

Can LLMs Judge Legal Accuracy? Reliability of LLM Evaluators for High-Stakes Insurance QA in a Low-Resource Language

David Beauchemin

david.beauchemin@ift.ulaval.ca

Université Laval

Richard Khoury

Université Laval

Research Article

Keywords: LLM-as-a-Judge, Automated evaluation, Insurance question answering, Legal NLP, Low-resource languages, Retrieval-augmented generation

Posted Date: September 7th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10304807/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Can LLMs Judge Legal Accuracy? Reliability of LLM Evaluators for High-Stakes Insurance QA in a Low-Resource Language

David Beauchemin^{1*} and Richard Khoury¹

¹Department of Computer Science and Software Engineering, Université Laval, Québec, QC, Canada.

*Corresponding author(s). E-mail(s): david.beauchemin@ift.ulaval.ca;
Contributing authors: richard.khoury@ift.ulaval.ca;

Abstract

Large language models (LLMs) are increasingly used as automated evaluators, a setting known as LLM-as-a-Judge, but their reliability outside general-domain English tasks is still poorly understood. We test this in a demanding case: judging the factual correctness of answers about Quebec automobile insurance, a specialized legal domain expressed in Quebec French. Evaluating 52 open-source and commercial judges against an expert-authored benchmark, we find that even the strongest is only moderately reliable and that different judges frequently disagree. Crucially, a judge’s overall quality does not predict how often it accepts a wrong answer as correct, the error that matters most in this setting; enabling step-by-step reasoning helps, and, contrary to the AI-favoritism reported in general domains, the judges are, if anything, stricter on AI-generated answers than on human ones. Selecting judges by their ability to catch errors, supplying an authoritative reference, and combining a few strong judges each reduces these dangerous acceptances. We conclude that a single LLM judge is not trustworthy enough to evaluate high-stakes legal answers on its own, and argue for cost-aware, human-in-the-loop use in deployment.

Keywords: LLM-as-a-Judge, Automated evaluation, Insurance question answering, Legal NLP, Low-resource languages, Retrieval-augmented generation

1 Introduction

Evaluating natural language generation (NLG) is a long-standing bottleneck: human judgement remains the gold standard, yet it is slow, costly, and difficult to reproduce at the scale required by modern model development [44]. To relieve this bottleneck, large language models (LLMs) are increasingly used as automated evaluators, a paradigm known as LLM-as-a-Judge [73], in which a capable model scores or compares candidate outputs in place of a human annotator. The approach has spread quickly because it is fast, inexpensive, and consistent across runs. It is now used in applications ranging from NLG evaluation, model leaderboards, and the preference-data pipelines used to align models, to high-stakes settings where an automated evaluator must vouch for user-facing answers whose errors carry real-world consequences. These uses differ in what they can tolerate: an offline leaderboard absorbs noisy verdicts that average out over many items, whereas an evaluator gating answers about a person’s legal rights cannot. Its appeal is reinforced by a growing literature reporting moderate-to-high correlation between LLM judges and human annotators on general-domain English benchmarks [32].

Three open questions, however, limit the applicability of these findings to applied settings. First, LLM judges exhibit well-documented biases: position bias [68], verbosity bias [70], and self-enhancement bias, where models rate their own generations more favourably [53, 57]. Second, most evaluations focus on general-domain tasks in English; whether LLM judges can reliably assess factual correctness in specialized domains (e.g., law [2], health [66], finance [47]) remains less well understood [4]. Third, evaluation quality appears to degrade for non-English content [34], raising questions about applicability beyond English.

These questions have practical relevance. In a 2024 ruling, the Civil Resolution Tribunal of British Columbia held Air Canada liable for erroneous information provided by its customer-facing chatbot regarding its bereavement fare policy [15]. More recently, in 2026, the Regional Court of Munich held Google directly liable for false statements produced by its Gemini-powered AI Overviews, ruling that such AI-generated answers are Google’s own content rather than neutral links to third-party sources [42]. These cases illustrate that factual errors in AI-generated answers about regulated products can have legal and financial consequences. In the insurance domain, where contracts are governed by jurisdiction-specific legislation, even a narrow factual error, such as confusing the public no-fault and private liability regimes, can misinform policyholders [2, 5]. Recent work on risk-weighted evaluation of AI errors [12] further shows that, at identical accuracy, differences in error-severity profiles can produce order-of-magnitude differences in expected cost, underscoring the importance of characterizing not only whether judges err but also how they err.

This paper studies LLM judges in a setting that combines all three challenges at once: a specialized, high-stakes professional domain (insurance), a non-English language (Quebec French), and a subtle factual-accuracy task in which the documented biases of LLM judges are most consequential. Concretely, we evaluate legal accuracy for French-language automobile insurance question answering in Quebec. A distinctive dual-regime system governs automobile insurance in the Province of Quebec. Bodily injury is covered by a public no-fault plan administered by the *Société de l’assurance*

automobile du Québec (SAAQ) [52], while property damage falls under private liability insurance governed by the Civil Code of Quebec [31] and standard policy forms (F.P.Q. No. 1) [3]. This structure creates precise terminological distinctions (e.g., the legally defined *automobile*, “car”, differs from the broader *véhicule*, “vehicle”) where superficially similar terms carry different legal implications. An LLM judge that conflates these regimes may validate an answer that the expert-authored key marks as incorrect. Our challenge is therefore one of reliability against this expert ground truth, namely whether a judge can dependably separate a correct answer from a subtly incorrect one in this high-stakes setting. It is not a question of human parity: we do not test whether LLM judges reproduce the agreement or error rates of human evaluators on the same task, a scope choice we return to in the [Limitations](#). The contributions of this work are:

1. A systematic reliability study of LLM-as-a-Judge for high-stakes legal question answering in a specialized domain and a non-English language (Quebec French), evaluating 52 judge configurations against a non-public expert benchmark.
2. A severity-aware analysis centred on the false-positive rate (FPR), which we tie to the severity-weighted cost of errors [12] and show is not significantly correlated with a judge’s aggregate quality, meaning the highest-ranked judge is not the safest to deploy.
3. A set of practical strategies for reducing risk and improving judge performance in high-stakes specialized-domain QA, with Quebec automobile insurance as our test case.

2 Related Work

2.1 LLM-as-a-Judge

Zheng et al [73] introduced the LLM-as-a-Judge paradigm with MT-Bench and Chatbot Arena, showing that a strong LLM can approximate human preferences on open-ended generation, and the approach was quickly adopted across NLG evaluation settings [17, 32, 48]. Because the paradigm’s credibility rests on agreement with human annotators, a meta-evaluation literature has grown to test that premise: Koo et al [40] formalize human agreement as the validity criterion, dedicated benchmarks such as JudgeBench [64] and broad surveys [46] catalogue judge models and their failure modes, and Guerdan et al [33] caution that judge-human agreement is itself difficult to interpret under inherent rating indeterminacy. This caveat bears directly on the only-slight agreement we report.

Systematic Biases.

A recurring theme of this scrutiny is that LLM judges are not neutral measuring instruments: their verdicts shift with cues unrelated to answer quality. Judges favour answers presented first in pairwise comparisons [68] and are swayed by surface features such as length and formatting [70]; they also show self-preference, scoring their own generations above equivalent human writing [53, 57]. Ye et al [71] quantify several such biases across many judge models, showing they are systematic rather than incidental.

Beyond General-Domain English.

This evidence, however, comes overwhelmingly from general-domain English settings, and the harder cases already hint at trouble. Bavaresco et al [4] find, across 20 NLP tasks, that judge-human agreement drops on factual-verification tasks relative to other categories, and Huang et al [37] document that fine-tuned judges do not substitute for stronger general models on legal evaluation, suggesting that general model capability, rather than domain-specific fine-tuning, is what such evaluation demands. Yet, to our knowledge, no study brings together these three different challenges: (1) reliability in a specialized, high-stakes domain, (2) in a non-English language, (3) with reasoning-enabled and standard judges compared side by side on the same task, a comparison existing work rarely makes since it seldom separates the effect of reasoning from the underlying model. The present study targets this gap across 52 judge configurations.

2.2 Automated Evaluation in Specialized Domains

Standard automatic metrics for NLG evaluation, such as BLEU [54], ROUGE, and BERTScore [72], have known limitations. Reiter [58] reviews the validity of BLEU and finds a weak correlation with human judgment across several generation tasks, and Tay et al [65] report the same for ROUGE in opinion summarization. The weakness extends to embedding-based metrics on specialized tasks: MeaningBERT [9] and JudgeBERT [10], which assess meaning preservation in text simplification and in legal text, respectively, find that many token- and vector-based metrics correlate weakly with human judgment, motivating purpose-built learned metrics. These limitations matter most in specialized domains where factual correctness outweighs surface fluency: a legally incorrect but fluent answer can score high on n -gram overlap while containing substantive errors.

RAG Evaluation Frameworks.

Retrieval-augmented generation (RAG) [45] grounds model answers in retrieved documents, and RAG-specific evaluation frameworks have emerged to address some of these gaps. Indeed, RAGAS [27] proposes reference-free metrics for faithfulness and answer relevance, while ARES [59] uses LLM judges with prediction-powered inference to estimate RAG quality with statistical guarantees. However, both frameworks [27, 59] were validated on general-domain English benchmarks and assume that LLM judges provide sufficiently reliable base signals, an assumption our work directly tests.

Evidence in Insurance QA.

Insurance question answering (QA) has a long NLP lineage. It began with retrieval and answer-selection models over the InsuranceQA benchmark [28, 63] and has since moved to large-language-model benchmarks of domain knowledge and answer generation [22]. These benchmarks measure whether a system *produces* a good answer; neither checks whether an automatic *evaluator* can reliably tell a correct insurance answer from a subtly incorrect one. That gap is sharpest for free-text answers: on Quebec automobile insurance, BLEU and ROUGE correlate weakly with expert human

judgment of RAG outputs [6], so surface-overlap metrics cannot serve as the evaluator. LLM-as-a-Judge is the natural candidate to fill this role, but its reliability has not been tested empirically in this domain, which is the gap our study addresses.

Legal NLP Benchmarks.

Legal NLP has produced a growing body of benchmarks and evaluation methodologies, ranging from broad surveys of the field [21] to task-specific resources such as contract-understanding datasets [36] and, more recently, RAG evaluation benchmarks for legal texts [55]. These efforts, however, focus overwhelmingly on English-language common-law jurisdictions. The intersection of legal NLP with non-English languages and civil-law traditions, as in Quebec [31], has received less attention. However, related work on French legal NLP exists, for example, case outcome prediction for the French and European courts [51, 62].

Positioning of This Work.

Closest to our work, a small but emerging line of research studies LLM-as-a-Judge reliability *within* the legal domain: Enguehard et al [26] (i.e., Legal LLM-as-a-Judge (LeMAJ)) bridge legal reasoning and LLM evaluation, Aftahee et al [1] compare LLM judges against legal experts in a non-English (Bengali) setting, and Karp et al [39] report that an LLM judge is unreliable when scoring answers to a Polish legal-qualification examination. Our study differs from both along three axes: the high-stakes Quebec automobile-insurance domain with its dual-regime civil-law terminology; a risk-weighted *false-positive* metric grounded in the differing costs of errors rather than agreement alone; and the scale and design of the judge roster (52 configurations, open and proprietary, reasoning and standard).

It also relates to a prior line of work on Quebec insurance question answering: where prior systems generate and compare answers [6, 8], the present paper turns the lens on the *evaluators* themselves, asking whether LLM judges are reliable enough to score such answers at all.

2.3 NLP for Low-Resource Languages and French

Resource availability in NLP is highly uneven across languages and varieties [38]. While French as a whole is well resourced, Quebec French in a specialized legal domain is not: it pairs a regional variety with specialized legal terminology that general multilingual models rarely encounter, which makes it a demanding setting for LLM evaluation. Quebec operates under a civil-law system (the Civil Code of Quebec [31], derived from the French civil-law tradition), with jurisdiction-specific terminology, legislation, and regulatory frameworks that differ from those of both France and other Canadian provinces.

Regional Variation.

Prior work reports performance disparities between LLMs on Quebec French versus Metropolitan French linguistic acceptability tasks, suggesting that regional variation affects model performance within the same language [7, 11, 13]. The combination

of specialized legal terminology, jurisdiction-specific regulations, and regional French thus creates an evaluation setting that differs from the general-domain English benchmarks used in most LLM-as-a-Judge research.

2.4 Risk-Aware AI Evaluation

Standard evaluation metrics such as accuracy and F1 treat all errors as equivalent, which is inadequate when different misclassifications carry different consequences. Two long-standing responses address this: cost-sensitive learning weights errors by their cost [25], and selective prediction, or the reject option, abstains on uncertain inputs to control risk at a chosen coverage level [30]. Building on this risk-aware view, Beauchemin et al [12] argue that AI evaluation should weight errors by their cost, decomposing operational risk into the rate at which a system errs and the cost incurred per error. Empirically, they show that two systems with identical 95% accuracy can differ by more than an order of magnitude in expected error cost depending on their severity profiles, and that a 99%-accurate system can be costlier than a 95%-accurate one when its errors concentrate in high-severity categories. Routing high-severity outputs to human reviewers, a domain-specific instance of the reject option [30], reduces expected cost by 86–98%.

Relation to This Work.

Our work complements this perspective: where Beauchemin et al [12] assume the availability of a reliable error classifier, we empirically evaluate how well LLM judges perform in that role. The false-positive rates we measure, where judges validate incorrect answers, correspond directly to the severity-bearing errors that drive tail risk in their cost model.

3 Resources and Data

3.1 AEPC-QA Benchmark

Our evaluation uses the Automobile and Expert in Property and Casualty Insurance Question Answering (AEPC-QA) benchmark [reference withheld for double-blind review],¹ a corpus of expert-authored multiple-choice questions drawn from printed professional examination preparation handbooks published by the *Autorité des marchés financiers* (AMF), Quebec’s financial and insurance regulatory body. The handbooks are sold primarily in print, with no widely-available digital editions. Table 1 reports the size of both corpora used in this study. AEPC-QA comprises 807 four-choice questions drawn from 19 source manuals, totalling roughly 223k whitespace-delimited tokens: a question stem averages 117 tokens, each answer choice 17 tokens, and each expert justification 92 tokens. The second corpus, the QAIERC-V2 reference corpus that feeds the RAG systems, is described in Section 3.2. Table 2

¹AEPC-QA is not publicly available due to intellectual property restrictions on the source examination materials. Access was granted under a research agreement with the corpus provider. Consequently, the originating reference is withheld for double-blind review, as citing it would reveal the authors’ identity. It will be restored in the camera-ready version.

Table 1 Statistics of the two corpora used in this study. AEPC-QA token counts are whitespace-delimited; QAIERC-V2 statistics are those reported by the source corpus [6].

Corpus	Statistic	Value
AEPC-QA	Questions	807
	Choices per question	4
	Source manuals	19
	Avg. tokens / question	117.20
	Avg. tokens / choice	16.81
	Avg. tokens / justification	92.21
	Total tokens	223,258
QAIERC-V2	Sentences	168,600
	Tokens	2.5M
	Document categories	7

details the five fields attached to every question and illustrates each with a representative item. Beyond the French question text, the four answer choices, and the answer key, every question carries an expert justification of why the keyed choice is correct and a reference to its source manual. The justification is what provides a judge with an authoritative rationale rather than the bare answer key. The illustrative item shown in Table 2 asks which regime compensates a pedestrian’s bodily injury after an automobile accident, with distractors that encode precisely the public-versus-private regime confusion that makes the domain error-prone.

Three properties of the AEPC-QA corpus are relevant for this evaluation:

1. The specific question and distractor formulations are highly unlikely to be present in LLM training data [60]: the corpus is not publicly available. It derives from preparation handbooks that exist mainly in print rather than digital editions. Hence, the items were very unlikely to be crawled from the web and ingested during pre-training. The underlying statutory content is, of course, public; it is the exact phrasing of the item, not domain knowledge, that resists contamination.
2. Domain experts write the questions and include plausible distractors with specific factual errors (e.g., stating that property damage is indemnified by the public no-fault plan, which in fact compensates only bodily injury).
3. The MCQ format provides binary ground truth, since each choice is either correct or incorrect, allowing measurement of judge accuracy without the subjectivity of rating scales.

3.2 Reference Corpus (QAIERC-V2)

For the RAG systems, we employ the Quebec Automobile Insurance Expertise Reference Corpus version 2 (QAIERC-V2) [6]. As summarized in Table 1, it gathers 168,600 sentences, an order of magnitude more text than the question set, organized into the seven document categories detailed in Table 3, which reports, for each category, the kind of material it holds and a concrete example.

This corpus serves as the retrieval knowledge base from which RAG systems generate their free-text answers. For example, a “laws” category passage encodes the no-fault

Table 2 Fields provided with each AEPC-QA question, with an illustrative item (synthetic and paraphrased to respect the corpus licence).

Field	Description	Example
Question	The exam question text, in French	Which regime compensates a pedestrian’s bodily injury after an automobile accident in Quebec? <i>Un piéton est blessé par une automobile au Québec; quel régime indemnise son préjudice corporel?</i>
Choices	Four answer choices (a, b, c, d), exactly one correct	(a) the public no-fault plan (SAAQ); (b) the driver’s civil-liability insurance; (c) the crime victims’ compensation fund; (d) no compensation <i>(a) le régime public sans égard à la responsabilité (SAAQ); (b) l’assurance de responsabilité civile du conducteur; (c) le Fonds d’indemnisation des victimes d’actes criminels; (d) aucune indemnisation</i>
Answer key	The ground-truth correct choice	(a)
Justification	Expert explanation of why the answer is correct	In Quebec, bodily injury from an automobile accident is compensated by the public no-fault SAAQ plan, regardless of fault. <i>Au Québec, le préjudice corporel résultant d’un accident d’automobile est indemnisé par le régime public sans égard à la responsabilité de la SAAQ.</i>
Source	Reference to the source preparation manual	AMF certification materials, automobile insurance tome <i>Documents de certification de l’AMF, tome assurance automobile</i>

principle of the Automobile Insurance Act, namely that bodily injury caused by an automobile is compensated by the public plan regardless of fault, while an educational-resource passage paraphrases the same rule for exam preparation; a RAG system retrieves such passages to ground its answer. The diversity of source types (primary legislation, secondary regulatory guidance, and tertiary educational material) mirrors the reference hierarchy that insurance professionals use in practice.

Table 3 The seven document categories of the QAIERC-V2 reference corpus, with a concrete example of the content of each.

Category	Content	Example
Laws	Quebec legislation: Civil Code of Québec (CQLR c. CCQ-1991), Automobile Insurance Act (CQLR c. A-25), Insurers Act (CQLR c. A-32.1), Code of Civil Procedure (CQLR c. C-25.01)	The Automobile Insurance Act provision that bodily injury caused by an automobile is compensated regardless of fault
Regulatory textbooks	AMF certification study materials for insurance agents, property and casualty brokers, and claims adjusters	A study-guide chapter explaining the coverages of the F.P.Q. No. 1 owners' form
Reports	Regulatory reports and compliance guides from Quebec's financial-sector regulators	An AMF report on the claims-settlement practices of property and casualty insurers
Articles	Academic and legal commentary on Quebec automobile insurance jurisprudence	A law-review note analyzing a Court of Appeal ruling on no-fault indemnities
Insurance contracts	Standard policy forms (F.P.Q. No. 1) and related contractual documents	The third-party liability clause (Section A) of the F.P.Q. No. 1 form
Regulatory documents	AMF directives, guidelines, and public notices	An AMF guideline on the sound commercial practices of insurers
Educational resources	Pedagogical materials on Quebec insurance law	An exam-preparation summary of the SAAQ public no-fault compensation rule

4 Methodology

4.1 Experimental Design

Establishing whether an LLM can be trusted to judge legal accuracy requires measuring two distinct things: whether a judge reliably separates a correct answer from a subtly incorrect one against an authoritative ground truth, and whether that behaviour carries over to the free-text answers a deployed system actually produces, which have no automatic ground truth. The two error directions are moreover far from symmetric in cost: validating a wrong answer lets it reach a policyholder, whereas rejecting a correct one merely routes it to a human re-check, so the evaluation must isolate the costly direction rather than average over both (we quantify this asymmetry in [Section 4.5](#) and [Section 5.1](#)). These requirements dictate our design. We employ a two-condition factorial design to evaluate LLM judges on both human-authored and AI-generated candidate answers.

Condition A: Human-Authored Text.

For each of the 807 AEPC-QA questions, we submit each of the four MCQ choices as an independent candidate answer to the LLM judge, yielding 3,228 evaluation samples per judge. Ground truth is determined directly from the answer key: one choice is correct, three are incorrect, producing a class imbalance of 25%/75% (correct/incorrect).

This condition measures the judge’s ability to detect factual errors in human-written, plausible distractors.

Condition B: AI-Generated Text.

We generate free-text answers to all 807 questions using each of the 20 open-source generation-capable models as a RAG generator, as detailed in [Section 4.3](#); the 6 judge-only critics and the API-served judges are excluded from generation. This yields $20 \times 807 = 16,140$ candidate answers, and every judge in the full roster then evaluates every generated answer, so each open generator both answers and is judged. Each RAG answer is submitted to the LLM judge along with the reference answer and expert justification from the AEPC-QA corpus, enabling comparison between the candidate and the known correct answer. This condition measures the judge’s ability to evaluate AI-generated free-text, a setting closer to practical RAG deployment. Since free-text correctness cannot be determined automatically, as a RAG answer may be partially correct, correct but differently worded, or subtly incorrect, Condition B analysis focuses on inter-judge agreement rather than accuracy against a ground-truth label.

Variance via 10-Fold Cross-Validation.

We assign each question to one of ten folds at the question level, so that all four MCQ choices of a question share a fold and no leakage occurs between a question’s correct answer and its distractors, and report every metric as the mean $\pm$ standard deviation across the ten held-out folds. Folds serve a dual purpose: they provide a proper train/test split for any tuned component of the judging framework, such as ensemble composition or decision threshold, and they yield the variance estimate. We deliberately use folds rather than repeated runs as the source of variance: with greedy decoding (`do_sample=False`), re-running a judge on the same input is nearly deterministic, so run-to-run variance is negligible rather than informative.

Scale.

Condition A comprises $52 \times 3,228 = 167,856$ evaluations, and Condition B generates $20 \times 807 = 16,140$ RAG answers, each judged by all 52 models, for $16,140 \times 52 = 839,280$ evaluations. The open-source subset is fully reproducible, while the API-served judges are queried through provider APIs. Serving hardware, configuration, and cost details are reported in [Appendix C](#).

4.2 LLM Judges

We evaluate 52 LLM-judge configurations, selected to span two complementary families: 26 open-source models served locally with vLLM [\[41\]](#) and fully reproducible from open weights, and 26 API-served judges spanning frontier proprietary and large open-weight models, queried through commercial APIs and the OpenRouter gateway. The roster was assembled to disentangle the factors that could plausibly drive judging reliability, rather than to maximize raw count. First, we cover the capability spectrum, from 0.5B parameters to 120B for the open models and up to frontier

proprietary systems, so that the effect of scale can be separated from that of specialization. Second, we include a fully open-source subset, both for reproducibility and to represent a judge a practitioner could realistically deploy locally. Third, we add models specialized for the task or the domain to test whether such specialization transfers to Quebec insurance law: the evaluation-tuned **Prometheus-2-7B** and **Skywork-Critic-8B**; six judge-only critics (**Selene-Mini-8B**, the three CompassJudge sizes, **Flow-Judge-3.8B**, and **GLIDER-3.8B**) that act as judges in both conditions but, not being trained for free-form generation, are excluded from the Condition B generation pool; and the legal-domain model **SauLLM-7B**, the closest domain match since, to our knowledge, no open insurance-specific LLM exists. Fourth, we span ten model vendors to avoid single-vendor idiosyncrasies, and restrict the roster to models available at the time of experimentation. A central design choice is that, wherever a model exposes a controllable reasoning mode, we evaluate it *both* with reasoning enabled, using the provider’s native thinking controls, and in standard mode (e.g. Claude Opus/Sonnet/Haiku 4.x, GPT-5 family, **Qwen3-235B**), so that the effects of inference-time reasoning on judging reliability can be measured within an otherwise identical model. [Table 4](#) and [Table 5](#) list the open-source and API-served rosters, respectively; serving hardware, quantization, constrained decoding, and the exact API model strings and query stack are reported in [Appendix C](#).

Reference Baselines.

To anchor every metric on an interpretable scale, we add four non-LLM baselines that issue verdicts without reading the candidate:

- **Oracle**: always returns the ground-truth label, marking the upper bound, i.e. a perfect Matthews correlation coefficient (MCC), our headline reliability metric, defined in [Section 4.5](#).
- **Always-accept**: labels every candidate “correct”, so it accepts every wrong answer, giving an FPR of 100%.
- **Always-reject**: labels every candidate “incorrect”, so it never makes a false positive but catches nothing genuine.
- **Random**: labels each candidate “correct” or “incorrect” by an unbiased coin flip.

Under Condition A’s 25%/75% correct/incorrect class balance, the three non-oracle policies all score an MCC near zero despite very different accuracies, so they pin down the trivial reference points and reveal which judges add genuine discriminative value over a constant rule. **Always-reject**, for instance, reaches 75% accuracy with zero discriminative power, exposing the conservative-rejecter behaviour that raw accuracy would reward.

Response Parsing.

Each judge’s output is parsed into a binary verdict, with a regex fallback when the structured field is missing. Outputs that cannot be resolved to “correct” or “incorrect” are labelled **UNPARSEABLE**; they are excluded from the accuracy metrics but reported separately as a per-judge unparseable rate, so that abstention is never silently scored as a correct or incorrect verdict.

Table 4 Open-source LLM judges (26), served locally with vLLM. “R” marks reasoning-capable models; the scales icon  marks judge-only critics excluded from the Condition B generation pool.

Judge	Provider	Size	R	Critic 
DeepSeek-R1-Distill-70B	DeepSeek/Meta	70B	✓	
Llama-3.3-70B	Meta	70B		
Qwen2.5-72B	Alibaba	72B		
gpt-oss-120b	OpenAI	120B	✓	
QwQ-32B	Alibaba	32B	✓	
Qwen3-32B	Alibaba	32B	✓	
Qwen3-30B-A3B	Alibaba	30B (3B act.)	✓	
Gemma-3-27B	Google	27B		
CompassJugder-32B	OpenCompass	32B		✓
gpt-oss-20b	OpenAI	20B	✓	
Phi-4	Microsoft	14B		
Selene-Mini-8B	Atla	8B		✓
Skywork-Critic-8B	Kunlun	8B		
CompassJugder-7B	OpenCompass	7B		✓
Prometheus-2-7B	KAIST	7B		
SaulLM-7B (legal)	Equall	7B		
Qwen2.5-7B	Alibaba	7B		
Flow-Judge-3.8B	Flow AI	3.8B		✓
GLIDER-3.8B	Patronus AI	3.8B		✓
Gemma-3-4B	Google	4B		
Qwen2.5-3B	Alibaba	3B		
Llama-3.2-3B	Meta	3B		
Qwen2.5-1.5B	Alibaba	1.5B		
CompassJugder-1.5B	OpenCompass	1.5B		✓
Llama-3.2-1B	Meta	1B		
Qwen2.5-0.5B	Alibaba	0.5B		

4.3 RAG Systems for Condition B

For Condition B, every open-source generation-capable model in Table 4 acts as a RAG generator, excluding the 6 judge-only critics: each of the 20 generators answers all 807 questions through the shared retrieval pipeline, and every judge in the 52-model roster then evaluates the answers produced by all generators. This design lets us measure, for every judge, whether it is more lenient toward AI-generated text than toward human MCQ distractors (*AI-favoritism*: a general tendency to favour machine-generated over human text, broadening the self-preference or self-enhancement bias documented for LLM judges [53, 73]), and whether models favour their own generations (*self-preference*), measured across the 20 models that both generate and judge. Restricting generation to open-source models keeps the generation step reproducible and free of closed-model contamination.

Retrieval.

Each RAG system follows the retrieve-then-generate architecture of the Quebec insurance QA pipeline of Beauchemin and Khoury [6]: a hybrid dense-sparse retriever

Table 5 API-served LLM judges (26), queried through native or OpenAI-compatible APIs. “R” marks reasoning-enabled configurations; “Batch” marks providers queried through an asynchronous Batch API. Models marked “via OpenRouter” are open-weight or proprietary models accessed through the OpenRouter gateway. Parameter counts for proprietary models are largely undisclosed.

Judge	Provider	R	Batch
Claude-Opus-4.8-reason	Anthropic	✓	✓
Claude-Opus-4.8	Anthropic		✓
Claude-Sonnet-4.6-reason	Anthropic	✓	✓
Claude-Sonnet-4.6	Anthropic		✓
Claude-Haiku-4.5-reason	Anthropic	✓	✓
Claude-Haiku-4.5	Anthropic		✓
GPT-5.1-reason	OpenAI	✓	✓
GPT-5-mini-reason	OpenAI	✓	✓
GPT-5-mini	OpenAI		✓
Magistral-Medium-reason	Mistral	✓	✓
Mistral-Medium	Mistral		✓
DeepSeek-Reasoner	DeepSeek	✓	
DeepSeek-Chat	DeepSeek		
Grok-4-reason	xAI	✓	
Grok-3-mini-reason	xAI	✓	
Gemini-3.1-Pro	Google (via OpenRouter)		
Gemini-2.5-Flash	Google (via OpenRouter)		
Qwen3-235B-thinking	Alibaba (via OpenRouter)	✓	
Qwen3-235B	Alibaba (via OpenRouter)		
GLM-4.6	Z-AI (via OpenRouter)		
Kimi-K2	Moonshot (via OpenRouter)		
Command-A	Cohere		
Command-R-Plus	Cohere		
Command-R	Cohere		
Command-R7B	Cohere		
Aya-Expansive-32B	Cohere		

selects, from a vector index of the QAIERC-V2 corpus, the passages most relevant to the question, and the top-ranked passages are concatenated into the generator’s context so that it answers from grounded legal text rather than parametric memory alone. Every generator queries the same index and receives the same passages, so differences between RAG answers reflect the generator rather than retrieval. The chunking, indexing, and retrieval parameters, together with the software stack, are detailed in [Appendix C](#).

Generation.

Each RAG system receives a domain-specific prompt ([Appendix B](#)) instructing it to answer in French, cite legal provisions, and avoid mentioning MCQ options. Each generator is served locally with vLLM, the same way as the open-source judges, and produces a free-text answer.

4.4 Judge Prompt Design

Our judge prompt design draws on empirical findings from the LLM-as-a-Judge literature. The full templates are provided in [Appendix A](#). The main design choices are:

1. **Binary verdict** (correct/incorrect): Koo et al [40] show that binary scales maximize inter-annotator agreement on factual verification tasks, avoiding the score-distribution biases observed with Likert scales.
2. **English meta-instructions, French content**: Our system prompts and evaluation instructions are in English, while the exam questions and candidate answers remain in their original French. We adopt English meta-instructions as a deliberate design choice, motivated by reports that LLM evaluation consistently degrades for non-English content [34].
3. **Chain-of-thought reasoning**: Following G-Eval [48], we require the judge to generate a reasoning explanation alongside its verdict, structured as a JSON object with `verdict` and `reasoning` fields.
4. **Domain-specific instructions**: The system prompt explicitly identifies the legal domain (Quebec automobile insurance), relevant legal frameworks (Civil Code of Quebec, AMF regulations), and warns against leniency on subtle legal inaccuracies.

Two prompt variants are used. **Without references** (Condition A), the judge receives only the question and the candidate’s answer. **With references** (Condition B, the default), it additionally receives the corpus reference answer and expert justification, enabling comparison between the RAG-generated answer and the known correct answer.

4.5 Evaluation Metrics

We evaluate LLM judges along four complementary axes and conduct seven supplementary analyses. All metrics are computed per judge within each cross-validation fold and aggregated as mean $\pm$ standard deviation across the 10 folds.

Accuracy Metrics (Condition A).

For each judge, we compute standard classification metrics treating “correct” as the positive class: accuracy, precision, recall, and F1. Since our primary concern is detecting incorrect answers, we additionally report error-detection F1: precision, recall, and F1 with “incorrect” as the positive class. Because the 25%/75% class imbalance makes raw accuracy misleading, since a constant “incorrect” verdict already scores 0.75, we adopt MCC [18, 49] and balanced accuracy (the mean of per-class recall) as the primary headline metrics. MCC is the correlation between a judge’s binary verdicts and the ground-truth labels, computed from all four cells of the confusion matrix; it ranges from -1 (systematic disagreement) through 0 (chance-level, no association) to $+1$ (perfect agreement). Because it is high only when a judge classifies *both* correct and incorrect answers well, it cannot be inflated by exploiting the majority class the way raw accuracy can [18]. Both headline metrics are therefore near zero for any constant or random rule and reward only genuine discrimination between correct and incorrect answers, exposing the conservative-rejecter behaviour that raw accuracy hides.

We also report the unparseable rate, defined as the fraction of all judge outputs that could not be parsed into a valid verdict.

False-Positive Rate.

The judge can err in two directions, and the two carry sharply asymmetric cost: a false positive lets a factually-incorrect legal answer be validated and reach a policyholder, where it may have negative consequences both for that policyholder by misleading them into a wrong course of action, and for the company by leaving them legally liable for the consequences of that bad advice. On the other hand, the opposite error, rejecting a correct answer, merely routes it to a human for review. A safety metric must therefore isolate the costly direction rather than average over both, which a symmetric score such as MCC cannot do. We accordingly report the false-positive rate:

$$\begin{aligned} \text{FPR} &= \text{P}(\text{judge says correct} \mid \text{answer is incorrect}) \\ &= \frac{|\{s : v_s = \text{correct} \wedge y_s = \text{false}\}|}{|\{s : y_s = \text{false}\}|} \end{aligned} \quad (1)$$

where v_s is the judge’s verdict for sample s and y_s is the ground-truth correctness label. This is the ordinary false-positive rate with “correct” as the positive class. The metric does not weight errors by severity, since all incorrect answers are treated equally; rather, it isolates the error direction that the cost model of Beauchemin et al [12] identifies as costly. A high FPR indicates that the judge frequently validates factually incorrect answers, which, in a deployed insurance QA system, could propagate incorrect information to policyholders. We stress that FPR is measured against the Condition A MCQ ground truth; whether this MCQ-measured behaviour transfers to the free-text answers a deployed system actually produces, which have no gold standard (Condition B), is an open question we do not resolve here (see [Limitations](#)). Given the 25%/75% class balance in Condition A, a naive “always correct” classifier would achieve 25% accuracy but 100% FPR. In that same cost model, FPR corresponds to the false-negative rate of the error classifier: missed errors that pass through to end users and contribute to severity-weighted aggregate cost.

Inter-Judge Agreement.

We measure pairwise agreement using Cohen’s κ [20] and overall agreement using Fleiss’ κ [29]. We additionally compute the phi coefficient between judge pairs on their binary error vectors, which captures whether judges tend to fail on the same or different questions. We interpret κ values on the Landis & Koch scale [43]: < 0.20 (slight), $0.21\text{--}0.40$ (fair), $0.41\text{--}0.60$ (moderate), $0.61\text{--}0.80$ (substantial), > 0.80 (almost perfect).

Statistical Tests.

We employ McNemar’s exact test [50] to assess the pairwise significance of differences among judges, with a conservative Bonferroni correction [23] across the $\binom{52}{2} = 1,326$ pairwise comparisons. Cochran’s Q test [19] assesses overall judge

equivalence across all 52 judges simultaneously. Bootstrap confidence intervals [24] ($n = 1,000$ resamples, seed = 42, percentile method) quantify metric uncertainty for accuracy, error-detection F1, MCC, and the false-positive rate; we verify these intervals are seed-stable by recomputing them under the ten-fold seed schedule (42–51) and confirming the bounds barely move (Appendix C).

Supplementary Analyses.

We conduct seven additional analyses to characterize judge behaviour beyond aggregate metrics:

1. **Choice label bias:** whether judges exhibit differential accuracy depending on the MCQ choice position (a, b, c, or d), measured as the rate of “correct” verdicts stratified by candidate label and ground truth;
2. **Answer length bias:** Pearson correlation between candidate answer length and judge verdict, computed separately for actually-correct and actually-incorrect samples;
3. **Calibration proxy:** point-biserial correlation between reasoning length and verdict correctness, testing whether judges produce different-length justifications when they are right or wrong;
4. **Question difficulty stratification:** judge accuracy stratified into 5 difficulty bins, where difficulty is defined as the proportion of judges that erred on a given sample;
5. **Error patterns:** identification of universal failures (samples where all judges err), universal successes, and divisive samples (top 10% by inter-judge variance);
6. **Latency analysis:** per-judge inference latency for the local judges, used as an efficiency signal alongside accuracy;
7. **Cross-fold stability:** the standard deviation of each judge’s accuracy and false-positive rate across the 10 folds, quantifying how stable each judge’s behaviour is over disjoint question subsets; with greedy decoding, run-to-run variance is negligible, so cross-fold spread is the meaningful stability signal.

4.6 Cross-Condition Analysis

To test for AI-favoritism, we compare judge behaviour between Condition A (human text) and Condition B (AI-generated text) along two dimensions.

Correct-Rate Comparison.

For each judge, we compute the rate at which it renders “correct” verdicts in each condition. We test for significant differences using a two-proportion z -test with a pooled proportion:

$$z = \frac{p_{\text{AI}} - p_{\text{human}}}{\sqrt{\hat{p}(1 - \hat{p}) \left(\frac{1}{n_{\text{human}}} + \frac{1}{n_{\text{AI}}} \right)}} \quad \text{where} \quad \hat{p} = \frac{p_{\text{human}}n_{\text{human}} + p_{\text{AI}}n_{\text{AI}}}{n_{\text{human}} + n_{\text{AI}}} \quad (2)$$

where p_{human} and p_{AI} are the judge’s rates of “correct” verdicts on the human MCQ choices (Condition A) and the AI-generated answers (Condition B), n_{human} and n_{AI} are the corresponding sample sizes, and $\hat{p}$ is the pooled rate; we use “human”/“AI”

rather than “A”/“B” subscripts to avoid confusion with the two conditions. We report the delta $\Delta = p_{AI} - p_{human}$ per judge (two-tailed, $\alpha = 0.05$). A positive Δ indicates that a judge renders “correct” more often on AI-generated text than on human MCQ choices. We stress that the two conditions differ in their underlying base rate of correct candidates (Condition A is fixed at 25% correct by the MCQ design, whereas RAG answers in Condition B are genuine attempts at the correct answer and are correct far more often), so Δ confounds this base-rate gap with any genuine leniency. We therefore read a large positive Δ as a behavioural tendency worth flagging, potentially consistent with the self-enhancement bias reported by Panickssery et al [53] and Raina et al [57], rather than as a bias-corrected effect size. The self-preference analysis enabled by the $N \times N$ design, comparing whether a model judges its own RAG answers more leniently than other models’ answers of comparable quality, holds answer provenance fixed within Condition B and thus provides a complementary, less confounded signal.

Verbosity Comparison.

We compare reasoning length, the character count of the `reasoning` field, across conditions using Welch’s t -test for unequal variances, following the observation by Wu and Aji [70] that judges may produce longer justifications for certain types of content.

4.7 Ensemble Strategies

Following Verga et al [67], who propose using panels of diverse models rather than single judges, we evaluate three ensemble voting strategies:

- **Majority vote:** the ensemble verdict is the majority among member judges, with ties on an even panel broken conservatively toward “incorrect”;
- **Conservative vote:** any judge flagging “incorrect” makes the ensemble verdict “incorrect”;
- **Unanimous vote:** all judges must agree on “correct” for the ensemble to render a “correct” verdict. Because each verdict is binary (exactly “correct” or “incorrect”), requiring unanimous “correct” is identical to the conservative rule, since either way a single “incorrect” vote sets the verdict to “incorrect”; the two rules would diverge only under a graded, non-binary verdict scale, which we do not use.

Choosing the panel that scores best on a dataset and then reporting that same score overstates performance: with many candidate panels, the maximum score partly reflects which panel happened to fit that dataset’s noise, so it does not generalize, a selection bias well documented in model selection [16]. We avoid it with nested cross-validation: within each fold, the best panel per voting rule is selected on the nine training folds and scored once on the disjoint held-out fold, and we report the mean $\pm$ standard deviation across the ten folds. Candidates are the panels of size two to five formed from the twelve judges with the highest individual error-detection F1 on the training folds; the four reference baselines are excluded so that the oracle cannot be selected. The vote ignores unparseable member verdicts, and a sample is excluded from the ensemble’s accuracy, F1, and FPR only when *every* member is unparseable; such cases are reported separately as the ensemble unparseable rate.

Table 6 Condition A reliability (mean over 10 folds), ordered by MCC. The two headline axes, overall quality (MCC) and the safety KPI (FPR, lower is better), are placed adjacent so their dissociation reads directly. MCC is a correlation coefficient in $[-1, 1]$; the other four columns are percentages. Header arrows mark whether higher ($\uparrow$) or lower ($\downarrow$) is better. “bAcc” is balanced accuracy; “eF1” is error-detection F1; “-R” denotes the reasoning-enabled configuration. Reference baselines are shown in the bottom block. The table lists the four baselines together with a representative selection of judges spanning the full range; **bold** marks the best judge per metric (baselines excluded), and the full MCC ranking of all 52 judges appears in [Appendix E](#).

Judge	MCC $\uparrow$	FPR (%) $\downarrow$	bAcc (%) $\uparrow$	Acc (%) $\uparrow$	eF1 (%) $\uparrow$
Gemini-3.1-Pro	0.65	14.82	84.76	84.97	89.46
Claude-Opus-4.8-R	0.54	15.23	78.22	81.50	87.30
GPT-5.1-R	0.50	13.42	75.21	80.93	87.21
Claude-Sonnet-4.6-R	0.49	23.64	77.26	76.82	83.15
Claude-Opus-4.8	0.48	10.26	73.15	81.46	87.90
Grok-4-R	0.44	20.67	74.02	76.67	83.61
DeepSeek-Reasoner	0.40	20.02	71.36	75.68	83.16
Claude-Sonnet-4.6	0.39	17.75	70.39	76.33	83.91
GPT-5-mini-R	0.37	14.95	68.12	76.61	84.51
GLM-4.6	0.36	18.84	68.56	74.86	82.86
Claude-Haiku-4.5-R	0.35	18.63	68.27	74.83	82.89
GPT-5-mini	0.34	11.24	65.58	77.19	85.38
Qwen3-235B-thinking	0.33	12.35	65.10	76.41	84.79
Claude-Haiku-4.5	0.24	6.27	58.35	76.08	85.47
gpt-oss-120b	0.24	17.10	61.54	72.26	81.77
Prometheus-2-7B	0.08	47.95	54.72	53.37	62.49
DeepSeek-R1-Distill-70B	0.08	65.75	54.38	44.30	47.89
SauLLM-7B (legal)	0.07	21.21	53.62	66.23	77.77
Skywork-Critic-8B	0.07	18.12	53.24	67.59	79.12
Baseline-Oracle	1.00	0.00	100.00	100.00	100.00
Baseline-AlwaysReject	0.00	0.00	50.00	75.06	85.75
Baseline-AlwaysAccept	0.00	100.00	50.00	24.94	0.00
Baseline-Random	0.00	50.18	50.25	50.04	59.88

5 Results

5.1 Judge Reliability on Human-Authored MCQ (Condition A)

[Table 6](#) reports judge reliability on the 3,228 MCQ evaluations of Condition A, with all metrics averaged over the 10 cross-validation folds and judges ordered by MCC. The four reference baselines anchor the scale: the oracle attains a perfect MCC of 1.0, while the three trivial policies (always-reject, always-accept, random) all collapse to an MCC of ≈ 0 , confirming that the 25%/75% class imbalance renders raw accuracy uninformative: always-reject reaches 75% accuracy with zero discriminative power.

Even the Best Judge Is Only Moderately Reliable.

The strongest judge, **Gemini-3.1-Pro**, reaches only an MCC of 0.65 with a balanced accuracy of 84.76%; its bootstrap interval $[0.62, 0.67]$ falls well short of the perfect oracle, by some fourteen standard errors, so even the best judge is only moderately reliable

rather than close to the ground truth. A cluster of frontier proprietary models follows, **Claude-Opus-4.8** at 0.54 with reasoning, **GPT-5.1** at 0.50, and **Claude-Sonnet-4.6** at 0.49, after which reliability degrades steadily. A large group of judges, including the legal-domain specialist **SaulLM-7B** at 0.07 and the evaluation-tuned models **Prometheus-2-7B** at 0.08 and **Skywork-Critic-8B** at 0.07, lands barely above the random baseline near zero: on this task, neither domain pre-training nor generic evaluation tuning transfers to Quebec insurance law.

Inference-Time Reasoning Consistently Helps.

For every model family evaluated in both modes, enabling reasoning improves MCC: **Claude-Opus-4.8** $0.48 \rightarrow 0.54$, **Claude-Sonnet-4.6** $0.39 \rightarrow 0.49$, **Claude-Haiku-4.5** $0.24 \rightarrow 0.35$, and **GPT-5-mini** $0.34 \rightarrow 0.37$. The gain is largest for the smaller models: extra reasoning lifts them more than it lifts the already-strong frontier judges, narrowing the gap to them without closing it. This fits the view that chain-of-thought reasoning allows a model to work through intermediate steps it would otherwise skip [69], and that spending more computational resources at inference can, in part, substitute for raw model capacity [61]. On a legally nuanced task such as this one, that substitution is only partial: reasoning sharpens a judge’s verdicts. However, it does not supply the domain knowledge a small model lacks, so a reasoning-enabled small judge still trails a frontier model.

Aggregate Quality Does Not Predict Dangerousness.

A judge’s overall quality does not predict how dangerous it is: across the 52 judges, FPR shows no significant rank correlation with MCC, with a Kendall τ of just 0.14, so the best-ranked judge is not the safest to deploy. Figure 1 makes this visible: judges of similar MCC span almost the full range of FPR, and the frontier judges are no safer than many weaker ones. The two failure modes are distinct. Some judges with a respectable MCC still wave through many errors, such as **Claude-Sonnet-4.6-reason**, which validates nearly a quarter of incorrect answers. Others reach a low FPR only by rejecting almost everything, like **Command-R-Plus**, whose near-always-reject behaviour leaves its MCC at 0.12; at the extreme, **DeepSeek-R1-Distill-70B** validates two-thirds of incorrect answers, worse than chance. Because quality and safety come apart this way, we report FPR separately rather than folding it into a single score.

Why Severity, Not Accuracy, Is the Objective.

This dissociation has a direct cost reading [12]: expected cost sums, over error types, how often each error occurs times its per-error severity, and a false positive, a wrong legal answer reaching a policyholder, carries a severity far above that of a false rejection, which merely routes a correct answer to human re-check. Accuracy weights the two error types equally and therefore misranks judges on cost. Concretely, **Claude-Haiku-4.5** and **Claude-Sonnet-4.6-reason** reach near-identical accuracy (76.08% vs 76.82%) yet their FPR differs almost fourfold (6.27% vs 23.64%). Because the severity-bearing term of expected cost scales with FPR, the higher-FPR judge incurs nearly four times the dangerous-error cost at equal accuracy, and the gap widens

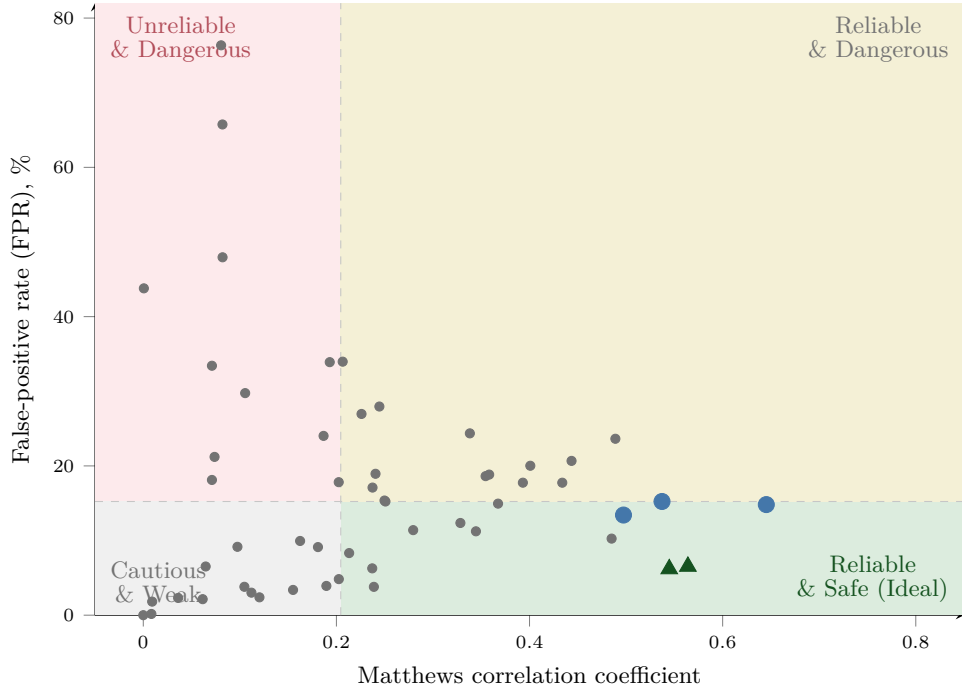


Figure 1 False-positive rate versus MCC across the 52 judges on Condition A (human-authored MCQ; excluding baselines). Markers encode judge type: black circles are the individual judges, blue circles the three highest-MCC single judges (*Gemini-3.1-Pro*, *Claude-Opus-4.8-reason*, *GPT-5.1-reason*), and green triangles the held-out ensembles (Section 5.4). Dashed lines split the judges at the median MCC (0.20) and the median FPR (15.23%) into four quadrants; the lower-right *Reliable & Safe (Ideal)* quadrant (high MCC, low FPR) is the target, the upper-left *Unreliable & Dangerous* quadrant the worst. The three top judges sit around the median FPR rather than in the ideal quadrant, so aggregate quality does not predict safety: they are no safer than many lower-MCC judges. The ensembles achieve the lowest FPR of any configuration and fall within the ideal quadrant, yet have only a moderate MCC (≈ 0.55 , below the best single judge), which again separates safety from aggregate quality.

as the severity ratio between a missed legal error and a false alarm increases; this is why we minimize FPR rather than maximize accuracy.

Judge Differences Are Statistically Significant.

The judges are genuinely different, not noisy copies of one underlying model (Table 7): Cochran’s Q test rejects equal accuracy across the 52 judges, and pairwise McNemar tests find 61% of judge pairs significantly different after Bonferroni correction. The estimates themselves are stable, with tight bootstrap confidence intervals. *Gemini-3.1-Pro*’s MCC sits above every other judge. Still, the frontier-reasoning models behind it have overlapping intervals, so we read them as a distinguishable top tier rather than a strict ranking.

Table 7 Inferential tests of judge differences on Condition A, computed on the 52 judges over the 2,743 samples every judge parsed.

Test	Question	Result
Cochran’s Q	Are all judges equally accurate?	$Q = 7,703$, $df = 51$, $p < 0.001$
McNemar, pairwise (Bonferroni)	Do two given judges differ?	805 of 1,326 pairs (61%) significant
Bootstrap 95% CI	How precise are the metrics?	median half-width 0.04 (MCC), 1 pp (FPR)

Metric Choice.

The Spearman rank correlations among metrics (over the 52 judges) confirm the methodological choices of [Section 4.5](#): MCC and balanced accuracy rank judges almost identically ($\rho = 0.97$), whereas plain accuracy correlates only moderately with MCC ($\rho = 0.63$) and error-detection F1 weakly ($\rho = 0.24$). Error-F1 is, in turn, strongly anti-correlated with FPR (Kendall $\tau = -0.67$), as expected, since both reward catching incorrect answers ([Figure 2](#)); note that Kendall τ runs lower in magnitude than the Spearman ρ reported above for an equally monotonic relationship, so this value reflects a tighter association than a Spearman coefficient of similar size would. This contrasts with the near-flat MCC–FPR cloud ([Figure 1](#)); the point is that error-detection F1, not aggregate quality, is what tracks safety.

Robustness to Abstention.

Excluding UNPARSEABLE verdicts could flatter a judge who abstains in hard cases, and abstention is mildly correlated with MCC; the leader **Gemini-3.1-Pro** abstains the most, at 6.8%. Recomputing MCC with every abstention scored as “incorrect”, the worst case, leaves the ranking unchanged and slips the leader’s MCC only from 0.65 to 0.61; [Appendix D](#) gives the full check.

5.2 Inter-Judge Agreement (Condition A)

If LLM judges were reliable proxies for legal correctness, they would largely agree with one another on this binary, ground-truth task. They do not. Overall agreement is only *slight* on the Landis & Koch scale: Fleiss’ κ is 0.19, and pairwise Cohen’s κ averages just 0.17, with most judge pairs below 0.20 and almost none reaching substantial agreement.

Agreement Tracks Capability, Not Architecture.

The most concordant pairs are exclusively frontier models that judge alike, for example, **Claude-Opus-4.8-reason** and **Gemini-3.1-Pro**, whereas weaker and evaluation-specialized judges agree with almost no one. Reasoning and standard variants of the same model agree closely, so enabling reasoning sharpens rather than reshapes a model’s verdicts. The practical implication is clear: the choice of judge significantly

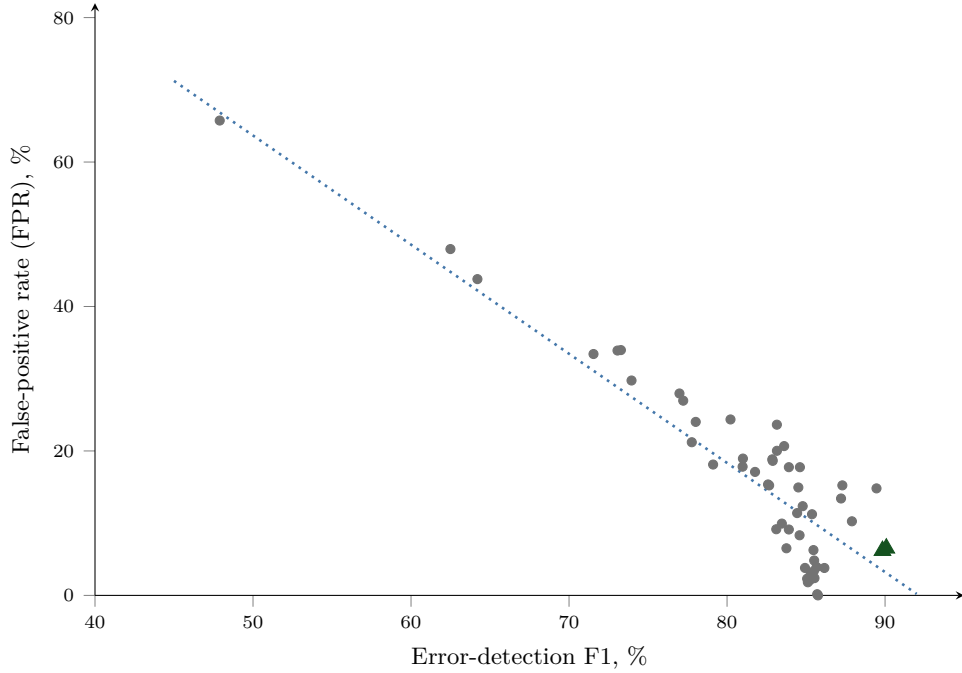


Figure 2 Error-detection F1 against the false-positive rate across the 52 judges on Condition A (human-authored MCQ; excluding baselines). Markers: black circles are the individual judges and green triangles the held-out ensembles; the dotted blue line is the least-squares fit (Kendall $\tau = -0.67$, $p < 0.001$). Unlike the flat MCC–FPR relation of Figure 1, the cloud falls steeply, so judges that catch more errors are safer. Judges *below* the line have a lower FPR than their error-F1 predicts (safer than expected); those *above* it are more dangerous than expected. The ensembles achieve the highest F1 score with the lowest FPR. Error-F1, not MCC, predicts FPR.

changes which answers are deemed correct, so single-judge evaluation of high-stakes legal QA is unreliable.

No Question Is Universally Easy or Hard.

Across the roster, there is no sample that every judge gets right and none that every judge gets wrong; instead, 314 samples are *divisive*, falling in the top decile for inter-judge disagreement. Errors are thus distributed across judges rather than concentrated on a few intractable questions, which is precisely the regime in which ensembling can help (Section 5.4).

5.3 AI-Generated Text and Cross-Condition Comparison (Condition B)

In Condition B, each judge evaluates every RAG answer and its reference from the 20 open-source generators, issuing 16,140 verdicts on AI-generated free text. Because free-text correctness has no automatic ground truth, we characterize Condition B through inter-judge agreement and through how each judge’s behaviour shifts relative to Condition A.

Agreement Stays Low on AI Text.

Pairwise agreement on AI text is *fair* (mean Cohen’s $\kappa = 0.26$), modestly above the *slight* agreement on human MCQ. This small rise is mostly a by-product of the harsher-on-AI behaviour reported next. Because most judges reject the large majority of RAG answers, they agree by uniformly voting “incorrect” rather than by converging on a shared notion of correctness. Either way, agreement never exceeds *fair*, so judges remain unreliable proxies for legal correctness on AI-generated text as well.

Judges Are Harsher on AI Text, Not More Lenient.

We compare each judge’s rate of “correct” verdicts between the two conditions with a two-proportion z-test (Equation 2). The shift is overwhelmingly negative: of the 51 judges present in both conditions, 42 accept AI answers *less* often than human MCQ candidates, and the drop is large for the strong judges; **Gemma-3-27B**, for instance, falls from 39% to 6%. Only a handful of weaker judges become more permissive. This runs *against* the AI-favoritism reported in general-domain settings [53, 57]. The two conditions differ in their base rate of correct candidates, so to check that the tendency is not merely this base rate, we recompute the comparison using only the *known-correct* MCQ choices, discarding the distractors that inflate the human side with answers a judge should reject. The gap does not close; it widens: across the roster, judges accept the known-correct human choices far more readily than the RAG answers, a mean acceptance gap of 27 percentage points (for example **Gemini-3.1-Pro**, 84% on correct human choices versus 5% on RAG answers), and the large majority of judges stay harsher on AI under this control. A residual gap could still reflect genuine RAG errors, which a Condition B gold set would disentangle (see Limitations); but the MCQ base rate alone does not explain the effect, so we read it as a behavioural tendency that judges do not favour AI text here, rather than a fully bias-corrected estimate.

No Self-Preference Once Provenance Is Controlled.

Using the $N \times N$ design to hold provenance and the candidate pool fixed, self-preference is essentially nil on average, with a mean $\Delta = +0.01$. We thus find *no* self-enhancement bias [53, 57], consistent with evaluators favouring human over machine text [35]; Appendix D details the controlled comparison.

Reasoning Grows on AI Text.

Judges also write substantially longer justifications on AI answers than on MCQ choices, consistent with free-text demanding more explanation than a single option.

5.4 Judging Frameworks: References, Ensembles, and Voting

Given that reliability and safety come apart, so that no single judge pairs high reliability with a low false-positive rate (the most reliable judges carry middling FPR, Section 5.1), we study three framework-level levers to reduce false positives, all tuned and evaluated using the 10-fold protocol.

Reference-Augmented Prompting.

Adding the corpus reference answer and expert justification to the prompt is a strong mitigation for weaker judges: on the open-source MCQ subset mean MCC rises from 0.12 to 0.67, and on free-text Condition B answers the reference rewrites between a fifth and a third of the verdicts of small and evaluation-specialized judges (McNemar $p < 0.001$) while leaving frontier judges largely unchanged, raising inter-judge agreement from Fleiss’ $\kappa = 0.13$ to 0.20. It is therefore a high-value lever for the cheaper judges most likely to be deployed at scale, not a universal substitute for judge quality. The full ablation, on both MCQ and free text, is reported in [Appendix D](#).

Ensembles.

Because errors are distributed across judges and questions rather than concentrated ([Section 5.2](#)), panels help. Under the nested-CV protocol ([Section 4.7](#)), the unbiased held-out search converges on a small panel built around the same frontier judges: for the conservative and unanimous rules it selects the *same* three in all ten folds, **Gemini-3.1-Pro**, **Claude-Opus-4.8-reason**, and **GPT-5.1-reason**, while the majority rule keeps these three and adds one or two weaker judges, most often **Command-R** and **Command-R-Plus** and occasionally **Qwen3-32B** or **Qwen2.5-3B**, for a mean size of 4.8. That the data-driven search independently recovers the top of the individual MCC ranking is reassuring in itself. Against the best single judge, **Gemini-3.1-Pro** at accuracy 84.97%, error-F1 89.46%, and FPR 14.82%, the held-out *majority* panel holds accuracy at $84.57\% \pm 1.96$ and slightly raises error-F1 to 90.08% while lowering FPR to $6.53\% \pm 2.63$.

Voting as a Decision Threshold.

Because judges emit a binary verdict rather than a calibrated score, there is no per-judge probability to threshold; the threshold instead lives in the ensemble vote rule, i.e. the fraction of members that must say “correct”. Making the rule *conservative*, so that a single member voting “incorrect” sets the panel verdict to “incorrect”, is the strictest such threshold: its held-out panel reaches accuracy $84.07\% \pm 2.08$, error-F1 89.84%, and FPR $6.19\% \pm 1.98$, roughly a $2.4\times$ reduction in false positives over the best single judge; the unanimous rule coincides with it in the binary case. For high-stakes deployment, such a conservative frontier panel, not any single judge, is the recommended operating point ([Figure 3](#)).

6 Discussion

Reliability, Not Capability, Is the Binding Constraint.

The frontier judges we tested are highly capable models, yet even the best one clears only a moderate reliability bar, as [Table 6](#) shows, and their chance-corrected agreement is only slight, as [Section 5.2](#) reports. The bottleneck is therefore not raw model quality but the calibration of judgement to a narrow, codified legal standard: a judge must apply the exact provisions of Quebec civil-law insurance, in French, where a fluent answer can be precisely wrong. This reframes the question for high-stakes domains;

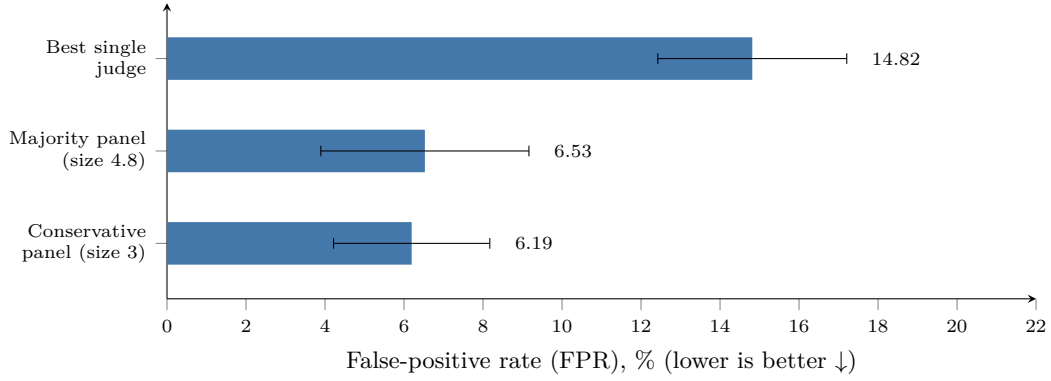


Figure 3 False-positive rate of the best single judge (**Gemini-3.1-Pro**) versus the held-out (nested-CV) ensembles, all on Condition A (human-authored MCQ). The majority panel roughly halves FPR, and the conservative panel (which coincides with the unanimous rule in the binary case) reaches a $\sim 2.4\times$ reduction, at the cost of a small drop in accuracy. Error bars are ± 1 standard deviation across the ten folds.

what matters is not whether a model is strong in general but whether any single model’s verdicts are stable enough to stand unaudited, and ours are not.

Why Aggregate Quality Misleads About Safety.

The dissociation between MCC and the false-positive rate, visible in [Figure 1](#), is structural rather than incidental. MCC rewards balanced discrimination across both error directions. In contrast, the operational risk here is one-sided: endorsing a wrong legal answer is far more costly than wrongly flagging a correct one, so a judge can score well on the symmetric metric while remaining unsafe on the asymmetric one. What covaries with safety is error-detection F1, as [Figure 2](#) shows: because both F1 and the FPR reward catching incorrect answers, F1 falls as the FPR falls, making it a usable selection proxy where MCC is not. The consequence generalizes beyond insurance: a high-stakes evaluator should be chosen and reported on a cost-weighted, direction-aware metric [\[12\]](#), never on a single aggregate score.

Specialization Does Not Survive Distribution Shift.

That the evaluation-tuned and legal-domain models fail to transfer is informative about what specialization actually buys. Evaluation-tuning encodes general rubrics, and the available legal models encode largely English common law, whereas Quebec insurance is codified civil law in French; both forms of specialization therefore sit on the wrong side of a distribution shift. The lesson is not that specialization is useless but that it must match the target jurisdiction and language; absent a corpus-matched specialist, a strong general reasoner is the safer default.

Bias Appears to Depend on the Setting, Not Only the Model.

The reversal of the widely-reported AI-favouritism suggests that evaluator bias depends heavily on task framing rather than being intrinsic to LLM judges. Under

strict legal-accuracy instructions, in a domain with a hard ground truth, the same class of models that flatters machine text on open-ended benchmarks [53, 57] becomes, if anything, harsher on it. Bias priors measured on general benchmarks should therefore not be carried over to specialized evaluation, since the sign of the effect can flip with the instructions and domain.

What Actually Reduces Risk, and Why.

The three levers that lower the false-positive rate act through different mechanisms. Firstly, conservative ensembles work because judge errors are distributed across judges rather than concentrated on universally hard questions, so a single veto suppresses individual lapses [67]. Secondly, an authoritative reference works by injecting the legal knowledge weak judges lack, which is why, as Appendix D reports, it helps cheap judges most and frontier judges least. Finally, routing high-severity cases to humans, a domain instance of the reject option [30], bounds the residual tail directly. None of them turns an LLM into a standalone adjudicator; each instead hardens a triage layer whose objective is to be safe under cost asymmetry.

Efficiency and Scalability.

The two families of LLMs differ sharply in operating cost. The open judges run locally at no per-query charge, with a median latency of 0.24 s per evaluation. In contrast, the API-served judges are billed per token, so a single Condition A pass reaches roughly USD \$10 per judge and far more across the full cross-condition design. This sets the economics of the triage design: the open roster, or a small frontier ensemble, can absorb bulk screening, leaving the most expensive judges and human experts for the high-severity residue.

Implications for High-Stakes Deployment.

These findings bear on the liability exposure that motivates the work: as the Air Canada and Gemini rulings show [15, 42], an organization that lets an unaudited model speak for it inherits the kind of tail risk that FPR is designed to surface, to the extent that our MCQ-based FPR estimates transfer to the free-text answers such systems actually produce, which our design measures only indirectly through inter-judge agreement. Our results argue for a concrete protocol: report MCC *and* FPR rather than accuracy alone; prefer a small conservative ensemble of frontier reasoning models to any single judge; supply an authoritative reference where one exists; and escalate uncertain or high-severity cases to human experts (a recommendation grounded in prior work on the reject option [30] and cost-weighted evaluation [12], not evaluated empirically in this study). More broadly, reliable automatic evaluation in a low-resource, high-stakes domain is attainable, but only as a cost-aware, human-in-the-loop system, not as a drop-in replacement for expert judgement.

Scope: The Offline Evaluation Setting.

Our deployment recommendations do not carry over wholesale to the offline uses that also drive LLM-as-a-Judge adoption, reward modelling for training, alignment, and leaderboards. There, a human-in-the-loop is impractical at scale, and individual

mislabels are diluted by averaging over many items, so the operational concern is aggregate bias rather than any single validated error. Our core findings still hold there: single-judge evaluation is unreliable, ensembles and inference-time reasoning help, and evaluators should be selected and reported on direction-aware metrics rather than a single aggregate score. What differs is the safety mechanism: an offline pipeline absorbs residual error through scale and re-sampling, whereas a deployed legal-accuracy gate must route it to a human. We therefore do not prescribe human adjudication for training-time judging; the human-in-the-loop protocol is specific to the high-stakes, user-facing deployment that is our focus.

7 Conclusion

We evaluated 52 LLM-judge configurations (open-sourced and API-served, reasoning and standard) for detecting factual errors in Quebec automobile insurance question answering, a task that combines a specialized legal domain, a civil-law tradition, and a regional variety of French. Against a non-public, expert-authored benchmark, even the strongest judge achieved only moderate reliability, with an MCC of 0.65; inter-judge agreement was no better than slight, and a judge’s aggregate quality failed to predict the rate of false positives, the error mode that MCQ evaluation identifies as the costly one. Inference-time reasoning consistently helped, while evaluation-specialized models and a legal-domain model did not transfer; and, contrary to general-domain findings, most judges were harsher, not more lenient, toward AI-generated answers. Selecting judges by error-detection F1 surfaces the safer ones; supplying an authoritative reference and aggregating frontier judges into conservative ensembles each reduces false positives, supporting a deployment pattern in which an LLM panel triages and humans adjudicate the high-severity cases. Taken together, these results caution against single-judge LLM evaluation for high-stakes legal question answering; for its user-facing deployment, they argue for severity-aware, human-in-the-loop protocols, whereas offline uses such as training and leaderboards inherit the same reliability caution but not that remedy. Future work includes extending the analysis to other insurance lines and jurisdictions, building a free-text gold standard so that the false-positive rate can be validated beyond the multiple-choice setting, and pairing judge reliability with the cost models that quantify its downstream impact.

Limitations

This study has several limitations. First, the AEPC-QA benchmark is not publicly available due to licensing restrictions on the AMF examination preparation materials. While this reduces the risk of contamination, it limits external reproducibility. We provide our evaluation code and prompt templates to support replication with other benchmarks. Second, our reliability estimates, including the headline MCC and the false-positive rate, are validated only against the binary MCQ ground truth of Condition A; the deployment-facing free-text setting (Condition B) has no gold standard, so we characterize it through inter-judge agreement alone and do not claim that the same accuracy or FPR guarantees transfer to free-text answers. The MCQ format

also reduces legal accuracy to binary correctness, which may not capture all dimensions of correctness in free-text legal reasoning. Third, our evaluation is limited to Quebec automobile insurance; generalization to other insurance lines, jurisdictions, or legal domains would require additional study. Fourth, model versions and capabilities evolve; our results reflect the models available at the time of experimentation (early 2026). The API-served judges, in particular, are served via opaque, versioned endpoints whose behaviour may change server-side, so only the open-source subset of our pipeline is fully and indefinitely reproducible. In particular, the leading judge, **Gemini-3.1-Pro**, was queried through the OpenRouter gateway on a *preview* endpoint ([Appendix C](#)); its headline figures should be read as a snapshot that may differ from the general-availability model. Fifth, the two judge families are not perfectly comparable: the open-source models are served under a controlled FP8/MXFP4 quantization and decoding configuration, whereas the API-served judges are queried through provider APIs whose internal precision, sampling, and reasoning controls we cannot fully observe; we mitigate this by mirroring context length and JSON constraints across families and by reporting both families on identical metrics, but residual configuration differences cannot be ruled out. Sixth, the reference-augmented result is an upper bound rather than a deployment estimate: for MCQ candidates, the supplied reference is close to the gold answer, so the measured gain partly reflects matching against a near-gold string, and any real-world benefit is contingent on an authoritative reference being retrievable for the question at hand. Seventh, ground truth is derived from a single official answer key without multi-annotator adjudication; we flagged two questions whose keys fall outside their choice sets and excluded them, but undetected key errors and the absence of inter-annotator agreement on correctness remain possible sources of label noise. Eighth, our reliability estimates are computed against an expert-authored ground truth, the examination answer key and its justification, not against human judges performing the same evaluation task. We therefore measure how faithfully LLM judges track the expert-defined correct answer, not whether they match the agreement, error rates, or blind spots of human evaluators, who themselves disagree and err; establishing human-judge parity would require a separate multi-annotator study and is beyond our scope. Ninth, the conservative held-out ensembles that most reduce the false-positive rate converge on frontier proprietary judges, so the safest configurations inherit the cost, latency, and reproducibility caveats of closed, versioned endpoints; the fully open subset is reproducible but does not reach the same safety.

Ethical Considerations

This work studies LLM evaluators in a high-stakes legal and insurance setting, where an erroneous “correct” verdict, a false positive, can let a wrong answer about a person’s rights or coverage pass unchallenged. We therefore frame reliability around this asymmetric harm rather than aggregate accuracy, and our central recommendation is explicitly cautionary: in such high-stakes legal settings, LLM judges should be deployed only as decision aids within a severity-aware, human-in-the-loop protocol, never as unsupervised adjudicators of legal correctness. The false-positive rate is

reported precisely so that the residual risk remains visible to practitioners and is not obscured by a single headline score, thereby counteracting the automation bias that can lead users to over-trust fluent model output.

The benchmark derives from Quebec regulatory examination materials accessed under a research agreement; it contains no personal data and is not released to respect the licensing of the source materials and to limit benchmark contamination. Our findings are specific to Quebec automobile insurance expressed in Quebec French, and transferring conclusions about judge reliability to other languages, jurisdictions, or legal domains without revalidation could yield inequitable or unsafe outcomes. Finally, large-scale LLM judging incurs nontrivial computational and energy costs; we mitigate them by serving open models locally with FP8/MXFP4 quantization and by showing that small, conservative ensembles of a few judges, rather than brute-force panels of many, achieve the largest safety gains. We release our code and prompts to enable scrutiny and replication, not to encourage unsupervised automated adjudication of legal advice.

Data Availability

The evaluation code, prompt templates, analysis pipeline, and RAG generation scripts will be made available upon acceptance. To support replication in the meantime, [Appendix C](#) records the complete pinned software stack, the fixed random seed, and the exact model identifiers (HuggingFace repositories and provider model strings) for all 52 judge configurations, and [Appendix A](#) gives the verbatim prompts. The AEPC-QA benchmark is not publicly available due to licensing restrictions.

Declarations

Funding.

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Conflict of Interest.

The authors declare no conflict of interest.

Ethics Approval.

This study uses open-source LLMs run locally and commercial LLM APIs, evaluated on a proprietary examination benchmark accessed under a research agreement. No human subjects were involved, and no personal data was processed.

Consent to Participate.

Consent to Participate declaration: not applicable.

Consent to Publish.

Consent to Publish declaration: not applicable.

References

- [1] Aftahee S, Farhad AFM, Mallik A, et al (2025) Assessing the Reliability of Large Language Models in the Bengali Legal Context: A Comparative Evaluation Using LLM-as-Judge and Legal Experts. arXiv:2511.05627
- [2] Ariai F, Mackenzie J, Demartini G (2025) Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges. ACM Computing Surveys
- [3] Autorité des marchés financiers (2025) Quebec Automobile Insurance Policy, Form F.P.Q. No. 1 (Owner’s Form). <https://lautorite.qc.ca/en/professionals/insurers/automobile-insurance/amf-approved-forms>, standard automobile insurance policy form approved by Québec’s financial-sector regulator under the Insurers Act (CQLR c. A-32.1)
- [4] Bavaresco A, Bernardi R, Bertolazzi L, et al (2025) LLMs instead of Human Judges? A Large Scale Empirical Study across 20 NLP Evaluation Tasks. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Short Papers, arXiv:2406.18403
- [5] Beauchemin D (2025) Automatic Explanation of Automobile Insurance Contracts Using Retrieval-Augmented Generation. PhD thesis, Université Laval
- [6] Beauchemin D, Khoury R (2024) Quebec Automobile Insurance Question-Answering with Retrieval-Augmented Generation. In: Proceedings of the Natural Legal Language Processing Workshop
- [7] Beauchemin D, Khoury R (2025) QFrCoLA: a Quebec-French Corpus of Linguistic Acceptability Judgments. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)
- [8] Beauchemin D, Khoury R (2026) Benchmarking Large Language Models for Quebec Insurance: From Closed-Book to Retrieval-Augmented Generation. arXiv:2603.07825
- [9] Beauchemin D, Saggion H, Khoury R (2023) MeaningBERT: assessing meaning preservation between sentences. *Frontiers in Artificial Intelligence* 6:1223924
- [10] Beauchemin D, Albert-Rochette M, Khoury R, et al (2025) JudgeBERT: Assessing legal meaning preservation between sentences. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing
- [11] Beauchemin D, Tremblay Y, Youssef MA, et al (2025) COLE: a Comprehensive Benchmark for French Language Understanding Evaluation. arXiv:2510.05046
- [12] Beauchemin D, Khoury R, Goulet V (2026) Machine Learning Evaluation Should Adopt the Actuarial Frequency-Severity Model. In: Advances in Neural

- [13] Beauchemin D, Veilleux PL, Khoury R, et al (2026) QFrBLiMP: a Quebec-French Benchmark of Linguistic Minimal Pairs. In: Findings of the Association for Computational Linguistics: EACL
- [14] BerriAI (2023) LiteLLM: Call all LLM APIs using the OpenAI format. <https://github.com/BerriAI/litellm>
- [15] British Columbia Civil Resolution Tribunal (2024) Moffatt v. Air Canada, 2024 BCCRT 149. <https://www.canlii.org/en/bc/bccrt/doc/2024/2024bccrt149/2024bccrt149.html>, tribunal decision holding Air Canada liable for its support chatbot’s inaccurate statements
- [16] Cawley GC, Talbot NLC (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research* 11:2079–2107
- [17] Chiang CH, Lee Hy (2023) Can Large Language Models Be an Alternative to Human Evaluations? In: Proceedings of the Annual Meeting of the Association for Computational Linguistics
- [18] Chicco D, Jurman G (2020) The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. *BMC Genomics* 21(1):6
- [19] Cochran WG (1950) The Comparison of Percentages in Matched Samples. *Biometrika* 37(3/4):256–266
- [20] Cohen J (1960) A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20(1):37–46
- [21] Cui J, Shen X, Wen S (2023) A Survey on Legal Judgment Prediction: Datasets, Metrics, Models and Challenges. *IEEE Access* 11:102050–102071
- [22] Ding J, Feng K, Lin B, et al (2025) InsQABench: Benchmarking Chinese Insurance Domain Question Answering with Large Language Models. *arXiv:2501.10943*
- [23] Dunn OJ (1961) Multiple comparisons among means. *Journal of the American Statistical Association* 56(293):52–64
- [24] Efron B, Tibshirani RJ (1994) *An Introduction to the Bootstrap*. Chapman & Hall/CRC, New York
- [25] Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of the International Joint Conference on Artificial Intelligence

- [26] Enguehard J, Van Ermengem M, Atkinson K, et al (2025) LeMAJ (Legal LLM-as-a-Judge): Bridging Legal Reasoning and LLM Evaluation. In: Proceedings of the Natural Legal Language Processing Workshop, EMNLP
- [27] Es S, James J, Espinosa-Anke L, et al (2024) RAGAS: Automated Evaluation of Retrieval Augmented Generation. In: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations
- [28] Feng M, Xiang B, Glass MR, et al (2015) Applying Deep Learning to Answer Selection: A Study and an Open Task. In: IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)
- [29] Fleiss JL (1971) Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin* 76(5):378–382
- [30] Geifman Y, El-Yaniv R (2017) Selective classification for deep neural networks. In: *Advances in Neural Information Processing Systems*
- [31] Government of Québec (1991) Civil code of québec. CQLR c. CCQ-1991
- [32] Gu J, Jiang X, Shi Z, et al (2025) A Survey on LLM-as-a-Judge. *The Innovation* arXiv:2411.15594
- [33] Guerdan L, Barocas S, Holstein K, et al (2025) Validating LLM-as-a-Judge Systems Under Rating Indeterminacy. In: *Advances in Neural Information Processing Systems*
- [34] Hada R, Gumma V, de Wynter A, et al (2024) Are Large Language Model-based Evaluators the Solution to Scaling Up Multilingual Evaluation? In: *Findings of the Association for Computational Linguistics: EACL*
- [35] Haverals W, Martin M (2025) Everyone Prefers Human Writers, Including AI. arXiv:2510.08831
- [36] Hendrycks D, Burns C, Chen A, et al (2021) CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. In: *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*
- [37] Huang H, Bu X, Zhou H, et al (2025) An Empirical Study of LLM-as-a-Judge for LLM Evaluation: Fine-tuned Judge Model is not a General Substitute for GPT-4. In: *Findings of the Association for Computational Linguistics: ACL*
- [38] Joshi P, Santy S, Budhiraja A, et al (2020) The state and fate of linguistic diversity and inclusion in the NLP world. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*

- [39] Karp M, Kubaszewska A, Król M, et al (2026) LLM-as-a-judge is bad, based on AI attempting the exam qualifying for the member of the Polish National Board of Appeal. Artificial Intelligence and Law <https://doi.org/10.1007/s10506-026-09505-w>
- [40] Koo R, Lee M, Raheja V, et al (2024) Benchmarking Cognitive Biases in Large Language Models as Evaluators. In: Findings of the Association for Computational Linguistics: ACL
- [41] Kwon W, Li Z, Zhuang S, et al (2023) Efficient Memory Management for Large Language Model Serving with PagedAttention. In: Proceedings of the Symposium on Operating Systems Principles
- [42] Landgericht München I (Regional Court of Munich) (2026) AI Overviews liability injunction, Case 26 O 869/26. <https://www.heise.de/en/news/LG-Munich-I-Google-ordered-to-pay-for-false-statements-in-AI-summaries-11327217.html>, ruling of 28 May 2026 holding Google directly liable for false statements produced by its AI Overviews. Reported by heise online, *the-decoder*, and the Transparency Coalition (which mirrors the decision text): <https://the-decoder.com/landmark-german-ruling-declares-googles-ai-overviews-are-googles-own-words-and-makes-it-liable-for-false-answers/>, <https://www.transparencycoalition.ai/news/german-court-holds-google-liable-for-ai-hallucination-read-the-full-decision-here>
- [43] Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. *Biometrics* 33(1):159–174
- [44] van der Lee C, Gatt A, van Miltenburg E, et al (2021) Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language* 67:101151
- [45] Lewis P, Perez E, Piktus A, et al (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems
- [46] Li H, Dong Q, Chen J, et al (2024) LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods. arXiv:2412.05579
- [47] Li Y, Wang S, Ding H, et al (2023) Large language models in finance: A survey. In: Proceedings of the Fourth ACM International Conference on AI in Finance
- [48] Liu Y, Iter D, Xu Y, et al (2023) G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing
- [49] Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) – Protein Structure* 405(2):442–451

- [50] McNemar Q (1947) Note on the Sampling Error of the Difference between Correlated Proportions or Percentages. *Psychometrika* 12(2):153–157
- [51] Medvedeva M, Vols M, Wieling M (2020) Using machine learning to predict decisions of the European court of human rights. *Artificial Intelligence and Law* 28(2):237–266
- [52] National Assembly of Québec (1977) Automobile Insurance Act, CQLR c. A-25. <https://www.legisquebec.gouv.qc.ca/en/document/cs/A-25>, québec’s public no-fault bodily-injury automobile regime, administered by the Société de l’assurance automobile du Québec (SAAQ)
- [53] Panickssery A, Bowman SR, Feng S (2024) LLM Evaluators Recognize and Favor Their Own Generations. *arXiv:2404.13076*
- [54] Papineni K, Roukos S, Ward T, et al (2002) BLEU: a Method for Automatic Evaluation of Machine Translation. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*
- [55] Pipitone N, Alami GH (2024) LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain. *arXiv:2408.10343*
- [56] Qdrant (2024) Qdrant: Vector Search Engine. <https://github.com/qdrant/qdrant>
- [57] Raina V, Liusie A, Gales M (2024) Is LLM-as-a-Judge Robust? Investigating Universal Adversarial Attacks on Zero-shot LLM Assessment. *arXiv:2402.14016*
- [58] Reiter E (2018) A Structured Review of the Validity of BLEU. *Computational Linguistics* 44(3):393–401
- [59] Saad-Falcon J, Khattab O, Potts C, et al (2024) ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*
- [60] Sainz O, Campos JA, García-Ferrero I, et al (2023) NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In: *Findings of the Association for Computational Linguistics: EMNLP*
- [61] Snell C, Lee J, Xu K, et al (2025) Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In: *International Conference on Learning Representations*
- [62] Şulea OM, Zampieri M, Vela M, et al (2017) Predicting the law area and decisions of French supreme court cases. *arXiv:1708.01681*
- [63] Tan M, dos Santos C, Xiang B, et al (2016) Improved Representation Learning for Question Answer Matching. In: *Proceedings of the Annual Meeting of the*

- [64] Tan S, Zhuang S, Montgomery K, et al (2025) JudgeBench: A Benchmark for Evaluating LLM-based Judges. In: International Conference on Learning Representations
- [65] Tay W, Joshi A, Zhang X, et al (2019) Red-faced ROUGE: Examining the Suitability of ROUGE for Opinion Summary Evaluation. In: Proceedings of the Annual Workshop of the Australasian Language Technology Association
- [66] Thirunavukarasu AJ, Ting DSJ, Elangovan K, et al (2023) Large language models in medicine. *Nature Medicine* 29(8):1930–1940
- [67] Verga P, Hofstatter S, Althammer S, et al (2024) Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. *arXiv:2404.18796*
- [68] Wang P, Li L, Chen L, et al (2023) Large Language Models are not Fair Evaluators. *arXiv:2305.17926*
- [69] Wei J, Wang X, Schuurmans D, et al (2022) Chain-of-thought prompting elicits reasoning in large language models. In: *Advances in Neural Information Processing Systems*
- [70] Wu M, Aji AF (2023) Style over Substance: Evaluation Biases for Large Language Models. *arXiv:2307.03025*
- [71] Ye J, Wang Y, Huang Y, et al (2025) Justice or Prejudice? Quantifying Biases in LLM-as-a-Judge. In: International Conference on Learning Representations
- [72] Zhang T, Kishore V, Wu F, et al (2020) BERTScore: Evaluating text generation with BERT. In: International Conference on Learning Representations
- [73] Zheng L, Chiang WL, Sheng Y, et al (2023) Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In: *Advances in Neural Information Processing Systems*

Appendix A Judge Prompt Templates

This appendix provides the complete prompt templates used for LLM judge evaluation. The design rationale is described in [Section 4.4](#).

Each template is shown as a system block stacked on the user block submitted to the judge. As motivated in [Section 4.1](#), the instructions are in English while the evaluated content is in French. The system prompt is identical across both conditions; the user prompt differs only in whether the corpus reference answer and expert justification are supplied.

A.1 Judge Prompt, Condition A (Without Reference)

```
<<system>> You are an expert evaluator specializing in Quebec automobile
insurance law and regulation. Your task is to determine whether a
candidate's answer to an insurance exam question is factually correct.

You must evaluate the candidate's answer strictly based on Quebec insurance
law, the Civil Code of Quebec, and the regulatory framework of the Autorité
des marchés financiers (AMF).

IMPORTANT:
• A candidate answer may sound plausible but contain subtle legal
inaccuracies.
• Pay close attention to legal nuances: specific terminology, exceptions,
conditions, and distinctions (e.g., "automobile" vs "véhicule", "préjudice
corporel" vs "préjudice matériel").
• Do not be lenient. In insurance law, small errors can have significant
legal and financial consequences.

Respond with a JSON object containing exactly two
fields: "verdict" ("correct" or "incorrect") and
"reasoning" (a brief 2--3 sentence justification).
```

```
<<user>> ## Question d'examen
{question}

## Réponse candidate
{candidate_answer}

## Votre évaluation
Is the candidate's answer factually correct for this
insurance exam question? Respond with JSON only:
{"verdict": "correct" or "incorrect", "reasoning": "...}"
```

A.2 Judge Prompt, Condition B (With Reference)

In Condition B, the system prompt remains unchanged; the user prompt additionally includes the corpus reference answer and an expert justification, enabling comparison between the RAG-generated answer and the known correct answer.

«system» You are an expert evaluator specializing in Quebec automobile insurance law and regulation. Your task is to determine whether a candidate's answer to an insurance exam question is factually correct. You must evaluate the candidate's answer strictly based on Quebec insurance law, the Civil Code of Quebec, and the regulatory framework of the Autorité des marchés financiers (AMF).

IMPORTANT:

- A candidate answer may sound plausible but contain subtle legal inaccuracies.
- Pay close attention to legal nuances: specific terminology, exceptions, conditions, and distinctions.
- Do not be lenient. In insurance law, small errors can have significant legal and financial consequences.

Respond with a JSON object containing exactly two fields: "verdict" ("correct" or "incorrect") and "reasoning" (a brief 2--3 sentence justification).

«user» ## Question d'examen

{question}

Réponse candidate

{candidate_answer}

Réponse de référence

{reference_answer}

Justification de référence

{justification}

Votre évaluation

Based on the reference answer and justification, is the candidate's answer factually correct for this insurance exam question? Respond with JSON only: {"verdict": "correct" or "incorrect", "reasoning": "..."}

Appendix B RAG Generation Prompt

The following prompt is used by every judge model acting as a RAG generator to produce free-text answers for Condition B. The {context} placeholder is replaced by the top-*k* retrieved documents, and {question} by the AEPC-QA question.

```
<<system>> You are an expert in Quebec automobile insurance law
and regulation. Answer the following exam question accurately
and in detail, based on the provided reference documents.
```

```
<<user>> ## Reference documents
{context}

## Exam question
{question}

## Instructions
• Answer in French.
• Provide a detailed, well-justified response based on the reference
documents above.
• Cite specific legal provisions, articles, or regulatory references when
applicable.
• Do not mention multiple-choice options or letters (a, b, c, d).
• If the reference documents do not contain enough information to answer
the question fully, state what you can determine from the available sources
and note any gaps.
• Structure your response clearly with the key legal principles
first, followed by their application to the question.
```

Appendix C Reproducibility: Software Environment and Model Identifiers

This appendix records the exact software stack and model identifiers needed to reproduce the open-source half of the pipeline and to query the same proprietary endpoints we used.

Software Environment.

All experiments were run on Python 3.11. Open-source judges were served with vLLM 0.23.0 on NVIDIA RTX 6000 ADA 48 GB GPUs, with constrained JSON decoding through vLLM’s guided-decoding backend ([xgrammar](#) 0.2.2; [lm-format-enforcer](#) 0.11.3) so that local outputs conform to the binary `{score,reason}` schema during decoding rather than being repaired afterwards; this is essential for the quantized checkpoints, whose unconstrained generation otherwise produces frequent parsing failures. API-served judges were queried through LiteLLM 1.82.0 [14], which provides a uniform OpenAI-compatible interface across providers. Verdict parsing builds on [DeepEval](#) 3.8.8; retrieval uses [qdrant-client](#) 1.15.1 over a [Qdrant](#) [56] index of the QAIERC-V2 corpus, which is chunked with overlap, queried by hybrid dense-sparse search, and the top-4 passages per question are concatenated into the generation context; metrics and statistical tests use [scikit-learn](#) 1.8.0 and [statsmodels](#) 0.14.6 (with [torch](#) 2.10.0 and [transformers](#) 4.49.0 underneath vLLM). All random seeds are pinned. Fold assignment is deterministic (a question’s corpus index modulo ten), and we fix a per-fold seed schedule, $\text{seed}_k = 42 + k$ for fold k , giving the ten seeds

42, 43, ..., 51; the bootstrap resampling uses the base seed 42. Judges decode greedily (`do_sample=False`, temperature 0), so re-runs are near-deterministic on identical hardware and weights, up to minor floating-point and batching nondeterminism. Local generation is capped at 1,024 output tokens. A seed-robustness check is reported in [Appendix D](#).

Hardware and Cost.

The open-source judges run on a single owned workstation with three NVIDIA RTX 6000 ADA 48 GB GPUs, with the largest models served tensor-parallel across two of them. Hence, the open subset is fully reproducible at no per-query cost beyond electricity. The API-served judges are billed per token and queried through their providers' APIs, using the asynchronous Batch API where available. Measured latency and per-evaluation cost are summarized in the Discussion.

Reproducibility Checklist.

For the open-source half of the study, every component needed to reproduce the results is specified: the exact model weights (HuggingFace checkpoints, [Table C1](#)), the serving stack and versions (above), the decoding configuration (greedy, constrained `{score, reason}` schema, 1,024-token cap), the data partitioning (deterministic ten-fold assignment), the random seeds (per-fold schedule 42–51; bootstrap seed 42), all evaluation prompts ([Appendix A](#)), and the metric and statistical-test code. The only components that are not indefinitely reproducible are the API-served judges, served through versioned provider endpoints, and the benchmark itself, withheld for licensing; both are documented so that the open subset reproduces exactly and the API-served subset can be re-queried against the same model strings ([Table C2](#)).

Model Identifiers.

[Table C1](#) lists the [HuggingFace](#) repository for every open-source judge (the precise quantized checkpoint, where applicable), and [Table C2](#) lists the provider model string queried for every API-served judge. Reasoning-enabled and standard configurations of the same model share an identifier and differ only in the provider's thinking control. Provider endpoints are versioned and may change server-side over time (see [Limitations](#)); the open-source identifiers and the pinned stack above make that subset fully and indefinitely reproducible.

Appendix D Ablation Studies and Robustness Checks

This appendix details the ablations and robustness checks summarized in [Section 5.1](#) and [Section 5.4](#).

Robustness to Abstention.

Because UNPARSEABLE verdicts are excluded from the headline metrics, a judge that abstains on hard cases could look artificially reliable, and abstention is indeed

Table C1 Open-source judges: exact HuggingFace model identifiers (FP8 / MXFP4 / native-precision checkpoints served with vLLM).

Judge	HuggingFace identifier
DeepSeek-R1-Distill-70B	RedHatAI/DeepSeek-R1-Distill-Llama-70B-FP8-dynamic
Llama-3.3-70B	RedHatAI/Llama-3.3-70B-Instruct-FP8-dynamic
Qwen2.5-72B	RedHatAI/Qwen2.5-72B-Instruct-FP8-dynamic
gpt-oss-120b	openai/gpt-oss-120b
QwQ-32B	RedHatAI/QwQ-32B-FP8-dynamic
Qwen3-32B	Qwen/Qwen3-32B-FP8
Qwen3-30B-A3B	Qwen/Qwen3-30B-A3B-FP8
Gemma-3-27B	RedHatAI/gemma-3-27b-it-FP8-dynamic
CompassJugder-32B	opencompass/CompassJugder-1-32B-Instruct
gpt-oss-20b	openai/gpt-oss-20b
Phi-4	RedHatAI/phi-4-FP8-dynamic
Selene-Mini-8B	AtlaAI/Selene-1-Mini-Llama-3.1-8B
Skywork-Critic-8B	Skywork/Skywork-Critic-Llama-3.1-8B
CompassJugder-7B	opencompass/CompassJugder-1-7B-Instruct
Prometheus-2-7B	prometheus-eval/prometheus-7b-v2.0
SaulLM-7B (legal)	Equall/Saul-7B-Instruct-v1
Qwen2.5-7B	Qwen/Qwen2.5-7B-Instruct
Flow-Judge-3.8B	flowaicom/Flow-Judge-v0.1
GLIDER-3.8B	PatronusAI/glider
Gemma-3-4B	RedHatAI/gemma-3-4b-it-FP8-dynamic
Qwen2.5-3B	Qwen/Qwen2.5-3B-Instruct
Llama-3.2-3B	RedHatAI/Llama-3.2-3B-Instruct-FP8-dynamic
Qwen2.5-1.5B	Qwen/Qwen2.5-1.5B-Instruct
CompassJugder-1.5B	opencompass/CompassJugder-1-1.5B-Instruct
Llama-3.2-1B	RedHatAI/Llama-3.2-1B-Instruct-FP8-dynamic
Qwen2.5-0.5B	Qwen/Qwen2.5-0.5B-Instruct

mildly correlated with MCC across judges, with the top judge abstaining the most: **Gemini-3.1-Pro** at 6.8%, against under 4% for the rest. To check that this is not driving the ranking, we recompute MCC scoring every abstention as “incorrect”, the worst case for the judge. The order is unchanged, **Gemini-3.1-Pro** still leading, its MCC slipping only from 0.65 to 0.61, so abstention narrows the leader’s margin slightly but overturns no conclusion.

Self-Preference Under a Controlled Base Rate.

To remove the base-rate confound, the $N \times N$ design of [Section 4.3](#) lets us compare, *within* Condition B, each generator-judge’s “correct” rate on its *own* answers versus on the other generators’, holding provenance and the candidate pool fixed. Self-preference is essentially nil on average, with a mean $\Delta = +0.01$ that is positive for only half the judges: individual models lean slightly for or against their own outputs, but the effect cancels across the roster. So in this strict legal domain we find *no* self-enhancement bias [\[53, 57\]](#), consistent with the broader observation that evaluators can favour human over machine text [\[35\]](#).

Reference-Augmented Prompting.

We repeat Condition A on the 26 open-source judges with the corpus reference answer and expert justification added to the prompt; this ablation is reported on the open-source subset. The effect is large and consistent: mean MCC jumps from 0.12 to 0.67 and FPR falls sharply, with almost every judge improving, and the gain is most dramatic for the smallest models, which go from near-baseline to near-oracle. The

Table C2 API-served judges: provider model strings (queried through LiteLLM). The `openrouter/` prefix denotes models reached through the OpenRouter gateway.

Judge	Provider model string
Claude-Opus-4.8(-reason)	anthropic/claude-opus-4-8
Claude-Sonnet-4.6(-reason)	anthropic/claude-sonnet-4-6
Claude-Haiku-4.5(-reason)	anthropic/claude-haiku-4-5-20251001
GPT-5.1-reason	openai/gpt-5.1
GPT-5-mini(-reason)	openai/gpt-5-mini
Magistral-Medium-reason	mistral/magistral-medium-latest
Mistral-Medium	mistral/mistral-medium-latest
DeepSeek-Reasoner	deepseek/deepseek-reasoner
DeepSeek-Chat	deepseek/deepseek-chat
Grok-4-reason	xai/grok-4
Grok-3-mini-reason	xai/grok-3-mini
Gemini-3.1-Pro	openrouter/google/gemini-3.1-pro-preview
Gemini-2.5-Flash	openrouter/google/gemini-2.5-flash
Qwen3-235B-thinking	openrouter/qwen/qwen3-235b-a22b-thinking-2507
Qwen3-235B	openrouter/qwen/qwen3-235b-a22b
GLM-4.6	openrouter/z-ai/glm-4.6
Kimi-K2	openrouter/moonshotai/kimi-k2
Command-A	cohere/command-a-03-2025
Command-R-Plus	cohere/command-r-plus-08-2024
Command-R	cohere/command-r-08-2024
Command-R7B	cohere/command-r7b-12-2024
Aya-Expanse-32B	cohere/c4ai-aya-expanse-32b

interpretation is direct: without a reference, a judge is limited by their own knowledge of Quebec insurance law, which is often weak; given the reference, the task collapses to comparing the candidate against a supplied answer, which even small models do well. For MCQ candidates, this is an upper bound, since the supplied reference is close to the gold answer. To test whether the gain is merely this matching artifact, we repeat the ablation on free-text Condition B answers, judging 648 matched answers both with and without the reference across eleven judges that span frontier proprietary, small open-source, and evaluation-specialized models. On free text the reference still rewrites a large share of verdicts for exactly the small and specialized judges that the MCQ ablation identified as benefiting most: **Qwen2.5-7B**, **Prometheus-2-7B**, and **Flow-Judge-3.8B** change between a fifth and a third of their verdicts, all with McNemar $p < 0.001$, whereas the frontier judges are largely insensitive, changing only a few percent, with no significant shift for **Gemini-3.1-Pro**, **Gemini-2.5-Flash**, **Claude-Haiku-4.5**, or **DeepSeek-Chat**. The reference consistently moves judges toward stricter verdicts and raises inter-judge agreement: among these eleven judges, adding it lifts Fleiss’ κ from 0.13 to 0.20.

Seed Robustness.

To confirm the confidence intervals are a property of the data rather than of the seed, we recompute every interval under all ten seeds; across all judges, the interval bounds move by at most 0.015 for MCC and 0.6 pp for FPR, so the reported intervals are seed-stable.

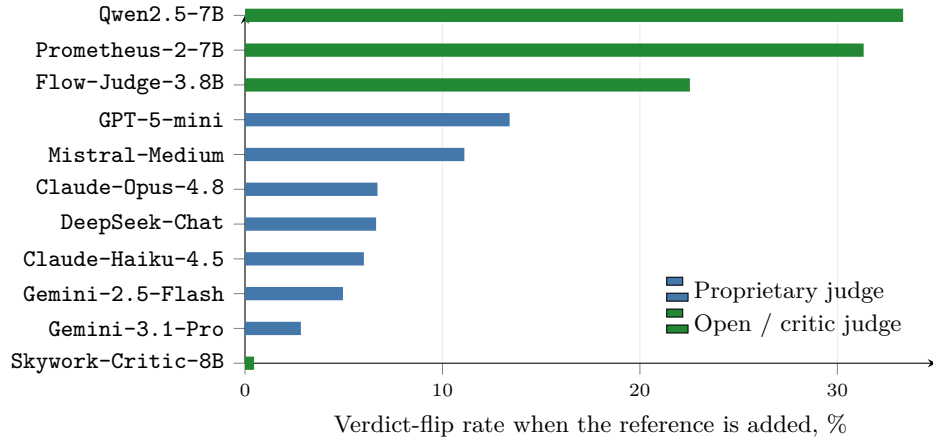


Figure D1 Effect of adding the corpus reference on free-text Condition B verdicts, on 648 matched answers. The bar is the share of a judge’s verdicts that *flip* when the reference is supplied, i.e. its sensitivity to the reference, not an accuracy gain. Small open-source and evaluation-specialized judges flip a fifth to a third of their verdicts, all with McNemar $p < 0.001$, whereas frontier proprietary judges are largely insensitive. The reference helps most exactly the judges that need it.

Appendix E Full Per-Judge Reliability

Figure E2 gives the MCC of every judge and reference baseline on Condition A, ordered by MCC; Table 6 lists only a representative subset.

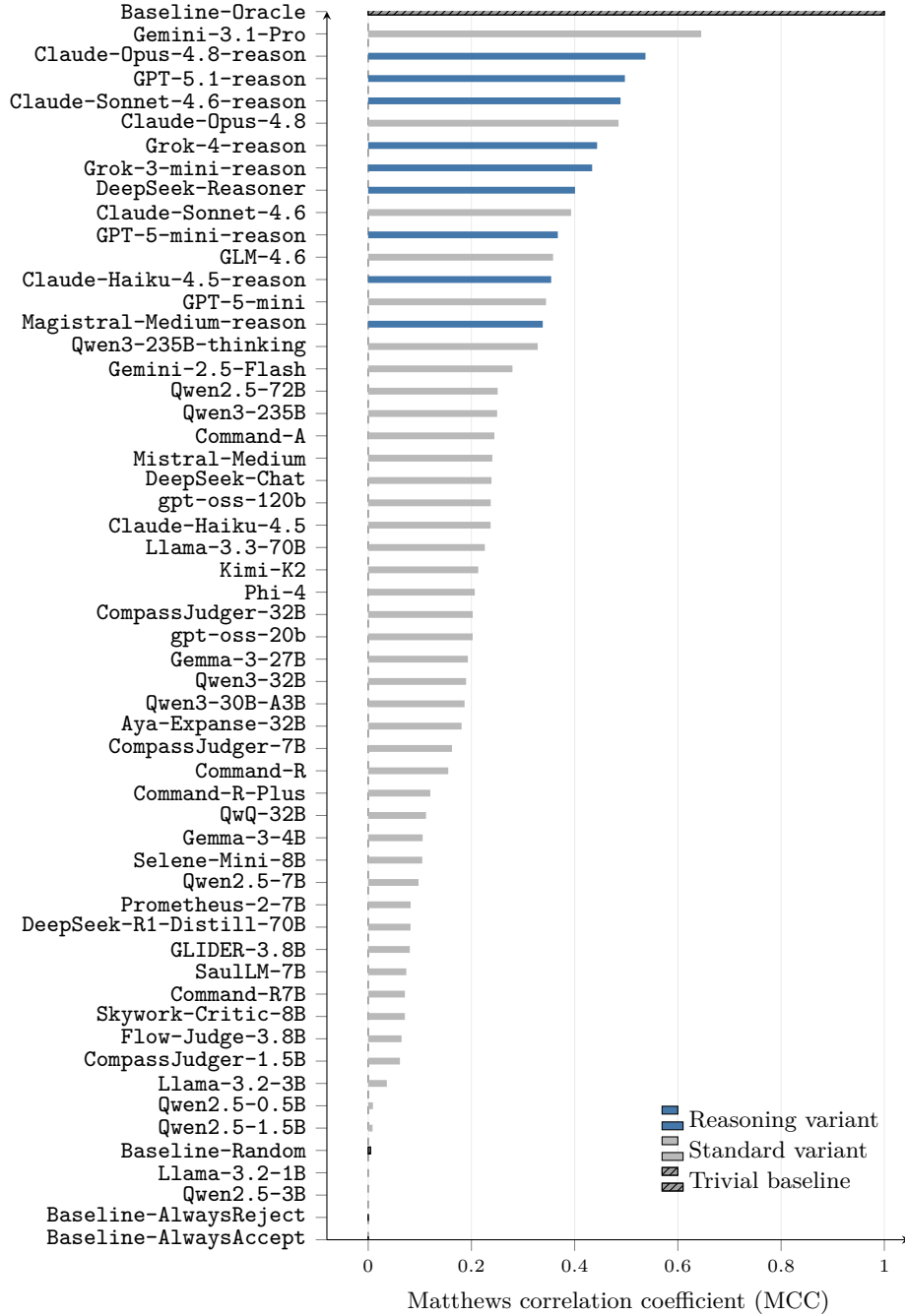


Figure E2 MCC for all 52 judges and the four reference baselines on Condition A (mean over 10 folds), ordered by MCC. Colour distinguishes reasoning from standard configurations; trivial baselines are hatched. The dashed line marks chance; the oracle attains 1.0. Reasoning variants concentrate in the upper tier.