GLOBENUM: BENCHMARKING CONTEXT-AWARE NUMERICAL FORMATTING REASONING ACROSS CULTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Numerical notation varies across cultures: 1.234 denotes one point two three four in India or the US, but one thousand two hundred and thirty-four in Belgium or Colombia. For many such expressions, the intended value cannot be recovered from the string alone; only context can resolve it. We present GLOBENUM, a benchmark that isolates these genuinely ambiguous representations and tests whether linguistic, geographic, and cultural context enables models to recover their intended value. Across 7 model families, 10 numerical formats, 12 countries, and 7 languages, using both symbolic operations and localised word problems, we find a strong preference for dot-decimal interpretations that persists despite geographic cues. Localisation further exposes a striking asymmetry: cultural context improves accuracy in dot-decimal locales while reducing it by up to 13 points in comma-decimal locales. Even with geographic and formatting cues provided, accuracy on genuinely ambiguous representations reaches only 32.78%. GLOBENUM reframes numerical reasoning as a cultural task: regional context must shape not only how numbers are written, but how their values are interpreted and computed, a distinction current models fail to make reliably.

1 INTRODUCTION

As Large Language Models (LLMs) are increasingly deployed across global user populations (Microsoft AI Economy Institute, 2026), they are routinely tasked with processing numerical data in daily applications, from following recipes to managing finances. In these operational contexts, accurate numerical parsing is essential for predictable system performance. Achieving true global competence requires LLMs to remain robust across diverse regional numerical formats. For instance, when evaluating the sum `1.234 + 5`, an Anglo-American or Indian system interprets the dot as a decimal marker, arriving at 6.234. Conversely, under Belgian or Colombian formatting rules, the dot acts as a thousands separator, converting `1.234` to 1234 and producing a final result of 1239. Misinterpreting these locale-specific conventions introduces severe quantitative errors that shift values by orders of magnitude.

Despite its importance, numerical notation has received little attention, even though it varies drastically across global cultures. At a broad level, significant progress has been made in multilingual and regional adaptation for natural language processing (Hernández-Cano et al., 2026; Dang et al., 2024; Martins et al., 2025), yet model sensitivity to regional numerical formats remains largely unexamined. On the other hand, standard numerical reasoning benchmarks systematically evaluate dimensions such as robustness under different number distributions, problem difficulty, and multi-step reasoning chains (Cobbe et al., 2021; Hendrycks et al., 2021; Sivakumar & Moosavi, 2023), but overlook numerical formatting or rely by default on Anglo-American representations.

Where numerical formatting has been studied, existing evaluations leave key real world factors unaddressed. While recent work on LLM numeracy confirms that models are sensitive to script and formatting variations (Reddy et al., 2026), these benchmarks rely strictly on isolated symbolic equations. Consequently, they fail to examine how natural language context in practical word problems influences performance, leaving format ambiguities unaddressed. In contrast, research on numerical benchmark localisation demonstrates that translating and localising names and cultural entities

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.

within word problems degrades model accuracy (Azime et al., 2026), yet these studies overlook regional numerical formatting entirely during the localisation process.

To address this gap, we introduce, GLOBENUM, a comprehensive benchmark designed to systematically evaluate format-induced numerical ambiguity in LLMs across diverse cultural contexts. Our framework evaluates performance across two parallel modalities: **Direct Symbolic Operations**, which tests core mathematical operations (addition, subtraction, multiplication, and division) using explicit formatting rules to measure basic notation parsing without natural language distractions; and **Localised Word Problems**, which tests realistic scenario-based problems localised to specific country-language pairs and validated by human annotators to measure whether models leverage contextual signal to infer the correct numerical standard. Central to our framework is a taxonomy distinguishing unambiguous, format-ambiguous, and value-ambiguous numerical expressions, which isolates the subset of representations whose correct value cannot be recovered without external context, the cases where contextual disambiguation is not merely helpful but necessary. Using this framework, we determine the extent to which explicit linguistic, geographic, and cultural context enables diverse models to dynamically adjust their numerical parsing rules.

Our evaluation spans 10 distinct numerical formatting standards, 12 target countries, 7 languages, and 7 LLM families. Our primary contributions are threefold: **(1) A Novel Benchmark Addressing Format Ambiguity:** We present the first benchmark[1] targeting locale-induced numerical ambiguity, featuring both symbolic operations and human-verified localised word problems across 7 leading model families; **(2) A Taxonomy Isolating Genuine Ambiguity:** We introduce a difficulty taxonomy separating numbers whose values are merely unfamiliar from those formally unrecoverable without context, enabling precise measurement of format-induced failures; and **(3) Analysis of Contextual Disambiguation:** We investigate the impact of linguistic, geographic, and cultural context on numerical parsing, identifying key failure modes where models override explicit contextual cues in favour of locale priors.

2 Related Work

2.1 Cultural adaptation and democratisation

Most efforts to widen access to language technology focus on language coverage. Translation models now support hundreds of languages (Team et al., 2022), and fine-tuned instruction models are available for many of them (Üstün et al., 2024). Multilingual testing has followed the same path, translating English reasoning benchmarks like GSM8K (Cobbe et al., 2021) and SVAMP (Patel et al., 2021) into other languages to assess performance outside English (Shi et al., 2023; Chen et al., 2024). However, translation alone does not make a benchmark culturally relevant. Direct translations retain the cultural context of their source, for instance, 28% of MMLU questions need culturally sensitive knowledge to answer (Singh et al., 2025). Therefore, recent work builds cultural benchmarks for everyday knowledge (Myung et al., 2024; Chiu et al., 2025), and for regional social rules (Rao et al., 2025). Yet models frequently default to globally dominant conventions rather than those appropriate to the situation, even on tasks contain local cultural clues (Permadi et al., 2026).

Instead of writing new questions from scratch, Azime et al. (2026) adapt GSM8K to local contexts by replacing source names, organisations, and currencies with local alternatives. They find that evaluating on native entities can reveal performance drops masked by English-centric benchmarks, and that fine-tuning models on entity-localised training data improves robustness to such cultural variation. Karim et al. (2026) apply the same idea at scale, replacing names, foods and places in 1,198 GSM8K questions for six countries while keeping every operation and number unchanged, and report accuracy drops of up to 5.9% across 14 models. Other approaches balance structure and regional variation by using fixed templates to generate local content (Elsetohy et al., 2026), or by relying on native speakers to write reasoning questions directly (Fabbri et al., 2025). Across this line of work, however, current adaptation usually stops at words and names. Even when a question adopts local entities, its numbers remain formatted according to the source country's conventions.

[1] Available at `https://anonymous.4open.science/r/anonymous-C6F8`)

2.2 NUMERIC CONVENTION

The focus of prior work on words and names overlooks numeric convention, which also varies by region. Writing the same quantity under a different convention changes the surface characters a model reads, making performance sensitive to visual presentation. How numbers are represented is a fundamental design choice (Thawani et al., 2021), and tokenisation alone alters arithmetic accuracy in frontier models (Singh & Strouse, 2024). More broadly, numeracy benchmarks reveal persistent failures across number types and suggest that models often process numbers as surface patterns rather than magnitudes (Yang et al., 2025b; Li et al., 2025).

Early work addresses such representational failures by modifying model design or prompting, for example by encoding number as aggregate digit embedding (Sivakumar & Moosavi, 2025), or noting digit counts to improve place-value reasoning (Schwartz et al., 2024) or adjusting numeral handling in chain-of-thought steps (Xu et al., 2024). Reddy et al. (2026) show that accuracy drops sharply when the same arithmetic expression uses a less common numeral script or an unfamiliar decimal and grouping format. Similarly, large-scale testing finds substantial variation in model accuracy when reading Eastern versus Western Arabic digits (Alhumud et al., 2026). These findings show that numerical reasoning is sensitive to changes in representation even when the arithmetic is unchanged. However, existing evaluations largely study this sensitivity in isolated numerical or symbolic settings, leaving unclear whether surrounding context can help models infer the intended numerical convention. Closest to this setting, Bui et al. (2025) find that models default to the measurement system dominant in their training data. Their task requires converting between measurement units. In contrast, we study cases where the same numerical string can correspond to different values depending on regional convention, and evaluate whether linguistic, geographic, and cultural context enables models to resolve this ambiguity.

3 THE GLOBENUM BENCHMARK

To systematically evaluate how models interpret and disambiguate numerical notation across linguistic and cultural contexts, we introduce the GLOBENUM (**GLOB**al **E**valuation of **NUM**erical Format) benchmark. GLOBENUM evaluates numerical reasoning across regional formatting conventions and, crucially, whether models can leverage linguistic, geographic, and cultural context to resolve representations whose intended value depends on the active convention. We first describe the typographical rules of global numerical formatting. Subsequently, we leverage those distinctions to develop a taxonomy of ambiguity between numerical format to generate a set of *Symbolic Operations*. A human-verified parallel *Localised Word Problems* set is then generated to evaluate contextual signal's impact on resolving numerical format ambiguity.

3.1 GLOBAL NUMERICAL FORMAT

Although numerical digits (0–9) are widely treated as a universal language, the typographical rules governing how real numbers are formatted vary substantially across regions, jurisdictions, and technical domains. Across global conventions, numerical formatting is primarily governed by three typographical parameters:

1. **The Decimal Marker:** The symbol used to separate the integer portion of a number from its fractional component, predominantly either a dot (`.`) or a comma (`,`).
2. **The Digit Group Separator:** The character inserted to group digits to improve readability, including commas (`,`), dots (`.`), non-breaking spaces, or apostrophes (`'`).
3. **The Grouping Interval:** The frequency at which separators are inserted. While Western standards adhere to a uniform 3-digit grouping (thousands, millions, billions), other regional standards employ distinct cadence rules, such as the Indian-subcontinental 2-2-2-3 grouping scheme (lakhs and crores) or 4-digit grouping in China and Japan.

To systematically measure model resilience to format variations, we codify these conventions into 10 representative formatting standards, summarised in Table 1 where the format code letters stand for D (dot), C (comma), S (space), A (apostrophe). These standards comprise dot-decimal conventions (`D-only`, `SD`, `CD`) and comma-decimal conventions (`C-only`, `SC`, `DC`, `AC`, `ADC`).

Table 1: Taxonomy of the 10 numerical formatting standards evaluated in our benchmark, detailing the decimal marker, representative example for the integer 123 456 789 with two fractional decimal places, and representative target locales. Format codes `D-decimal` and `C-decimal` represent the canonical, separator-free baseline standards for dot-decimal and comma-decimal conventions, respectively; these can be used with any locale that adopts dot or comma as their decimal marker. Format code letters indicate: D (Dot), C (Comma), S (Space), and A (Apostrophe).

Format Code	Format	Representative Example	Example Locales
`D-only`	**D**ot	`123456789.00`	Canonical Dot Baseline
`SD`	**S**pace-**D**ot	`123 456 789.00`	`US`, `UK`
`CD(333)`	**C**omma-**D**ot	`123,456,789.00`	`US`, `UK`, `South Korea`
`CD(2223)`	**C**omma-**D**ot	`12,34,56,789.00`	`India`
`CD(444)`	**C**omma-**D**ot	`1,2345,6789.00`	`China`
`C-only`	**C**omma	`123456789,00`	Canonical Comma Baseline
`SC`	**S**pace-**C**omma	`123 456 789,00`	`France`, `South Africa`
`DC`	**D**ot-**C**omma	`123.456.789,00`	`Belgium`, `Spain`, `Germany`
`AC`	**A**postrophe-**C**omma	`123'456'789,00`	`Switzerland`
`ADC`	**A**postrophe-**D**ot-**C**omma	`123'456.789,00`	`Colombia`

A country may use or encounter multiple numerical formatting conventions across administrative, technical, and software contexts. For example, while national standards may mandate space-separated digits (`SC` or `SD`) for official publications, commercial software often defaults to period or comma separators (`CD` or `DC`). We therefore include the unseparated formats `D-only` and `C-only` as canonical baselines for dot- and comma-decimal conventions, respectively. To systematically measure model performance under these real-world variations, our benchmark controls for notation by applying a single, consistent formatting standard per problem instance, allowing us to evaluate whether a model can correctly follow a specific regional convention without intra-prompt ambiguity.

3.2 Symbolic Operations Framework

We distinguish ambiguity along two dimensions: whether the active formatting convention can be identified from the numerical representation, and whether the scalar value itself can be uniquely recovered. This yields three categories: **Unambiguous Numbers** contain structural attributes that uniquely identify their format. For example, `6789,2` identifies a comma-decimal convention (`C-only`), allowing unambiguous deduction of the underlying notation rule; **Format-Ambiguous Numbers** have a uniquely recoverable scalar value, but their specific formatting convention remains unidentifiable in isolation. For instance, the integer `789` represents the same value across all 10 formatting standards, despite providing no evidence about which standard is active; **Value-Ambiguous Numbers** have scalar values that cannot be determined without knowing the active formatting convention. For example, `1,234` represents 1234 under comma-separated formats (`CD(333)` and `CD(2223)`), but 1.234 under comma-decimal formats (`C-only`, `SC`, `DC`, `AC`, `ADC`).

Pairwise Interaction Tiers. Each arithmetic problem pairs two operands, and the ambiguity categories of the two operands define its interaction tier. Using combinations of these three categories, we construct pairwise arithmetic problems from a core set of 100 base templates across all four primary operators ($+$, $-$, $\times$, $\div$). , designed to isolate how models resolve notation under varying amounts of structural context. An Unambiguous operand fixes the format of the whole expression, which also resolves the value of the other operand. In contrast, when neither operand provides sufficient evidence to identify the format, the intended value may only be recoverable from external context. As detailed in Table 2, this pairwise design yields six distinct interaction tiers, five of which we evaluate. We refer to the four evaluated tiers whose result is resolvable from the expression as *Standard Ambiguity*, and to the Format-Ambiguous $\times$ Value-Ambiguous tier as *High Ambiguity*. The exact construction constraints are detailed in Appendix A.

Format-Ambiguous $\times$ Value-Ambiguous Interactions. This interaction tier represents the key context-dependent evaluation setting in our symbolic framework. A format-ambiguous operand (e.g., `789`) provides no structural clue to identify the active notation, while a value-ambiguous operand (e.g., `1,234`) yields multiple valid numeric values depending on the convention. Their interaction therefore makes the calculation intrinsically underdetermined: no evidence within the

Table 2: Taxonomy of pairwise operand interaction tiers in the GLOBENUM symbolic evaluation framework. Ticks (✓) and crosses (×) indicate whether the format standard (F) and scalar value (V) are intrinsically known for each operand, and whether the overall expression format (F_{expr}) and arithmetic result (V_{expr}) are resolvable without external prompt context.

		Operand 1		Operand 2		Resolved	
Interaction Tier	**Example**	F_1	V_1	F_2	V_2	F_{expr}	V_{expr}
Unambig. × Unambig.	`6789,2 + 1234,5`	✓	✓	✓	✓	✓	✓
Unambig. × Format-Ambig.	`6789,2 + 789`	✓	✓	×	✓	✓	✓
Unambig. × Value-Ambig.	`6789,2 + 1,234`	✓	✓	×	×	✓	✓
Format-Ambig. × Format-Ambig.	`789 + 456`	×	✓	×	✓	×	✓
Format-Ambig. × Value-Ambig.	`789 + 1,234`	×	✓	×	×	×	×
Value-Ambig. × Value-Ambig.[†]	`1,234 + 5,678`	×	×	×	×	×	×

[†] The *Value-Ambiguous* × *Value-Ambiguous* tier is intentionally excluded because every candidate reading yields the same digit sequence, so the answer cannot reveal which notation a model applied.

arithmetic expression can resolve the calculation. Performance in this tier consequently measures whether models can use contextual signals, such as explicit format instructions or geographic locale information, to resolve the ambiguity.

This structural ambiguity arises from the role inversion of dots and commas across formatting conventions, where the same symbol alternates between acting as a decimal marker or a digit group separator. We categorise these typographical conflicts into three primary ambiguity pairings (see Appendix B for further detail and examples):

1. **Four-digit grouping vs. Comma-decimal:** Conflicts between `CD(444)` and all comma-decimal format (`C-only`, `SC`, `DC`, `AC`, `ADC`) e.g `1,2345`.
2. **Two/Three-digit grouping vs. Comma-decimal vs. Comma-decimal:** Conflicts between (`CD(333)`, `CD(2223)`) and comma-decimal (`C-only`, `SC`, `DC`, `AC`, `ADC`) e.g. `1,234`.
3. **Dot-decimal vs. Dot-grouping:** Conflicts between standard dot-decimal format (`D-only`, `SD`, `CD(333)`, `CD(2223)`, `CD(444)`) and (`DC`, `ADC`) e.g. `1.234`.

3.3 LOCALISED WORD PROBLEMS

To evaluate whether models can leverage contextual cues to infer formatting convention and resolve the intended numerical values, we construct a parallel set of scenario-based word problems. These word problems require performing the same mathematical operations on identical numerical values within realistic narratives. We select base templates from the English FERMAT dataset (Sivakumar & Moosavi, 2023) and systematically localise them across 12 target countries (23 country–language pairs). All localised instances, cultural references, and numerical adaptations were reviewed, edited, and verified by expert human annotators who are native speakers of the respective target languages and familiar with the target locales.

To evaluate cultural reasoning and context-driven numerical disambiguation, we transform symbolic operations into localised word problems while controlling the mathematical task. Across locales, each instance preserves the source template's operation, structure, and scalar values, while linguistic and cultural context is systematically varied. This design isolates whether models can use contextual cues to infer the intended numerical convention without confounding changes in mathematical difficulty. We construct parallel instances across 23 country-language pairs covering 12 countries and seven languages (Table 3). Our construction framework consists of four controlled stages:

1. **Template Standardisation:** We select 100 core templates from FERMAT (Sivakumar & Moosavi, 2023), balanced equally across the four arithmetic operations (25 per operator). Each template is decomposed into named entities, objects, and measurement units using a fixed 14-tag schema, with annotations initially generated by `Gemini-3.6-Flash` and manually verified (Appendix C.1).
2. **Controlled Cultural Substitution:** For each locale, tagged slots are instantiated from a curated word bank of culturally familiar entities. Substitutions are constrained to preserve semantic type, measurement dimension, and plausible numerical scale, without altering scalar values (Appendix C.2).

3. **Locale-Specific Rendering:** Culturally adapted templates are translated into the target language using `Gemini-3.6-Flash`. Numerical placeholders (`[NUM1]`, `[NUM2]`) remain unpopulated throughout translation, separating linguistic localisation from numerical formatting and preventing translation from altering the target notation.
4. **Human Validation:** Native speakers from each target locale evaluate every template for naturalness, rewriting unnatural phrasing according to standardised guidelines. Additionally, final templates are back-translated into English using DeepL to verify semantic fidelity to the source problem (Appendix C.3).

Table 3: Coverage of GLOBENUM across 23 country–language pairs, 12 countries, and seven languages. Each locale is paired only with number-format conventions attested for that setting.

Locale Code	Country / Region	Language	Evaluated Format Codes
`UK-en`	United Kingdom	English	`D-only`, `SD`, `CD(333)`
`US-es/en`	United States	Spanish / English	`D-only`, `SD`, `CD(333)`
`IN-ta/en`	India	Tamil / English	`D-only`, `CD(2223)`
`CH-fr/de/en`	Switzerland	French / German / English	`C-only`, `AC`
`ZA-en`	South Africa	English	`C-only`, `SC`
`ES-es/en`	Spain	Spanish / English	`C-only`, `DC`
`CO-es/en`	Colombia	Spanish / English	`C-only`, `ADC`, `DC`
`BE-fr/en`	Belgium	French / English	`C-only`, `DC`
`CN-zh/en`	China	Mandarin / English	`D-only`, `CD(444)`
`FR-fr/en`	France	French / English	`C-only`, `SC`
`DE-de/en`	Germany	German / English	`C-only`, `DC`
`KR-ko/en`	South Korea	Korean / English	`D-only`, `CD(333)`

4 Experimental Setup

Models. We test a suite of 11 open- and closed-weight models. Specifically, `gpt-5.6-luna` (OpenAI, 2026) and `deepseek-v3-flash` (DeepSeek-AI et al., 2024) serve as representative proprietary models due to their widespread adoption. For open-weight baselines, we include two Mistral family model: `ministral3-8b` (Liu et al., 2026) and `mixtral-8x22b` (Jiang et al., 2024), three Qwen family models: `qwen2.5-32b/72b` (Qwen et al., 2025), and `qwen3.8-27b` (Yang et al., 2025a) for their multilingual capabilities. Finally, we evaluate models designed with an explicit focus on global or regional outreach, including `apertus-8b/70b` (Hernández-Cano et al., 2026), `aya-expanse-32b` (Dang et al., 2024), and `eurollm-22b` (Martins et al., 2025).

Prompting Strategies. We evaluate each model under five prompting conditions (P1–P5) that progressively control the contextual information available for interpreting numerical notation. **P1 (Equation-only Baseline)** provides a context and language free baseline, presenting only the expression (e.g., `789 + 1.234 =`). **P2 (Standard Instruction)** adds a standard task instruction (*"Solve the following problem"*) but no information about the notation. **P3 (Location)** introduces geographic context by specifying the target location (e.g., *"Location: Colombia"*), testing whether models associate locale with its numerical conventions. **P4 (Format hint)** instead provides an explicit formatting cue (*"Pay attention to the numerical formatting"*) without identifying the locale, testing whether directing attention to notation alone changes the interpretation. Finally, **P5 (Format + Location)** combines both geographic and explicit signal for disambiguation. In addition to English language prompts, we conduct multilingual experiments using Spanish, French, and German prompts to isolate the effect of language and compare cultural variations between symbolic operations and word problems. Exact prompt templates and system prompts are provided in Appendix D.

Dataset and Evaluation Scale. The resulting evaluation comprises 7,000 symbolic expressions and 35,300 localised word problems spanning 10 numerical formatting standards and 23 country–language pairs across 12 countries and 7 languages. Combined with the prompting conditions and locale-specific expansions described above, these yield 71,800 symbolic and 141,200 localised word problem evaluations, for a total of 213,000 main evaluation instances per model. We additionally evaluate 45,400 instances in a multilingual symbolic ablation. Detailed instance counts and execution environments are provided in Appendices E, and F.

Evaluation Metrics. To distinguish arithmetic correctness from adherence to the target numerical notation, we evaluate model outputs using four complementary metrics. First, **Digit Sequence Correct (DS)** measures pure numerical accuracy independent of formatting by stripping non-significant zeroes, whitespaces, and structural symbols, comparing only the remaining digit sequence against the ground-truth value. Second, **Format Correct (FC)** measures adherence to the target numerical convention by verifying whether the decimal marker, group separators, and digit-grouping intervals strictly match the target format code. Third, **Target Value Correctness (VC)** measures exact overall correctness, requiring the model prediction to be both arithmetically correct (DS) and properly formatted according to the specific target locale standard (FC). Finally, **Any-Format Correctness (V-Any)** serves as a relaxed value metric evaluating whether the prediction is arithmetically correct under *any* valid formatting standard within the benchmark.

5 Analysis

5.1 Model Number Format Biases

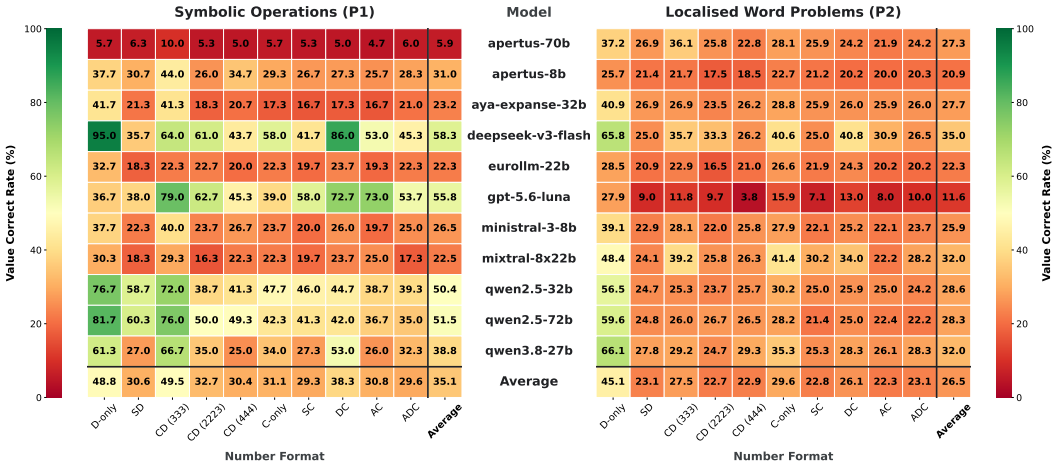

Figure 1: **Cross-Model Performance Across Regional Number Formats.** Comparison of `value_correct_rate` (%) aggregated across combined unambiguity tiers for Direct Symbolic Operations (left) and Localised Word Problems (right).

To evaluate model sensitivity to regional numerical representations, performance across numerical formats was averaged across three difficulty tiers: *Unambiguous* × *Unambiguous*, *Unambiguous* × *Format-Ambiguous*, and *Format-Ambiguous* × *Format-Ambiguous*. Because the underlying numerical values and arithmetic operations remain strictly identical across formats within these tiers, any variance in accuracy directly reflects model format bias.

Figure 1 shows substantial format-dependent variation. In symbolic operations, familiar dot-decimal conventions generally perform best, but the pattern is not explained by the decimal marker alone: formats using the same decimal marker can differ by nearly 20 percentage points. Models are therefore sensitive to the structure of the notation, including the separator and grouping pattern.

Localisation does not remove these biases, but changes how they manifest. The standard dot-decimal format remains robust in word problems, while other high-performing symbolic formats degrade substantially. The effect also varies markedly across models: `DeepSeek-V3-Flash` remains comparatively strong across both modalities, whereas `GPT-5.6-Luna` performs strongly on symbolic expressions but degrades sharply on localised problems. Regionally or multilingually oriented models such as `Apertus`, `EuroLLM`, and `Aya-Expanse` show no consistent advantage over general-purpose models. Overall, format sensitivity reflects an interaction between notation and task modality rather than arithmetic performance alone.

Table 4: **Format bias under increasing contextual grounding.** Difference in Value Correctness (percentage points) between competing format groups for each High-Ambiguity split. Smaller absolute differences indicate reduced format bias; positive (negative) values indicate higher performance for the first (second) format group.

Format Comparison	**Example Ambiguity**	**Baseline** Symbolic P1	**Culture** Word P2	**+ Geog.** Word P3	**+ Lang.** Word P3
Four-digit grouping vs. Comma-dec.	$1,2345$	5.87	3.09	2.43	–0.44
Two/Three-digit grouping vs. Comma-dec.	$1,234$	43.82	9.57	9.24	5.29
Dot-dec. vs. Dot-grouping	1.234	53.53	45.13	43.08	37.49

5.2 Ambiguous Tier

In the most challenging tier, *Format-Ambiguous* $\times$ *Value-Ambiguous*, the intended numerical value cannot be recovered from the expression alone and must instead be resolved from external context. To quantify format bias under ambiguity, Table 4 presents the difference in Value Correctness between competing format groups as increasingly rich contextual grounding is introduced. At the **Baseline**, models exhibit a pronounced preference toward dot-decimal interpretations. This is most evident for comma-grouped dot-decimal vs. comma-decimal and dot-decimal vs. dot-grouped comma-decimal, where the performance gaps reach 43.82 and 53.53 percentage points, respectively. Thus, when the expression itself cannot determine the intended value, models disproportionately interpret ambiguous notation as using a dot decimal marker.

As more contextual cues are provided, format bias decreases across all three comparisons, but the effect is highly format-dependent. For four-digit grouping vs. comma-decimal, the initial preference is effectively eliminated under full context, while the much larger bias for comma-grouped dot-decimal vs. comma-decimal is also substantially reduced. In contrast, dot-decimal vs. dot-grouped comma-decimal remains strongly asymmetric, with a 37.49 percentage-point gap even under full contextual information. Thus, contextual cues can help models resolve ambiguous notation, but do not overcome all format preferences equally. In addition, reduced bias does not imply successful reasoning: Value Correctness remains below 20% for most conditions in this tier (Appendix H).

5.3 Impact of Contextual Signal

Table 5: **Prompt Context for Disambiguation** Averaged across 11 models for Digit Sequence Correct (`DS`), Format Correct (`FC`), Value Correct (`VC`), and Value Correct Under Any Format (`V-Any`). High Ambiguity (Format-Ambig. $\times$ Value-Ambig.), Standard Ambiguity (all other difficulty tier).

		Symbolic Operations				**Localised Word Problems**			
Difficulty Tier	**Prompt Strategy**	**DS (%)**	**FC (%)**	**VC (%)**	**V-Any (%)**	**DS (%)**	**FC (%)**	**VC (%)**	**V-Any (%)**
Standard Ambiguity	P1 (Equation-only)	38.28	60.35	26.09	36.58	–	–	–	–
	P2 (Instruction)	36.99	42.18	24.06	35.92	36.11	40.81	20.44	31.32
	P3 (Location hint)	40.44	50.01	28.43	39.94	36.03	39.32	19.77	30.57
	P4 (Format hint)	40.79	46.44	27.74	39.91	37.61	45.48	23.01	33.57
	P5 (Format + Location)	45.32	63.23	37.02	44.98	37.92	51.37	25.06	33.22
High Ambiguity	P1 (Equation-only)	41.53	72.28	27.79	34.63	–	–	–	–
	P2 (Instruction)	40.95	41.78	22.14	33.02	33.95	43.38	15.08	24.17
	P3 (Location hint)	39.89	51.26	23.71	32.89	34.92	41.90	14.77	24.38
	P4 (Format hint)	42.88	48.48	25.28	35.09	36.12	46.67	17.00	26.72
	P5 (Format + Location)	45.39	63.05	32.78	39.75	37.69	49.55	19.86	28.07

Geography and Format Cues As shown in Table 5, geographic context alone provides limited support for numerical disambiguation. Adding a location cue (**P3**) improves Value Correctness in symbolic tasks under Standard Ambiguity, but provides no corresponding improvement in localised worded problems. A formatting cue (**P4**) produces more consistent gains across modalities, suggesting that explicitly directing attention to numerical notation is more effective than relying on models to infer the relevant convention from location alone.

Combining both cues (**P5**) yields the strongest overall performance, particularly under High Ambiguity, where external context is necessary to determine the intended value. Yet performance remains low even when both signals are available: Value Correctness reaches only 32.78% for symbolic oper-

ations and 19.86% for localised worded problems. Thus, providing relevant contextual information does not ensure that models will use it successfully to resolve numerical ambiguity.

Language and Culture

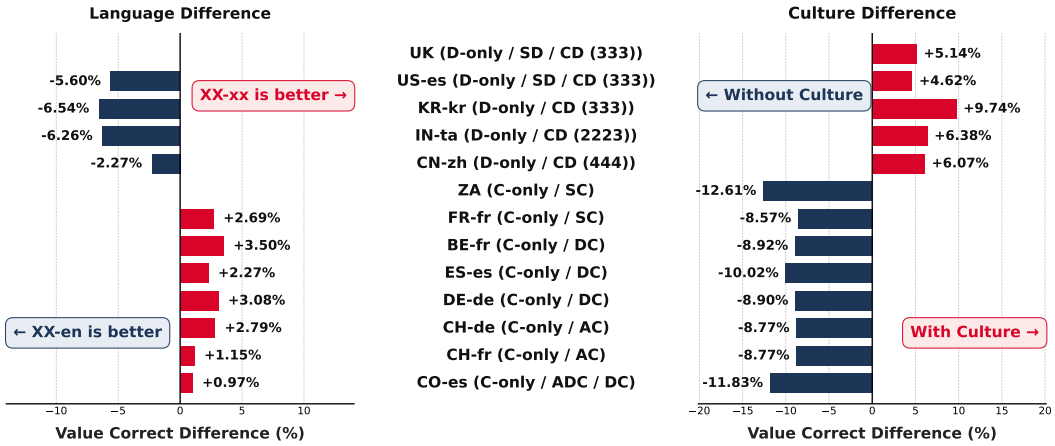

Figure 2: Value Correctness difference for **(Left)** Language Difference, calculated as Word problems evaluated in target languages (`XX-xx`) minus their English counterparts (`XX-en`), **(Right)** Culture Difference, calculated as localised Word problem (*With Culture*) minus Symbolic Operation (*Without Culture*) with prompt in official language (`XX-xx`).

Figure 2 shows that the effect of linguistic and cultural context depends strongly on the numerical convention. Translating localised word problems from English (`XX-en`) into the locale language (`XX-xx`) generally benefits comma-decimal locales, while producing smaller or negative changes for the dot-decimal locales. Language therefore does not provide a uniform improvement; its benefit is greatest where the numerical convention differs from the dominant dot-decimal convention.

Cultural framing exhibits the opposite asymmetry. Relative to symbolic operations in the same language, localised word problems improve Value Correctness for the dot-decimal locales shown, but reduce it across the comma-decimal locales. This indicates that adding semantically meaningful context can either support or interfere with numerical interpretation depending on the notation embedded within it. Notably, target-language translation partially offsets this degradation for several comma-decimal locales, suggesting that linguistic and numerical conventions interact rather than contributing independently to model performance.

Overall, these findings reveal a failure mode that is not captured by prior studies of numerical representation or benchmark localisation. In GLOBENUM's ambiguous settings, regional context is not merely an additional source of information: it is necessary to determine the numerical value itself. Yet models frequently fail to exploit this information, and linguistic or cultural localisation can either mitigate or amplify the resulting errors depending on the numerical convention. This exposes a distinct challenge for multilingual numerical reasoning: robustness requires not only performing consistently across languages and formats, but correctly grounding ambiguous numerical representations in the regional context that gives them meaning.

6 Conclusion

We introduced GLOBENUM to test a step that numerical reasoning benchmarks largely take for granted: reading the number itself. Our results show that LLMs exhibit strong defaults in how they interpret numerical notation. When a numerical string is genuinely ambiguous, models frequently favour dot-decimal interpretations and often fail to recover the intended value, even when given the user's location and directed to attend to formatting. These errors are particularly concerning because they are silent: an answer can be mathematically consistent with the wrong interpretation while appearing entirely valid. Numerical reasoning cannot be considered truly multilingual until models can recognise when numerical meaning depends on regional convention and use context to resolve it.

AI USE STATEMENT

In this work, we used generative AI tools for dataset generation code development, initial scenario localisation and translation via API calls, subsequent backtranslation into English via API calls, script creation for figures and tables, and dataset auditing to identify data inconsistencies. We have not used generative AI tools for final data verification or manual human annotations, and fully autonomous dataset creation without human validation is not applicable to this work. Additionally, we used generative AI tools for proofreading and editing sections of the manuscript. We have reviewed all AI-assisted work and take full responsibility for the final content.

REFERENCES

Anas Alhumud, Abdulaziz Alhammadi, and Muhammad Badruddin Khan. Arabicnumbench: Evaluating arabic number reading in large language models, 2026. URL https://arxiv.org/abs/2602.18776.

Israel Abebe Azime, Tadesse Destaw Belay, Dietrich Klakow, Philipp Slusallek, and Anshuman Chhabra. Bridging the culture gap: A framework for LLM-driven socio-cultural localization of math word problems in low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 855–870, San Diego, California, United States, 2026. Association for Computational Linguistics. URL https://aclanthology.org/2026.findings-acl.42/.

Minh Duc Bui, Kyung Eun Park, Goran Glavaš, Fabian David Schmidt, and Katharina von der Wense. On generalization across measurement systems: LLMs entail more test-time compute for underrepresented cultures. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21262–21276, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1032. URL https://aclanthology.org/2025.acl-long.1032/.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7001–7016, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.411. URL https://aclanthology.org/2024.findings-emnlp.411/.

Yu Ying Chiu, Liwei Jiang, Bill Yuchen Lin, Chan Young Park, Shuyue Stella Li, Sahithya Ravi, Mehar Bhatia, Maria Antoniak, Yulia Tsvetkov, Vered Shwartz, and Yejin Choi. CulturalBench: A robust, diverse and challenging benchmark for measuring LMs' cultural knowledge through human-AI red-teaming. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25663–25701, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1247. URL https://aclanthology.org/2025.acl-long.1247/.

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. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan

Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker. Aya expanse: Combining research breakthroughs for a new multilingual frontier, 2024. URL https://arxiv.org/abs/2412.04261.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. URL https://arxiv.org/abs/2412.19437.

Alaa Elsetohy, Sama Hadhoud, Haryo Akbarianto Wibowo, Chenxi Whitehouse, Genta Indra Winata, Fajri Koto, and Alham Fikri Aji. Macaron: Controlled, human-written benchmark for multilingual and multicultural reasoning via template-filling. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 47885–47906, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2211. URL https://aclanthology.org/2026.acl-long.2211/.

Alexander R. Fabbri, Diego Mares, Jorge Flores, Meher Mankikar, Ernesto Hernandez, Dean Lee, Bing Liu, and Chen Xing. Multinrc: A challenging and native multilingual reasoning evaluation benchmark for llms, 2025. URL https://arxiv.org/abs/2507.17476.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Alejandro Hernández-Cano, Alexander Hägele, Allen Hao Huang, Angelika Romanou, Antoni-Joan Solergibert, Barna Pásztor, Bettina Messmer, Dhia Garbaya, Eduard Frank Ďurech, Ido Hakimi, Juan Garcia Giraldo, Mete Ismayilzada, Negar Foroutan, Skander Moalla, Tiancheng Chen, Vinko Sabolčec, Yixuan Xu, Michael Aerni, Badr AlKhamissi, Inés Altemir Marinas, Mohammad Hossein Amani, Matin Ansaripour, Ilia Badanin, Harold Benoit, Emanuela Boros, Nicholas John Browning, Fabian Bösch, Maximilian Böther, Niklas Canova, Camille Challier, Clément Charmillot, Jonathan Coles, Jan Milan Deriu, Arnout Devos, Lukas Drescher, Daniil Dzenhaliou, Maud Ehrmann, Dongyang Fan, Simin Fan, Silin Gao, Miguel Gila, María Grandury,

Diba Hashemi, Alexander Miserlis Hoyle, Jiaming Jiang, Mark Klein, Andrei Kucharavy, Anastasiia Kucherenko, Frederike Lübeck, Roman Machacek, Theofilos Ioannis Manitaras, Andreas Marfurt, Kyle Matoba, Simon Matrenok, Henrique Mendonça, Fawzi Roberto Mohamed, Syrielle Montariol, Luca Mouchel, Sven Najem-Meyer, Jingwei Ni, Gennaro Oliva, Matteo Pagliardini, Elia Palme, Andrei Panferov, Léo Paoletti, Marco Passerini, Ivan Pavlov, Auguste Poiroux, Kaustubh Ponkshe, Nathan Ranchin, Javier Rando, Mathieu Sauser, Jakhongir Saydaliev, Mukhammadali Sayfiddinov, Marian Schneider, Stefano Schuppli, Marco Scialanga, Andrei Semenov, Kumar Shridhar, Raghav Singhal, Anna Sotnikova, Alexander Sternfeld, Ayush Kumar Tarun, Paul Teiletche, Jannis Vamvas, Xiaozhe Yao, Hao Zhao, Alexander Ilic, Ana Klimovic, Andreas Krause, Caglar Gulcehre, David Rosenthal, Elliott Ash, Florian Tramèr, Joost VandeVondele, Livio Veraldi, Martin Rajman, Thomas C. Schulthess, Torsten Hoefler, Antoine Bosselut, Martin Jaggi, and Imanol Schlag. Apertus: Democratizing open and compliant LLMs for global language environments. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 46877–46955, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2172. URL https://aclanthology.org/2026.acl-long.2172/.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. URL https://arxiv.org/abs/2401.04088.

Aabid Karim, Abdul Karim, Bhoomika Lohana, Matt Keon, Jaswinder Singh, and Abdul Sattar. Lost in cultural translation: Do llms struggle with math across cultural contexts?, 2026. URL https://arxiv.org/abs/2503.18018.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Haoyang Li, Xuejia Chen, Zhanchao Xu, Darian Li, Nicole Hu, Fei Teng, Yiming Li, Luyu Qiu, Chen Jason Zhang, Li Qing, and Lei Chen. Exposing numeracy gaps: A benchmark to evaluate fundamental numerical abilities in large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20004–20026, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1026. URL https://aclanthology.org/2025.findings-acl.1026/.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra,

Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026. URL https://arxiv.org/abs/2601.08584.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. EuroLLM: Multilingual language models for Europe, 2025.

Microsoft AI Economy Institute. Global AI diffusion in Q1 2026: Trends and insights. Research report, Microsoft, 2026. URL https://www.microsoft.com/en-us/corporate-responsibility/topics/ai-economy-institute/reports/global-ai-adoption-2026-q1/.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, Víctor Gutiérrez-Basulto, Yazmín Ibáñez García, Hwaran Lee, Shamsuddeen Hassan Muhammad, Kiwoong Park, Anar Sabuhi Rzayev, Nina White, Seid Muhie Yimam, Mohammad Taher Pilehvar, Nedjma Ousidhoum, Jose Camacho-Collados, and Alice Oh. Blend: A benchmark for llms on everyday knowledge in diverse cultures and languages. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 78104–78146. Curran Associates, Inc., 2024. doi: 10.52202/079017-2483. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8eb88844dafefa92a26aaec9f3acad93-Paper-Datasets_and_Benchmarks_Track.pdf.

OpenAI. Gpt-5.6 system card. https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf, July 2026.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Vynska Amalia Permadi, Xingwei Tan, Nafise Sadat Moosavi, and Nikolaos Aletras. No shortcuts to culture: Indonesian multi-hop question answering for complex cultural understanding. *Transactions of the Association for Computational Linguistics*, 14:1243–1265, 07 2026. ISSN 2307-387X. doi: 10.1162/TACL.a.726. URL https://doi.org/10.1162/TACL.a.726.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Abhinav Sukumar Rao, Akhila Yerukola, Vishwa Shah, Katharina Reinecke, and Maarten Sap. NormAd: A framework for measuring the cultural adaptability of large language models. 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. 2373–2403, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.120. URL https://aclanthology.org/2025.naacl-long.120/.

Varshini Reddy, Craig W Schmidt, Seth Ebner, Adam Wiemerslage, Yuval Pinter, and Chris Tanner. 1,729 vs. 1729: The effect of scripts and formats on LLM numeracy. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 10679–10696, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.518. URL https://aclanthology.org/2026.findings-acl.518/.

Eli Schwartz, Leshem Choshen, Joseph Shtok, Sivan Doveh, Leonid Karlinsky, and Assaf Arbelle. NumeroLogic: Number encoding for enhanced LLMs' numerical reasoning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 206–212, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.12. URL https://aclanthology.org/2024.emnlp-main.12/.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=fR3wGCk-IXp.

Aaditya K. Singh and DJ Strouse. Tokenization counts: the impact of tokenization on arithmetic in frontier llms, 2024. URL https://arxiv.org/abs/2402.14903.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18761–18799, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.919. URL https://aclanthology.org/2025.acl-long.919/.

Jasivan Sivakumar and Nafise Sadat Moosavi. FERMAT: An alternative to accuracy for numerical reasoning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15026–15043, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.838. URL https://aclanthology.org/2023.acl-long.838/.

Jasivan Alex Sivakumar and Nafise Sadat Moosavi. How to leverage digit embeddings to represent numbers? In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7685–7697, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.514/.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation, 2022. URL https://arxiv.org/abs/2207.04672.

Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro Szekely. Representing numbers in NLP: a survey and a vision. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 644–656, Online,

June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.53. URL https://aclanthology.org/2021.naacl-main.53/.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. Aya model: An instruction finetuned open-access multilingual language model. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15894–15939, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.845. URL https://aclanthology.org/2024.acl-long.845/.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6.

Ancheng Xu, Minghuan Tan, Lei Wang, Min Yang, and Ruifeng Xu. NUMCoT: Numerals and units of measurement in chain-of-thought reasoning using large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14268–14290, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.848. URL https://aclanthology.org/2024.findings-acl.848/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. URL https://arxiv.org/abs/2505.09388.

Haotong Yang, Yi Hu, Shijia Kang, Zhouchen Lin, and Muhan Zhang. Number cookbook: Number understanding of language models and how to improve it. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=BWS5gVjgeY.

APPENDIX

A ALGORITHMIC CONSTRUCTION OF DIFFICULTY TIERS

To rigorously evaluate how numerical formatting influences model parsing, we employ a rejection-sampling procedure to generate numbers for each format code $f \in \mathcal{F}$ and difficulty tier $t \in \mathcal{T}$. The generation logic samples exact `Decimal` values within bounded integer digit counts N_{int} and fractional decimal precisions N_{dec}, validating that the output string satisfies the exact structural constraints of the target tier.

A.1 DIFFICULTY TIER GENERATION RULES

1. **Unambiguous Tier:** Numbers are generated with sufficient integer magnitude N_{int} or fractional length N_{dec} to force the serialised string to display unique typographical markers (e.g. multi-group thousand separators or specific cadence patterns). For example, under `CD(2223)`, integer lengths of $N_{\text{int}} \in [6, 9]$ guarantee the appearance of the South Asian 2-2-2-3 grouping cadence (`12,34,56,789`).
2. **Value-Ambiguous Tier:** Numbers are constrained to specific fractional precisions (typically $N_{\text{dec}} = 3$ or $N_{\text{dec}} = 4$) and restricted integer lengths. This forces identical string representations across conflicting conventions; for instance, producing `1.234`, which represents 1.234 under dot-decimal standards but $1\,234$ under comma-decimal standards.
3. **Format-Ambiguous Tier:** Numbers are generated with short integer lengths ($N_{\text{int}} \leq 3$ or $N_{\text{int}} \leq 5$ depending on the format code) without triggering thousands separators. This ensures the scalar value is universally readable across all models (e.g., `789`), while leaving the underlying format code latent.

Table 6: Sampling parameters for generating numbers across formatting standards and difficulty tiers, specifying integer digit bounds (N_{int}), decimal presence (H_{dec}), and fractional decimal precision (N_{dec}).

Target Tier	Format Code	Integer Digits (N_{int})	Has Decimal (H_{dec})	Decimal Places (N_{dec})
Unambiguous	`CD(444)`	$[5, 8]$ or $[9, 11]$	Random / False	$[1, 2]$
	`D-only`, `C-only`	$[5, 8]$	True	$[1, 2]$
	`SD`, `SC`	$[4, 7]$	True	$[1, 2]$
	`CD(333)`	$[6, 8]$ or $[7, 9]$	Random / False	$[1, 2]$
	`CD(2223)`	$[6, 9]$	Random	$[1, 2]$
	`AD`	$[4, 6]$ or $[7, 9]$	Random / False	$[1, 2]$
	`DC`	$[7, 10]$	Random	$[1, 2]$
	`ADC`	$[7, 10]$	True	$[1, 2]$
Value-Ambiguous	`CD(444)`	$[1, 4]$	True	4
	`D-only`, `SD`, `C-only`, `SC`	$[1, 3]$	True	3
	`CD(333)`, `AD`, `DC`	$[1, 5]$	True	3
	`CD(2223)`	$[1, 5]$	True	3
	`ADC`	$[3, 6]$	True	3
Format-Ambiguous	`D-only`, `SD`, `C-only`, `SC`, `AD`	$[1, 3]$ or $[1, 4]$	False	0
	`CD(333)`, `CD(2223)`	$[1, 5]$	Random	$\{1, 2, 5\}$
	`CD(444)`	$[1, 4]$	Random	$\{1, 2, 3, 5\}$
	`DC`	$[1, 6]$	Random	$\{1, 2, 5\}$
	`ADC`	$[4, 6]$	True	$\{1, 2, 5\}$

B AMBIGUITY CONFLICT MODES AND WORKED PARSING EXAMPLES

This appendix provides detailed parsing truth tables and worked step-by-step arithmetic examples for the three primary typographical conflict modes occurring in *Format-Ambiguous* × *Value-Ambiguous* interactions. In these settings, external contextual signals (e.g., explicit system prompts or locale directives) are required to select the correct active format code, as the input arithmetic string contains no intrinsic structural features to disambiguate the operands.

B.1 Conflict Mode 1: Comma-Decimal vs. Myriad Dot-Decimal

This conflict mode occurs between comma-decimal conventions (`C-only`, `SC`, `DC`, `AC`, `ADC`) and Chinese myriad dot-decimal formatting (`CD 444`). Here, the comma character (`,`) undergoes character role inversion, shifting between a fractional decimal marker and a 4-digit (myriad) group separator.

Table 7: Parsing outcome for expression `789 + 1,2345` across competing format standards under Conflict Mode 1.

Active Format Code	Role of Comma (`,`)	Operand 1	Operand 2	Evaluated Result
`C-only`, `SC`, `DC`, `AC`, `ADC`	Decimal Marker	789	1.2345	790.2345
`CD(444)`	Myriad Separator	789	12 345	13 134.0000

Worked Example. Consider the expression `789 + 1,2345`:

- Under `DC` (e.g., Germany or Belgium), `789` is an integer and `1,2345` is parsed with a decimal comma as $1 + \frac{2345}{10000} = 1.2345$. The resulting sum is $789 + 1.2345 = 790.2345$.
- Under `CD(444)` (e.g., China), `1,2345` uses a comma to mark four integer digits (1 wan 2345). It is parsed as the integer $12\,345$. The resulting sum is $789 + 12\,345 = 13\,134$.

B.2 Conflict Mode 2: Comma-Decimal vs. Standard / South Asian Dot-Decimal

This conflict mode occurs between comma-decimal conventions (`C-only`, `SC`, `DC`, `AC`, `ADC`) and 3-digit or 2-2-2-3 dot-decimal formats (`CD 333`, `CD 2223`). The comma alternates between a fractional decimal marker and a standard thousands or lakhs/crores separator.

Table 8: Parsing outcome for expression `789 + 1,234` across competing format standards under Conflict Mode 2.

Active Format Code	Role of Comma (`,`)	Operand 1	Operand 2	Evaluated Result
`C-only`, `SC`, `DC`, `AC`, `ADC`	Decimal Marker	789	1.234	790.234
`CD(333)`, `CD(2223)`	Thousands Separator	789	1 234	2 023.000

Worked Example. Consider the expression `789 + 1,234`:

- Under `SC` (e.g., France or South Africa), the comma acts as a decimal marker. Operand 2 is parsed as 1.234, yielding $789 + 1.234 = 790.234$.
- Under `CD(333)` or `CD(2223)` (e.g., United Kingdom or India), the comma acts as a digit group separator. Operand 2 is parsed as the integer $1\,234$, yielding $789 + 1\,234 = 2\,023$.

B.3 Conflict Mode 3: Dot-Decimal vs. Separator-Inverted Comma-Decimal

This conflict mode occurs between standard dot-decimal conventions (`D-only`, `SD`, `CD 333`, `CD 2223`, `CD 444`) and Continental European standards that use periods as digit group separators (`DC`, `ADC`). Here, the period (`.`) experiences character role inversion.

Table 9: Parsing outcome for expression `789 + 1.234` across competing format standards under Conflict Mode 3.

Active Format Code	Role of Period (`.`)	Operand 1	Operand 2	Evaluated Result
`D-only`, `SD`, `CD(333)`, `CD(2223)`, `CD(444)`	Decimal Marker	789	1.234	790.234
`DC`, `ADC`	Thousands Separator	789	1 234	2 023.000

Worked Example. Consider the expression `789 + 1.234`:

- Under `D-only` or `CD(333)`, the period is a decimal point. Operand 2 is parsed as 1.234, producing $789 + 1.234 = 790.234$.
- Under `DC` or `ADC` (e.g., Spain or Colombia), the period is a thousands separator. Operand 2 is parsed as $1\,234$, producing $789 + 1\,234 = 2\,023$.

C Localised Word Problem Generation Details

C.1 Entity and Unit Tagging Taxonomy

The 14-tag entity taxonomy covers five primary semantic categories, listed in Table 10.

Category	Tags
People	`[NAME]`, `[NAME2]`, `[PRONOUN.subject]`, `[PRONOUN.object]`, `[PRONOUN.possessive]`
Locations	`[LOCATION.town]`, `[LOCATION.place]`
Time	`[WEEKDAY]`
Objects	`[OBJECT.measured]`, `[OBJECT.counted]`, `[OBJECT.container]`
Measurement	`[UNIT]`, `[UNIT2]`, `[CURRENCY]`

Table 10: The 14-tag entity taxonomy. A second distinct filler of the same category takes a numbered variant, so a template naming two measured objects uses `[OBJECT.measured]` and `[OBJECT.measured2]`.

Measured vs. Counted Distinctions. An entity adjacent to a measurement unit is categorised as MEASURED, even if the underlying object is discretely countable (e.g., *"NUM1 g of bottle caps"* is tagged as MEASURED, whereas *"NUM1 bottle caps"* is tagged as COUNTED). The tag reflects the grammatical presentation of quantity rather than intrinsic physical properties, ensuring that the word bank replaces measured objects only with items sold by weight, length, or volume within that locale.

Unit Localisation. Measurement units are tagged analogously to named entities, enabling each locale to employ its native unit systems. Only the unit descriptor is adapted (e.g., *"NUM1 inches"* in the source becomes *"NUM1 cm"* in metric locales, and *"NUM1 g"* shifts to *"NUM1 ounces"* in US English). Crucially, raw numerical values are preserved without unit conversion, ensuring that the underlying arithmetic problem remains identical across all locales while changing only the narrative framing.

Tagging prompt. The prompt gives the tag list and the rules that decide where each tag goes. The number placeholders `NUM1` and `NUM2` are already in the template, and the rules keep them, and the wording outside the tags, exactly as written. Only the entity words change.

```
Tag this maths question using only the tags below.

Tags:
[NAME] [NAME2] [PRONOUN] [PRONOUN.obj] [PRONOUN.poss]
[LOCATION.town] [LOCATION.place] [WEEKDAY]
[OBJECT.measured] [OBJECT.counted] [OBJECT.container]
[UNIT] [UNIT2] [CURRENCY]

Rules:
-- The same tag twice means the same word twice.  A second, different filler
takes a 2:  [NAME2], [OBJECT.measured2].  Add .pl for a plural.
-- Keep NUM1 and NUM2 exactly as they are.
-- A number beside a unit means the object is measured, even if you could
count it:  NUM1g of bottle caps is measured, NUM1 bottle caps is counted.
-- A title stays outside the tag:  Mrs.  Hilt is Mrs.  [NAME].
-- A possessive keeps its ending outside the tag:  Emily's is [NAME]'s.
-- Do not rewrite the sentence or change the maths.

Question:  {text}
```

```
Return JSON only:
{"tagged": the sentence with tags,
  "slots": {tag: {"was": the original word, "why": one line}},
  "kept": [words left untagged on purpose]}
```

C.2 Word Bank Construction Constraints

Word bank entries are manually curated under four strict constraints:

1. **Type Consistency:** Candidates must match the source entity class, ensuring measured objects are never substituted with counted items.
2. **Dimensional Alignment:** Candidates must share the same measurement dimension (e.g., replacing mass-measured items with mass-measured alternatives).
3. **Plausible Scale:** Numerical quantities must remain realistic for the substituted entity (e.g., *"NUM1 kg of chicken"* admits *"paneer"* in `IN-en`, but excludes *"yeast"*, which is typically packaged in small quantities).
4. **Cultural Everyday Relevance:** Substituted items must represent familiar, everyday objects or places within the target locale.

C.3 Human Review and Verification Protocol

Native speakers of each target locale review the localised templates assigned to their country–language pair, following the guideline reproduced below. Annotators are instructed not to solve the arithmetic: they read, verify, and correct the text, and return their corrections through a form. Finalised localised templates are back-translated into English using DeepL and manually audited against the original source templates to confirm semantic equivalence without introducing mathematical drift.

Annotation guideline. Each annotator reviews the batch of word problems assigned to their language and region. They are told not to solve or calculate the answers: the task is to read, verify and correct the text, and to return the corrections through a form. Everything in a question must sound entirely natural to a native speaker living in that country. The guideline asks four checks of every item.

1. **Vocabulary.** Are the names, objects, spellings and units the ones a native speaker would meet in a school maths book here? *Budi went to the store to buy 5 gallons of gas* is not a UK question, because the name is not one used there and *store*, *gallons* and *gas* are American. It becomes *Brenda went to the shop to buy 5 litres of petrol*.
2. **Numerical formatting.** Are large numbers and decimals punctuated and grouped the way this region writes them? The UK marks a decimal with a point, so *8,5 kg of flour* becomes *8.5 kg of flour*, while France and Germany mark it with a comma, so *4.5 Meter* becomes *4,5 Meter*. The UK also groups by three digits, so *1.00.000 residents* becomes *100,000 residents* rather than a lakh or a European point grouping.
3. **Number and object.** Does the quantity fit the thing it counts? A decimal has to attach to something measurable such as time, weight or distance, and not to a whole item. *Oliver has 4.5 cars to build* becomes *Oliver has 4.5 hours of car repairs to do*, so the object changes and the numbers stay as they are.
4. **Fluency.** Does the sentence read as though it was written here? Where it sounds robotic, awkwardly translated or poorly phrased, rewrite it, keeping what the question asks. A misspelled or foreign word counts here as well, so *John kauft 2 Pfund Tomaten im Supermarket* becomes *Lukas kauft 2 Kilo Tomaten im Supermarkt*.

Checks 2 and 3 operate on a rendered value, so the review copy carries numbers rather than slots. Review is therefore a human check on the numerical format itself, and not only on the language around it. Check 3 also lets a reviewer replace an entity rather than its wording, which is the one stage where an entity can come from the annotator instead of the word bank.

Annotators are compensated with vouchers for their regional marketplace, and each signs a consent form covering participation, compensation, withdrawal and data use before starting.

D Prompt Templates and System Prompt

For conditions P2–P5, all user prompts are paired with a standardised system directive: *"You are a highly competent mathematician."* To preserve localised separator strings and prevent decoding parsers from stripping regional typography, outputs are mandated to return valid string-encoded JSON in the exact structure `{"answer": "#####"}`. The P1 baseline omits system prompt and JSON constraints entirely, evaluating bare mathematical strings.

Table 11 specifies the exact text templates used across prompting conditions P1–P5 for English main experiments. Multilingual prompt variants (French, Spanish, German) used in the ablation studies maintain identical system prompt and JSON output structures.

Table 11: Prompt templates across evaluation conditions P1–P5. System prompt and JSON formatting instructions apply to P2–P5.

Condition	User Prompt Template
P1	`{problem} =`
P2	`Solve the following problem. Return the final answer as JSON: {"answer": "#####"}.\n\n{problem}`
P3	`Location: {location}\nSolve the following problem. Return the final answer as JSON: {"answer": "#####"}.\n\n{problem}`
P4	`Pay attention to the numerical formatting and solve the following problem. Return the final answer as JSON: {"answer": "#####"}.\n\n{problem}`
P5	`Location: {location}\nPay attention to the numerical formatting and solve the following problem. Return the final answer as JSON: {"answer": "#####"}.\n\n{problem}`

Table 12: System prompt and prompt template components across multilingual evaluation conditions.

Language	System Directive	Location Prefix (*L*)	Format Hint (*F*)	Task Instruction (*T*)
French	`Vous êtes un mathématicien hautement compétent.`	`Lieu : {location}.`	`Faites attention au formatage des nombres`	`résolvez le problème suivant : {problem}`
Spanish	`Eres un matemático altamente competente.`	`Ubicación: {location}.`	`Presta atención al formato de los números`	`resuelve el siguiente problema: {problem}`
German	`Sie sind ein äußerst kompetenter Mathematiker.`	`Standort: {location}.`	`Beachten Sie die Zahlenformatierung`	`lösen Sie die folgende Aufgabe: {problem}`

E Dataset Coverage and Evaluation Instance Breakdown

The evaluation design contains 258,400 prompt instances per evaluated model across three primary experimental tracks:

1. **Main Symbolic Operations (71,800 instances):** Comprises 7,000 equations across 10 formatting standards and nine ambiguity strata, balanced equally across addition, subtraction, multiplication, and division (1,750 equations per operator). Evaluating P1, P2, and P4 yields 7,000 instances each, while expanding P3 and P5 across all format-compatible locales yields 25,400 instances each.

2. **Localised Word Problems (141,200 instances):** Applies P2–P5 to 35,300 scenario-based questions across 23 country–language combinations in seven languages: English (18,500), Spanish (5,800), French (4,400), German (3,000), and 1,200 each in Mandarin Chinese, Korean, and Tamil. The evaluated country–language combinations are: `BE-en`, `BE-fr`, `CH-de`, `CH-en`, `CH-fr`, `CN-en`, `CN-zh`, `CO-en`, `CO-es`, `DE-de`, `DE-en`, `ES-en`, `ES-es`, `FR-en`, `FR-fr`, `IN-en`, `IN-ta`, `KR-en`, `KR-ko`, `UK-en`, `US-en`, `US-es`, and `ZA-en`.
3. **Multilingual Symbolic Ablation (45,400 instances):** Applies P2–P5 using translated prompt instructions and location cues across French (14,800 instances across 3,000 eligible equations), Spanish (20,000 instances across 4,200 eligible equations), and German (10,600 instances across 2,300 eligible equations).

F SOFTWARE

We implement our evaluation framework in Python. For local inference, we use `vLLM` (Kwon et al., 2023) with `Transformers` (Wolf et al., 2020) and model specific chat templates. We access hosted models through their providers' APIs.

G MODEL LICENCES

The released weights of `EuroLLM-22B-Instruct-2512`, `Qwen2.5-32B-Instruct`, `Apertus-8B-Instruct-2509`, and `Apertus-70B-Instruct-2509` are distributed under the Apache Licence 2.0.[2] The original `Mixtral-8x22B-Instruct-v0.1` and `mistralai/Ministral-3-8B-Instruct-2512` releases also use Apache 2.0.[3] For Mixtral, the configured AWQ checkpoint is a third-party quantisation of the original release. Its repository identifies the upstream model but does not provide a separate licence declaration.[4]

`Qwen2.5-72B-Instruct` is distributed under the Qwen Licence Agreement.[5] `Ministral-8B-Instruct-2410` is released under the Mistral Research License, which restricts use to research purposes.[6] `Aya-Expanse-32B` are distributed under CC BY-NC 4.0, together with Cohere's acceptable-use conditions.[7]

Access to hosted models is governed by the respective providers' service terms. OpenAI API use is covered by the OpenAI Services Agreement,[8] Hosted Mistral, DeepSeek, and Cohere models are accessed under their respective service terms.[9]

[2] `https://huggingface.co/utter-project/EuroLLM-22B-Instruct-2512`; `https://huggingface.co/Qwen/Qwen2.5-32B-Instruct`; `https://huggingface.co/swiss-ai/Apertus-8B-Instruct-2509`; `https://huggingface.co/swiss-ai/Apertus-70B-Instruct-2509`.

[3] `https://huggingface.co/mistralai/Mixtral-8x22B-Instruct-v0.1`; `https://huggingface.co/mistralai/Ministral-3-8B-Instruct-2512`.

[4] `https://huggingface.co/MaziyarPanahi/Mixtral-8x22B-Instruct-v0.1-AWQ`.

[5] `https://huggingface.co/Qwen/Qwen2.5-72B-Instruct/blob/main/LICENSE`.

[6] `https://huggingface.co/mistralai/Ministral-8B-Instruct-2410`.

[7] `https://huggingface.co/CohereLabs/aya-expanse-32b`

[8] `https://openai.com/policies/services-agreement/`.

[9] `https://legal.mistral.ai/terms/commercial-terms-of-service/`; `https://cdn.deepseek.com/policies/en-US/deepseek-terms-of-use.html`; `https://cohere.com/terms-of-use`.

H AMBIGUOUS TIER PERFORMANCE

Table 13: Value Correctness (VC %) performance across ambiguity split groups under incremental levels of contextual framing.

Split Group	Baseline	Culture	+Location	+ Language
Split 1: CD(444) vs. Comma-Decimal				
Comma Group (AC, ADC, C, DC, SC)	11.77	1.55	1.93	4.49
CD(444)	17.64	4.64	4.36	4.05
Split 2: [CD(333), CD(2223)] vs. Comma-Decimal				
Comma Group (AC, ADC, C, DC, SC)	11.82	2.00	2.08	5.88
CD(333), CD(2223)	55.64	11.57	11.32	11.17
Split 3: Dot-Decimal vs. [DC, ADC]				
ADC, DC	5.59	3.51	3.85	5.16
Dot Group (CD_444, CD_2223, CD_333, D, SD)	59.12	48.64	46.93	42.65