LOST IN FORMULATION: DISENTANGLING AND MEASURING PROMPT SENSITIVITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Language model outputs vary with both the phrasing of a prompt (*prompt formulation*) and the amount of information it provides about the intended question (*prompt specificity*). Separating these sources of sensitivity helps users decide how to rewrite a prompt, but existing prompt sensitivity metrics conflate them. We vary prompt formulation and prompt specificity systematically on questions from the AmbigQA dataset, collect repeated answers from three open-weight models, and compute ten existing metrics on these answers. A principal component analysis separates the metrics into three components, which we call *correctness*, *formulation dependence*, and *output dispersion*. While existing metrics capture correctness and output dispersion well, none captures formulation dependence without also counting decoding noise, the variation among repeated answers to the same prompt. We therefore introduce ρ_F, a new metric based on intraclass correlation of correctness across formulations that subtracts decoding noise. Although it does not enter the decomposition, ρ_F tracks the formulation dependence component more closely than existing metrics, also on held-out AmbigQA questions. Varying specificity and formulation confirms that the three components respond to different changes. Disambiguating a question raises correctness and lowers output dispersion but produces no detectable change in ρ_F, whereas more varied formulations tend to raise ρ_F but not the other two. Correctness, formulation dependence, and output dispersion are thus distinct parts of prompt sensitivity, and ρ_F measures formulation dependence separately from decoding noise.

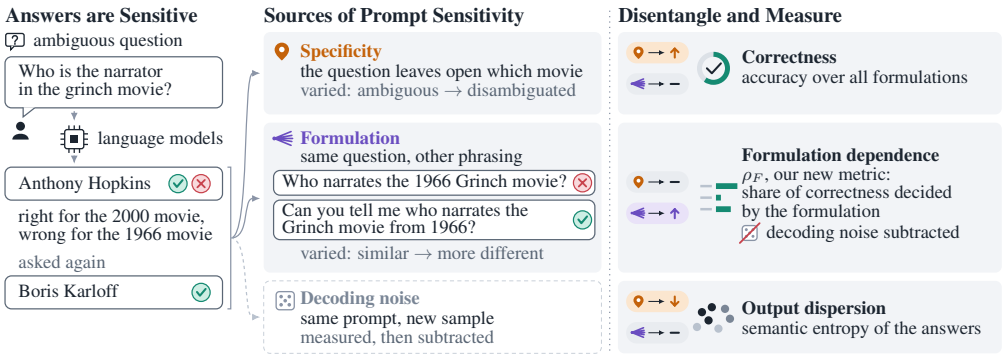

Figure 1: An answer changes with what the question leaves open (specificity), with its phrasing (formulation), and with sampling (decoding noise). We measure three factors, correctness with accuracy, formulation dependence with our metric ρ_F, which subtracts decoding noise, and output dispersion with semantic entropy. Icons show each factor's response to disambiguating the question (pin) and to more varied formulations (fan), an arrow marking a monotone change in all three models and a dash no consistent change (Section 4.3).

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

1 INTRODUCTION

Language models can perform tasks ranging from question answering to code generation and are guided by prompts, which are inputs that state goals, provide context, and set constraints in natural language (Schulhoff et al., 2024). Because every output depends on and is sensitive to its prompt, small changes to it can change the output in unintended ways (Mizrahi et al., 2024; Zhu et al., 2024; Sclar et al., 2024). This prompt sensitivity has become consequential because language models increasingly serve as measurement instruments rather than only as objects of evaluation: they annotate data and judge other models, also inside agentic evaluations (Messing, 2026; Mustahsan et al., 2025). In these roles, prompt sensitivity carries over into the conclusions drawn from their outputs (Messing, 2026), and multi-prompt evaluation is increasingly recommended as a remedy (Mizrahi et al., 2024; Polo et al., 2024).

We distinguish three sources of this prompt sensitivity. First, outputs vary with how much information a prompt gives about what is asked, its *prompt specificity* (Pecher et al., 2026; Keluskar et al., 2024). For example, the question "Who is the narrator in the Grinch movie?" fits three movies with different narrators (Figure 1). The answer "Anthony Hopkins" is correct for the 2000 movie but wrong for the 1966 movie the user meant, because the question does not specify the movie. Second, outputs vary with the phrasing of a prompt, its *prompt formulation*, even when the specificity is fixed (Sclar et al., 2024; Cao et al., 2024). For example, "Who narrates the 1966 Grinch movie?" and "Can you tell me who narrates the Grinch movie from 1966?" ask for the same information but may lead to different answers (Cao et al., 2024). We call the degree to which the correctness of an output depends on the formulation *formulation dependence*. A user who can tell the two sources apart knows what to change, a more specific question in the first case and a different formulation in the second. Third, even a fixed prompt does not fix the output, because language models sample a new output each time (Kuhn et al., 2023; Taparia et al., 2026). We call this variation, which is not a property of the prompt, *decoding noise*.

Existing work measures prompt sensitivity by how the correctness or consistency of outputs varies across prompt variants (Mizrahi et al., 2024; Polo et al., 2024; Sclar et al., 2024), including metrics such as the worst-prompt performance (Cao et al., 2024), correctness or output agreement (Zhuo et al., 2024; Errica et al., 2025), or likelihood divergence (Chatterjee et al., 2024), and work on uncertainty quantification measures how widely repeated outputs scatter for one prompt (Kuhn et al., 2023; Farquhar et al., 2024). These metrics have three shortcomings. First, they score different outcomes. Some ask whether the outputs are correct, others only whether they agree with each other. Second, most compare one output per prompt. Under sampling, this mixes the formulation effect with decoding noise, and under greedy decoding it shows only the most likely output. The few metrics that address decoding noise score answer embeddings (Cox et al., 2025) or numeric responses (Kunievsky & Evans, 2026), which show how much answers change but not whether they stay correct, vary no formulation (Mustahsan et al., 2025), or pool over whole corpora (Messing, 2026; Żatuchin, 2026), which hides which questions are affected. Third, whether these metrics measure one quantity or conflate several has, to our knowledge, not been tested. A single prompt sensitivity metric cannot separate how often the model is correct, how much the formulation matters, and how widely the answers scatter.

To address these shortcomings, we first disentangle what prompt sensitivity metrics measure. We do this by applying ten existing metrics to 89,730 answers of three open-weight models to 150 AmbigQA questions (Min et al., 2020); each question is asked at two levels of specificity (an ambiguous and a disambiguated form) and in up to ten formulations of the same meaning (Section 3). Our analysis shows that existing metrics measure three different things, and none isolates the effect of formulations on correctness from decoding noise. Building on the decomposition results, we introduce ρ_F to measure formulation dependence independently of decoding noise. To evaluate it, we test how well it captures variations in formulations (Section 4).

In this paper, we make four contributions. First, we show that ten prompt sensitivity metrics measure three factors: correctness, formulation dependence, and output dispersion. Accuracy and semantic entropy reflect correctness and output dispersion, while the two formulation dependence metrics agree only weakly and neither measures correctness net of decoding noise. Second, we introduce ρ_F, a per-question metric of formulation dependence that checks correctness and subtracts decoding noise. It represents the formulation dependence component more closely than either existing metric.

Third, we show that the three factors respond to different changes of a prompt. Disambiguating a question raises accuracy by 22 to 25 points and lowers semantic entropy but does not noticeably change ρ_F, whereas formulations that differ more tend to raise ρ_F. Finally, we release a dataset of 259,560 sampled answers from three open-weight models to 7,652 formulations of 250 AmbigQA questions at two specificity levels, with ten answers per formulation, all scored against the intended answer.

2 Related Work

Existing prompt sensitivity metrics differ in what they score, whether they subtract decoding noise, at what level they report sensitivity, and whether their reliability has been analyzed (Table 1). Beyond these metrics, we review how prior work varies specificity and formulation.

Table 1: Comparison of existing prompt sensitivity metrics. Among the approaches compared here, only our ρ_F combines a correctness check, subtraction of decoding noise, per-question estimation, and a reliability analysis.

	What is scored		Formulation and decoding			
Approach	Outcome	Checks correctness	Repeated decoding	Noise subtracted	Reported per	Reliability analysis
Prompt-set spread and extrema (Sclar et al., 2024; Zhu et al., 2024; Cao et al., 2024)	score spread, extrema, drop	◐	○	○	task, item	◐
Agreement and entropy metrics (Zhuo et al., 2024; Errica et al., 2025; Lu et al., 2024)	answer or correctness agreement	◐	○	○	item, prompt	○
POSIX (Chatterjee et al., 2024)	likelihood divergence	○	○	○	prompt set	○
Semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024)	meaning entropy	○	●	○	question	○
ρ_u, embedding variance share (Cox et al., 2025)	answer embedding	○	●	◐	question	○
Brittleness decomposition (Romanou et al., 2026)	binary correctness	●	○	○	model, benchmark	○
Uncertainty decomposition (Taparia et al., 2026)	semantic entropy	○	◐	○	prompt, dataset	○
Intent/articulation decomposition (Kunievsky & Evans, 2026)	numeric response	○	●	●	task, model	◐
Trial-to-trial intraclass correlation (Mustahsan et al., 2025)	binary correctness	●	●	◐	pooled	◐
G-theory formulation facet (Messing, 2026; Żatuchin, 2026)	judge score, correctness, polarity	◐	●	●	corpus, aggregate	◐
This work: three factors, ρ_F	correctness	●	●	●	question, model	●

● holds ◐ partial or some variants ○ absent or unreported. "Noise subtracted" denotes an explicit within-formulation decoding term.

What existing metrics score. Besides benchmark metrics, which check correctness or only agreement (Zhuo et al., 2024; Lu et al., 2024; Errica et al., 2025), decomposition approaches split output variation into parts due to the formulation, the question itself, repeated decoding, or other sources (Cox et al., 2025; Kunievsky & Evans, 2026; Romanou et al., 2026; Taparia et al., 2026; Mustahsan et al., 2025; Messing, 2026; Żatuchin, 2026). They work with semantic entropy, embeddings, numeric responses, single greedy outputs, item-level or pooled correctness, or corpus-level judge or sentiment scores, and none estimates, per question, the share of correctness variance due to the formulation after decoding noise is removed.

Varying specificity and formulation. Underspecified prompts can drive prompt sensitivity (Pecher et al., 2026), and the AmbigQA dataset pairs ambiguous questions with a disambiguated question for each interpretation (Min et al., 2020). Prior studies that vary specificity answer closed-book (Keluskar et al., 2024) or remove task guidance along with detail (Kim, 2025), and neither checks that the same answer stays correct at every level. Beyond varying the number of typos or padding characters (Romanou et al., 2026), we find no prior work that varies how much meaning-preserving rewrites of a question differ. Table 1 shows that no existing approach both checks correctness and subtracts decoding noise per question. Our ρ_F does both, and we analyze its reliability and test it with accuracy and semantic entropy by varying specificity and formulation separately.

3 DISENTANGLING PROMPT SENSITIVITY

This section addresses which factors of prompt sensitivity existing metrics measure. We generate language model outputs and compute existing metrics (Section 3.1) and then decompose these metrics into components (Section 3.2).

3.1 METHODS

Distinguishing the sources of prompt sensitivity requires that specificity and formulation vary independently, that every answer can be scored against the answer the user intended, and that every formulation is answered multiple times, so that decoding noise can be estimated.

Questions and specificity. As data, we use questions from AmbigQA (Min et al., 2020), which builds on real search queries (Kwiatkowski et al., 2019) and provides two specificity levels with annotated answers. We use the term *answers* to refer to language model outputs for questions in the dataset. For each ambiguous query, human annotators list its possible interpretations and write, for each, a disambiguated question that states the interpretation explicitly, with its answer. For example, "Who is the narrator in the Grinch movie?" has one interpretation each for the 1966, 2000, and 2018 movies, as each has a different narrator. The two specificity levels, the ambiguous question and a disambiguated question, are thus written by humans rather than generated by a model (Figure 2), and disambiguating a question increases its specificity. For the disambiguated question, we choose one of the m_0 interpretations of each question by a seeded hash of its identifier, so that every model and rerun evaluates the same one. We call it the *intended interpretation* and its annotated answer the *intended answer*. Answers to both the ambiguous and the disambiguated questions are scored against the intended answer, because a user who asks the ambiguous question means the same interpretation without saying which. If the 1966 movie is the intended interpretation of the Grinch question, "Anthony Hopkins" thus counts as wrong at both specificity levels.

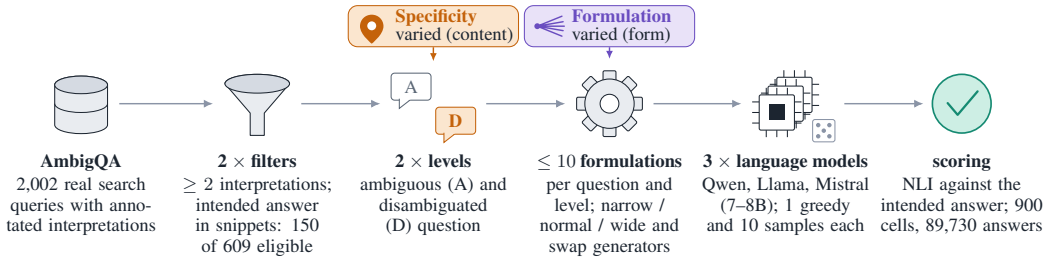

Figure 2: The experimental pipeline. Two filters and dataset order select 150 questions, each asked at two specificity levels in up to ten formulations, answered by three models with one greedy and ten sampled answers, and scored against the intended answer.

Each prompt contains the question and *snippets*, the web search results that AmbigQA's annotators retrieved for the question (Min et al., 2020). The snippets are the same across levels and formulations, so only the question text changes. They hold the available information fixed, so that correctness reflects how a model uses this information rather than what it has memorized, and they make the questions answerable for models of this size (7–8B parameters), which could not answer 84 % of the questions without them in a pilot run. We keep only questions whose intended answer appears verbatim in the snippets, a filter that uses no model output. Of the 1,172 validation-set questions with at least two interpretations, 609 (52 %) pass, and we analyze the first 150 in dataset order, which match the other 459 in number of interpretations and question length.

Formulations. To vary the formulation of a question while its specificity stays fixed, we use Phi-4 (Abdin et al., 2024) to rewrite each question in the voice of eight writing personas, such as a journalist, a domain expert, and a student, so that the formulations differ in style as questions from different users do (Appendix A.2). Two checks ensure that a formulation still asks the same question. A natural language inference (NLI) model (Williams et al., 2018; He et al., 2023) must find that the new formulation and the original imply each other, and Phi-4, acting as a judge, must confirm that the formulation still has the intended answer (for an ambiguous question, at least one of the annotated answers). Entailment is well-defined for statements but ill-defined for questions (Jiang

& de Marneffe, 2022), so neither check is perfect on its own, which is why we use both. We drop near-duplicates, formulations that differ by fewer than six character edits from one already kept, and keep up to ten formulations per question and level.

Models and scoring. We use three open-weight models, Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, and Mistral-7B-Instruct-v0.3 (Qwen Team et al., 2024; Grattafiori et al., 2024; Jiang et al., 2023), to generate the answers. We run them locally on a GPU cluster rather than through an API, which keeps decoding parameters and random seeds under experimental control. Each formulation receives one greedy answer (temperature 0) and $k = 10$ sampled answers (temperature 1), within the 8 to 16 repeated runs after which the pooled intraclass correlation of correctness of Mustahsan et al. (2025) converged on structured tasks. An answer counts as correct when the NLI model finds that the intended answer implies it (entailment probability at least 0.7) and does not contradict it (contradiction probability below 0.5). We call one question at one specificity level, answered by one model, a *cell* of the grid of 150 questions, two levels, and three models. The 900 cells hold up to ten formulations with ten sampled answers each, 89,730 scored answers in total (one question keeps a single formulation at one level). The *success rate* of a formulation x is the share of its k answers scored correct,

$$F(x) = \frac{1}{k} \sum_{j=1}^{k} \mathbf{1}[y_{x,j} \text{ is correct}], \tag{1}$$

and the correctness-based metrics below summarize these success rates.

Existing metrics. We compute ten existing metrics per cell (definitions in Appendix B), chosen from Table 1 because they are defined per question and can be computed from the answers alone. Four *correctness-based* metrics summarize the success rates of a cell. They are accuracy $\bar{F}$, the mean success rate over the formulations, the best- and worst-formulation accuracies $F_{\max}$ (Sclar et al., 2024; Mizrahi et al., 2024) and $F_{\min}$ (Cao et al., 2024), and the spread $F_{\max} - F_{\min}$ (Sclar et al., 2024; Cao et al., 2024). The other six are *dispersion-based* and do not check correctness. Five summarize how the answers of a cell cluster by meaning. The first is semantic entropy $H_{\text{sem}} = -\sum_c p_c \log_2 p_c$ (Kuhn et al., 2023; Farquhar et al., 2024), the entropy of a formulation's answers over the meaning clusters c. It thus measures how widely repeated answers to one prompt scatter. The others are S_τ and total-variation consistency (both adapted from Errica et al., 2025), the number of observed meaning clusters $|\mathcal{A}_q|$ (Kuhn et al., 2023), and the variation ratio (adapted from Lu et al., 2024). The sixth, the embedding variance share ρ_u (Cox et al., 2025), is the share of the variance of answer embeddings that lies between formulations. POSIX (Chatterjee et al., 2024) is excluded because it needs answer probabilities under other formulations (Appendix D.5). Finite samples miss rare answers, so semantic entropy and the number of observed meaning clusters are biased low (McCabe et al., 2026).

Component analysis. To determine how many distinct quantities the ten metrics measure, we compute the Spearman correlations between them within each of the six combinations of model and specificity level and average the six correlation matrices. Computing the correlations within each combination prevents differences between models or levels from creating correlations between metrics. A principal component analysis of the averaged matrix, with varimax rotation, groups the metrics into components. Horn's parallel analysis (Horn, 1965) counts the components whose eigenvalue exceeds what random data of the same size produce.

3.2 Results

The ten metrics group into three components, and only two of them have an adequate existing metric. Three components explain 87 % of the variance of the ten metrics, and each metric loads mainly on one of them (Table 2; correlations and all loadings in Appendix B).

A loading is the correlation between a metric and a component. We name each component after what its metrics have in common. *Correctness* collects accuracy, $F_{\max}$, and $F_{\min}$. A low value indicates that the model answers another interpretation or fails to use the snippets. *Formulation dependence* collects the spread and ρ_u. A high value indicates that the same question is answered correctly in one formulation and wrongly in another. *Output dispersion* collects the five clustering metrics, H_{sem}, S_τ, total-variation consistency, $|\mathcal{A}_q|$, and the variation ratio. A high value indicates that the answers scatter across meanings, whether they are correct or not. The components reflect what the

Table 2: Loadings of the ten existing metrics on the three varimax-rotated components (largest per metric in bold). ρ_F (Section 4.1) did not enter the decomposition and is projected afterwards.

	Component				Component		
Metric	Corr.	Form.	Disp.	Metric	Corr.	Form.	Disp.
Accuracy $\bar{F}$	**0.97**	0.04	−0.15	Semantic entropy H_{sem}	−0.15	0.03	**0.96**
Best formulation $F_{\max}$	**0.95**	0.21	−0.07	S_τ	−0.10	0.24	**0.91**
Worst formulation $F_{\min}$	**0.82**	−0.35	−0.29	Total-variation consistency	−0.14	0.28	**0.94**
Spread $F_{\max} - F_{\min}$	0.32	**0.76**	0.24	Observed clusters $\lvert\mathcal{A}_q\rvert$	−0.18	0.05	**0.89**
Embedding share ρ_u	−0.11	**0.80**	0.30	Variation ratio	−0.07	0.36	**0.81**
ρ_F (our metric)	0.09	**0.89**	0.06				

Corr. = correctness, Form. = formulation dependence, Disp. = output dispersion. The spread is computed from the greedy answers.

metrics measure rather than whether they check correctness. The spread checks correctness, yet it shares a component with ρ_u, which does not, because both measure how answers differ between formulations. Varimax keeps the components uncorrelated by design, so the analysis shows which metrics covary, not whether the components reflect different causes. Section 4 tests that.

Two components have an adequate existing metric, the third does not. Accuracy loads 0.97 on correctness, and semantic entropy loads 0.96 on output dispersion and is established as a measure of uncertainty in language model outputs (Kuhn et al., 2023; Farquhar et al., 2024). We therefore keep them as the metrics of these two components. Formulation dependence has no such metric. Its two metrics agree only weakly, and neither measures correctness net of decoding noise. Parallel analysis retains only the first two components, because the two metrics correlate only 0.42 with each other and the third eigenvalue (1.00) therefore stays below its random-data threshold (1.26). The spread loads 0.32 on correctness besides 0.76 on its own component. Computed here from the greedy answers, it shows only whether the most likely answers differ, and computed from sampled answers, it counts decoding noise as a formulation effect, because every success rate $F(x)$ contains both (Section 4.1). ρ_u is specific to the component (0.80) but scores what the model said rather than whether it was right. In answer to the question of this section, existing metrics measure correctness and output dispersion well, but formulation dependence needs a new metric.

4 Measuring Prompt Sensitivity

This section addresses how each factor can be measured individually. We introduce ρ_F (Section 4.1) and test whether it represents the formulation dependence component and whether accuracy, ρ_F, and semantic entropy respond to different changes of a prompt (Sections 4.2 and 4.3).

4.1 A Metric of Formulation Dependence

Of the existing metrics that check correctness, the spread comes closest to measuring formulation dependence, but each success rate $F(x)$ rests on only $k = 10$ sampled answers. Even if a formulation's true chance of a correct answer is fixed, its $F(x)$ varies from one set of samples to the next, so $F_{\max} - F_{\min}$ is positive even when all formulations are equally good. We therefore measure formulation dependence, i.e., how much of correctness depends on the formulation, as the share of the variance in correctness that lies between formulations rather than within them. If the answers of a cell are grouped by formulation, the variance within a group is decoding noise, and the variance between groups beyond that noise is the effect of the formulation. This share is an intraclass correlation (ICC), the standard measure of how much of an outcome's variance lies between groups (Shrout & Fleiss, 1979; McGraw & Wong, 1996). For a cell with N formulations of k answers each, success rates $\hat{p}_i = F(x_i)$, and their mean $\bar{p}$, the between-formulation (SS_b) and within-formulation (SS_w) sums of squared deviations are

$$\text{SS}_\text{b} = k \sum_i (\hat{p}_i - \bar{p})^2, \qquad \text{SS}_\text{w} = k \sum_i \hat{p}_i(1 - \hat{p}_i), \tag{2}$$

where SS_w, the squared deviations of the binary answers from their formulation's success rate, reduces to this form for binary outcomes (Appendix C.2). Dividing each by its degrees of freedom,

$N-1$ and $N(k-1)$, gives the mean squares $\mathrm{MS_b}$ and $\mathrm{MS_w}$, and

$$\rho_F = \frac{\mathrm{MS_b} - \mathrm{MS_w}}{\mathrm{MS_b} + (k-1)\mathrm{MS_w}}. \tag{3}$$

$\mathrm{MS_w}$ estimates decoding noise, whereas $\mathrm{MS_b}$ contains the formulation effect plus the same noise, because each success rate rests on only k answers. Their difference removes the noise and estimates k times the variance due to the formulation, and the denominator estimates k times the total variance of correctness, so ρ_F is the share of that variance due to the formulation. For example, if five formulations of a question are always answered correctly and five never, all variance lies between formulations and $\rho_F = 1$. If every formulation is answered correctly in half of its samples, all variance is decoding noise and $\rho_F = 0$ (negative estimates are set to zero). Because ρ_F is estimated per cell, it shows for each question and model how much of correctness depends on the formulation, which estimates pooled over a whole task or corpus cannot (Kunievsky & Evans, 2026; Messing, 2026).

Equation 3 is undefined when all answers of a cell are correct or all are wrong. We therefore estimate ρ_F with a hierarchical beta-binomial model, in which each formulation's chance of a correct answer varies around the cell's accuracy by an amount set by ρ_F and the prior on ρ_F is fitted across all cells of a model, so that it is defined on every cell and stabilizes cells with extreme success rates (Appendix C.3). We report this estimate unless stated otherwise and use the value of Equation 3, the method-of-moments (MoM) estimate, and the absolute between-formulation variance $\sigma_B^2 = (\mathrm{MS_b} - \mathrm{MS_w})/k$ to check it. The component analysis and the holdout rule use the MoM estimate (Appendix C.3).

4.2 Methods

Testing ρ_F against the components. To ensure that the analysis justifying ρ_F does not include it, we first project ρ_F onto the fixed components of Section 3 and only then add it as an eleventh metric and repeat the parallel analysis. Since ρ_F is chosen on the same 150 questions on which it is evaluated, we also repeat the whole pipeline on 100 additional AmbigQA questions that pass the same filters and were not included in any earlier analysis. By a rule fixed before any of their answers existed, ρ_F passes if, projected onto the components of the 150 questions, it loads highest on formulation dependence and its margin, that loading minus its largest other loading, exceeds those of the spread and of ρ_u.

Varying specificity and formulation. Next, we test whether the three metrics respond to different changes of a prompt. The two specificity levels of Section 3.1 vary specificity with the formulation generator fixed, so a more specific question should raise accuracy and lower H_{sem} without changing ρ_F. To vary the formulation with the specificity fixed, we regenerate the formulations of 50 questions with a *narrow* generator (three personas, minimal edits), the *normal* generator of the main experiment, and a *wide* generator (eight additional personas, larger edits), holding the meaning fixed (Appendix A.2); more varied formulations should raise ρ_F without changing accuracy or H_{sem}. All three use Phi-4, so a fourth generator, OLMo-2-13B-Instruct (Team OLMo et al., 2024), which also replaces the judge, tests whether the results depend on the model that writes the formulations. Because the narrow generator keeps fewer formulations, we repeat the comparisons of ρ_F and σ_B^2 with every generator subsampled to the same number of formulations per cell.

Statistics and reliability. We test the effect of specificity with nine paired Wilcoxon tests over the 150 questions, one per model for accuracy (the primary test), H_{sem}, and ρ_F, corrected jointly for multiple testing (Holm and Benjamini–Hochberg), with question-clustered bootstrap 95 % intervals; all other tests are descriptive. Because a cell's ρ_F rests on at most ten formulations, we measure its *split-half reliability*, how far the ranking of cells by ρ_F reproduces with a new set of formulations. We estimate ρ_F (MoM) on two random halves of each cell's formulations, rank-correlate the halves across cells, average over 200 splits, and apply the Spearman–Brown correction.

4.3 Results

ρ_F represents formulation dependence. ρ_F loads 0.89 on the formulation dependence component and at most 0.09 on the other two, higher than the spread (0.76) and ρ_u (0.80; last row of Table 2). Added as an eleventh metric, it loads 0.97, no loading of an existing metric changes by more than

0.08, and parallel analysis retains all three components (third eigenvalue 1.50 against a threshold of 1.29). The rule fixed in advance passes on the 100 holdout questions. Projected onto the components of the 150 questions, the margin of ρ_F exceeds that of the spread by 0.25 and that of ρ_u by 0.32, with bootstrap intervals above zero, and a separate decomposition of the holdout reproduces the components (Appendix D.3). On the holdout, shuffling the answers across a cell's formulations, which removes any formulation effect, leaves ρ_F at 0.02 but keeps 47 to 68 % of the spread of the sampled success rates, which is thus decoding noise. With ρ_F, each component has one metric, and we call the components with their metrics the three *factors* of prompt sensitivity, correctness (accuracy), formulation dependence (ρ_F), and output dispersion (H_{sem}).

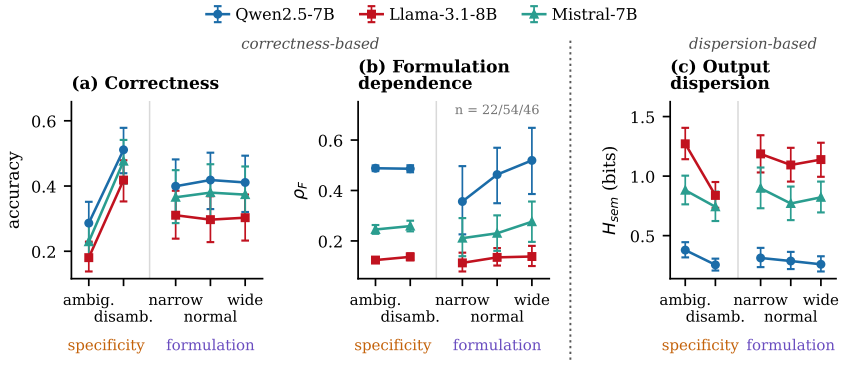

Figure 3: The three factors when specificity and formulation vary, per model. Panels (a) and (b) are correctness-based, and panel (c) is dispersion-based, so lower is not better there. Left blocks vary specificity (means over 150 questions, question-clustered bootstrap 95 % intervals), right blocks vary formulation (means over paired cells, cell-level bootstrap intervals). In (b), the specificity block uses the hierarchical and the formulation block the MoM estimator ($n = 22/54/46$ cells).

Increasing specificity raises accuracy and lowers semantic entropy. Accuracy rises in all three models when the question is disambiguated, by 22 to 25 points, Holm-significant in every model (Figure 3a; Table 4, Appendix D.1). Accuracy on the ambiguous questions ($0.29/0.18/0.23$ for Qwen, Llama, and Mistral) is close to what a model that picked one interpretation at random would reach, and on the 15 questions whose intended answer is also an answer of another interpretation, where knowing the interpretation matters less, the gain vanishes (Appendix D.1). Semantic entropy falls in all three models (Figure 3c; averaged over models $p = 6.1 \times 10^{-5}$). Disambiguated questions are 43 % longer, but within a cell success does not track question length. A more specific question thus helps mainly because it tells the model which interpretation the user means, and the answers then vary less.

ρ_F shows no detectable change as specificity increases. In Figure 3b, ρ_F barely changes between the ambiguous and the disambiguated questions ($-0.002/+0.013/+0.013$, $p \geq .22$). Pooling the posterior draws of the hierarchical model with Rubin's rules (Rubin, 1987), the 95 % intervals of the change lie within about ± 0.07 in every model, on the scale of the shrunken hierarchical estimates. The bootstrap intervals resolve changes of about 0.1 to 0.2, the wider Rubin intervals a change of 0.2 only for Llama (Appendix D.1), so the result is consistent with disambiguation changing what a question asks rather than how much correctness depends on the formulation, without proving that ρ_F is exactly invariant. Formulation dependence thus does not appear to stem from ambiguity in the question.

More varied formulations raise ρ_F, with model-dependent strength. The generators produce increasingly different formulations, as the mean pairwise token edit distance among a cell's formulations confirms (Appendix D.2). ρ_F (Figure 3b) and σ_B^2 rise from the narrow to the wide generator in all three models, also at equal formulation counts (Table 5, Appendix D.2), with one-sided $p = .004/.079/.051$ for ρ_F. Accuracy does not change consistently, and semantic entropy falls only slightly, and monotonically only for Qwen (Figure 3a,c). With the swap generator, per-cell ρ_F agrees with the normal generator at Spearman $+0.27$ to $+0.51$ and σ_B^2 at $+0.64$ to $+0.80$. Absolute values of ρ_F thus depend on the generator, whereas the ranking of models does not, so values of ρ_F should be compared only under one generator.

Reliability limits what single cells show. ρ_F is $0.487/0.130/0.252$ for Qwen, Llama, and Mistral. This ranking holds under both estimators and for σ_B^2 (Table 6, Appendix D.4), and it survives the generator swap for ρ_F, whereas Qwen and Mistral trade places for σ_B^2. Split-half reliability is $0.38/0.52/0.57$, which leaves single-cell values too noisy to report on their own at ten formulations per cell. Per-question values transfer weakly across models (Spearman 0.10 to 0.28), so formulation dependence is a property of the question and the model together. ρ_F correlates only weakly with accuracy and semantic entropy for every model and level ($|\rho| \leq 0.09$ over posterior draws, ≤ 0.28 on posterior means), and its MoM correlation with accuracy lies inside a simulated null that holds the true ρ_F constant (Appendix D.4). Accuracy and semantic entropy, by contrast, are negatively associated (Spearman -0.03 to -0.68).

Existing metrics track ρ_F, and the spread suffices to find where another formulation helps. The spread and ρ_u correlate with ρ_F for every model and level ($+0.23$ to $+0.39$ and $+0.17$ to $+0.50$). Yet the spread ranks the gain from switching to a cell's best formulation, measured on held-out samples, as well as MoM ρ_F or better in two of three models and in all three on disjoint formulation sets, and better than hierarchical ρ_F in all three (Appendix D.4), so the spread suffices to find where another formulation helps most, whereas measuring how much correctness depends on the formulation requires ρ_F. Thus, each factor has a metric of its own, and ρ_F responds to other changes of a prompt than accuracy and semantic entropy do.

5 Conclusion

Prompt sensitivity is not one quantity but three factors: correctness, formulation dependence, and output dispersion. Of the ten existing metrics we analyzed, accuracy measures correctness and semantic entropy measures output dispersion, but none isolates formulation dependence. Our metric ρ_F subtracts decoding noise and represents formulation dependence more closely than existing metrics. Varying specificity and formulation separately, we find that disambiguation moves correctness and output dispersion but not formulation dependence, whereas more varied formulations tend to raise only formulation dependence, which therefore needs a metric of its own. Prompt sensitivity thus has at least three distinct sources: low prompt specificity, which disambiguating the question resolves; formulation dependence, which persists once the question is disambiguated; and decoding noise.

Uncertainty estimates, accuracy, and retries can all miss failures caused by the formulation. ρ_F correlates only weakly with semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024) and accuracy ($|\rho| \leq 0.28$ for any model and specificity level), and the most accurate model in our study, Qwen2.5-7B, is also the most formulation-dependent ($\rho_F = 0.49$, against 0.13 for Llama-3.1-8B; Section 4.3). Retrying the same prompt does not expose them either, because it samples the same formulation again and thus varies only decoding noise, which ρ_F excludes by construction. In research, studies that use language models for annotation, judging, or benchmarking rarely assess prompt sensitivity, although it can change model rankings (Mizrahi et al., 2024; Sclar et al., 2024) and reverse conclusions (Messing, 2026). We therefore recommend that studies report one metric per factor, accuracy, ρ_F, and semantic entropy, together with the formulations, the decoding parameters, and the grader (Appendix C.1).

6 Limitations and Future Work

We tested three open-weight models of 7 to 8 billion parameters on 250 short-answer AmbigQA questions (Min et al., 2020), scored correctness and formulation equivalence with an NLI model, although entailment is ill-defined for questions (Jiang & de Marneffe, 2022), and estimated ρ_F from ten formulations per cell, which ranks models stably but is too unreliable for single-cell values (split-half reliability 0.38 to 0.57). Future work should test whether the factors decompose in the same way for larger and closed models, open-ended tasks, and human-written formulations. Comparing ρ_F across personas and writing styles could show whether formulation dependence affects some groups of users more than others. Future work should also test whether more formulations per cell make ρ_F reliable enough to report per question, whether the grader holds up under human validation, and whether the results hold at other decoding temperatures and on other datasets.

AI USE STATEMENT

Generative AI tools were used to generate and check the question formulations described in Sections 3.1 and 4.2; to implement and refactor parts of the experimental code and to support literature discovery, manuscript structuring, prose revision, and reference formatting. The research contribution was not delegated: the research question, the design of the measurement framework, the choice of dataset and guardrails, and the interpretation of the results are the authors' own. AI output was treated as a draft requiring verification rather than as a result; all AI-assisted code, text, and claims were reviewed, and every reported number was traced to an executed experimental artifact. The authors take responsibility for the final content of this work, including text, claims, code, and artifacts produced with AI assistance.

REPRODUCIBILITY STATEMENT

Sections 3.1 and 4.2 report the question sample, model identifiers, decoding settings, the scoring rule, statistical tests, and how specificity and formulation are varied; Appendices A–D provide the evidence, scoring, and generator details, the metric definitions and component analysis, the population estimand, the derivation and estimators of ρ_F, the specificity and formulation endpoints, the held-out questions, and further measurement properties of ρ_F. The analysis code, the configurations, the per-cell result files the analyses read, a run manifest pinning seeds, thresholds, the question sample, and file hashes, and the dataset of all scored answers are attached to this submission as supplementary material and are public along with the paper (will be deanonymized after review). A single script regenerates the analysis results from these files. Intended interpretations are selected by a seeded hash of the question identifier, so the sample and the levels reproduce exactly; all model requests are cached under a hash of the full request, and every pipeline stage is seeded. Analysis outputs are bit-reproducible within a platform. The derivation the main text states without proof is given in Appendix C.2.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

Bowen Cao, Deng Cai, Zhisong Zhang, Yuexian Zou, and Wai Lam. On the worst prompt performance of large language models. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, volume 37, pp. 69022–69042. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7fa5a377b7ffabcce43cd00231bb3f9c-Abstract-Conference.html.

Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14550–14565, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.852. URL https://aclanthology.org/2024.findings-emnlp.852/.

Kyle Cox, Jiawei Xu, Yikun Han, Rong Xu, Tianhao Li, Chi-Yang Hsu, Tianlong Chen, Walter Gerych, and Ying Ding. Mapping from meaning: Addressing the miscalibration of prompt-sensitive language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (22):23696–23703, 2025. doi: 10.1609/aaai.v39i22.34540. URL https://ojs.aaai.org/index.php/AAAI/article/view/34540.

Federico Errica, Davide Sanvito, Giuseppe Siracusano, and Roberto Bifulco. What did I do wrong? Quantifying LLMs' sensitivity and consistency to prompt engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1543–1558, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.73. URL https://aclanthology.org/2025.naacl-long.73/.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0.

Aaron Grattafiori et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

M. C. Gulliford, G. Adams, O. C. Ukoumunne, R. Latinovic, S. Chinn, and M. J. Campbell. Intraclass correlation coefficient and outcome prevalence are associated in clustered binary data. *Journal of Clinical Epidemiology*, 58(3):246–251, 2005. doi: 10.1016/j.jclinepi.2004.08.012.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA. arXiv:2111.09543.

John L. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30 (2):179–185, 1965. doi: 10.1007/BF02289447.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

Nan-Jiang Jiang and Marie-Catherine de Marneffe. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374, 2022. doi: 10.1162/tacl_a_00523. URL https://aclanthology.org/2022.tacl-1.78/.

Aryan Keluskar, Amrita Bhattacharjee, and Huan Liu. Do LLMs understand ambiguity in text? A case study in open-world question answering. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 7485–7490, Washington, DC, USA, 2024. IEEE. doi: 10.1109/BigData62323.2024.10825265. arXiv:2411.12395.

Olivia Kim. DETAIL matters: Measuring the impact of prompt specificity on reasoning in large language models, 2025. URL https://arxiv.org/abs/2512.02246.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve. arXiv:2302.09664.

Nadav Kunievsky and James A. Evans. Measuring intent comprehension in LLMs. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, Seoul, South Korea, 2026. PMLR. URL https://arxiv.org/abs/2506.16584.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and BERT-NLI. *Political Analysis*, 32(1):84–100, 2024. doi: 10.1017/pan.2023.20.

Sheng Lu, Hendrik Schuff, and Iryna Gurevych. How are prompts different in terms of sensitivity? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5833–5856, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.325. URL https://aclanthology.org/2024.naacl-long.325/.

Lucas H. McCabe, Rimon Melamed, Thomas Hartvigsen, and H. Howie Huang. Estimating semantic alphabet size for LLM uncertainty quantification. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=uYK6GPVg1O. arXiv:2509.14478.

Kenneth O. McGraw and S. P. Wong. Forming inferences about some intraclass correlation coefficients. *Psychological Methods*, 1(1):30–46, 1996. doi: 10.1037/1082-989X.1.1.30. A correction appeared in *Psychological Methods* 1(4), 390 (1996), DOI 10.1037/1082-989X.1.4.390.

Solomon Messing. Hidden measurement error in LLM pipelines distorts annotation, evaluation, and benchmarking, 2026. URL https://arxiv.org/abs/2604.11581.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5783–5797, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.466. URL https://aclanthology.org/2020.emnlp-main.466/.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681. URL https://aclanthology.org/2024.tacl-1.52/.

Zairah Mustahsan, Abel Lim, Megna Anand, Saahil Jain, and Bryan McCann. Stochasticity in agentic evaluations: Quantifying inconsistency with intraclass correlation, 2025. URL https://arxiv.org/abs/2512.06710.

Branislav Pecher, Michal Spiegel, Robert Belanec, and Jan Cegin. Revisiting prompt sensitivity in large language models for text classification: The role of prompt underspecification, 2026. URL https://arxiv.org/abs/2602.04297.

Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. Efficient multi-prompt evaluation of LLMs. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, volume 37, pp. 22483–22512. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/28236482f64a72eec43706b6f3a6c511-Abstract-Conference.html.

Qwen Team, An Yang, et al. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Angelika Romanou, Mark Ibrahim, Candace Ross, Chantal Shaib, Kerem Oktar, Samuel J. Bell, Anaelia Ovalle, Jesse Dodge, Antoine Bosselut, Koustuv Sinha, and Adina Williams. Brittlebench: Quantifying LLM robustness via prompt sensitivity, 2026. URL https://arxiv.org/abs/2603.13285.

Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, New York, 1987.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, et al. The Prompt Report: A systematic survey of prompt engineering techniques, 2024. URL https://arxiv.org/abs/2406.06608.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations (ICLR)*, pp. 25055–25083, 2024. URL https://openreview.net/forum?id=RIu5lyNXjT. arXiv:2310.11324.

Patrick E. Shrout and Joseph L. Fleiss. Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428, 1979. doi: 10.1037/0033-2909.86.2.420.

James H. Steiger. Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, 87 (2):245–251, 1980. doi: 10.1037/0033-2909.87.2.245.

Aditya Taparia, Ransalu Senanayake, Kowshik Thopalli, and Vivek Narayanaswamy. The anatomy of uncertainty in LLMs, 2026. URL https://arxiv.org/abs/2603.24967. ICBINB Workshop at ICLR 2026.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, et al. 2 OLMo 2 Furious, 2024. URL https://arxiv.org/abs/2501.00656. Shorter version in the Second Conference on Language Modeling (COLM), 2025.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Dmitrij Żatuchin. Where does the noise come from? A variance-components decomposition of non-determinism in LLM brand answers, 2026. URL https://arxiv.org/abs/2607.13304.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, and Xing Xie. PromptRobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis (LAMPS '24)*, pp. 57–68, Salt Lake City, UT, USA, 2024. Association for Computing Machinery. doi: 10.1145/3689217.3690621.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. ProSA: Assessing and understanding the prompt sensitivity of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1950–1976, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.108. URL https://aclanthology.org/2024.findings-emnlp.108/.

APPENDIX A
EXPERIMENTAL DETAILS

This appendix completes the description of the pipeline in Section 3.1 and of the formulation generators in Section 4.2. Throughout the appendix, number triples give Qwen2.5-7B, Llama-3.1-8B, and Mistral-7B, in that order.

A.1 EVIDENCE, SCORING, AND ANSWER CLUSTERING

Evidence snippets. The snippets in every prompt are the search results that AmbigQA's annotators retrieved while annotating (Min et al., 2020), and the dataset release notes that these records should not be the input to a system. We use them deliberately as evidence that stays fixed across specificity levels and formulations, so that correctness reflects answering from the given evidence rather than recall. The filter of Section 3.1 keeps only questions whose intended answer appears in the snippets.

Scoring. One NLI model checks the formulations (Appendix A.2), scores the answers, and clusters them by meaning. It is DeBERTa-v3-large (He et al., 2023), fine-tuned on MultiNLI (Williams et al., 2018) and four further NLI datasets and released as `MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli` (Laurer et al., 2024). AmbigQA lists several accepted forms of each annotated answer. An answer is scored correct when, for at least one form of the intended answer as premise and the model's answer as hypothesis, the entailment probability is at least 0.7 and the contradiction probability is below 0.5. The greedy answers are also scored with the entailment threshold lowered to 0.5, a permissive rule that Appendix D.1 uses as a robustness check.

Answer clustering. Two answers are linked when entailment is the most probable label in both directions, and the meaning clusters are the connected components of these links, a transitive variant of the clustering of Kuhn et al. (2023). The clustering runs once per cell over the answers of all its formulations, so that a cluster denotes the same answer in every formulation. All five clustering-based metrics of Appendix B read this one clustering. The embedding variance share ρ_u uses answer embeddings from `sentence-transformers/all-mpnet-base-v2`.

A.2 FORMULATION GENERATORS

The normal generator, used in the main experiment, prompts Phi-4 (Abdin et al., 2024) under eight writing personas (`neutral`, `journalist`, `casual_user`, `domain_expert`, `student`, `terse_keyword`, `formal_academic`, `second_language`) at temperature 0.8. The narrow generator uses three minimal-edit personas (`minimal_synonym`, `minimal_reorder`, `minimal_function_words`) at temperature 0.5. Each asks for a different kind of minimal edit, so that the candidates can still pass the edit-distance deduplication. The wide generator uses eight stylistically diverse personas (`colloquial_chatty`, `verbose_polite`, `cleft_or_passive`, `indirect_statement`, `headline_style`, `spoken_conversational`, `narrative_context`, `exam_question`) at temperature 1.0, restricted to fluent rewrites. The swap generator keeps the personas and the temperature of the normal generator and replaces both the generator and the answer-preservation judge with OLMo-2-13B-Instruct (Team OLMo et al., 2024). The judge changes with the generator because each generator judges its own formulations and because the two models do not fit on one GPU together.

A candidate passes the NLI check when the entailment probability between it and the original question is at least 0.9 in both directions. For questions where fewer than ten candidates pass, the threshold is relaxed to 0.85. The NLI check is the same for every generator, and its rejection rates are recorded for the narrow, wide, and swap generators, so that a difference in strictness is visible rather than silent (Appendix D.2). The prompt templates and judge configurations are part of the released code.

APPENDIX B
THE METRICS AND THEIR STRUCTURE

This appendix defines the ten existing metrics of Section 3.1, shows which of them share inputs, and reports the two-stage component analysis of Sections 3.2 and 4.3 in full.

Correctness-based metrics. Every metric is computed per cell and correlated separately for each model and specificity level, so that differences between models or levels cannot create correlation. The four correctness-based metrics summarize the success rates $F(x)$ of a cell's formulations (Equation 1). Accuracy $\bar{F}$ is their mean, $F_{\max}$ the best-formulation accuracy (Sclar et al., 2024; Mizrahi et al., 2024), and $F_{\min}$ the worst-formulation accuracy (Cao et al., 2024). The spread is $F_{\max} - F_{\min}$ (Sclar et al., 2024), the gap that Cao et al. (2024) also report for judged win rates. In the component analysis, the spread is computed from the greedy answers. Each formulation's greedy answer is either correct or wrong, so this spread is 1 when the greedy answers of a cell's formulations differ in correctness and 0 otherwise. Computed from the sampled success rates instead, the spread leaves the number of retained components unchanged in both stages below and still loads highest on formulation dependence (in the second stage 0.77, against 0.46 on correctness). The greedy version also enters the correlations of the spread with ρ_F in Section 4.3, the rule on the holdout questions, and the payoff check on 20 answers per formulation (Appendix D.4). The sampled version enters the decoding-noise checks on the holdout questions (Appendix D.3) and the payoff check on disjoint formulations.

Dispersion-based metrics. Five metrics read the meaning clusters of Appendix A.1. Semantic entropy H_{sem} (Kuhn et al., 2023; Farquhar et al., 2024) is the entropy of a formulation's answers over the clusters, averaged over the formulations. S_τ (Errica et al., 2025) is the entropy of the answer distribution pooled over the formulations, divided by its maximum $\log_2|\mathcal{A}_q|$ (Errica et al. divide by the number of classes, which open answers do not fix), and it is set to 0 when a cell has a single cluster (7 to 26 % of cells). Total-variation consistency, adapted from the consistency of Errica et al. (2025), is one minus the total-variation distance between the answer distributions of two formulations, averaged over all pairs of formulations. It enters the analysis reversed, as the mean distance, so that all five metrics rise with dispersion. The number of observed clusters $|\mathcal{A}_q|$ (Kuhn et al., 2023) counts the meaning clusters of a cell. The variation ratio, adapted from the input-perturbation variation ratio of Lu et al. (2024), is the share of formulations whose most frequent answer differs from the most frequent answer across formulations. The sixth dispersion-based metric, the embedding variance share ρ_u (adapted from Cox et al., 2025, with sentence embeddings), is the share of the variance of answer embeddings that lies between formulations. POSIX (Chatterjee et al., 2024) requires one teacher-forced pass per ordered pair of formulations, so it is computed on a subset of 100 cells per model and analyzed separately (Appendix D.5).

Shared inputs. Several metrics share an input, so part of their agreement is arithmetic. The five clustering-based metrics are functions of one clustering, and $S_\tau \log_2|\mathcal{A}_q|$ equals H_{sem} plus the Jensen–Shannon divergence between the formulations' answer distributions, so S_τ adds the disagreement between formulations to H_{sem}. Accuracy, $F_{\max}$, $F_{\min}$, and ρ_F summarize one vector of per-formulation success rates, as does the spread when it is computed from them. ρ_u reads answer embeddings and POSIX reads answer likelihoods, so neither shares an input with the other metrics. Agreement among metrics with a shared input is therefore not evidence that they measure the same thing, and a shared input does not make them interchangeable. On the specificity test of Section 4.2 (paired Wilcoxon, disambiguated minus ambiguous question, $n = 150$ per model), the five clustering-based metrics reach different decisions. H_{sem} responds at $p = .011/3.7\times10^{-7}/.034$, S_τ at $p = .056/3.5\times10^{-5}/.18$, the variation ratio at $p = .006/.042/.48$, total-variation consistency at $p = .002/2.8\times10^{-6}/.16$, and $|\mathcal{A}_q|$ at $p = 8.9\times10^{-5}/3.6\times10^{-8}/.052$. They summarize the clustering in different ways, and the paper therefore keeps a single dispersion metric rather than a group of them.

Correlations and the two-stage component analysis. Figure 4 shows the absolute Spearman correlations of the ten existing metrics, averaged over the six combinations of model and specificity level. The dense dispersion block reflects the shared clustering. The analysis that selects one metric per component must not contain the metric it selects, so it proceeds in two stages. In the first stage, the averaged correlation matrix of the ten existing metrics is decomposed into principal components, and the first three are rotated by varimax. ρ_F is held out and projected afterwards, so that

its loadings are correlations with component scores it did not shape. It enters as the MoM estimate (Appendix C.3) on the cells where that estimate is defined (56 to 117 of 150 per model and level), and each of its correlations uses the cells where both metrics are defined. The first three components explain 87.4 % of the variance and the first two 77.4 %, and every existing metric loads most strongly on the component it was expected to measure (Table 3). Horn's parallel analysis (Horn, 1965) compares the eigenvalues 5.25, 2.49, and 1.00 with the 95th percentiles of the eigenvalues of 500 random normal data sets, 1.55, 1.38, and 1.26, and retains two components for every assumed sample size from 56 to 150 and ten simulation seeds. The formulation-dependence component falls short because its two existing metrics correlate only 0.42 with each other. In a two-component solution, the spread splits between the retained components (0.52 on dispersion and 0.47 on correctness), and ρ_u joins dispersion (0.62). Projected onto the three-component solution, ρ_F correlates 0.89 with the formulation-dependence component and 0.06 and 0.09 with the other two. Scoring each model and level separately with the pooled weights reproduces the projection. The Spearman correlation of ρ_F with the formulation-dependence component score is then 0.54 to 0.90 (mean 0.77), and with the other two -0.19 to 0.20.

In the second stage, ρ_F enters as an eleventh metric. Parallel analysis then retains three components (5.37, 2.76, and 1.50 against 1.59, 1.40, and 1.29), again for every sample size and seed, and the third threshold is 1.47 at the smallest sample size of 56. The three components explain 87.2 %, ρ_F loads 0.97 on formulation dependence, and no loading of an existing metric moves by more than 0.08 (Table 3). Without $F_{\max}$ and $F_{\min}$, the existing set would hold a single correctness metric, and parallel analysis would retain two components in either stage, because a component needs several metrics to be retained. The hierarchical estimate of ρ_F, which is also defined on the cells where all answers are correct or all are wrong, projects more weakly onto the first-stage components, at 0.47 on formulation dependence and at most 0.12 in absolute value on the other two (0.65 and at most 0.08 on the holdout questions of Appendix D.3).

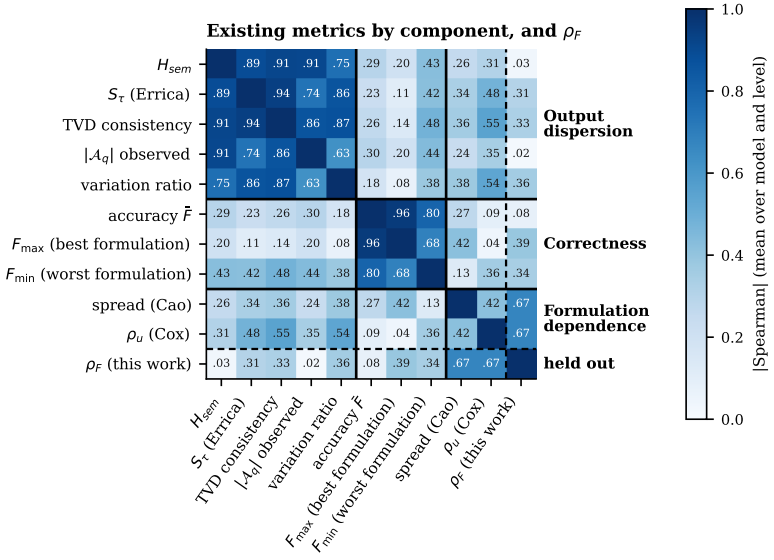

Figure 4: Absolute Spearman correlations among the ten existing metrics, computed separately for each model and specificity level and averaged, ordered by component. The spread is computed from the greedy answers. The bottom row adds ρ_F (Section 4.1) as the MoM estimate on the cells where it is defined; it is not part of the decomposition. ρ_F correlates 0.67 with both the spread and ρ_u, which correlate 0.42 with each other, and at most 0.39 with any other metric.

Table 3: Loadings on the three varimax-rotated components in the two stages of the component analysis (largest per metric in bold). In the first stage, ρ_F is held out and projected; in the second, it is an active metric.

	First stage: ten existing metrics			Second stage: with ρ_F		
Metric	Corr.	Form.	Disp.	Corr.	Form.	Disp.
Accuracy $\bar{F}$	**0.97**	0.04	−0.15	**0.97**	0.05	−0.15
Best formulation $F_{\max}$	**0.95**	0.21	−0.07	**0.94**	0.28	−0.08
Worst formulation $F_{\min}$	**0.82**	−0.35	−0.29	**0.83**	−0.36	−0.29
Spread $F_{\max} - F_{\min}$ (greedy)	0.32	**0.76**	0.24	0.31	**0.71**	0.28
Embedding share ρ_u	−0.11	**0.80**	0.30	−0.12	**0.74**	0.35
Semantic entropy H_{sem}	−0.15	0.03	**0.96**	−0.15	−0.02	**0.97**
S_τ	−0.10	0.24	**0.91**	−0.11	0.23	**0.91**
Total-variation consistency	−0.14	0.28	**0.94**	−0.15	0.26	**0.94**
Observed clusters $\lvert\mathcal{A}_q\rvert$	−0.18	0.05	**0.89**	−0.17	−0.01	**0.90**
Variation ratio	−0.07	0.36	**0.81**	−0.08	0.33	**0.83**
ρ_F (MoM)	0.09	**0.89**	0.06	0.06	**0.97**	0.06
Eigenvalues 1–3	5.25, 2.49, 1.00			5.37, 2.76, 1.50		
Parallel thresholds	1.55, 1.38, 1.26			1.59, 1.40, 1.29		
Retained components	2			3		

Corr. = correctness, Form. = formulation dependence, Disp. = output dispersion. The first-stage columns repeat Table 2. Total-variation consistency enters reversed, as the mean distance between formulations, so that all five clustering-based metrics rise with dispersion.

APPENDIX C
FORMULATION DEPENDENCE: ESTIMAND, DERIVATION, AND ESTIMATORS

C.1 THE POPULATION ESTIMAND

Let P_U be a declared distribution over the accepted formulations of question q, D a decoding policy, S the scoring rule, and $p(x) = \Pr_{r\sim D}[S(r) = 1 \mid x]$ the probability that model m answers formulation x correctly. The population quantity that ρ_F (Equation 3) estimates is

$$\rho_F^\star(q, m; P_U, D, S) = \frac{\mathrm{Var}_{x\sim P_U}[p(x)]}{\mathrm{Var}_{x\sim P_U}[p(x)] + \mathrm{E}_{x\sim P_U}[p(x)(1-p(x))]}, \tag{4}$$

whose denominator is the total variance of correctness when first a formulation and then an answer are drawn. $\rho_F^\star$ is therefore conditional on the question, the model, the set of formulations, the decoder, and the grader. Changing any of them changes the estimand rather than adding noise, which is why the generator swap of Section 4.3 compares rankings across generators rather than values. Our analysis takes P_U to be uniform over the accepted formulations and estimates both terms by repeated decoding.

C.2 THE WITHIN-FORMULATION SUM OF SQUARES FOR BINARY OUTCOMES

Equation 2 states the within-formulation sum of squares of ρ_F as $k\sum_i \hat{p}_i(1-\hat{p}_i)$. This expression is exact, not an approximation, because the outcomes are binary. Let $y_{ij} \in \{0, 1\}$ be the correctness of the j-th of k answers sampled from formulation i, and let $\hat{p}_i = \frac{1}{k}\sum_{j=1}^{k} y_{ij}$ be the observed success rate of that formulation. Because $y_{ij}^2 = y_{ij}$ for binary outcomes,

$$\begin{aligned}\sum_{j=1}^{k}(y_{ij} - \hat{p}_i)^2 &= \sum_{j=1}^{k} y_{ij}^2 - 2\hat{p}_i \sum_{j=1}^{k} y_{ij} + k\hat{p}_i^2 \\ &= k\hat{p}_i - 2k\hat{p}_i^2 + k\hat{p}_i^2 \\ &= k\,\hat{p}_i(1-\hat{p}_i).\end{aligned} \tag{5}$$

Summing Equation 5 over the N formulations gives $\mathrm{SS_w}$ as stated. ρ_F can therefore be computed from the per-formulation success rates alone, and the individual outcomes y_{ij} need not be retained.

C.3 Estimators and Boundary Behavior

The method-of-moments (MoM) value of Equation 3 is clamped to zero when it is negative. Its denominator vanishes only when every formulation in the cell has the same success rate and that rate is 0 or 1. It is therefore undefined on all-correct and all-wrong cells, as well as on cells with a single formulation, where the between-formulation mean square has no degrees of freedom. On these cells it is recorded as missing rather than as zero, and it is defined on 45/66/57 % of all cells. This missingness depends on the outcome, because the accuracy extremes are where the intraclass correlation of a binary outcome degenerates. Analyses on the defined cells are therefore range-restricted on accuracy, and a simulated null largely reproduces this selection (Appendix D.4).

The hierarchical estimator embeds the same estimand in a beta-binomial model. The correct counts of a cell's formulations follow $y_i \sim \mathrm{Binomial}(k, p_i)$, and the p_i follow a beta distribution with mean μ and intraclass correlation ρ_F, that is, with shape parameters μs and $(1-\mu)s$ for $s = (1-\rho_F)/\rho_F$. The mean μ is integrated out under a flat prior, and ρ_F has a $\mathrm{Beta}(a, b)$ prior whose parameters are fitted by type-II maximum likelihood across all cells of a model (empirical Bayes). The reported value is the posterior mean of ρ_F, computed on a grid. It is defined on every cell, at the cost that its values on the boundary cells partly reflect the prior, and its fit is deterministic. The absolute between-formulation variance $\sigma_B^2 = (\mathrm{MS_b} - \mathrm{MS_w})/k$ complements both estimates of the share. Noisier decoding does not push σ_B^2 down, so agreement in the model ordering between the share and σ_B^2 guards against a conclusion driven by the denominator alone.

Which estimator each analysis uses. The hierarchical estimate enters the specificity test, the model ordering, the transfer of per-question values across models, the correlations with the spread and ρ_u (computed on the posterior means), the correlations with accuracy and semantic entropy (pooled over 200 posterior draws per cell with Rubin's rules, and on the posterior means), the check against greedy answers in Table 6, and the per-generator fits of the generator swap. The MoM estimate, on the cells where it is defined, enters both stages of the component analysis, the rule and the decomposition on the holdout questions, the permutation and k-halving checks, the split-half reliability, the simulated null of Appendix D.4, the response to varying formulation (together with σ_B^2), the payoff checks, and the comparison with POSIX. The model ordering holds under both estimators.

Relation to nearby metrics. Five nearby quantities differ from ρ_F by construction rather than by degree, as Table 1 records column by column. The embedding variance share ρ_u (Cox et al., 2025) registers what the model said rather than whether it was right, and it compares the share with its no-sensitivity baseline $1/k$ instead of subtracting the within-formulation term. The correctness spread $F_{\max} - F_{\min}$ (Sclar et al., 2024), computed from sampled success rates, mixes the formulation effect with decoding noise, because each observed success rate carries both. Computed from greedy answers, it contains no decoding noise but records only whether the most likely answers differ. Romanou et al. (2026) decompose correctness scores across items but take one deterministic answer per prompt, so there is no within-formulation term to subtract. Mustahsan et al. (2025) estimate a trial-to-trial term on binary correctness, but their intraclass correlation contrasts questions rather than formulations and takes the plain variance of question means. It therefore neither isolates a formulation effect nor removes decoding noise from its between term, and as a pooled quantity it cannot say for which question another formulation would help. Kunievsky & Evans (2026) sketch the same kind of decomposition for binary correctness (their Appendix H) but report it only for numeric responses and pool it over a task.

Appendix D
Additional Results

D.1 Varying Specificity

How much of the gain is learning the intended interpretation. For the ambiguous question, the model cannot know which interpretation the intended answer belongs to. If it answered a uniformly chosen interpretation as well as it answers the disambiguated question, its accuracy on the ambiguous question would be its accuracy on the disambiguated question divided by m_0, which gives $0.218/0.180/0.199$ against the observed $0.286/0.180/0.229$. On the 15 questions whose intended answer is also an answer of another interpretation, where being told the interpretation changes the

Table 4: Specificity tests (disambiguated minus ambiguous question; paired $n = 150$ questions per model; nine tests corrected jointly for multiple testing). Intervals are question-clustered bootstrap 95 % intervals of the paired difference. Markers come from the nine Wilcoxon tests († significant after Benjamini–Hochberg, ‡ also after Holm), so a marker can differ from what the interval alone suggests. The ρ_F row is the hierarchical estimate, whose resolution is discussed below. Averaged over the three models, the change in H_{sem} is -0.232 $[-0.330, -0.137]$ ($p = 6.1 \times 10^{-5}$, outside the nine tests).

Endpoint	Qwen2.5-7B	Llama-3.1-8B	Mistral-7B
Accuracy (primary)	$+0.225$ $[0.161, 0.292]^{\ddagger}$	$+0.238$ $[0.183, 0.297]^{\ddagger}$	$+0.247$ $[0.183, 0.314]^{\ddagger}$
ρ_F (hierarchical)	-0.002 $[-0.017, 0.014]$	$+0.013$ $[-0.003, 0.029]$	$+0.013$ $[-0.011, 0.038]$
H_{sem} (bits)	-0.124 $[-0.205, -0.045]^{\dagger}$	-0.433 $[-0.583, -0.284]^{\ddagger}$	-0.139 $[-0.277, 0.001]$

score less (for 5 of them, whose intended answer is shared by every interpretation, not at all), the gain is -0.079 $[-0.245, +0.052]$, -0.061 $[-0.166, +0.039]$, and $+0.009$ $[-0.151, +0.145]$, against $+0.259$ $[+0.194, +0.326]$, $+0.271$ $[+0.213, +0.329]$, and $+0.274$ $[+0.204, +0.343]$ on the other 135 (Mann–Whitney $p = .001/.0005/.036$). The gain is therefore mostly the model learning which interpretation the user means, which is part of what a user gains from a more specific question. The remaining effect on shared-answer questions is not distinguishable from zero at $n = 15$. The intended interpretation is chosen by a seeded hash, and the gain does not depend on whether the first-listed interpretation was chosen ($+0.244/+0.225/+0.233$ against $+0.212/+0.247/+0.256$). Propagated through this difference, a new hash seed would move the gain by about ± 0.001 (one standard deviation), less than a fiftieth of the half-width of its 95 % interval.

Resolution of the ρ_F test. The hierarchical estimates shrink toward each model's prior, so a change in ρ_F appears attenuated. When the model is refitted to data simulated with each cell's accuracy held at its observed value and a known change between the levels, a true increase of 0.10 appears as $+0.02/+0.04/+0.02$ and one of 0.20 as $+0.04/+0.08/+0.04$ (means over five seeds). The upper ends of the intervals in Table 4 ($+0.014/+0.029/+0.038$) therefore lie below the attenuated value of a true increase of 0.10 in Qwen2.5-7B and Llama-3.1-8B and of 0.20 in Mistral-7B, and 0.20 is more than half the range of ρ_F between the models. Pooling 200 posterior draws per cell with Rubin's rules (Rubin, 1987) carries each cell's posterior uncertainty into the test and widens the intervals to $[-0.057, +0.054]$, $[-0.021, +0.048]$, and $[-0.041, +0.070]$ ($p \geq .45$), whose upper ends lie below the attenuated value of a true increase of 0.20 only in Llama-3.1-8B.

Scoring robustness. Scored with the permissive rule of Appendix A.1, the accuracy of the greedy answers rises with disambiguation by $+0.205/+0.261/+0.247$ ($p \leq 1.2 \times 10^{-7}$), outside the nine corrected tests.

D.2 Varying Formulation

Manipulation check. The mean pairwise token edit distance among a cell's formulations orders the generators $4.6 < 8.4 < 9.2$ for narrow, normal, and wide, and the full ordering holds in 66 % of cells (Wilcoxon, narrow below wide, $p = 4.2 \times 10^{-18}$). The narrow, normal, and wide generators share the NLI check and the judge, so the different rejection rates of the narrow and wide generators reflect the generator. The NLI check rejects 64 % of wide candidates against 7 % of narrow ones, whereas deduplication removes 84 % of narrow candidates against 23 % of wide ones. Narrow formulation sets therefore end smaller, with 6.7 formulations on average against 10.0 for the other generators, which is why the comparisons are repeated at equal formulation counts.

Identifiability of the per-generator hierarchical fit. Simulating a known constant ρ_F on each generator's realized formulation sets, at equal cell counts and over five seeds, separates a regime problem from a sample-size problem. On the normal generator's formulation sets, the hierarchical estimator tracks the truth and recovers a true 0.3 as $0.26/0.28/0.33$. On the narrow generator's sets, which are smaller and more often all correct or all wrong, the same true 0.3 is recovered as $0.57/0.30/0.25$, with deviations of up to 0.23 across seeds. Fitted per generator, the hierarchical estimate falls from narrow to wide for Qwen2.5-7B (0.68, 0.50, 0.52), the model whose narrow sets the simulation inflates most, and rises for Llama-3.1-8B (0.08, 0.09, 0.12) and Mistral-7B (0.19,

Table 5: Response to varying formulation (50 questions, each ambiguous and disambiguated). MoM ρ_F requires variance under all three generators, which selects the n paired cells depending on the outcome. The unpaired per-generator means and the equal-count tests (five subsample seeds) in the note bound that selection. p is from a one-sided Wilcoxon test of narrow $<$ wide. The per-generator hierarchical fit is unreliable for the narrow generator, as the simulation below shows, and is therefore not reported here.

Model	Quantity	Narrow	Normal	Wide	p (n)	Δ(w−n) [95 % CI]
Qwen2.5-7B	σ^2_B	0.026	0.026	0.030	.054 (98)	$+.005$ $[-.006, +.015]$
	ρ_F MoM	0.356	0.463	0.519	**.004** (22)	$+.163$ $[+.063, +.278]$
Llama-3.1-8B	σ^2_B	0.011	0.014	0.015	**.021** (98)	$+.005$ $[+.001, +.008]$
	ρ_F MoM	0.113	0.135	0.138	.079 (54)	$+.025$ $[-.015, +.066]$
Mistral-7B	σ^2_B	0.021	0.021	0.025	.060 (98)	$+.004$ $[-.004, +.012]$
	ρ_F MoM	0.211	0.230	0.276	.051 (46)	$+.065$ $[-.006, +.139]$

Note: unpaired per-generator ρ_F means (each generator's own defined cells): 0.334/0.389/0.395 (n=32/42/40, Qwen); 0.098/0.131/0.114 (62/62/71, Llama, non-monotone); 0.195/0.197/0.226 (51/62/65, Mistral). Equal-count one-sided p, median [range] over five subsample seeds: ρ_F MoM .007 [.003, .016] / .037 [.021, .143] / .113 [.065, .293]; σ^2_B .036 [.029, .246] / .032 [.012, .069] / .159 [.041, .298].

0.28, 0.24). The MoM contrast on the defined cells and the equal-count contrasts are therefore the interpretable quantities for varying formulation, and Table 5 reports them.

Generator swap. With OLMo-2-13B-Instruct as generator and judge (the same 50 questions at both levels), the per-generator hierarchical estimate of ρ_F is $0.360/0.129/0.307$, against $0.503/0.085/0.279$ for the normal generator on these questions, so the ordering Qwen2.5-7B $>$ Mistral-7B $>$ Llama-3.1-8B holds under both generators (MoM under the swap generator: $0.285/0.112/0.283$). σ^2_B is $0.022/0.016/0.028$ under the swap generator, which puts Mistral-7B above Qwen2.5-7B. Per cell, ρ_F agrees between the two generators at Spearman $+0.41/+0.27/+0.51$ and σ^2_B at $+0.68/+0.64/+0.80$.

D.3 Held-out Questions

The 100 holdout questions (600 cells) pass the same filters as the 150 main questions and entered no earlier analysis. The rule of Section 4.2 is evaluated on MoM ρ_F, projected onto the frozen first-stage components of the 150 questions. ρ_F loads highest on formulation dependence, and its margin exceeds that of the spread by 0.25 $[0.13, 0.38]$ and that of ρ_u by 0.32 $[0.17, 0.43]$ (bootstrap over holdout questions). The frozen components were recomputed once after the holdout answers existed, when S_τ was corrected to follow Errica et al. (2025), and the rule passed under both versions (margins 0.34 and 0.31 before the correction). A separate decomposition of the holdout questions retains two components for the ten existing metrics and three with ρ_F (third eigenvalue 1.40 against a threshold of 1.36, two or three when the assumed sample size is varied), with loadings nearly identical to those of the 150 questions (Tucker congruence at least 0.995). There ρ_F loads 0.97 on formulation dependence and at most 0.08 on the other two, whereas the spread and ρ_u load up to 0.36 on other components.

Two checks show that the MoM estimate excludes decoding noise. Permuting the answers of a cell across its formulations removes any formulation effect but keeps the cell's accuracy. It leaves MoM ρ_F at 0.02 but keeps 47/68/55 % of the spread of the sampled success rates, which is therefore decoding noise. Halving k from ten to five, using the first and the second five answers of every formulation, changes MoM ρ_F by at most 1 % but raises the spread by 3/15/8 %. The hierarchical estimate, refitted at $k = 5$, rises by 8/56/23 %, so its values are comparable only at equal k.

The main results recur on the holdout questions. Disambiguation produces no detectable change in hierarchical ρ_F ($-0.018/+0.001/+0.001$, $p \geq .54$), the model ordering Qwen2.5-7B $>$ Mistral-7B $>$ Llama-3.1-8B holds, and split-half reliability is $0.55/0.57/0.64$. On disjoint halves of five formulations, the spread again predicts the payoff of switching formulation better than MoM ρ_F ($+0.40/+0.55/+0.51$ against $+0.33/+0.41/+0.44$).

Table 6: Measurement properties of ρ_F. Every row names its estimator, and all rows score against the intended answer. The switching payoff is the best formulation's success rate minus the mean over formulations, measured on answers or formulations that the predictor did not use.

Property	Qwen2.5-7B	Llama-3.1-8B	Mistral-7B
ρ_F, hierarchical (mean over cells)	0.487	0.130	0.252
ρ_F, MoM (mean over defined cells)	0.400	0.128	0.233
σ_B^2 (mean over cells)	0.031	0.014	0.023
MoM coverage	45 %	66 %	57 %
Spearman correlation of the MoM and hierarchical estimates	0.87	0.89	0.86
Split-half reliability (MoM)	0.38	0.52	0.57
Greedy answers differ in correctness (partial Spearman, hierarchical)	$+0.51$	$+0.40$	$+0.54$
Switching payoff on answers 11 to 20, predicted by MoM ρ_F	$+0.62$ (n=42)	$+0.40$ (n=62)	$+0.70$ (n=62)
… by hierarchical ρ_F ($n = 100$)	$+0.20$	$+0.01$	$+0.11$
… by the spread of the greedy answers ($n = 100$)	$+0.80$	$+0.63$	$+0.62$
Switching payoff on five held-out formulations, predicted by MoM ρ_F	$+0.29$	$+0.37$	$+0.38$
… by the spread of the sampled success rates	$+0.39$	$+0.53$	$+0.50$

D.4 Measurement Properties of ρ_F

The model ordering Qwen2.5-7B > Mistral-7B > Llama-3.1-8B holds for the hierarchical estimate, the MoM estimate, and σ_B^2 (Table 6), and the two estimators agree at Spearman 0.86 to 0.89 on the cells where both are defined. Greedy answers contain no decoding noise, so when the greedy answers of a cell's formulations differ in correctness, the formulation alone decides whether the answer is correct. The hierarchical ρ_F predicts this disagreement at partial Spearman $+0.51/+0.40/+0.54$, controlling for how far the cell's accuracy lies from one half.

The correlation with accuracy is within estimator mechanics. The intraclass correlation of a binary outcome is known to be associated with the outcome's prevalence (Gulliford et al., 2005), so an association between ρ_F and accuracy can appear with no relation between the constructs. We simulated the full formulation-by-answer grid from the beta-binomial model with the true ρ_F held constant across cells and each cell's mean success held at its observed value, and re-estimated ρ_F with the pipeline's own MoM estimator (500 simulations). All six observed correlations of MoM ρ_F with accuracy on the defined cells, three models at two specificity levels, fall inside the 5th-to-95th-percentile band of this null. The null also largely reproduces the coverage selection of Appendix C.3, because under a constant true ρ_F the cells near the accuracy extremes are again the ones where the estimate is undefined. The association on the defined cells is therefore uninformative about the constructs at this design size, which is why the hierarchical estimator is primary.

The payoff of switching formulation. For 50 questions at both specificity levels, every formulation received 20 sampled answers. The metrics are computed from the first ten answers, and the payoff of switching formulation is measured on the other ten. MoM ρ_F predicts the payoff on the cells where it is defined, but the spread of the greedy answers on all 100 cells does as well or better in two of the three models, and the hierarchical estimate predicts it only weakly (Table 6). On disjoint halves of five formulations (all 150 questions), which do not reuse the formulations of the estimate, the spread of the sampled success rates ranks the payoff better in all three models. The spread therefore suffices to find where another formulation helps most, as Section 4.3 states.

D.5 POSIX

POSIX (Chatterjee et al., 2024) requires one teacher-forced pass per ordered pair of formulations, so it is computed for 50 questions at both specificity levels (100 cells per model). Within this subset, its index ψ correlates with H_{sem} at $+0.41/+0.38/+0.56$ and with MoM ρ_F at $+0.24/+0.30/+0.32$ ($n = 42/62/62$ cells where ρ_F is defined). A Williams test of the difference (Steiger, 1980) is not significant in any model ($p = .42/.63/.054$), and the correlations with ρ_F are attenuated by its split-half reliability of 0.38 to 0.57. POSIX also falls with disambiguation in all three models, the direction of lower output dispersion. It therefore does not separate output dispersion from formulation dependence at this sample size, which is why it is reported here rather than as the metric of either factor.