1.29**
Absolute scoring	1.01	0.31	0.85	1.41	1.48
Pairwise judging	**0.26****	**0.21**	**0.42**	**0.14*****	**0.26****

5 RELATED WORK

Benchmark saturation and evaluation design. Previous works on benchmarks report the impact of score saturation and the downstream limit to discrimination amongst strong models (Bowman & Dahl, 2021; Vania et al., 2021; Akhtar et al., 2026). Attempts to resolve this have typically involved introducing new challenging versions of existing benchmarks (Hendrycks et al., 2021; Suzgun et al., 2023; Rein et al., 2024; Wang et al., 2024; Center for AI Safety et al., 2026). Other lines of work also explore the idea of dynamic evaluation, which instead uses adversarial data collection or some continuously updated test set to maintain difficulty over time (Kiela et al., 2021; White et al., 2025; Jain et al., 2025), but costs for running these methods are much higher than alternatives.

Information from model responses. Prior works on transcript-analysis extract capability-relevant signals through pairwise comparison and rubric-based scoring (Zheng et al., 2023; Liu et al., 2023; Kim et al., 2024), process-level reasoning assessment (Uesato et al., 2022; Lightman et al., 2024), and confidence-based prediction of correctness (Lin et al., 2022; Kadavath et al., 2022; Tian et al., 2023). These approaches point to model transcripts as a promising source of information. To our knowledge, no previous work has explored whether correctly solved transcripts from saturated benchmarks can signal capabilities.

Capability estimation and prediction. Previous works use Item Response Theory (IRT) for latent ability estimation, item calibration, and test-set comparison (Lalor et al., 2016; 2019; Rodriguez et al., 2021; Vania et al., 2021; Zhou et al., 2026). By accounting for differences in item difficulty, IRT can be used to provide a more structured capability estimate than accuracy (Baker, 1998). Additional works on adaptive testing and representative item selection focus on a different direction of reducing evaluation cost by identifying informative subsets of tasks (Zhuang et al., 2025; Maia Polo et al., 2024; Fonseca Rivera et al., 2026), although estimation reliability depends on the number and distribution of evaluated models (Jiang et al., 2026). Finally, forecasting work predicts benchmark performance from signals on other tasks (Owen, 2024; Ruan et al., 2024).

6 DISCUSSION

We explore whether benchmarks can remain useful for evaluating model capabilities, even after final scores are saturated. We show that transcripts from saturated benchmarks retain capability-predicting information. Amongst the protocols we assessed, pairwise judging was able to extract this information most reliably, outperforming predictions from saturated benchmark scores in predicting performance on more difficult held-out tasks. More broadly, these results suggest that it is possible to extend the usefulness of saturated benchmarks for tracking model capabilities: a critical capacity for informing safety and governance decisions, when constructing harder evaluations cannot keep pace with model progress.

Demonstrating that capability information is present in transcripts is of course a first step. A useful capability-estimation protocol must also extract this information reliably when the relationship between transcript behaviour and capability changes; our current protocols do not yet meet this requirement. Prediction performance varies substantially across time, and the transition to reasoning models in early 2025 provides a clear failure case in which judging-based protocols systematically underestimate the capabilities of the new model class. Future work should focus not only on extracting more signal from transcripts, but on designing protocols that are robust to distribution shifts.

AI USE STATEMENT

We used generative AI tools to assist with editing and restructuring the manuscript along with code development and debugging. All AI-assisted research code and manuscript text were reviewed and verified by the authors. The authors take responsibility for the final content of this work, including all claims and artefacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the full experimental code used in the main text and appendix in an anonymised code repository in https://anonymous.4open.science/r/scalable-capability-evaluation. Implementation details for each prediction protocol and complete lists of evaluated models and benchmarks are provided in the appendix.

REFERENCES

Mubashara Akhtar, Anka Reuel, Prajna Soni, Sanchit Ahuja, Pawan Sasanka Ammanamanchi, Ruchit Rawal, Vilém Zouhar, Srishti Yadav, Chenxi Whitehouse, Dayeon Ki, Jennifer Mickel, Leshem Choshen, Marek Šuppa, Jan Batzner, Jenny Chim, Jeba Sania, Yanan Long, Hossein A. Rahmani, Christina Knight, Yiyang Nan, Jyoutir Raj, Yu Fan, Shubham Singh, Subramanyam Sahoo, Eliya Habba, Usman Gohar, Siddhesh Pawar, Robert Scholz, Arjun Subramonian, Jingwei Ni, Mykel Kochenderfer, Sanmi Koyejo, Mrinmaya Sachan, Stella Biderman, Zeerak Talat, Avijit Ghosh, and Irene Solaiman. When AI benchmarks plateau: A systematic study of benchmark saturation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.16763.

Anthropic. Anthropic's responsible scaling policy. https://www.anthropic.com/responsible-scaling-policy, 2026. URL https://www.anthropic.com/responsible-scaling-policy. Version 3.4, effective July 8, 2026. Accessed September 17, 2026.

Frank B Baker. Item response theory: Parameter estimation techniques. *Applied Psychological Measurement*, 22(1):89–91, 1998.

Yehonatan Bitton, Elad Bitton, and Shai Nisan. Detecting stylistic fingerprints of large language models. *arXiv preprint arXiv:2503.01659*, 2025. URL https://arxiv.org/abs/2503.01659.

Michael Borenstein, Larry Hedges, Julian Higgins, and Hannah Rothstein. *Random-Effects Model*, chapter 12, pp. 69–75. John Wiley & Sons, Ltd, 2009. ISBN 9780470743386. doi: 10.1002/9780470743386.ch12. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470743386.ch12.

Samuel R. Bowman and George E. Dahl. What will it take to fix benchmarking in natural language understanding? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4843–4855, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.385. URL https://aclanthology.org/2021.naacl-main.385/.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2334029.

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649(8099):1139–1146, January 2026. doi: 10.1038/s41586-025-09962-4. URL https://www.nature.com/articles/s41586-025-09962-4.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs. *arXiv preprint arXiv:2605.00674*, 2026. URL https://arxiv.org/abs/2605.00674.

Four Flynn, Helen King, and Anca Dragan. Strengthening our Frontier Safety Framework. Google DeepMind blog, September 2025. URL https://deepmind.google/blog/strengthening-our-frontier-safety-framework/. Published September 22, 2025. Updated April 17, 2026 (Frontier Safety Framework v3.1).

Joshua Fonseca Rivera, Neil Shah, David Demitri Africa, and Konstantinos Voudouris. Item Response Theory for AI safety. arXiv preprint arXiv:2608.05086, 2026. URL https://arxiv.org/abs/2608.05086.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Maxwell Jia. AIME 2024 dataset. Hugging Face dataset, December 2024. URL https://huggingface.co/datasets/Maxwell-Jia/AIME_2024.

Han Jiang, Sunbeom Kwon, Jinwen Luo, Ziang Xiao, and Susu Zhang. Can we trust Item Response Theory for AI evaluation? arXiv preprint arXiv:2607.15190, 2026. URL https://arxiv.org/abs/2607.15190.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. arXiv preprint arXiv:2207.05221, 2022. URL https://arxiv.org/abs/2207.05221.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, Franck Ndzomga, Dheeraj Oruganty, Sophie Luskin, Kangheng Liu, Botao Yu, Amit Arora, Dongyoon Hahm, Harsh Trivedi, Huan Sun, Juyong Lee, Tengjun Jin, Yifan Mai, Yifei Zhou, Yuxuan Zhu, Rishi Bommasani, Daniel Kang, Dawn Song, Peter Henderson, Yu Su, Percy Liang, and Arvind Narayanan. Holistic agent leaderboard: The missing infrastructure for ai agent evaluation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 98778–98849, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a0928f924a344aaebbb7f6cd8d56e34c-Paper-Conference.pdf.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and

Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324/.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

Peter Kirgis, Sayash Kapoor, Stephan Rabanser, Nitya Nadgir, Cozmin Ududec, Magda Dubois, JJ Allaire, Conrad Stosz, Marius Hobbhahn, Jacob Steinhardt, and Arvind Narayanan. Log analysis is necessary for credible evaluation of AI agents. *arXiv preprint arXiv:2605.08545*, 2026. URL https://arxiv.org/abs/2605.08545.

John P. Lalor, Hao Wu, and Hong Yu. Building an evaluation scale using Item Response Theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 648–657, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1062. URL https://aclanthology.org/D16-1062/.

John P. Lalor, Hao Wu, and Hong Yu. Learning latent parameters without human response patterns: Item Response Theory with artificial crowds. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4249–4259, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1434. URL https://aclanthology.org/D19-1434/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=8s8K2UZGTZ.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.03688.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, et al. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. arXiv preprint arXiv:2601.11868, 2026. URL https://arxiv.org/abs/2601.11868.

OpenAI. Our updated Preparedness Framework. https://openai.com/index/updating-our-preparedness-framework/, April 2025. URL https://openai.com/index/updating-our-preparedness-framework/. Published April 15, 2025.

David Owen. How predictable is language model benchmark performance? arXiv preprint arXiv:2401.04757, 2024. URL https://arxiv.org/abs/2401.04757.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4486–4503, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.346. URL https://aclanthology.org/2021.acl-long.346/.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0506. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/1cded4f97cf5f01a284c574110b7e3b9-Abstract-Conference.html.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. arXiv preprint arXiv:2211.14275, 2022. URL https://arxiv.org/abs/2211.14275.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. Comparing test sets with Item Response Theory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1141–1158, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.92. URL https://aclanthology.org/2021.acl-long.92/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024. URL https://arxiv.org/abs/2406.01574.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger,

and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sKYHBTAxVa.

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. Intercode: Standardizing and benchmarking interactive coding with execution feedback, 2023. URL https://arxiv.org/abs/2306.14898.

Jiahao Ying, Yixin Cao, Yushi Bai, Qianru Sun, Bo Wang, Wei Tang, Zhaojun Ding, Yizhe Yang, Xuanjing Huang, and Shuicheng Yan. Automating dataset updates towards reliable and timely evaluation of large language models. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, pp. 17106–17132, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/1e89c12621c0315373f20f0aeabe5dbe-Abstract-Datasets_and_Benchmarks_Track.html.

Weiqi Zhai, Zhihai Wang, Jinghang Wang, Boyu Yang, Xiaogang Li, Xiang Xu, Bohan Wang, Peng Wang, Xingzhe Wu, Anfeng Li, Qiyuan Feng, Yuhao Zhou, Shoulin Han, Wenjie Luo, Yiyuan Li, Yaxuan Wang, Ruixian Luo, Guojie Lin, Peiyao Xiao, Chengliang Xu, Ben Wang, Zeyu Wang, Zichao Chen, Jianan Ye, Yijie Hu, Jialong Chen, Zongwen Shen, Yuliang Xu, An Yang, Bowen Yu, Dayiheng Liu, Junyang Lin, Hu Wei, Que Shen, and Bing Zhao. Hle-verified: A systematic verification and structured revision of humanity's last exam, 2026. URL https://arxiv.org/abs/2602.13964.

Andy K Zhang, Neil Perry, Riya Dulepet, Joey Ji, Celeste Menders, Justin W Lin, Eliot Jones, Gashon Hussein, Samantha Liu, Donovan Julian Jasper, Pura Peetathawatchai, Ari Glenn, Vikram Sivashankar, Daniel Zamoshchin, Leo Glikbarg, Derek Askaryar, Haoxiang Yang, Aolin Zhang, Rishi Alluri, Nathan Tran, Rinnara Sangpisit, Kenny O Oseleononmen, Dan Boneh, Daniel E. Ho, and Percy Liang. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tc90LV0yRL.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Hongli Zhou, Hui Huang, Ziqing Zhao, Lvyuan Han, Huicheng Wang, Kehai Chen, Muyun Yang, Wei Bao, Jian Dong, Bing Xu, Conghui Zhu, Hailong Cao, and Tiejun Zhao. Lost in benchmarks? rethinking large language model benchmarking with Item Response Theory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(41):35085–35093, March 2026. doi: 10.1609/aaai.v40i41.40814. URL https://ojs.aaai.org/index.php/AAAI/article/view/40814.

Yan Zhuang, Qi Liu, Zachary A. Pardos, Patrick C. Kyllonen, Jiyun Zu, Zhenya Huang, Shijin Wang, and Enhong Chen. Position: AI evaluation should learn from how we test humans. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2306.10512.

APPENDIX CONTENTS

A Detailed Methodology **16**

A.1 Quantification of protocol predictions 16

A.2 Item splitting and capability measurement 17

B Additional Experimental Configuration **18**

B.1 Evaluated Models 18

B.2 Collected Benchmark Data 20

C Protocol Implementation **26**

C.1 Shared Inputs 26

C.2 Pairwise Judging 26

C.3 Single-Transcript Absolute Scoring 28

C.4 Deterministic Transcript Features 29

C.5 Transcript Embeddings 30

C.6 Worked Example: Pairwise and Absolute Judging 30

D Additional Results **34**

D.1 Protocol performance 34

D.2 Prediction performance across historical target periods 34

D.3 Specific choice of judge model does not affect prediction performance 34

D.4 Pairwise judge predictions are best in CODING and CYBER domains 36

D.5 Transfer of predictions across domains 37

D.6 Number of Evaluated Items 37

D.7 Shifts in the Reference Population 39

D.8 Capability-Predicting Signal in Single Transcript Sections 40

D.9 Capability-Predicting Signal Across Transcript Sections For Other Protocols 40

D.10 Capability-Predicting Signal Across Transcript Channels For Other Protocols . . . 42

D.11 Results are robust to the Saturation Boundary 42

D.12 Predicting Across Longer Time Differences 44

D.13 Cost of the Protocols 44

D.14 Judge Ensemble 45

A Detailed Methodology

A.1 Quantification of protocol predictions

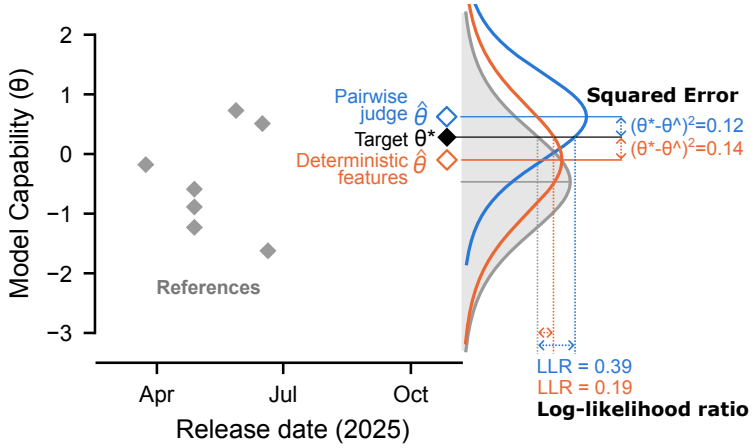


Figure 10: **Quantification of squared error and log-likelihood ratio for an example target model.** Prediction of one target model's capability (*black*: measured θ^*_m) from reference models in the preceding semester (*grey diamonds*), plotted against release date. For each protocol, predictive uncertainty is represented by a Student-t distribution with $n-1$ degrees of freedom, centred on the protocol's prediction and scaled using leave-one-model-out prediction errors on the reference models (*grey*: baseline protocol; *blue*: pairwise judging; *orange*: deterministic features). Pairwise judging and deterministic features miss θ^*_m by similar amounts in opposite directions (squared error 0.12 and 0.14, respectively), but the narrower distribution for pairwise judging assigns greater density to θ^*_m, resulting in a larger log-likelihood ratio relative to the baseline protocol (0.39 versus 0.19).

For each prediction protocol $\mathcal{P}$ and target model m, we first measure prediction accuracy using squared error:

$$\epsilon_{m,\mathcal{P}} = \left(\hat{\theta}_{m,\mathcal{P}} - \theta^*_m\right)^2. \tag{1}$$

We additionally estimate how stable each protocol's predictions are given the finite reference set. For each of the n reference models r (7 or 8 per domain and semester), we refit protocol $\mathcal{P}$ on the remaining $n-1$ references and predict the held-out reference capability θ^*_r. We summarise these leave-one-model-out errors using

$$s_{\mathcal{P}} = \sqrt{\frac{1}{n}\sum_{r\in\mathcal{M}_{\text{ref}}}\left(\hat{\theta}^{(-r)}_{r,\mathcal{P}} - \theta^*_r\right)^2}. \tag{2}$$

Thus, $s_{\mathcal{P}}$ estimates the protocol's out-of-sample prediction error on models drawn from the reference population.

We use this quantity to model predictive uncertainty for each target. Specifically, we define a Student-t distribution with $n-1$ degrees of freedom, centred on the target prediction $\hat{\theta}_{m,\mathcal{P}}$ and scaled by $s_{\mathcal{P}}$:

$$p_{\mathcal{P}}(\theta) = \frac{1}{s_{\mathcal{P}}} t_{n-1}\left(\frac{\theta - \hat{\theta}_{m,\mathcal{P}}}{s_{\mathcal{P}}}\right). \tag{3}$$

We use a Student-t rather than a normal distribution because the prediction scale is estimated from only 7 or 8 reference models per domain and semester. We apply the same construction to the baseline protocol.

For each target, we then compute the log-likelihood ratio (LLR) between the protocol and baseline predictive distributions:

$$\text{LLR}_{m,\mathcal{P}} = \log p_{\mathcal{P}}(\theta^*_m) - \log p_{\text{baseline}}(\theta^*_m). \tag{4}$$

A positive LLR indicates that the protocol assigns greater predictive density to the target's measured capability than the baseline (Figure 10). Unlike squared error, the LLR also accounts for the protocol's estimated prediction uncertainty: two protocols with similar squared error can receive different LLRs if one is substantially more or less stable across the reference models.

A.2 Item splitting and capability measurement

For each domain, we split items and estimate model capabilities as follows:

1. **Responses.** We collect every model's scored response to every item in the domain's benchmarks.
2. **Global IRT fit.** We fit a two-parameter logistic IRT model with a fixed guessing floor to all model responses. Each item has a discrimination parameter a_i, a difficulty parameter b_i, and a fixed guessing floor c_i, set to $1/k$ for a multiple-choice item with k options and to 0 for a free-response item. Each model receives a global capability estimate.
3. **Reference and target sets.** For each target semester, the reference set consists of models from the preceding semester, while the target set consists of models from the target semester.
4. **Saturation boundary.** The three reference models with the highest global capability define the split. An item is *saturated* if all three answer it correctly, and *unsaturated* if at least one answers it incorrectly. We ablate this in Appendix D.11.
5. **Judged items.** From the saturated set, we draw a fixed random sample of up to 25 items from each quartile of item difficulty b_i, for a maximum of 100 items. For section and channel removal (Section 4.2), we used a 40-item subset (10 per quartile). The same candidate items are used for every model and protocol; a model contributes a transcript for an item only when it answers that item correctly.
6. **Measured capability.** For every reference and target model, we estimate θ^* using only the unsaturated items while holding a_i, b_i, and c_i fixed at their global values. This places all semester-specific capability estimates within a domain on a common scale.

Protocols observe only correctly solved transcripts from the sampled saturated items and are evaluated on how well they predict θ^* measured on the unsaturated items.

B Additional Experimental Configuration

This appendix provides additional details on the models and benchmark data used in our retrospective evaluations.

B.1 Evaluated Models

We organise our evaluation framework into six-month periods (semesters). From each semester, we select eight models: these serve as both the target models (for protocols calibrated on the preceding semester) and the reference models for target models in the next semester. For our main analyses, we restricted our model selection to only those that show unencrypted reasoning traces (but see Section 4.3 for additional experiments with closed-source models with hidden reasoning). We include closed-source models in 2023 and 2024, as open-weight reasoning models with visible traces were not yet available in these periods.

Table 4 lists the models included in each semester. Within each period, we select models spanning a range of capability levels rather than only the strongest available models to penalise strategies that always estimate high capabilities. The capability is estimated based on the accuracy on a set of collected benchmarks and transcripts, full collected data in Appendix B.2.

Within each period and domain, the most capable model (derived from IRT across all items in a domain; see Appendix A.2) is shown in bold and the second and third are underlined; closed-source models are shown in italics. For models with explicit reasoning-effort controls, we use the provider default where available and "medium" otherwise. A dash indicates that the model is not evaluated in that domain.

Table 4: **Models used in the retrospective evaluation.** Models are grouped by evaluation period. Entries give the estimated global capability $\theta \pm$ the half-width of its 95% credible interval. Within each period and domain, the highest θ is in bold and the second and third are underlined, with shading of matching strength. Closed-source models are shown in italics. A dash indicates that the model is not evaluated in that domain.

Period	Model	SCIENCE	MATHS	CODING	CYBER
2023	*gpt-4-0613*	**−0.91 ±0.10**	**−1.85 ±0.21**	–	–
	mixtral-8x7b-instruct-v0.1	−1.27 ±0.12	−2.75 ±0.28	–	–
	gpt-3.5-turbo-0125	−1.68 ±0.13	−2.52 ±0.25	–	–
	nous-hermes-2-yi-34b	−1.64 ±0.17	−3.07 ±0.31	–	–
	qwen-72b-chat	−2.03 ±0.18	−2.90 ±0.29	–	–
	llama-2-70b-chat-hf	−1.85 ±0.13	−3.18 ±0.32	–	–
	llama-2-13b-chat-hf	−2.16 ±0.17	−4.23 ±0.63	–	–
	mpt-30b	−2.60 ±0.18	−4.52 ±0.67	–	–
2024-I	*gpt-4o-2024-05-13*	**−0.16 ±0.09**	**−1.40 ±0.20**	–	–
	gpt-4-turbo-2024-04-09	−0.31 ±0.10	−1.41 ±0.20	–	–
	qwen2-72b-instruct	−0.63 ±0.10	−1.53 ±0.21	–	–
	wizardlm-2-8x22b	−0.57 ±0.17	−1.78 ±0.21	–	–
	meta-llama-3-70b-instruct	−0.62 ±0.10	−1.91 ±0.22	–	–
	gemma-2-27b-it	−0.89 ±0.10	−1.96 ±0.23	–	–
	qwen1.5-72b-chat	−1.24 ±0.12	−2.32 ±0.23	–	–
	phi-3-medium-128k-instruct	−1.34 ±0.12	−2.50 ±0.25	–	–
2024-II	deepseek-v3	**0.09 ±0.09**	**−0.73 ±0.18**	–	–
	phi-4	−0.02 ±0.09	−1.16 ±0.18	–	–
	llama-3.3-70b-instruct	−0.32 ±0.09	−1.21 ±0.20	–	–
	qwen2.5-72b-instruct	−0.22 ±0.09	−1.33 ±0.19	–	–
	qwen2.5-32b-instruct	−0.35 ±0.09	−1.22 ±0.21	–	–
	hermes-3-llama-3.1-405b	−0.27 ±0.15	−1.65 ±0.21	–	–
	llama-3.1-8b-instruct	−1.37 ±0.12	−2.00 ±0.24	–	–
	open-mistral-nemo-2407	−1.35 ±0.13	−2.54 ±0.26	–	–
2025-I	minimax-m1	**0.98 ±0.14**	0.46 ±0.14	**0.87 ±0.25**	0.70 ±0.40
	deepseek-r1-0528	0.92 ±0.10	**0.76 ±0.14**	−0.43 ±0.23	**0.90 ±0.33**
	qwen3-235b-a22b	0.64 ±0.09	0.44 ±0.15	−1.15 ±0.29	0.10 ±0.36
	qwen3-32b	0.35 ±0.11	0.11 ±0.12	−1.28 ±0.27	0.04 ±0.28
	deepseek-v3-0324	0.48 ±0.10	−0.54 ±0.15	−1.35 ±0.27	0.03 ±0.36
	qwen3-30b-a3b	0.37 ±0.12	0.08 ±0.14	−1.31 ±0.27	−0.75 ±0.30
	gemma-3-27b-it	−0.34 ±0.15	−1.11 ±0.17	–	–
	mistral-small-3.2-24b-instruct	−0.18 ±0.13	−0.88 ±0.16	−1.72 ±0.29	−1.19 ±0.30
2025-II	kimi-k2-thinking	**1.55 ±0.14**	**1.64 ±0.16**	1.07 ±0.25	1.01 ±0.39
	deepseek-v3.2	1.13 ±0.12	1.28 ±0.16	0.95 ±0.26	**1.30 ±0.36**
	glm-4.7	1.31 ±0.15	1.61 ±0.17	0.77 ±0.24	0.74 ±0.39
	glm-4.5	1.23 ±0.14	1.33 ±0.15	0.53 ±0.27	0.67 ±0.36
	minimax-m2	1.24 ±0.14	0.78 ±0.15	**1.12 ±0.25**	0.48 ±0.38
	qwen3-235b-a22b-thinking-2507	0.98 ±0.13	1.05 ±0.16	−0.80 ±0.26	0.33 ±0.40
	gpt-oss-120b	0.65 ±0.13	0.32 ±0.14	−0.30 ±0.26	0.22 ±0.35
	gpt-oss-20b	0.10 ±0.14	0.16 ±0.14	−0.90 ±0.26	0.41 ±0.35
2026-I	glm-5.1	1.50 ±0.16	1.14 ±0.16	**1.91 ±0.26**	**2.12 ±0.42**
	glm-5.2	1.68 ±0.13	1.43 ±0.16	1.70 ±0.29	1.73 ±0.37
	kimi-k2.6	**1.76 ±0.13**	**1.68 ±0.17**	1.46 ±0.26	1.49 ±0.37
	qwen3.5-397b-a17b	1.48 ±0.15	1.58 ±0.16	1.27 ±0.27	1.19 ±0.40
	kimi-k2.5	1.54 ±0.12	1.54 ±0.17	1.25 ±0.27	1.11 ±0.40
	nvidia-nemotron-3-ultra-550b-a55b	1.29 ±0.11	1.55 ±0.15	0.86 ±0.23	1.47 ±0.35
	qwen3.6-27b	1.22 ±0.14	1.03 ±0.16	1.13 ±0.25	1.60 ±0.40
	gemma-4-31b-it	1.33 ±0.12	1.13 ±0.15	1.03 ±0.26	1.43 ±0.37

B.2 COLLECTED BENCHMARK DATA

Our main evaluation retains benchmark items across four domains.

Science. We use all 198 questions in GPQA-Diamond. For ARC-Challenge, we select a fixed random subset of 200 questions from the 1,172-question `test` split, using seed 0. For HLE, only the 575 text-only questions in the HLE-Verified gold subset are used for analysis. The science domain retains 210 questions tagged Physics, Chemistry, Biology/Medicine, Computer Science/AI, or Engineering.

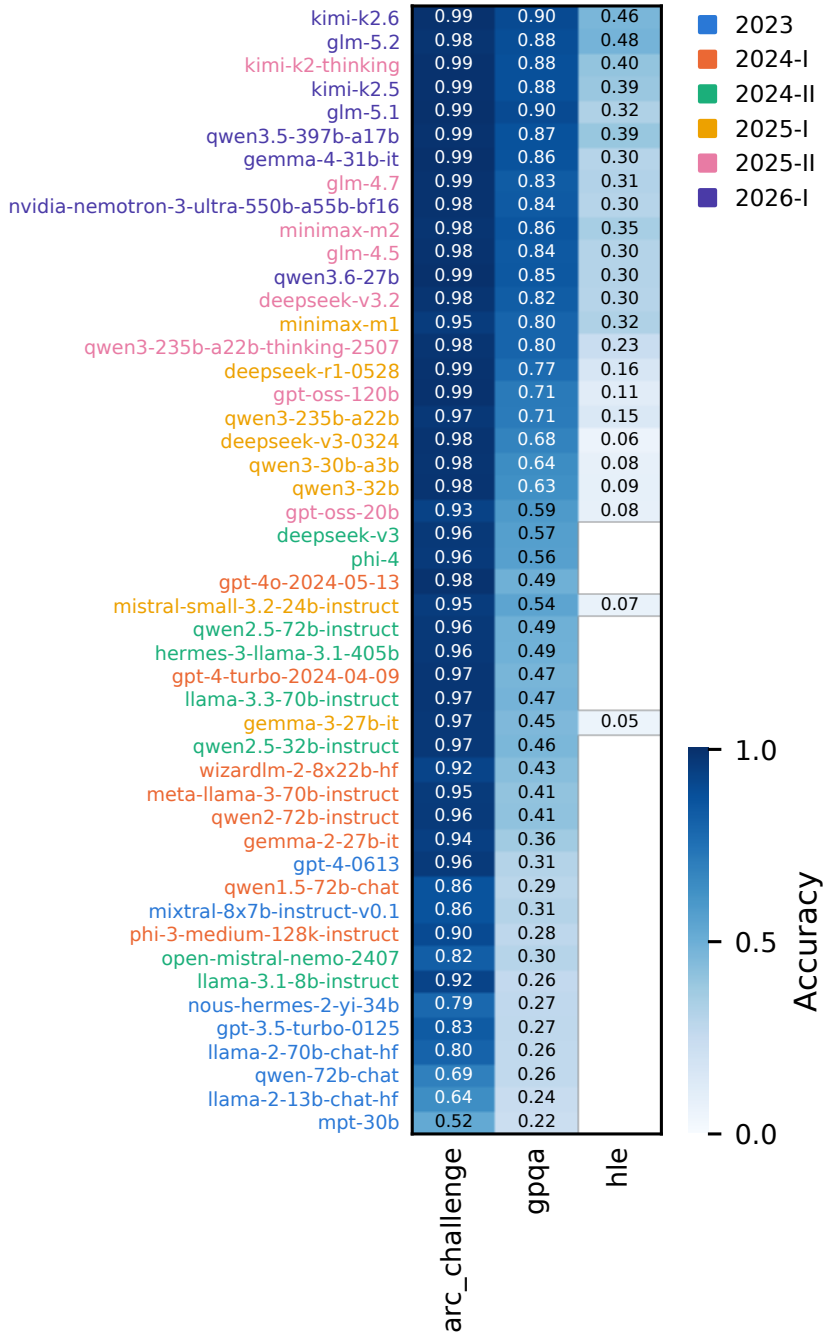
Mathematics. We select 150 questions from the `math` category of the MMLU-Pro `test` split, using a fixed random sample with seed 42. AIME 2024, AIME 2025, and AIME 2026 each contribute all 30 problems in their respective datasets. We also use all 12 problems from APEX 2025 and all 47 problems from APEX Shortlist, each provided in a `train` split. HLE-Gold contributes the 263 text-only gold questions tagged `Math`.

Coding. For AgentBench OS, we use the revised `test` split from Inspect's AgentBench OS task version 4-A. It contains 119 tasks after six ill-posed tasks were removed from the original 125-task split. For SWE-Bench, we use all 50 tasks in the `test` split of SWE-bench Verified Mini, distributed as `MariusHobbhahn/swe-bench-verified-mini`. For Terminal-Bench, we use all 89 tasks from version 2.1, pinned to revision 6.

Cybersecurity. We use the hard variant of CyBench with 39 challenges; the Inspect implementation omits the `motp` challenge for licensing reasons. For InterCode, we use Inspect's GDM InterCode-CTF implementation with 78 challenges. This subset excludes 22 internet-dependent challenges from the original 100-task CTF collection.

Figures 11–14 summarise the collected benchmark data. Accuracy generally increases across evaluation periods, causing the easier benchmarks to approach saturation for later models (Figure 14). The global IRT fit places benchmarks within each domain across distinct ranges of item difficulty (Figure 12), while estimated global capability increases with model release date across all four domains (Figure 13).

Figure 11: **Benchmark accuracy by model and benchmark.** Models are coloured by evaluation period. SCIENCE is shown here; the remaining domains continue on the following pages.



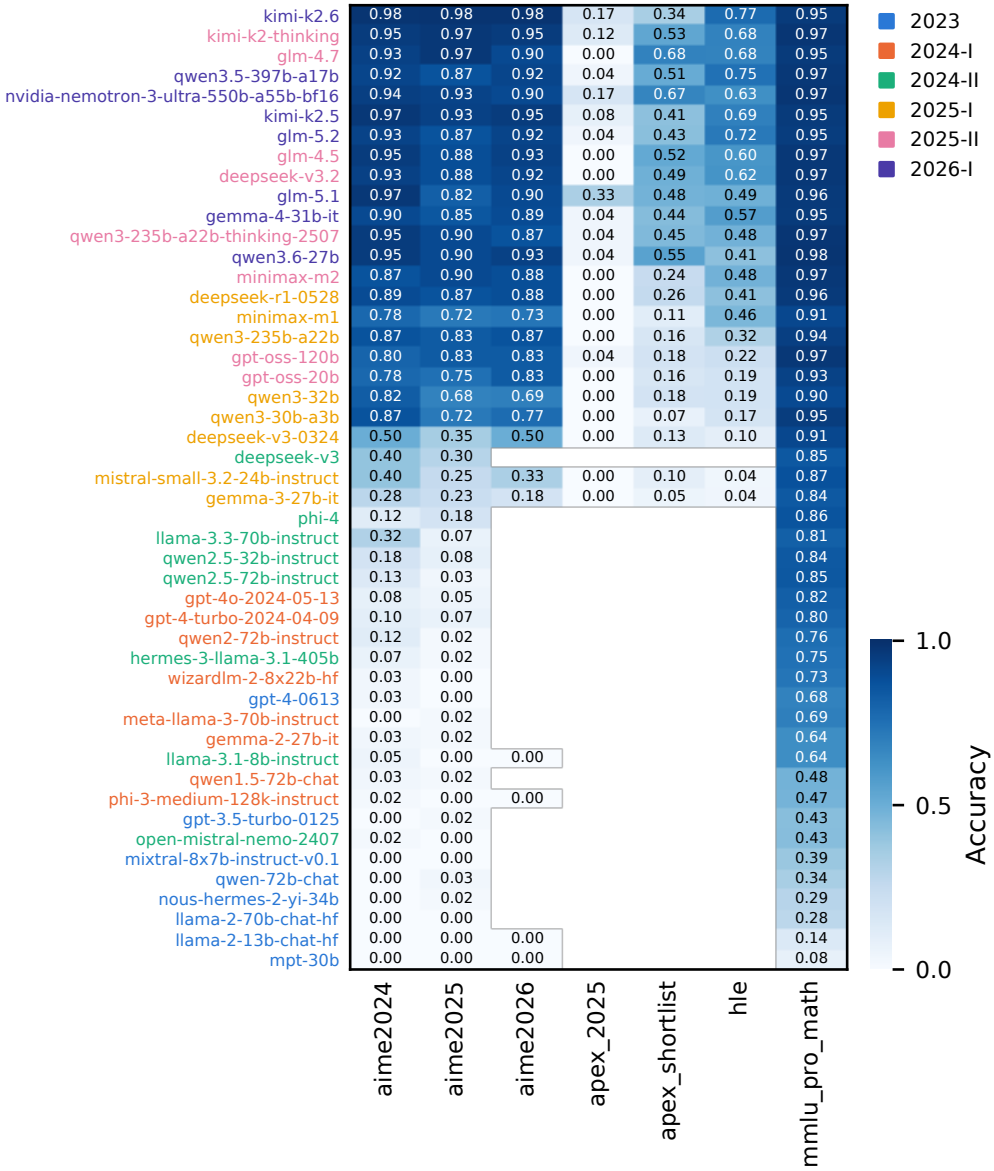


Figure 11: (Continued.) MATHS.

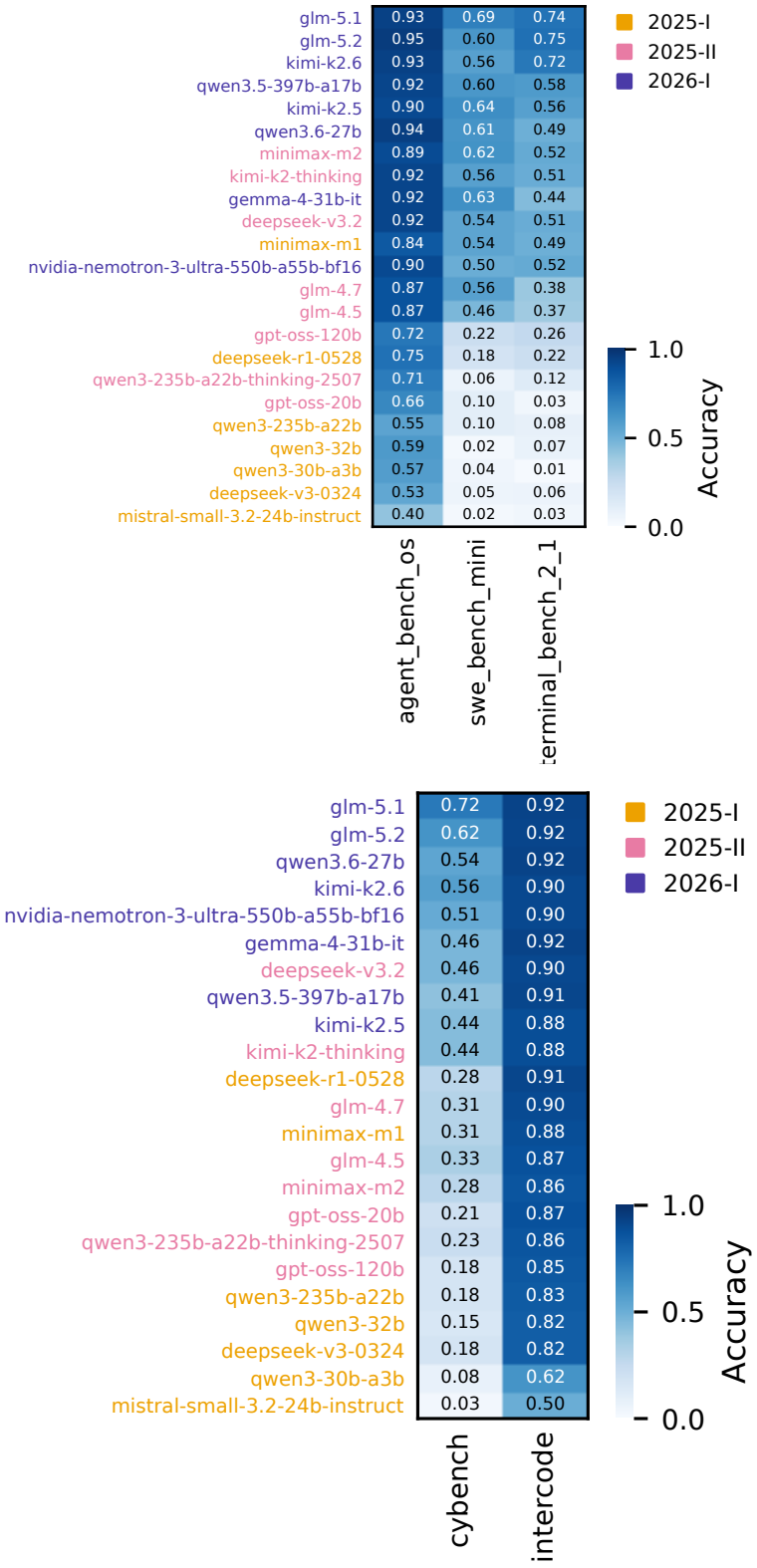


Figure 11: (Continued.) CODING (top) and CYBER (bottom).

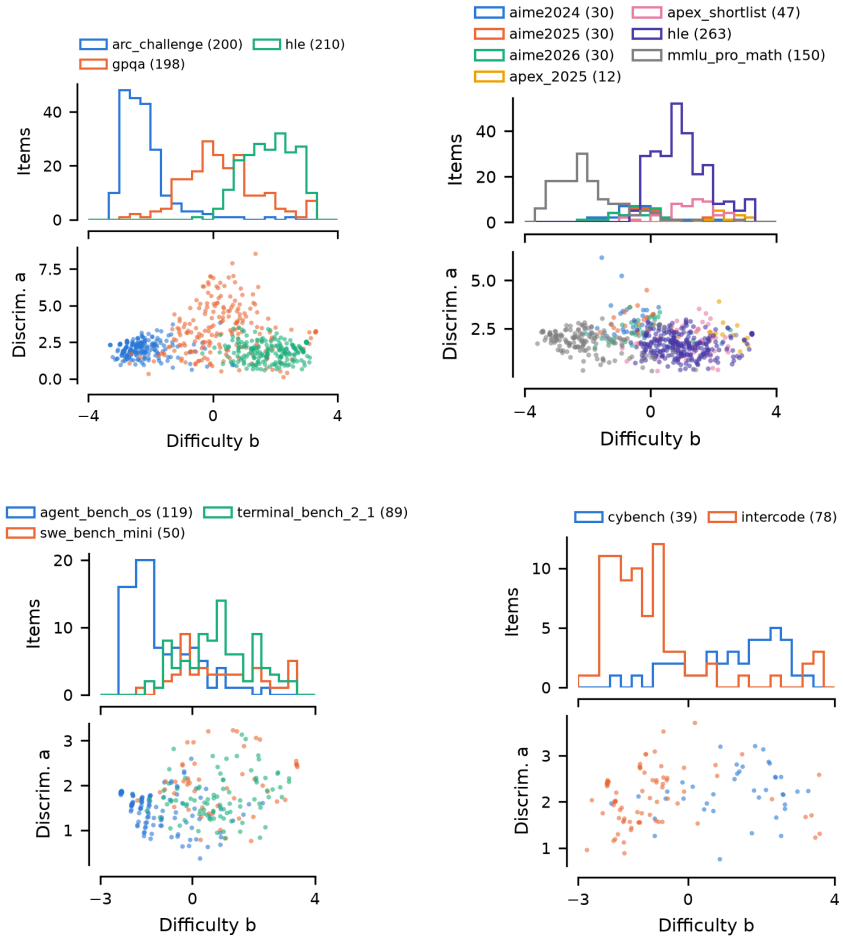


Figure 12: **Item difficulty parameters from the global IRT fit.** SCIENCE and MATHS are shown in the top row; CODING and CYBER in the bottom row. For each domain, the upper panel shows the distribution of item difficulty b_i by benchmark, with the number of items shown in brackets. The lower panels show discrimination a_i against difficulty b_i.

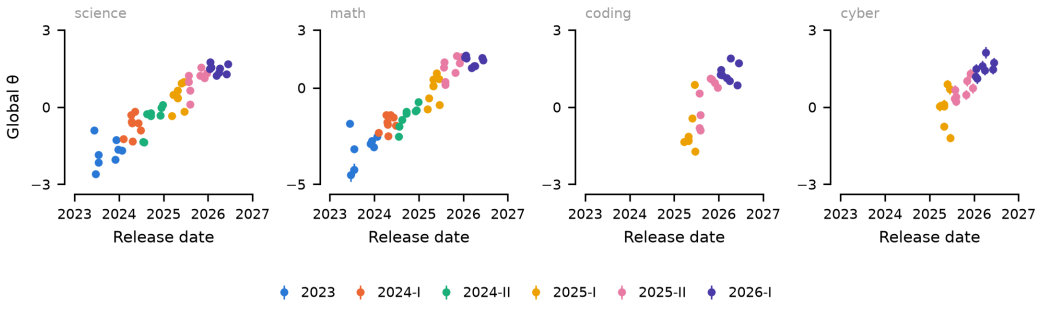


Figure 13: **Model capabilities by release date.** Estimated model capabilities in each domain against release date, based on the global IRT fit (see Appendix A.2), coloured by evaluation period. Error bars show posterior standard deviation.

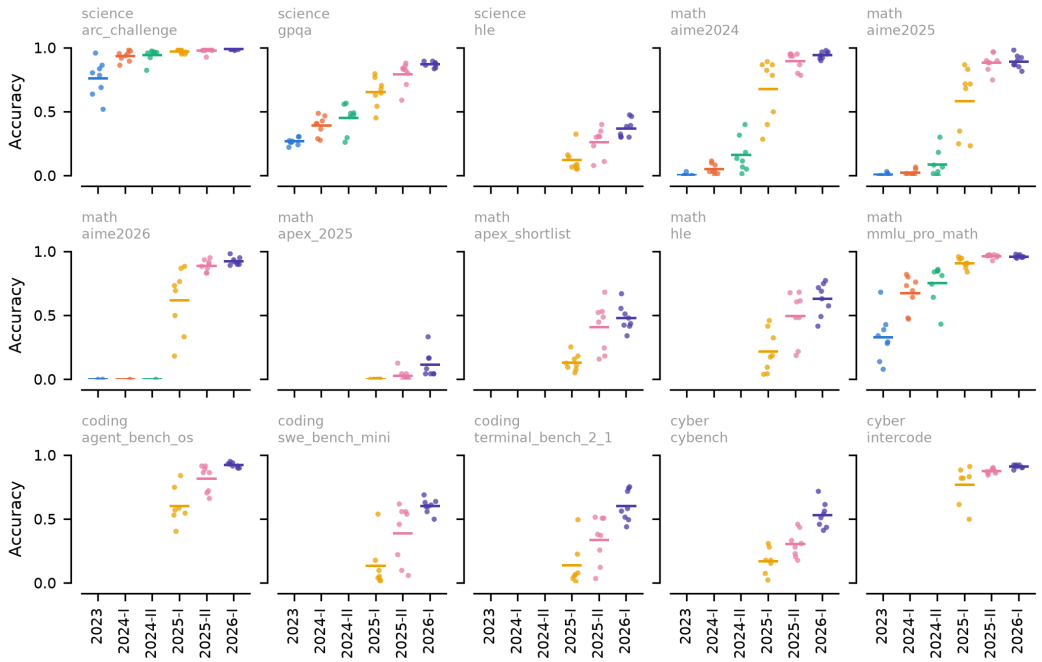


Figure 14: **Benchmark accuracy by evaluation period.** Points show individual models and lines show period means.

C PROTOCOL IMPLEMENTATION

This section describes the implementation of each prediction protocol. All protocols use the shared inputs and calibration data described below.

C.1 SHARED INPUTS

Items. For each prediction task, we sample up to 25 saturated items from each quartile of global item difficulty b_i using a fixed random seed, for a maximum of 100 items. Sampling is performed once at the item level so that all models and protocols use the same candidate items. For the position and channel analyses (Section 4.2; Appendix D.8, D.9, D.10) and the reference-period sweep (Appendix D.12), we use a fixed subset of 10 items per difficulty quartile. Protocols never observe unsaturated items or whether a model failed an item.

Transcripts. For each model–item pair, we use the first attempt and retain the transcript only if the model answers the item correctly. Transcripts contain the message history excluding the system message and initial user prompt; subsequent user messages in agentic tasks are retained. Assistant messages, reasoning traces, tool calls, and tool outputs are marked by role. Unless otherwise stated, visible reasoning is included. For multiple-choice tasks, each transcript is paired with the answer-option ordering shown to that model.

Calibration data. Each reference model $r \in \mathcal{M}_{\text{ref}}$ is associated with its measured capability θ_r^* on the unsaturated items. Every protocol is calibrated using only the reference models for that prediction task, ensuring that all protocols and the baseline use the same labelled models. Target capabilities θ_m^* are used only for evaluation.

Baseline. The baseline assigns every target model the mean measured capability of the reference models:

$$\hat{\theta}_m = \bar{\theta}_{\text{ref}} = \frac{1}{|\mathcal{M}_{\text{ref}}|} \sum_{r \in \mathcal{M}_{\text{ref}}} \theta_r^*.$$

Because protocols observe only correctly solved saturated items, benchmark accuracy provides no information for distinguishing target models in this setting.

C.2 PAIRWISE JUDGING

Comparisons. For each benchmark, we compare every reference–target pair and every pair of reference models. Reference–reference comparisons calibrate the relationship between judge preferences and capability differences, while reference–target comparisons locate each target relative to the calibrated references. Target–target comparisons are not used. A pair is evaluated on an item only when both models answer it correctly and produce a non-trivial response. Each comparison is presented in both transcript orders to reduce sensitivity to position bias.

Judge prompt. The judge is an LLM sampled at temperature 0. It receives the task followed by two correct transcripts and is asked which one was produced by the more capable model. For multiple-choice tasks, the answer-option ordering shown to each model is also provided. The prompts are reproduced verbatim; angle brackets mark filled-in fields, and the options blocks appear only for multiple-choice tasks.

Pairwise judge: system prompt

```
You are an expert LLM evaluator, tasked with judging the responses of two AI models to the
      same task.
You will be given the question and the responses from two models, labeled Transcript 1 and
      Transcript 2, both coming to the correct solution.

One of the two models is more capable than the other, and your task is to identify which
     is the more capable model (1 or 2). You MUST follow this reply format:

EXPLANATION: <Your explanation of which response was produced by the more capable model
     and why. 2-4 sentences.>
ANSWER: <1 or 2, indicating which response was produced by the more capable model.>
```

Pairwise judge: user prompt

```
Input question:
---

<question>

---

Multiple choice options as shown to Model 1:
A) ...

Transcript 1:
---

<transcript 1>

---

Multiple choice options as shown to Model 2:
A) ...

Transcript 2:
---

<transcript 2>

---

Which of the two transcripts is from the more capable model? Answer in the instructed
    format.
```

Unparsable responses are treated as missing. The main condition uses complete transcripts, including visible reasoning. Comparisons that exceed the judge's context window are omitted rather than truncated.

Aggregating comparisons. To aggregate the pairwise win/lose comparisons into a single prediction of the capability score of the target model we use a weighted average with inverse precision scaling, adapted from Borenstein et al. (2009); Bradley & Terry (1952).

Let $w_b^{(m,o)}$ denote the number of times model m is preferred over opponent o on benchmark b, out of $n_b^{(m,o)}$ judgements. Then the empirical log odds of model m winning against model o is:

$$\text{logit}\left(P(m \succ o)\right) \approx \ell_b^{(m,o)} = \log \frac{w_b^{(m,o)} + 1}{n_b^{(m,o)} - w_b^{(m,o)} + 1}.$$

The +1 terms smooth the estimate and prevent infinite log odds.

For each benchmark b, we calibrate the relationship between judge preference and capability difference using reference–reference comparisons:

$$\ell_b^{(r,r')} = k_b \cdot (\theta_r^* - \theta_{r'}^*)$$

This represents the relationship between the ground-truth difference of capability and how often a judge actually picks the more capable model. In this relation k_b is the *sensitivity* of the judge to differences in capability. Higher sensitivity means that the judge will pick the better model more frequently.

We fit k_b as the slope of an ordinary least squares fit of $\ell_b^{(r,r')}$ on $\theta_r^* - \theta_{r'}^*$. The fit includes an intercept which we discard: since $\ell_b^{(r,r')} = -\ell_b^{(r',r)}$, the true intercept is zero.

Benchmarks with insufficient reference comparisons or a non-positive fitted sensitivity k_b are excluded. Given a single benchmark b and reference r, the estimate of the target capability is:

$$\hat{\theta}_{m|b,r} = \theta_r^* + \frac{\ell_b^{(m,r)}}{k_b},$$

and the variance of this estimate across reference models is:

$$\text{Var}_r\left(\hat{\theta}_{m|b,r}\right) = \frac{\tau_b^2}{k_b^2} \quad \Longrightarrow \quad \text{precision } \omega_b = \frac{k_b^2}{\tau_b^2}.$$

We estimate the residual variance τ_b^2 from the reference–reference calibration and combine the target–reference comparisons using weighted average with a Gaussian prior centred on the mean and variance of all reference models $\mathcal{N}(\bar{\theta}_{\text{ref}}, s^2_{\text{ref}})$. The average is weighted on the precision parameter ω_b we just estimated.

$$\hat{\theta}_m = \frac{\sum_{(b,r)} \omega_b \cdot \hat{\theta}_{m|b,r} + \bar{\theta}_{\text{ref}}/s^2_{\text{ref}}}{\sum_{(b,r)} \omega_b + 1/s^2_{\text{ref}}},$$

where s^2_{ref} is the sample variance of reference capabilities. A target with no usable comparisons receives the prior mean, which is equal to the baseline prediction.

Leave-one-model-out calibration. For each prediction task, the three highest-capability reference models are considered as candidate judges. To evaluate each candidate judge, we remove one reference model at a time, recalibrate the protocol on the remaining references, and predict the capability of the held-out model. We select the judge with the lowest mean squared leave-one-model-out error across the reference models. Ablations on this are available in Appendix D.3.

Sensitivity and Ablations. We conduct extensive sensitivity analyses to examine how protocol performance depends on judge aggregation, evaluation budget, and reference calibration. First, combining judgements across a panel of models gives little benefit over a single well-selected judge (Appendix D.14). Second, scaling the number of judged items provides diminishing returns, as performance is often bounded by systematic judge biases rather than finite-sample noise (Appendix D.6). Third, this protocol is notably more expensive than others, and we analyse the trade off between cost and prediction accuracy in Appendix D.13. Finally, calibration quality is highly dependent on the composition of the reference set: anchoring predictions requires high-capability reference models, particularly when evaluating frontier targets (Appendix D.7).

C.3 Single-Transcript Absolute Scoring

Ratings. The judge receives one correct transcript at a time and assigns a capability score from 1 to 10. Candidate judge models, temperature, transcript representation, and reasoning-access condition match those used for pairwise judging. The prompts are reproduced verbatim; the options block appears only for multiple-choice tasks.

Absolute judge: system prompt

```
You are an expert LLM evaluator, tasked with judging the capability of an AI model from
     its response to a task.
You will be given the question and one response, labelled Transcript 1, which comes to the
      correct solution.

For multiple choice questions, interpret any answer letter using the options as shown to
     this model, listed immediately before the transcript.

Estimate the model's capabilities.

Rate capability on a scale from 1 to 10, where 1 indicates very low capability (worse than
      the average human), 3 is the average person, 6 is on par with the top 1% of all
     humans, and 7-10 indicates very high capabilities beyond human experts.

You MUST follow this reply format:

EXPLANATION: <Explain the capability demonstrated by the model and why, in 2-4 sentences.>
ANSWER: <One number from 1 to 10, indicating the model's capability.>
```

Absolute judge: user prompt

```
Input question:
---

<question>

---

Multiple choice options as shown to Model 1:
A) ...

Transcript 1:
---

<transcript>

---

How capable is the model that produced this transcript, on a scale from 1 to 10? Answer in
     the instructed format.
```

Unparsable responses and ratings outside the 1–10 range are treated as missing.

Calibration. Let $\bar{s}_m$ denote the mean transcript rating for model m. We fit

$$\theta_r^* = \gamma_0 + \gamma_1 \bar{s}_r$$

using the reference models and predict target capability as

$$\hat{\theta}_m = \gamma_0 + \gamma_1 \bar{s}_m.$$

The judge is selected using leave-one-model-out prediction error on the reference models, following the same procedure as pairwise judging.

C.4 Deterministic Transcript Features

Each transcript is represented using deterministic features derived from model-written text and tool-use metadata. These include transcript length, visible-reasoning length, the presence of visible reasoning, tool-call count, and lexical rates capturing pronouns, hedging, backtracking, uncertainty, confidence, questions, verification language, causal and inferential connectives, and contrast markers. Lexical rates are computed per 1,000 words after removing code, mathematical spans, and final-answer lines.

Table 5: **Deterministic transcript features.**

Feature	Definition
Length	Log model-written transcript length
Reasoning length	Log visible-reasoning length
Visible reasoning	Indicator for non-empty visible reasoning
Personal pronouns	Rate of first-person pronouns
Hedging	Rate of hedging expressions
Backtracking	Rate of self-correction expressions
Uncertainty	Rate of uncertainty expressions
Confidence	Rate of confidence expressions
Questions	Question-mark rate
Verification	Rate of checking and verification expressions
Connectives	Rate of causal and inferential connectives
Contrast	Rate of contrastive expressions
Tool calls	Log number of tool calls

Regression. For each model, we average transcript features across its correctly solved sampled items to obtain a model-level representation. Features are standardised using the reference models, and ridge regression is fitted to predict θ^*. The regularisation coefficient is selected by leave-one-model-out cross-validation over 13 logarithmically spaced values from 10^{-3} to 10^3.

C.5 TRANSCRIPT EMBEDDINGS

We embed each rendered transcript, including visible reasoning when available, using Qwen3-Embedding-8B at a fixed model revision. Transcripts that exceed the encoder context window are divided into overlapping chunks of at most 8,192 tokens. Chunk embeddings are combined using a token-weighted mean and normalised to produce one representation per transcript.

For each model, we average its transcript embeddings across correctly solved sampled items to obtain a model-level representation. We then fit ridge regression from this representation to θ^* using the reference models. The regularisation coefficient is selected by leave-one-model-out cross-validation over 13 logarithmically spaced values from 10^{-2} to 10^4. The main results use ridge regression on full-transcript embeddings.

C.6 WORKED EXAMPLE: PAIRWISE AND ABSOLUTE JUDGING

We show one GPQA-Diamond item from the SCIENCE 2026-I prediction task, answered correctly by the reference model gpt-oss-20b ($\theta^* = 0.01$) and the target model glm-5.2 ($\theta^* = 1.71$). Both protocols use the same judge, kimi-k2-thinking, with visible reasoning included, and the prompts of Appendix C.2 and C.3. Transcripts and judge outputs are reproduced verbatim except that Unicode symbols are typeset as mathematics; each judge box shows the judge's reasoning (grey) above the explanation and verdict it returned.

GPQA-Diamond item

Consider a system with Hamiltonian operator $H = \varepsilon\vec{\sigma}.\vec{n}$. Here, $\vec{n}$ is an arbitrary unit vector, ε is a constant of dimension energy, and components of $\vec{\sigma}$ are the Pauli spin matrices. What are the eigenvalues of the Hamiltonian operator?

Options shown to both models: A) $+\varepsilon, -\varepsilon$ B) $+\hbar/2, -\hbar/2$ C) $+1, -1$ D) $+\varepsilon\hbar/2, -\varepsilon\hbar/2$

gpt-oss-20b (reference, $\theta^* = 0.01$, 722 characters)

REASONING

We need eigenvalues of $H = \varepsilon(\sigma \cdot n)$. For any unit vector n, $\sigma \cdot n$ has eigenvalues ± 1. Because $\sigma \cdot n = (n \cdot \sigma)$ has eigenvalues ± 1. Then H has eigenvalues $\pm\varepsilon$. So answer is A.

But double-check: The $\sigma \cdot n$ is a 2x2 hermitian matrix with eigenvalues ± 1. Classic. So H eigenvalues are $\pm\varepsilon$.

Given choices:

A) $+\varepsilon, -\varepsilon$. B $+\hbar/2, -\hbar/2$. C $+1, -1$. D $+\varepsilon\hbar/2, -\varepsilon\hbar/2$. So A.

Thus answer: A.

Now produce final.

ANSWER

The operator $\vec{\sigma} \cdot \vec{n}$ is a 2×2 Hermitian matrix whose eigenvalues are ± 1 (since $(\vec{\sigma} \cdot \vec{n})^2 = \mathbb{I}$). Multiplying by the scalar ε simply scales these eigenvalues, giving eigenvalues $+\varepsilon$ and $-\varepsilon$.

ANSWER: A

glm-5.2 (target, $\theta^* = 1.71$, 2,141 characters)

REASONING

The Hamiltonian is $H = \varepsilon\vec{\sigma} \cdot \vec{n}$, where $\vec{n}$ is a unit vector.
The Pauli spin matrices satisfy the property that for any unit vector $\vec{n}$, the eigenvalues of $\vec{\sigma} \cdot \vec{n}$ are +1 and -1.
Proof: $(\vec{\sigma} \cdot \vec{n})^2 = \sigma_i n_i \sigma_j n_j = \frac{1}{2}(\sigma_i\sigma_j + \sigma_j\sigma_i)n_i n_j = \delta_{ij} n_i n_j = n_i n_i = |\vec{n}|^2 = 1$.
Since the trace of $\vec{\sigma} \cdot \vec{n}$ is 0, the sum of eigenvalues is 0.
Since the square has eigenvalues 1, the eigenvalues must be 1 and -1.
Therefore, the eigenvalues of $H = \varepsilon\vec{\sigma} \cdot \vec{n}$ are $\varepsilon \times 1$ and $\varepsilon \times (-1)$, which are $+\varepsilon$ and $-\varepsilon$.

Let's check the options:
A) $+\varepsilon, -\varepsilon$

B) $+\hbar/2, -\hbar/2$
C) $+1, -1$
D) $+\varepsilon\hbar/2, -\varepsilon\hbar/2$
Option A matches perfectly.

ANSWER

The Hamiltonian is given by $H = \varepsilon\vec{\sigma}\cdot\vec{n}$, where $\vec{n}$ is a unit vector and $\vec{\sigma}$ represents the Pauli spin matrices.

To find the eigenvalues of H, we first determine the eigenvalues of the operator $\vec{\sigma}\cdot\vec{n}$.
A well-known property of the Pauli matrices is that for any arbitrary unit vector $\vec{n}$, the operator $\vec{\sigma}\cdot\vec{n}$ has eigenvalues $+1$ and -1. This can be seen because:
1. The trace of $\vec{\sigma}\cdot\vec{n}$ is zero, meaning the sum of its eigenvalues is 0.
2. $(\vec{\sigma}\cdot\vec{n})^2 = \sum_{i,j}\sigma_i\sigma_j n_i n_j = \sum_{i,j}(\delta_{ij}+i\epsilon_{ijk}\sigma_k)n_i n_j = \sum_i n_i^2 + i\sum_{i,j,k}\epsilon_{ijk}n_i n_j\sigma_k = |\vec{n}|^2 + 0 = 1$.
Since the sum of the eigenvalues is 0 and the sum of their squares is 1, the eigenvalues must be $+1$ and -1.

Therefore, the eigenvalues of the Hamiltonian $H = \varepsilon\vec{\sigma}\cdot\vec{n}$ are obtained by multiplying the eigenvalues of $\vec{\sigma}\cdot\vec{n}$ by ε:
$\lambda_1 = \varepsilon(+1) = +\varepsilon$
$\lambda_2 = \varepsilon(-1) = -\varepsilon$
ANSWER: A

Pairwise judge. The pair is judged in both orders (Appendix C.2); the judge prefers glm-5.2 in both.

Order 1: gpt-oss-20b = Transcript 1, glm-5.2 = Transcript 2 **Verdict: glm-5.2**

JUDGE REASONING

The question: $H = \varepsilon\vec{\sigma}\cdot\vec{n}$, find eigenvalues. Both models correctly identify that $\vec{\sigma}\cdot\vec{n}$ has eigenvalues ±1, so H has eigenvalues $\pm\varepsilon$. Both select option A. Now comparing the quality of reasoning:

Transcript 1: Brief thought process; recognizes that $\sigma\cdot n$ has eigenvalues ±1 because it's a standard result. Mentions that $(\vec{\sigma}\cdot\vec{n})^2 = \mathbb{I}$ (implied by "eigenvalues are ±1"). Quick verification of options. Final answer is concise and correct.

Transcript 2: More detailed derivation. Shows the proof that $(\vec{\sigma}\cdot\vec{n})^2 = 1$. Uses the trace argument (sum of eigenvalues is 0). Explicitly calculates $(\vec{\sigma}\cdot\vec{n})^2$ using Pauli matrix algebra: $\sigma_i\sigma_j = \delta_{ij} + i\epsilon_{ijk}\sigma_k$. Then concludes eigenvalues are ±1. Then scales by ε.

Transcript 1 is brief but correct; it relies on "classic" knowledge. Transcript 2 provides a rigorous derivation from first principles (Pauli matrix commutation relations) and explicitly mentions the trace condition. In an evaluation of capability, demonstrating the ability to derive from first principles is generally considered a higher level of capability than just recalling a fact, assuming both are correct. Transcript 2 also explicitly mentions the trace condition and the squaring property, which are the key ingredients.

Therefore, Transcript 2 is from the more capable model.

EXPLANATION

Transcript 2 provides a rigorous, step-by-step derivation of the eigenvalues from the algebraic properties of the Pauli matrices, explicitly proving that $(\vec{\sigma}\cdot\vec{n})^2 = \mathbb{I}$ and using the trace condition to confirm the eigenvalues are ±1 before scaling by ε. In contrast, Transcript 1 relies on recalling the result as a "classic" property without derivation. While both arrive at the correct answer, the ability to derive the result from first principles demonstrates deeper physical and mathematical understanding.

ANSWER: 2

Order 2: glm-5.2 = Transcript 1, gpt-oss-20b = Transcript 2 **Verdict: glm-5.2**

JUDGE REASONING

Comparing two transcripts from two models answering a physics question about Hamiltonian eigenvalues. Both models correctly identify that $\vec{\sigma}\cdot\vec{n}$ has eigenvalues ±1, making the Hamiltonian's eigenvalues $\pm\varepsilon$, and both select option A.

Transcript 1: Demonstrates thorough command of the formalism. Explicitly proves $(\vec{\sigma}\cdot\vec{n})^2 = \mathbb{I}$ using index notation. Invokes the traceless condition to rigorously fix the spectrum as $\{+1, -1\}$ rather than

merely allowing eigenvalues ± 1 individually. Uses the calculation: $(\vec{\sigma} \cdot \vec{n})^2 = \sigma_i n_i \sigma_j n_j = \frac{1}{2}(\sigma_i \sigma_j + \sigma_j \sigma_i) n_i n_j = \delta_{ij} n_i n_j = n_i n_i = |\vec{n}|^2 = 1$. Since the trace of $\vec{\sigma} \cdot \vec{n}$ is 0, the sum of eigenvalues is 0. Since the square has eigenvalues 1, the eigenvalues must be 1 and -1.

Transcript 2: Reaches the correct answer with less detail. Omits the trace argument needed to fully exclude degenerate spectra. More concise but less rigorous.

Transcript 1's explanation is more complete and educational. It shows the mathematical steps clearly using index notation and explicitly mentions both the trace condition and the squaring property. The logic is sound and the derivation is complete.

Therefore, Transcript 1 is from the more capable model.

EXPLANATION

Transcript 1 demonstrates a more thorough and rigorous command of the formalism. It explicitly proves $(\vec{\sigma} \cdot \vec{n})^2 = \mathbb{I}$ using index notation and, crucially, invokes the traceless condition to rigorously fix the spectrum as $\{+1, -1\}$ rather than merely allowing eigenvalues ± 1 individually. Transcript 2 reaches the correct answer with less detail and omits the trace argument needed to fully exclude degenerate spectra.

ANSWER: 1

Absolute judge. Rated one at a time (Appendix C.3), the two transcripts receive the same score, so this item carries no signal for absolute scoring while it separates the two models under pairwise judging.

gpt-oss-20b **Verdict: 6**

JUDGE REASONING

Evaluating the capability of an AI model based on its response to a physics question about quantum mechanics. The question asks for the eigenvalues of the Hamiltonian $H = \varepsilon \vec{\sigma} \cdot \vec{n}$, where $\vec{n}$ is a unit vector and $\vec{\sigma}$ are the Pauli matrices.

The model correctly identifies that:

1. $\vec{\sigma} \cdot \vec{n}$ is a spin operator along direction $\vec{n}$
2. The eigenvalues of $\vec{\sigma} \cdot \vec{n}$ are ± 1 (since $(\vec{\sigma} \cdot \vec{n})^2 = I$)
3. Therefore, the eigenvalues of $H = \varepsilon(\vec{\sigma} \cdot \vec{n})$ are $\pm\varepsilon$
4. This corresponds to option A

The reasoning is correct and concise. The model shows understanding of:

- Pauli matrices properties
- Spin operators in quantum mechanics
- Eigenvalue scaling under scalar multiplication
- The relationship between the dot product of Pauli matrices with a unit vector and the resulting operator's eigenvalues

This is a standard quantum mechanics problem, typically taught in undergraduate or early graduate quantum mechanics courses. Getting this right demonstrates solid understanding of quantum mechanics concepts.

The model's explanation is clear, concise, and correct. It shows understanding of why the eigenvalues are what they are, not just memorization. The operator $\vec{\sigma} \cdot \vec{n}$ is correctly identified as a 2×2 Hermitian matrix whose eigenvalues are ±1 (since $(\vec{\sigma} \cdot \vec{n})^2 = \mathbb{I}$), and the scaling by ε is correct.

Rating: 6 (on par with top 1% of humans — those who have studied quantum mechanics).

EXPLANATION

The model correctly identifies that the operator $\vec{\sigma} \cdot \vec{n}$ (representing spin along direction $\vec{n}$) has eigenvalues ± 1 since $(\vec{\sigma} \cdot \vec{n})^2 = \mathbb{I}$ for any unit vector $\vec{n}$, and correctly applies scalar multiplication to obtain eigenvalues $\pm\varepsilon$. This demonstrates solid understanding of quantum spin physics and linear algebra concepts typically taught in undergraduate quantum mechanics courses.

ANSWER: 6

glm-5.2 **Verdict: 6**

JUDGE REASONING

Evaluating the capability of the AI model that produced this transcript in response to the physics question about Hamiltonian eigenvalues.

First, analyze the response: The model correctly identifies that we need to find the eigenvalues of $\vec{\sigma} \cdot \vec{n}$ first. It correctly states that the eigenvalues of $\vec{\sigma} \cdot \vec{n}$ are +1 and -1. It provides a rigorous proof:

- Uses the property that $(\vec{\sigma} \cdot \vec{n})^2 = 1$ (since $\vec{n}$ is a unit vector and Pauli matrices satisfy $\sigma_i \sigma_j = \delta_{ij} + i\epsilon_{ijk}\sigma_k$).
- Correctly notes that the trace is 0, so eigenvalues sum to 0.
- Correctly concludes that since $(\vec{\sigma} \cdot \vec{n})^2 = I$, the eigenvalues squared are 1, so they are ± 1.
- Then correctly scales by ε to get eigenvalues $\pm\varepsilon$.
- Correctly identifies option A as the answer.

The reasoning is sound, mathematically correct, and physically accurate. This is a standard result in quantum mechanics (spin-1/2 systems), related to the fact that $\vec{\sigma} \cdot \vec{n}$ represents spin projection along direction $\vec{n}$, which has eigenvalues ± 1 (for Pauli matrices specifically).

The model not only solves it but provides clear reasoning, checks the trace condition, verifies the square of the operator, and correctly maps to the multiple choice options. The solution is complete and rigorous.

The model provides a perfect solution with clear mathematical derivation. This suggests capability around 6. It's not "beyond human experts" because this is a basic problem that any physics expert would know, but it's certainly better than the average human. An average person wouldn't know this, and an undergraduate physics student should. A graduate student in physics definitely knows this.

Rating: 6 (on par with top 1% of humans — those who have studied quantum mechanics).

EXPLANATION

The model correctly solves the quantum mechanics problem by recognizing that the operator $\vec{\sigma} \cdot \vec{n}$ represents spin projection along the unit vector $\vec{n}$ and has eigenvalues ± 1. It rigorously derives this using the Pauli matrix identity $(\vec{\sigma} \cdot \vec{n})^2 = |\vec{n}|^2 = 1$ and the traceless property, then correctly scales by ε to obtain eigenvalues $\pm\varepsilon$, matching option A. This demonstrates accurate physical reasoning and mathematical competence at the level of a trained physicist.

ANSWER: 6

None of the four verdicts notices a slip in glm-5.2's answer: it states that "the sum of their squares is 1", whereas the squared eigenvalues sum to $\mathrm{tr}(\vec{\sigma} \cdot \vec{n})^2 = \mathrm{tr}\,\mathbb{I} = 2$. The pairwise preference rests on the presence of an explicit derivation rather than on its correctness, and the absolute judge gives both transcripts the same score.

D Additional Results

D.1 Protocol performance

Table 6: **Prediction error per protocol.** *(Accompaniment to Figure 3) MSE*: Mean squared error over the 16 target models of late 2025 and early 2026 (averaged over four domains). ΔSE: mean squared error minus the baseline's, paired t-test (n=16 target models). *LLR*: mean log likelihood ratio of the protocol's prediction against the baseline protocol (standard error in parentheses). p: one-sample t-test on LLR against 0. We report statistics based on LLR in Section 4.1, to penalise confidently incorrect predictions.

Protocol	**MSE [95% CI]**	**ΔSE**	p	**LLR**	p
Baseline protocol	1.37 [0.88, 1.87]	–	–	–	–
Deterministic features	0.86 [0.52, 1.21]	−0.51	.064	−0.26 (0.30)	.404
Embedding	1.01 [0.61, 1.41]	−0.36	<.001	0.04 (0.06)	.464
Absolute scoring	1.01 [0.67, 1.35]	−0.36	.002	−0.21 (0.16)	.200
Pairwise judge	0.26 [0.18, 0.34]	−1.12	<.001	0.68 (0.21)	.005

D.2 Prediction performance across historical target periods

Table 7: **Prediction error per protocol across target periods.** *Accompaniment to Figure 6.* Mean squared error over the 8 target models of each period, with 95% percentile-bootstrap intervals (2,000 resamples). Lowest error per row underlined. Cell colours show mean squared error on a logarithmic scale from 0.1 (blue) to 10+ (red).

Domain	**Target period**	**Baseline protocol**	**Deterministic features**	**Embedding**	**Absolute scoring**	**Pairwise judge**
Science	Early 2024	1.65 [1.05, 2.27]	1.66 [1.07, 2.29]	1.65 [1.05, 2.25]	1.61 [1.00, 2.20]	0.11 [0.01, 0.24]
	Late 2024	0.54 [0.36, 0.74]	0.58 [0.37, 0.84]	0.54 [0.37, 0.74]	0.46 [0.32, 0.63]	0.54 [0.36, 0.75]
	Early 2025	1.16 [0.61, 1.72]	38.06 [4.01, 82.83]	0.77 [0.38, 1.20]	4.14 [1.52, 6.84]	0.65 [0.24, 1.08]
	Late 2025	0.85 [0.46, 1.23]	0.28 [0.14, 0.46]	0.45 [0.26, 0.66]	0.42 [0.24, 0.63]	0.17 [0.02, 0.37]
	Early 2026	0.45 [0.27, 0.64]	0.57 [0.18, 1.11]	0.35 [0.18, 0.54]	0.20 [0.06, 0.37]	0.25 [0.13, 0.38]
Maths	Early 2024	1.61 [0.98, 2.25]	1.97 [0.81, 3.13]	1.42 [0.78, 2.14]	2.41 [1.19, 3.76]	1.61 [0.97, 2.23]
	Late 2024	0.53 [0.22, 0.92]	0.42 [0.09, 0.84]	0.35 [0.14, 0.62]	0.29 [0.07, 0.54]	0.11 [0.02, 0.22]
	Early 2025	3.41 [1.95, 4.95]	578.38 [71.89, 1377.78]	2.59 [1.30, 3.89]	3.03 [1.56, 4.53]	3.97 [1.76, 6.10]
	Late 2025	2.23 [1.21, 3.24]	0.89 [0.40, 1.41]	1.04 [0.50, 1.57]	1.51 [0.98, 2.02]	0.27 [0.07, 0.51]
	Early 2026	0.41 [0.24, 0.58]	0.63 [0.26, 1.05]	0.29 [0.18, 0.40]	0.20 [0.08, 0.34]	0.57 [0.30, 0.85]
Coding	Late 2025	2.65 [1.18, 3.91]	0.30 [0.09, 0.57]	2.65 [1.28, 3.94]	2.03 [1.05, 2.97]	0.17 [0.01, 0.43]
	Early 2026	1.64 [1.03, 2.34]	1.09 [0.49, 1.73]	0.74 [0.40, 1.11]	0.78 [0.39, 1.25]	0.11 [0.05, 0.20]
Cyber	Late 2025	1.08 [0.48, 1.75]	0.82 [0.27, 1.48]	1.08 [0.50, 1.76]	1.01 [0.45, 1.82]	0.32 [0.06, 0.69]
	Early 2026	1.70 [1.16, 2.34]	2.31 [1.41, 3.33]	1.49 [0.98, 2.09]	1.95 [1.28, 2.75]	0.20 [0.08, 0.33]

D.3 Specific choice of judge model does not affect prediction performance

Our main analyses use one judge for each prediction task. We typically consider the 3 strongest models of the previous reference semester, and pick as judge the one with the lowest leave-one-model-out prediction error on the other references. Table 8 shows the selected judge and other candidate judges.

We then test whether the results depend strongly on this judge-selection step. For each task, we repeat judging with each of the three strongest reference models, which are also the models used to define the saturated-item boundary. We compare squared prediction error across the 16 target models from 2025-II and 2026-I.

The three judges produce similar prediction errors (Figure 15). Changing the judge shifts a task's mean squared error by a median of 0.30, much smaller than the gap between pairwise judging and either absolute scoring (median 0.78) or the baseline (median 1.09). The main pairwise-judging results therefore do not depend on selecting a particular judge.

Table 8: **Candidate and picked judges.** For each target semester and domain, the three references (models from the preceding semester) with the highest global capability θ, which serve as candidate judges for pairwise judging and absolute scoring, ordered by reference leave-one-out error. The candidate with the lowest error on the reference set ("*Best*") is the judge used for the task.

Semester	Domain	Protocol	Best	2nd Best	3rd Best
2024-I	SCIENCE	Pairwise	*gpt-4-0613*	mixtral-8x7b-instruct-v0.1	nous-hermes-2-yi-34b
		Absolute	nous-hermes-2-yi-34b	*gpt-4-0613*	mixtral-8x7b-instruct-v0.1
	MATHS	Pairwise	*gpt-3.5-turbo-0125*	mixtral-8x7b-instruct-v0.1	*gpt-4-0613*
		Absolute	*gpt-3.5-turbo-0125*	*gpt-4-0613*	mixtral-8x7b-instruct-v0.1
2024-II	SCIENCE	Pairwise	*gpt-4o-2024-05-13*	*gpt-4-turbo-2024-04-09*	wizardlm-2-8x22b
		Absolute	*gpt-4o-2024-05-13*	*gpt-4-turbo-2024-04-09*	wizardlm-2-8x22b
	MATHS	Pairwise	*gpt-4-turbo-2024-04-09*	*gpt-4o-2024-05-13*	qwen2-72b-instruct
		Absolute	*gpt-4-turbo-2024-04-09*	qwen2-72b-instruct	*gpt-4o-2024-05-13*
2025-I	SCIENCE	Pairwise	qwen2.5-72b-instruct	deepseek-v3	phi-4
		Absolute	phi-4	qwen2.5-72b-instruct	deepseek-v3
	MATHS	Pairwise	llama-3.3-70b-instruct	deepseek-v3	phi-4
		Absolute	phi-4	deepseek-v3	llama-3.3-70b-instruct
2025-II	SCIENCE	Pairwise	minimax-m1	deepseek-r1-0528	qwen3-235b-a22b
		Absolute	deepseek-r1-0528	minimax-m1	qwen3-235b-a22b
	MATHS	Pairwise	deepseek-r1-0528	qwen3-235b-a22b	minimax-m1
		Absolute	qwen3-235b-a22b	minimax-m1	deepseek-r1-0528
	CODING	Pairwise	minimax-m1	deepseek-r1-0528	qwen3-235b-a22b
		Absolute	qwen3-235b-a22b	deepseek-r1-0528	minimax-m1
	CYBER	Pairwise	deepseek-r1-0528	qwen3-235b-a22b	minimax-m1
		Absolute	minimax-m1	qwen3-235b-a22b	deepseek-r1-0528
2026-I	SCIENCE	Pairwise	minimax-m2	glm-4.7	kimi-k2-thinking
		Absolute	glm-4.7	minimax-m2	kimi-k2-thinking
	MATHS	Pairwise	kimi-k2-thinking	glm-4.5	glm-4.7
		Absolute	glm-4.7	glm-4.5	kimi-k2-thinking
	CODING	Pairwise	kimi-k2-thinking	minimax-m2	deepseek-v3.2
		Absolute	deepseek-v3.2	minimax-m2	kimi-k2-thinking
	CYBER	Pairwise	kimi-k2-thinking	deepseek-v3.2	glm-4.7
		Absolute	deepseek-v3.2	glm-4.7	kimi-k2-thinking

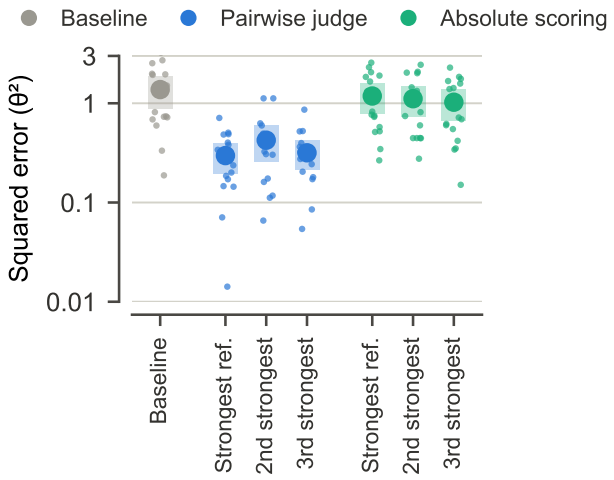


Figure 15: **Pairwise judging is robust to the choice of judge.** Squared prediction error for the 16 target models from 2025-II and 2026-I when using the strongest, second-strongest, or third-strongest reference model as judge. Small points show individual target models, averaged across domains; large circles show means and boxes show 95% t-intervals.

D.4 Pairwise judge predictions are best in Coding and Cyber domains

The main analysis pools prediction error across the four domains. To test whether protocol performance varies systematically by domain, we fit a hierarchical Gamma generalised linear model with a log link function to the squared prediction errors of the 16 target models from 2025-II and 2026-I across all domains. This gives 64 target-by-domain combinations, with one squared error per protocol and combination:

$$\texttt{error}^2 \sim \texttt{protocol} + \texttt{domain} + (\texttt{protocol} \mid \texttt{domain}) + (1 \mid \texttt{target} : \texttt{domain}) \quad (5)$$

The protocol term also includes a length-only protocol (which includes only the length terms of the deterministic features; see Table 5), fitted jointly with the other protocols.

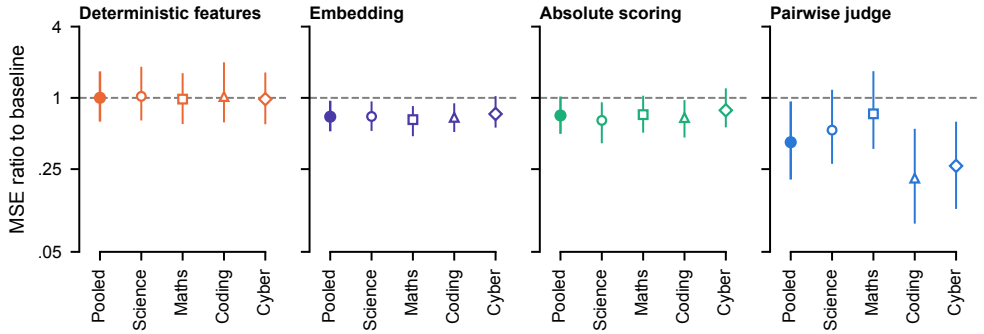


Figure 16: **Protocol efficacy varies across domains.** Posterior MSE ratio of each protocol relative to the baseline protocol, pooled across domains (*filled circles*) and separately by domain (*open markers*), estimated using the hierarchical Gamma GLM. Error bars show 95% credible intervals; the dashed line marks equal error to the baseline.

We use the baseline protocol as our protocol reference, with its effect fixed at zero; domain effects are normalised to sum to zero. Protocol-by-domain deviations $(\texttt{protocol} \mid \texttt{domain})$ are partially pooled separately for each protocol and drawn from a normal distribution with protocol-specific standard deviation σ_p. This captures variation of that protocol across domains. The target-by-domain term $(1 \mid \texttt{target} : \texttt{domain})$ is a random intercept accounting for the fact that all protocols predict the same target within each domain.

We use $\mathcal{N}(0, 2)$ priors for the intercept, $\mathcal{N}(0, 1)$ priors for protocol and domain effects, $\text{HalfNormal}(0.5)$ priors for σ_p, and a $\text{HalfNormal}(1)$ prior for the standard deviation of the target-by-domain random intercepts. We fit the model using NUTS with four chains, 1,000 warm-up draws, and 2,000 retained draws per chain. The model fitting showed no divergences, with $\hat{R} \leq 1.005$ and bulk effective sample size $\geq 1{,}090$.

Pairwise judging has the lowest pooled MSE ratio at 0.42 (95% credible interval 0.20–0.93), but this varies substantially by domain (Figure 16): it has clear predictive ability in Coding and Cyber, but is less reliable in Science and Maths (Table 9).

Table 9: **Hierarchical estimates of protocol performance by domain.** Posterior median MSE ratio relative to the Baseline protocol, with 95% credible intervals in brackets. σ_p shows the between-domain standard deviation.

Protocol	Pooled	Science	Maths	Coding	Cybersecurity	σ_p
Deterministic features	1.00 [0.63, 1.68]	1.03 [0.64, 1.83]	0.98 [0.60, 1.62]	1.03 [0.62, 2.00]	0.98 [0.60, 1.64]	0.17 [0.01, 0.70]
Embedding	0.69 [0.52, 0.95]	0.70 [0.53, 0.93]	**0.65 [0.47, 0.85]**	0.68 [0.51, 0.90]	0.73 [0.56, 1.03]	0.13 [0.01, 0.57]
Absolute scoring	0.71 [0.50, 1.03]	0.65 [0.41, 0.92]	0.72 [0.51, 1.04]	0.68 [0.46, 0.96]	0.79 [0.56, 1.20]	0.18 [0.01, 0.68]
Pairwise judging	**0.42 [0.20, 0.93]**	**0.53 [0.28, 1.17]**	0.73 [0.37, 1.68]	**0.21 [0.09, 0.55]**	**0.27 [0.12, 0.63]**	0.62 [0.12, 1.22]

In Coding and Cyber, pairwise judging also outperforms every other protocol, with pairwise-to-comparator MSE ratios between 0.20 and 0.36 and all corresponding intervals below one (Figure 17).

Embeddings provide a smaller but more consistent reduction in error, with a pooled MSE ratio of 0.69 (0.52–0.95) and domain-specific ratios between 0.65 and 0.73. Absolute scoring improves slightly on the baseline in SCIENCE and CODING (though less so than pairwise judging), while deterministic features show no clear improvement in any domain.

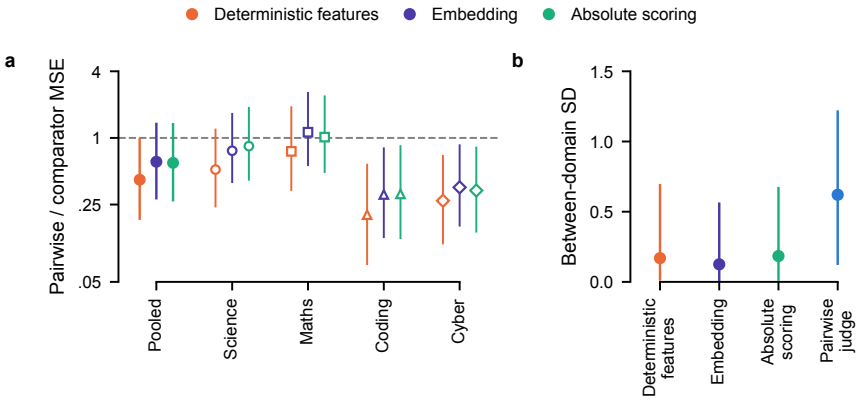


Figure 17: **Pairwise judges' prediction performance beats other protocols reliably in CODING and CYBER. a,** Ratio of pairwise judging relative to each other protocol, pooled and by domain; values below one favour pairwise judging. **b,** Between-domain standard deviation σ_p for each protocol on the log-MSE scale. Pairwise judging performance is also the most variable across domains. Error bars show 95% credible intervals.

The advantage of pairwise judging being concentrated in CODING and CYBER is consistent with this protocol working best in agentic tasks, but could also be related to subject matter. Additional data to directly compare agentic and non-agentic tasks within a domain is needed to disentangle these possibilities.

D.5 TRANSFER OF PREDICTIONS ACROSS DOMAINS

We ask whether capability signals extracted from one domain's transcripts transfer to capability prediction in another domain. For each ordered pair of domains, we calibrate pairwise verdicts from the source domain against the reference models' measured capability in the target domain, while keeping the judge fixed to the best judge selected for the source domain. We compare each cross-domain prediction with both the target domain's own transcripts and the baseline protocol.

Transcripts from the agentic domains transfer well to other domains (Figure 18). CODING and CYBER transcripts predict each other's capability better than the baseline, with squared errors of 0.35 and 0.33 compared with baseline errors of 2.14 and 1.39 ($p \leq .001$). They also predict MATHS and SCIENCE capability about as well as those domains' own transcripts, with cross-domain errors between 0.08 and 0.31 compared with within-domain errors of 0.42 and 0.21 ($p \geq .055$).

The reverse transfer is substantially weaker. MATHS and SCIENCE transcripts predict CODING capability with 18 and 11 times the within-domain error, respectively (2.52 and 1.59; $p \leq .015$), and predict CYBER capability with 4.4 and 3.6 times the within-domain error (1.16 and 0.94; $p \leq .025$). Neither transfer performs significantly better than the baseline ($p \geq .12$). Cross-domain transfer is therefore asymmetric: agentic transcripts retain capability information that transfers to non-agentic domains, while the reverse is not observed.

D.6 NUMBER OF EVALUATED ITEMS

We next test how prediction performance changes with the number of saturated items available to each protocol. This measures whether the protocols benefit from additional transcript evidence or whether performance is primarily limited by systematic errors in the extracted signal.

For both absolute scoring and pairwise judging, additional items improve prediction in some settings but provide little benefit in others. In at least two cases (pairwise judging in MATHS 2026-I

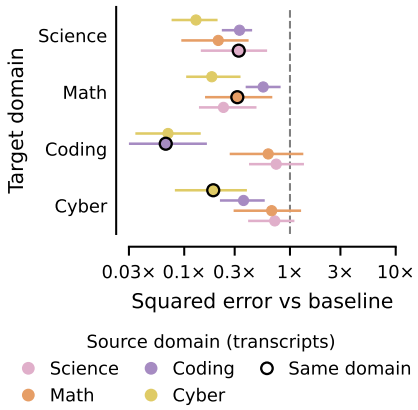


Figure 18: **Pairwise judging transfers asymmetrically across domains.** Squared prediction error relative to the saturated-item baseline when transcripts from one source domain are used to predict capability in another target domain. Outlined markers indicate within-domain predictions using transcripts from the target domain itself. Error bars show 95% intervals; the dashed line marks equal error to the baseline.

and absolute scoring in SCIENCE 2025-I), performance worsens as more items are added and does not subsequently recover. This is consistent with systematic judging errors being reinforced as additional judgements accumulate, rather than averaged away.

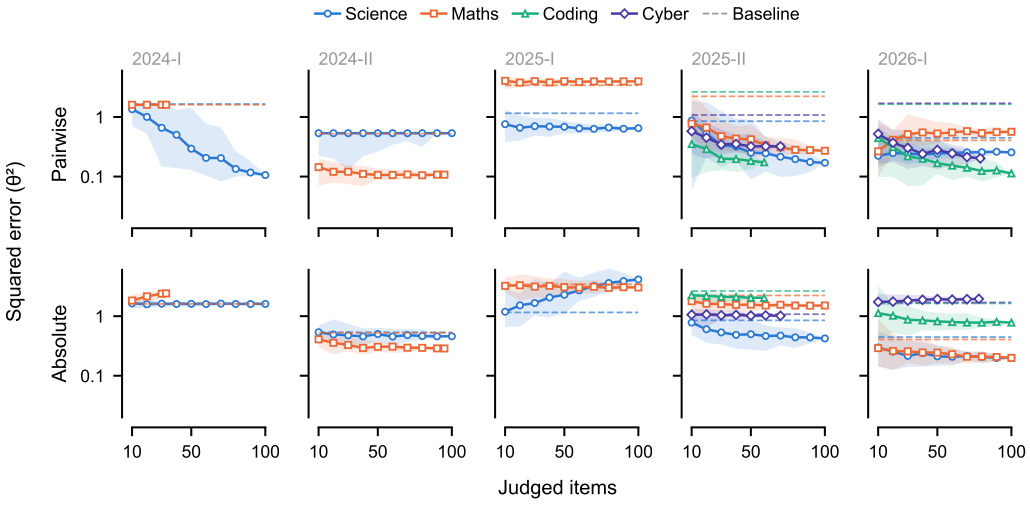


Figure 19: **Effect of the number of evaluated items.** Squared prediction error of pairwise judging and absolute scoring (the judge selected by reference leave-one-model-out error, as in the main analysis) as the number of judged saturated items varies across periods and domains: median over 30 random item subsets, band 2.5–97.5 percentile; *dashed line* is baseline protocol error of each task.

Overall, these results suggest that collecting more saturated transcripts does not necessarily resolve the main limitations of transcript-based capability estimation. For pairwise judging in particular, systematic judge bias can dominate sampling noise in some settings.

D.7 Shifts in the Reference Population

The prediction protocols are calibrated using a small set of reference models, so their performance may depend on which models are available for calibration. We test this sensitivity for pairwise judging by progressively restricting the reference population in two directions: retaining only the stronger reference models or retaining only the weaker reference models.

Removing the stronger references is substantially more damaging than removing the weaker ones. This suggests that high-capability references provide important anchors for locating target models on the capability scale, particularly when targets exceed much of the reference population.

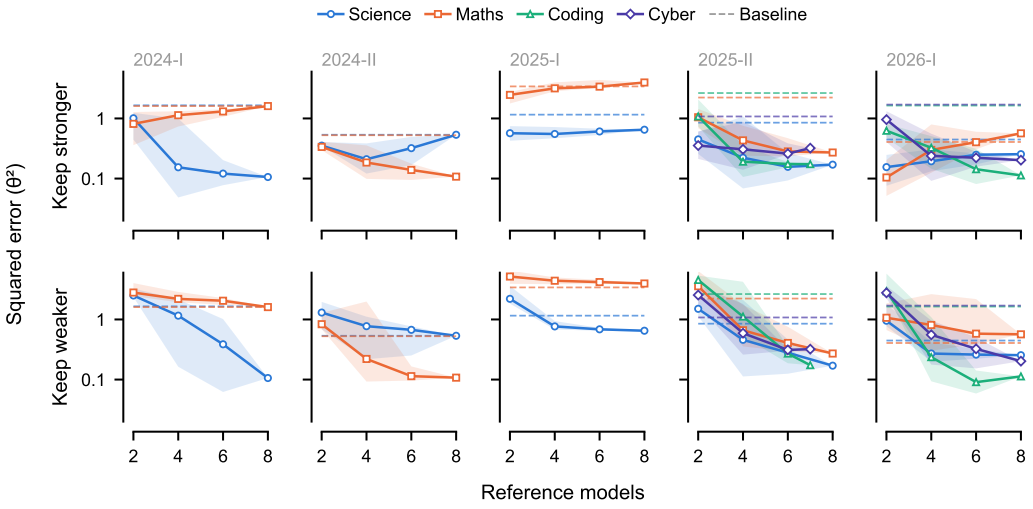


Figure 20: **Sensitivity to the reference population.** Squared prediction error of pairwise judging as the reference set is progressively restricted to stronger or weaker reference models: median over the third of reference subsets keeping the strongest (top) or weakest (bottom) models, band 2.5–97.5 percentile; *dashed-line*: the baseline protocol prediction for each task (all references).

Additionally, we tested whether excluding cases of self judging (removing any cases where pairwise judges and absolute scoring judged their own performance) modified predictions.

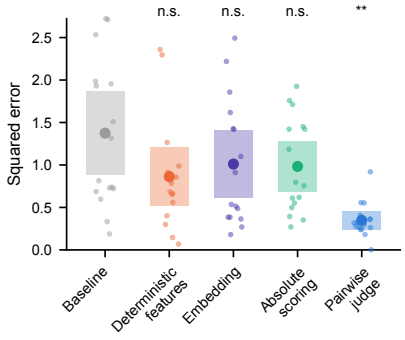


Figure 21: **Removal of self-judgements does not influence results.** Prediction performance of each protocol for 16 target models (squared error: lower is better), without self-judgements for absolute scoring and pairwise judging. This did not influence our results (cf. Figure 3). Each point is one target model; circle and box: mean and 95% CI.

D.8 Capability-Predicting Signal in Single Transcript Sections

We complement the section-removal tests in Section 4.2 with its converse: whether any third is *sufficient*, by showing judges only the first, middle, or last third of every transcript.

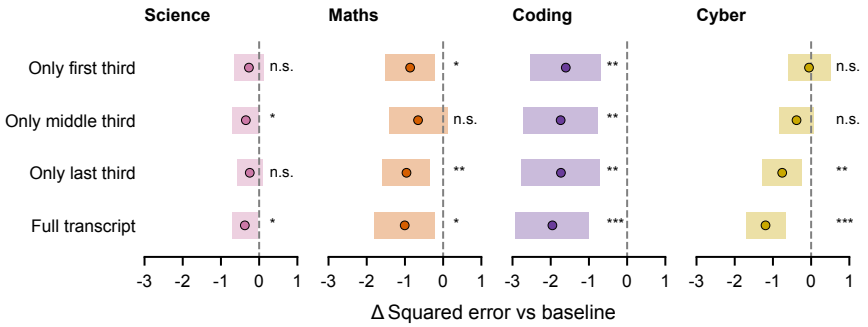


Figure 22: **Single transcript sections carry part of the capability information.** Change in prediction error of pairwise judging when only the first, middle, or last third of every transcript is shown, or the full transcript, relative to the baseline protocol (*dashed line*). Points show mean changes across target models and boxes show 95% paired t-intervals. *: $p < .05$; **: $p < .01$; ***: $p < .001$.

Overall, capability signals were distributed: shown any single third of transcripts, pairwise judging beat the baseline in most cases (Figure 22). In Coding, any third was sufficient (-1.61 to -1.74, $p \leq .003$). In Maths, the first and last thirds were (-0.86, $p = .012$; -0.96, $p = .005$), and in Science only the middle third was (-0.35, $p = .048$).

In Cyber however, capability-predicting information was concentrated towards the end of the transcript: only the final third outperformed the baseline (-0.76, $p = .007$); this is consistent with our results in Section 4.2, which show that removing the final third significantly degrades performance (Figure 4).

D.9 Capability-Predicting Signal Across Transcript Sections For Other Protocols

We repeat the transcript-position ablation from Section 4.2 for the remaining transcript-based protocols. Each protocol is refitted after removing the first, middle, or final third of every transcript, to test whether the positional pattern observed for pairwise judging holds for alternative representations of transcript information.

As for pairwise judging, most removals left prediction error unchanged (Figure 23). For deterministic features, removing the middle third increased error slightly in Science ($+0.19$, $p = .037$) and Maths ($+0.43$, $p < .001$). Embedding error changed by at most 0.07 in any domain: removing the first third increased it slightly in Science ($+0.04$, $p = .032$), and removing the middle (-0.05, $p = .012$) or last third (-0.07, $p = .044$) *decreased* it in Cyber– though note that the full-transcript error (1.34) is barely below that of the baseline protocol's prediction (1.39). For absolute scoring, no removal significantly increased error; removing the last third *decreased* it in Maths (-0.30, $p = .022$). No single third is consistently necessary for any of the three protocols across domains.

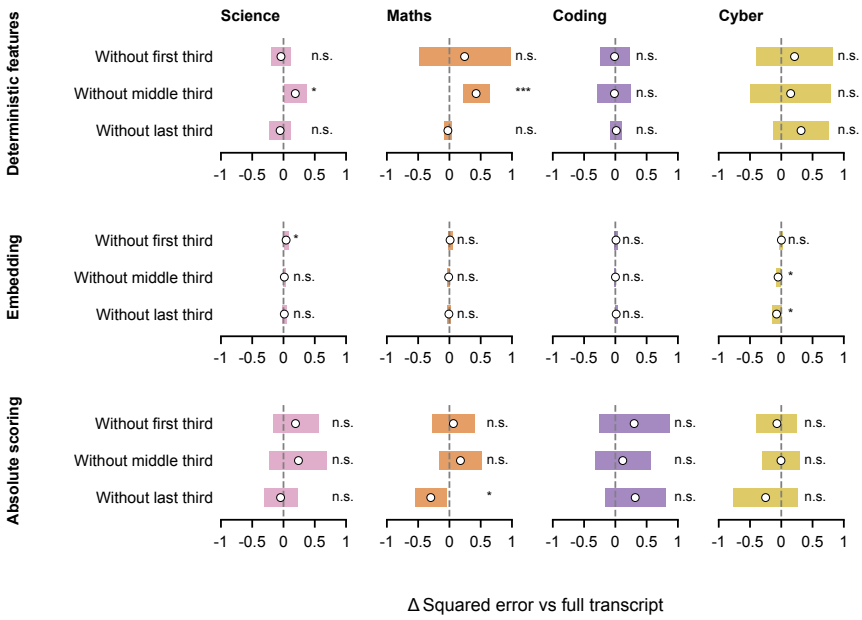


Figure 23: **Capability signals across transcript positions for deterministic features, embeddings and absolute scoring.** Change in squared prediction error for deterministic features (*top*), embeddings (*middle*) and absolute scoring (*bottom*) after removing the first, middle, or last third of every transcript, relative to the complete-transcript condition (*dashed line*). Points show mean changes across target models and boxes show 95% paired t-intervals. *: $p < .05$; ***: $p < .001$.

D.10 Capability-Predicting Signal Across Transcript Channels For Other Protocols

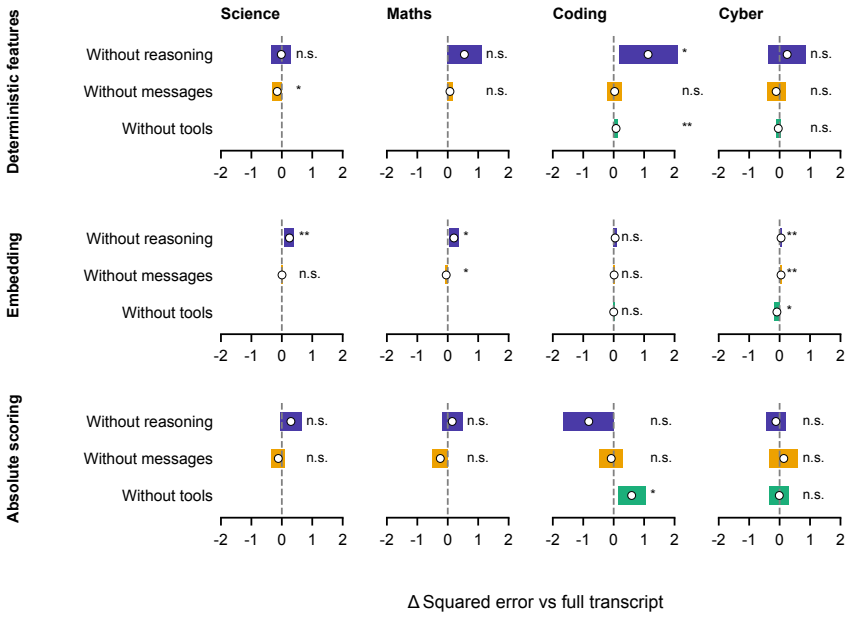


Figure 24: **Capability signals across channels for deterministic features, embeddings and absolute scoring.** Change in squared prediction error for deterministic features (*top*), embeddings (*middle*) and absolute scoring (*bottom*) after removing one transcript channel, relative to the complete-transcript condition (*dashed line*). Deterministic features and embeddings are refitted using the corresponding channel-removed reference transcripts; for absolute scoring, the judge re-rates the channel-removed transcripts and is recalibrated on the same references. MATHS and SCIENCE contain no tool calls. Points show mean changes across target models and boxes show 95% paired t-intervals. *: $p < .05$; **: $p < .01$.

We repeat the channel analysis from Section 4.2 for deterministic features, transcript embeddings, and absolute scoring to test whether the channels that matter for pairwise judging also carry signal for these alternative protocols. Each protocol is refitted after removing one channel from every transcript. Embeddings are recomputed from the channel-removed transcript, while deterministic features are recomputed from the remaining text and tool-use metadata.

We use the 16 target models from 2025-II and 2026-I in each domain. For each target, we compare squared prediction error under the channel-removed condition with error under the complete transcript using two-sided paired t-tests.

Unlike pairwise judging, no channel consistently carried signal for these protocols (Figure 24). For deterministic features, removing reasoning increased error only in CODING ($+1.13$, $p = .024$), and removing tools increased error slightly in CODING ($+0.08$, $p = .006$). Removing reasoning increased embedding error in SCIENCE, MATHS and CYBER ($+0.05$ to $+0.25$, $p \leq .012$), but all changes were small. For absolute scoring, only removing tools in CODING significantly increased error ($+0.59$, $p = .015$).

D.11 Results are robust to the Saturation Boundary

The main analysis defines an item as saturated when the three strongest reference models in a semester answer it correctly on their first attempt. Under this definition, weaker reference models and target models may still fail some saturated items. Since all protocols can only access correctly answered items, this can result in different models contributing slightly different subsets of transcripts. We therefore test whether the main results depend on this choice of saturation boundary.

We progressively tighten the definition. At level k, an item is considered saturated only if the k strongest reference models answer it correctly, for $k = 3, \ldots, 8$. In the strictest "all" condition, every reference *and* target model in the period must answer the item correctly. Tightening the boundary only removes items from the observed saturated set, so all analyses can be repeated using the existing transcripts, judge verdicts, ratings, features, and embeddings.

At each level, we refit θ^* for the reference, target, and historical models using the corresponding unsaturated set while holding the global IRT item parameters fixed. The level-3 refit reproduces the frozen main-analysis capability estimates to within 0.001. Because the unsaturated set changes with the boundary, the baseline prediction error also changes. We then refit and evaluate every protocol using the surviving observed items, selecting the judge independently at each level using leave-one-model-out error on the reference models. We report squared prediction error across the late-2025 and early-2026 target models relative to the corresponding baseline (Figure 25; Table 10).

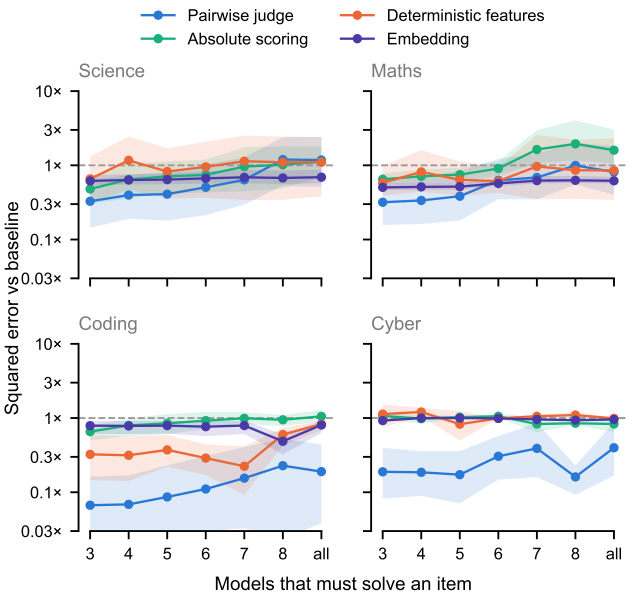


Figure 25: **Tightening the saturation boundary.** Squared prediction error of each protocol relative to the baseline as the number of models required to solve an item increases. Levels 3–8 require the k strongest reference models to solve the item; "all" requires every reference and target model in the period to solve it. θ^* is refitted at every level, and level 3 corresponds to the main-text setting. Bands show bootstrap 95% intervals across the 16 target models from late 2025 and early 2026.

Table 10: **Pairwise judging under stricter saturation boundaries.** Squared prediction error relative to the corresponding baseline at each saturation level, with θ^* refitted separately at every level. Late 2025 and early 2026 are pooled. The reported n values give the number of observed items summed across the two periods at level 3 and under the "all" condition.

	Models that must solve an item						
Target domain	3	4	5	6	7	8	all 16
Science (n: 200 → 140)	0.33	0.40	0.41	0.51	0.64	1.20	1.18
Maths (n: 200 → 102)	0.32	0.34	0.38	0.63	0.69	1.00	0.81
Coding (n: 159 → 52)	0.07	0.07	0.09	0.11	0.16	0.23	0.19
Cybersecurity (n: 149 → 94)	0.19	0.19	0.17	0.31	0.39	0.16	0.40
Pooled	0.19	0.20	0.22	0.32	0.39	0.65	0.49

Tightening the boundary progressively reduces the number of observed items, although the rate differs by domain. Across the two periods, SCIENCE retains 146 of 200 sampled items at $k = 8$

and 140 under the "all" condition; MATHS retains 109 and 102; CODING retains 37 and 52; and CYBER retains 66 and 94. Note that for CODING and CYBER, the 2025-II cohort contains only seven reference models, so the $k = 8$ condition includes only 2026-I.

The refitted capability estimates change most for weaker target models. For example, the SCIENCE capability of gpt-oss-20b falls from 0.01 at level 3 to -0.95 at level 8, while the strongest target models move by less than 0.05. Baseline error therefore also decreases as the boundary tightens, from 0.65 to 0.51 in SCIENCE and from 1.32 to 0.97 in MATHS.

Importantly, the ordering of protocols' performance is largely preserved. Pairwise judging has the lowest error at every boundary level in every domain except the strictest SCIENCE and MATHS conditions, while absolute scoring, deterministic features, and embeddings remain closer to the baseline. However, the magnitude of the pairwise advantage decreases under stricter saturation criteria. Pooled across domains, pairwise error rises from $0.19\times$ baseline error at level 3 to $0.32\times$ at $k = 6$, $0.65\times$ at $k = 8$, and $0.49\times$ under the "all" condition.

The agentic domains are comparatively robust: pairwise judging changes from $0.07\times$ to $0.19\times$ baseline error in CODING and from $0.19\times$ to $0.40\times$ in CYBER. In MATHS, the ratio reaches $1.00\times$ at $k = 8$ and $0.81\times$ under the "all" condition. SCIENCE is the main exception, increasing from $0.33\times$ at level 3 to $1.18\times$ under the strictest condition (95% CI 0.51–2.41).

Two effects occur simultaneously as the boundary tightens. First, the observed saturated set loses items that weaker models fail, which can remove items on which differences between strong and weak correct solutions are easiest to detect. Second, those items move into the unsaturated set used to measure θ^*, changing the capability target itself and generally compressing the target population towards the references. These effects are largest in SCIENCE, where tasks often produce short transcripts, and smallest in the agentic domains, where trajectories can differ substantially even when all models ultimately solve the same task.

D.12 PREDICTING ACROSS LONGER TIME DIFFERENCES

In our main analyses, reference models and judges are always the models of the preceding period. This emulates a scenario where a newly released target model deviates only moderately from the references in its release date. Here, we ask how far these capability predictions can be stretched, by progressively increasing the distance between reference and target models. We sweep the reference period back from late 2025 to 2023, while keeping our target period fixed in early 2026.

In SCIENCE, pairwise judging performance degrades as reference models get older. Prediction error rises from 0.25 with late 2025 references to 0.47 (early 2025) to 10.39 with 2023 reference models (Figure 26, left). It beats the baseline protocol with early 2025 ($p = .002$) and late 2024 references ($p < .001$), but not earlier references. Importantly, pairwise judging two periods back is worse than the baseline one period back.

This degradation in prediction could reflect the fact that future targets are far more capable than reference models, or simply a discrepancy between targets and references. To disentangle these possibilities, we fix the references at 2024-II or 2025-I and predict both older and newer targets, up to two periods away (Figure 26, right). We find that error grows with distance in both directions: targets two periods away were predicted worse than targets one period away (0.59 vs 0.14, $p = .024$), while older and newer targets at the same distance did not differ ($p \geq .52$). Degradation is thus dependent on the distance between reference and target models, irrespective of direction.

D.13 COST OF THE PROTOCOLS

We price each protocol by the tokens its judge consumed, at prices listed on OpenRouter (September 2026). For pairwise judging and absolute scoring, we trace costs as the number of judged items varies, similarly to Appendix D.6.

The protocols differ in cost by orders of magnitude. Deterministic features are free and embeddings cost a few cents per semester, while absolute scoring costs \$2–7 and pairwise judging \$55–105, because every item is judged once per target–reference pair (quadratic number of comparisons). Pairwise judging is the only protocol where additional spending generally buys accuracy. At the full

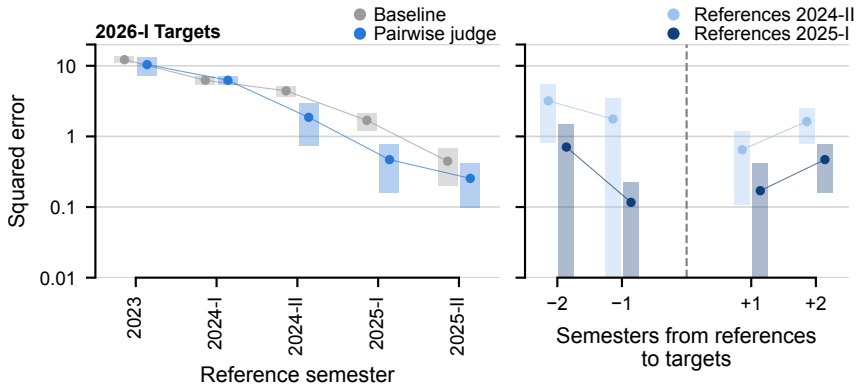


Figure 26: **Pairwise judging degrades with the distance between references and targets.** In SCIENCE, squared error per target; circle and box: mean and 95% t-interval over the eight targets of a task. *Left*: early 2026 targets predicted from each earlier reference period. *Right*: references fixed late 2024 (light) or early 2025 (dark), targets up to two periods older (negative) or newer (positive). Tasks one period ahead are those of the main text.

item set, it reaches $0.07\times$ baseline error in CODING, $0.19\times$ in CYBER, and $0.37\times$ in SCIENCE, and a tenth of the items (about $6 per semester) already beats the baseline in the agentic domains.

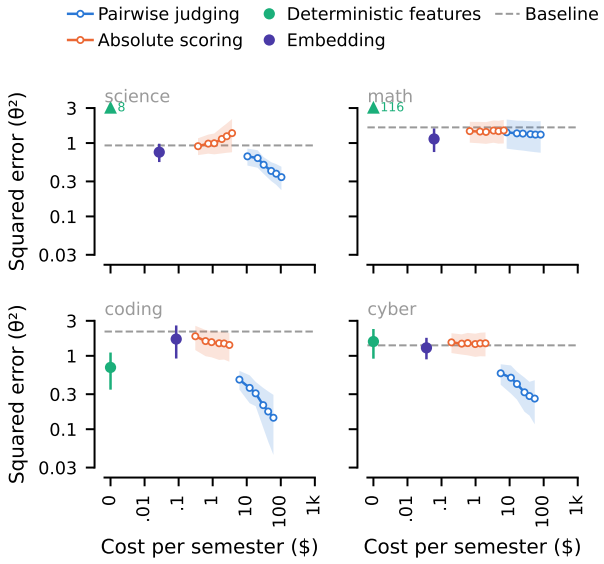


Figure 27: **Only pairwise judging turns cost into accuracy.** Prediction error against USD cost, pooled over every task of each domain and averaged over time periods. The judge is the reference with the lowest leave-one-out error. (*Dashed line*) is the baseline protocol. Pairwise judging and absolute scoring are shown as curves over 10–100% of the judged items, with cost proportional to the items judged. Deterministic features and embeddings are single points.

D.14 JUDGE ENSEMBLE

We additionally evaluate a judge ensemble that averages the predictions of all three candidate judges for each prediction task (a single domain and semester). We compare this ensemble with the single judge selected using leave-one-model-out prediction error (Figure 28).

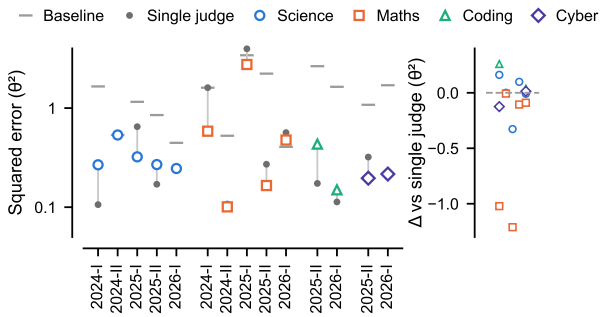


Figure 28: **Performance of the pairwise judge ensemble.** Squared prediction error of the ensemble of the three candidate judges (mean of their predictions; coloured markers) and of the judge selected by leave-one-model-out error (grey dots), on the items all three judged; grey tick, the baseline protocol. Right panel shows the paired difference (ensemble minus single judge; negative = ensemble is better). Ensemble error is generally similar to the single judge's, but substantially reduces the error of the selected judge in MATHS 2024-I (1.61 to 0.58) and 2025-I (3.97 to 2.75).