Meeseeks-Multilingual: A Multi-Turn, Multi-Constraint Benchmark Evaluating Multilingual Instruction-Following and Self-Correction

Anonymous EMNLP submission

Abstract

Large Language Models (LLMs) fail to reliably follow multi-constraint instructions across languages, and the failure rate rises when regional linguistic rules interact with general constraints. To address this challenge, we present **Meeseeks-Multilingual**, a multi-turn, feedback-driven benchmark that extends Meeseeks to multilingual and cross-regional evaluation. The benchmark separates general constraints (e.g., word counts, keyword rules) from regional constraints (e.g., Japanese kana requirements) so that the two failure modes can be measured independently. Meeseeks-Multilingual automatically identifies unmet constraints and provides targeted feedback to test iterative self-correction. All data are produced by trained annotators with no LLM involvement in data generation. Experiments across mainstream commercial and open-source LLMs show that even with iterative feedback, models consistently fail to meet all multilingual constraints. Aggregate model rankings stabilize by Round 5, which lets the benchmark be run at reduced cost for overall comparisons, although regional rankings settle more slowly and need more rounds. Meeseeks-Multilingual has been open-sourced on `https://anonymous.4open.science/r/Meeseeks-0A4A/`.

1 Introduction

Large Language Models (LLMs) are increasingly deployed in user-facing settings that span multiple languages and writing systems. In these settings, users issue prompts that combine several interdependent constraints, and many of those constraints depend on language-specific conventions rather than language-agnostic rules. Existing evidence shows that current frontier models still fail to consistently satisfy combined constraints, and the failure rate grows when prompts require numerical precision, structural adherence, or linguistic reasoning across languages (Zhou et al., 2023; He et al., 2024b; Jiang et al., 2024).

Existing instruction-following benchmarks, including IF-Eval (Zhou et al., 2023), CELLO (He et al., 2024a), FollowBench (Jiang et al., 2024), Multi-IF (He et al., 2024b), and the original Meeseeks benchmark (Zhao et al., 2025), have characterized model weaknesses primarily in single-turn or bilingual (Chinese and English) settings. They do not separate constraints that are language-agnostic, such as word counts, keyword requirements, and structural formatting, from constraints that are tied to a specific language, such as kana usage in Japanese, French grammatical gender agreement, or Arabic diacritical mark conventions. Multilingual self-correction, in which a model receives constraint-level feedback and rewrites in a non-English language, is also rarely covered despite being the common interaction pattern for non-English users. Iterative self-correction methods such as Self-Refine (Madaan et al., 2023), Reflexion (Shinn et al., 2023), and CRITIC (Gou et al., 2024) have been studied in English task settings, but whether their gains transfer to multilingual linguistic constraints remains an open question. The combined effect is that current benchmarks lack the resolution to diagnose where multilingual failure originates.

To address these gaps, we present **Meeseeks-Multilingual**, a multilingual extension of the Meeseeks benchmark designed to evaluate instruction-following under multi-constraint, multi-turn, and cross-lingual conditions. Our design introduces a dual-layer constraint framework that separates general constraints shared across languages from regional constraints tied to specific linguistic systems. This separation lets us locate where multilingual LLMs fail: at the universal level (e.g., miscounting words across languages) or at the regional level (e.g., misusing Japanese hiragana vs. katakana). As in Meeseeks, the framework identifies violated constraints,

generates script-driven feedback, and runs the model for multiple correction turns.

Our experiments across nine mainstream LLMs and 11 languages show that models struggle with multi-constraint multilingual instructions. Even after 9 rounds of feedback, the best model satisfies only 52% of all constraints, and averaged across models regional constraints (30.2%) lag behind general ones (58.5%). Models frequently misinterpret regional linguistic rules, and aggregate model rankings stabilize by Round 5 even as absolute scores continue to drift upward, while regional rankings settle more slowly.

Our contributions are as follows:

(1) **Multilingual multi-constraint instruction-following benchmark.** We extend Meeseeks to 11 languages spanning six language families and four writing systems, so that multi-constraint instruction-following can be evaluated outside Chinese and English.

(2) **Dual-layer constraint design separating general and regional knowledge.** We label each constraint as either general (e.g., word counts, keywords) or regional (e.g., Japanese hiragana vs. katakana selection, French grammatical gender), so that language-agnostic failures and language-specific failures can be measured separately rather than aggregated.

(3) **Round 5 lock-in for aggregate feedback-driven evaluation.** We show that the aggregate model ranking stabilizes by Round 5, with Kendall $\tau(R_5, R_9) \geq 0.9$ in all 9 leave-one-model-out replicates, whereas the regional-only ranking settles more slowly and is only approximately stable by Round 7. Future iterations can drop the last four rounds for overall comparisons, while regional comparisons benefit from keeping more rounds.

(4) **Failure-mode typology from iterative feedback.** By tracking per-constraint-type failure counts across rounds, we separate failures that respond to feedback (theme coverage, format, keyword embedding) from failures that do not (character-level counting, morphological agreement). The latter persist even when the feedback explicitly names the offending tokens, which points at a generation-time rather than knowledge-time limitation.

2 Related Work

2.1 Instruction-Following Evaluation

Instruction-following evaluation has moved from rule-verifiable single-turn tasks toward multi-turn and feedback-driven settings. IF-Eval (Zhou et al., 2023) introduced automatic constraint checking for 25 instruction types in synthetic one-turn prompts. Constraint complexity was then increased by COLLIE (Yao et al., 2024), with 1,132 compositional constraints at multiple textual levels, InFoBench (Qin et al., 2024), with a Decomposed Requirements Following Ratio for multi-criteria instructions, and ComplexBench (Wen et al., 2024), with multiple combined constraints, all within single-turn formats. Multi-turn evaluation appears in FollowBench (Jiang et al., 2024), with open-ended instructions judged by GPT-4, MT-Bench-101 (Bai et al., 2024), with multilingual three-turn dialogs derived from IF-Eval, and StructFlowBench (Li et al., 2025), with six common dialogue patterns under constraint-based scoring. Iterative self-correction methods, including Self-Refine (Madaan et al., 2023), Reflexion (Shinn et al., 2023), and CRITIC (Gou et al., 2024), demonstrate that LLMs can improve outputs through targeted feedback loops, though prior work has also shown fundamental limits of intrinsic self-correction without external grounding (Huang et al., 2024). Recent reinforcement-learning approaches such as SCoRe (Kumar et al., 2024) train models to self-correct via online RL rather than prompt-level feedback, pointing toward a complementary training-time solution. The remaining gap is feedback-driven instruction-following in multilingual settings, which Meeseeks-Multilingual addresses by extending the Meeseeks iterative self-correction setup beyond Chinese and English.

2.2 Multilingual Evaluation

Multilingual evaluation has shifted from English-centric benchmarks toward dedicated multilingual tests. MMLU (Hendrycks et al., 2020), XTREME (Siddhant et al., 2020), and XGLUE (Liang et al., 2020) assess cross-lingual knowledge and understanding using translated test sets and do not target generative compliance. Holistic evaluation frameworks such as HELM (Liang et al., 2022) complement these with multi-metric assessments spanning accuracy, robustness, and fairness. XNLI (Conneau et al., 2018) provides a foundational cross-lingual inference benchmark

in 15 languages. Broader multilingual suites including MEGA (Ahuja et al., 2023) and MEGAVERSE (Ahuja et al., 2024) evaluate generative LLMs across 70 and 83 languages respectively, while Global-MMLU (Singh et al., 2025) addresses cultural and linguistic biases in multilingual knowledge evaluation. Belebele (Bandarkar et al., 2024) extends reading comprehension to 122 language variants and MGSM (Shi et al., 2022) probes chain-of-thought reasoning in ten typologically diverse languages. INCLUDE (Romanou et al., 2025) provides 197k questions from regional exams across 44 written languages with region-specific knowledge requirements. Multilingual instruction-tuned models such as BLOOM (Workshop et al., 2022) and the Aya model (Üstün et al., 2024) highlight the need for instruction-following benchmarks that match their broad linguistic range. Multi-IF (He et al., 2024b) translates IF-Eval into seven languages with 4,500 three-turn dialogs and reports sharper declines on non-Latin languages. M-IFEval (Dussolle et al., 2025) extends IF-Eval to French, Japanese, and Spanish with verifiable metrics. SEA-HELM (Susanto et al., 2025) introduces culturally grounded evaluation for Southeast Asian languages. These benchmarks address multilingual coverage or multi-turn interaction but not the combination of the two with feedback-driven self-correction, which is the regime Meeseeks-Multilingual targets.

3 Meeseeks-Multilingual

Meeseeks-Multilingual extends the original Meeseeks benchmark (Zhao et al., 2025) to multilingual settings while preserving its core iterative self-correction framework. Given a multi-constraint instruction in a target language, each evaluation run proceeds through repeated cycles of (1) response collection, (2) automatic constraint checking, and (3) feedback prompting. We adapt the data and constraint taxonomy to a multilingual context by (i) extending to 11 languages, (ii) separating **general constraints** shared across languages from **regional constraints** capturing language-specific phenomena, and (iii) performing cross-lingual analysis over languages and scripts.

3.1 Dual-Layer Constraint Taxonomy

The original Meeseeks benchmark organizes constraints into a cognitive-inspired taxonomy with three evaluation dimensions (Intent Recognition, Granular Content Validation, and Output Structure Validation) spanning 32 capability tags. We retain this taxonomy but add a split between two constraint layers:

General Constraints capture instruction-following requirements that apply uniformly across all languages. These include word or character count control, keyword inclusion or exclusion, output format requirements (e.g., lists, tables, headings), rhetorical style specifications, and structural organization rules. A general constraint might require, for instance, that the response contain between 150 and 200 words, or that a particular keyword appear at least three times. Constraints of this kind test instruction-following abilities that should not depend on the surface language of the prompt.

Regional Constraints capture language-specific linguistic phenomena that require knowledge of a particular language's writing system, morphology, or stylistic conventions. Examples include Japanese hiragana vs. katakana script selection, Chinese classical vs. modern prose style, Korean honorific level requirements (Son et al., 2024), French accent-mark counts and aspirated-H usage, German compound word formation, Arabic dual-number and broken-plural agreement (Mousi et al., 2025), and Russian verbal aspect. These constraints test whether a model has the language-specific knowledge needed to produce a compliant output, rather than the surface-level competence sufficient for translation.

The split supports a targeted diagnosis: a model that performs well on general but poorly on regional constraints likely has working instruction-following machinery but lacks the relevant language knowledge, whereas a model that fails on both has broader compliance issues.

3.2 Evaluation

Meeseeks-Multilingual adopts a two-stage evaluation pipeline. In the first stage, an LLM evaluator (Zheng et al., 2023) normalizes the tested model's output into a structured representation and extracts the components relevant to each constraint. In the second stage, language-aware Python scripts verify whether each constraint is satisfied. Each constraint is labeled with a Capability Rule (e.g., word-count control), a Constraint Rule (e.g., range, exact match), and optionally a Regional Rule (e.g.,

Japanese kana usage). General constraints are checked by language-independent scripts, and regional constraints are checked by language-specific modules that handle script identification, orthographic conventions, and morphological analysis. Constraints with no rule label are judged directly by the LLM evaluator.

Due to hallucinations in LLM-based evaluation, directly relying on models to determine constraint satisfaction is unreliable (Zhao et al., 2025). Therefore, our primary evaluation method uses deterministic scripts wherever possible, with LLM judgment reserved only for subjective or context-dependent constraints that resist rule-based checking.

3.3 Feedback Prompting

After evaluation, the pipeline generates constraint-level feedback for each unmet constraint. A feedback message is specific and actionable, for example reporting that the output contains 130 words against a required range of 150 to 180, or that a term written in katakana should be written in hiragana. The majority of feedback is synthesized by deterministic scripts from the evaluation result. The remainder is produced by the LLM evaluator for constraints that require natural language explanation.

Feedback is injected into the chat template for the next turn, forming a multi-turn dialogue where the model receives its previous response along with constraint-level error descriptions. Feedback is also language-aware. It uses the conventions of the target language, counting words for English and characters for Chinese, and referencing keywords in the original script rather than in translation. This keeps the feedback format consistent with the language and cultural context of the original instruction.

3.4 Code-Guided Evaluation

Following the original Meeseeks code-guided evaluation approach, we instruct the evaluator to write extraction code rather than regenerate entire responses. A Python function might parse the response into a list of section titles for counting, or extract all instances of a particular script type. Two properties make this useful in the multilingual setting. Language-specific logic, such as counting Japanese kana or identifying Arabic diacritical marks, is isolated in dedicated modules that can be independently validated. Extraction also targets only the spans needed for checking, which reduces token consumption. The original Meeseeks work reports a token reduction of over 70% and higher evaluation accuracy relative to regeneration-based formatting.

3.5 Data Collection and Quality

Meeseeks-Multilingual extends the original Meeseeks dataset (Chinese and English, 32 capability tags) to 11 languages spanning six language families (Sino-Tibetan, Japonic, Koreanic, Indo-European, Austronesian, Afro-Asiatic) and four writing systems (Latin, CJK, Cyrillic, Arabic). We retain the cognitive-inspired three-dimensional taxonomy but reorganize it into general and regional layers, so that each language carries both shared and language-specific constraints.

All instances are constructed by professional annotators and language experts, each holding at least a bachelor's degree in a relevant field. Regional constraints are authored natively in the target language rather than translated, which avoids artifacts from source-language interference. No LLM is involved in data generation or translation, which removes the risk of contamination from model-produced text. Each instance is reviewed by at least two annotators, with disagreements resolved through discussion. Annotators also verify that constraints within an instance are logically consistent, for example that a word-count requirement does not conflict with a brevity requirement.

3.6 Metrics

The primary metric is the **Meeseeks-Multilingual Score**, computed as the proportion of constraints satisfied by a model's response. The score is reported in aggregate and decomposed by language, by constraint type (general or regional), and by capability dimension. The General and Regional scores are the satisfaction proportions within each constraint layer, and the reported Overall score is the unweighted mean of these two layer scores rather than the pooled proportion over the combined constraint set. We use this macro-average so that the general and regional layers contribute equally even when they contain different numbers of constraints. All numbers reported in this paper use this metric unless explicitly noted.

4 Experiments

4.1 Experimental Setup

Languages. To ensure broad coverage of typologically diverse writing systems and resource

levels, we select 11 languages spanning six language families and four scripts: Chinese (ZH), English (EN), Japanese (JA), Korean (KO), Spanish (ES), French (FR), German (DE), Portuguese (PT), Russian (RU), Arabic (AR), and Indonesian (ID). Language selection is guided by a resource analysis combining Wikipedia article counts and Hugging Face model metadata (details in Appendix A).

Models. We evaluate nine mainstream LLMs covering both commercial and open-source families: GPT-4.1 and GPT-4o (OpenAI), DeepSeek-V3 and DeepSeek-V3-0324, Doubao-1.6 (ByteDance), Longcat-Chat and Longcat-Eco, Qwen-32B-Think and Qwen-A22B (Alibaba). All models are evaluated via their respective APIs with default generation parameters.

Evaluation Protocol. Following the Meeseeks framework, each model receives a multi-constraint instruction in a target language and generates a response (Round 0). The evaluation pipeline then checks every constraint and provides targeted feedback for unmet ones. The model receives this feedback and generates a corrected response, repeating for up to 9 rounds of iterative self-correction (10 total responses). We use GPT-4.1 as the LLM evaluator across all experiments to ensure consistent judgment quality.

4.2 Main Results

Table 1 presents the overall performance of all models at Round 0 (initial response) and Round 9 (after 9 rounds of feedback-driven self-correction), decomposed into general and regional constraints.

Across the 11 languages, regional constraints are consistently harder than general ones throughout the protocol. At Round 0 the average regional score is 7.0% against 33.7% for general, and the gap shrinks only modestly with feedback. By Round 9 general reaches 58.5% and regional reaches 30.2%, a 28.3 percentage-point (pp) gap. Even the strongest model, GPT-4.1, satisfies only 52.0% of all constraints after nine feedback turns. Average Round 0 to Round 9 gains are 24.8pp on general and 23.2pp on regional, so feedback raises both constraint types by a similar amount even though the regional ceiling remains low.

The top of the leaderboard reshuffles between the two constraint types. GPT-4.1 leads on overall Round 9 score (52.0%) and on regional Round 9 score (38.5%), while Doubao-1.6 has the highest general Round 9 score (65.7%). Qwen-32B-Think is an outlier with an overall Round 9 score of 27.2%, well below the other eight systems.

Figure 1 shows the self-correction trajectories.

Model	General		Regional		Overall
	R0	R9	R0	R9	R9
GPT-4.1	37.4	65.5	**10.9**	**38.5**	**52.0**
Doubao-1.6	34.0	**65.7**	6.2	30.7	48.2
DeepSeek-V3-0324	32.2	64.3	6.6	30.1	47.2
DeepSeek-V3	34.0	60.4	7.1	32.7	46.5
GPT-4o	31.8	60.3	6.7	31.7	46.0
Qwen-A22B	34.5	57.1	5.7	32.5	44.8
Longcat-Chat	33.9	57.4	6.8	30.0	43.7
Longcat-Eco	39.1	61.1	7.1	26.0	43.6
Qwen-32B-Think	26.1	35.1	6.3	19.3	27.2
Average	*33.7*	*58.5*	*7.0*	*30.2*	*44.4*

Table 1: Main results: average constraint satisfaction scores (%) across 11 languages. R0 is the initial response and R9 is the response after 9 rounds of feedback-driven self-correction. Models are ranked by overall R9 score. General constraints are shared across languages, while regional constraints are language-specific. The Overall column is the unweighted mean of the General and Regional R9 scores.

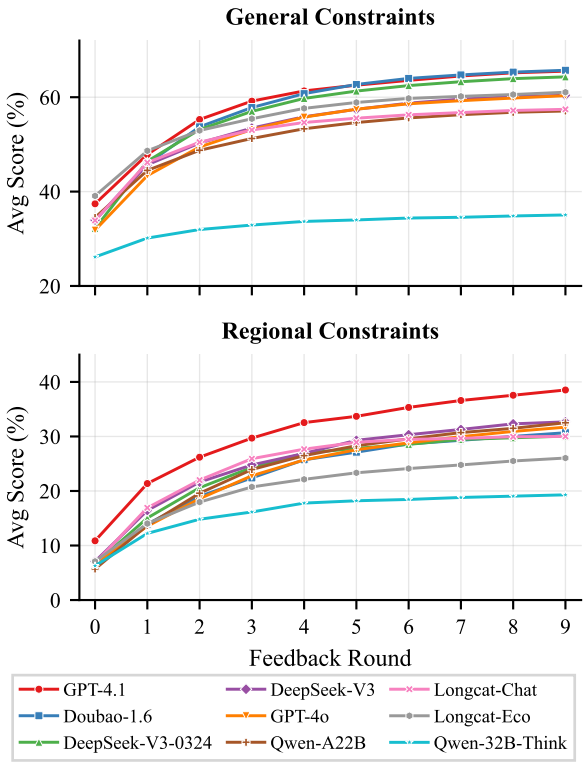

Figure 1: Self-correction trajectories from Round 0 through Round 9, averaged over 11 languages. Top: general constraints. Bottom: regional constraints. All models show diminishing returns, with most gains realized in the first few rounds of feedback.

Most models improve quickly during the first few rounds of feedback and approach a plateau by Round 7. GPT-4.1 and DeepSeek-V3-0324 show the steepest initial slope on general constraints, while the regional curves are flatter and more spread out across systems. Qwen-32B-Think barely improves on the regional side at any round.

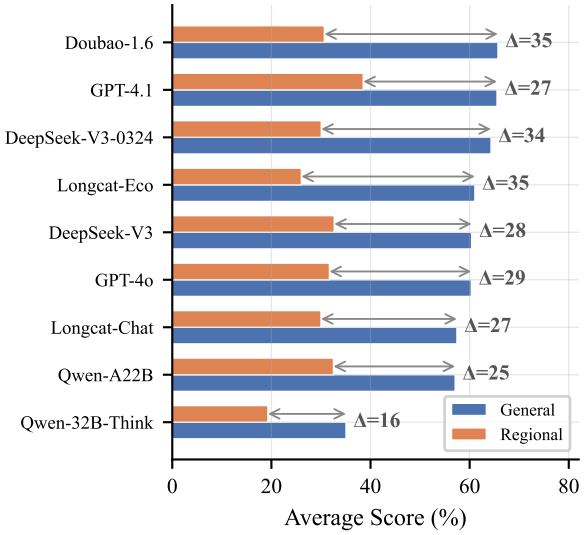

Figure 2: General versus regional scores at Round 9 for each model. Models converge toward a similar general-constraint band while remaining widely dispersed on regional constraints.

By Round 9, general scores cluster between 55% and 66% across most systems, whereas regional scores remain widely dispersed from 19% to 39% (Figure 2). The regional dispersion is consistent with the failure-mode analysis in Section 5.3. The per-cell breakdown in Figure 5 shows that French and Japanese regional cells stay below 20% for all systems, while Spanish and Korean cells reach into the 50% to 70% range.

5 Analysis

5.1 Ranking Stability and the Round 5 Lock-in

How many feedback rounds does the benchmark need before model comparisons are reliable? We answer this by computing Kendall τ between the model ranking at Round k and the final Round 9 ranking. Averaged across the 11 languages and over the full constraint set, $\tau(R_5, R_9) = 0.94$ on the full nine-model pool, and $\tau(R_5, R_9) \geq 0.93$ in all 9 leave-one-model-out replicates, where each replicate drops one model from the pool and recomputes the rank correlation on the remaining

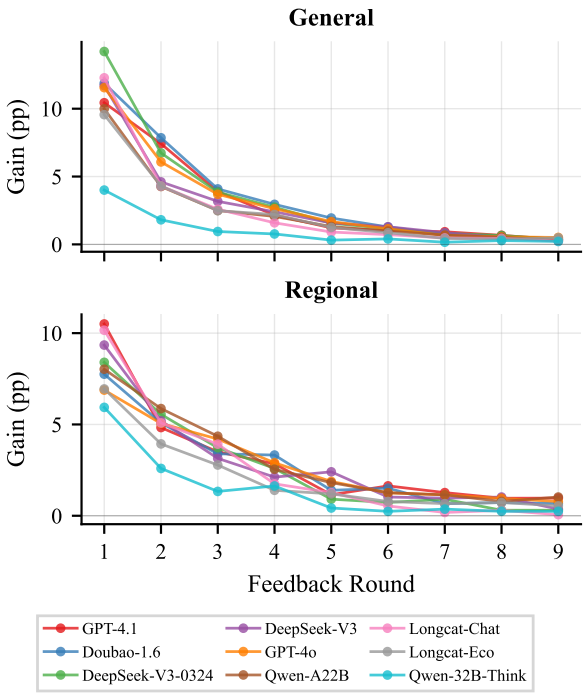

Figure 3: Marginal score gain per feedback round across all models. The plateau after Round 5 coincides with the convergence of the aggregate inter-round model rankings.

8 systems. The regional-only ranking is less stable: $\tau(R_5, R_9) = 0.67$ across all 11 languages, and $\tau(R_7, R_9) = 0.83$, indicating that regional cells need more rounds than general ones to settle. Absolute scores continue to drift upward beyond Round 5 (Figure 3), but the aggregate relative ordering of systems is already settled. Truncating the benchmark at Round 5 preserves this aggregate ordering while removing 4 of the 10 evaluation rounds. Studies that compare systems at the regional level should keep additional rounds, since the regional ordering is only approximately stable by Round 7.

5.2 Per-Language Convergence and the Three-Tier Regional Pattern

The aggregate trajectories in Figure 1 can hide language-specific structure, so we also average within each language over all nine models. Figure 4 shows that general constraints behave almost uniformly across languages, with every curve entering the 53% to 62% band by Round 9. Regional constraints separate into three groups. Spanish, Korean, and German lead with R9 averages of 58%, 47%, and 40%. Chinese, English, Portuguese, Indonesian, and Russian sit between 25% and 37%. Arabic, Japanese, and French stay below

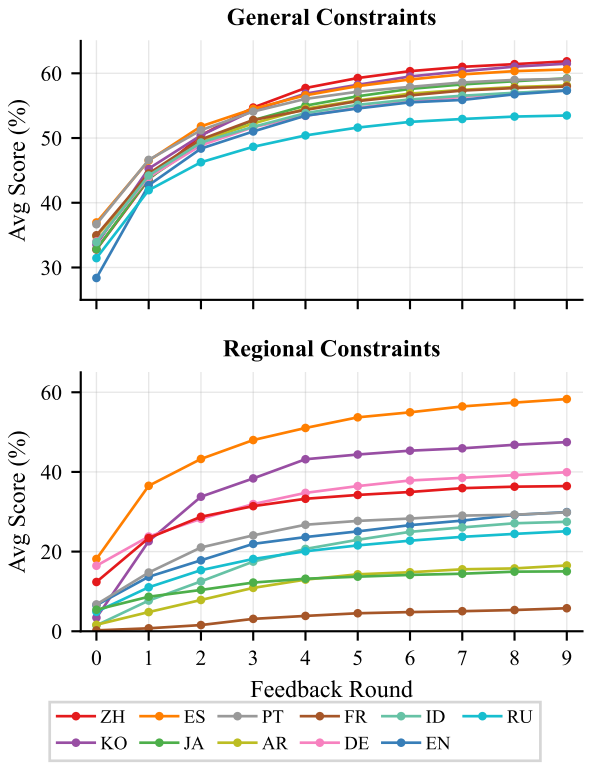

Figure 4: Per-language self-correction curves averaged across all nine models. Top panel shows general constraints, where every language climbs into the 53% to 62% band by Round 9. Bottom panel shows regional constraints. Spanish, Korean, and German top out at 58%, 47%, and 40%, while Arabic, Japanese, and French stay below 20%.

20% even after nine rounds of feedback. The split is broadly stable across rounds ($\tau(R_1, R_9) = 0.78$ on language-level regional scores). The lowest group is dominated by character-level and morphological constraints (French accent counting and aspirated-H, Japanese kana selection, Arabic verbal morphology), which we analyze in the next subsection.

5.3 Failure Modes Separated by Feedback Responsiveness

The 10-round protocol with script-derived feedback exposes a second axis along which constraints differ. We track per-constraint-type failure counts across rounds and group constraints by how much their failure count drops over the 9 feedback turns.

One group improves quickly and accounts for most of the R0 to R5 gain on general tasks. The relevant constraints are theme coverage, format requirements, and keyword embedding. Once the feedback names the missing element, the model adds it on the next pass. We refer to these as *instructable failures*.

French constraint type	R0	R5	R9
Exact-count	37	29	26
Multi-paragraph range	111	96	89
Accent-mark counting	99	91	86
Aspirated-H	30	29	29

Table 2: Failure counts on four French regional constraint types across feedback rounds for DeepSeek-V3. Exact-count and multi-paragraph constraints respond to feedback. Character-level constraints such as accent-mark counting and aspirated-H usage barely move even though the feedback explicitly lists the offending words.

A second group barely responds to feedback even when the feedback is precise. Table 2 reports per-round failure counts for four French regional constraint types under DeepSeek-V3. Exact-count constraints lose 30% of their R0 failures over 9 rounds and multi-paragraph range constraints lose 20%. Accent-mark counting loses only 13%, and the aspirated-H failure count moves from 30 to 29 over the same 9 rounds. The pattern recurs on Arabic morphological constraints such as dual-number forms, broken plurals, and gender-agreement counts. We refer to these as *architectural failures*. The model receives the correct error description but cannot produce a compliant response.

A French and an Arabic case make this concrete. A French prompt requires four paragraphs of at least 120 words each, exactly 20 acute or grave accented words per paragraph, and at least three occurrences per paragraph of each of the keywords *extrêmement*, *seulement*, and *puis*. After nine rounds of feedback that names the specific tokens carrying the wrong accent and the exact undercount on each keyword, DeepSeek-V3 still returns paragraphs in which *extrêmement* occurs zero times in three of the four paragraphs and the accent count is off in every paragraph. An Arabic prompt asks for a five-paragraph article where each paragraph contains exactly five third-person masculine singular present-tense verb forms. DeepSeek-V3 produces zero such verb forms in any paragraph at Round 9, even though each round of feedback enumerates the verbs the model used and the forms it should have used instead. The French failure is orthographic and the Arabic failure is morphological, yet in both cases the feedback message reaches the model unchanged across rounds and the response does not improve.

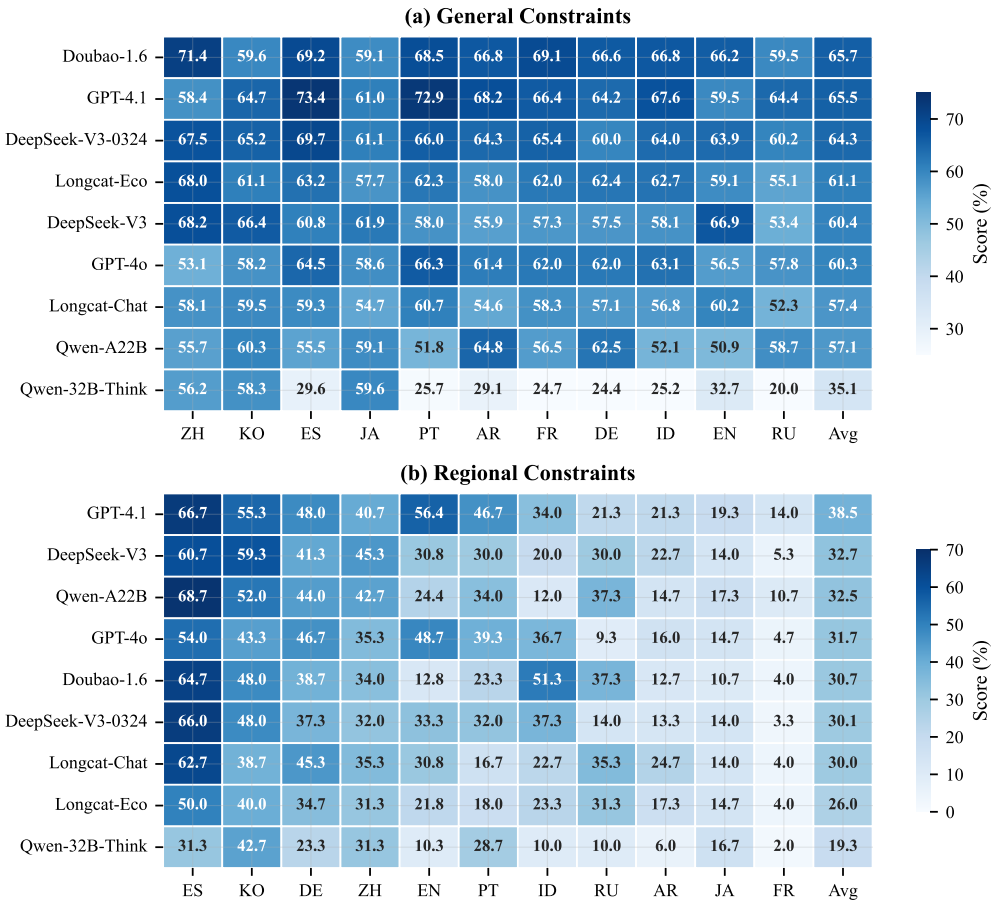

Figure 5: Model×language heatmaps of Round 9 scores. Top panel shows general constraints, where all but Qwen-32B-Think cluster between 50% and 70%. Bottom panel shows regional constraints, where French and Japanese cells stay below 20% for every system while Spanish and Korean cells reach 50% to 70%.

The two classes have different implications. Instructable failures can be closed by targeted post-training or prompt design. Architectural failures reflect a generation-time limitation. An autoregressive decoder cannot reliably track a running character-level or morphological count while producing fluent text, no matter how precise the error message is. The separation is only visible when feedback is both script-derived and applied for enough rounds to let the architectural class plateau. Multilingual benchmarks that cap at three turns (He et al., 2024b; Bai et al., 2024) cannot make this distinction, because the instructable class has not yet finished improving by Round 3 and the two groups still overlap in the aggregate.

6 Discussion

The split between instructable and architectural failures suggests where progress will come from. Targeted prompting and post-training (Ouyang et al., 2022; Chung et al., 2024) are likely to close the instructable class. The architectural class is harder, since an autoregressive decoder cannot reliably track a running character-level or morphological count while producing fluent text. Constraint-aware decoding, tool-augmented generation (Gou et al., 2024), and RL regimes that optimize for self-correction (Kumar et al., 2024) are more promising for that class. Other extensions include per-instance logging for finer error analysis, a no-feedback regeneration control to separate feedback from added generation budget, and adding more morphologically rich languages.

7 Conclusion

We introduce Meeseeks-Multilingual, a multilingual, multi-constraint benchmark with a dual-layer general/regional taxonomy and a feedback-driven self-correction protocol across 11 languages. Across nine mainstream LLMs, regional constraints lag general ones by 28.3pp at Round 9, aggregate model rankings stabilize at Round 5, and character-level and morphological failures barely respond to feedback. The benchmark, evaluation scripts, and per-instance counts are open-sourced.

Limitations

Meeseeks-Multilingual measures multilingual instruction-following compliance and does not attempt to score task correctness or output quality independently of the stated constraints. The dual-layer taxonomy was designed to capture the most common general and regional phenomena across the 11 target languages but does not enumerate every language-specific construct of interest. The dataset is professionally curated, and per-cell instance counts are reported in the released artifact rather than in the main text.

Two methodological caveats apply to the experimental results. The LLM evaluator across all runs is GPT-4.1, and GPT-4.1 itself is one of the nine tested models, so we cannot rule out evaluator self-preference, a recognized limitation in LLM-as-judge evaluation (Zheng et al., 2023). The use of deterministic scripts for the majority of constraint checks does limit the room for this bias to express itself. The reported Round 0 to Round 9 gains also conflate the effect of feedback with the effect of additional generation budget, since we do not run a no-feedback regeneration control. We also refrain from introducing additional prompting strategies such as chain-of-thought templates (Wei et al., 2022), in order to keep the evaluation comparable across models.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, and 1 others. 2023. Mega: Multilingual evaluation of generative ai. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267.

Sanchit Ahuja, Divyanshu Aggarwal, Varun Gumma, Ishaan Watts, Ashutosh Sathe, Millicent Ochieng, Rishav Hada, Prachi Jain, Mohamed Ahmed, Kalika Bali, and 1 others. 2024. Megaverse: Benchmarking large language models across languages, modalities, models and tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2598–2637.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and 1 others. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2475–2485.

Antoine Dussolle, Andrea Cardeña Díaz, Shota Sato, and Peter Devine. 2025. M-ifeval: Multilingual instruction-following evaluation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6176–6191.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, and 1 others. 2024. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pages 57734–57811.

Qianyu He, Jie Zeng, Wenhao Huang, Lina Chen, Jin Xiao, Qianxi He, Xunzhe Zhou, Jiaqing Liang, and Yanghua Xiao. 2024a. Can large language models understand real-world complex instructions? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18188–18196.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, and 1 others. 2024b. Multi-if: Benchmarking llms on multi-turn and multilingual instructions following. *arXiv preprint arXiv:2410.15553*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *International conference on learning representations*, volume 2024, pages 32808–32824.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. Followbench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, and 1 others. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.

Jinnan Li, Jinzhe Li, Yue Wang, Yi Chang, and Yuan Wu. 2025. Structflowbench: A structured flow benchmark for multi-turn instruction following. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9322–9341.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and 1 others. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2025. Aradice: Benchmarks for dialectal and cultural capabilities in llms. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4186–4218.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048.

Angelika Romanou, Negar Foroutan, Anna Sotnikova, Sree Harsha Nelaturu, Shivalika Singh, Rishabh Maheshwary, Micol Altomare, Zeming Chen, Mohamed Haggag, Alfonso Amayuelas, and 1 others. 2025. Include: Evaluating multilingual language understanding with regional knowledge. In *International Conference on Learning Representations*, volume 2025, pages 83291–83322.

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. 2022. Language models are multilingual chain-of-thought reasoners.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Aditya Siddhant, Junjie Hu, Melvin Johnson, Orhan Firat, and Sebastian Ruder. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of the International Conference on Machine Learning*, volume 2020, pages 4411–4421.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, and 1 others. 2025. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799.

Guijin Son, Hanwool Lee, Suwan Kim, Huiseo Kim, Jae cheol Lee, Je Won Yeom, Jihyu Jung, Jung woo Kim, and Songseong Kim. 2024. Hae-rae bench: Evaluation of korean knowledge in language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7993–8007.

Yosephine Susanto, Adithya Venkatadri Hulagadri, Jann Railey Montalan, Jian Gang Ngui, Xianbin Yong, Wei Qi Leong, Hamsawardhini Rengarajan, Peerat Limkonchotiwat, Yifan Mai, and William Chandra Tjhi. 2025. Sea-helm: Southeast asian holistic evaluation of language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12308–12336.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, and 1 others. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd annual meeting of the Association for Computational Linguistics (volume 1: Long papers)*, pages 15894–15939.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, and 1 others. 2024. Benchmarking complex instruction-following with multiple constraints composition. *Advances in Neural Information Processing Systems*, 37:137610–137645.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, and 1 others. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Shunyu Yao, Howard Chen, Austin Hanjie, Runzhe Yang, and Karthik Narasimhan. 2024. Collie: Systematic construction of constrained text generation tasks. In *International Conference on Learning Representations*, volume 2024, pages 38595–38606.

Yunke Zhao, Peng Ding, Jun Kuang, Yibin Shen, Zhe Tang, Yilin Jin, ZongYu Wang, Xiaoyu Li, Xuezhi Cao, Xunliang Cai, and 1 others. 2025. Meeseeks: A feedback-driven, iterative self-correction benchmark evaluating llms' instruction following capability. *arXiv preprint arXiv:2504.21625*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Language Selection via Resource Statistics

To ensure that the dataset covers mainstream languages with reasonable proportions, we conduct a language-resource analysis using publicly available statistics. We compute approximate language distributions from Wikipedia, using the number of articles per language as a proxy for written content volume. We also analyze Hugging Face model metadata, counting how many released models primarily target each language as an indicator of community and tooling support.

Based on this analysis, we select 11 languages spanning six language families (Sino-Tibetan, Japonic, Koreanic, Indo-European, Austronesian, Afro-Asiatic) and four writing systems (Latin, CJK, Cyrillic, Arabic). The selected languages are Chinese, English, Japanese, Korean, Spanish, French, German, Portuguese, Russian, Arabic, and Indonesian, covering both high-resource and moderately-resourced settings.

B Detailed Per-Language Results

Tables 3 and 4 present the full per-language breakdown of Round 9 scores for general and regional constraints respectively.

C Feedback Prompt Template

The feedback prompt used in each turn after Round 0 is the verbatim template from the released benchmark code. The model receives its prior question, a list of failed sub-question descriptions with their script-derived error explanations, and its previous answer.

```
In the previous round of conversation, your answer did not meet the user's requirements. Now you need to correct your previous answer. I will provide you with the [Original Question], [Unmet Requirements from Previous Round], and [Previous Answer]. Please provide your corrected answer based on this information. Note: Only output the answer, do not output any additional information.
[Original Question]: ⟨question⟩
[Unmet Requirements from Previous Round]: ⟨errors⟩
[Previous Answer]: ⟨previous_answer⟩
[Your Corrected Answer]:
```

Each entry in `errors` is built from a sub-question that failed evaluation, using the form "`⟨sub-question⟩ was not satisfied. Specific unmet information: ⟨eval_explanation⟩`". The `eval_explanation` is generated by the deterministic rule-based checker for token-level checks (it lists offending tokens, observed and required counts, or character-range violations) and by the LLM evaluator for sub-questions that need a linguistic judgement. The same template is reused in every round and in every target language.

D Representative Regional Constraints

For each of the 11 target languages, we list capability tags used by the regional checks in the released benchmark and give one real prompt from

Model	ZH	EN	JA	KO	ES	FR	DE	PT	RU	AR	ID	Avg
GPT-4.1	58.4	59.5	61.0	64.7	**73.4**	66.4	64.2	**72.9**	**64.4**	**68.2**	**67.6**	65.5
Doubao-1.6	**71.4**	66.2	59.1	59.6	69.2	**69.1**	**66.6**	68.5	59.5	66.8	66.8	**65.7**
DeepSeek-V3-0324	67.5	63.9	61.1	65.2	69.7	65.4	60.0	66.0	60.2	64.3	64.0	64.3
Longcat-Eco	68.0	59.1	57.7	61.1	63.2	62.0	62.4	62.3	55.1	58.0	62.7	61.1
DeepSeek-V3	68.2	**66.9**	**61.9**	**66.4**	60.8	57.3	57.5	58.0	53.4	55.9	58.1	60.4
GPT-4o	53.1	56.5	58.6	58.2	64.5	62.0	62.0	66.3	57.8	61.4	63.1	60.3
Longcat-Chat	58.1	60.2	54.7	59.5	59.3	58.3	57.1	60.7	52.3	54.6	56.8	57.4
Qwen-A22B	55.7	50.9	59.1	60.3	55.5	56.5	62.5	51.8	58.7	64.8	52.1	57.1
Qwen-32B-Think	56.2	32.7	59.6	58.3	29.6	24.7	24.4	25.7	20.0	29.1	25.2	35.1

Table 3: Per-language general constraint scores (%) at Round 9. Bold indicates the best score per column.

Model	ZH	EN	JA	KO	ES	FR	DE	PT	RU	AR	ID	Avg
GPT-4.1	40.7	**56.4**	**19.3**	55.3	66.7	**14.0**	**48.0**	**46.7**	21.3	21.3	34.0	**38.5**
DeepSeek-V3	**45.3**	30.8	14.0	**59.3**	60.7	5.3	41.3	30.0	30.0	22.7	20.0	32.7
Qwen-A22B	42.7	24.4	17.3	52.0	**68.7**	10.7	44.0	34.0	**37.3**	14.7	12.0	32.5
GPT-4o	35.3	48.7	14.7	43.3	54.0	4.7	46.7	39.3	9.3	16.0	36.7	31.7
Doubao-1.6	34.0	12.8	10.7	48.0	64.7	4.0	38.7	23.3	**37.3**	12.7	**51.3**	30.7
DeepSeek-V3-0324	32.0	33.3	14.0	48.0	66.0	3.3	37.3	32.0	14.0	13.3	37.3	30.1
Longcat-Chat	35.3	30.8	14.0	38.7	62.7	4.0	45.3	16.7	35.3	**24.7**	22.7	30.0
Longcat-Eco	31.3	21.8	14.7	40.0	50.0	4.0	34.7	18.0	31.3	17.3	23.3	26.0
Qwen-32B-Think	31.3	10.3	16.7	42.7	31.3	2.0	23.3	28.7	10.0	6.0	10.0	19.3

Table 4: Per-language regional constraint scores (%) at Round 9. Bold indicates the best score per column. French regional constraints are the most challenging across all models.

the dataset with an English translation. The capability tags below are taken from the per-sub-question capability field of the released dataset and the prompts are verbatim instances (some shortened with `[. . . ]` for space). For Russian and Arabic, prompts in the dataset are written in Cyrillic and Arabic script. We give a plain transliteration here for typographic reasons.

Chinese (ZH). Regional capability tags include special prosody, character shape structure, strokes, tones, palindrome layout, pinyin ordering, forward and reverse word chains, homophones, number chains, radicals, and rhyme.

> [...] 请你围绕场景创作一副对联，对联要求如下：上联和下联的总字数是18字，且上联和下联必须都满足回文的结构，横批和对联都要尽量贴近主题。
> *[...] Write a couplet on the given scene. The combined length of the two lines is 18 characters, both lines must form a palindrome, and the horizontal banner and the couplet should stay close to the theme.*
> Capability tag: Chinese palindrome layout. Rule: `has_palindrome`.

English (EN). Regional capability tags include English compounds, English capitalization, and English-Chinese code-mixing.

> [...] Your response must contain at least 5 compound words, here compound words only refer to words like *Mike's* or *follow-up*, where each word has either a hyphen (-) or an apostrophe (') in the middle.
> (Prompt is in English; no separate translation.)
> Capability tag: English compound words. Rule: `compound_word_num:[5,5]`.

Japanese (JA). Regional capability tags include small kana, hiragana–katakana–kanji mixing, long-vowel mark, dakuten and handakuten, the Japanese repetition mark 々, okurigana patterns, and furigana patterns.

> 李さんは日本語を勉強していて、今日はちょうど仮名の送り仮名を学んでいます。日本語の五十音図におけるあ行を語彙の読み始めとしてください。 5 のふりかなパターン (漢字の後にかな) が正確に存在する必要があります，また、各語彙の文字数は2文字数とする。
> *Mr. Li is learning Japanese and today is studying okurigana. Use the あ-row of the Japanese syllabary as the reading-onset of the vocabulary. Exactly 5 okurigana patterns (kana following kanji) must appear, and each word must be 2 characters long.*
> Capability tag: Japanese okurigana pattern. Rule: `has_kanji_okurigana_pattern:5`.

Korean (KO). Regional capability tags include interrogative words, double consonants, honorific levels, and consonant-only abbreviations.

> 저와 여동생은 암호 해독 게임을 좋아합니다. [...] 오늘 제 여동생이 【ㄴㄴㅂㅂㄷ】이라는 줄인말을 저에게 보냈는데 5 가지 합리적인 추측을 부탁드립니다. 1. xxx, 2. xxx, 3. xxx 형식으로 써 주세요.

> *My sister and I love code-breaking games. [...] Today my sister sent me the abbreviation "ㄴㄴㅂㅂㄷ". Please make 5 reasonable guesses, formatted as "1. xxx, 2. xxx, 3. xxx".*
> Capability tag: Korean consonant-only abbreviation. Rule: `has_korean_abbreviation:ㄴㄴㅂㅂㄷ`.

Spanish (ES). Regional capability tags include written-accent counts, interrogative-word stress, and Spanish-specific number, currency, and phone formats.

> [...] Debe crear una lista de precios completa que incluya: 1. el precio de venta de cada oficina y su número de planta correspondiente [...]. 2. una descripción detallada de cada oficina (que no supere los 70 palabras). 3. Contiene8 acentos.
> *[...] You must produce a complete price list including: 1. the sale price of each office and its corresponding floor number [...]. 2. a detailed description of each office (no more than 70 words). 3. Contains 8 written accents.*
> Capability tag: Spanish written-accent count. Rule: `has_spanish_accent_count:[8,8]`. The original prompt writes "Contiene8 acentos" without a space; we keep that form verbatim.

French (FR). Regional capability tags include French word count, diacritics, special prosody, aspirated vs. mute H, adverbial pronouns, partitive articles, the passé composé, French spacing rules, pronominal verbs, and filler words.

> [...] Veuillez composer un court poème sur le thème de l'enfance. Le poème comprend 2 strophes, chacune composée d'un quatrain complet. Dans chaque strophe, le nombre de mots avec la crédille doit être exactement de 3. De plus, chaque strophe doit respecter le schéma des rimes plates. [...]
> *[...] Write a short poem on the theme of childhood. The poem has 2 stanzas, each a complete quatrain. In each stanza, the number of words containing a cedilla must be exactly 3. Each stanza must also follow the AABB rhyme scheme. [...]*
> Capability tag: French diacritics. Rule: `french_cedilla_total:[3]`. The original prompt writes "crédille" (a misspelling of *cédille*); we keep it verbatim.

German (DE). Regional capability tags include diminutive identification with singular/plural counts, total definite-article counts, and German numerical constraints on length, parity, and monotonicity.

> Schreibe eine Märchengeschichte mit ungefähr 150 Wörtern. Die Hauptfigur der Geschichte ist ein kluger kleiner Fuchs, der in einem magischen Wald lebt. [...] Bitte beachten Sie, dass Ihre Antwort mindestens 5 Diminutivformen im Singular enthalten muss.
> *Write a fairy tale of about 150 words. The protagonist is a clever little fox living in a magical forest. [...] Please note that your answer must contain at least 5 singular diminutive forms.*
> Capability tag: German diminutive identification with number and count. Rule: `german_text_diminutive_words:[5,5]`.

Portuguese (PT). Regional capability tags include acute and circumflex accents, double negation, address abbreviations, nasal-vowel marks (∼), Portuguese number expressions, ordinal abbreviations, rhyme, date formats, negative-reply formats, and the cedilla *ç*.

> [...] Por favor, escreva um diário para o dia da perspectiva de Pedro e Sofia, cada diário deve conter pelo menos 400 palavras, e exatamente 7 palavras que contenham vogais nasalizadas marcadas com o til (∼).
> *[...] Please write a diary for the day from the perspective of Pedro and Sofia, each diary must contain at least 400 words and exactly 7 words containing nasalized vowels marked with the tilde (∼).*
> Capability tag: Portuguese nasal-vowel mark. Rule: `has_nasal_vowel:[7,7]`.

Russian (RU). Regional capability tags include stress-distinguishing homographs, accusative time expressions, prepositional time expressions, and evaluative suffixes.

> Napishite rech' dlya vyborov starosty klassa prodolzhitel'nost'yu okolo 4 minut. V tekste dolzhno byt' 2 vkhozhdenii slova *muka*, prichem napisanie slova ostaetsya neizmennym, no ego znachenie i udarenie razlichayutsya v zavisimosti ot konteksta. Tekst dolzhen sostoyat' iz 5 abzatsev.
> *Write a campaign speech for class representative, about 4 minutes long. The text must contain 2 occurrences of the word* muka*, written identically but with different meanings and stresses depending on context. The text must consist of 5 paragraphs.*
> Capability tag: Russian stress-distinguishing homographs. Rule: `rus_stress_homonym_usage:`*`muka`*`:2`.

Arabic (AR). Regional capability tags include dual-form endings, broken plurals, independent personal pronouns, masculine–feminine ratio, definite nouns, verb conjugation, idafa (genitive construct), rhyme, and sound masculine plurals.

> Ana ata'allam jam' al-taksir fi al-lugha al-'arabiyya. Yurja taqdim 18 kalima min naw' jam' al-taksir [...]. La yukarrar ayy min al-kalimat. [...]
> *I am learning broken plurals in Arabic. Please give me 18 broken-plural words [...]. None of the words may repeat. [...]*
> Capability tag: Arabic broken plurals. Rule: `arabic_broken_plurals_each:[1,1]`.

Indonesian (ID). Regional capability tags include the articles *si* and *sang*, formal abbreviations, loanwords with initial-letter and letter-count constraints, negation particles, the *se-* quantifier prefix, passive voice, and plural forms.

> Baru-baru ini supermarket baru saya sudah dibuka, tolong bantu saya memikirkan nama toko. [. . .] semua nama toko secara keseluruhan harus mengandung 1 kata serapan bahasa Inggris, Prancis, atau Belanda dalam bahasa Indonesia yang dimulai dengan huruf K dan terdiri dari 8 huruf.
> *My new supermarket just opened, please help me name the store. [. . .] All store names together must contain 1 loanword from English, French, or Dutch in Indonesian that starts with the letter K and is 8 letters long.*
> Capability tag: Indonesian loanwords with initial-letter, letter-count, and unit-count constraints. Rule: `check_indonesian_loanwords_each:1,K,8`.