LACE: Locality-Aware (syntactic) Complexity Estimation

Anonymous ACL submission

Abstract

Syntactic complexity contributes to language processing difficulty, yet most evaluation metrics rely on shallow proxies or conflate syntactic and lexical difficulty, leaving the structural contribution unmeasured. We address this gap by introducing two locality-aware syntactic complexity metrics inspired by Dependency Locality Theory (DLT): LACE-Core, which quantifies the memory load and integration difficulty of syntactic dependencies, and LACE-Full, which additionally captures the cost of introducing new discourse referents. We benchmark these metrics across three dimensions: (1) agreement with human judgments of text simplification, (2) overlap with other complexity metrics, and (3) prediction of difficulty in downstream QA tasks. Our results show that LACE-Full aligns more closely with human judgments of simplified text, while LACE-Core provides the most length-independent signal of structural complexity. These findings establish LACE as a theory-grounded benchmark for syntactic complexity, with potential applications to text simplification, readability, and accessibility.

1 Introduction

Syntactic complexity is a well–established determinant of processing difficulty in human language comprehension (Yngve, 1960; Chomsky and Miller, 1968; Miller and Chomsky, 1963). Psycholinguistic work shows that constructions involving long–distance dependencies, center embeddings, and the introduction of new discourse referents impose substantial memory and integration demands (Bever, 2013; Miller and Isard, 1964; Gibson, 1998, 2000). These effects cannot be inferred solely from surface features such as sentence length or lexical choice. Rather, they arise from the *syntactic structural configuration* of a sentence, such as its dependency distances, unresolved relations, and points of integration (Abney and Johnson, 1991; Kimball, 1973; Levy, 2008; Futrell et al., 2020).

Prior work has revealed that Large Language Models (LLMs) struggle to parse structures such as center-embeddings, *garden-path* sentences, long-distance and referential dependencies (Amouyal et al., 2025; Irwin et al., 2023; Li et al., 2024). These failures arise even when the vocabulary is simple and the length is short, indicating that *syntactic load*, rather than lexical complexity, contributes to model errors. Despite this limitation, there is no standardized metric for quantifying structural complexity in naturalistic text or systematically relating this complexity to model behavior. This motivates the need for *a practical metric that directly quantifies syntactic load.*

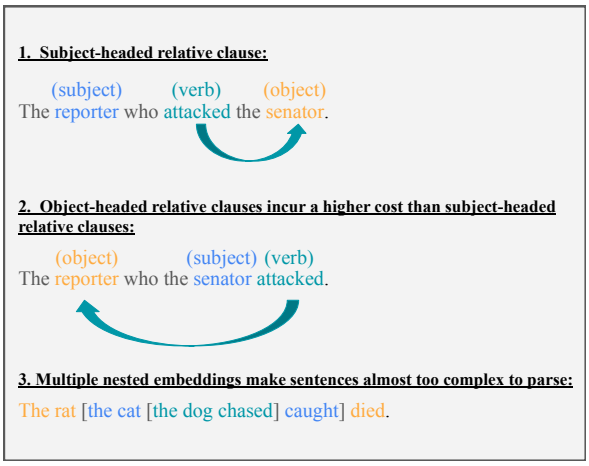


Figure 1: Different syntactic structures, such as relative clauses and center embeddings, can contain varying degrees of syntactic complexity. Sentences 1 and 2 are taken from Gibson (1998). Sentence 3 is modified from Hudson (1996).

Automatic evaluation of text complexity (and, by extension, the quality of simplified text), has traditionally relied on two categories of metrics. (1) *Readability formulas*, such as Flesch Reading Ease (FRE), Flesch–Kincaid Grade Level (FKGL), Fog Index (FOG), and SMOG grade (Kincaid et al., 1975), estimate difficulty from surface features such as sentence length or proportions of polysyllabic words. (2) *Machine–translation–inspired metrics*, including BLEU, iBLEU, and SARI (Sun and Zhou, 2012; Xu et al., 2016) compare system outputs to human references using n–gram overlap. Although useful for certain evaluation settings, these metrics provide no principled way to isolate the contribution of *syntactic structure* to overall complexity. They are driven primarily by lexical or

surface patterns, not by dependency configuration or structural load.

A theory that directly models structural load is therefore needed. To address this gap, we draw on *Dependency Locality Theory* (DLT; Gibson, 1998, 2000), a theoretical framework that models structural difficulty in terms of storage and integration costs over syntactic dependencies. DLT explains classic contrasts in human processing difficulty, for instance, why object–extracted relative clauses are harder to understand than subject–extracted ones, and why multiple center embeddings quickly become uninterpretable, as shown in Figure 1. Crucially, DLT offers an explicit account of the structural determinants of complexity, independent of surface length or lexical frequency.

We operationalize DLT into two automatic syntactic complexity metrics: LACE-CORE, which quantifies the memory load and integration cost associated with syntactic dependencies, and LACE-FULL, which augments LACE-CORE with the *explicit, cumulative* cost of introducing new discourse referents. We evaluate these metrics across three dimensions: (1) correspondence with human judgments of text simplification; (2) overlap and complementarity with established complexity measures; and (3) associations with model difficulty in downstream reasoning tasks. By bridging psycholinguistic theory and NLP evaluation, we contribute a theory–grounded, computationally tractable benchmark for syntactic complexity based on DLT and adapted for modern NLP evaluation.[1]

2 Background and Related Work

We situate our work within three areas of prior research: (1) psycholinguistic theories of syntactic complexity, (2) computational metrics for measuring complexity in text, and (3) syntactic robustness in neural language models. Together, these threads motivate the need for a theory-grounded, reference-free metric that directly quantifies structural load in naturalistic text.

2.1 Psycholinguistic Accounts of Syntactic Complexity

A long tradition in psycholinguistics links structural properties of sentences to processing difficulty. Classic work showed that humans incur substantial cognitive load when resolving long-distance dependencies, center embeddings, or unresolved syntactic relations (Yngve, 1960; Miller and Chomsky, 1963; Chomsky and Miller, 1968; Miller and Isard, 1964; Bever, 2013). Such effects may arise independently of sentence length or lexical difficulty, reflecting load in working memory and integration processes (Abney and Johnson, 1991; Kimball, 1973; Levy, 2008; Futrell et al., 2020).

Dependency Locality Theory (DLT) (Gibson, 1998, 2000) provides a formal account of these cognitive costs. DLT models processing difficulty as arising from (i) the introduction of new discourse referents, (ii) the storage of incomplete syntactic predictions, and (iii) integration across syntactic distance. Experimental evidence shows that DLT explains contrasts in relative clause difficulty, pronoun facilitation, and the breakdown of center embeddings, making it one of the most influential theories of sentence processing.

Despite its explanatory power in cognitive science, DLT has not been operationalized into a general-purpose, computable metric for syntactic complexity in NLP. Our work directly addresses this gap.

2.2 Metrics for Measuring Syntactic Complexity

Surface-based and readability metrics. Automatic evaluation of text complexity has largely borrowed from readability research. Formulas such as Flesch Reading Ease, Flesch-Kincaid Grade Level, Fog Index, and SMOG grade (Kincaid et al., 1975) estimate difficulty from surface proxies including sentence length, syllable counts, or proportions of polysyllabic words. While widely used, these metrics capture lexical and length-based difficulty, not structural load.

Reference-based metrics. Evaluation in text simplification has been dominated by machine translation-inspired metrics such as BLEU, iBLEU, and SARI (Sun and Zhou, 2012; Xu et al., 2016). SARI rewards n-gram additions, deletions, and retentions, making it the standard metric for simplification. Recent work has extended evaluation to semantic fidelity and fluency using BERTSCORE (Zhang et al., 2020), MOVERSCORE (Zhao et al., 2019), and UNIEVAL (Zhong et al., 2022), or tested factual consistency using QA-based approaches such as QUESTEVAL (Scialom et al., 2021). However, these methods emphasize lexical or semantic similarity and do not isolate syntactic structural load.

Structural proxies. A smaller body of work has attempted to capture syntactic complexity through structural heuristics. Parse tree depth and mean dependency length correlate with processing difficulty (Temperley, 2007; Liu, 2008), and dependency-based compression methods implicitly reduce syntactic load (Filippova and Strube, 2008). Metrics such as T-UNIT length (Hunt, 1965) and SUB-CLAUSE counts (Lu, 2010) also track structural variation. Yet these proxies are coarse, task-dependent, and lack grounding in cognitive theory (Al-Thanyyan and Azmi, 2021).

2.3 Syntactic Complexity as a Probe of Model Robustness

Syntactic phenomena have long been used to diagnose brittleness in neural language models. Early work showed that RNNs struggle with syntax-sensitive generalizations such as subject–verb agreement (Linzen et al., 2016) and hierarchical contrasts (Marvin and Linzen, 2018), motivating controlled benchmarks including BLIMP (Warstadt et al., 2020) and SYNTAXGYM (Gauthier et al., 2020) that target phenomena such

[1] We will release all code for our experiments upon acceptance.

as center embeddings, agreement attraction, and garden-path effects. While large pretrained models exhibit some hierarchical generalization, they remain vulnerable to increasing structural load (Wilcox et al., 2019; Irwin et al., 2023; Li et al., 2024).

Recent probing work shows that these limitations persist in modern LLMs. Diego-Simón et al. (2025) find that syntactic representations degrade with increasing dependency distance and syntactic depth, while Amouyal et al. (2025) show systematic misinterpretation and reanalysis failures on classic garden-path constructions in both humans and LLMs. Nandi et al. (2025) further demonstrate weak syntactic generalization in transformer models unless explicitly biased toward hierarchical structure, and Williamson et al. (2025) report correlations between syntactic complexity and mathematical reasoning errors without establishing cross-domain generality. Together, these studies use syntactic complexity diagnostically, via controlled probes or minimal pairs, rather than as a general-purpose, computable metric applicable to natural text.

Gap in the Literature Although syntactic complexity is central to psycholinguistics, text simplification, and LLM evaluation, no widely adopted, theory-driven metric exists for quantifying structural load in arbitrary text: readability metrics emphasize surface difficulty, MT-style metrics require references, structural proxies are coarse, and diagnostic evaluations are not general-purpose. To our knowledge, no prior work operationalizes Dependency Locality Theory as a scalable, reference-free metric or benchmarks it against human judgments, structural baselines, and LLM reasoning difficulty.

3 Methodology: A Locality-Aware Measure of Syntactic Complexity

Syntactic complexity has long been studied in psycholinguistics as a determinant of human language processing difficulty. Among formal models, DLT (Gibson, 1998, 2000) provides one of the most influential accounts. DLT attributes processing cost to three cognitively motivated sources: (i) the introduction of new discourse referents, (ii) the storage of incomplete dependencies in working memory, and (iii) the integration of dependents with their syntactic heads across distance.

We adapt these components into a practical, computable metric of syntactic complexity that can be applied to arbitrary text corpora. To separate structural from discourse-related contributions, we define two variants of our DLT-inspired metric: **LACE-Core**, which measures only storage and integration costs, and **LACE-Full**, which additionally incorporates discourse referent introduction.

3.1 Discourse Cost

Discourse cost reflects the introduction of new discourse referents, realized by content words such as nouns, proper nouns, verbs, and numerals. For a sequence s with tokens $w_{1\ldots N}$, we formally define discourse cost in Equation 1 and visualize it in Figure 2.

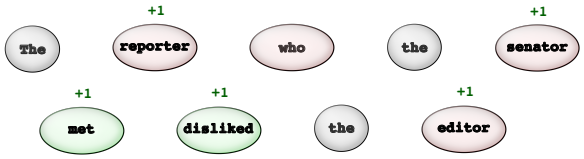


Figure 2: Discourse Cost on an example sentence. Each referent increments the cost by 1. Sentence modified from Gibson (2000).

$$C_{\text{discourse}}(s) = \sum_{i=1}^{N} \mathbb{1}[w_i \in \texttt{REFERENT\_POS}], \quad (1)$$

where a token is considered a referent if it has a POS tag $\in \{\texttt{NOUN}, \texttt{PROPN}, \texttt{NUM}, \texttt{VERB}\}$. Each such token increments the discourse cost by 1 (Gibson, 1998, 2000).

3.2 Storage Cost

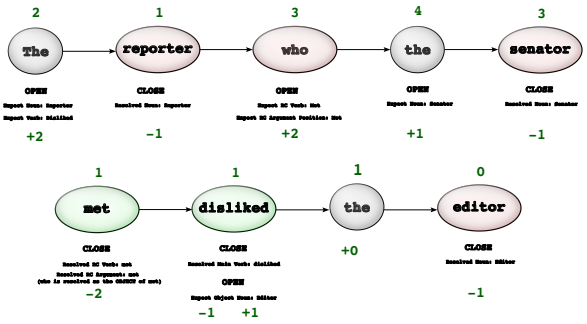


Figure 3: Storage cost of sequence. Each syntactic head introduces *Memory Units* (MUs) to complete the string as a grammatical sentence. Sentence modified from Gibson (2000).

Comprehension requires maintaining predictions about upcoming syntactic dependents until they are resolved. For example, encountering a subject noun phrase creates an expectation for a verb to complete the dependency. At each token position i, let $O(i)$ denote the set of unresolved dependencies at that position.

In classical DLT (Gibson, 1998, 2000), storage cost is defined as the cumulative number of such open dependencies. For large-scale automatic evaluation across heterogeneous corpora, however, cumulative counts can be unstable and highly sensitive to local parse variation. Inspired by the same underlying intuition as DLT but adapted for robust corpus-level scoring, we approximate storage cost using the *peak* number of unresolved dependencies, normalized by sentence length.

$$C_{\text{storage}}(s) = \frac{\max_{1 \le i \le N} |O(i)|}{N}, \quad (2)$$

This emphasizes the maximum momentary working-memory load, the point of greatest resource demand,

while preserving the DLT intuition that unresolved syntactic commitments drive processing difficulty (see Figure 3).

3.3 Integration Cost

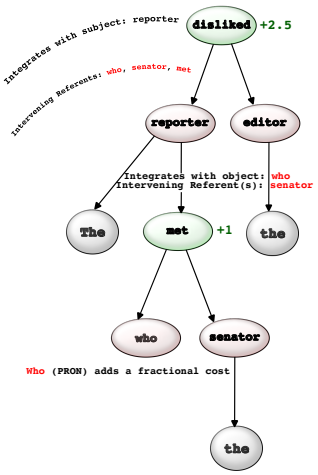


Figure 4: Integration cost is computed by counting the discourse referents that intervene between a dependent and its syntactic head; each such referent contributes one unit of integration cost. Sentence modified from Gibson (2000).

Integration occurs when a dependent is attached to its syntactic head, and processing cost increases with the distance between the two. In DLT, this distance is defined as the number of intervening discourse referents between the dependent and its head. For a sentence s with tokens $w_{1\ldots N}$, let $D(s)$ denote the set of dependencies in its parse, where each element is a dependent-head pair (w_i, h_i). Let $I(w_i, h_i)$ be the set of intervening referents between w_i and h_i.

Following Gibson (2000), who note that integrating over pronouns is less costly than integrating over non-pronominal referents, we assign each intervening referent $r \in I(w_i, h_i)$ a cost of 0.5 if it is a pronoun and 1 otherwise. We formally define the cost of each referent in Equation 3:

$$\text{REF\_COST(I)} = \sum_{r \in I} \begin{cases} 0.5 & \text{if } POS(r) = \text{PRON} \\ 1 & \text{otherwise} \end{cases}, \quad (3)$$

where $POS()$ is a function that returns the part-of-speech tag of an input referent r.

Classical DLT sums the costs of referents directly across all dependencies. For large-scale automatic evaluation across corpora with wide variation in sentence length and referential density, raw sums are difficult to compare and can obscure structural effects. To retain the core DLT intuition while ensuring cross-sentence comparability, we normalize the summed referential costs by the total number of discourse referents $C_{\text{discourse}}$ to get the final integration cost in Equation 4:

$$C_{\text{integration}}(s) = \frac{\sum_{(w_i,h_i)\in D(s)} REF\_COST(I(w_i, h_i))}{C_{\text{discourse}}}. \quad (4)$$

This normalized formulation preserves the theoretical basis of DLT that longer and referentially denser dependencies impose greater integration cost, while yielding a stable, scale-invariant measure suitable for heterogeneous NLP datasets.

3.4 LACE-Core and LACE-Full Metrics

We define two aggregate metrics from these components: **LACE-Core** captures the syntactic structural load:

$$\text{LACE-Core}(s) = C_{\text{storage}}(s) + C_{\text{integration}}(s). \quad (5)$$

LACE-Full augments this by adding discourse cost directly:

$$\begin{aligned} \text{LACE-Full}(s) &= C_{\text{storage}}(s) \\ &+ C_{\text{integration}}(s) \\ &+ C_{\text{discourse}}(s) \end{aligned} \quad (6)$$

In this formulation, storage and integration costs are normalized while discourse cost is raw. This design reflects a balance between structural load and referent density. Together, these variants provide a principled, psycholinguistically inspired benchmark for syntactic complexity that is directly computable from dependency parses.

4 Evaluation Setup

To validate our proposed metrics, we design an evaluation protocol aligned with three research questions (RQs). Each RQ targets a complementary dimension of validity:

RQ 1: To what extent does LACE align with human judgments of simplification?

RQ 2: How does LACE compare with established surface and structural baselines in capturing syntactic complexity?

RQ 3: How reliably does LACE predict difficulty for downstream NLP tasks?

Together, these RQs form a comprehensive evaluation framework: RQ1 anchors the metrics to human perception, RQ2 situates them relative to canonical syntactic complexity metrics, and RQ3 tests whether they capture the structural difficulty that challenges state-of-the-art models.

4.1 Baseline Metrics

To contextualize the performance of LACE, we compare it against four canonical syntactic complexity metrics:

- **Sentence length**(Length): defined as the number of tokens.

- **Mean dependency length**(MDL): the average linear distance between heads and dependents (Liu, 2008).
- **Mean T-unit length**(T-UNIT): the average length of a minimal terminable unit (an independent clause plus its dependents) (Hunt, 1965).
- **Subordinate clause count**(SUB-CLAUSE): the number of dependent clauses per sentence (Lu, 2010).

We report all evaluations for both LACE-CORE and LACE-FULL alongside these baselines.

4.2 Datasets

Our evaluation spans two categories of datasets corresponding to the three research questions.

4.2.1 Simplification Corpora

For **RQ1** and **RQ2**, we evaluate syntactic complexity metrics on established sentence simplification corpora spanning multiple languages, including **ASSET** (Alva-Manchego et al., 2020) (English crowdsourced complex–simple pairs with multiple references), **WIKIAUTO** (Jiang et al., 2020) (large-scale automatically aligned English Wikipedia–Simple Wikipedia pairs), and **MULTISIM** (Ryan et al., 2023), a professionally curated multilingual benchmark. For **MULTISIM**, we use selected subcorpora from **German**, **French**, and **Italian**.[2]

We focus on these non-English subcorpora because they are predominantly SVO languages for which our DLT-based complexity computation applies directly, with cross-lingual variation arising primarily from the dependency parser rather than from language-specific algorithmic changes. Moreover, these languages are supported by reliable, high-quality dependency parsers in SPACY, enabling consistent and robust computation of metrics across corpora.

4.2.2 QA Benchmarks

For **RQ3**, we evaluate the impact of syntactic complexity on downstream reasoning using three challenging QA benchmarks: **SUPERGPQA** (Team et al., 2025), a large-scale multiple-choice benchmark of graduate-level questions across academic disciplines; **MMLU** (Hendrycks et al., 2021), a broad multiple-choice benchmark covering professional and academic knowledge; and **GSM8K** (Cobbe et al., 2021), a dataset of linguistically diverse grade-school math word problems.

For RQ1 and RQ2, LENGTH is computed as the number of whitespace-separated tokens. For RQ3, LENGTH is determined using each model's native tokenizer. In all cases, dependency parses are generated using the same parsing pipeline, and all complexity metrics (LACE and baseline) are computed over these parses to ensure comparability across experiments.

[2]The selected MULTISIM subcorpora include CLEAR and WIKILARGEFR (French), GEOLINOTEST and TEXTCOMPLEXITYDE (German), and ADMINIT, SIMPITIKIWIKI, PACCSS-IT, TEACHER, and TERENCE (Italian).

5 Results and Analysis

5.1 RQ1: Alignment with Human Judgments of Simplification

To assess whether LACE reflects human intuitions of syntactic complexity, we evaluate their ability to assign higher complexity to the more complex sentence in aligned complex-simple sentence pairs from the datasets introduced in subsubsection 4.2.1.

For each aligned pair $(s_{\text{orig}}, s_{\text{simp}})$ we compute the difference:

$$\Delta_{\text{METRIC}} = \text{METRIC}(s_{\text{orig}}) - \text{METRIC}(s_{\text{simp}}),$$

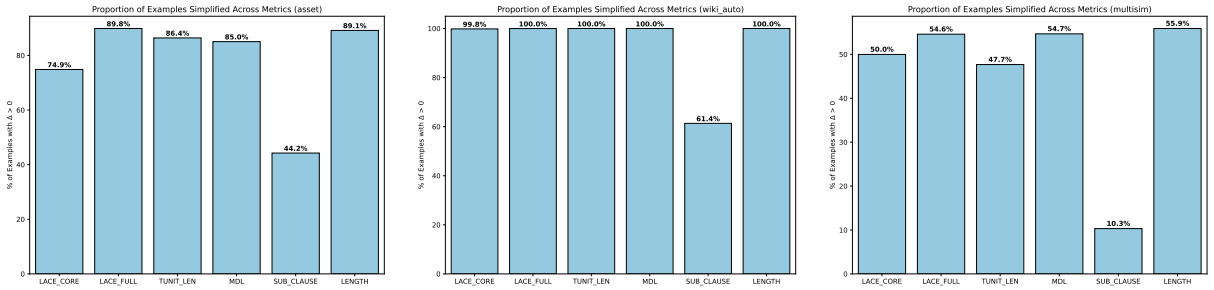
where a positive value indicates that the metric assigns higher complexity to the original (unsimplified) sentence. For each metric and dataset, we report the proportion of pairs where $\Delta_{\text{METRIC}} > 0$, which corresponds to agreement with the gold simplification direction.

Figure 5 presents these results across ASSET, WIKIAUTO, and MULTISIM. On ASSET, LACE-CORE and LACE-FULL agree with human simplification in 74.9% and 89.8% of pairs, respectively. Performance increases further to 99.8% and 100% respectively on WIKIAUTO, a large-scale corpus. Traditional structural baselines show more variable alignment. While T-UNIT achieves strong agreement on WIKIAUTO (>99%), its performance drops substantially on ASSET (86.4%) and it is the second-lowest performing metric on MULTISIM (47.7%). MDL is one of the highest-performing metrics on WIKIAUTO and MULTISIM, but only the fourth-highest performing metric on ASSET. Subordinate clause count is the least reliable, falling to 44.2% on ASSET, 61.4% on WIKIAUTO, and 10.3% on MULTISIM. Sentence length performs well overall.

Performance of all metrics is lower on MULTISIM than on ASSET and WIKIAUTO, suggesting that our metrics work better for English than for German, French, and Italian. We provide a per-language breakdown of the MULTISIM results in appendix section A.6. We find that LACE-CORE has the highest positive complexity delta for German, LACE-FULL has the highest complexity delta for French, and LENGTH is the best predictor for Italian.

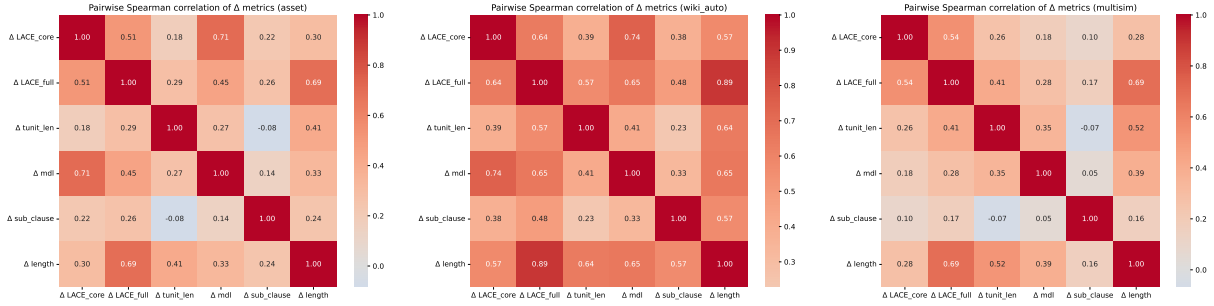
5.2 RQ2: Construct Overlap Among Complexity Metrics

To determine whether LACE captures overlapping or distinct aspects of syntactic complexity, we compute pairwise Spearman (ρ) correlations among LACE-CORE, LACE-FULL, LENGTH, MDL, SUB-CLAUSE, and T-UNIT across the simplification corpora introduced in subsubsection 4.2.1. High correlations indicate redundancy in how metrics rank sentences, as the metrics are capturing mostly the same dimensions of structural



(a) ASSET (b) WIKIAUTO (c) MULTISIM

Figure 5: Proportion of examples with positive complexity deltas ($\Delta > 0$) across metrics for ASSET, WikiAuto, and MultiSim. Higher values indicate that the metric agrees with the human simplification direction (original > simplified).



(a) ASSET (b) WIKIAUTO (c) MULTISIM

Figure 6: Pairwise Spearman correlations between complexity metrics for ASSET, WIKIAUTO, and MULTISIM. Darker colors indicate stronger positive correlations, suggesting that the metrics capture overlapping aspects of syntactic complexity. Lower or inconsistent correlations highlight divergence in what each metric measures.

variation; low correlations indicate that the metrics capture different dimensions of structural variation. Based on their definitions, we expect LACE-FULL to correlate more strongly with surface length, whereas LACE-CORE should provide a more length-independent signal.

Figure 6 reports these correlations across ASSET, WIKIAUTO, and MULTISIM. As expected, LACE-FULL shows strong correlations with sentence length ($\rho = 0.69$ on ASSET, 0.89 on WIKIAUTO, 0.69 on MULTISIM), reflecting its dependence on discourse referent counts. In contrast, LACE-CORE exhibits only low-to-moderate correlations with LENGTH ($\rho = 0.30, 0.57, 0.28$). LACE-CORE correlates most strongly with MDL on ASSET ($\rho = 0.71$), and WIKIAUTO($\rho = 0.74$), consistent with its theoretical grounding in integration cost. On MULTISIM, LACE-CORE correlates most strongly with LACE-FULL and demonstrates a lack of significant overlap with the baseline metrics, suggesting that its overlap with MDL is specific to English. The baseline metrics show heterogeneous relationships: MDL and T-UNIT display moderate-to-high correlations with length but lower correlations with eachother. SUB-CLAUSE is the weakest and most inconsistent signal, displaying negative correlations with T-UNIT on ASSET and MULTISIM and a small positive correlation with T-UNIT on WIKIAUTO. This suggests that it captures a distinct but unreliable structural dimension.

To summarize, LACE-FULL, which incorporates discourse referent cost, achieves consistently high alignment, rivaling or surpassing surface length metrics. LACE-CORE correctly ranks the majority of complex-simple pairs on ASSET, achieves almost perfect performance on WIKIAUTO, and outperforms clause-based baselines T-UNIT and SUB-CLAUSE on MULTISIM. LACE-CORE complements surface proxies by isolating dependency-distance–driven structural load on English text in ASSET and WIKIAUTO, while LACE-FULL captures a larger share of length- and discourse-density–associated variation and is correspondingly more sensitive to global sentence-level load. This distinction motivates evaluating both variants: LACE-FULL for length-sensitive judgments and LACE-CORE for isolating structure that is not attributable to LENGTH.

5.3 RQ3: Predicting Downstream Model Difficulty

Our third research question asks whether syntactic complexity metrics can *predict model failures* on QA benchmarks. To evaluate this, we first analyze whether model accuracy is worse on examples deemed more complex by our metrics (section 5.3.1). We then investigate whether model surprisal (specifically, *peak surprisal*) on syntactically complex sentences differs between instruct and base variants of the model (section 5.3.2). We evaluate several base and instruction-tuned model

pairs: **LLaMa (8B), Mistral (8B), OLMo (7B)** and **Qwen (7B)**.[3]

5.3.1 Accuracy Trends

We perform a controlled Monte Carlo randomization test (Nichols and Holmes, 2002) that isolates the most syntactically complex examples according to each metric and compares model performance on this subset against randomly sampled subsets of equal size, allowing us to assess whether higher syntactic load directly corresponds to reduced accuracy. For each dataset, we compute syntactic complexity using each of the metrics and select the top 20% most complex examples as the *hard* subset. We then repeatedly sample random subsets of equal size and compute the performance difference $\Delta = \bar{Y}_{\text{random}} - \bar{Y}_{\text{hard}}$, where $Y \in \{0, 1\}$ denotes example-level model correctness and $\bar{Y}$ is the mean correctness (accuracy) over the corresponding subset. Positive Δ values indicate worse performance on syntactically complex instances, implying that the metric has isolated examples that successfully predict model difficulty. We conduct experiments using the benchmarks introduced in subsubsection 4.2.2.

Table 1 summarizes the change in model accuracy (ΔAccuracy) between syntactically hard and random subsets across the benchmarks introduced in section 4.2.2. Overall, we observe that syntactic complexity exerts markedly different effects depending on the task . SuperGPQA and MMLU exhibit broadly significant positive accuracy deltas, indicating systematic performance degradation on syntactically complex questions. GSM8K results reveal fewer significant positive deltas than SuperGPQA and MMLU, but the majority of deltas are still significant and positive. Both LACE-Core and LACE-Full yield statistically significant accuracy drops for all models, demonstrating that LACE reliably isolates structurally challenging questions in both datasets. On GSM8K, half of the deltas for LACE-Full and three out of four deltas for LACE-Core are positive and significant, demonstrating a weaker yet still significant trend of LACE isolating more complex inputs. All baseline metrics, with the exception of T-Unit, yield significant accuracy drops on almost all models for all datasets. The largest accuracy drops alternate between length-sensitive metrics, specifically LACE-Full and Length, across model families on all three datasets, indicating that accumulated discourse referents and sequence length are dominant sources of difficulty in these settings. The significant results observed for LACE-Core on all models except for OLMo on GSM8K also point to localized structural load modulating accuracy degradation on each dataset.

5.3.2 Peak Surprisal and Instruction Tuning

To complement accuracy-based analysis, we examine model surprisal during generation to assess whether instruction tuning alters sensitivity to syntactic complexity beyond task performance. Surprisal is a measurement of how uncertain a model is about a token in a text. The less probable the token is, the more "surprised" the model is by it. We compare *base* and *instruction-tuned* variants of each model on identical inputs. At generation step t, token-level surprisal is defined as:

$$S_t = -\log p_t(w_t), \tag{7}$$

where $p_t(w_t)$ is the probability assigned to the generated token. We summarize sequence-level uncertainty using *peak surprisal*:

$$S_{\text{peak}} = \max_{1 \le t \le T} S_t. \tag{8}$$

S_{peak} captures the point of maximal predictive uncertainty during generation. Although generation-time surprisal may be influenced by decoding constraints(inference-time constraints arising from decoding algorithms, their hyperparameters, and imposed output or prompt formats), prior work shows it remains sensitive to syntactic structure (Gauthier et al., 2020), motivating its use as a proxy for processing difficulty. For each dataset, we compute S_{peak} for outputs generated by base and instruction-tuned variants in response to the same QA inputs.

We analyze peak surprisal as a function of syntactic complexity by computing linear fits between the two (visualized in Figures 16 to 18). The intercept of each line of best fit reflects a model's *baseline surprisal* on syntactically simple inputs, while the slope of the line captures the strength of the relationship between model surprisal and the syntactic complexity of its input. A positive slope indicates that the model becomes *more surprised* as syntactic complexity increases, while a negative slope indicates that the model becomes *less surprised* as syntactic complexity increases. Base-to-instruct shifts in this slope–intercept space are visualized using *Structural Surprisal Dual-Space* plots, where each point plotted represents a line of best fit for a model with its slope on the x-axis and its intercept on the y-axis. Because points represent linear functions, they are plotted in a *dual space*, hence the name of our plots. We plot points for both base and instruct variants of models in the same space and connect the points in each pair via a dotted line.

Figures 7 to 9 display dual space plots for SuperGPQA, MMLU, and GSM8K, respectively. Instruction-tuning induces a consistent downward shift in intercept across models on all three datasets, indicating lower baseline surprisal on syntactically simpler inputs. Instruction tuning generally increases the alignment between residual surprisal variation and localized syntactic complexity as measured by LACE-Core, indicating that while models are more confident overall, residual variation in output surprisal exhibits a stronger statistical association with localized syntactic complexity. Changes in slope differ among models and datasets. For LACE-Core, slopes increase unanimously on SuperGPQA and GSM8K, indicating that peak surprisal increases

[3] Prompting templates are detailed in subsection A.7.

Model	LACE-Core	LACE-Full	T-Unit	MDL	Length	Sub-Clause
			SuperGPQA			
LLaMa 3-8B	**0.040** ± 0.004	**0.056** ± 0.005	**0.052** ± 0.004	**0.027** ± 0.005	**0.054** ± 0.005	**0.045** ± 0.005
Mistral-7B	**0.016** ± 0.005	**0.026** ± 0.005	**0.023** ± 0.005	**0.010** ± 0.005	**0.027** ± 0.005	**0.019** ± 0.005
OLMo 3-7B	**0.030** ± 0.005	**0.060** ± 0.005	**0.047** ± 0.005	**0.030** ± 0.005	**0.063** ± 0.005	**0.044** ± 0.005
Qwen 2.5-7B	**0.017** ± 0.005	**0.044** ± 0.005	**0.038** ± 0.006	0.003 ± 0.005	**0.039** ± 0.006	**0.028** ± 0.005
			MMLU			
LLaMa 3-8B	**0.036** ± 0.008	**0.074** ± 0.008	**0.018** ± 0.008	**0.036** ± 0.008	**0.079** ± 0.008	**0.073** ± 0.008
Mistral-7B	**0.054** ± 0.008	**0.067** ± 0.008	-0.000 ± 0.008	**0.067** ± 0.008	**0.069** ± 0.009	**0.073** ± 0.008
OLMo 3-7B	**0.018** ± 0.008	**0.111** ± 0.008	-0.002 ± 0.008	**0.048** ± 0.008	**0.116** ± 0.008	**0.115** ± 0.009
Qwen 2.5-7B	**0.022** ± 0.008	**0.124** ± 0.007	-0.001 ± 0.007	**0.044** ± 0.008	**0.124** ± 0.007	**0.119** ± 0.008
			GSM8K			
LLaMa 3-8B	**0.041** ± 0.023	**0.073** ± 0.023	0.020 ± 0.022	**0.041** ± 0.023	**0.083** ± 0.022	**0.057** ± 0.023
Mistral-7B	**0.048** ± 0.027	**0.116** ± 0.027	0.014 ± 0.028	**0.071** ± 0.027	**0.115** ± 0.027	**0.108** ± 0.026
OLMo 3-7B	0.011 ± 0.020	0.022 ± 0.019	-0.027 ± 0.019	-0.0005 ± 0.019	**0.064** ± 0.019	0.018 ± 0.020
Qwen 2.5-7B	**0.039** ± 0.021	0.023 ± 0.021	-0.004 ± 0.020	**0.050** ± 0.021	**0.050** ± 0.021	**0.038** ± 0.021

Table 1: Change in task-level correctness (Δ, ± SE) between syntactically hard and random subsets. Bold values denote statistically significant effects ($p < 0.05$). Underlined values indicate the largest degradation per model–dataset pair. *All models are instruction-tuned.*

† Full distributional visualizations (box plots and example-level analyses) are provided in Section A.1.

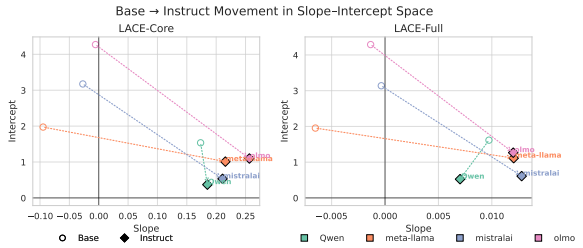


Figure 7: Base-to-instruct movement in surprisal dual space on SuperGPQA for LACE-Core and LACE-Full.

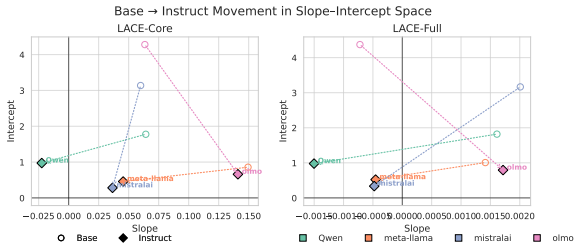


Figure 8: Base-to-instruct movement in surprisal dual space on MMLU for LACE-Core and LACE-Full.

with localized syntactic complexity after instruction-tuning. On MMLU, however, LACE-Core slopes decrease after instruction-tuning for all models except OLMo, indicating weakened coupling between localized syntactic complexity and peak surprisal on this dataset. Slopes for LACE-Full vary depending on the model and dataset. For SuperGPQA, most models exhibit modest increases in sensitivity to accumulated discourse-level complexity, while Qwen shows a reduction in slope. On MMLU, slopes remain near zero or decrease for most models, with OLMo being a notable exception. For GSM8K, slopes barely increase for LLaMa and OLMo, increase more noticeably for Mistral, and decrease for Qwen. Overall, changes in slope are larger for LACE-CORE than for LACE-FULL on each dataset, indicating stronger post-training amplification of localized rather than discourse-level complexity.

6 Conclusion

This work introduces and evaluates LACE, a new computable metric for syntactic complexity based on Dependency Locality Theory (DLT), with two variants: LACE-Core and LACE-Full. The evaluation, spanning human simplification judgments, existing complexity baselines, and LLM reasoning tasks, shows that LACE-Full effectively captures discourse-level complexity while LACE-Core isolates localized syntactic load. These metrics provide a detailed tool for analyzing model robustness under syntactic stress. LACE-Full is ideal for discourse-heavy tasks, and LACE-Core helps identify specific structural failure modes, laying the groundwork for future research in readability and model evaluation.

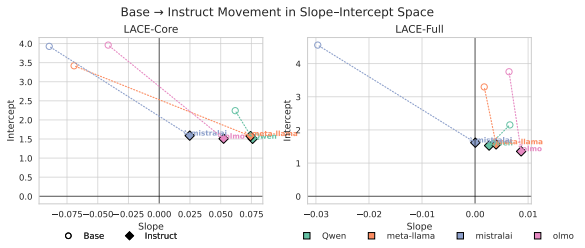


Figure 9: Base-to-instruct movement in surprisal dual space on GSM8K for LACE-Core and LACE-Full.

7 Ethical Considerations

Our work does not involve training, finetuning, or deploying language models, so concerns pertaining to releasing

new models are not applicable. All of the models we use are open-source and the datasets we use are publicly available. We do not conduct studies with human participants, so there are no concerns about experiments with human subjects.

8 Limitations

Reported analyses are correlational in nature and do not establish causal mechanisms underlying model behavior. While the observed relationships between syntactic complexity and performance are consistent with theoretical expectations from Dependency Locality Theory, they do not directly reveal how such structural load is represented or processed within language models.

Another limitation concerns our reliance on automatic dependency parses to estimate DLT-based complexity measures. Although modern parsers are generally reliable, parsing errors may introduce noise into complexity estimates, particularly for long or syntactically irregular inputs. Such noise would be expected to attenuate observed effects rather than systematically favor LACE over alternative metrics.

Finally, our evaluation is conducted on existing benchmarks that, while widely used and carefully selected to span diverse reasoning settings, may not fully reflect all downstream applications of language models. As with any benchmark-driven analysis, dataset-specific properties and potential artifacts may influence observed trends.

References

Steven P. Abney and Mark Johnson. 1991. Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20(3):233–250.

Suha S. Al-Thanyyan and Aqil M. Azmi. 2021. Automated text simplification: A survey. *ACM Comput. Surv.*, 54(2).

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Samuel Joseph Amouyal, Aya Meltzer-Asscher, and Jonathan Berant. 2025. When the LM misunderstood the human chuckled: Analyzing garden path effects in humans and language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8235–8253, Vienna, Austria. Association for Computational Linguistics.

Thomas G. Bever. 2013. The cognitive basis for linguistic structures1.

Noam Chomsky and George A. Miller. 1968. Introduction to the formal analysis of natural languages.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Pablo J. Diego-Simón, Emmanuel Chemla, Jean-Rémi King, and Yair Lakretz. 2025. Probing syntax in large language models: Successes and remaining challenges. *Preprint*, arXiv:2508.03211.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32, Salt Fork, Ohio, USA. Association for Computational Linguistics.

Richard Futrell, Roger Philip Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96:371 – 412.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Edward Gibson. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Edward Gibson. 2000. The Dependency Locality Theory: A distance-based theory of linguistic complexity. In *Image, language, brain: Papers from the first mind articulation project symposium*, pages 95–126, Cambridge, MA. MIT Press.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Richard A. Hudson. 1996. The difficulty of (so-called) self-embedded structures *.

Kellogg W. Hunt. 1965. Grammatical structures written at three grade levels. ncte research report no. 3.

Tovah Irwin, Kyra Wilson, and Alec Marantz. 2023. BERT shows garden path effects. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3220–3232, Dubrovnik, Croatia. Association for Computational Linguistics.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960, Online. Association for Computational Linguistics.

John Kimball. 1973. Seven principles of surface structure parsing in natural language. *Cognition*, 2:15–47.

Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Andrew Li, Xianle Feng, Siddhant Narang, Austin Peng, Tianle Cai, Raj Sanjay Shah, and Sashank Varma. 2024. Incremental comprehension of garden-path sentences by large language models: Semantic interpretation, syntactic re-analysis, and attention. *Preprint*, arXiv:2405.16042.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *The Journal of Cognitive Science*, 9:159–191.

Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15:474–496.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

George A. Miller and Noam Chomsky. 1963. Finitary models of language users.

George A. Miller and Stephen D. Isard. 1964. Free recall of self-embedded english sentences. *Inf. Control.*, 7:292–303.

Ananjan Nandi, Christopher D Manning, and Shikhar Murty. 2025. Sneaking syntax into transformer language models with tree regularization. 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)*, pages 8006–8024, Albuquerque, New Mexico. Association for Computational Linguistics.

Thomas E. Nichols and Andrew P. Holmes. 2002. Nonparametric permutation tests for functional neuroimaging: A primer with examples. *Human Brain Mapping*, 15(1):1–25.

Michael Ryan, Tarek Naous, and Wei Xu. 2023. Revisiting non-English text simplification: A unified multilingual benchmark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4898–4927, Toronto, Canada. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual SMT system for paraphrase generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–42, Jeju Island, Korea. Association for Computational Linguistics.

P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shawn Gavin, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, and 78 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *Preprint*, arXiv:2502.14739.

David Temperley. 2007. Minimization of dependency length in written english. *Cognition*, 105(2):300–333.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 181–190, Florence, Italy. Association for Computational Linguistics.

Dane A Williamson, Yangfeng Ji, and Matthew B. Dwyer. 2025. Syntactic blind spots: How misalignment leads to LLMs' mathematical errors. In *Proceedings of The 3rd Workshop on Mathematical Natural Language Processing (MathNLP 2025)*, pages 1–14, Suzhou, China. Association for Computational Linguistics.

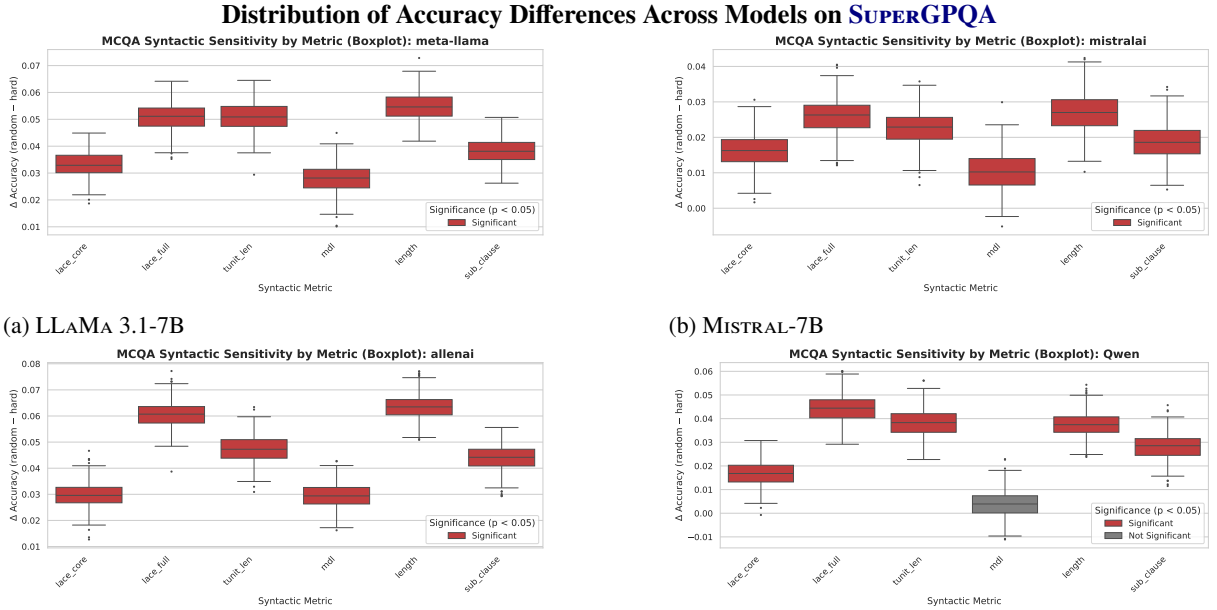
Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Victor H. Yngve. 1960. A model and an hypothesis for language structure.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *CoRR*, abs/1909.02622.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.



(a) LLaMa 3.1-7B

(b) Mistral-7B

(c) OLMo 3-7B

(d) Qwen 2.5-7B

Figure 10: Distribution of ($\Delta = \bar{Y}_{\text{random}} - \bar{Y}_{\text{hard}}$) across syntactic complexity metrics on SuperGPQA. Each subplot summarizes delta (y-axis) over 1,000 Monte Carlo draws for a single model for the top-20% most complex examples as determined by each metric (x-axis).

A Appendix

A.1 Benchmark Performance Variability

We report distributional statistics underlying the summary results for benchmark accuracy in § 5.3.1. For Monte Carlo stress tests, performance differences are computed over 1,000 random draws, and we visualize the resulting distributions using box plots. Central tendencies reported in the main text correspond to the mean of these distributions, with variability summarized via standard errors or confidence intervals as noted. Figures 10 to 12 visualize the appropriate descriptive statistics. For experimental details see subsubsection A.4.1.

A.2 Peak Surprisal Distributions

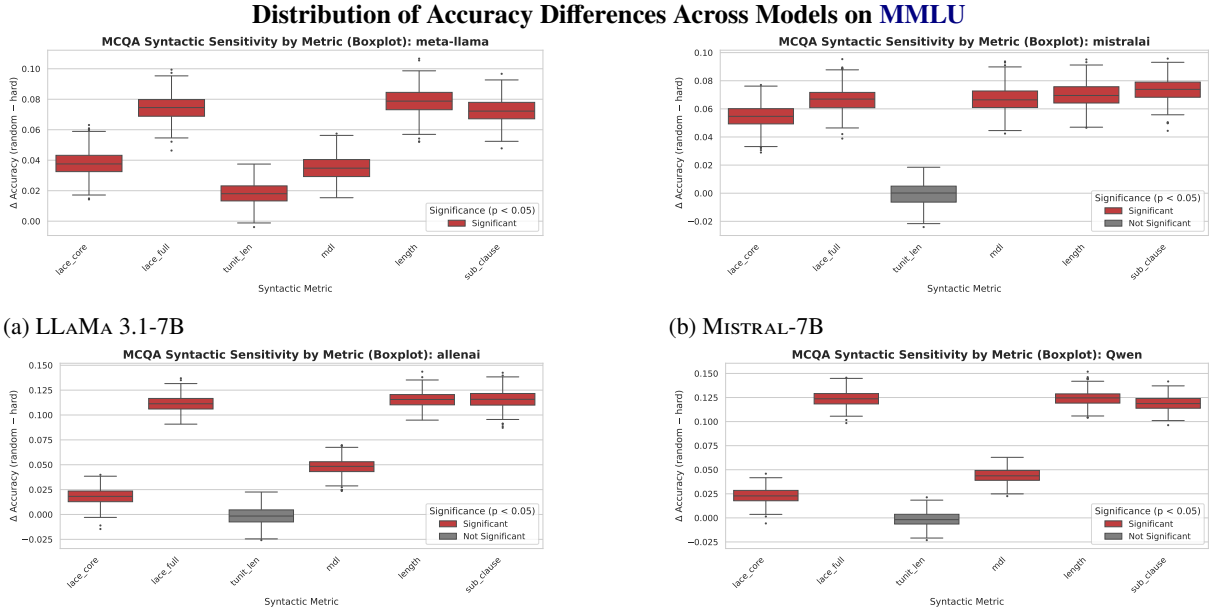
Peak surprisal captures the point of maximal predictive uncertainty during generation and is commonly interpreted as a signal of processing difficulty or expectation violation. Figures 13 to 15 present paired distributions of peak surprisal for base and instruction-tuned variants of each model, computed over identical input sets and restricted to valid responses only.

These distributions provide a complementary view to the slope-intercept analysis in § 5.3.2. Whereas the regression analysis characterizes how surprisal scales with increasing syntactic complexity, the distributional plots visualize how instruction tuning reshapes the overall surprisal landscape. In particular, shifts in the mode and mass of the distribution reflect changes in baseline uncertainty, while changes in tail behavior indicate how extreme uncertainty events are redistributed after instruction tuning. Across datasets, instruction-tuned models consistently exhibit a leftward shift in the surprisal distribution relative to their base counterparts, indicating reduced baseline surprisal on structurally simpler inputs. At the same time, overlap between base and instruction-tuned distributions varies by model and dataset, highlighting heterogeneity in how post-training modulates predictive uncertainty. These plots contextualize the regression results by making visible the full distributional effects underlying changes in slope and intercept.

A.3 Effects of Instruction-Tuning on DLT Sensitivity

This subsection provides example-level visualizations illustrating how instruction tuning alters model sensitivity to syntactic complexity as measured by DLT-based metrics. Figures 16 to 18 show representative scatter plots of peak surprisal as a function of syntactic complexity, with separate linear fits for base and instruction-tuned variants of the same model.

Each point corresponds to an individual input, plotted by its syntactic complexity (x-axis) and peak surprisal (y-axis). Solid lines denote ordinary least squares fits computed separately for base and instruction-tuned models. The annotated slopes quantify surprisal sensitivity to increasing syntactic load. These plots make explicit two systematic effects summarized in the main text. First, instruction tuning consistently reduces baseline surprisal, reflected in downward shifts of the fitted intercepts. Second, and more importantly, instruction tuning often increases the slope of the surprisal-complexity relationship, indicating heightened sensitivity to syntactic structure. This slope increase is most pronounced and consistent for LACE-Core, suggesting that post-training



(a) LLaMa 3.1-7B

(b) Mistral-7B

(c) OLMo 3-7B

(d) Qwen 2.5-7B

Figure 11: Distribution of ($\Delta = \bar{Y}_{\text{random}} - \bar{Y}_{\text{hard}}$) across syntactic complexity metrics on MMLU.

sharpens responsiveness to localized dependency-based complexity rather than to length- or discourse-driven complexity captured by LACE-Full. Effects are heterogenous, and often task/model specific.

A.4 Experimental Setup and Hyperparameters

This subsection documents experimental parameters and design choices required for reproducibility. Most parameters in our experiments are *fixed by protocol or theory*, rather than tuned via hyperparameter search.

A.4.1 Monte Carlo Stress Tests.

For the downstream QA evaluation (§5.3.1), we assess whether syntactic complexity metrics isolate systematically harder examples using Monte Carlo stress tests. For each dataset and complexity metric, we select the ***top 20% most complex examples*** to form a syntactically hard subset. This threshold balances isolating high-complexity inputs while retaining sufficient sample size for stable estimation.We verified that qualitative trends remain stable for nearby thresholds (e.g., 20–30%). For each hard subset, we draw **1,000 random subsets** of equal size and compute the performance difference:

$$\Delta = \bar{Y}_{\text{random}} - \bar{Y}_{\text{hard}},$$

where Y denotes task correctness. All reported confidence intervals and significance tests are derived from this Monte Carlo distribution.

A.4.2 Surprisal-Complexity Regression.

Slope and intercept values in the surprisal dual-space analysis (§5.3.2) are obtained via **ordinary least squares (OLS) regression**. For each model, dataset, and complexity metric, we fit peak surprisal as a linear function of syntactic complexity:

$$S_{\text{peak}} = \alpha + \beta \cdot \text{Complexity}.$$

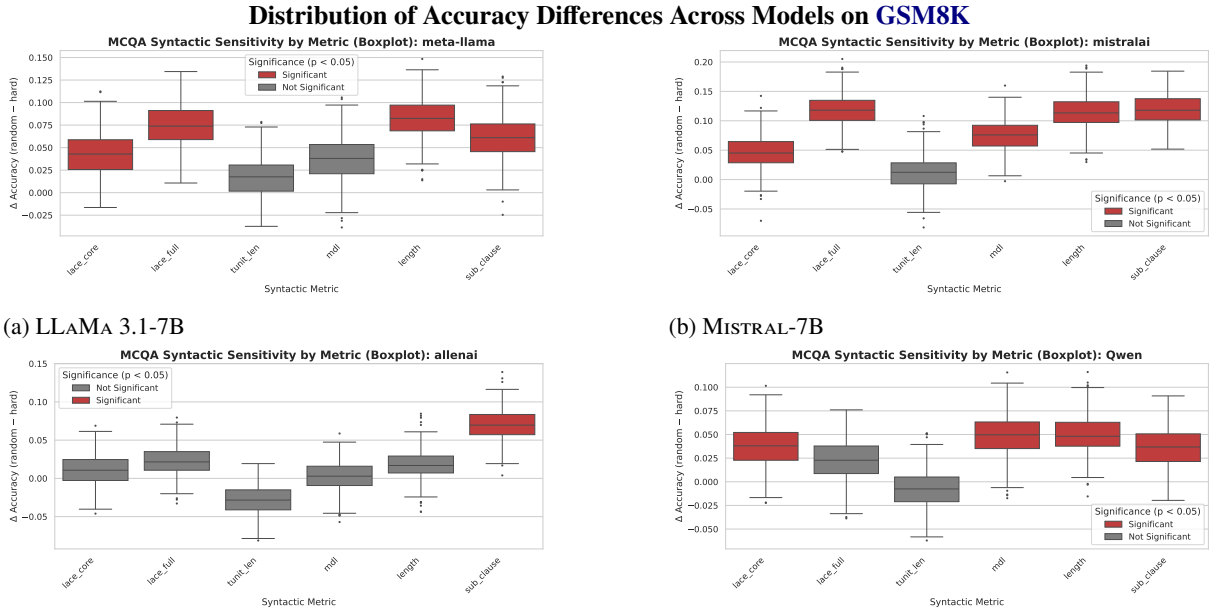
The intercept α reflects baseline surprisal at low syntactic complexity, while the slope β captures sensitivity to increasing structural load. No regularization or hyperparameter tuning is applied. Base and instruction-tuned models are evaluated on identical input sets to ensure comparability.

Parsing, Tokenization, and Decoding. Dependency parses are generated deterministically from *spaCy's* pretrained English dependency parser. For German, French, and Italian, *spaCy* parsers specific to each language are used. Sentence length is computed using whitespace tokenization for simplification corpora and each model's native tokenizer for QA benchmarks. All prompting experiments use greedy decoding (temperature = 0) with no sampling. Model architectures and decoding parameters are listed in Table 2.

Model	Params	Max	Temp	Top-p	Sample
LLaMa 3.3	8B	8192	0.0	1.0	False
Mistral	7B	8192	0.0	1.0	False
OLMo 3	7B	8192	0.0	1.0	False
Qwen 2.5	7B	8192	0.0	1.0	False
Gemma	7B	8192	0.0	1.0	False
Qwen 2	72B	8192	0.0	1.0	False
LLaMa 3	72B	8192	0.0	1.0	False

(a) QA models and decoding hyperparameters. All models use greedy decoding (temperature = 0, no sampling).

Table 2: LLMs used in experiments.



(a) LLaMa 3.1-7B

(b) Mistral-7B

(c) OLMo 3-7B

(d) Qwen 2.5-7B

Figure 12: Distribution of ($\Delta = \bar{Y}_{\text{random}} - \bar{Y}_{\text{hard}}$) across syntactic complexity metrics on GSM8K.

Dataset	RQ	# Ex.	Split	Format	Description
ASSET	RQ1, RQ2	2,000	Validation	Paired	Human-written sentence simplifications with multiple references
WikiAuto	RQ1, RQ2	118,074	Test	Paired	Automatically aligned Wikipedia–Simple Wikipedia sentence pairs
MultiSim	RQ1, RQ2	~2,500	Test	Paired	Simplified text pairs across 12 languages.
SuperGPQA	RQ3	26,529	Test	MCQA	Graduate-level multiple-choice questions across academic domains
MMLU	RQ3	14,042	Test	MCQA	Broad multi-domain academic and professional knowledge benchmark
GSM8K	RQ3	1,319	Test	Open-ended	Grade-school math word problems with single- and multi-step reasoning

Table 3: Summary of datasets used across research questions. All prompting experiments are zero-shot.

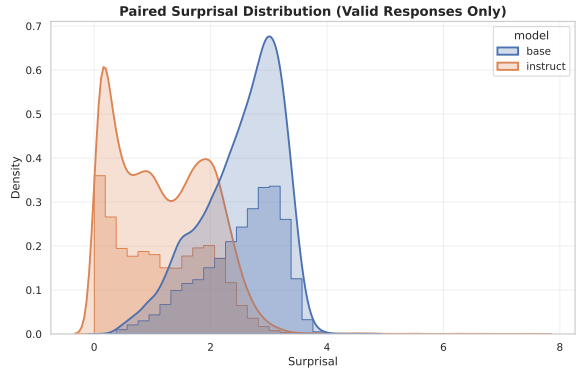
A.5 Dataset Statistics and Splits

This appendix summarizes the datasets used across all experiments, including the number of examples and evaluation splits, (see Table 3). All analyses in this paper are conducted in a *zero-shot evaluation* setting. Unless otherwise noted, results are computed on the official test splits provided by each benchmark.

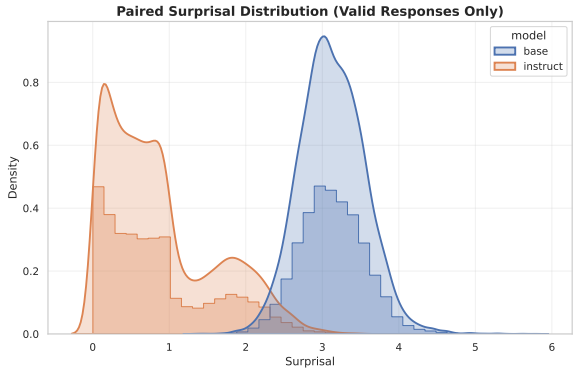
A.6 Individual Language Results on MultiSim

Figure 19 displays the positive complexity deltas across metrics for German, French, and Italian individually. On German, LACE-Core achieves the highest positive complexity delta whereas LACE-Full and Length display lower performance. This suggests that simplification in the German dataset targets more localized features as opposed to sentence length or the number of new discourse referents. The opposite is observed for French: LACE-Full achieves the highest positive complexity delta, and Length also displays a relatively strong signal. Results differ significantly on Italian, where Length achieves the highest delta value. MDL and T-Unit achieve similar performance to one another on German and French, with both metrics performing relatively high. On Italian, however, MDL performs significantly better than T-Unit and rivals the performance of Length. Sub-Clause remains the lowest-performing metric across all three languages.

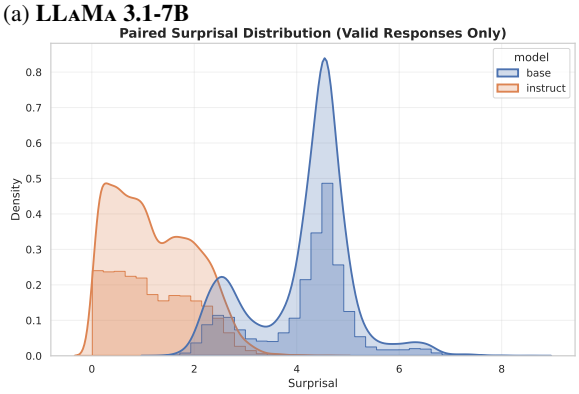
Figure 20 displays the pairwise Spearman correlations between complexity metrics for each language. LACE-Full correlates more highly with Length than LACE-Core on all three languages, highlighting its increased sensitivity to length compared to LACE-Core. LACE-Core has low-to-moderate overlap with the other baseline metrics on German and French, and low-to-no overlap with the baselines on Italian. LACE-Core is most highly correlated with LACE-Full on all three languages, suggesting that it captures distinct features from the baselines.



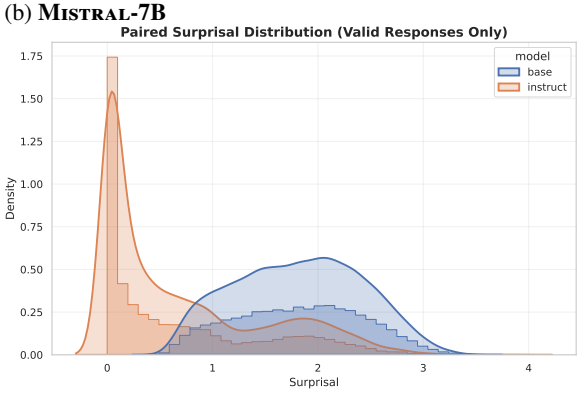
(a) **LLaMa 3.1-7B**



(b) **Mistral-7B**



(c) **OLMo 3-7B**



(d) **Qwen 2.5-7B**

Figure 13: Base vs Instruct surprisal distributions across models on SuperGPQA

A.7 Prompting Templates

§ Sections A.7.1 to A.7.3 document the prompting templates used in our evaluation pipeline to ensure reproducibility and clarity across datasets and model classes. We distinguish between *role-conditioned chat prompting* for instruction-tuned models and plain-text prompting for base models, reflecting the different input interfaces these model classes are trained to expect.

Crucially, the instructional and question content is held *constant across prompting formats*: the chat-based and plain-text templates differ only in surface representation (i.e., role annotations and formatting), not in the information provided to the model. No additional guidance, examples, or task-specific cues are introduced for instruction-tuned models beyond those present in the corresponding base-model prompts.

For instruction-tuned models, we adopt a structured chat-style format that separates dataset-level instructions from question content using explicit system and user roles, and constrains model outputs to a parseable JSON schema for evaluation consistency. In all cases, the required answer is a single multiple-choice option label (e.g., A–D), with the JSON wrapper serving solely as an output-formatting constraint.

For base models, which do not support conversational roles, we instead use a single plain-text prompt that concatenates the same instruction (when present), question, and answer options, followed by an explicit **Answer:** completion cue to elicit the *same* single option label. Unless otherwise noted, these templates are applied uniformly within each dataset.

A.7.1 SuperGPQA

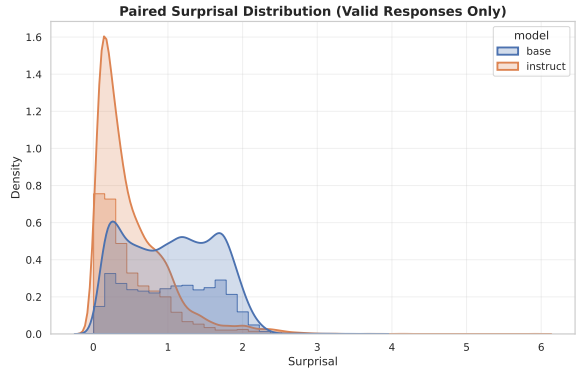
Figures 21 and 22 illustrate the prompting templates used for base and instruction-tuned models, respectively, on SuperGPQA. While the surface form differs between plain-text and chat-based prompts, both templates present the same multiple-choice question and options and constrain the model output to a single option letter.

A.7.2 MMLU

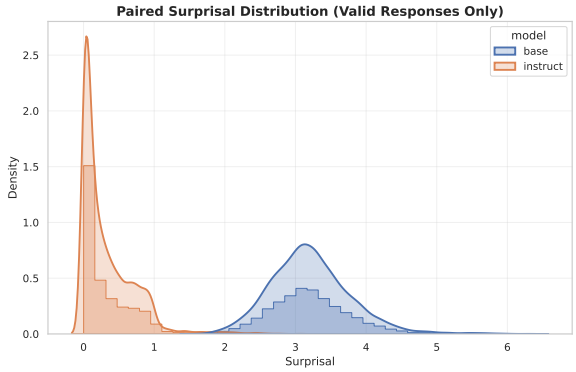
Figures 23 and 24 illustrate the prompting templates used for base and instruction-tuned models, respectively, on MMLU. As with SuperGPQA, instruction-tuned models receive a chat-style prompt with an explicit system instruction, whereas base models are prompted using a single plain-text template terminating in an **Answer:** cue.

A.7.3 GSM8K

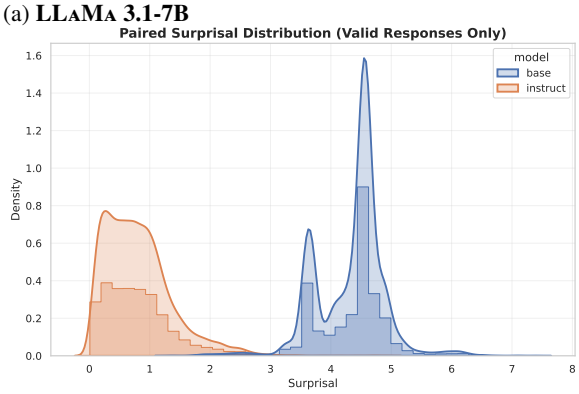
Figures 25 and 26 illustrate the prompting templates used for base and instruction-tuned models, respectively, on GSM8K. As with SuperGPQA and MMLU, instruction-tuned models receive the text in a chat-style prompt and base models receive all parts of the input as a plain-text template with the addition of an **Answer:** cue at the end. Differently from SuperGPQA and MMLU, both base and instruct prompts for GSM8K ask the model



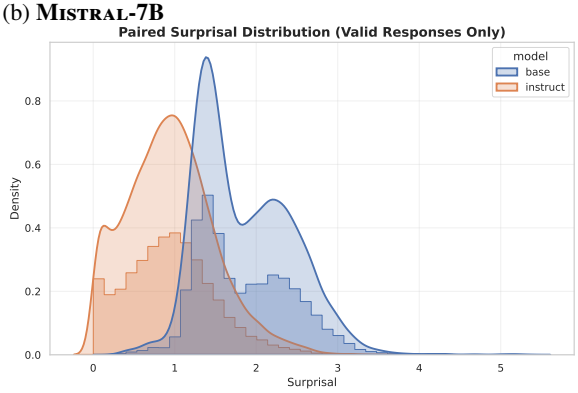
(a) **LLaMa 3.1-7B**



(b) **Mistral-7B**



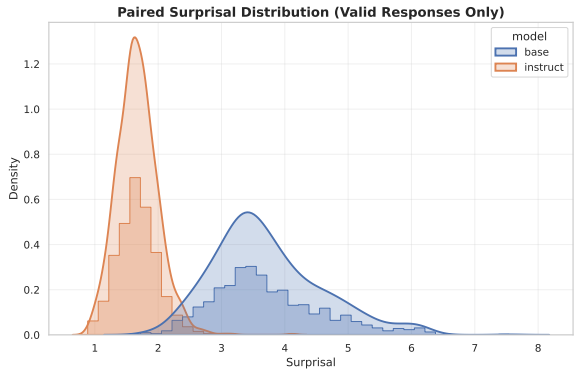
(c) **OLMo 3-7B**



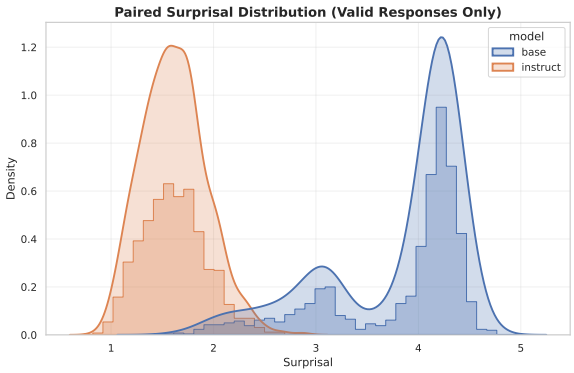
(d) **Qwen 2.5-7B**

Figure 14: Base vs Instruct surprisal distributions across models on MMLU

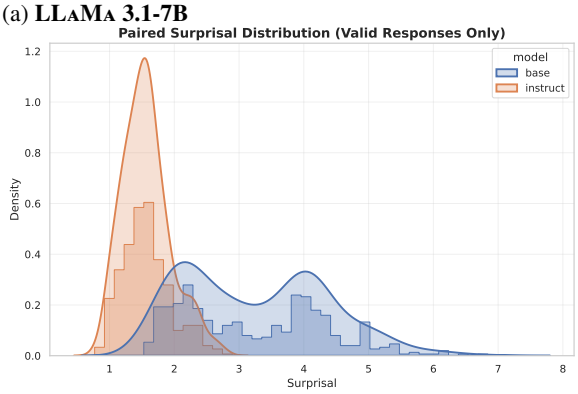
to respond using JSON format. If the answer is not in JSON format the first number in the answer is chosen.



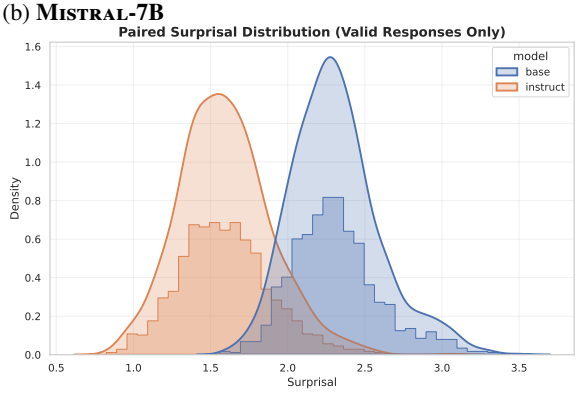
(a) **LLaMa 3.1-7B**



(b) **Mistral-7B**

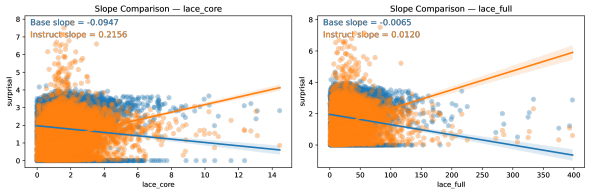


(c) **OLMo 3-7B**

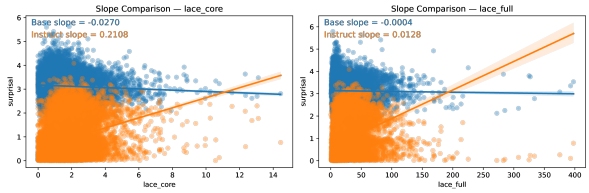


(d) **Qwen 2.5-7B**

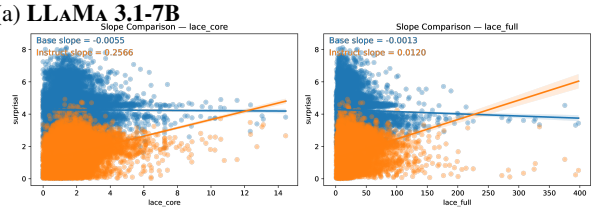
Figure 15: Base vs Instruct surprisal distributions across models on GSM8K



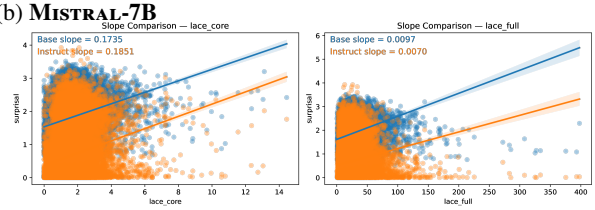
(a) **LLaMa 3.1-7B**



(b) **Mistral-7B**

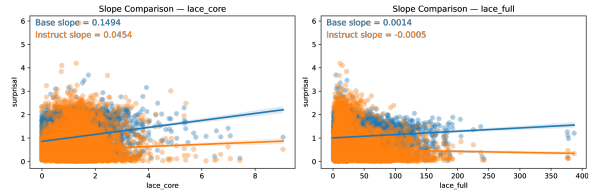


(c) **OLMo 3-7B**

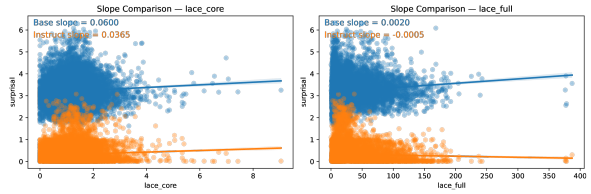


(d) **Qwen 2.5-7B**

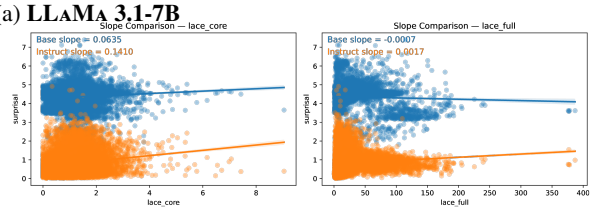
Figure 16: Base vs. instruction-tuned sensitivity across models on SuperGPQA, measured via slope shifts for LACE-Core and LACE-Full.



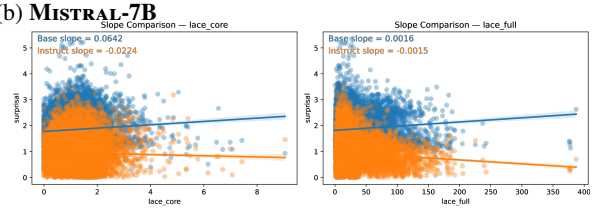
(a) **LLaMa 3.1-7B**



(b) **Mistral-7B**

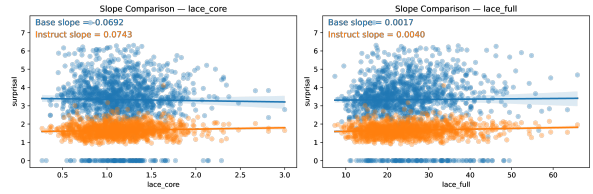


(c) **OLMo 3-7B**

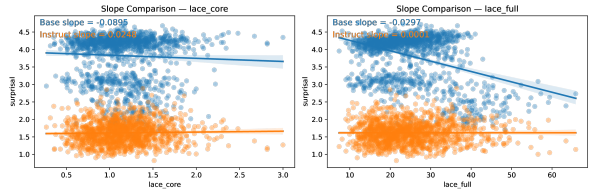


(d) **Qwen 2.5-7B**

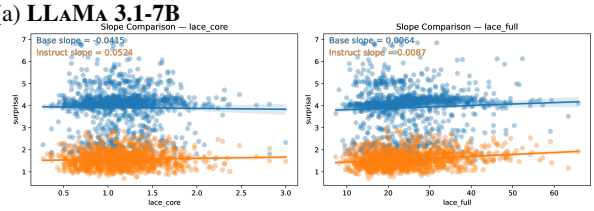
Figure 17: Base vs. instruction-tuned sensitivity across models on MMLU, measured via slope shifts for LACE-Core and LACE-Full.



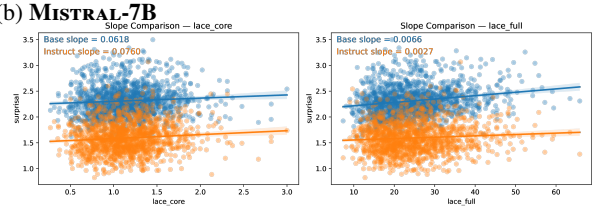
(a) **LLaMa 3.1-7B**



(b) **Mistral-7B**

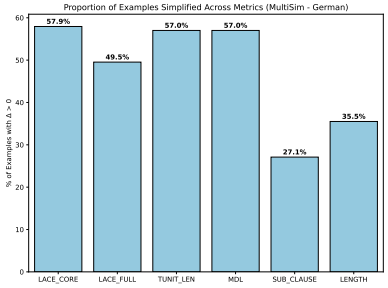


(c) **OLMo 3-7B**

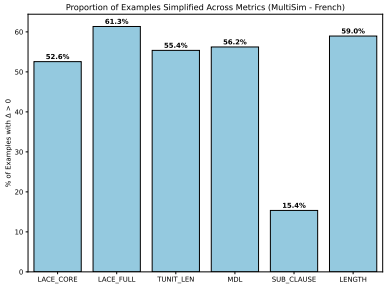


(d) **Qwen 2.5-7B**

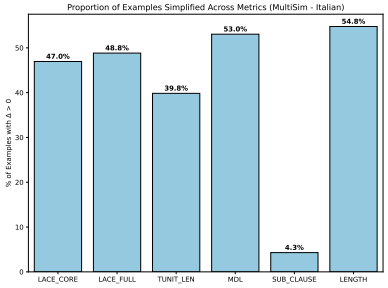
Figure 18: Base vs. instruction-tuned sensitivity across models on GSM8K, measured via slope shifts for LACE-Core and LACE-Full.



(a) German

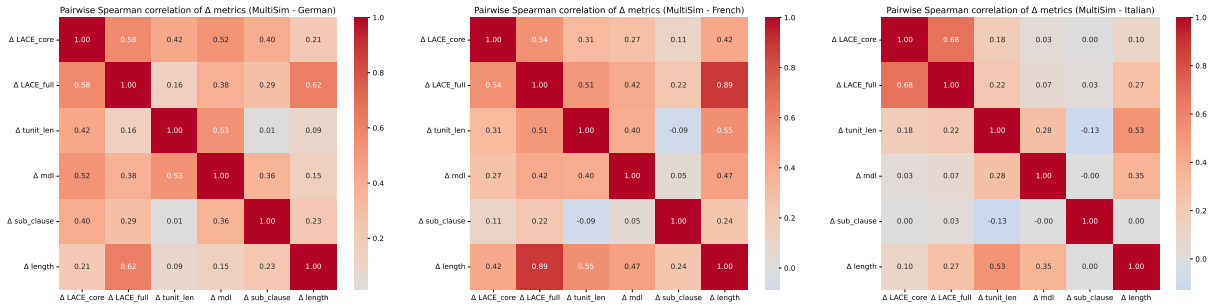


(b) French



(c) Italian

Figure 19: Proportion of examples with positive complexity deltas ($\Delta > 0$) across German, French, and Italian for MultiSim. Higher values indicate that the metric agrees with the human simplification direction (original > simplified).



(a) German (b) French (c) Italian

Figure 20: Pairwise Spearman correlations between complexity metrics for German, French, and Italian on MULTISIM. Darker colors indicate stronger positive correlations, suggesting that the metrics capture overlapping aspects of syntactic complexity. Lower or inconsistent correlations highlight divergence in what each metric measures.

SUPERGPQA Base Prompt

Prompt construction (per example):

PROMPT
Answer the multiple-choice question based on the given context. Return the answer in JSON format: { `"answer": "<option_letter>"` }.

Question:
The common-mode rejection ratio of the first stage amplification circuit in a three-op-amp differential circuit is determined by ( ).

Options:
A. the absolute value of the difference in the common-mode rejection ratio of A1 and A2 themselves
B. all of the above
C. the average of A1 and A2's common-mode rejection ratios
D. the sum of A1 and A2's common-mode rejection ratios
E. the product of A1 and A2's common-mode rejection ratios
F. the square root of the product of A1 and A2's common-mode rejection ratios
G. the size of A2's common-mode rejection ratio
H. the size of A1's common-mode rejection ratio
I. The difference in the common-mode rejection ratio of A1 and A2 themselves
J. input resistance

Answer:

Figure 21: Illustration of the SUPERGPQA *base* prompting format used in our evaluation pipeline.

SuperGPQA Instruct Prompt (Chat Template)

Prompt construction (per example):

SYSTEM `example.instruction`

USER **Question:**
The common-mode rejection ratio of the first stage amplification circuit in a three-op-amp differential circuit is determined by ( ).

Options:
A. the absolute value of the difference in the common-mode rejection ratio of A1 and A2 themselves
B. all of the above
C. the average of A1 and A2's common-mode rejection ratios
D. the sum of A1 and A2's common-mode rejection ratios
E. the product of A1 and A2's common-mode rejection ratios
F. the square root of the product of A1 and A2's common-mode rejection ratios
G. the size of A2's common-mode rejection ratio
H. the size of A1's common-mode rejection ratio
I. The difference in the common-mode rejection ratio of A1 and A2 themselves
J. input resistance

ASSISTANT `{ "answer": "<option_letter>" }`

Figure 22: Illustration of the SuperGPQA *instruct* prompting format used in our evaluation pipeline.

MMLU Base Prompt

Prompt construction (per example):

PROMPT
Answer the following multiple-choice question.

Question:
A women's action group attempted for many months, unsuccessfully, to reach an agreement with the local professional men's club to admit women to membership. The women's group instituted a suit for a declaratory judgment in federal court to determine whether the men's club was subject to the state's anti-discrimination act. Prior to the elections for city officials, four members of the women's group were sent to picket the offices of the mayor and district attorney, both prominent members of the men's club. Two members walked outside the front of the mayor's office building, carrying signs that read, "The mayor is supposed to serve all the people but his lunch club is for men ONLY. So don't vote for him." The other two pickets walked outside the rear of the district attorney's office building, carrying similar signs, telling the public not to vote for him. This picketing was carried on from 9 A.M. to 5 P.M. The same day, two more pickets were assigned to carry identical signs in front of the mayor's official residence. Two pickets also carried duplicate signs in front of the district attorney's suburban home during the early evening hours. The picketing at all sites was held peacefully without any disturbance. The relevant city ordinances concerning picketing read as follows: "Section 201. No picketing shall be permitted inside of, or on any sidewalk or street immediately adjacent or contiguous to, city hall, without express permission of the mayor. Applications for such permission shall be filed at least three days before such picketing is intended to begin and shall state the purpose, place, and time of the proposed picketing. Section 202. It shall be unlawful for any person to engage in picketing before or about the residence of an individual. Nothing herein shall be deemed to prohibit the holding of a meeting or assembly on any premises used for the discussion of subjects of general public interest." The federal district court will most likely avoid making a decision on the merits of the suit for declaratory judgment because

Options:
A. the case lacks adequate ripeness.
B. there is no case or controversy.
C. the relief sought is essentially for an advisory opinion.
D. the women's group lacks standing.

Answer:

Figure 23: Illustration of the MMLU base prompting format used in our evaluation pipeline.

MMLU Instruct Prompt (Chat Template)

Prompt construction (per example):

SYSTEM Answer the following multiple-choice question. Return the answer in JSON format:
`{ "answer": "<option_letter>" }`.

USER
Question:
A women's action group attempted for many months, unsuccessfully, to reach an agreement with the local professional men's club to admit women to membership. The women's group instituted a suit for a declaratory judgment in federal court to determine whether the men's club was subject to the state's anti-discrimination act. Prior to the elections for city officials, four members of the women's group were sent to picket the offices of the mayor and district attorney, both prominent members of the men's club. Two members walked outside the front of the mayor's office building, carrying signs that read, "The mayor is supposed to serve all the people but his lunch club is for men ONLY. So don't vote for him." The other two pickets walked outside the rear of the district attorney's office building, carrying similar signs, telling the public not to vote for him. This picketing was carried on from 9 A.M. to 5 P.M. The same day, two more pickets were assigned to carry identical signs in front of the mayor's official residence. Two pickets also carried duplicate signs in front of the district attorney's suburban home during the early evening hours. The picketing at all sites was held peacefully without any disturbance. The relevant city ordinances concerning picketing read as follows: "Section 201. No picketing shall be permitted inside of, or on any sidewalk or street immediately adjacent or contiguous to, city hall, without express permission of the mayor. Applications for such permission shall be filed at least three days before such picketing is intended to begin and shall state the purpose, place, and time of the proposed picketing. Section 202. It shall be unlawful for any person to engage in picketing before or about the residence of an individual. Nothing herein shall be deemed to prohibit the holding of a meeting or assembly on any premises used for the discussion of subjects of general public interest." The federal district court will most likely avoid making a decision on the merits of the suit for declaratory judgment because

Options:
A. the case lacks adequate ripeness.
B. there is no case or controversy.
C. the relief sought is essentially for an advisory opinion.
D. the women's group lacks standing.

ASSISTANT `{ "answer": "<option_letter>" }`

Figure 24: Illustration of the MMLU *instruct* prompting format used in our evaluation pipeline.

GSM8K Base Prompt

Prompt construction (per example):

PROMPT
Solve the following math problem step by step and then return your answer in JSON format: { "answer": "<your answer>" }.

Question:
Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts for $5.00. How much does he spend on yogurt over 30 days?

Answer:

Figure 25: Illustration of the GSM8K *base* prompting format used in our evaluation pipeline.

GSM8K Instruct Prompt (Chat Template)

Prompt construction (per example):

SYSTEM Solve the following math problem step by step and then return your answer in JSON format: { "answer": "<your answer>" }.

USER
Question:
Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts for $5.00. How much does he spend on yogurt over 30 days?

ASSISTANT { "answer": "<answer>" }

Figure 26: Illustration of the GSM8K *instruct* prompting format used in our evaluation pipeline.