

Agreement Alone Selects the Wrong Grader: A Six-Axis Evaluation of 45 Language-Model Deployments for Rubric-Referenced Grammar Scoring of Learner Spanish and French

Jasper Vande Moortel
Orion-Solution, Belgium
jaspervandemoortel@orion-solution.net

Manuscript version 0.6. Reading edition: single column, one section per page.

Abstract

Published comparisons of large language models as graders rank them by agreement with a rubric or a rater. We show that for a grader that has to ship, agreement alone selects the wrong model. On an authored corpus of 24 transcribed learner utterances in Spanish and French with 29 planted grammatical errors, 45 chat-capable deployments on one cloud resource graded every item against a count-based teacher rubric rather than the descriptor-based rubrics common in language-proficiency grading, and six axes were recorded per deployment: quadratic weighted kappa, call success, parseability, latency, and cost. 22 deployments clear the conventional QWK 0.70 bar on the point estimate and 7 on the lower bound of the interval; four gates applied together ($\text{QWK} \geq 0.90$, every call returned, every response parseable, median latency under three seconds) leave 4 on the point estimate; none can be shown to clear the gate on the lower bound, because the sole passer has zero disagreements and a degenerate interval. Under item resampling three of the four point-estimate passers clear the QWK gate in 83–93% of resamples, so that count is not a stable shortlist. The three deployments with the highest agreement had exhausted the completion budget on three calls in ten, a shortfall a larger budget mostly resolves; another reached near-perfect agreement with a median latency above half a minute. A preliminary comparison of five grading architectures favours a typed error list scored by code over asking the model for the band, from the paired interval. All data are synthetic and every figure is an upper bound. The corpus, raw logs, analysis code and manuscript are released so the sweep can be re-run on any set of deployments.

Contents

1	Introduction	3
2	Related work	4
3	Method	6
3.1	Task and rubric	6
3.2	Corpus	6
3.3	Grading architectures	7
3.4	Deployment sweep	7
3.5	Statistics	8
4	Results	9
4.1	Which architecture	9
4.2	Repeatability	9
4.3	The sweep	10
4.4	Failures, re-examined	13
5	Discussion	14
6	Conclusion	15
	Limitations	16
A	Corpus	21
B	Prompts	23
C	Supplementary figures and tables	24
D	Failure classification	28
E	Completion-budget re-run	30
F	Full sweep	31
F.1	Robustness to the weighting scale	33
G	Cost derivation	35
H	Repeatability, per item	37
I	Literature search log	39

1 Introduction

A teacher who sets a two-minute speaking task for a class of thirty cannot mark it on a train. The recordings have to be listened to, the grammar has to be counted against whatever rubric the teacher wrote, and the judgement is the least reproducible of the four skills. The Flemish national assessment of French at the end of secondary education found this borne out in practice: speaking results were problematic in the arts and technical tracks, with fewer than half of pupils reaching most speaking goals, and the assessment called for sound speaking instruction in every school type (Vlaams Ministerie van Onderwijs en Vorming, 2013). Automated grading of the transcript is the obvious response, and large language models make the grammar part of it look easy: give the model the transcript and the rubric, get a score.¹

This paper is about choosing the model for that step, and its claim is that the obvious way to choose is wrong. Published comparisons of language models as graders rank them by agreement with a human or a rubric (Mizumoto and Eguchi, 2023; Yamashita, 2025). Agreement is necessary but not sufficient: a grader deployed in a school must return a score for every submission, in a form the system can read, within the time a teacher will wait, at a cost the school will pay. We evaluated 45 chat-capable deployments available on one cloud resource on six axes at once: the three deployments with the highest agreement had each exhausted the completion budget on three calls in ten, and another reached near-perfect agreement with a median latency above half a minute. Of 22 deployments that clear the conventional $\text{QWK} \geq 0.70$ bar on the point estimate, and 7 on the lower bound of the interval, 4 clear four deployment gates together on the point estimate, and none can be shown to clear them on the lower bound, because the sole zero-disagreement passer has a degenerate interval; under item resampling the point-estimate passers are not a stable set. The paper’s claim is that agreement alone would pass deployments the other gates remove, not that any fixed count is the shortlist a reader should deploy.

A second, preliminary result concerns how the grading call is structured. For a rubric that counts errors, asking the model to return a typed list of errors and computing the score by code agreed with the rubric better than asking the model for the score directly, and did so with fewer tokens per call in total. It is scoped to count-based rubrics and fixes the architecture the sweep uses.

The evidence base is small and synthetic: 24 authored utterances in Spanish and French with 29 planted errors, because no error-annotated corpus of spoken learner Spanish or French exists. Every number in the paper is therefore an upper bound and carries a bootstrap interval that is wide; we report the intervals and treat them, not the point estimates, as the result. The contribution is not a ranking of models, which will be stale within months, but the demonstration that model selection for grading has to gate on parseability, reliability and latency as well as agreement, with the measurements, the corpus, the code and the raw logs released so that the sweep can be re-run on any set of deployments.

Companion papers in the same series cover pronunciation assessment, tense detection on transcripts, and the correctness taxonomy and computable rubric the grader implements.

¹The corpus, raw logs, analysis code and manuscript source are at <https://github.com/anonymised/spoken-assessment-1ab>; a DOI is minted at camera-ready.

2 Related work

Language models as graders. Since Mizumoto and Eguchi (2023) scored L2 essays with GPT-3, general-purpose models have reached moderate to high agreement with human raters on holistic scales, with the agreement depending on how the task is posed: GPT-4 with calibration examples matches a modern automated writing evaluation system on CEFR-rated short essays, varying by the learner’s first language (Yancey et al., 2023); zero-shot GPT-4 analytic scores of L2 spoken English correlate with independent complexity features (Bannò et al., 2024); GPT-4o ratings carry a verbosity bias (Yamashita, 2025). Closest to our architecture question, Kucia et al. (2026) report a stable negative bias in analytic grammar scoring, so decomposing a rubric does not by itself calibrate grammar grading, and Yu et al. (2025) report, in legal question answering, that decomposed-criteria evaluation correlates far better with experts than pointwise holistic judgement. The nearest written-modality precedent for Spanish is Miller De Rutté and Correa (2026), who grade A1 Spanish learner writing with a task-customised GPT against a provided rubric. We found no study that tests an extract-errors-then-count architecture against holistic scoring on the same learner-language data, and none that grades *spoken* Spanish learner production with a language model; the queries are logged in Appendix I.

Judge consistency. LLM judges are inconsistent across samples and biased by position, verbosity and anchoring (Zheng et al., 2023; Stureborg et al., 2024); G-Eval and self-consistency voting reduce that noise by sampling and averaging (Liu et al., 2023; Wang et al., 2022), but temperature 0 does not make a commercial API deterministic (Ouyang et al., 2025) and determinism does not make a judge reliable (Schroeder and Wood-Doughty, 2024). Halder and Hockenmaier (2025) report low intra-rater reliability of LLM judges under sampling on their SummaC benchmark (Krippendorff’s alpha 0.33–0.79 across three models, lower still on MT-Bench); disabling sampling buys determinism at a cost of one to three points of agreement with humans depending on the model, and their remedy is majority voting over sampled runs with intra-rater reliability reported alongside agreement. Our repeatability measurement in Section 4.2 is the deterministic end of that trade-off: one model, one architecture, temperature 0, without their remedy.

Deployment axes beyond accuracy. Format-restricted generation degrades reasoning performance (Tam et al., 2024) and constrained-decoding schema compliance does not hold reliably across real-world schemas (Geng et al., 2025), so a grader’s ability to return the requested structure is a property to measure. Selecting models on accuracy against cost is established in NLP (Ong et al., 2024); Maurya et al. (2024) select on accuracy and report latency gains as a consequence rather than a selection criterion. Huang et al. (2025) score judge configurations on accuracy, reliability, cost and latency percentiles under a stated load (Agrawal et al., 2025); Frohn (2026) trade scoring accuracy against API cost for LLM scoring of mathematics assessments across five models and multiple configurations, reporting intra-model consistency but keeping their efficiency frontier to accuracy and cost, with no parseability or latency axis. The operational scoring tradition already treats a scorer as a deployed component: e-rater’s advisory flags route ungradeable responses away from machine scoring before agreement is computed (Attali and Burstein, 2006; Zhang et al., 2016), Williamson et al. (2012) make reliability, validity and practicality properties of a deployed system, the Duolingo English Test governs scoring as one pipeline stage (Settles

et al., 2020), and Speak & Improve, built on the corpus of Knill et al. (2024), runs its grader as a shared-task challenge (Qian et al., 2024). Sapkota and Murshed (2026) compares six deployments on a mathematics exam without a quantified reliability, latency or cost axis; this paper applies the operational stance to LLM graders.

Learner corpora and error counting. Learner corpora define an error either as one tag per surface deviation, the Cambridge Learner Corpus convention (Nicholls, 2003), or as the difference from a minimal target hypothesis, the MERLIN and Falko convention (Boyd et al., 2014; Reznicek et al., 2013); agreement on where errors are and how many there are is a separate question from agreement on their type (Dobrić, 2023). COWS-L2H is a corrected corpus of learner Spanish *writing* (Davidson et al., 2020); no spoken counted-error corpus exists for Spanish or French, and the ERRANT family has no port for either, in Qiu et al. (2025) or the twelve MultiGEC-2025 languages (Masciolini et al., 2025). Synthetic errors are the standard substitute, at the known cost that rule-based artificial errors are distributionally simpler than authentic ones (Felice and Yuan, 2014; Stahlberg and Kumar, 2021; Bryant et al., 2023); in speech, Korzekwa et al. (2022) report detector AUC rising from 0.928 to 1.00 with the severity of synthetic mispronunciations, above the same detector’s results on real non-native speech. That is why the figures in this paper are upper bounds.

Agreement statistics. Quadratic weighted kappa is the standard agreement statistic for ordinal rubric scores since the 2012 ASAP competition (Shermis and Hamner, 2013); the $QWK \geq 0.70$ deployment bar is conventionally attributed to Williamson et al. (2012), and the verbal bands of Landis and Koch (1977) are borrowed for it although defined for unweighted kappa. Both conventions are used here with those caveats.

3 Method

3.1 Task and rubric

The task is the one a teacher sets when a speaking assignment is graded for grammar: given a transcript of a learner utterance, count the grammatical errors and map the count to a band. We use a single rubric throughout (Table 4): a score of 10 for no error, then 9 for one, 8 for two, 6 for three, 4 for four and 2 for five or more. This count-based rubric is a deliberate simplification chosen for automated grading, not a transcription of Flemish classroom practice: the Flemish sources we could find frame grading of speaking, where they address it at all, as criteria with descriptor levels anchored to the CEFR rather than as an error tally (Centrum voor Taal en Onderwijs, 2026; Katholiek Onderwijs Vlaanderen, 2026). An error is one surface deviation that a native speaker would correct, so two separate wrong agreements count as two errors. This follows the Cambridge Learner Corpus convention of one tag per surface deviation (Nicholls, 2003) rather than the target-hypothesis convention of MERLIN and Falko (Boyd et al., 2014; Reznicek et al., 2013), in which the errors are whatever a minimal correction changes; the two conventions can disagree on a single item, and we return to this in Section 4.1.

The rubric scores grammatical accuracy only. Content, pronunciation, vocabulary range and fluency are outside the task and are handled by other components of the system this work belongs to (Section 1).

3.2 Corpus

No error-annotated corpus of spoken learner Spanish or French with counted errors exists (Section 2), so the corpus was authored. It holds 24 utterances, 12 per language, written as a Dutch-speaking Flemish secondary-school learner would say them: without sentence-final punctuation, with capitalisation only on proper nouns, and with the errors that the contrastive literature documents for this first language (gender and number agreement, tense and aspect, subjunctive triggers, clitic placement, false friends, elision, auxiliary choice). Two lexical items, *es09* and *fr09*, are English-mediated false friends rather than Dutch-mediated ones (*embarazadalembarrassed*, *actuellement/actually*): Flemish pupils learn English before Spanish or French, and a learner's second language transfers into lexical choices in a third language more readily than the first does (Jarvis and Pavlenko, 2008), matching Bardel and Falk (2007)'s L2-status factor for L3 transfer generally; Dutch teaching inventories list both pairs as English-mediated false friends (Enforex, 2026). The datasheet released with the corpus carries the full provenance, including an erratum on *es06* (its text lowercases a proper noun against the corpus's own capitalisation convention, so the model's second flagged edit was a defect in the item, not a grading error) and a documented key ambiguity on *es09* (the gold key counts the item's false friend as one error; the model consistently counts it as two surface deviations under the corpus's own one-tag-per-deviation convention). Table 8 reports the sweep's headline counts recomputed under both alternative keys. Each item carries a known set of planted errors, 29 in total across 17 types, and a minimal correction in the MERLIN sense. 2 items are deliberately correct and serve as false-positive controls. The full corpus is released with the packet under CC BY 4.0 and reproduced in Appendix A.

Because the errors are planted, the ground truth is exact by construction and the gold score is

a deterministic function of the planted count. Two consequences follow and bound every number below. First, authored errors are easier to detect than authentic learner errors (Felice and Yuan, 2014; Stahlberg and Kumar, 2021; Korzekwa et al., 2022), so every agreement figure is an upper bound on what the same grader would reach on real transcripts. Second, a rubric that counts errors structurally favours a grader that counts; the architecture comparison in Section 4.1 is therefore a result about count-based rubrics, not about holistic judgement in general.

3.3 Grading architectures

Five architectures were compared, each a different decomposition of the task rather than a prompt variant:

Holistic one call; the rubric is in the system prompt; the model returns a score and an error count as JSON.

Chain of thought as holistic, but the model lists the errors it finds before the score.

Extract-then-score the model returns only a typed list of errors (span, correction, label) as JSON; the count and the band are computed by code from the rubric. The rubric never passes through the model.

Target-hypothesis diff the model returns a minimally corrected utterance; the errors are the token-level edits between input and correction, counted by code.

Anchored as holistic, with three teacher-scored exemplar responses per language in the prompt, as a teacher calibrating a grader would supply them.

All five ran on the same model (gpt-5.4, Azure AI Foundry, global standard deployment) over all 24 items at temperature 0, one call per item. Prompts are reproduced verbatim in Appendix B. Repeatability was measured separately by running the extract-then-score architecture 5 times on the same items with the same parameters.

3.4 Deployment sweep

Every chat-capable deployment on one Azure AI Foundry resource in the Switzerland North region, 45 in total, graded all 24 items with the extract-then-score prompt, for 1080 calls, 6 at a time, with a completion budget of 1500 tokens. Parameter names differ across model families, so the client tried `max_completion_tokens` with and without a temperature, then `max_tokens`, and recorded which shape succeeded; every request, response, latency and token count is in the packet unchanged.

Six axes were recorded per deployment: agreement (quadratic weighted kappa, exact agreement, mean absolute error), call success, parseability, median and 95th-percentile latency over successful calls, and cost per thousand gradings from the mean token counts against the Azure retail price API, with the matched meter named for every priced deployment (Appendix G). Deployments returning fewer than 12 parseable responses are reported as under-sampled rather than scored, since a kappa on fewer items is noise. Failures were re-examined rather than accepted: deployments with failed calls were re-run serially with backoff, to separate rate limiting on this

account from model failure; Section 4.4 reports the classification and the one defect the re-run found in our own client.

Completion-budget re-run. Every deployment with at least one unparsed response at the 1500-token budget (8) was re-run on all 24 items, same prompt and parameters, at 8000 tokens, serially with backoff, under a pre-registration written before the run. One amendment was recorded during the run, before any result was read: a deployment whose first eight calls all returned and none parsed was stopped there as not a budget effect, which applied to Phi-4-reasoning alone. Results are in Section 4.3 and Appendix E.

3.5 Statistics

Quadratic weighted kappa (QWK) is computed on integer scores clipped to the rubric range over the items that parsed. Every QWK, exact-agreement and adjacent-agreement estimate carries a bias-corrected and accelerated bootstrap interval over items (2000 resamples, seed 20260909); call-success and parseability proportions carry Wilson intervals, a lower bound since failures cluster on three items (Section 4.3). Architecture differences are paired over items: the QWK difference carries a paired bootstrap interval and the mean-absolute-error difference a paired permutation test (5000 permutations). In the sweep, each deployment’s paired MAE comparison with gpt-5.4 is Benjamini–Hochberg corrected at 0.05 (details and the dependence assumption in Appendix F). The $\text{QWK} \geq 0.70$ bar is the field’s working convention, attributed to Williamson et al. (2012) although unconfirmed in that paper’s text; the $\text{QWK} \geq 0.90$ deployment gate is our own operational bar. With 24 items the intervals are wide, and the analysis plan, fixed before re-derivation, treats the interval rather than the point estimate as the result.

Table 1: Grading-architecture comparison on gpt-5.4, 24 items (95% BCa bootstrap intervals over items). p (MAE vs $extract$) is a paired sign-flip permutation test on absolute error against the extract row, uncorrected (– for extract itself). fp is the mean reported error count on the two deliberately correct control items (0 is ideal).

architecture	n	QWK [CI]	exact (%)	adjacent (%)	MAE	p (vs extract)	fp (controls)	mean tokens
extract	24	0.976 [0.667, 1.000]	95.8	100.0	0.04	–	0.00	128
cot	24	0.908 [0.655, 1.000]	83.3	100.0	0.17	0.239	0.00	244
anchored	24	0.863 [0.591, 0.984]	83.3	95.8	0.21	0.307	0.00	310
holistic	24	0.722 [0.593, 0.954]	70.8	91.7	0.46	0.032	0.00	190
th1	24	0.634 [0.112, 0.963]	79.2	95.8	0.29	0.187	0.50	105

4 Results

4.1 Which architecture

Table 1 and Fig. 1 give the five architectures on gpt-5.4. Extract-then-score has the highest point estimate on every axis: QWK 0.976 [0.667, 1.000] against 0.722 [0.593, 0.954] for the single holistic call, with the lowest mean absolute error, no phantom errors on the two control items, and fewer total tokens per call than the holistic call (128 against 190), because the rubric is not in its prompt; it spends more completion tokens than the holistic call (30 against 17) and fewer than chain of thought (61). The paired difference in QWK, holistic minus extract-then-score, is -0.254 with a bootstrap interval of [-0.388, -0.064], which excludes zero; the paired permutation test on mean absolute error against extract-then-score is reported in the p (vs $extract$) column of Table 1. With 24 items the intervals on the individual kappas overlap, and the reader should weigh the paired interval over the marginal ones. The gap between the two calls is a counting gap, not a rubric-application gap: mapping the holistic call’s own reported error count through the rubric by code gives the same score the holistic call gave, and the call miscounts 7 of 24 items against 1 for extract-then-score.

Two secondary observations. The target-hypothesis diff is the cheapest architecture and the only one that invented errors on both deliberately correct control items (0.50 phantom errors per control on average): a model asked to rewrite minimally still improves text that is merely plain, and every improvement becomes a counted error. Anchoring with three teacher-scored exemplars did not improve on chain of thought (paired interval [-0.279, 0.063] includes zero): a counting rule leaves exemplars nothing to calibrate, though they should matter for judged bands.

4.2 Repeatability

Five independent runs of extract-then-score on gpt-5.4 at temperature 0 returned an identical score on 23 of 24 items; the largest spread on any item was 1 point, and a majority vote of the five runs changed no score, on this one model and architecture (Fig. 4, Appendix C; per-item scores in Table 11). That is not the whole story on the two items that behaved unusually. es06 was scored 8 against a gold of 9 in every run: the model flagged the planted error and a second edit, the missing capital on a proper noun that this corpus’s own convention requires capitalised. With the item corrected to follow that convention, the model returns exactly the planted error and score 9 in 5 of

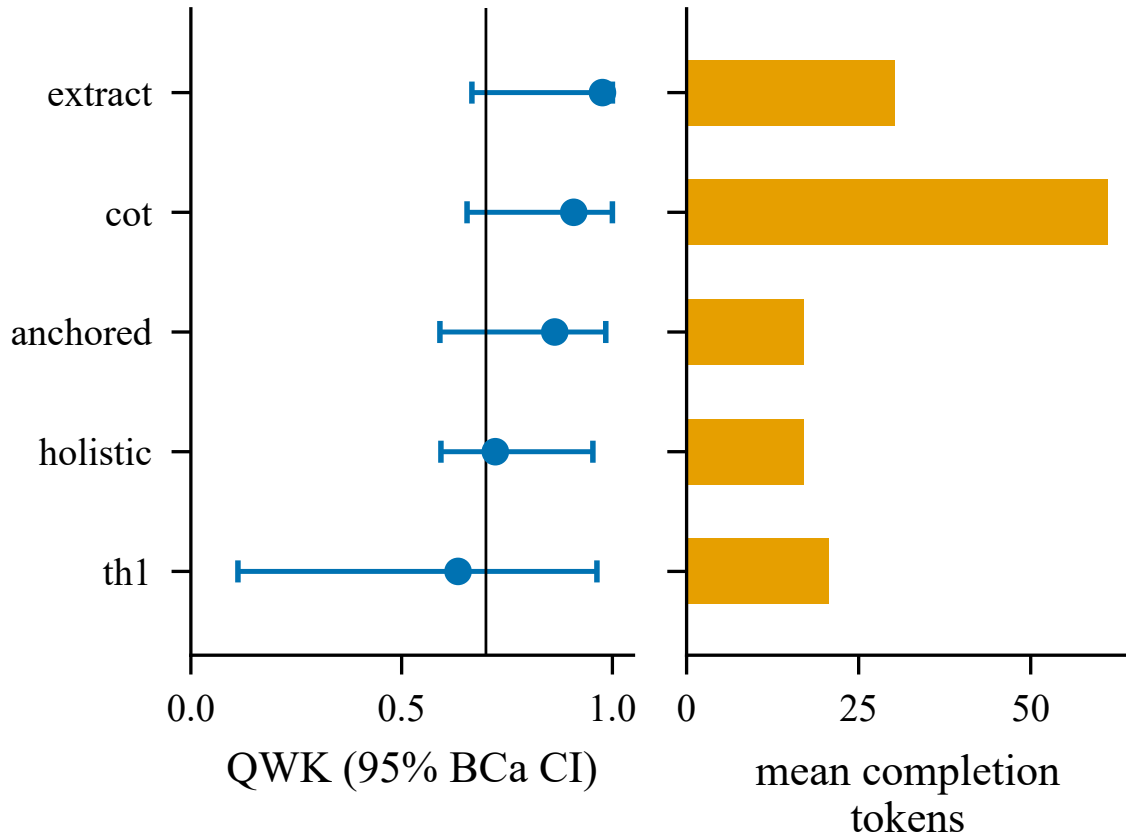


Figure 1: Five grading architectures on gpt-5.4 over the same 24 items. Left: QWK against the rubric with 95% BCa bootstrap intervals over items. Right: mean completion tokens per call. Extract-then-score returns a typed error list and leaves the arithmetic to code.

5 runs; the original item was a corpus defect, not a grading error. es09 was scored 8 with 2 flagged surface deviations in 5 of 5 runs, where the gold key counts the item’s false friend as one error, a key ambiguity the datasheet documents (Section 3.2); es09 also drifted across days, at 9, 9, 9, 8, 8 in the run reported in Table 11 and at 8 in all five re-runs. Haldar and Hockenmaier (2025) report the complementary case: low intra-rater reliability under sampling, with greedy decoding at temperature 0 buying self-consistency at a cost in accuracy. Their remedy, majority voting at temperature above zero with intra-rater reliability reported, was not tested here.

4.3 The sweep

Figure 2 and Table 7 (Appendix F) give every scored deployment and axis. On the point estimate, 22 of 45 deployments reach $\text{QWK} \geq 0.70$; on the lower bound of the interval, 7 do; scoring unparsed calls as failures instead of dropping them, 13 do. Applying the four gates together ($\text{QWK} \geq 0.90$, every call returned, every response parseable, median latency under three seconds) leaves 4 on the point estimate (Table 2); none can be shown to clear them on the lower bound, because the sole zero-disagreement passer, gpt-5.6-sol, has a degenerate bootstrap interval, and a rule-of-three bound on zero of 24 keeps its disagreement rate below 11.7% at 95%. Replacing the median-latency gate by a 95th-percentile gate of ten seconds admits 5. None of these counts is a stable

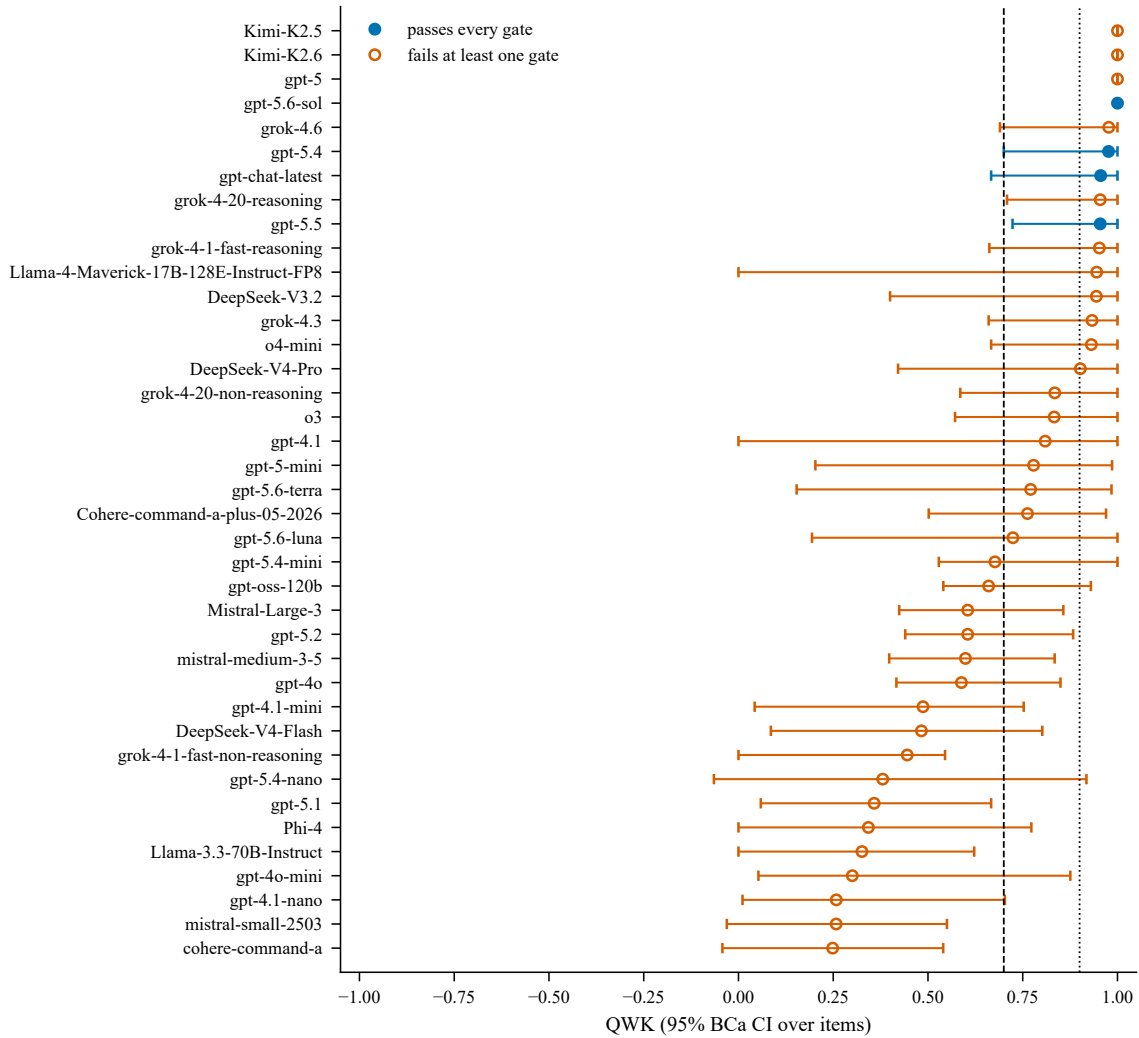


Figure 2: QWK against the rubric for every scored deployment (39 of 45; 6 returned fewer than 12 parseable responses and are not scored), with 95% BCa bootstrap intervals over items. Filled markers pass all four deployment gates. Dashed line: the conventional 0.70 bar; dotted: the 0.90 gate.

shortlist: under 2,000 item-bootstrap resamples, the three point-estimate passers with a genuine interval clear the 0.90 QWK gate in 93% (gpt-5.4), 83% (gpt-5.5) and 84% (gpt-chat-latest) of resamples, and the 95th-percentile-latency admission o4-mini clears it in 67%. Table 8 shows the same instability under the two disputed-item keys (Section 3.2): the passing set changes composition, dropping gpt-chat-latest under one key and admitting gpt-5.6-terra under the other, while the count stays near four, though 2 passers under no_es06 are zero-disagreement and degenerate on the 23 remaining items. QWK’s quadratic weights run over the raw, non-uniform 0–10 score; re-computing on band rank moves the 0.70-bar count to 28 of 45 against 22 on the raw score, though the gate shortlist is unchanged (Appendix F.1).

Three findings a single-axis comparison would have missed.

Table 2: Gate scoreboard: the models that pass every gate, plus five notable near-misses (95% BCa bootstrap intervals over items for QWK). Gates: QWK ≥ 0.90 , 100% call success, 100% parseability, p50 latency < 3 s. †: zero disagreements on all items; the interval is degenerate and carries no uncertainty.

model	QWK [CI]	call (%)	parsed (%)	p50 (s)	p95 (s)	EUR/1k	gates
gpt-5.4	0.976 [0.700, 1.000]	100.0	100.0	0.84	1.22	0.60	4/4
gpt-5.5	0.954 [0.723, 1.000]	100.0	100.0	2.26	5.86	4.81	4/4
gpt-5.6-sol	1.000 [1.000, 1.000]†	100.0	100.0	1.49	3.30	2.66	4/4
gpt-chat-latest	0.956 [0.667, 1.000]	100.0	100.0	1.25	2.13	–	4/4
Kimi-K2.5	1.000 [1.000, 1.000]†	100.0	70.8	7.46	13.92	–	2/4
Kimi-K2.6	1.000 [1.000, 1.000]†	100.0	70.8	7.95	20.47	–	2/4
gpt-5	1.000 [1.000, 1.000]†	100.0	75.0	11.26	17.76	–	2/4
Llama-4-Maverick-17B-128E-Instruct-FP8	0.945 [0.000, 1.000]	87.5	83.3	0.58	0.90	0.05	2/4
grok-4.6	0.977 [0.690, 1.000]	100.0	100.0	34.06	91.59	–	3/4

Parseability is a separate failure mode from agreement, measured under this budget. Kimi-K2.5, Kimi-K2.6 and gpt-5 each reach QWK 1.000 on the responses that parsed, and returned parseable JSON on 71, 71 and 75% of calls. Of the 20 unparsed calls from these three deployments, 19 ended exactly at the 1500-token completion budget, all but one with an empty body and that one truncated mid-object (7, 7 and 5 calls respectively; Table 5, *at budget* column): these are reasoning-style deployments that exhaust the completion budget on their reasoning tokens before emitting the answer, not a readability limit inherent to the model. The failures cluster on three items rather than spreading evenly: fr12 (15 of 45 models), fr11 (12) and es08 (11). Scoring the unparsed calls at the rubric floor of 2 gives QWK 0.130–0.148 for the trio; scoring them at each model’s median score gives 0.663–0.708. A benchmark that scores only parsed responses ranks these three deployments first. The re-run at 8000 tokens settles the mechanism (Table 6): the three returned parseable JSON on 100% of calls and reached QWK 0.953 [0.727, 1.000], 0.929 and 0.976 over all 24 items, with 95th-percentile completions of up to 3610 tokens. The larger budget made every other re-run deployment fully parseable except the Phi-4 pair: Phi-4-mini-reasoning returned every call and parsed none at 8000 tokens, while Phi-4-reasoning, stopped after eight calls under the futility rule (Appendix E), also parsed none. Parseability under a fixed budget is a property of the budget for the reasoning-style deployments and of the model for the Phi-4 pair, and one budget for every deployment misreads the first as the second.

Tail latency disqualifies otherwise strong deployments. grok-4.6 reaches QWK 0.977 with a median of 34.06 s and a 95th percentile of 91.59 s (Fig. 5, Appendix C).

Cheap open-weight deployments are competitive on agreement and fail on the operational axes. Llama-4-Maverick reaches QWK 0.945 at a median of 0.58 s and about EUR 0.05 per thousand gradings, and fails the gates on call success (88%) and parseability (83%). Figure 3 (Appendix C) places every priced deployment on cost against agreement.

The sweep does not separate a top group on mean absolute error at this sample size: of the 44 paired MAE comparisons against gpt-5.4, 7 are nominally significant at raw $p < 0.05$ and 0 survive Benjamini–Hochberg correction (q (BH) column, Table 7).

4.4 Failures, re-examined

10 deployments had at least one call fail outright in the concurrent run, and a further 10 answered every call without ever producing the requested JSON. Serial re-runs with backoff classified them (Table 5, Appendix D). Most were rate limits on this account’s small quota for non-OpenAI deployments under 6 concurrent calls: Mistral-Large-3 and mistral-small-2503 completed every call when paced and are reported from the paced run; Ministral-3B could not be measured at any pacing. Four apparent failures were a client defect, retrying the parameter shape on HTTP 400 when the Mistral endpoints return HTTP 422; after the fix all three answered normally. Three failures are genuine: the Phi-4 reasoning and mini deployments answer every call and essentially never produce the requested JSON, indistinguishable from a harness fault until re-run serially.

5 Discussion

The sweep’s headline is neither a ranking nor a shortlist: with 24 items the intervals are too wide to separate the top group: 22 deployments clear the 0.70 bar on the point estimate and 7 on the lower bound. The four-gate count is no more stable: 4 pass on the point estimate, and none can be shown to pass on the lower bound, because the sole zero-disagreement passer has a degenerate interval; under item resampling the three passers with a genuine interval clear the 0.90 QWK gate in as few as 83% of resamples (Section 4.3). The highest-agreement deployments were the least usable; gates belong in the same run, on the same items and client, or the ranking describes a different system than the one that ships.

For the architecture question, the result is narrower than the point estimates suggest and should be read from the paired interval. A typed error list scored by code agreed with the rubric better than asking for the band directly, with fewer tokens, and every score stayed auditable. That recommendation is for count-based rubrics (Section 3.1); it says nothing about judged bands, where exemplars may matter. The repeatability result cautions the same way: self-consistency is not correctness. The item that stayed off gold in every run was a defect in the item’s own convention, and the item that drifted turned on a disputable key; resampling changed nothing, so the remedy for a disputed score is an unbuilt teacher-override interface, not built or timed here. A school running fifteen hundred gradings a year pays under a euro on gpt-5.4 and about seven on gpt-5.5 (Table 2).

What the evidence licenses for a school is modest: a gated extract-then-score deployment agrees with a count-based rubric on authored transcripts near the conventional bar, and which deployments qualify moves under resampling and the disputed keys (Section 4.3). Whether it agrees on real transcripts from Flemish pupils, blind-graded by two teachers, is the planned next study.

6 Conclusion

We evaluated 45 deployments as graders of learner Spanish and French against a count-based grammar rubric on six axes. Agreement alone selects deployments that fail on parseability, reliability or latency: of 22 deployments (28 on band rank) that clear the agreement bar on the point estimate and 7 on the lower bound, 4 clear all four gates on the point estimate and none demonstrably on the lower bound, and under resampling the count is not a stable shortlist. A preliminary comparison favours code-scored typed error lists for count-based rubrics. All corpora are synthetic and every figure an upper bound.

Limitations

All corpora are synthetic. The 24 items were authored to carry known errors; authored errors are easier to detect than authentic learner errors, and no figure in this paper is an estimate of agreement on real transcripts. The human criterion is absent: agreement is measured against a rubric applied mechanically to the planted count, which is reliability against a specification, not validity against teacher judgement. With 24 items the bootstrap intervals on kappa span several tenths for the weaker deployments, and the tables report those intervals rather than the point estimates. The rubric counts errors, which favours a grader that counts, so the architecture result applies to count-based rubrics and says nothing about judged bands, for which anchored exemplars may well matter. The sweep reflects one Azure AI Foundry resource, one region, one account's quota and one week in September 2026; availability, prices and behaviour of every deployment change, and the reliability column in particular reflects this account's provisioning as much as any model. Latency was measured from a single client under light concurrency, not under classroom load. Cost is computed for global standard deployments from a retail-price snapshot, and thirteen deployments could not be priced. The comparison covers models available on one provider's catalogue; models not deployable there were not tested. The sweep used a fixed 1500-token completion budget; reasoning-style deployments exhaust that budget on their reasoning tokens before emitting an answer on some items, so the parseability figures for those deployments are a property of the budget as much as of the model; the re-run at 8000 tokens measured that for 8 deployments and one budget, and its latency and cost figures are not comparable with the sweep's. The planned follow-up (Section 5) would score speech from pupils, who are minors; the harm the teacher-override interface is meant to prevent is a wrong count reaching a pupil's record unreviewed. The EU AI Act's Annex III lists education and vocational training, including systems used to evaluate learning outcomes, as high-risk; a deployed grader in that role would fall under it, and this paper evaluates the grading engine in isolation, not a compliant deployment. Consent and DPIA procedures for the planned pupil study have not begun.

References

- Amey Agrawal, Nitin Kedia, Anmol Agarwal, Jayashree Mohan, Nipun Kwatra, Souvik Kundu, Ramachandran Ramjee, and Alexey Tumanov. 2025. On evaluating performance of LLM inference serving systems. arXiv:2507.09019. *arXiv preprint*.
- Attali and Burstein. 2006. Automated essay scoring with e-rater V.2. <https://ejournals.bc.edu/index.php/jtla/article/view/1650>. Accessed September 10, 2026.
- Stefano Bannò, Hari Krishna Vydana, Kate M. Knill, and Mark J. F. Gales. 2024. Can GPT-4 do L2 analytic assessment? arXiv:2404.18557. *arXiv preprint*.
- Camilla Bardel and Ylva Falk. 2007. The role of the second language in third language acquisition: the case of germanic syntax. *Second Language Research*, 23(4):459–484.
- Boyd, Hana, Nicolas, Meurers, Wisniewski, Abel, Schöne, Štindlová, and Vettori. 2014. The MERLIN corpus: Learner language and the CEFR. <https://aclanthology.org/L14-1488/>. Accessed September 10, 2026.
- Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, pages 1–59.
- KU Leuven Centrum voor Taal en Onderwijs. 2026. Rubrics (aan de slag met taalbeleid). <https://www.taalbeleid.be/nl/aan-de-slag/rubrics>. Accessed September 10, 2026.
- Davidson, Yamada, Fernández-Mira, Carando, Sánchez-Gutiérrez, and Sagae. 2020. Developing NLP tools with a new corpus of learner spanish. <https://aclanthology.org/2020.lrec-1.894/>. Accessed September 10, 2026.
- Nikola Dobrić. 2023. Identifying errors in a learner corpus - the two stages of error location vs. error description and consequences for measuring and reporting inter-annotator agreement. *Applied Corpus Linguistics*, 3(1):100039.
- Enforex. 2026. Spaanse valse vrienden. <https://www.enforex.com/spaans/taal/spaanse-valse-vrienden.html>. Accessed September 10, 2026.
- Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126. Association for Computational Linguistics.
- Scott Frohn. 2026. The impact of LLM Self-Consistency and reasoning effort on automated scoring accuracy and cost. arXiv:2604.26954. *arXiv preprint*.
- Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. 2025. Jonschemabench: A rigorous benchmark of structured outputs for language models. arXiv:2501.10868. *arXiv preprint*.
- Rajarshi Haldar and Julia Hockenmaier. 2025. Rating roulette: Self-Inconsistency in LLM-As-A-Judge frameworks. arXiv:2510.27106. *arXiv preprint*.

- Donghao Huang, Shila Chew, Anna Dutkiewicz, and Zhaoxia Wang. 2025. Llm-as-a-judge for scalable test coverage evaluation: Accuracy, operational reliability, and cost. *arXiv:2512.01232. arXiv preprint*.
- Scott Jarvis and Aneta Pavlenko. 2008. *Crosslinguistic Influence in Language and Cognition*. Routledge.
- Katholiek Onderwijs Vlaanderen. 2026. Taalonderwijs frans: aan de slag in de klas (evaluatie). <https://pro.katholiekonderwijs.vlaanderen/taalonderwijs-frans/aan-de-slag-in-de-klas>. Accessed September 10, 2026.
- Kate Knill, Diane Nicholls, Mark J. F. Gales, Mengjie Qian, and Pawel Stroinski. 2024. Speak & improve corpus 2025: an L2 english speech corpus for language assessment research. *arXiv:2412.11986. arXiv preprint*.
- Daniel Korzekwa, Jaime Lorenzo-Trueba, Thomas Drugman, and Bozena Kostek. 2022. Computer-assisted pronunciation training. speech synthesis is almost all you need. *Speech Communication*, 142:22–33.
- Filip J. Kucia, Anirban Chakraborty, and Anna Wróblewska. 2026. Llm essay scoring under holistic and analytic rubrics: Prompt effects and bias. *arXiv:2604.00259. arXiv preprint*.
- J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522. Association for Computational Linguistics.
- Masciolini, Caines, De Clercq, Kruijsbergen, Kurfalı, Muñoz Sánchez, Volodina, and Östling. 2025. The MultiGEC-2025 shared task on multilingual grammatical error correction. <https://aclanthology.org/2025.nlp4call-1.1/>. Accessed September 10, 2026.
- Kaushal Kumar Maurya, KV Aditya Srivatsa, and Ekaterina Kochmar. 2024. Selectllm: Query-aware efficient selection algorithm for large language models. *arXiv:2408.08545. arXiv preprint*.
- Alyssia Miller De Rutté and Maite Correa. 2026. Evaluating GPT-generated feedback for beginning-level spanish writing: An exploratory study. *ReCALL*, 38(2):274–290.
- Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an AI language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.
- Diane Nicholls. 2003. The cambridge learner corpus: Error coding and analysis for lexicography and ELT. <https://ucrel.lancs.ac.uk/publications/CL2003/papers/nicholls.pdf>. Accessed September 10, 2026.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. 2024. Routellm: Learning to route LLMs with preference data. *arXiv:2406.18665. arXiv preprint*.

- Shuyin Ouyang, Jie M. Zhang, Mark Harman, and Meng Wang. 2025. An empirical study of the non-determinism of ChatGPT in code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2):1–28.
- Mengjie Qian, Kate Knill, Stefano Banno, Siyuan Tang, Penny Karanasou, Mark J. F. Gales, and Diane Nicholls. 2024. The speak & improve challenge 2025: Tasks and baseline systems. arXiv:2412.11985. *arXiv preprint*.
- Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, Siliang Liu, and Jungyeul Park. 2025. Multilingual grammatical error annotation: Combining Language-Agnostic framework with Language-Specific flexibility. arXiv:2506.07719. *arXiv preprint*.
- Marc Reznicek, Anke Lüdeling, and Hagen Hirschmann. 2013. *Competing target hypotheses in the Falko corpus: A flexible multi-layer corpus architecture*, pages 101–124. John Benjamins Publishing Company.
- Aastha Sapkota and M. G. Sarwar Murshed. 2026. Llms as teaching assistants for mathematics exam grading: Reliability, and practical usability. arXiv:2607.01247. *arXiv preprint*.
- Kayla Schroeder and Zach Wood-Doughty. 2024. Can you trust LLM judgments? reliability of LLM-as-a-judge. arXiv:2412.12509. *arXiv preprint*.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.
- Mark D. Shermis and Ben Hamner. 2013. Contrasting state-of-the-art automated scoring of essays. <https://www.routledge.com/Handbook-of-Automated-Essay-Evaluation-Current-Applications-and-New-Directions/Shermis-Burstein/p/book/9780415810968>. Accessed September 10, 2026.
- Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. arXiv:2105.13318. *arXiv preprint*.
- Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. arXiv:2405.01724. *arXiv preprint*.
- Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? a study on the impact of format restrictions on performance of large language models. arXiv:2408.02442. *arXiv preprint*.
- Vlaams Ministerie van Onderwijs en Vorming. 2013. Peiling frans in de derde graad aso, kso en tso van het secundair onderwijs: luisteren en spreken. https://docs.vlaamsparlement.be/docs/biblio/ependigibib/monografie/2013/728_peiling_frans.pdf. Accessed September 10, 2026.
- Vlaamse Overheid. 2021. Toolkit breed evalueren. <http://web.archive.org/web/20160508172016/http://onderwijs.vlaanderen.be:80/sites/default/files/atoms/files/Toolkit%20breed%20evalueren%20competenties%20Nederlands%20in%20het%20secundair%20onderwijs.pdf>. Accessed September 10, 2026.

- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. arXiv:2203.11171. *arXiv preprint*.
- David M. Williamson, Xiaoming Xi, and F. Jay Breyer. 2012. A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, 31(1):2–13.
- Taichi Yamashita. 2025. Exploring potential biases in GPT-4o’s ratings of english language learners’ essays. *Language Testing*, 42(3):344–358.
- Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short L2 essays on the CEFR scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584. Association for Computational Linguistics.
- Fangyi Yu, Nabeel Seedat, Dasha Herrmannova, Frank Schilder, and Jonathan Richard Schwarz. 2025. Beyond pointwise scores: Decomposed Criteria-Based evaluation of LLM responses. arXiv:2509.16093. *arXiv preprint*.
- Mo Zhang, Jing Chen, and Chunyi Ruan. 2016. Evaluating the advisory flags and machine scoring difficulty in the e-rater automated scoring engine. *ETS Research Report Series*, 2016(2):1–14.
- 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. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. arXiv:2306.05685. *arXiv preprint*.

A Corpus

Table 3 lists every item with its minimal correction and planted errors. The JSON release carries the same fields plus the gold score and the gloss explaining why each error is typical of the population.

Table 3: Every corpus item as spoken, its minimal correction (target hypothesis) and the planted errors. Control items carry no error.

id	utterance as spoken	minimal correction	planted errors
es01	mi hermana es muy cansada porque trabaja mucho	mi hermana está muy cansada porque trabaja mucho	ser_estar (es → está)
es02	ayer yo voy al mercado y compro tres manzana	ayer yo fui al mercado y compré tres manzanas	tense_preterite (voy → fui); tense_preterite (compro → compré); number_agreement (manzana → manzanas)
es03	el problema es que la mesa blanco no es muy grande	el problema es que la mesa blanca no es muy grande	gender_agreement (blanco → blanca)
es04	espero que mi amigo viene a la fiesta mañana	espero que mi amigo venga a la fiesta mañana	subjunctive (viene → venga)
es05	estudio español para dos años y me gusta mucho	estudio español desde hace dos años y me gusta mucho	duration_expression (para → desde hace)
es06	cuando era pequeño yo fui al colegio en bruseles todos los días	cuando era pequeño yo iba al colegio en bruseles todos los días	aspect_pret_imperf (fui → iba)
es07	me gusta mucho los perros pero no tengo un	me gustan mucho los perros pero no tengo ninguno	verb_agreement (gusta → gustan); pronoun (un → ninguno)
es08	mi madre cocina la cena y después nosotros comemos lo juntos	mi madre cocina la cena y después nosotros la comemos juntos	clitic_form_position (comemos lo → la comemos)
es09	soy embarazada de hablar español delante de la clase	me da vergüenza hablar español delante de la clase	false_friend (soy embarazada → me da vergüenza)
es10	hace dos semanas he visitado a mi abuela en el pueblo	hace dos semanas visité a mi abuela en el pueblo	tense_preterite (he visitado → visité)
es11	la comida estaba muy rico y también muy barato	la comida estaba muy rica y también muy barata	gender_agreement (rico → rica); gender_agreement (barato → barata)
es12	normalmente yo me levanto a las siete y desayuno con mi familia	normalmente yo me levanto a las siete y desayuno con mi familia	<i>none (control)</i>
fr01	ma soeur est très fatigué parce qu'elle travaille beaucoup	ma soeur est très fatiguée parce qu'elle travaille beaucoup	gender_agreement (fatigué → fatiguée)
fr02	hier je vais au marché et j'achète trois pomme	hier je suis allé au marché et j'ai acheté trois pommes	tense_passe_compose (vais → suis allé); tense_passe_compose (achète → ai acheté); number_agreement (pomme → pommes)
fr03	j'ai allé à l'école en bus ce matin	je suis allé à l'école en bus ce matin	auxiliary_choice (j'ai allé → je suis allé)
fr04	il faut que je fais mes devoirs avant de sortir	il faut que je fasse mes devoirs avant de sortir	subjunctive (fais → fasse)
fr05	je ne mange pas de la viande mais je mange du poisson	je ne mange pas de viande mais je mange du poisson	partitive_article (de la viande → de viande)
fr06	quand j'étais petit j'ai joué au football tous les mercredis	quand j'étais petit je jouais au football tous les mercredis	aspect_pc_imparfait (j'ai joué → je jouais)
fr07	je l'ai vu hier et je lui ai donné le livre que il voulait	je l'ai vu hier et je lui ai donné le livre qu'il voulait	elision (que il → qu'il)
fr08	ma mère prépare le dîner et après nous mangeons le ensemble	ma mère prépare le dîner et après nous le mangeons ensemble	clitic_position (mangeons le → le mangeons)
fr09	actuellement je pensais que le film était très ennuyeux	en fait je pensais que le film était très ennuyeux	false_friend (actuellement → en fait)
fr10	je suis étudiant depuis trois ans et je aime beaucoup ça	je suis étudiant depuis trois ans et j'aime beaucoup ça	elision (je aime → j'aime)
fr11	les grand maisons blanches sont très belle	les grandes maisons blanches sont très belles	number_agreement (grand → grandes); number_agreement (belle → belles)
fr12	normalement je me lève à sept heures et je prends mon petit déjeuner	normalement je me lève à sept heures et je prends mon petit déjeuner	<i>none (control)</i>

B Prompts

All prompts are reproduced verbatim from the experiment scripts in the packet. {lang} is “Spanish” or “French”. The user turn is always Transcript: <utterance>, except for the target-hypothesis architecture, whose user turn is the bare utterance.

Rubric (shared by holistic, chain of thought and anchored).

Score the response from 0 to 10 for GRAMMATICAL ACCURACY only. Do not judge content, ideas, pronunciation or vocabulary range. Band definitions the teacher has set: 10 = at most 0 grammatical errors; 9 = exactly 1 grammatical error; 8 = exactly 2 grammatical errors; 6 = 3 grammatical errors; 4 = 4 grammatical errors; 2 = 5 or more grammatical errors. An error is one surface deviation that a native speaker would correct. Two separate wrong agreements are two errors, not one.

Holistic.

You grade {lang} learner speech. <rubric> Reply with JSON only: {"score": <int 0-10>, "n_errors": <int>}

Chain of thought.

You grade {lang} learner speech. <rubric> First list each grammatical error you find, then give the score. End with JSON only: {"score": <int>, "n_errors": <int>}

Extract-then-score (also the sweep prompt).

You find grammatical errors in {lang} learner speech. Return JSON only: {"errors":[{"span":"<exact text>","correction":"<text>","type":"<short label>"}]}. One entry per surface deviation a native speaker would correct. Do not flag style, register, word choice or missing punctuation. If the sentence is correct, return an empty list.

Target-hypothesis diff.

Correct this {lang} learner utterance with the MINIMUM possible changes. Fix only grammar and morphology. Keep every word choice, every idea and the word order unless the word order is itself ungrammatical. Do not improve style. Do not add punctuation. Return JSON only: {"corrected": "<text>"}

Anchored.

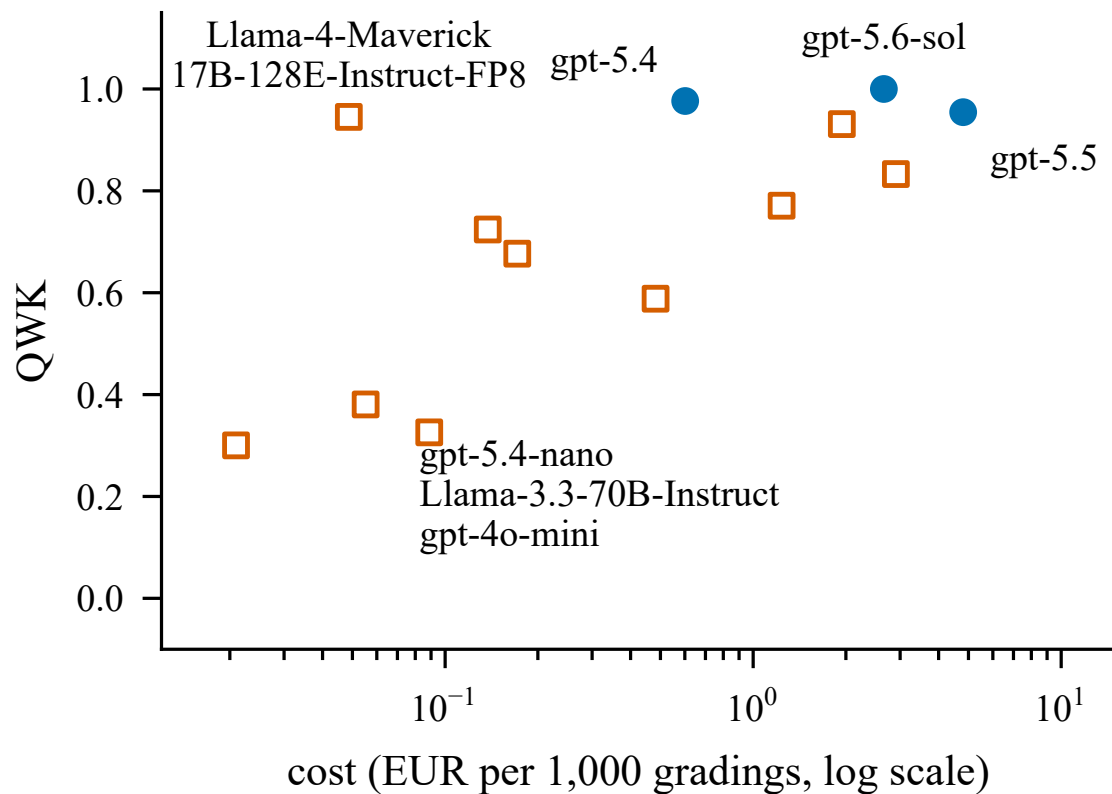
You grade {lang} learner speech the way THIS teacher grades. <rubric> Here is how the teacher scored three real responses: <three exemplars, each as: Example (score N/10): "<text>" -> <reason> Reply with JSON only: {"score": <int 0-10>, "n_errors": <int>}

The three exemplars per language are in the packet (experiments/errorset.py, ANCHORS).

Table 4: Grammar-error corpus used for Experiments 2 and 6: every item carries a known, planted error set and a gold score under the teacher’s rubric (Section 3.2).

quantity	value
items	24
Spanish items	12
French items	12
planted errors	29
distinct error types	17
control (error-free) items	2
gold score range	6–10

C Supplementary figures and tables



No matched Azure meter (omitted): Kimi-K2.5, Kimi-K2.6, gpt-5 -- each reaches QWK = 1.00 but has no priced deployment.

Figure 3: Cost per thousand gradings (log scale, from the Azure retail price API, meters in Appendix G) against QWK for every priced deployment. Filled markers pass all four gates; labelled points are discussed in Section 4.3.

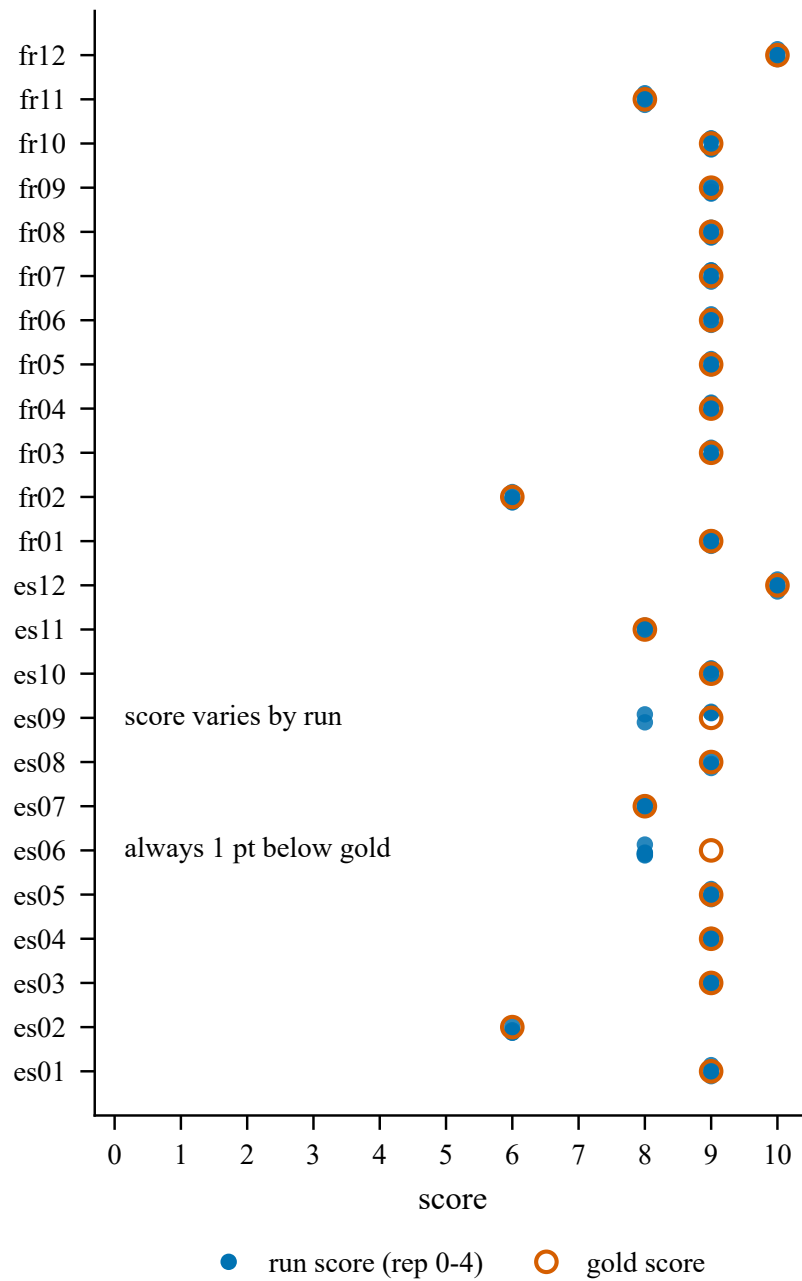


Figure 4: Per-item scores across five runs (filled points, jittered) against the gold score (hollow), extract-then-score on gpt-5.4 at temperature 0. One item (es09) varies within this run; one item (es06) scores one point below its original gold in every run, a corpus-convention defect rather than a grading error (Section 4.2).

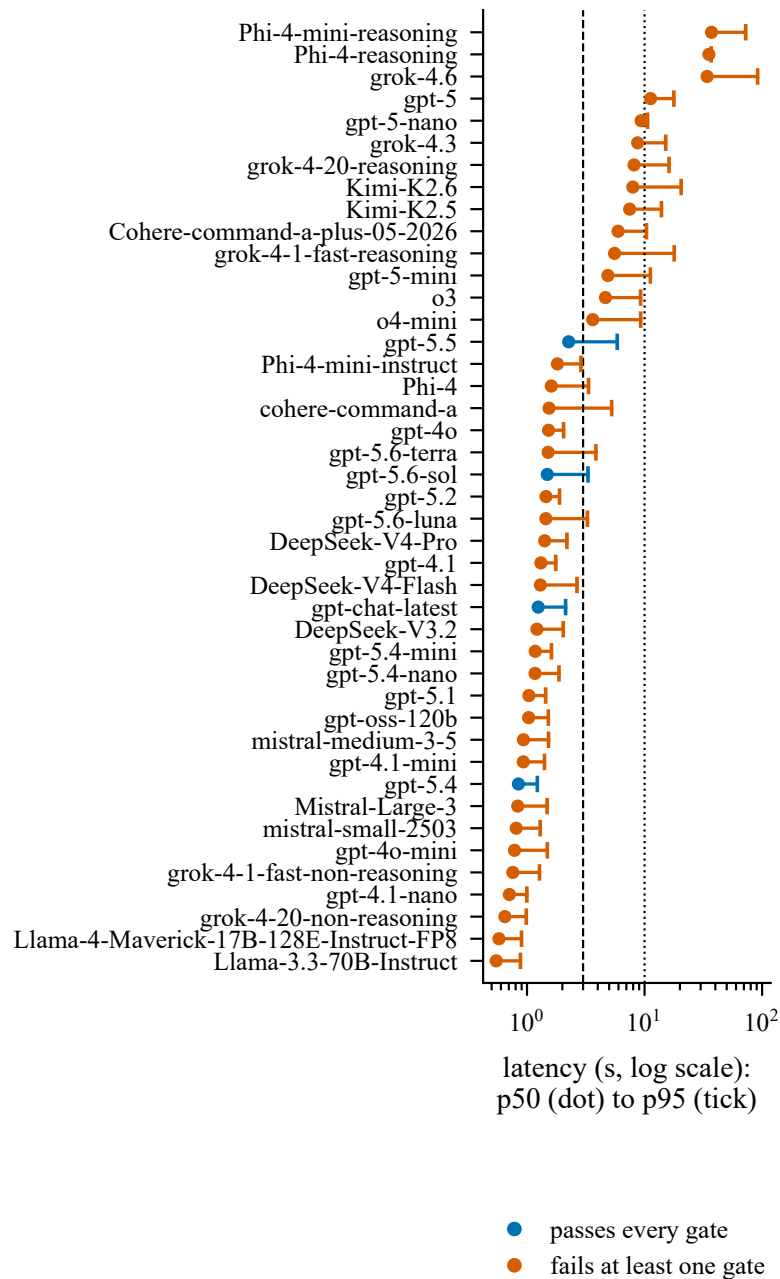


Figure 5: Median (point) and 95th-percentile (bar end) latency per deployment over successful calls, log scale, single client, 6 concurrent calls, including deployments not scored on agreement (Section 3.4). Reference lines at 3 s and 10 s.

D Failure classification

Table 5 classifies every call that did not return a scoreable response in the concurrent sweep, reclassified from the serial re-runs described in Section 4.4: rate limits under this account’s quota, the client’s HTTP-400 retry against Mistral’s HTTP-422 error shape (fixed before the reported run), and genuine reasoning-model failures to emit the requested JSON.

Table 5: Every model with at least one failed or unparseable grading call in the 45-model sweep (24 calls each). *at budget* is how many of those failures hit the sweep’s completion-token budget exactly ($ct = 1500$), the signature of a reasoning model exhausting its budget before emitting an answer rather than a genuine readability failure. Classification is quota, harness (parameter-shape), genuine (model limitation) or unknown, cross-checked against the serial retest where one exists.

model	n failed	at budget	classification	top error class
MAI-Thinking-1	24	0/24	harness	HTTPN: {"error":{"message":"Reques...
Ministral-3B	24	0/24	unmeasurable	HTTPN: {"error":{"code":"RateLimit...
Phi-4-mini-instruct	24	0/24	genuine	(call ok, output not valid JSON)
Phi-4-reasoning	24	24/24	genuine	(call ok, output not valid JSON)
Phi-4-mini-reasoning	23	7/23	genuine	(call ok, output not valid JSON)
gpt-5-nano	17	17/17	genuine	(call ok, output not valid JSON)
Kimi-K2.5	7	7/7	genuine	(call ok, output not valid JSON)
Kimi-K2.6	7	7/7	genuine	(call ok, output not valid JSON)
gpt-5	6	5/6	genuine	(call ok, output not valid JSON)
Llama-3.3-70B-Instruct	4	0/4	quota	HTTPN: {"error":{"code":"RateLimit...
Llama-4-Maverick-17B-128E-Instruct-FP8	4	0/4	quota	HTTPN: {"error":{"code":"RateLimit...
Phi-4	4	0/4	quota	(call ok, output not valid JSON)
gpt-4.1	4	0/4	genuine	(call ok, output not valid JSON)
DeepSeek-V3.2	3	0/3	quota	HTTPN: {"error":{"code":"RateLimit...
cohere-command-a	3	0/3	quota	(call ok, output not valid JSON)
gpt-4o	3	0/3	genuine	(call ok, output not valid JSON)
Cohere-command-a-plus-05-2026	2	2/2	genuine	(call ok, output not valid JSON)
DeepSeek-V4-Pro	2	0/2	quota	HTTPN: {"error":{"code":"RateLimit...
gpt-5-mini	1	1/1	genuine	(call ok, output not valid JSON)
grok-4-1-fast-non-reasoning	1	0/1	genuine	(call ok, output not valid JSON)

Table 6: Completion-budget re-run for the deployments with at least one unparsed response at the sweep’s 24-item, 1500-token budget: calls made at 8000 tokens (fewer than 24 where the futility rule stopped the run), parseability at each budget, QWK at the largest budget attempted (95% BCa bootstrap intervals over items, only once at least 12 rows parsed), the 95th-percentile completion length in tokens and the median latency at 8000. No model needed the pre-registered third budget of 16000 tokens: every deployment that parsed at all at 8000 parsed every item.

model	calls	par@1500 (%)	par@8000 (%)	QWK [CI]	p95 ct@8000	p50 (s)@8000
Cohere-command-a-plus-05-2026	24	91.7	100.0	0.824 [0.452, 0.953]	1239	5.10
Kimi-K2.5	24	70.8	100.0	0.953 [0.727, 1.000]	3341	13.84
Kimi-K2.6	24	70.8	100.0	0.929 [0.654, 1.000]	3610	6.24
Phi-4-mini-reasoning	24	4.2	0.0	–	3992	39.73
Phi-4-reasoning	8	0.0	0.0	–	8000	185.90
gpt-5	24	75.0	100.0	0.976 [0.654, 1.000]	2278	11.94
gpt-5-mini	24	95.8	100.0	0.762 [0.188, 0.972]	1201	6.29
gpt-5-nano	24	29.2	100.0	0.725 [0.427, 0.910]	3295	9.60

E Completion-budget re-run

Table 6 reports the re-run described in Section 3.4: the 8 deployments with at least one unparsed response at the 1500-token budget, called again on every item at 8000 tokens, serially with backoff, under a pre-registration that fixed the selection rule, the budgets, the cost cap and a futility rule before the first call. The pre-registration also provided a third budget of 16000 tokens for any deployment still unparsed at 8000; no deployment needed it, because every deployment that parsed at all at 8000 parsed every item, and the two that parsed nothing did so with completions far below the budget (Phi-4-mini-reasoning) or by exhausting the budget on every call at about three minutes per call (Phi-4-reasoning, stopped after eight calls under the futility rule). Latency and cost at 8000 are reported for completeness and are not comparable with the sweep, which ran 6 calls concurrently at a different time. The raw log is `results/raw/exp7_budget.jsonl`.

F Full sweep

Table 7 lists every axis for every deployment. Each deployment is compared with gpt-5.4 on mean absolute error over the items both parsed, with a paired sign-flip permutation test; 6 of 45 deployments are under-sampled and enter the correction imputed at $p = 1$ (conservative), so 7 of the 44 tests are nominally significant and 0 survive Benjamini–Hochberg at 0.05. The 44 p -values share one baseline and mostly the same items, so they are positively dependent; BH is valid under that condition (PRDS), and Benjamini–Yekutieli, the conservative alternative, computed directly rather than assumed from the BH result, also rejects 0.

Table 7: Full 45-model sweep, sorted by QWK descending (95% BCa bootstrap intervals over items; call and parsed are Wilson 95% intervals on the same proportion, whole percentage points). MAE is in Table 1 and Table 2 in the main text; it is dropped here to keep the table within the page width. q (BH) is the Benjamini–Hochberg-adjusted p-value for the paired MAE difference against gpt-5.4 (– for gpt-5.4 itself). Under-sampled models (fewer than 12 parsed rows out of 24) are listed at the bottom with –, since a QWK on that few points is not meaningful. †: zero disagreements on all items; the interval is degenerate and carries no uncertainty. QWK on band rank is in Table 9.

model	QWK [CI]	call (%) [95% CI]	parsed (%) [95% CI]	p50 (s)	p95 (s)	EUR/1k	q (BH)
gpt-5.6-sol	1.000 [1.000, 1.000]†	100 [86, 100]	100 [86, 100]	1.49	3.30	2.66	1.000
gpt-5	1.000 [1.000, 1.000]†	100 [86, 100]	75 [55, 88]	11.26	17.76	–	1.000
Kimi-K2.6	1.000 [1.000, 1.000]†	100 [86, 100]	71 [51, 85]	7.95	20.47	–	1.000
Kimi-K2.5	1.000 [1.000, 1.000]†	100 [86, 100]	71 [51, 85]	7.46	13.92	–	1.000
grok-4.6	0.977 [0.690, 1.000]	100 [86, 100]	100 [86, 100]	34.06	91.59	–	1.000
gpt-5.4	0.976 [0.700, 1.000]	100 [86, 100]	100 [86, 100]	0.84	1.22	0.60	–
gpt-chat-latest	0.956 [0.667, 1.000]	100 [86, 100]	100 [86, 100]	1.25	2.13	–	1.000
gpt-5.5	0.954 [0.723, 1.000]	100 [86, 100]	100 [86, 100]	2.26	5.86	4.81	1.000
grok-4-20-reasoning	0.954 [0.709, 1.000]	100 [86, 100]	100 [86, 100]	8.14	16.14	–	1.000
grok-4-1-fast-reasoning	0.952 [0.662, 1.000]	100 [86, 100]	100 [86, 100]	5.56	17.88	–	1.000
Llama-4-Maverick-17B-128E-Instruct-FP8	0.945 [0.000, 1.000]	88 [69, 96]	83 [64, 93]	0.58	0.90	0.05	1.000
DeepSeek-V3.2	0.944 [0.400, 1.000]	92 [74, 98]	88 [69, 96]	1.21	2.03	–	1.000
grok-4.3	0.933 [0.660, 1.000]	100 [86, 100]	100 [86, 100]	8.73	15.14	–	1.000
o4-mini	0.931 [0.667, 1.000]	100 [86, 100]	100 [86, 100]	3.62	9.27	1.94	0.920
DeepSeek-V4-Pro	0.902 [0.421, 1.000]	92 [74, 98]	92 [74, 98]	1.41	2.18	–	0.800
grok-4-20-non-reasoning	0.834 [0.585, 1.000]	100 [86, 100]	100 [86, 100]	0.65	0.99	–	0.920
o3	0.833 [0.571, 1.000]	100 [86, 100]	100 [86, 100]	4.65	9.26	2.91	0.920
gpt-4.1	0.809 [0.000, 1.000]	100 [86, 100]	83 [64, 93]	1.31	1.76	–	1.000
gpt-5-mini	0.779 [0.203, 0.986]	100 [86, 100]	96 [80, 99]	4.88	11.20	–	0.676
gpt-5.6-terra	0.771 [0.154, 0.984]	100 [86, 100]	100 [86, 100]	1.51	3.86	1.24	0.676
Cohere-command-a-plus-05-2026	0.763 [0.502, 0.970]	100 [86, 100]	92 [74, 98]	5.96	10.38	–	0.323
gpt-5.6-luna	0.724 [0.194, 1.000]	100 [86, 100]	100 [86, 100]	1.45	3.27	0.14	1.000
gpt-5.4-mini	0.677 [0.529, 1.000]	100 [86, 100]	100 [86, 100]	1.18	1.61	0.17	0.800
gpt-oss-120b	0.660 [0.540, 0.930]	100 [86, 100]	100 [86, 100]	1.03	1.52	–	0.676
gpt-5.2	0.605 [0.440, 0.883]	100 [86, 100]	100 [86, 100]	1.45	1.89	–	0.584
Mistral-Large-3	0.605 [0.424, 0.857]	100 [86, 100]	100 [86, 100]	0.83	1.48	–	0.584
mistral-medium-3-5	0.599 [0.398, 0.834]	100 [86, 100]	100 [86, 100]	0.93	1.52	–	0.323
gpt-4o	0.588 [0.417, 0.850]	100 [86, 100]	88 [69, 96]	1.52	2.05	0.48	0.800
gpt-4.1-mini	0.487 [0.043, 0.753]	100 [86, 100]	100 [86, 100]	0.93	1.41	–	0.525
DeepSeek-V4-Flash	0.483 [0.086, 0.802]	100 [86, 100]	100 [86, 100]	1.30	2.66	–	0.676
grok-4-1-fast-non-reasoning	0.445 [0.000, 0.545]	100 [86, 100]	96 [80, 99]	0.76	1.28	–	0.240
gpt-5.4-nano	0.381 [-0.065, 0.918]	100 [86, 100]	100 [86, 100]	1.17	1.87	0.06	0.240
gpt-5.1	0.358 [0.059, 0.667]	100 [86, 100]	100 [86, 100]	1.04	1.44	–	0.240
Phi-4	0.343 [0.000, 0.773]	96 [80, 99]	83 [64, 93]	1.61	3.34	–	0.800
Llama-3.3-70B-Instruct	0.326 [0.000, 0.622]	88 [69, 96]	83 [64, 93]	0.55	0.88	0.09	0.308
gpt-4o-mini	0.300 [0.053, 0.876]	100 [86, 100]	100 [86, 100]	0.78	1.49	0.02	0.676
gpt-4.1-nano	0.258 [0.011, 0.703]	100 [86, 100]	100 [86, 100]	0.71	1.00	–	0.240
mistral-small-2503	0.258 [-0.031, 0.550]	100 [86, 100]	100 [86, 100]	0.81	1.29	–	0.240
cohere-command-a	0.249 [-0.042, 0.540]	96 [80, 99]	88 [69, 96]	1.54	5.25	–	0.240
MAI-Thinking-1	–	–	–	–	–	–	–
Ministral-3B	–	–	–	–	–	–	–
Phi-4-mini-instruct	–	–	–	–	–	–	–
Phi-4-mini-reasoning	–	–	–	–	–	–	–
Phi-4-reasoning	–	–	–	–	–	–	–
gpt-5-nano	–	–	–	–	–	–	–

Table 8: Key sensitivity: gpt-5.4 extract/holistic QWK point estimates and the 45-model sweep’s $n(\geq 0.70)$ counts (point estimate and the 95% BCa-CI lower bound) and full-gate pass count, recomputed under each alternative scoring key against the same raw logs used throughout Experiments 2 and 6. *of which degenerate* is how many of the gate-passing models have a zero-disagreement (degenerate) QWK interval. †: zero disagreements on all items; the interval is degenerate and carries no uncertainty.

key variant	extract QWK	holistic QWK	$n(\geq 0.70 \text{ pt})$	$n(\geq 0.70 \text{ CI-lo})$	$n(\text{gates})$	of which degenerate
es09 gold _n =2 (score 8)	0.953	0.733	21	7	3	0
es06 excluded	1.000†	0.731	22	8	5	2
both	0.976	0.743	22	8	4	0

F.1 Robustness to the weighting scale

QWK’s quadratic weights run over the raw 0–10 score, so a one-band miss costs 1/100 near the ceiling (10 against 9) and 4/100 lower down (6 against 4), an asymmetry never stated in the main text. Table 9 recomputes QWK on band rank (10/9/8/6/4/2 mapped to ranks 0–5, quadratic weights on the rank) for every scored deployment. 28 of 45 deployments clear the 0.70 bar on rank QWK, against 22 on the raw score, and 4 clear all four gates on rank QWK, the same count and the same four deployments (gpt-5.4 0.959, gpt-5.5 0.923, gpt-5.6-sol 1.000, gpt-chat-latest 0.925) as on the raw score. The weighting scale changes individual point estimates but not the shortlist or which intervals are degenerate.

Table 9: QWK on the raw 0–10 score against QWK on band rank, for every scored deployment, sorted by rank QWK descending (95% BCa bootstrap intervals over items). †: zero disagreements on all items; the interval is degenerate and carries no uncertainty.

model	QWK [CI] (raw)	QWK [CI] (rank)
gpt-5.6-sol	1.000 [1.000, 1.000]†	1.000 [1.000, 1.000]†
gpt-5	1.000 [1.000, 1.000]†	1.000 [1.000, 1.000]†
Kimi-K2.6	1.000 [1.000, 1.000]†	1.000 [1.000, 1.000]†
Kimi-K2.5	1.000 [1.000, 1.000]†	1.000 [1.000, 1.000]†
grok-4.6	0.977 [0.690, 1.000]	0.961 [0.667, 1.000]
gpt-5.4	0.976 [0.700, 1.000]	0.959 [0.652, 1.000]
gpt-chat-latest	0.956 [0.667, 1.000]	0.925 [0.610, 1.000]
gpt-5.5	0.954 [0.723, 1.000]	0.923 [0.666, 1.000]
grok-4-20-reasoning	0.954 [0.709, 1.000]	0.923 [0.657, 1.000]
grok-4-1-fast-reasoning	0.952 [0.662, 1.000]	0.916 [0.599, 1.000]
DeepSeek-V3.2	0.944 [0.400, 1.000]	0.892 [0.351, 1.000]
Llama-4-Maverick-17B-128E-Instruct-FP8	0.945 [0.000, 1.000]	0.892 [0.000, 1.000]
grok-4.3	0.933 [0.660, 1.000]	0.885 [0.609, 1.000]
o4-mini	0.931 [0.667, 1.000]	0.882 [0.592, 1.000]
o3	0.833 [0.571, 1.000]	0.867 [0.643, 1.000]
grok-4-20-non-reasoning	0.834 [0.585, 1.000]	0.865 [0.656, 1.000]
gpt-4.1	0.809 [0.000, 1.000]	0.840 [0.000, 1.000]
gpt-5.4-mini	0.677 [0.529, 1.000]	0.835 [0.643, 1.000]
DeepSeek-V4-Pro	0.902 [0.421, 1.000]	0.823 [0.377, 1.000]
gpt-oss-120b	0.660 [0.540, 0.930]	0.796 [0.603, 0.938]
gpt-5-mini	0.779 [0.203, 0.986]	0.786 [0.343, 0.972]
gpt-5.6-luna	0.724 [0.194, 1.000]	0.774 [0.261, 1.000]
gpt-5.6-terra	0.771 [0.154, 0.984]	0.771 [0.270, 0.971]
gpt-4o	0.588 [0.417, 0.850]	0.769 [0.462, 0.901]
Cohere-command-a-plus-05-2026	0.763 [0.502, 0.970]	0.758 [0.519, 0.939]
gpt-5.2	0.605 [0.440, 0.883]	0.725 [0.455, 0.889]
Mistral-Large-3	0.605 [0.424, 0.857]	0.725 [0.400, 0.889]
mistral-medium-3-5	0.599 [0.398, 0.834]	0.715 [0.455, 0.896]
gpt-4.1-mini	0.487 [0.043, 0.753]	0.672 [0.182, 0.900]
DeepSeek-V4-Flash	0.483 [0.086, 0.802]	0.632 [0.205, 0.889]
gpt-5.1	0.358 [0.059, 0.667]	0.530 [0.180, 0.813]
grok-4-1-fast-non-reasoning	0.445 [0.000, 0.545]	0.498 [0.000, 0.718]
Phi-4	0.343 [0.000, 0.773]	0.496 [0.000, 0.872]
gpt-4o-mini	0.300 [0.053, 0.876]	0.457 [0.125, 0.905]
Llama-3.3-70B-Instruct	0.326 [0.000, 0.622]	0.394 [0.035, 0.757]
gpt-4.1-nano	0.258 [0.011, 0.703]	0.381 [0.083, 0.761]
gpt-5.4-nano	0.381 [-0.065, 0.918]	0.336 [-0.049, 0.838]
mistral-small-2503	0.258 [-0.031, 0.550]	0.322 [-0.024, 0.668]
cohere-command-a	0.249 [-0.042, 0.540]	0.298 [-0.039, 0.653]

G Cost derivation

Prices are EUR per million tokens for a global standard deployment, read from the Azure retail price API on 9 September 2026 and matched to a named meter by an explicit table rather than fuzzy matching, because meter names are not uniform across model families. Deployments whose meter could not be matched with confidence are reported as unpriced rather than estimated. Table 10 gives the meter for every deployment; cost per thousand gradings in the main tables is mean prompt tokens times the input price plus mean completion tokens times the output price, times one thousand.

Table 10: Price meters matched to each deployment (EUR per million tokens, global standard deployment, Azure retail price API snapshot of 9 September 2026). Unpriced deployments state why.

deployment	input	output	matched meter
Cohere-command-a-plus-05-2026	–	–	<i>no meter</i>
DeepSeek-V3.2	–	–	<i>no DeepSeek meter in the fetched regions</i>
DeepSeek-V4-Flash	–	–	<i>no DeepSeek meter</i>
DeepSeek-V4-Pro	–	–	<i>no DeepSeek meter</i>
Kimi-K2.5	–	–	<i>no Moonshot meter</i>
Kimi-K2.6	–	–	<i>no Moonshot meter</i>
Llama-3.3-70B-Instruct	0.60	0.60	Llama 3.3 70B Inp glbl Tokens Llama 3.3 70B Outp glbl Tokens
Llama-4-Maverick-17B-128E-Instruct-FP8	0.20	0.90	Llama 4 Maverick 17B Inp glbl Tokens Llama 4 Maverick 17B Outp glbl Tokens
MAI-Thinking-1	–	–	<i>no meter</i>
Minstral-3B	–	–	<i>no meter</i>
Mistral-Large-3	–	–	<i>no meter</i>
Phi-4	–	–	<i>Phi-4 meters ambiguous across variants</i>
Phi-4-mini-instruct	–	–	<i>ambiguous</i>
Phi-4-mini-reasoning	–	–	<i>ambiguous</i>
Phi-4-reasoning	–	–	<i>ambiguous</i>
cohere-command-a	–	–	<i>no meter</i>
gpt-4.1	–	–	<i>meter substring not found: gpt 4.1 input global</i>
gpt-4.1-mini	–	–	<i>meter substring not found: gpt 4.1 mini input global</i>
gpt-4.1-nano	–	–	<i>meter substring not found: gpt 4.1 nano input global</i>
gpt-4o	2.10	8.60	gpt-4o-0806-Inp-glbl Tokens gpt-4o-0806-Outp-glbl Tokens
gpt-4o-mini	0.10	0.50	gpt-4o-mini-0718-Inp-glbl Tokens gpt-4o-mini-0718-Outp-glbl Tokens
gpt-5	–	–	<i>only 'pp' meters found</i>
gpt-5-mini	–	–	<i>only 'pp' meters found</i>
gpt-5-nano	–	–	<i>no meter found</i>
gpt-5.1	–	–	<i>only 'pp' meters found</i>
gpt-5.2	–	–	<i>only 'pp' meters found</i>
gpt-5.4	2.15	12.88	5.4 inp Gl 1M Tokens 5.4 opt Gl 1M Tokens
gpt-5.4-mini	0.64	3.86	5.4 mini Inp Gl 1M Tokens 5.4 mini Opt Gl 1M Tokens
gpt-5.4-nano	0.17	1.07	5.4 nano Inp Gl 1M Tokens 5.4 nano Opt Gl 1M Tokens
gpt-5.5	4.29	25.76	5.5 ShortCo inp Gl 1M Tokens 5.5 ShortCo opt Gl 1M Tokens
gpt-5.6-luna	0.17	1.03	5.6 luna ShortCo Inp Std Gl 1M Tokens 5.6 luna ShortCo Opt Std Gl 1M Tokens
gpt-5.6-sol	4.29	25.76	5.6 sol ShortCo Inp Std Gl 1M Tokens 5.6 sol ShortCo Opt Std Gl 1M Tokens
gpt-5.6-terra	1.72	10.30	5.6 terra ShortCo Inp Std Gl 1M Tokens 5.6 terra ShortCo Opt Std Gl 1M Tokens
gpt-chat-latest	–	–	<i>no dedicated meter</i>
gpt-oss-120b	–	–	<i>no OSS 120b meter in the fetched regions</i>
grok-4.1-fast-non-reasoning	–	–	<i>Grok4 Fast meters exist but variants not distinguished</i>
grok-4.1-fast-reasoning	–	–	<i>Grok4 Fast meters exist but reasoning/non-reasoning not distinguished</i>
grok-4.20-non-reasoning	–	–	<i>no Grok 4.20 meter</i>
grok-4.20-reasoning	–	–	<i>no Grok 4.20 meter</i>
grok-4.3	–	–	<i>no Grok 4.3 meter published</i>
grok-4.6	–	–	<i>no Grok 4.6 meter published</i>
mistral-medium-3-5	–	–	<i>no meter</i>
mistral-small-2503	–	–	<i>no meter</i>
o3	1.70	6.90	o3 0416 Inp glbl Tokens o3 0416 Outp glbl Tokens
o4-mini	0.90	3.80	o4-mini 0416 Inp glbl Tokens o4-mini 0416 Outp glbl Tokens

H Repeatability, per item

Table 11 gives the per-item score across the five repeatability runs described in Section 4.2, so es06's one-point deficit and es09's day-to-day drift can be checked item by item against the summary in the main text.

Table 11: gpt-5.4 / extract, 5 repeated runs per item, 24 items. *spread* is max—min across the five runs; a * marks an item whose five runs disagree with each other.

item	gold	scores (rep 0–4)	spread
es01	9	9, 9, 9, 9, 9	0
es02	6	6, 6, 6, 6, 6	0
es03	9	9, 9, 9, 9, 9	0
es04	9	9, 9, 9, 9, 9	0
es05	9	9, 9, 9, 9, 9	0
es06	9	8, 8, 8, 8, 8	0
es07	8	8, 8, 8, 8, 8	0
es08	9	9, 9, 9, 9, 9	0
es09	9	9, 9, 9, 8, 8	1*
es10	9	9, 9, 9, 9, 9	0
es11	8	8, 8, 8, 8, 8	0
es12	10	10, 10, 10, 10, 10	0
fr01	9	9, 9, 9, 9, 9	0
fr02	6	6, 6, 6, 6, 6	0
fr03	9	9, 9, 9, 9, 9	0
fr04	9	9, 9, 9, 9, 9	0
fr05	9	9, 9, 9, 9, 9	0
fr06	9	9, 9, 9, 9, 9	0
fr07	9	9, 9, 9, 9, 9	0
fr08	9	9, 9, 9, 9, 9	0
fr09	9	9, 9, 9, 9, 9	0
fr10	9	9, 9, 9, 9, 9	0
fr11	8	8, 8, 8, 8, 8	0
fr12	10	10, 10, 10, 10, 10	0

I Literature search log

The related-work section and every absence claim in this paper rest on searches run on 9 September 2026 against OpenAlex, Crossref, arXiv abstract pages, the ACL Anthology and general web search; the Semantic Scholar API was rate-limited on that date and was not queried. The full log (database, query, hit count, screened, included) is in the packet (papers/sa-llm-grading/Literature.md). The absence claims are: no published study evaluates a language model as a rubric-referenced grader of *spoken* Spanish learner production (OpenAlex full-text search, 1,818 hits screened at the top page; three targeted web searches over arXiv and the ACL Anthology; the nearest counter-example found is Miller De Rutté and Correa (2026), which grades A1 Spanish learner *writing* with a task-customised GPT against a provided rubric, outside the spoken scope of this claim); no study tests an extract-errors-then-count architecture against holistic scoring on the same learner-language data; no ERRANT-style error-typing framework exists for Spanish or French (the languages of Qiu et al. (2025) and of MultiGEC-2025 (Masciolini et al., 2025) were checked directly; OpenAlex returned 144 hits for the query *grammatical error correction Spanish French ERRANT*, none a port); and no evaluation of graders for learner language gates on parseability, reliability, latency and cost together (four targeted searches). Each is a statement about what these searches found, not a proof of nonexistence.

Two further searches informed sections outside the main absence claims. For the Flemish grading-practice question (Section 3.1), the queries "*maximum twee fouten*", "*foutentelling*" *spreekvaardigheid evalueren Vlaanderen* and "*aantal fouten*" *beoordelingsschaal* returned no official or semi-official Flemish source that tallies grammar errors for spoken tasks; taalbeleid.be, the Katholiek Onderwijs Vlaanderen Frans classroom and leerplan pages, and the archived Toolkit breed evalueren (Vlaamse Overheid, 2021) all describe grading through rubric criteria and CEFR-anchored descriptor levels instead (Centrum voor Taal en Onderwijs, 2026; Katholiek Onderwijs Vlaanderen, 2026). For the operational-tradition paragraph (Section 2), the queries that located the e-rater and Speak & Improve precedents were run against OpenAlex and ETS's own publications pages; the same pass could not confirm a DOI for Attali and Burstein (2006) independent of a mismatched JTLA record, so that entry is cited as a grey source with two independently read mirrors instead.