Fraunhofer IESE

Fraunhofer-Institut für Experimentelles Software Engineering IESE

LLM-as-a-Judge for assessing maintainability. First results with a causal approach.

Julien Siebert. AI4SE Workshop, KI2026, 12.08.2026

Motivation

AI coding assistants are producing increasing amounts of software artefacts (code, requirements, tests, docs…).

Whereas some aspects like functional correctness can be checked automatically (e.g. through tests), other aspect like maintainability, readability, or clarity are harder to measure.

Possibilities:

- **Human assessment: good quality (expected), expensive and not automatable.**
- **Syntactic (e.g., ROUGE, BLEU) and Semantic similarity metrics: automatable but poor quality (and reliance on a ground truth).**
- **LLM-as-a-Judge: potentially good quality and automatable.**

LLM-as-a-Judge

Problem

Objective:

- How good are LLM-as-a-Judge?
- What influences the quality of the agreement between models and humans?
- What can we do in order to improve the agreement quality?

Challenges:

- Existing studies evaluate different models on different benchmarks: very small overlap (if ever!),
- Models evaluated are quickly obsolete,
- Benchmarks might be too easy to distinguish between different model size.
- Most studies focus on correlation and might not distinguish causal effects from spurious correlations (confounding, collider, selection biases). For example, model size may be associated with quantization or architecture.

Can we still achieve our objective despite the challenges?

Our Approach: Causal Inference

1. **Modelling: explicit causal assumptions in a causal graph (DAG).**
2. **Identification: Use the causal graph to determine whether effects can be estimated from data.**
3. **Estimation: estimate the causal effect (statistical methods, causal ML).**
4. **Refutation: Try to falsify the causal effect.**

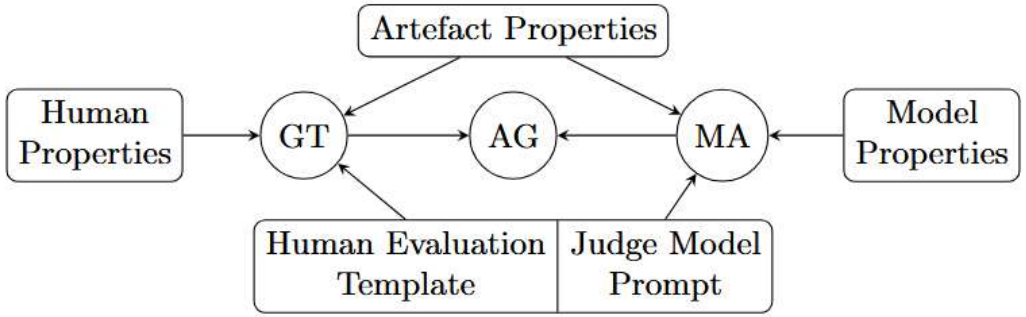


Fig. 1: High level overview of aspects influencing LLM-as-a-Judge agreement quality. **GT**: Ground Truth, **MA**: Model Answer, **AG**: Agreement.

Pearl, J., & Mackenzie, D. (2018). The book of why. Basic Books
Huntington-Klein, N. (2021). The effect: An introduction to research design and causality. Chapman and Hall/CRC.

Research Questions

RQ1: What is the effect of model properties on agreement quality?

RQ2: What is the effect of prompt properties on agreement quality?

RQ3: What is the effect of code characteristics on agreement quality?

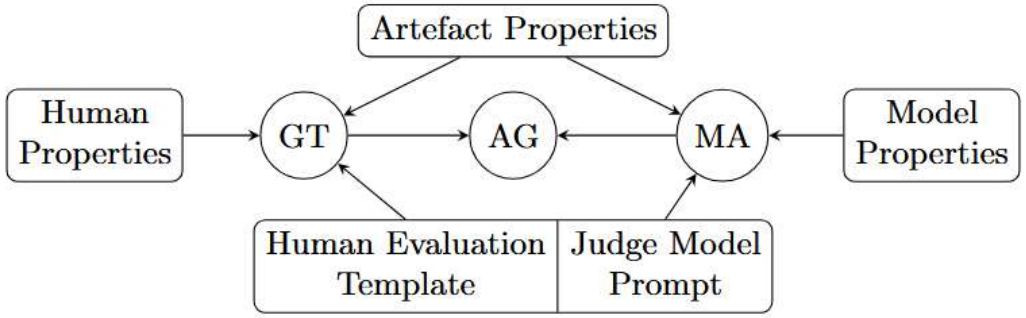


Fig. 1: High level overview of aspects influencing LLM-as-a-Judge agreement quality. **GT**: Ground Truth, **MA**: Model Answer, **AG**: Agreement.

Dataset

Schnappinger et al.

Dataset contains 304 Java classes from five projects (ArgoUML, AOI, Diary Management, JSweet, JUnit 4).

Classes were assessed by multiple human experts on five dimensions:

- **Readability (Rd, “this code is easy to read”),**
- **Understandability (Ud, “the semantic meaning of this code is clear”),**
- **Complexity (Cx, “this code is complex”),**
- **Modularity (Md, “this code should be broken down into smaller sections”),**
- **and Overall maintainability (Ov, “overall, this code is maintainable”)**

Expert answers (strongly-agree, weakly-agree, weakly-disagree, strongly-disagree) were aggregated to get a probability.

	Ov.	Rd.	Ud.	Cx.	Md.
strongly agree	174	183	157	22	29
weakly agree	64	79	76	41	31
weakly disagree	41	38	51	60	49
strongly disagree	25	4	20	181	195

Number of time the probability a given answer (strongly agree, weakly agree, weakly disagree, strongly disagree) is maximum (≥ 0.5).

Schnappinger, M., Fietzke, A., & Pretschner, A. (2020, September). Defining a software maintainability dataset: collecting, aggregating and analysing expert evaluations of software maintainability. In 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 278-289). IEEE.
Schnappinger, M., Fietzke, A., Pretschner, A.: A software maintainability dataset (Sep 2020). https://doi.org/10.6084/m9.figshare.12801215

Experimental Setup

Prompt:

- Code = java code (no preprocessing).
- Question = same questions as the dataset.

Hardware / Software:

- NVIDIA DGX B300 system.
- Ollama (1 GPU), vLLM (2 GPUs).

```
Here is some code:
{code}

Question: {question}

Please provide your answer and a brief explanation.
Answer with exactly one of these options: strongly agree, weakly agree, weakly disagree,
strongly disagree

Format your response as JSON as follows:
{
  "answer": "your answer",
  "explanation": "your explanation"
}
```

Fig. 2: Prompt used for the experiments.

Name	Size	Quant.	Arch.	Server
Qwen3.5-0.8B	0.87B	Q8_0	Dense	Ollama
Qwen3.5-2B	2.27B	Q8_0	Dense	Ollama
Qwen3.5-4B	4.66B	Q4_K_M	Dense	Ollama
Qwen3.5-9B	9.65B	Q4_K_M	Dense	Ollama
Qwen3.5-35B-A3B	36B (3B active)	Q4_K_M	MoE	Ollama
Qwen3.5-122B-A10B	125B (10B active)	Q4_K_M	MoE	Ollama
Qwen3.5-397B-A17B	397B (17B active)	NVFP4	MoE	vLLM

Table 2: Models used. All models were run with temperature 0.6 and 1.

Metrics

We measure agreement between the reference answer (i.e., the answer in the ground truth dataset with the maximum probability) and the LLM's answer using four categories:

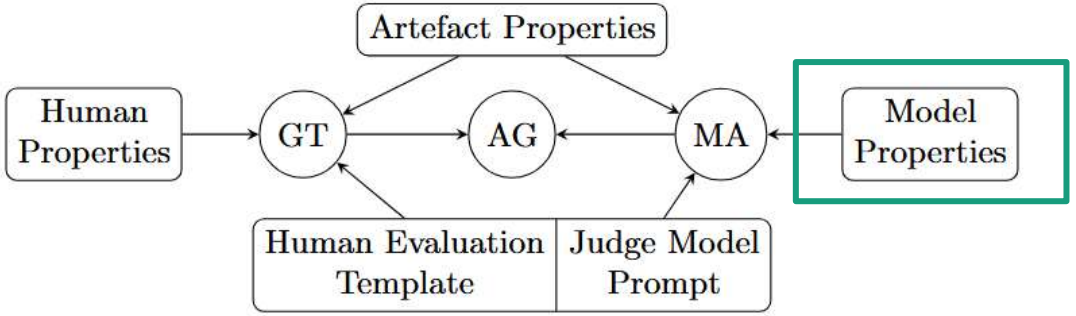
- **Exact match: Both answers are identical.**
- **Partial match: Both answers fall on the same side of the scale but differ in intensity. Partial agree (resp. disagree) means that one answer was strongly agree and the other weakly agree (resp. strongly disagree and weakly disagree).**
- **Wrong direction: The answers fall on opposite sides of the scale (e.g., one indicates agreement while the other indicates disagreement).**

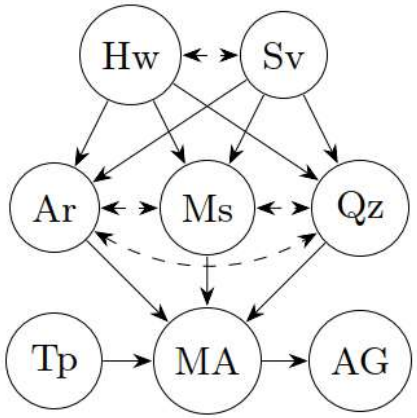
Baseline (uniformly distrib.): Wrong direction 50%, Exact match 25%, Partial match (25%)

Human	LLM	Categories
Strongly agree	Strongly agree	Exact Match (4)
Weakly agree	Weakly agree	
Weakly disagree	Weakly disagree	
Strongly disagree	Strongly disagree	
Strongly agree	Weakly agree	Partial agree (2)
Weakly agree	Strongly agree	
Weakly disagree	Strongly disagree	Partial disagree (2)
Strongly disagree	Weakly disagree	
Strongly agree	Strongly disagree	Wrong direction (8)
Strongly agree	Weakly disagree	
Weakly agree	Strongly disagree	
Weakly agree	Weakly disagree	
Weakly disagree	Strongly agree	
Weakly disagree	Weakly agree	
Strongly disagree	Strongly agree	
Strongly disagree	Weakly agree	

RQ1: Effect of Model Properties

Causal Assumptions





(a) Subgraph for model properties (RQ1). **Hw**: Hardware, **Sv**: Server, **Ms**: Model size, **Qz**: Quantization, **Ar**: Model architecture, **Tp**: Temperature. Dashed double arrows represent latent confounder.

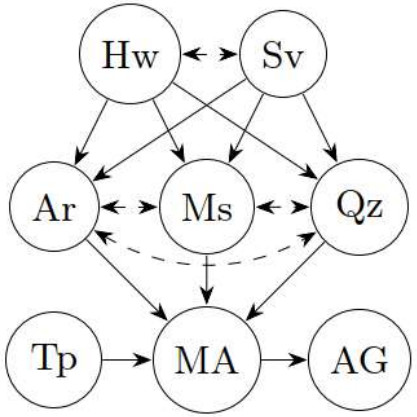
RQ1: Effect of Model Properties

Identification

Temperature (Tp) is not confounded (correlation = causation).

Model size (Ms), architecture (Ar) and quantisation (Qz) are all covariates. To estimate the effect of one, the other two must be controlled for.

We only have enough data for estimating the effect of model Size.



(a) Subgraph for model properties (RQ1). **Hw**: Hardware, **Sv**: Server, **Ms**: Model size, **Qz**: Quantization, **Ar**: Model architecture, **Tp**: Temperature. Dashed double arrows represent latent confounder.

Name	Size	Quant.	Arch.	Server
Qwen3.5-0.8B	0.87B	Q8_0	Dense	Ollama
Qwen3.5-2B	2.27B	Q8_0	Dense	Ollama
Qwen3.5-4B	4.66B	Q4_K_M	Dense	Ollama
Qwen3.5-9B	9.65B	Q4_K_M	Dense	Ollama
Qwen3.5-35B-A3B	36B (3B active)	Q4_K_M	MoE	Ollama
Qwen3.5-122B-A10B	125B (10B active)	Q4_K_M	MoE	Ollama
Qwen3.5-397B-A17B	397B (17B active)	NVFP4	MoE	vLLM

Table 2: Models used. All models were run with temperature 0.6 and 1.

RQ1 Results

Estimation

1. **Temperature:** **No significant effect (permutation test Mean(Tp=0.6) – Mean(Tp=1.0))**

2. **Model size (permutation tests):**

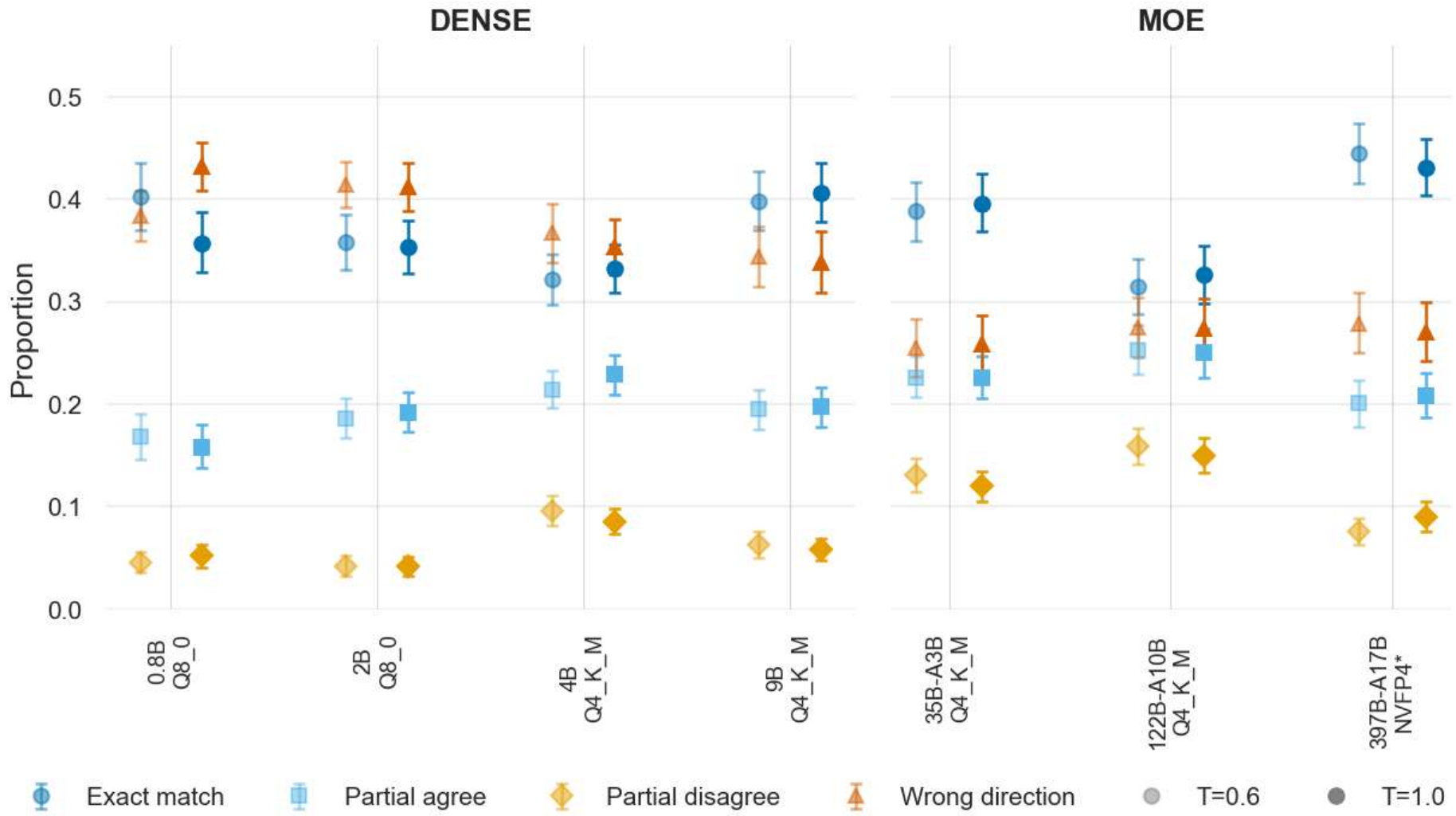
- **0.8B vs 2B (Dense, Q8_0): no significant differences.**
- **4B vs 9B (Dense, Q4_K_M); 9B outperformed 4B in exact matches (Δ = +7.6%, p_fdr < 0.01)**
- **35B vs 122B (MoE, Q4_K_M). 35B outperformed 122B in exact matches (Δ = +7.4%, p_fdr < 0.01)**

RQ1: Results

wrong direction
baseline: 0.5

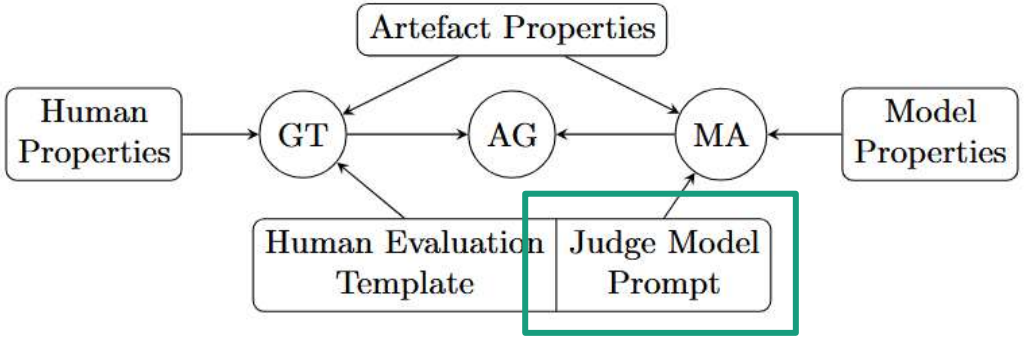
Exact match
baseline: 0.25

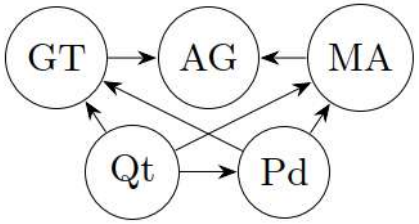
Partial match
baseline: 0.125



RQ2: Effect of Prompt Properties

Causal Assumptions





(b) Subgraph for prompt properties (RQ2). **Qt**: Question type, **Pd**: Phrasing direction

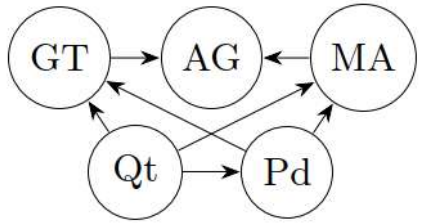
Fraunhofer IESE

RQ2: Effect of Prompt Properties

Identification

Question type (Qt) is not confounded (correlation = causation)

Phrasing direction (Pd) is confounded by question type but since there is only one phrasing direction per question type, we cannot estimate the effect of Pd (no variance left after controlling).



(b) Subgraph for prompt properties (RQ2). **Qt**: Question type, **Pd**: Phrasing direction

Question Types (Qt)	Pd.
Rd. This code is easy to read	+
Ud. The semantic meaning of this code is clear	+
Cx. This code is too complex	-
Md. This code should be broken down into smaller sections	-
Ov. Overall, this code is maintainable	+

RQ2: Results

Estimation

Effect of question type:

For each pair of question types (qa, qb), the causal effect is defined as the difference in conditional probability distributions: $\Delta = P(AG \mid Qt = qa) - P(AG \mid Qt = qb)$.

Statistical significance was assessed using a permutation test (n=10000) under Fisher's sharp null hypothesis of zero individual treatment effects ($p < 0.05$).

Our results (see next slide) show that some question formulations produce more aligned judgments than others:

- **compared to other types of questions Rd. and Ud. tend to produce more aligned results,**
- **Md. and Ov. tends to decrease agreement,**
- **Cx. increase exact match and decreases wrong direction but decreases partial agree and increases partial disagree.**

RQ2: Results

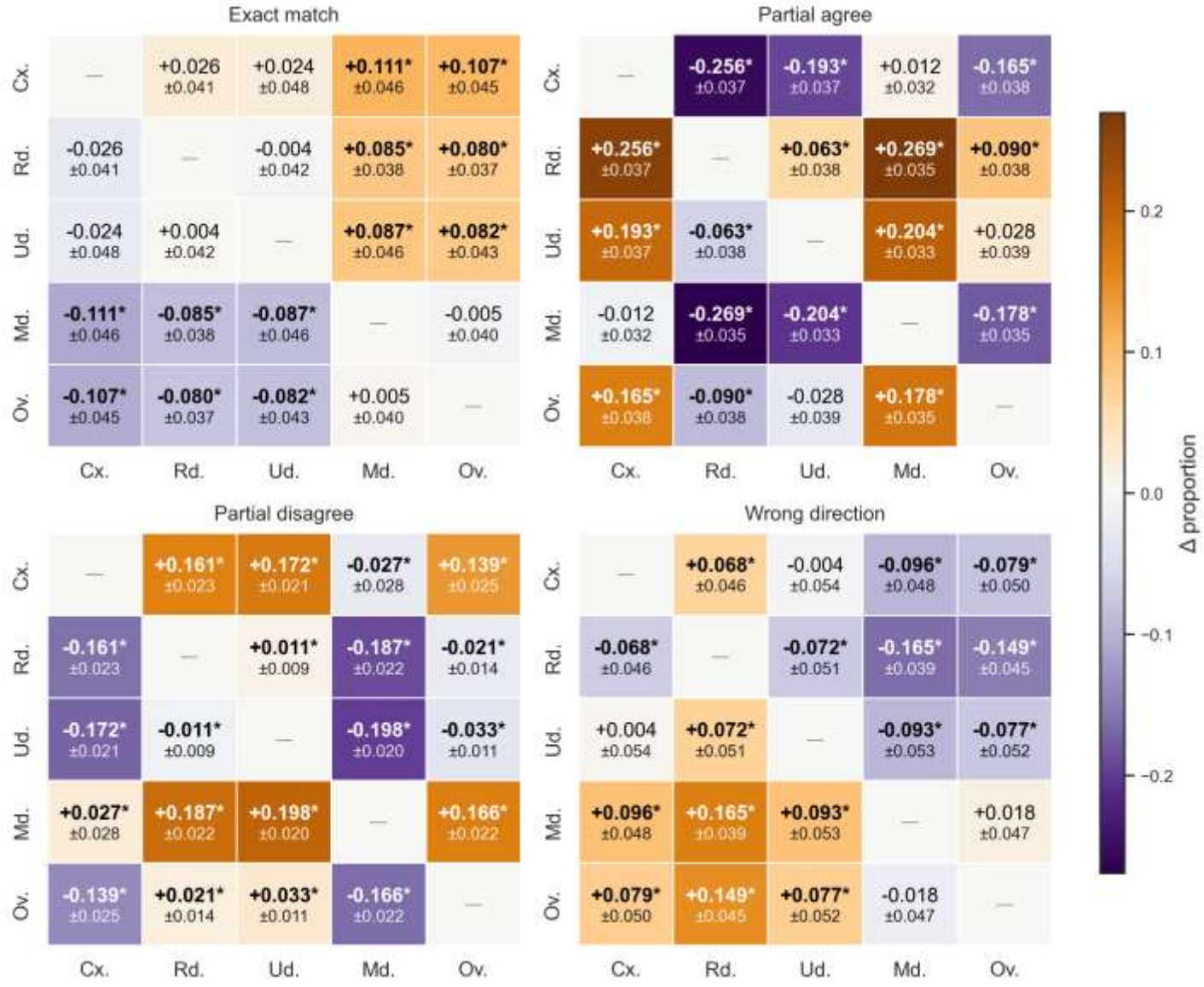
Effects of question type on agreement.

Cells show the difference in proportions
Δ = P (AG | row)–P (AG | column),
with 95% bootstrap CI.

Asterisks indicate FDR-corrected significance (p < 0.05).

Positive Δ (orange) means the row question type yields higher exact match (top left), partial agree (top right), partial disagree (bottom left), or wrong directions (bottom right) than the column.

Orange in top panel left (resp. purple in lower panel right) show better agreement.

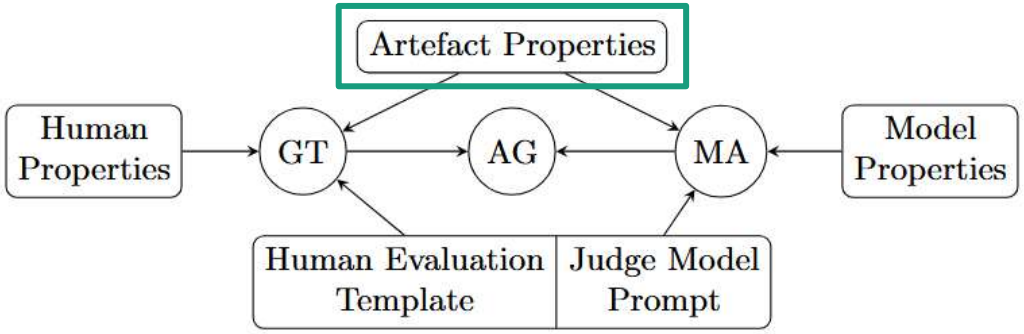


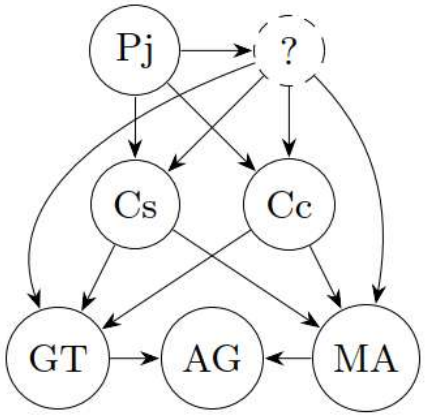
RQ3: Effects of Code Properties

Causal Assumptions

Code size (Cs) is measured as lies of code (LOC)

Code complexity (Cc) is measured as response for class (RFC)





(c) Subgraph for artefact properties (RQ3). **Cs**: Code size, **Cc**: Code complexity, **Pj**: Project, **?**: within-project latent confounders.

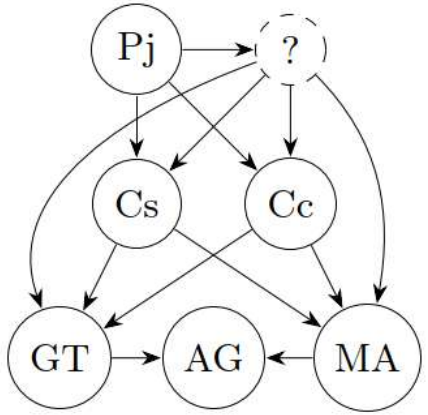
RQ3: Effects of Code Properties

Identification

To estimate the effect of code size (resp. complexity), we controlled for project and complexity (resp. size).

This leaves within-project confounding unaddressed.

To assess robustness to unmeasured confounding, we computed E-values. (E-values answers the question "How strong would a hidden confounder need to be to completely explain away the observed effect?“)



(c) Subgraph for artefact properties (RQ3). **Cs**: Code size, **Cc**: Code complexity, **Pj**: Project, **?**: within-project latent confounders.

RQ3 Results
Estimation

Effect of code size (controlling for complexity):

- **an increase of 216 lines of code (1 SD LOC) implies 17% lower odds of wrong prediction vs exact match.**

Effect of code complexity (controlling for size):

- **an increase in 49 more methods (1 SD RFC) implies 54% higher odds of wrong prediction vs exact match.**

RQ3 Results

Estimation

Predictor	Outcome	OR	95% CI	E-value	p-value
LOC (per 216 lines)	partial_agree	0.771	0.707–0.842	1.92	$<0.001^{***}$
LOC (per 216 lines)	partial_disagree	0.979	0.880–1.088	1.17	0.688
LOC (per 216 lines)	wrong_direction	0.834	0.783–0.888	1.69	$<0.001^{***}$
RFC (per 49 methods)	partial_agree	1.311	1.204–1.428	1.95	$<0.001^{***}$
RFC (per 49 methods)	partial_disagree	0.964	0.856–1.086	1.23	0.549
RFC (per 49 methods)	wrong_direction	1.539	1.441–1.644	2.45	$<0.001^{***}$

Table 3: Effect of code properties on agreement. Multinomial logistic regression with project fixed effects (N=21,280 predictions, 304 files). Reference category: *exact_match*. Predictors standardized (z-scored); effect sizes represent per 1 SD increase (LOC: 216 lines; RFC: 49 methods). E-value quantifies robustness to unmeasured confounding (higher = more robust). Significance: *** $p < 0.001$.

Limitations and Open Questions

Limitations:

- Only 5 Java projects (304 Java classes).
- Only 7 models, 1 Model family (Qwen-3.5).
- Only one request per question per model per temperature (no sampling).
- Cost: total of 21,280 requests amounting to 114,348,974 tokens (87,653,651 input tokens and 26,695,323 output tokens).
- No refutation tests, yet (slide 5).

Open Questions:

- Temperature effect? Maybe with smaller values?
- Models size: role of quantization?
- Prompt: effect of phrasing, language, order of elements… ?
- Artefacts' properties: is the effect of complexity and size the same with other judge models? Does the same applies to other artefacts types (such as as docs, reqs, etc.)?

Conclusion

To my knowledge: first results using causal inference to evaluate LLM-as-a-Judge systems (for software maintainability).

Making causal assumptions explicit through a DAG helped to find out which factors are identifiable and which are not within the current experimental design.

Causal inference methods (not only *Pearlian* – graph based) have the potential to help overcome some of the empirical challenges listed in slide 4.

Contact

Dr. Julien Siebert

julien.siebert@iese.fraunhofer.de
https://www.iese.fraunhofer.de/blog/author/julien-siebert/
https://de.linkedin.com/in/dr-siebert-julien
https://sites.google.com/view/dr-julien-siebert

Fraunhofer IESE
Data Science Department
Fraunhofer-Platz 1
67663 Kaiserslautern
www.iese.fraunhofer.de

Fraunhofer IESE

Fraunhofer-Institut für Experimentelles Software Engineering IESE