

Jais 3: Introducing Arabic-centric Agentic AI

Jais Team*

Institute of Foundation Models, Mohamed bin Zayed University of Artificial Intelligence
Cerebras Systems, Inception42

Overview

We present **Jais 3**, a fleet of six open, Arabic-centric agentic language models ranging from 0.9B to 375B parameters. **Jais 3** is built to complete tasks for Arabic speakers, not only to answer questions in plain text: it is trained to plan multi-step tasks, call tools and Model Context Protocol (MCP) servers, operate a terminal, search the web, and read, edit, and produce PDF, spreadsheet, and presentation files, while communicating with the user in Arabic. Instead of pretraining from scratch, we adapt each model of the fully open **K2 Horizon** fleet to Arabic through continual pretraining, followed by supervised fine-tuning (SFT), final token preference optimization (FTPO), and reinforcement learning with verifiable rewards (RLVR). We combine natural Arabic text with translated, transliterated, and synthetic Arabic, and replay the base model's own training data to keep its reasoning, agentic, and English abilities. **Jais 3** covers Modern Standard Arabic and the major regional dialects, follows instructions with Arabic-specific constraints, and inherits the 512K-token context window of **K2 Horizon** (all models except the 0.9B). To our knowledge, its 375B model is the largest open Arabic-centric language model to date. For evaluation, we revisit how Arabic models are assessed, moving to fully generative evaluation on audited Arabic benchmarks. Across model sizes, **Jais 3** matches or outperforms open-weight models of comparable size on Arabic and agentic benchmarks, with particularly strong agentic capabilities and Arabic instruction following, and far fewer hallucinations. We release the weights and the checkpoints from every training stage of all six models, together with samples of the training data, the data recipes and training configurations, the training logs, the audited benchmarks, and the evaluation code.

Jais 3 جيس

	Jais 3	hf.co/collections/IFM/jais-3
	Jais 3 (Web)	jaischat.ai
	Jais 3 (Training Code)	github.com/ifm-ai/xllm
	Jais 3 (RL Code)	github.com/ifm-ai/RL360
	Jais 3 (Eval Code)	github.com/ifm-ai/gen-eval-arabic
	Jais 3 (Audited Benchmarks)	hf.co/collections/IFM/arabic-benchmark-auditing
	Jais 3 (Training Logs)	W&B Logs

*Correspondence to: {haonan.li,hector.liu,eric.xing}@mbzuai.ac.ae

Contents

1	Introduction	4
2	Background and Related Work	5
2.1	Arabic-Centric and Multilingual LLMs	5
2.2	Language Adaptation and Continual Pretraining	6
2.3	K2 Horizon	7
2.4	Problems with Current Arabic Evaluation	8
3	Data Processing	9
3.1	Processing Pipeline	10
3.2	Limitations and Exclusions	13
4	Continual Pretraining	14
4.1	Main-Stage Data	14
4.2	Cooldown Data	17
4.3	Training Recipe and Runs	18
5	Post-Training	20
5.1	SFT Data	20
5.2	Supervised Fine-Tuning	30
5.3	Final Token Preference Optimization	31
5.4	Reinforcement Learning	35
5.5	Cascade versus Mixed-Task RLVR	43
6	Evaluation	49
6.1	Arabic Evaluation	50
6.2	Agentic Evaluation	61
6.3	English Evaluation	67
6.4	Factuality and Hallucination	71
6.5	Safety Evaluation	73
7	Release Notes	76
7.1	Open-Source Statement and Commitment	76
7.2	Intended Use and Limitations	76
8	Conclusion	77
A	Reward Hyperparameters	91
B	Multi-Turn Data: Additional Details	92
B.1	Anatomy of a Targeted Conversation	92
B.2	Qualification Criteria	92
B.3	Reasoning Replay and Fallbacks	92
C	Complete Evaluation Setup	94
C.1	Decoding Configurations	94
C.2	Complete Benchmark Inventory	95
C.3	Arabic Benchmark Details	97

C.4	English Benchmark Details	98
C.5	Run-to-Run Variance	100
C.6	Low-Effort Thinking Results	100
C.7	Construction of the Arabic Linguistics Benchmarks	109
C.8	Scoring, Judging, and Failure Handling	111
C.9	Impact of Benchmark Auditing on Evaluation Scores	112

1 Introduction

Arabic is used daily by more than 400 million people (United Nations, 2026), yet it remains under-served by open large language models (LLMs). Part of the reason lies in the language itself. Arabic is a family of varieties: Modern Standard Arabic (MSA) for writing and formal speech, and regional dialects (Gulf, Egyptian, Levantine, Maghrebi, and others) for everyday use, increasingly also written in Latin script as Arabizi. Natural Arabic text on the web is limited and noisy, and much of what exists is classical, religious, or news prose. Arabic-centric models such as **Jais** (Sengupta et al., 2023; Anwar et al., 2026), **AceGPT** (Huang et al., 2023), **ALLaM** (Bari et al., 2024), and **Fanar** (Fanar Team et al., 2025, 2026) have narrowed the gap. These models, like the benchmarks used to compare them, are judged mainly on whether they can answer a question in Arabic. Meanwhile, open agentic models, which plan multi-step work, call tools, and handle files, are trained and benchmarked mostly in English, and how well they serve Arabic users is rarely a design goal. As a result, open models that can complete real tasks for Arabic speakers, rather than only converse with them, are largely missing.

This report presents **Jais 3**, a fleet of six Arabic-centric agentic models that range from 0.9B to 375B parameters: four dense models (0.9B, 3.7B, 7B, 32B) and two sparse Mixture-of-Experts models: a 36B model with Mixture-of-Value Attention (36B MoVA, 4B active parameters) and a 375B model (23B active). Instead of pretraining from scratch, we build **Jais 3** on **K2 Horizon** (IFM Team, 2026), a fully open model fleet pretrained on up to about 22T tokens with extensive reasoning and agentic data. Each **Jais 3** model starts from the final, post-trained checkpoint of the **K2 Horizon** model of the same size and keeps its architecture and tokenizer unchanged (Section 2.3). We adapt it to Arabic through continual pretraining, which ends with a short cooldown, then post-train it with supervised fine-tuning (SFT), final token preference optimization (FTPO), and reinforcement learning with verifiable rewards (RLVR) (Figure 1). Our goal is to add Arabic language, dialectal, and cultural competence while keeping the reasoning, long-context, agentic, and English abilities of the base fleet. To our knowledge, **Jais 3** is the first language adaptation of **K2 Horizon**, and its 375B model is the largest open Arabic-centric language model released to date.

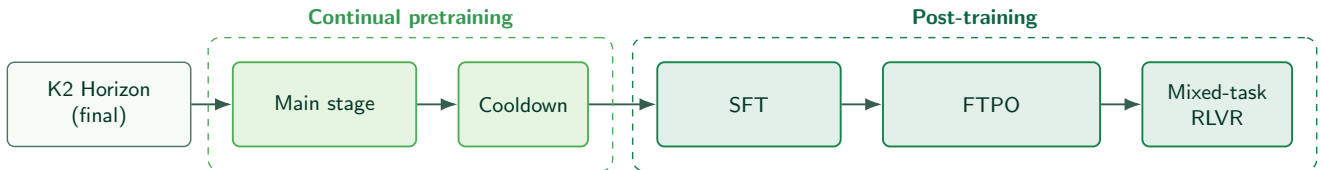


Figure 1: Training pipeline of **Jais 3**. Each model starts from the final, post-trained **K2 Horizon** checkpoint of the same size and passes through Arabic and English continual pretraining (ending with a short cooldown), supervised fine-tuning (SFT), final token preference optimization (FTPO), and mixed-task reinforcement learning with verifiable rewards (RLVR).

Contributions. The main contributions of this work are:

- **An open Arabic-centric agentic model fleet.** We release six models from 0.9B to 375B parameters, covering on-device to large-scale server deployments. All models except the 0.9B inherit the 512K-token context window of **K2 Horizon**. On Arabic, **Jais 3** leads or matches the best open models at every size, with the largest margins at small sizes, and the 375B model is competitive with **Claude Sonnet 5** (Section 6.1).
- **Agentic capabilities for Arabic users.** **Jais 3** models can plan multi-step tasks, call tools and Model Context Protocol (MCP) servers, operate a terminal, and search the web while communicating

with the user in Arabic. Connected to file tools, they can read, edit, and produce PDF, spreadsheet, and presentation files directly in Arabic (Sections 5.1.7 and 6.2). They also follow complex instructions with Arabic-specific constraints on orthography, diacritics, and morphology (Section 5.1.4).

- **A documented recipe for adapting a strong base model to a new language.** We describe the full adaptation pipeline and the decisions that mattered (Sections 4 and 5). Natural Arabic text runs out long before the token budget, so translated, transliterated, and synthetic Arabic make up a large share of continual pretraining. A fixed share of replay from the base model’s own mid-training data limits forgetting. Each data source needs its own filters, because rules tuned on English web text silently discard large amounts of valid Arabic.
- **A generative, audited Arabic evaluation suite.** Most Arabic leaderboards score multiple-choice log-probabilities, which do not fit reasoning models, and many widely used benchmarks contain wrong, ambiguous, or outdated ground truths. We move all evaluation to a generative protocol, and we audit the Arabic benchmarks in our suite with three LLM judges, removing the items that all three flag as flawed (Section 6.1.1). We also introduce new Arabic benchmarks: AraIFBench for instruction following, two linguistic benchmarks for grammatical analysis (*i’rab*) and vocalization (*tashkeel*), Arabic versions of AA-LCR and SimpleQA-Verified (Section 6.1.2), and an Arabic file-processing benchmark for agents (Section 6.2.1). We release the cleaned benchmarks, the discarded items, and the evaluation code.
- **An open release beyond the weights.** For each model we release checkpoints from every training stage: continual pretraining (including the cooldown), SFT, FTPO, and RLVR. We also release samples of the training data, the data recipes and training configurations, the training and RL code, the W&B training logs, the audited benchmarks, and the evaluation code, so that others can study, reproduce, or branch from any stage of the pipeline (Section 7).

Outline of the report. Section 2 reviews related work on Arabic and multilingual models, language adaptation, K2 Horizon, and Arabic evaluation. Section 3 describes the data processing shared across training stages. Sections 4 and 5 present continual pretraining, SFT, FTPO, and RLVR, together with their data and training recipes; the FTPO and RLVR subsections also include stage-level results. Section 6 presents the Arabic, agentic, English, factuality, and safety evaluations, including our benchmark auditing and newly introduced benchmarks. Section 7 discusses release details, intended use, and limitations, and Section 8 concludes the report. Additional details are provided in the appendices, including multi-turn data details (Appendix B), and the complete evaluation setup (Appendix C).

2 Background and Related Work

2.1 Arabic-Centric and Multilingual LLMs

Frontier open and closed models have demonstrated strong performance across many tasks, but multilingual evaluations still show gaps between English and other languages (Xuan et al., 2025; Caciolai et al., 2026). Arabic, as a medium-resource language, remains below English for several evaluated models and tasks (Xuan et al., 2025). These gaps are especially clear in Arabic dialects. Recent evaluations report lower performance in dialectal settings than in Modern Standard Arabic (MSA), including cultural reasoning, translation, and dialect-controlled generation (Al Kautsar et al., 2026). At the same time, efforts to create Arabic-centric models have progressed to address these gaps. Jais 1 introduced Arabic-centric bilingual models, with pretrained and instruction-tuned versions of up to 30B parameters. These models combine large-scale Arabic, mainly MSA, with English and code during pretraining, followed by Arabic and English instruction tuning (Sengupta et al., 2023; Inception, 2024). The Jais-adapted series adds Arabic capability to English-

focused Llama 2 models through vocabulary expansion and continual bilingual pretraining, followed by bilingual instruction tuning (Gosal et al., 2024; Inception, 2024). Jais 2 extends this line with 8B and 70B models trained from scratch. The family places greater emphasis on dialects and Arabic cultural knowledge, including poetry, cuisine, and Islamic question answering (Anwar et al., 2026).

Other Arabic-centric efforts include Fanar 2.0, which combines Arabic model adaptation with multimodal and tool-use support (Fanar Team et al., 2025, 2026); AceGPT, which focuses on Arabic adaptation and cultural alignment (Huang et al., 2023); and ALLaM, which combines vocabulary expansion, bilingual training, and preference alignment (Bari et al., 2024). Atlas-Chat targets Moroccan Darija written in Arabic script (Shang et al., 2025b), whereas Nile-Chat targets Egyptian Arabic in both Arabic and Latin scripts (Shang et al., 2025a). Falcon-H1-Arabic uses a hybrid Mamba-Transformer architecture (Boussaha et al., 2026), and Command R7B Arabic focuses on retrieval-augmented generation and contextual faithfulness (Alnumay et al., 2025).

BLOOM was one of the early open efforts in multilingual language modeling, with pretraining across 46 natural languages, including Arabic (BigScience Workshop et al., 2022). BLOOMZ followed with multilingual instruction tuning to improve generalization across tasks and languages (Muennighoff et al., 2023). Later releases include Llama 3 (Grattafiori et al., 2024), Gemma 3, with reported support for over 140 languages (Gemma Team et al., 2025), and Gemma 4 (Gemma Team, 2026). The Qwen family expanded support from 119 languages and dialects in Qwen3 to 201 in Qwen3.5 (Yang et al., 2025; Qwen Team, 2026a), followed by Qwen3.8 (Qwen Team, 2026c). However, broad language coverage does not ensure equal performance. On MMLU-ProX, Arabic accuracy trails English by 14.7, 7.8, and 6.9 percentage points for Llama-3.3-70B, Gemma-3-27B, and Qwen3-32B, respectively (Xuan et al., 2025). Evaluations of earlier models from these families also show gaps between Modern Standard Arabic and Arabic dialects (Al Kautsar et al., 2026).

Gains in Arabic knowledge and generation do not by themselves establish reliable Arabic tool calling or multi-step agent performance. A recent Arabic adaptation of BFCL reports lower tool-calling accuracy with Arabic prompts for Llama 3.3, Qwen3, and gpt-oss (Kubrak et al., 2026). Telco-GAIA extends evaluation to Arabic and English tool-using agents, and finds remaining weaknesses in tasks involving documents and images (Khizbullin et al., 2026). However, these studies cover limited sets of models and tasks, and neither reports a systematic evaluation of tool use across Arabic dialects (Kubrak et al., 2026; Khizbullin et al., 2026). Jais 3 addresses both gaps: it is an Arabic-centric agentic model trained for multi-step tool use in Arabic and bilingual settings.

2.2 Language Adaptation and Continual Pretraining

Continual pretraining (CPT) adapts an existing language model by continuing its language-modeling objective on a new data distribution, typically next-token prediction on target-language text (Gururangan et al., 2020). It has become a practical way to bring open foundation models to new languages without pretraining from scratch. For Japanese, Swallow combines vocabulary expansion with large-scale CPT and reports continued improvements on Japanese benchmarks up to 100 billion training tokens, alongside decreases on English benchmarks (Fujii et al., 2024). For Arabic, Gosal et al. (2024) combine vocabulary expansion, an embedding-alignment stage, and full-model bilingual CPT, obtaining substantial Arabic gains while preserving or improving aggregate English performance. Both works expand the vocabulary because the inherited tokenizer segments the target language poorly, which requires initializing new embeddings (Minixhofer et al., 2022; Dobler and de Melo, 2023). Jais 3 does not need this step, because the K2 Horizon tokenizer already covers Arabic well (Section 2.3). We also train all parameters, rather than restricting updates to adapters or low-rank modules (Pfeiffer et al., 2020; Hu et al., 2022).

A central challenge in CPT is balancing the acquisition of new capabilities with the retention of existing ones. Mixing target-language text with source-language or general-domain data is a common mitigation,

and its benefits extend beyond source-language perplexity: [Elhady et al. \(2025\)](#) show that excluding English during language adaptation can impair in-context learning and downstream target-language performance despite similar target-language validation perplexity. Optimization interacts with these data choices: [Ibrahim et al. \(2024\)](#) show that learning-rate rewarming and decay combined with replay are effective under distribution shift, and the CMR scaling law models the relationship between mixture proportions, training duration, and general-domain versus target-domain loss ([Gu et al., 2024](#)). Following this line of work, **Jais 3** replays a fixed share of **K2 Horizon**’s own mid-training data in every stage and ends continual pretraining with a short cooldown (Section 4).

Most prior adaptations start from a base model and add instruction tuning afterward. Continuing to train an already post-trained model carries a further risk: fine-tuning can erode a model’s general and instruction-following abilities ([Luo et al., 2023](#)), and even benign fine-tuning can weaken safety alignment ([Qi et al., 2023](#)). A second common ingredient for under-resourced languages is machine-translated data, used both in pretraining ([Sengupta et al., 2023](#)) and in instruction tuning ([Lai et al., 2023](#); [Üstün et al., 2024](#)). **Jais 3** combines these settings. It applies full-model CPT to a post-trained, agentic **K2 Horizon** checkpoint with an unchanged tokenizer, mixes natural Arabic with translated, transliterated, and synthetic Arabic, and replays reasoning and agentic traces so that it keeps these capabilities, not only its English language modeling.

2.3 K2 Horizon

K2 Horizon ([IFM Team, 2026](#)) is a fully open fleet of six language models developed at the Institute of Foundation Models (IFM) and released under the Apache 2.0 license. The models use a standard Transformer backbone with grouped-query attention, rotary position embeddings, and SwiGLU feed-forward layers, and add components of their own: grouped RMSNorm across the fleet and, in the 36B model, Mixture-of-Value Attention (MoVA) with a softplus-activated attention gate. The fleet has four dense models (0.9B, 3.7B, 7B, and 32B) and two sparse ones (Table 1). The 375B model is a fine-grained Mixture-of-Experts (MoE) model with 192 routed experts and one shared expert per layer, of which eight routed experts are active per token. The 36B MoVA model combines MoE feed-forward layers with MoVA, which applies expert routing to the value projections in attention: each token selects 4 of 64 value experts, at no additional KV-cache cost. All models share a 250,624-token vocabulary except the 0.9B, which uses a 64,256-token vocabulary to save memory on device.

	0.9B	3.7B	7B	32B	36B MoVA	375B
Architecture	Dense	Dense	Dense	Dense	MoE + MoVA	MoE
Active backbone params (B)	0.88	3.78	6.95	32.21	4.67	23.59
Total backbone params (B)	0.88	3.78	6.95	32.21	36.16	376.09
Layers	28	36	36	64	48	61
Hidden dimension	1,536	2,560	4,096	5,120	2,560	6,144
Query / KV heads	32 / 8	32 / 8	32 / 8	64 / 8	32 / 8	48 / 8
Vocabulary size	64,256	250,624	250,624	250,624	250,624	250,624
Pretraining tokens (T)	5.0	21.9	21.9	21.9	21.9	15.1
Context after mid-training	128K	512K	512K	512K	512K	512K

Table 1: **K2 Horizon** configurations ([IFM Team, 2026](#)), the starting point of **Jais 3**. Backbone parameter counts exclude the input embedding and output projection. The 375B model was pretrained in two phases (7.05T and 8.05T tokens).

Training. The models are pretrained with an 8K-token context on about 22T tokens ([IFM Team, 2026](#)) (15T for the 375B model and 5T for the 0.9B model). The pretraining mixture combines web, code,

mathematics, science, multilingual, and domain-specific data with about 10T synthetic tokens, and nearly 17% of it consists of problem-solving trajectories with explicit reasoning. Mid-training then extends the context in stages from 32K to 128K and 512K tokens (1.1T, 500B, 110B, and 200B tokens over four stages), while adding reasoning and agentic data; the 0.9B model follows a shorter two-stage schedule that ends at 128K. Post-training trains domain-specific RL specialists, merges them into one policy, and adds supervised fine-tuning, with a short final token preference optimization (FTPO) stage on the small models to remove degenerate repetition. The release includes final and intermediate checkpoints, training code, data or data recipes, and training logs.

Arabic in K2 Horizon. Arabic is already well represented in K2 Horizon (IFM Team, 2026). Multilingual text makes up about 7% of the pretraining mixture across 17 languages, and Arabic is the second-largest of them. The tokenizer was also trained on a corpus that is roughly a quarter Arabic, to give Arabic text an efficient segmentation. Even so, Arabic is not a target language of K2 Horizon, and adapting it to Arabic pays off: Jais 3 raises the Arabic average by 9 to 35 points over the K2 Horizon model it starts from.

From K2 Horizon to Jais 3. Each Jais 3 model starts from the final, post-trained checkpoint of the K2 Horizon model of the same size, with its architecture and tokenizer unchanged. Jais 3 therefore inherits the long-context, reasoning, and agentic abilities of its base, including the 512K-token context window of K2 Horizon (all models except the 0.9B), and the adaptation described in Sections 4 and 5 aims to add Arabic competence without losing them. Because adaptation starts from a post-trained model rather than a pretrained one, continual pretraining must also preserve instruction-following and agentic behavior. To this end, we replay K2 Horizon’s own mid-training data, which includes reasoning traces and agentic trajectories, in continual pretraining, cooldown, and SFT (Sections 4.1 and 5.2). To our knowledge, Jais 3 is the first adaptation of K2 Horizon to a new language, and since every part of the starting point is public, the adaptation can be repeated or extended by others.

2.4 Problems with Current Arabic Evaluation

Most Arabic leaderboards, such as the Open Arabic LLM Leaderboard (El Filali et al., 2024), score models by the log-probability of multiple-choice options, which does not reflect how modern reasoning models are used. Such models respond freely: they may state the answer first and justify it afterward, or reason at length before concluding, in plain text or Markdown. Instructions to answer in a fixed format are not reliably followed, so rigid answer extraction can mark correct responses as wrong. We therefore convert all benchmarks to generative evaluations and use an LLM judge to identify and grade the final answer, using the original benchmark gold label as a reference, regardless of its position or format.

Several widely used benchmarks also contain defective items that penalize correct answers. For instance, an ArabicMMLU (Koto et al., 2024) item keys 15 as the number of NATO members, although the alliance now has 32; an Arabic GSM8K (Omartificial-Intelligence-Space, 2025) reference solution miscounts the coins in its own question; and an ArabCulture (Sadallah et al., 2025) item offers three equally plausible options but keys only one as correct. We filter such items with a three-judge LLM audit, which removes 5.3% of the items in the 15 benchmark sets we use (Section 6.1.1).

3 Data Processing

Key Takeaways

- **Plan for the natural-text ceiling.** Natural Arabic web text and literature supply about 19% of the continual-pretraining mixture; translated and synthetic Arabic supply about 38% of the run, of which about 16 points is Arabizi (Latin script) and about 22 points Arabic script. The parallel dialect text is an LLM translation of English conversations, and the English side of those conversations adds a further 6 points.
- **Replay runs through every stage.** Data from the base model’s own mid-training (web, reasoning, and agentic traces) makes up 25% of continual pretraining and SFT and is part of every cooldown mixture, where it is drawn from the most recent mid-training phase of the replay data used in continual pretraining.
- **Every source needs its own filters and quality checks.** One shared rule set removed 12% of a dialect corpus and 34% of a diacritized one; per-source rules kept over 99% of both.

Jais 3 is trained by continuing the pretraining of **K2 Horizon** (Section 2.3), so the data work has a different goal from building a corpus from scratch: add Arabic knowledge, dialects, and assistant behavior without eroding what the base model already does well. This shapes every decision in this section. Arabic data is selected and weighted for what the base model lacks, and a share of the base model’s own mid-training data (web text, reasoning traces, and agentic trajectories) is replayed in each stage. **Jais 3** builds on the Arabic data work of **Jais 2** (Anwar et al., 2026). For this release we name each source, report tokens under one tokenizer, and state what was filtered, deduplicated, and decontaminated, including the cases where a step was not run.

The inventory we draw from holds approximately 1.63 T tokens, counted with the **Jais 3** tokenizer, which is unchanged from **K2 Horizon**: 1.41 T of pretraining-style text across 39 datasets and 224 B of chat-formatted data across 35 datasets. These are sizes of what we prepared, not what the model saw; the training mixtures sample from the inventory with per-source weights, and we report inventory and exposure separately throughout. The Arabic side combines **ArabicTxT360** and style rewrites of it, Arabic translations of the base model’s reasoning data, a synthetic Arabizi corpus, parallel dialect text, and small diacritization and search-trajectory sets. Coverage therefore spans Modern Standard Arabic, regional dialects, Arabizi, and diacritized text.

The data description is organized by training stage (Table 2). Continual pretraining samples 201.5 B tokens, with each source weighted for about one pass, 75% Arabic-side and 25% replay. Cooldown combines small, high-quality sets (chat-formatted cooldown data, monthly Arabic Wikipedia, and knowledge-infusion and explanation-enriched sets; Section 4.2 describes their decontamination and its limits) with mid-training replay; it samples 3.47 B tokens (4.0 B with the 0.9B’s tokenizer) in a three-to-one ratio. SFT samples about 41 B tokens: Arabic and English instruction data in high- and low-thinking modes, long-context Arabic question answering (about 17% of the mixture), agentic trajectories, most of them Arabic translations (about 14%), dialect and Arabizi conversations, culturally grounded sets (Islamic QA, inheritance law, poetry, UAE and country knowledge), safety data, and 25% replay. Data for reinforcement learning with verifiable rewards (RLVR) is described last.

Section 3.1 describes the processing shared across sources: extraction to one document per record, quality filtering, MinHash deduplication within and across sources, screening against evaluation sets, and token accounting. The data used in each training stage is described with that stage: continual pretraining and cooldown data in Sections 4.1 and 4.2, and SFT, FTPO, and RLVR data in Section 5. Section 3.2 lists limitations.

Stage	Used in training	Mixture	Section
Continual pretraining	201.5 B tokens	151.1 B Arabic-side tokens from 32 sources, one pass (75%); 50.4 B mid-training replay (25%)	§4.1
Cooldown	3.47 B tokens	chat-format data, Arabic Wikipedia, knowledge and task QA sets (75%); mid-training replay (25%)	§4.2
SFT	40.8 B tokens per epoch, 22.2 M examples	138 mixture entries, one epoch; includes mid-training replay (25%)	§5.1
RLVR	19,769 prompts	five verifiable tasks: instruction following, grammatical analysis, mathematics, diacritization, code (plus about 10.9 K instruction-following prompts for the 375B)	§5.4.1

Table 2: Data used in each training stage. Token counts are under the Jais 3 tokenizer; shares are fractions of tokens sampled in the stage.

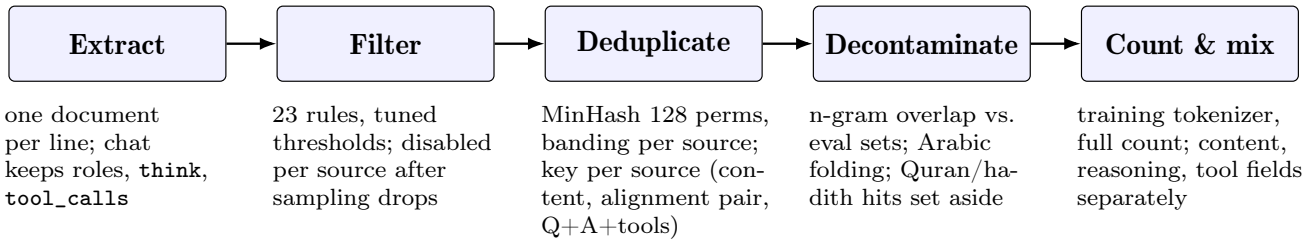


Figure 2: Processing pipeline. The five stages are shared by most sources; the settings under each stage are chosen per source.

3.1 Processing Pipeline

Most sources pass through the same five stages: extraction, quality filtering, deduplication, decontamination, and token accounting (Figure 2), with configurations tailored to each source. These configurations account for differences in linguistic content and corpus structure: boilerplate removal heuristics can exclude valid diacritized classical prose, while deduplication criteria can collapse parallel dialect sentences whose high similarity is intentional. We therefore treat the pipeline as a fixed set of tools and a per-source profile, and we accept a profile only after inspecting samples of what it keeps and what it removes. The rest of this section describes the tools; the figures quoted are the cases that forced a profile change.

Extraction. Each source is converted to JSONL with one self-contained document per line, `{id, text, metadata}`, so that later stages can hash, filter, and count documents independently of the vendor’s packaging. Chat and agentic records keep their structure (`messages` with roles, and `think` and `tool_calls` fields where present); for document-oriented tools we project them to text by joining the turn contents, and the projection never feeds `think`, `tool_calls`, or tool schemas into that text. Generation and annotation metadata, including the evaluation questions used to seed synthetic data, stay outside `text`.

Quality filtering. The Arabic filter applies 23 document-level rules: length (word count, mean word length), composition (alphabetic fraction, Arabic-character ratio, symbol-to-word ratio, stop-word count against 55 Arabic function words), repetition (duplicate lines, paragraphs, and 2- to 10-gram fractions), boilerplate (navigation phrases, booking listings, bullet and ellipsis lines, short-line tables, lorem ipsum), and content (an adult-URL list and a clitic-aware bad-word list). Measurements are taken on a normalized copy, with diacritics stripped and clitics such as *wa-* separated, while the emitted text is unchanged.

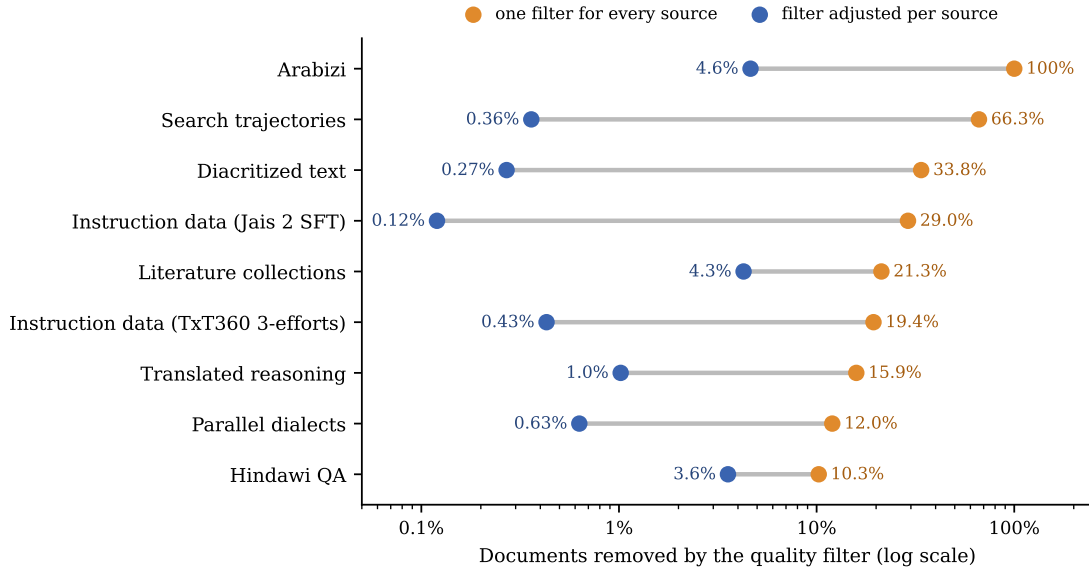


Figure 3: Documents removed by the quality filter when one filter is applied to every source (orange) and when the filter is adjusted per source (blue), for the nine sources whose audits measured both. Log scale. In each audit, samples of the documents the one-size filter removed were read; they were overwhelmingly valid text (classical passages, dictionaries, Latin-script Arabic, reasoning traces), which is why the adjusted profiles were accepted.

Defaults were calibrated on high-quality Arabic text rather than imported from English pipelines: lowering the mean-word-length floor from 3.5 to 3.2 characters reduced rejection of classical and religious prose from 19.7% to zero; lowering the alphabetic-fraction floor from 0.80 to 0.65 reduced rejection from 54.4% to 9.6% on the same subset. Inspection of rejected documents revealed three defects. The undiacritized stop-word list failed on fully vocalized text, rejecting 56% of a hadith collection. The uppercase check considered only cased characters, so acronyms such as *CNN* or *OPEC* triggered rejection of Arabic paragraphs, removing 22.4% of one news source. The bad-word list matched inside ordinary words after clitic expansion (for example *zabūn*, customer), and correcting it recovered 76.4% of the dialect documents it had removed.

Each source then gets its own profile (Figure 3). The diacritization corpus lost 33.8% of its documents to the sentence-count rule alone, because classical passages are written on one line without terminators, and its final profile removes 0.27%. The parallel-dialect corpus lost 12% at defaults, e.g., records on drug information or scientific abstracts whose repetition is structural, and retained 99.2% after nine rules were disabled. For search-trajectory data, the Arabic-ratio rule would have rejected 66.3% of documents because of their English reasoning traces; the adjusted profile instead adds a stuck tool-call loop check, which removes 305 trajectories (0.36%).

Deduplication. We run MinHash (Broder, 1997) locality-sensitive hashing with 128 permutations over 13-character shingles on lowercased, punctuation-stripped text, banded per source: 9 bands of 13 rows for most sources (candidate pairs at Jaccard ≈ 0.85), and 16 bands of 8 rows (≈ 0.71) for the style rewrites and Arabizi. Candidate pairs are grouped into connected components and one representative per component is kept. The comparison key is chosen per source. For search-trajectory data the default content key would have removed 2,177 records with a 99.95% false-positive rate, because retrieved Wikipedia passages dominate the hash even when the question, reasoning, and answer differ; keying on question, reasoning, answer, and tool calls, and excluding retrieved text, flagged two records, one of them a false positive, so one record was removed. For parallel dialect text, each record holds an English conversation together with

Scan	Coverage	Removed	Rate
Continual pretraining	13 datasets, 149.4 M lines	15,689	0.0105%
Cooldown	11 datasets, 4.3 M lines	1,952	0.045%
SFT	33 datasets, 33.1 M samples	3,272	0.0099%

Table 3: Contamination screening of the three training stages against the evaluation benchmarks. Counts are lines, one JSONL record each, not tokens; line length varies roughly 130-fold across datasets, from 2,784 bytes to 362,075 per line, so rates are not comparable across rows.

its dialect translation, and we hash the whole record rather than either side, so two translations of the same conversation are not collapsed into one. We do not deduplicate questions alone in synthetic QA sets, where one question with several distinct answers is intended.

Decontamination. The training sets we prepared for continual pretraining, cooldown, and SFT (Table 3) are screened against our evaluation benchmarks before use with AllenAI’s `decon` n-gram overlap detector¹ (Rust, with pyo3 bindings) wrapped in a DataTrove (Penedo et al., 2024) pipeline,² after folding Arabic orthographic variants (alef and ya forms, ta marbuta, tatweel, diacritics) on both the evaluation index and the scanned text. Overlap is not the same as leakage: Quranic verses, hadith and shared instruction boilerplate appear in thousands of documents and in the benchmarks alike, and the single largest such item matched 11,485 separate documents. These are excluded on review before anything is removed. Table 3 shows information after this exclusion. All replay was screened: the replay used in SFT is drawn from the same K2 Horizon mid-training data as the 50.4 B continual-pretraining replay, so the continual-pretraining scan covers both.

In the continual-pretraining scan, 111,089 lines matched: 15,689 matched at least one discriminative item and were removed from the training copies, and the other 95,400 matched only non-discriminative text; the rewrite touched 3,447 files, whose cleaned versions were verified byte for byte against the originals. The overlap concentrates in poetry benchmarks with none in the diacritization or search-context sets. The cooldown scan ran against 554 of the 567 delivered evaluation files, covering 210,269 questions indexed as 287,416 entries after deduplication; of 1,955 flagged lines, 3 were kept after individual review as false positives and 1,952 were removed, and about half of the Arabic Wikipedia hits were canonical religious text quoted by both the article and the evaluation item. The SFT scan removed 3,272 samples across 33 datasets, with the poetry set highest at 1.45%, and every scanned file is written through byte-identical to its source apart from the removed records. A re-scan of the cleaned cooldown release flags only the three kept false positives and nothing else. Screening is lexical throughout and operates on whole records, so a flagged record is dropped in full.

Token accounting. After processing we count every document with the training tokenizer, without subsampling. We recompute all counts, as source-provided values differed from audited counts by up to a factor of eight; in one case, undecoded binary files were assigned token counts equal to their character lengths. For chat data we count assistant turns as the number of assistant messages and tokenize content,

¹<https://github.com/allenai/decon>

²Token 8-grams under the `Jais 3` tokenizer, testing an n-gram every 3 tokens, with hits expanded to the full matching span and scored as an IDF-weighted overlap of question, answer and passage; anything at or above 0.8 is reported, the upstream default, unchanged. The stride of 3 is the largest with no blind spot, since a region is certain to be found when $m \leq \ell_{\min} - n + 1$, where m is the stride, n the n-gram length, and $\ell_{\min}$ the length in tokens of the shortest indexed evaluation span; it leaves 0.00% of indexed evaluation entries at risk against 7.61% at the tool’s default of 10. Using the model’s own tokenizer matters here: `cl100k` needs 4.17 tokens per Arabic word against 1.86 for the `Jais 3` tokenizer, so an 8-token n-gram would span under two Arabic words.

reasoning, and tool-call fields separately: a content-only count missed 28% of the tokens of the translated agentic data and 631 M reasoning tokens of one Arabic instruction set. These are inventory counts; exposure depends on mixture weights and repetition and is reported per stage.

Chat-format data. Conversational data, the SFT pool and the chat-formatted cooldown set, is stored as `messages[]` with no `text` field, so the document filter above drops all of it, and it would in any case reject turns that are correct for chat: a bare option letter as the answer, or English reasoning on an Arabic prompt. These sets go through a chat-aware pipeline instead, with five checks:

- *Format gate.* Every assistant turn must carry exactly one of `think` or `think_faster`. A stratified sample is read before any detector is trusted.
- *Deduplication.* Runs over six keys (byte-identical record, conversation after Unicode and diacritic normalization, first user prompt, final answer, the generator’s seed hash, and MinHash near-duplicate prompts), with exact families reduced to one record per cluster and near-duplicate clusters judged as complete records.
- *Content.* A role-aware bad-word pass flags assistant turns for a judge and a human read.
- *Identity.* Scans assistant content and reasoning for another model’s name next to a first-person cue.
- *Consistency.* Every reasoning turn with a delivered answer is judged for agreement between the end of its trace and that answer.

Within the SFT and chat-format cooldown sets, flags are candidates, never removals: each drop is decided after reading the whole record, refusals and educational answers to sensitive prompts are kept, and a prompt that recurs with different answers is kept in full.

On the chat cooldown set this removed 1,233 of 1,844,798 conversations (0.067%): 453 exact duplicates, 154 explicit completions out of 348 candidates read, 126 reasoning–answer contradictions, 68 records with the reasoning trace serialized into the answer, 430 tutor and multiple-choice dialogues that were wrong on reading, and 2 overlong failed traces; 69,225 system prompts naming the base model were rewritten to the `Jais 3` identity. The reasoning–answer judge ran over 10.4 M reasoning turns of the SFT pool and flagged 2.5%. Most flags, 87%, sat in the two Arabic and Arabizi instruction sets, which were republished without 3.55% and 2.40% of their rows (contradictions, substituted personal names, output-language and keyword-constraint failures, and 14,738 Arabizi answers with Arabic letters fused inside words); 19,193 flagged records were removed from 28 further sets before decontamination. The agentic sets, which hold about 14,000 of the consistency flags, were not filtered for consistency.

Identity leakage concentrates in synthetic data and inside reasoning traces. The mid-training chat replay was handled differently: records were removed on the name match alone, so all 789,812 of 13.7 M records (5.75%) that carried another model’s name in an assistant turn were dropped, and a judge confirmed 27,120 of them as self-identification, 84% inside the reasoning trace and 81% naming ChatGPT. A regenerated TxT360 (Tang et al., 2024) delivery carried the teacher’s name in 1,554 traces and the generation prompt’s own instructions in 23,440, and was rebuilt without 2,883 rows.

3.2 Limitations and Exclusions

Screening is lexical. The contamination scan matches token n-grams after orthographic folding, so translated or paraphrased benchmark items are not detected, and MinHash over character shingles measures lexical overlap only, so paraphrased duplicates survive deduplication.

4 Continual Pretraining

Key Takeaways

- **One main-stage dataset for every size.** Each model starts from its post-trained K2 Horizon checkpoint and trains on the same main-stage data (201.5 B tokens; 233.6 B with the 0.9B’s tokenizer), 75% Arabic-side data and 25% replay, then a short cooldown; batch size, learning rate, and weight decay change with size, and the 0.9B also differs in tokenizer and position encoding.
- **Constant learning rate, then an annealing cooldown.** The main stage keeps the learning rate flat, so its last checkpoint is not annealed; the cooldown decays it to a tenth of the peak on curated data, much of it in chat format, that is closer in shape to SFT.
- **Benchmark-derived cooldown data uses training splits only.** One small set adds written explanations to the training splits of HellaSwag, ARC, and DROP (Table 6).

Continual pretraining adapts each K2 Horizon model to Arabic in two stages: a main stage and a cooldown. Sections 4.1 and 4.2 describe the data of each stage, and Section 4.3 the training recipe and the runs.

4.1 Main-Stage Data

The main stage samples 201.5 B tokens, with each Arabic source weighted for one pass over its retained tokens. On the Arabic side the mixture has 32 entries: the 19 subsets of the translated reasoning data, each weighted separately, and the 13 other sources of Table 4, counting the targeted and instruction rows as their individual sets. A quarter is replay from K2 Horizon’s own mid-training data, and three quarters come from the sources in Table 4. Natural Arabic web text and literature make up 19% of the run. Translated and synthetic sources make up 44%, of which about 6 points is the English side of the parallel dialect data. The remaining 12% is small targeted sets, Arabic instruction data, English domain data, and further Arabic texts. Every source went through the pipeline of Section 3.1 with its own filter settings; Table 5 records the settings and removal rates for each one.

Natural Arabic. ArabicTxT360 is our Arabic extraction of 83 web crawl snapshots, refiltered with the Arabic rule set; older crawls were markedly noisier, with 25–44% of documents removed from the 2014–2017 snapshots against about 14% from 2023–2024. The literature collections include, among others, classical texts, poetry, news, and religious scholarship with the longest record being 16.3 M tokens long.

Translated and synthetic Arabic. This is the largest group, because natural Arabic runs out long before the token budget. We translated 19 subsets of K2 Horizon’s pretraining data (reasoning, planning, question answering, instruction following, and code reasoning). A pilot showed that long mathematical derivations were often truncated in translation, so we cut mathematics from 31% to about 14% of the tokens planned for translation. A human rating covered only 50 short samples, which averaged 4.95 out of 5.

The Arabizi corpus transliterates 30 B tokens of English TxT360 into Gulf, Egyptian, Levantine, and Darija Arabizi (about 27.5 B Arabizi tokens before our filtering and deduplication, 25.1 B after). For the parallel dialect corpus, GLM-5 translates English conversations into the target dialect and script, and Jais 2 writes a second version. An LLM judge scores both versions from 1 to 4 on dialect accuracy, meaning, and naturalness, and an automated check flags unwanted switches into other languages. To build fluency in both scripts, a subset was also translated from the English source into Arabizi. The style rewrites turn web passages into formal, humorous, and simplified versions with Qwen to add register variety; the humorous style is about 88% dialectal. Finally, QA augmentation appends generated question-answer pairs to web and Hindawi passages. We did not deduplicate these against the original web text, because the passage is meant to appear again in question-answer form; only 17.5% of the web QA tokens are new.

Group	Source	Origin	Tokens	Share
Natural Arabic	ArabicTxT360	83 web crawl snapshots, 2013–2024	28.4 B	14.1%
	Literature collections	Classical texts, poetry, news	9.65 B	4.8%
Translated and synthetic	Translated reasoning	19 subsets of K2 Horizon’s pretraining data	30.9 B	15.3%
	Parallel dialects	English conversations paired with dialect translations from GLM-5 and Jais 2, judge-filtered for dialect fidelity, meaning and naturalness	26.9 B	13.4%
	Arabizi	English TxT360 rendered as Arabizi in four dialect groups: Gulf, Egyptian, Levantine and Darija	25.1 B	12.5% [‡]
	Style rewrites	ArabicTxT360 passages rewritten in three styles	4.43 B	2.2%
	QA augmentation	Web and Hindawi passages with generated questions and answers	1.42 B	0.7%
Targeted	Diacritization, search, exams	Sadeed/Tashkeela diacritization, and Arabic exam Q&A	0.58 B	0.3%
Instruction data	Arabic chat data	TxT360 instruction data at three reasoning-effort levels (Arabic); Jais 2 SFT with regenerated answers	5.5 B	2.7%
English domain	Investment and economy	Finance and economics documents	6.33 B	3.1%
Other	Further Arabic documents	Partner-supplied Arabic books, cross-document deduplicated and de-watermarked	11.9 B	5.9%
Replay	Mid-training replay	K2 Horizon mid-training data	50.4 B	25.0%

Table 4: Main-stage sources. Tokens are the inventory after filtering and deduplication, counted with the training tokenizer; share is the source’s fraction of the 201.5 B tokens sampled in the main stage. [‡]A further ~3.5 points of Arabizi is inside the parallel-dialect data.

Targeted sets. Three small sets target specific skills. The diacritization set is instruction-formatted over fully vocalized classical and modern text, with loss on the answer only. The search-trajectory set records an agent that searches a bilingual Wikipedia index to answer factual questions the model had previously answered wrong. Only 17% of its tokens are written by the model; the rest are retrieved passages. The third set holds questions and answers from exam papers of 13 Arab countries extracted with Gemma 4, and judged faithful to the source in 94.6–99.9% of cases.

Instruction data and replay. About 3% of the run is Arabic instruction data in chat format, drawn from the same sets used later in SFT. Because each model starts from a post-trained checkpoint, this data keeps its chat behavior exercised during document-style training. The replay quarter is K2 Horizon’s own mid-training data, which holds English web text, reasoning traces, and agentic trajectories; it is there to limit forgetting of English, code, and reasoning while the model takes in new Arabic text. Before using it we removed every record in which the assistant might identify as another model: name matching flagged 790 K records, a judge confirmed 27,120 of them as real leaks (84% inside the reasoning trace), and we

Source	Filter settings (changes from Section 3.1 defaults)	Removed by filter (default settings in brackets)	Removed by dedup	Final docs
ArabicTxT360	nothing off; added page-template boilerplate rules; three passes over the Section 3.1 defaults	24.6% of documents, 33.5% of tokens	upstream, global	73.8 M
Literature collections	word cap off; stop-word rule matched on undiacritized text	4.3% (21.3%)	23.5%	9.03 M
Further Arabic documents	no changes	0.0% (0.9% of characters)	20.6%	199 K
Investment and economy	Arabic-ratio rule off; added binary-content rule; English thresholds	56.9% of web pages; 3.1% of words (64.7%)	53.6%	4.37 M
Translated reasoning	line cleaning off outside web text; six per-subset profiles	1.0% (15.9%)	0.04%	17.6 M
Parallel dialects	nine repetition and content rules off; bad-word list corrected	0.6% (12%)	0.2%	8.73 M
Arabizi	Arabic-ratio, stop-word and alphabetic-fraction rules off	4.6% (100%)	0.08%	11.8 M
Style rewrites	nothing off; added rules for model refusals, data dumps and script splices	12.6%	0.005%	12.4 M
Web QA	bullet rule off; checks scoped to the passage only	10.7%	0.2%	2.97 M
Hindawi QA	bullet and toxic-word rules off; one-sentence minimum	3.6% (10.3%)	5.3%	462 K
Exams	line cleaning, ellipsis, bullet and short-table rules off	38.7%	4.0%	24 K
Diacritization	sentence-count and eight other rules off	0.27% (33.8%)	0.08%	925 K
Search trajectories	default rules relaxed for English reasoning traces; added a stuck tool-call loop check	0.36% (66.3%)	1 record	85 K
Instruction (TxT360)	checks off for reasoning text; thresholds per category and effort level	0.4% (19.4%)	0.7%	2.82 M
Instruction (Jais 2 SFT)	nothing off; added a judge-based semantic filter; thresholds per task family	0.1% (29.0%), then 21% by judge	1.1%	1.81 M
Mid-training replay	identity-leak removal	5.8%	not run	12.9 M

Table 5: Processing record per main-stage source. Percentages are of documents entering each stage unless noted. The bracketed figure is what the default settings removed on the same input; it is omitted where no default run was measured. One step outside the quality filter is not listed: 16 documents carrying personal data were dropped from Further Arabic documents.

removed all 790 K.

Mixture. Every Arabic source is weighted for exactly one pass over its retained tokens. The QA-augmentation sources are converted to one chat per question–answer pair, so a passage with several questions appears once per question. The only data seen more than once is replay, where the data of the second phase of the final mid-training stage is repeated about twice. The 75/25 split between new data and replay is the same in continual pretraining and SFT; the cooldown mixture is described in Section 4.2. The English domain set came to us inside an Arabic crawl; after filtering it is 99% English.

4.2 Cooldown Data

The cooldown mixes curated cooldown sets with replay from the most recent mid-training phase; Table 6 gives the composition used for every model, three quarters cooldown sets and a quarter replay, 3.47 B tokens (4.0 B with the 0.9B’s tokenizer). The cooldown sets are small and mostly curated or synthetic: chat-format instruction data, monthly Arabic Wikipedia, English QA, and multiple-choice questions with written explanations. Table 7 records the processing of each set.

Source	Content	Language	Tokens	Share
Chat-format cooldown	instruction conversations, half with reasoning traces	Ar, En	1.64 B	47.4%
Arabic Wikipedia	monthly snapshot, redirects and disambiguation pages removed	Ar	0.61 B	17.5%
Knowledge infusion	synthetic texts and questions on 47 subjects at four levels	En	0.15 B	4.3%
Synthetic multiple choice	new questions with explanations, modeled on Arabic school-exam and general-knowledge benchmarks and on MMLU	Ar, En	0.17 B	5.0%
Benchmark training items	training splits of HellaSwag, ARC, and DROP with written explanations	Ar, En	0.03 B	0.8%
Mid-training replay	most recent phase of the main-stage replay data	En	0.87 B	25.0%

Table 6: Cooldown sources. Tokens after processing, counted with the training tokenizer; share is the fraction of the 3.47 B tokens sampled in the stage.

Chat-format data. The largest cooldown set is 1.8 M instruction conversations, split almost evenly between Arabic and English, half of them with a reasoning trace. More than half are multiple-choice questions. It went through the chat-aware checks used for SFT data instead of the document filter; these checks removed 0.07% of conversations: exact duplicates, overlong records, conversations with inconsistent reasoning and answers, and explicit content. Where the same prompt appeared with different answers we kept all of them.

Arabic Wikipedia. Wikipedia is where the web-text tools failed most visibly. The stock filter would have removed 38% of articles; reading samples of what it removed showed that only about 0.2% of the removed tokens were actually junk. The rest were valid articles tripped by rules that assume web boilerplate: repeated category lines, Latin citation titles read as shouting, and a rule meant for JavaScript notices that matched the Arabic word for JavaScript. Deduplication failed the same way: the stock MinHash run removed 63,485 articles, because similar stub articles chained together (one component linked 12,706 articles about Moroccan villages), while a pairwise check found 6 true duplicates. We therefore applied neither the filter nor MinHash, and removed only redirects, disambiguation pages, empty shells, and those 6 duplicates, 5.8% of records and 0.8% of tokens.

Knowledge infusion. This English set pairs synthetic text sections with generated questions and answers, on 47 subjects at four levels from high school to researcher. A three-pass audit with a large judge model estimated that 18.6% of answers were wrong, rising to about 75% for researcher-level abstract algebra and

physics against about 10% for high-school humanities. We repaired 22,577 answers, dropped 23,667, and estimate the remaining error rate at no less than 9%.

Synthetic multiple choice. These sets contain new multiple-choice questions with worked explanations, generated from Arabic materials and the training sets of general-knowledge benchmarks in variants written for a generic reader and for a student who chose a wrong answer. We found that generation makes questions easier and leaves a shortcut. The correct option is the longest one in 38–57% of items, against about 25% by chance; in the source Arabic benchmark it is the longest less often than chance. Generated questions are also shorter and more definitional than their sources. We removed byte-identical copies but kept rephrasings of the same question, which are intended as augmentation. We did not rebalance option lengths, which would require rewriting the options; the bias remains in the data. Another small cooldown set attaches written explanations to the training splits of HellaSwag, ARC, and DROP.

Source	Processing (changes from Section 3.1 defaults)	Removed	Final records
Chat-format cooldown	chat-aware SFT checks instead of the document filter; reasoning-aware exact dedup	0.07%	1.84 M
Arabic Wikipedia	stock filter and MinHash not applied; structural removal and verified pairwise dedup	5.8% (stock filter: 38%)	1.27 M
Knowledge infusion	judge-based answer verification, repair, and removal	14.3% of answers	142 K answers
Synthetic multiple choice	minimal length filter; structural validator; byte-identical dedup only	0.1–13.6% by set	0.84 M
Benchmark training items	Arabic-ratio rule only; byte-identical dedup	5.1%	99 K

Table 7: Processing record per cooldown source. Removed is the share of records removed by processing, or of answers where noted.

Decontamination. The released cooldown data was scanned against 554 evaluation files: 1,955 of 4.3 M lines were flagged and 1,952 removed (0.045%). About half of the Wikipedia matches were Quranic verses and other shared religious text rather than copied benchmark items. The scan could not cover HellaSwag, ARC, and DROP, whose reference files were not available to it, and it cannot see translated or paraphrased items.

4.3 Training Recipe and Runs

Table 8 summarizes the settings of each model for the main stage and the cooldown. The main-stage data is identical across sizes.

Recipe. Every model starts from its K2 **Horizon** checkpoint, which is the final post-trained model rather than the pretrained base (Section 2.3). We train all parameters with the next-token prediction loss and start from a fresh optimizer state rather than reusing the one saved with the checkpoint. The optimizer is AdamW (Loshchilov and Hutter, 2017) with $\beta_1=0.9$ and $\beta_2=0.95$, and gradients are clipped at a global norm of 1.0. Each update covers 2,400 to 4,096 sequences of 8,192 tokens (19.7–33.6 M tokens; Table 8), so one pass over the main-stage data takes 6,004 to 11,141 updates.

The learning rate rises linearly over the first 48 updates (30 for the 375B), about 1B tokens, and then stays flat until the cooldown, so the main stage ends at full learning rate and the annealing is left to the

Model	Vocab	Main stage		Cooldown		Batch (tok / seqs)	Peak LR	Weight decay	RoPE	Max context
		Tokens	Updates	Tokens	Updates					
Jais-3-0.9B	64,256 [†]	233.6 B [†]	11,141	4.0 B [†]	189	21.0 M / 2,560	8.0×10^{-5}	0.1	$10^6 + \text{YaRN}$	128K
Jais-3-3.7B	250,624	201.5 B	10,248	3.47 B	177	19.7 M / 2,400	4.0×10^{-5}	0.06	10^7	512K
Jais-3-7B	250,624	201.5 B	10,248	3.47 B	177	19.7 M / 2,400	3.0×10^{-5}	0.06	10^7	512K
Jais-3-32B	250,624	201.5 B	9,606	3.47 B	166	21.0 M / 2,560	1.5×10^{-5}	0.06	10^7	512K
Jais-3-MoVA-36B-A4B	250,624	201.5 B	9,606	3.47 B	166	21.0 M / 2,560	4.0×10^{-5}	0.06	10^7	512K
Jais-3-375B-A23B	250,624	201.5 B	6,004	3.47 B	104	33.6 M / 4,096	1.4×10^{-5}	0.073	10^7	512K

Table 8: Continual-pretraining settings for each Jais 3 model. Tokens and updates are given for the main stage and the cooldown. All models are trained with AdamW on 8K-token sequences; the main stage uses a short warmup and a constant learning rate, and the cooldown decays it linearly to a tenth of the peak. Max context is the context window inherited from K2 Horizon; YaRN uses a scaling factor of 16 from an original 8K context. [†]Token counts depend on the tokenizer: the 0.9B’s smaller vocabulary splits the same text into more tokens.

cooldown and its own data. The peak ranges from 1.4×10^{-5} for the 375B to 8.0×10^{-5} for the 0.9B, and weight decay is 0.06, except 0.073 for the 375B and 0.1 for the 0.9B (Table 8).

The mixture weights are fixed for the whole stage (Section 4.1). The replay portion, 50.4 B tokens, is drawn from K2 Horizon’s final two mid-training stages, which comprise high-quality, SFT-like data. We chose the 75/25 split after ablations with 90/10, 75/25, and 50/50 mixtures of new data and replay, in which 75/25 gave the best balance between learning Arabic and retaining what K2 Horizon already knew. The token budget and the batch size follow our earlier continual-pretraining runs. We also conduct ablation studies on the learning rate and batch size, confirming that the hyperparameters inspired by K2 Horizon are optimal.

Cooldown. The cooldown anneals the model on curated data and re-exposes it to the chat format before SFT begins. It picks up from the last main-stage checkpoint; the learning rate decays linearly from the main-stage peak to 10%, with no warmup, and every other setting stays the same. For every model, it is a single pass over the cooldown mixture; Table 8 gives the tokens and updates of each model.

Following the same 75/25 mixing ratio, the cooldown dataset (Section 4.2) combines replay data from the main stage with a mixture of Wikipedia and chat-formatted data across both thinking modes; incorporating these during the annealing phase both strengthens the model’s factual knowledge and familiarizes it with the thinking formats prior to SFT.

Infrastructure. We trained the 0.9B, 32B, 36B MoVA, and 375B models, in both the main stage and the cooldown, with our in-house framework (xLLM) in bfloat16 with fully sharded data parallelism. The 0.9B ran on NVIDIA H100 80GB GPUs without tensor parallelism; the 32B, 36B MoVA, and 375B ran on NVIDIA H200 141GB GPUs, with 2-way tensor parallelism for the 36B MoVA and 8-way for the 375B. The 3.7B and 7B were trained on Cerebras CS-2 systems in cbfloat16.

To fit the batch at 8K context, we recompute the first and third MLP projections and the attention value projection in the backward pass instead of keeping their activations. The training data is tokenized, packed into 8,192-token sequences, and shuffled across the full dataset before the run starts, so the data loader only reads pre-built shards. Padding accounts for less than 0.02% of tokens, and about 5% of sequences end in a document that was cut at the sequence boundary. Checkpoints hold the model weights and the optimizer state. They were written every 250 updates in the main stage and every 100 in the cooldown, and we kept the last two.

Runs. The main stage took about 1.8K H100 GPU hours for the 0.9B (233.6 B tokens), 10.9 K H200 GPU hours for the 36B MoVA, and 76.9K H200 GPU hours for the 375B; on Cerebras, it took 3.3 K and 4.7K CS-2 system hours for the 3.7B and 7B. The 32B main-stage run did not log its runtime.

The 0.9B main stage ran its 11,141 updates in one uninterrupted job, and the cooldown took another 30 GPU hours. Training loss dropped quickly at first, from 2.81 at the first update to an average of 1.74 around update 500, as the model adjusted to the new mixture. After that it fell slowly, reaching 1.61 by update 2,500, 1.57 by update 5,000, and 1.54 at the end of the stage. There were no loss spikes: no update exceeded the average of the preceding 50 by more than 10%, and after the first 500 updates the gradient norm stayed between about 0.2 and 0.6. Over the cooldown, the loss went from 1.44 to 1.22 as the learning rate came down.

The larger models ended the main stage at a training loss of 1.56 (3.7B), 1.49 (7B), 1.33 (32B), 1.35 (36B MoVA), and 1.21 (375B), and the cooldown at 1.27, 1.21, 0.98, 0.99, and 0.88, each averaged over the last 5% of updates. For SFT we used the final cooldown checkpoint of each model and did not select among intermediate checkpoints.

5 Post-Training

Post-training turns each base model into a chat assistant. Starting from the cooldown checkpoint, we apply three stages in sequence (Figure 1). Each stage targets a different gap left by the one before it: supervised fine-tuning (SFT) establishes the desired behaviors, final token preference optimization (FTPO) corrects a failure mode that SFT introduces, and reinforcement learning with verifiable rewards (RLVR) sharpens performance on tasks whose outputs can be checked automatically. The following sections describe the training data and method for each stage, and report stage-level results for FTPO and RLVR.

The first stage, SFT (Sections 5.1 and 5.2), teaches the model the **Jais 3** chat format, its high- and low-thinking modes, and the Arabic, agentic, and safety behaviors we target. To keep the base models’ existing abilities intact, a quarter of the SFT tokens are replayed from **K2 Horizon**’s mid-training data.

SFT alone, however, leaves the model prone to degenerate repetition loops. The second stage, FTPO (Section 5.3), is a short preference-optimization round that addresses this directly, reducing the average loop rate from 13.8% to 1.2%.

Finally, RLVR (Section 5.4) trains the model with programmatic rewards across five domains: instruction following, mathematics, code, Arabic grammatical analysis, and diacritization.

5.1 SFT Data

SFT trains for one epoch on a mixture of 40.8 B tokens (44.9 B with the 0.9B tokenizer) in 22.2 M examples from 138 mixture entries, a quarter of it replay of **K2 Horizon** mid-training data. The mixture is sampled from a pool of 33 datasets with 174.4 B tokens (Table 9). Most of the pool is the general chat and replay family (153.0 B), of which the run uses only a small part. Most of the pool’s tokens are in completions, and approximately half of the completion tokens are reasoning traces. Reasoning is present throughout the pool. Nearly every dataset is sampled in paired high- and low-thinking variants, 98.2% of rows come from datasets containing at least some reasoning trace, and only six datasets contain none at all.

Table 9 summarizes the pool by dataset family. Each of the 138 mixture entries is a dataset or one of its splits with its own sampling weight; Table 18 gives the mixture’s composition by category, and Section 5.2 describes the training run. The mixture subsets split further by reasoning mode and file shard and add the replay band. The subsections below describe the datasets we built for this release; families without a subsection are summarized only in Table 9.

Family	Samples	Tokens	Trace
General chat and replay	19,263,061	153.03 B	54.3%
Long context	150,890	7.11 B	99.9%
Agentic and tool use	425,251	5.83 B	8.3%
Dialect (Arabic script, Arabizi, ID)	5,919,316	3.54 B	58.8%
Translation and transliteration	1,033,966	0.72 B	20.6%
Instruction following and multi-turn	2,543,981	2.22 B	61.3%
Regional and religious knowledge	2,858,783	1.40 B	44.4%
Robustness and factuality	475,503	0.26 B	90.9%
Safety and identity	259,813	0.11 B	64.3%
Arabic linguistics	285,921	0.07 B	35.0%
Synthetic ArabicMMLU	102,988	0.07 B	78.8%
Law, finance, medical	98,430	0.04 B	98.2%
Total	33,417,903	174.40 B	53.3%

Table 9: The SFT pool by family. Tokens are prompts plus completions; *trace* is the share of completion tokens inside a reasoning trace.

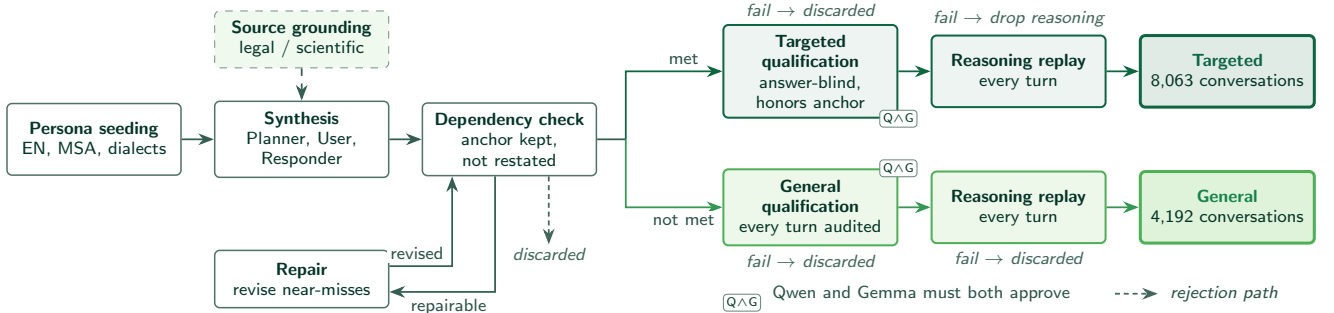


Figure 4: Multi-turn data pipeline. Conversations that realize a long-range dependency, directly or after repair, enter the targeted lane; those that do not, but are otherwise sound, enter the general lane. The lanes differ when a reasoning replay fails: a targeted conversation keeps its original answer without a reasoning trace, whereas a general conversation is discarded.

5.1.1 Multi-Turn Data

Many failures of chat assistants are failures to carry state across a conversation: an early instruction is dropped, a fact the user shared is ignored, or a correct answer is abandoned when the user pushes back. We synthesize English and Arabic multi-turn conversations of two kinds. *Targeted conversations* are built around these failures: each ends with a request whose correct answer depends on a specific earlier turn. *General conversations* carry no such dependency but pass the same quality checks. Both come from one pipeline (Figure 4), which produced 8,063 targeted and 4,192 general conversations, 12,255 in total, each with 4–10 user turns. The SFT mixture (Table 18) includes all of them once, across three entries totaling 79.8 M tokens (0.19% of the mixture).

Synthesis. We follow the structure of the MultiChallenge pipeline (Deshpande et al., 2025). Each conversation starts from a persona sampled from a pool of personas, localized to English, MSA, or an Arabic dialect. A *Planner* writes a blueprint specifying the scenario, the dependency type, and where

the anchoring information is placed. A *User* simulator and a *Responder* then realize the blueprint as a dialogue of 8–20 messages; the Responder sees only the visible conversation, never the blueprint. Qwen3.6-27B (Qwen Team, 2026b) and gemma-4-31B-it (Gemma Team, 2026) serve as both generators and judges. All conversations are synthesized from scratch, without benchmark prompts as seeds, and pass our decontamination pipeline (Section 3.1).

Targeted conversations. We adopt the four dependency types of Deshpande et al. (2025): instruction retention, user-information recall, versioned editing, and self-coherence. An *anchor* turn introduces an instruction, fact, document version, or commitment, and the final user turn depends on it without restating it (Appendix B.1). A structural audit of the visible conversation checks that the anchor is present, that at least one exchange separates it from the final turn, that the final turn does not restate it, and that the anchor is unambiguous. Under strict qualification, the audit also confirms that changing the anchor would change the correct answer. Conversations that narrowly fail the audit are repaired and audited again rather than discarded. Legal and scientific conversations can additionally be grounded in pinned source passages.

General conversations. Many conversations that fail the dependency audit are still good conversations, so we re-qualify them for general quality instead of discarding them. Without the structural guarantee, quality must be established turn by turn: both judges audit every assistant turn for correctness, coherence, language quality, factual accuracy, safety, privacy, and consistent model identity, and check that no turn uses information introduced only later in the conversation.

Verification. Three mechanisms apply to both kinds of data. First, *answer-blind regeneration*: the final answer is regenerated from the visible conversation alone, so the judges evaluate an independent attempt rather than approve an existing one. Second, *unanimous cross-family judging*: a conversation is accepted only if both judges, one from each model family, approve it. This favors precision, since an error specific to one family is unlikely to be shared by the other, and no conversation is accepted on its generator’s judgment alone. Third, *reasoning replay*: every assistant turn is replayed with explicit reasoning, again answer-blind. The replayed answer must be of good quality, consistent with its reasoning, and in agreement with the original answer, because later turns build on it. If a replayed turn fails, a targeted conversation keeps the original answer without a reasoning trace, whereas a general conversation is rejected. Finally, the assembled conversation receives a whole-chat review; Appendix B gives the full criteria and fallbacks.

5.1.2 Country Knowledge Data

We built a question-answer dataset on the 22 member states of the Arab League, covering politics, history, geography, the economy, religion, food, sports, famous people, landmarks, folklore, customs, and local expressions. Approximately 70% of the examples are in the dialect of the country concerned and 30% in MSA; Somalia, Djibouti, and Comoros are covered in MSA only. Every example includes a reasoning trace. Of the 1.38 M examples, the final SFT run uses 1.37 M (0.64 B tokens, 1.6% of the SFT tokens), and 220,511 of the set went through evidence-based verification.

Generation. Questions come from a taxonomy of 29 categories and their subtopics; subtopics name areas of knowledge rather than entities, so the generator chooses what to ask about. For each country, register, and category–subtopic pair, gemma-4-31B-it writes first-person questions with an asker profile (e.g., resident, student, traveler, researcher), a target difficulty, and one of six forms: factual question, explanation, comparison, scenario, opinion, or list. Questions are deduplicated within each country and register by exact match and MinHash. The same model then answers each question with reasoning enabled, in the requested dialect or in MSA.

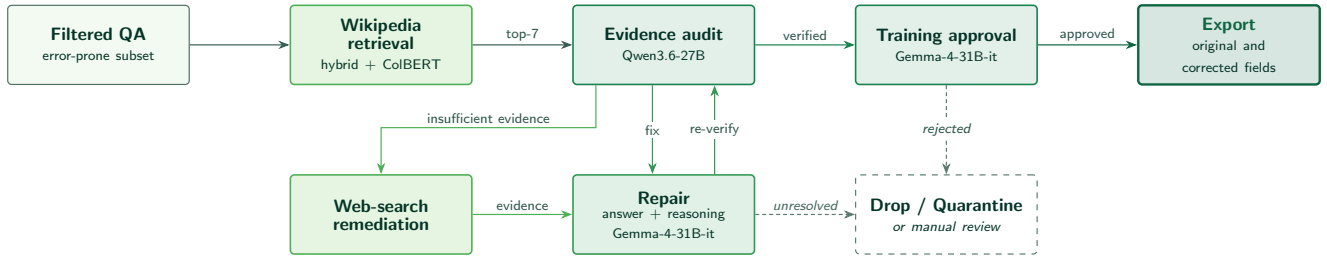


Figure 5: Evidence-based verification of error-prone examples. Wikipedia retrieval is the default evidence source and web search is a fallback. Every rewritten answer is audited again before it can reach the training judge; examples that cannot be verified are quarantined rather than accepted.

Quality filtering. Rule-based filters remove empty, truncated, repetitive, and refused answers, answers less than 55% Arabic, and answers with leaked reasoning. `gemma-4-31B-it` then scores the majority of examples on a five-point scale, keeping those that score at least 4 on every criterion. `gemma-4-31B-it` and `Qwen3.6-27B` independently fact-check every example, and examples flagged by both are rejected. `Qwen3.6-27B` flags dialectal and general answers more often; examples flagged only by `gemma-4-31B-it` more often contain a genuine error, usually a misattributed named entity.

Evidence-based verification. Parametric fact-checking misses recent changes and facts that the generator and both judges get wrong. For the most error-prone category–subtopic pairs, we therefore verify answers against a fixed snapshot of Arabic Wikipedia (Figure 5), chosen over the web for reproducibility, source quality, and control over retrieval. For each question, hybrid retrieval returns 150 passages, which `LFM2.5-ColBERT-350M` reranks to seven. `Qwen3.6-27B` audits the answer against them; answers that are not *correct* are rewritten by `gemma-4-31B-it` from the passages alone and audited again, with targeted web search as a fallback when Wikipedia lacks the evidence. Examples whose answers remain unsupported are quarantined.

Final approval. Before training, `gemma-4-31B-it` checks that each rewrite fixes the audited issues and that every answer is natural Arabic in the requested register, consistent with its reasoning, and free of truncation, repetition, and leaked audit text. *Unverifiable* examples are kept only for non-factual questions.

5.1.3 Translation and Transliteration Data

We built a synthetic corpus for two capabilities: *translation* between English and Arabic (MSA, classical and literary Arabic, and seven regional dialects), and *transliteration* between Arabic script and Arabizi (Franco), the Latin-script convention of informal written Arabic. `gemma-4-31B-it` generated all 812,160 examples; deterministic checks and a blinded LLM judge keep 741,262 (91.3%) for SFT.

Sources. The translation data has two parts. The first reuses the source passages and translation directions of 465,692 samples from the *Jais 2* translation data (Anwar et al., 2026), covering subtitles and dialogue, short sentence pairs, EU and legal text, and classical and literary Arabic, re-rendered into standardized prompts with newly generated answers. The second has 140,219 dialectal samples: Palestinian, North Levantine, and Yemeni samples are built from the public corpora of Table 10, and Egyptian, Moroccan, Tunisian, and Iraqi samples from internal data. For Iraqi, internal English seed sentences are first rendered into synthetic Iraqi Arabic and then translated back into English. The transliteration data (206,249 samples) covers both directions between Arabic and Arabizi across the same seven dialects, seeded from internal Egyptian, Moroccan, and MSA data and from the normalized Arabic passages of the translation data.

Direction	Source corpus	License
Palestinian → English	Curras (El Haff et al., 2022; Jarrar et al., 2017, 2014)	CC BY 4.0 (Creative Commons, 2013)
North Levantine → English	Nabra (Nayouf et al., 2023) and Baladi (El Haff et al., 2022)	CC BY 4.0 (Creative Commons, 2013)
Yemeni → English	Lisan-Yemeni (Jarrar et al., 2022)	CC BY 4.0 (Creative Commons, 2013)

Table 10: Public seed corpora for the dialectal translation data. Egyptian, Moroccan, Tunisian, and Iraqi samples are seeded from internal data.

Axis	Translation	Transliteration
Quality (1–5)	Overall quality, semantic adequacy, dialect/register fidelity, fluency	Transliteration fidelity
Task compliance	Correct direction, language variety, and target script	Correct direction and script; transliterated, not translated
Preservation	Entities, numbers, negation, modality, tense, pronouns, relations	Meaning-bearing words, entities, numbers, names, punctuation, URLs, relations
Output defects	Omissions, additions, source copying, commentary, repetition, multi-passages misalignment	Omissions, additions, accidental translation, commentary, script errors
Source integrity	Actual variety, label agreement, usability, artifacts, placeholders, malformed fragments	Actual variety and agreement with the declared dialect
Content flags	Sexual content, hate, slurs, sectarianism, racism, harassment, threats	

Table 11: Judge criteria. Quality dimensions are scored 1–5; all other fields are categorical or boolean.

Generation. We sample from English and Arabic prompt templates at temperature 1.0, top- p 0.95, and top- k 64, with reasoning enabled or disabled (Figure 6a). Translation prompts require full meaning preservation, a translation-only answer, and order-aligned output for multi-passages inputs. Transliteration prompts require changing only the script and preserving words, names, numbers, and punctuation, while allowing natural Arabizi spelling variation. The reasoning trace is stored separately from the final answer, with provenance and decoding metadata.

Quality filtering. Deterministic checks first drop mechanical failures, such as unresolved placeholders, injected foreign text, repetition, boilerplate, untranslated source copies, and model commentary. **gemma-4-31B-it** (temperature 0, schema-constrained JSON), blinded to the generator’s reasoning trace, then scores the remaining rows against the criteria of Table 11. A row is kept if every quality score is at least 4, every compliance and preservation check passes, and the declared and observed language varieties agree. This leaves 571,151 translation (94.3%) and 170,111 transliteration (82.5%) examples (Figure 6b).

5.1.4 Arabic Instruction Following

Open instruction-following data is mostly English, and its constraints rarely test what makes Arabic text correct: *hamza* and *tā’ marbūṭa*, diacritics, Arabic-Indic digits and punctuation, or the plural, dual, and relational forms of words. We therefore built a synthetic set of 43,780 Arabic prompts, each pairing a natural request with one or two program-checkable constraints; the final SFT run uses the reasoning version of 43,443 of them (0.12 B tokens).

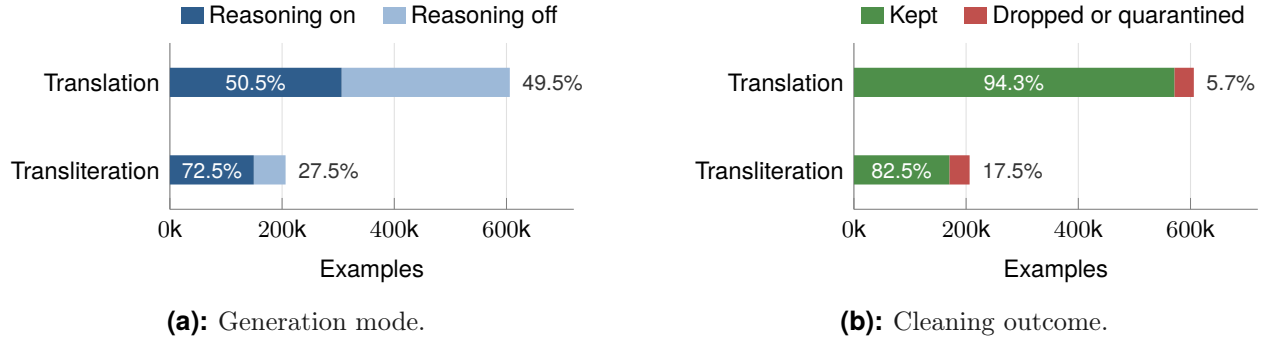


Figure 6: (a) Generations with reasoning enabled and disabled, per task. (b) Kept versus dropped or quarantined examples, as a share of each task’s generated total.

Category	Example constraints
Orthography	exact number of <i>hamza</i> , <i>alif maqṣūra</i> , <i>tā’ marbūṭa</i> , or definite articles
Diacritics and script	exact count or absence of <i>sukun</i> , <i>shadda</i> , and <i>tanwīn</i> ; <i>kashida</i> ; Arabic-Indic digits; Arabic punctuation
Sequence	word counts that increase from sentence to sentence; equal character counts across sentences
Copy and transform	repeat, reverse, or substitute parts of the request; return only the requested output
Format and position	required first or last word, nested brackets or custom separators, paragraph counts; counts of words, numbers, or items from a closed list
Morphology	produce the broken plural, dual, feminine, relational adjective (<i>nisba</i>), participle, or verbal noun of a given word
Combined	two exact-count constraints in one prompt, such as <i>kashida</i> and diacritic counts

Table 12: Constraint categories in the Arabic instruction-following set. Each constraint type has a deterministic validator.

Prompts. Qwen3.8-27B (Qwen Team, 2026c) writes chat-style requests in MSA or in Gulf, Egyptian, or Levantine Arabic, across genres, domains, and response lengths. We append constraints as numbered Arabic instructions, drawn from 70 constraint types in seven categories (Table 12). Approximately 7% of prompts carry two constraints, either a light second one, such as a closing phrase, or a demanding pair of exact counts; constraints that fix the output, such as copy-and-transform or morphology, are never combined.

Filtering and responses. Each constraint type has a deterministic validator, using a curated lexicon of gold forms for morphology. Filters remove prompts with formatting artifacts, prompts less than 55% Arabic, near-duplicates, and prompts that overlap with evaluation data (a shared 8-word sequence, token Jaccard similarity ≥ 0.5 , or a normalized exact match). Because validators do not check that the request was answered, Qwen3.8-27B also judges each prompt with a constraint-satisfying response, rejecting unnatural prompts and responses that ignore the request, the most common failure. The released response is sampled from the same model with reasoning enabled and resampled until it passes every validator in strict mode without formatting artifacts. Each prompt is released single-turn, with and without the reasoning trace.

5.1.5 Long-Context Data

Open post-training data rarely asks a model to reason over long Arabic text. We therefore built multiple-choice questions over long, natively Arabic documents: classical texts, legislation and government publications, financial and academic documents, news, and fiction. Contexts reach 128K tokens, with a median of

Domain	Sources	Question formats
Law	legislation, regulations, and court rulings from Egypt, Saudi Arabia, Qatar, and Kuwait; legal portals	case-based multiple choice
Finance	publications of central banks and market regulators in several Arab countries; further Arabic financial texts	numerical calculation, yes/no statements, multiple choice
Medical	Arabic medical documents; drug regulator guidance; physician-reviewed health portals	multiple choice, clinical vignettes, fill-in-the-blank, short answer

Table 13: Sources and question formats of the law, finance, and medical set. All sources are public Arabic documents.

approximately 43K, and the multiple-choice format makes every answer checkable. The final SFT run uses approximately 111 K of these questions (5.5 B tokens); with earlier builds and a smaller question-answering set over Arabic documents, long-context data totals 7.1 B tokens, approximately 17% of SFT (Table 18).

Construction. Each question is written from one long document or several related ones. Approximately a quarter combine evidence from across the context; the rest are written from one or two passages placed inside a long text, so the model must first locate them. `Qwen3.8-27B` and `gemma-4-31B-it` generate and check the questions. Every question is grounded in source passages, and its distractors are plausible, such as a sentence from the text that does not answer the question, an option with one false detail, or a one-word change. A small share of questions have “the context does not determine the answer” as the correct option. A question is kept only if an independent solve recovers its key, and options that can be ruled out by surface cues, such as length, are rejected.

Responses. `Qwen3.8-27B` solves each final question with reasoning enabled. A response is kept only if its answer matches the key and its reasoning supports that answer with evidence from the context, rather than with test-taking heuristics such as preferring the longest option.

5.1.6 Law, Finance, and Medical Data

Professional questions in law, finance, and medicine depend on local statutes, regulators, and clinical terminology, which open Arabic instruction data covers poorly. We built 98.4K Arabic questions in these domains (45M tokens), grounded in public Arabic sources (Table 13): 10.0 K in law, 6.7 K in finance, and 81.7 K in medicine. The final SFT run uses 97.8K of them.

Construction. We collected public Arabic documents from government, regulatory, and professional websites, following each site’s crawling rules, and removed duplicates and short or mostly non-Arabic documents. From each excerpt, the generator extracts facts with a verbatim supporting quote, such as rules and deadlines in law, rates and formulas in finance, or diagnoses and treatments in medicine, and turns each fact into one self-contained MSA question in a format suited to its domain. `gemma-4-31B-it` and `Qwen3.8-27B` generate and check the questions. A question is kept only if an independent solve recovers its key; filters also remove malformed items, near-duplicates, and legal questions on sensitive political or sectarian topics.

Responses. `Qwen3.8-27B` solves each final question with reasoning enabled, seeing only the question and, for multiple-choice items, its options. A response is kept only if it matches the key: the same option or yes/no label, a number within 0.5% of the key for calculations, the key term or an accepted variant for fill-in-the-blank items, and the key facts of the reference answer for short-answer items.

Task type	Trajectories	Source file	Trajectories
Creation	3,881	DOCX	2,754
Extraction	1,228	CSV	1,835
Editing	3,791	PDF	1,699
		XLSX	1,600
		PPTX	1,012
Total	8,900		8,900

Table 14: The Arabic file-manipulation set by task type and by format of the source file. The agent’s outputs are Excel workbooks in 5,119 trajectories, PowerPoint decks in 1,992, Word documents in 1,553, and CSV files in 223; a trajectory can produce several files.

5.1.7 File Manipulation in Arabic

This set teaches the model to work on Arabic office files: to read an Arabic document, carry out a professional task based on it, and deliver the result as a native file with correct Arabic text and right-to-left layout. It contains 8.9 K multi-turn agent trajectories (0.72 B tokens), generated with **MiniMax-M3** in the OpenCode agent harness (Table 14); the statistics below describe this release. The final SFT run uses an earlier build of 2.2 K trajectories (0.16 B tokens) made with the same pipeline.

Generation. Each example starts from one Arabic source file. **MiniMax-M3** reads every candidate file in full and keeps only those it grades excellent or good. For each kept file, it writes MSA tasks of three types: *creation*, where the file is a reference for producing a new spreadsheet, document, or presentation; *extraction*, where its content is turned into a derived product such as a memo, summary, ranking, or table; and *editing*, where the file itself is revised or converted. Requests follow the style of GDPval (Patwardhan et al., 2026), with a professional role, a business situation, concrete constraints, and a named deliverable, and each task comes with an Arabic grading rubric written by the same model.

Trajectories. A **MiniMax-M3** agent solves each task in OpenCode with shell, Python, file, and web-search tools, while a second instance plays the user. After the agent’s first answer, the simulated user reviews the work against the rubric, which the agent never sees, and asks for corrections for up to two rounds; 7.9 K trajectories contain three user turns. In 1.6 K trajectories, a document skill (a reference guide for producing the requested file format) is installed in the agent’s workspace. Trajectories have a median of 38 assistant turns and 40 tool calls and keep the agent’s reasoning, tool calls, and tool results; 20.6% of completion tokens are reasoning trace.

5.1.8 Safety

Safety data is a small, targeted part of SFT: five sets totaling approximately 0.22% of the tokens sampled in the final SFT run, roughly 92 M tokens. Rather than add a large generic refusal corpus, we built most of it around failures we observed in earlier **Jais** models and around content that is sensitive in our region: religion, national leadership, and the model’s own identity (Table 15).

General safety. We translated an openly released safety SFT set (Nemotron Safety v2) into Arabic with **gemma-4-31B-it**, keeping two thirds Arabic and one third English, and removed 3.4% of rows that failed translation checks. Native Arabic speakers also wrote a set of harmful prompts in MSA, to cover requests that do not arise from translation; we generated safe responses for these and quarantined 27 of approximately 5,800 that reviewers could not agree on.

Set	Content	Languages	Rows	Share
General safety	public safety SFT data translated into Arabic, plus MSA harmful prompts written by native speakers	Ar, En	47.5 K	0.07%
Jais 2 safety	general and regional safety sets inherited from Jais 2	Ar	115.8 K	0.10%
Religious and regional safety	variants of prompts the model had failed, with benign controls	Ar, En	15.6 K	0.03%
Self-identity	questions about the model’s name, maker, and nature	26 languages, Arabizi	79.3 K	0.03%
National leadership	factual questions about UAE leadership	Ar, En	0.6 K	<0.01%

Table 15: Safety data in SFT. Rows are examples in the final SFT run; share is the fraction of tokens sampled in that run.

Religious and regional safety. This set starts from 1,761 prompts on Islamic and UAE-related topics that an earlier Jais model answered unsafely or inaccurately. For each prompt we generated twelve variants: adversarial rephrasings that should still be refused, and benign questions on the same topic that should be answered, so that the model learns to distinguish the two rather than refuse the topic outright. A response entered training only when two independent judge models from different families both approved it. A first review with a single judge had approved only 38.9% of responses and flagged over 10,000 factual errors and approximately 1,400 unsafe compliances, which is why we moved to agreement between two judges. This set draws no prompts from HarmBench, XSTest, or AraSafe, which we use for evaluation.

Self-identity. Jais 3 is trained from a base model with its own name and maker, and part of its training data was written by other assistants. We therefore trained on questions about the model’s identity in 26 languages, and removed identity leaks from the K2 Horizon mid-training replay set (Section 4.1). An off-the-shelf identity filter missed 41% of the leaks a judge confirmed there, because most of them sit in reasoning traces it does not read.

Limitations. Our system-prompt ablation covers safety behavior only: we compared the deployed prompt against no system prompt on various red-team safety sets, and did not measure the prompt’s effect on capability benchmarks. Safety scores are also sensitive to decoding: most prompts score the same way in every run at temperature 1.0, but a minority split across runs, so single-run numbers should be read with that variance in mind. Safety evaluation results, including over-refusal on XSTest, are reported in Section 6.5.

5.1.9 Robustness

Users do not type like benchmark questions. Arabic input in particular carries misspellings that change meaning (a missing dot, *tā’ marbūṭa* written as *ha*), dialect mixed into MSA, and words run together. We want the model to do what a careful person does: answer the intended question when it can be recovered, and ask when it cannot. The robustness data has two parts, one for each behavior (Table 16): mutated questions, which are in the final SFT run, and ambiguous misspellings, which were built but not used in it. Both parts were generated with `gemma-4-31B-it`.

Mutated questions. This part targets questions whose wording is corrupted but whose intent can still be recovered. When native speakers probed an earlier Jais model with mutated questions, it recovered the intent almost every time; the exception was real-word substitution. In “what are the physical benefits of

Part	Input	Target behavior	Examples	Final run
Mutated questions	Typos, dialect mixing, run-together words, false premises, or moral dilemmas; intent is recoverable	Answer the intended question	5,131	✓
Ambiguous misspellings	Misspellings with more than one plausible correction	Ask for clarification when a guess could mislead	20,000	–

Table 16: The robustness SFT set. Only the mutated questions are used in the final SFT run (5,131 examples, 0.007% of its tokens). The ambiguous misspellings consist of 10,000 prompts with two responses each.

al-ṣiyām (fasting)?”, changing one letter gives *al-ṣayyād* (the hunter), a valid word, and the model answered about the benefits of hunting.

Prompts (mutated questions). Native speakers wrote 1,392 seed prompts: 893 with misspellings or false premises, covering real-word substitutions, *tā’ marbūṭa*, letter dots and shapes, dialect mixed into MSA, run-together words, and English typos and reordered words, and 499 moral or pragmatic dilemmas. *gemma-4-31B-it* generated five variants of each seed, and we removed near-duplicates, keeping 6,000 of about 7,000 prompts.

Responses (mutated questions). The same model answered each prompt once, half of the prompts with a reasoning trace and half without. Three judge models from different families checked whether each answer recovered the intended question; they removed 14.5% of answers, leaving 5,131 examples (2,521 with a reasoning trace, 2,610 without), and a manual check of 100 accepted items found 97 correct.

Ambiguous misspellings. This part targets misspellings with more than one plausible correction, where silently correcting the word can change the request. In “Can you compare the startup ecosystem growth in Riyadh versus the trends in Duba?”, *Duba* may be a misspelling of Dubai or refer to the Saudi city of Duba. In an Arabic prompt asking for traffic data in the *al-riyāḍa* (‘sport’) area, an extra *tā’ marbūṭa* may have turned Riyadh (*al-riyāḍ*) into a common noun.

Prompts (ambiguous misspellings). We used *gemma-4-31B-it* to generate 12,619 candidate prompts, half in English and half in Arabic. Each request sampled a topic, cycling through the 26 domains of a topic taxonomy (1,122 of its 1,125 leaf topics appear), an ambiguity category to emphasize, and two of six fixed examples in the target language. The categories are near-neighbor named entities, acronyms and tickers, technical terms, version numbers and codes, domain-dependent words, cross-language homographs, Latin–Arabic transliterations, and, for Arabic only, letter shapes and dots, including *hamza* and *tā’ marbūṭa*. The instructions excluded misspellings with an obvious correction, such as *teh* for *the*. Each record marks the misspelling, its intended correction and alternative readings, and three labels: whether an assistant should ask for clarification, how hard the intent is to recover, and how confident the correction is. We dropped records whose misspelling was absent from the prompt, identical to its correction, or a repeat of an earlier (misspelling, correction) pair, keeping 10,000 of 12,619, and manually inspected random subsamples of the result.

Responses (ambiguous misspellings). The same model answered every prompt twice, with and without a reasoning trace, under a system prompt telling it to assume the most likely reading when that is safe and to ask when a guess could make the answer unhelpful. Answers cut off by the token limit were

regenerated with a larger budget. This set was not used in the final SFT run, because the final model no longer showed this weakness.

5.2 Supervised Fine-Tuning

SFT starts from each model’s final cooldown checkpoint. All six models follow one recipe: one epoch over a 40.8B-token mixture drawn from the pool of Section 5.1, packed into 128K-token sequences so that long-context data is trained at full length. The models differ in learning rate, weight decay, hardware, and parallel layout, and the 0.9B also in its tokenizer, which splits the same mixture into more tokens (Table 17). The 3.7B was trained on an earlier revision of the mixture, whose token count differs slightly.

5.2.1 Training Setup

Chat format and loss. All data is rendered with the K2 Horizon chat template, so the released models share one format. Each turn sits between `<|ifm|im_start|>` and `<|ifm|im_end|>` markers with a role name, and tool definitions go in the system turn. The loss covers every assistant turn in a conversation, not only the last one; system, user, and tool-result turns are masked. Tool calls are part of the assistant turn, so in all agentic data the model trains on the tool calls it has to produce. Each conversation starts with a BOS token and ends with a supervised EOS token. The loss is averaged over all supervised tokens in an optimizer step, so each conversation contributes in proportion to its number of trained tokens.

The models have two modes, high-thinking and low-thinking, and the mode is set by a tag at the start of the assistant turn. In high-thinking mode the turn opens with `<|ifm|think>`, followed by the reasoning trace, the closing tag and then the answer. In low-thinking mode the turn opens with an empty `<|ifm|think_faster>` block and goes straight to the answer, with no reasoning trace. These are the high- and low-effort tags of the K2 Horizon template; its medium-effort tag `<|ifm|think_fast>` is not used. At inference time the template writes the opening tag for the requested mode, with high-thinking as the default. During training we mask the opening tag, since the prompt supplies it, but train on the closing tag so the model learns when to stop reasoning and start answering. Of the 31.4M assistant turns in the mixture, 16.4M are in high-thinking mode and 15.0M in low-thinking mode, so the two are close to balanced. The 16.7K turns that had no reasoning field were put in low-thinking mode. The high- and low-thinking results in Section 5.5 refer to these two modes.

Sequence length and packing. Long-context QA makes up 17% of the mixture, so all sizes train with 131,072-token sequences. Conversations are packed into these sequences with a best-fit packer that never cuts a conversation in two; samples longer than the limit are removed before packing. Conversations are separated by EOS tokens and attention does not cross from one conversation into the next. The 6,917 conversations (0.03%) longer than 131,072 tokens were removed. Padding is negligible, 0.001–0.002% of tokens in every run.

Hyperparameters. We train all parameters from a fresh optimizer state with AdamW ($\beta_1=0.9$, $\beta_2=0.95$) and gradient clipping at 1.0. Peak learning rate and weight decay follow continual pretraining (Table 8), except that the 375B uses a lower peak of 1.0×10^{-5} instead of 1.4×10^{-5} . The learning rate warms up linearly over the first 2% of optimizer steps and then decays linearly to 10% of its peak. Each optimizer step covers 32 sequences of 131,072 tokens (approximately 4.2M tokens), and one epoch of the mixture takes 9,732 updates (40.8B tokens). The MoVA and 375B runs continued for 72 more updates (9,804 updates, 41.1B tokens), slightly more than one epoch. Table 17 gives the per-model settings.

Mixture. The SFT mixture has 138 mixture entries and 22.2M examples.

Model	Optimization				Compute		
	Peak LR ($\times 10^{-5}$)	WD	Steps	Tokens	GPUs	TP / CP	Time
Jais-3-0.9B	8.0	0.1	10,711	44.9 B	256	1 / 8	10.5 h
Jais-3-3.7B	4.0	0.06	9,754	40.9 B	256	1 / 8	21.5 h
Jais-3-7B	3.0	0.06	9,732	40.8 B	256	1 / 8	26.1 h
Jais-3-32B	1.5	0.06	9,732	40.8 B	512	2 / 8	53.3 h
Jais-3-MoVA-36B-A4B	4.0	0.06	9,804	41.1 B	256	2 / 4	33.7 h
Jais-3-375B-A23B	1.0	0.073	9,804	41.1 B	1,024	8 / 4	45.2 h

Table 17: SFT settings per model. WD is weight decay; TP / CP are the tensor- and context-parallel degrees. Tokens are those processed (optimizer steps $\times$ 32 sequences $\times$ 131,072 tokens); the 0.9B processes more because its 64K-vocabulary tokenizer splits the same data into more tokens. Time is the wall-clock training time, excluding the intervals between restarts.

Category	Tokens	Share
Arabic instructions, culture, safety	10.3 B	25.1%
Replay (K2 Horizon mid-training)	10.3 B	25.0%
Long-context QA	7.1 B	17.3%
English chat, code, math, STEM	6.0 B	14.5%
Agentic and tool use	5.8 B	14.2%
Translation and Arabizi	1.6 B	3.9%
Total	41.1 B	100%

Table 18: SFT mixture by category, in tokens sampled over the 9,804 updates of the MoVA and 375B runs (41.1 B, slightly more than one 40.8 B epoch; 250K-vocabulary tokenizer). Shares are the same for every run. Agentic and tool use comprises 425 K examples.

Table 18 groups them by type. A quarter of the tokens are replay from the last K2 Horizon mid-training phase, which we kept to protect the general abilities the model had before SFT. The rest is split across Arabic instruction data, long-context QA, agentic and tool-use trajectories, English instruction data, and translation and Arabizi. Most sources are seen once; a few small, high-value sets are repeated up to about six times.

Infrastructure. All six models were fine-tuned with our in-house training framework, xLLM, on NVIDIA H100 80GB GPUs (dense models) or H200 141GB GPUs (Jais-3-MoVA-36B-A4B and Jais-3-375B-A23B); Table 17 gives the GPU counts and parallelism degrees per model. To fit 128K-token sequences in memory, every run combines fully sharded data parallelism with context parallelism, and the three largest models also use tensor parallelism. All runs compute in bfloat16; the two sparse models keep parameters, gradients, and gradient reductions in float32. The 32B run was restarted 23 times and the 0.9B run once; the 3.7B and 7B runs completed without a restart.

Training loss. Averaged over the first and last 5% of optimizer steps, training loss on the supervised tokens fell from 1.10 to 0.80 for the 0.9B, 0.80 to 0.71 for the 3.7B, 0.72 to 0.64 for the 7B, 0.64 to 0.58 for the 32B, 0.66 to 0.58 for the 36B MoVA, and 0.61 to 0.53 for the 375B. For the 7B and 32B, most of the drop came in the first half of training.

5.3 Final Token Preference Optimization

Reasoning models can fall into *doom loops*, in which they repeat a span of text until the output budget is exhausted (Pipis et al., 2025). Such loops sustain themselves: each repetition of a span raises the

probability of the next one (Xu et al., 2022). A looping response never reaches an answer, and in agentic use a single looping call can stall an entire episode. Greedy and low-temperature decoding make loops more frequent. Liquid AI attributes them to three interacting causes: over-trained tokens such as “Wait” that become disproportionately likely, a context in which a recently generated sequence becomes more likely to recur, and low-temperature sampling, which removes the randomness that would let the model exit the cycle (Liquid AI, 2026).

We address doom loops with a short Final Token Preference Optimization (FTPO) stage between SFT and RLVR (Figure 1). FTPO was introduced in Antislop to suppress overused phrases (Paech et al., 2025), and the Antidoom recipe of Liquid AI (2026) applies it to repetition. The training signal comes from the model’s own looping generations. At the token where a loop begins, that token is the rejected choice, and alternatives that the model itself ranked as plausible at the same position are the chosen ones. A LoRA adapter (Hu et al., 2022) learns these single-token preferences under a tether that keeps the remaining logits close to those of the starting model. All six models, from Jais-3-0.9B to Jais-3-375B-A23B, go through the same procedure with the same objective and optimizer settings. Only the number of mined pairs, the maximum training length, and the adapter placement in the mixture-of-experts models differ between models.

5.3.1 Data

FTPO data is mined on-policy: every model generates from its own SFT checkpoint, so every rejected token is one that the model itself produces. A training example is a triple (x, r, C) , where the context x is the prompt followed by the generation up to the start of a loop, r is the rejected token, and C is the set of chosen tokens.

Prompts. Prompts come from Antidoom Mix v1.0³, a prompt-only mixture of public instruction-following, chat, mathematics, code, and question-answering datasets. We use a fixed snapshot of 474,229 prompts from 14 source datasets. The eight largest sources (MMLU, PubMedQA, AutoIF, UltraChat, Evol-CodeAlpaca, MetaMathQA, UltraInteract, and LMSYS Arena conversations) each account for 10–11% of the prompts. Prompts longer than 2,000 tokens are discarded. Each prompt is followed by the instruction `Think through the problem step by step, then respond with your final answer as "Answer: <your answer>".` and rendered with the model’s chat template in high-thinking mode.

Mining. We sample at temperatures 0.1, 0.2, 0.3, and 0.4 with top- k 50, min- p 0.01, and top- p 1.0, generate up to 8,192 new tokens, and record the top-20 log-probabilities at every position. Low temperatures are used because loops are most frequent there. The 8,192-token budget is sufficient because a pair is extracted where a loop starts, and later tokens contribute nothing. Mining stops once at least 18,000 pairs (models up to 7B), 24,000 pairs (Jais-3-32B and Jais-3-MoVA-36B-A4B), or 16,000 pairs (Jais-3-375B-A23B) have been collected.

Loop detection and pair construction. Loops are identified with the Antidoom detector, without modification. The detector samples a 16-character fingerprint of the decoded text every 128 characters and searches for a nearby recurrence of it, which yields a candidate period of 1 to 1,024 characters. A loop is confirmed when a unit of that period repeats at least four times consecutively and covers at least 60 characters. The rejected token is the first non-whitespace token of the first repetition, the point at which the model begins to copy the unit again. The period is aligned in token space, and tokens consisting only of whitespace or punctuation are skipped. The chosen tokens are drawn from the top-20 log-probabilities at the same position, renormalized at the sampling temperature after the rejected token and its case variants

³<https://huggingface.co/datasets/LiquidAI/antidoom-mix-v1.0>

are removed. We keep up to 20 tokens whose probability is at least 1% of that of the most likely remaining token and discard whitespace-only tokens. Every chosen token is therefore a continuation that the model already considered plausible at that position.

Training subset. For each model we sample 12,000 of the mined pairs. The sampling flattens the distribution of rejected tokens: a token whose count exceeds the median count m is sampled with weight $(m/\text{count})^{0.3}$, so that no single loop opener, such as “Wait”, dominates the data. Chosen tokens are deduplicated within each pair by normalized surface form. Chosen tokens whose count exceeds the 95th percentile of chosen-token counts (and at least 50) are removed from some pairs, which moves their count toward that reference with strength 0.5. Finally, we discard pairs whose context exceeds the maximum training length of 6,000 tokens (4,096 for **Jais-3-32B**), pairs whose rejected or chosen token is not a single token, and pairs with no chosen token left. Between 10,530 and 11,486 pairs per model remain.

5.3.2 Training Setup

FTPO acts only on the next-token distribution at the end of each context. We train a LoRA adapter on the SFT checkpoint and merge it into the weights after training.

Objective. Let z denote the logits of the model being trained at the last position of x , and let z^{ref} denote the logits of the same model with the adapter disabled, that is, of the SFT checkpoint. The loss has two parts: a preference term that changes the decision between the rejected token and its alternatives, and a tether term that keeps everything else close to the SFT checkpoint. Expectations are taken over the pairs in a batch.

Preference term. For a chosen token $c \in C$, let $\Delta_c = z_c - z_r$ be its logit margin over the rejected token r . The preference term pushes every chosen token above the rejected one:

$$\mathcal{L}_{\text{pref}} = \mathbb{E} \left[\frac{1}{|C|} \sum_{c \in C} w_c \text{softplus}(\varepsilon - \Delta_c) \right], \quad w_c = \text{clip} \left(\frac{\varepsilon - \Delta_c}{\varepsilon}, 0, 1 \right). \quad (1)$$

The weight w_c decays to zero as the margin approaches ε , so pairs that are already resolved stop contributing.

Tether term. Let $T = \{r\} \cup C$ be the set of target tokens and $\bar{T}$ the rest of the vocabulary. The tether penalizes any change in the non-target logits, but lets the target logits move by up to τ_t at no cost:

$$\mathcal{L}_{\text{tether}} = \lambda \mathbb{E} \left[\frac{1}{|\bar{T}|} \sum_{v \in \bar{T}} (z_v - z_v^{\text{ref}})^2 \right] + \lambda_t \mathbb{E} \left[\frac{1}{|T|} \sum_{v \in T} \max(|z_v - z_v^{\text{ref}}| - \tau_t, 0)^2 \right]. \quad (2)$$

Total loss. The FTPO loss is the sum of the two terms:

$$\mathcal{L} = \mathcal{L}_{\text{pref}} + \mathcal{L}_{\text{tether}}. \quad (3)$$

Together, they confine the update to the decision between the rejected token and its alternatives. We use $\varepsilon = 2$, $\lambda = 0.4$, $\lambda_t = 0.05$, and $\tau_t = 0.5$ for all models.

Hyperparameters. The adapter has rank 256, $\alpha = 128$, and no dropout. It covers the attention projections (query, key, value, and output), the MLP projections (gate, up, and down), and the LM head. For **Jais-3-MoVA-36B-A4B**, it covers the query, key, and output projections, the value projections of the three dense layers (MoVA layers use value experts instead), the shared expert, the MLPs of the dense layers, and the LM head; the routed experts and value experts remain frozen. For **Jais-3-375B-A23B**, it covers the query, key, value, and output projections of every layer, the shared expert, the MLPs of the

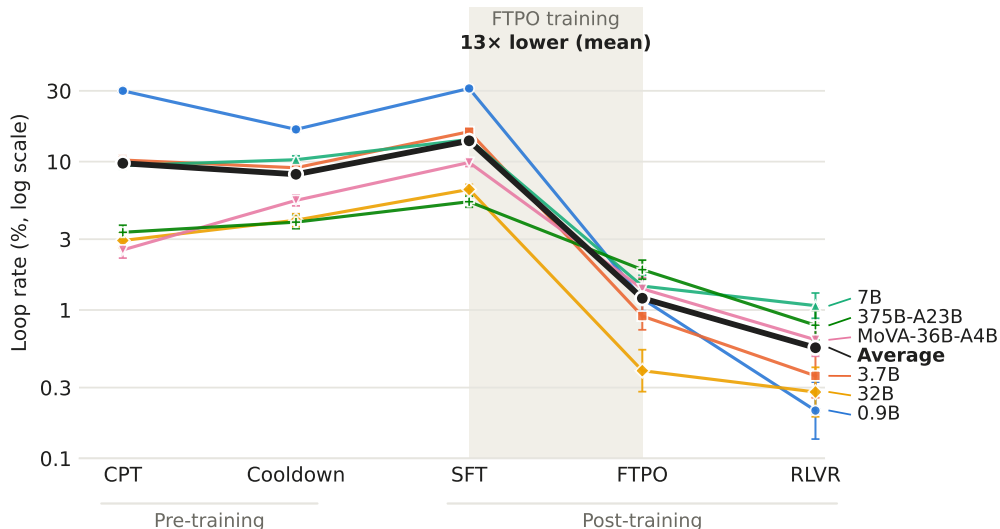


Figure 7: Loop rate (% on log scale) of the released checkpoint of each training stage on the internal evaluation set, averaged over temperatures 0.1, 0.3, and 0.6. RLVR denotes the final model. Error bars are Wilson 95% intervals, and the shaded band marks the FTPO checkpoint.

three dense layers, and the LM head; the routed experts remain frozen. We optimize with AdamW at a learning rate of 1.5×10^{-5} with a linear schedule, 10% warmup, and weight decay 0.01, for at most one epoch with an effective batch size of 16 pairs.

Early stopping. Training is controlled by *chosen-win*, the fraction of a pair’s chosen tokens whose log-probability exceeds that of the rejected token, averaged over pairs. We evaluate it every 5 optimizer steps, averaged over the steps since the previous evaluation, and stop at the first evaluation at which it reaches 0.35. [Liquid AI \(2026\)](#) recommend starting from 0.4 and report strong loop reduction for values between 0.15 and 0.3. Stopping early also bounds how far the model moves from its SFT checkpoint: in every run, the tether terms of Equation 2 grow steadily up to the stopping step.

5.3.3 Results

Evaluation set. We measure looping on an internal evaluation set of 3,061 prompts. The prompts come from the same public sources as the mining prompts but do not overlap with them: no evaluation prompt shares its source identifier or its normalized text with any mining prompt, so none was generated on or trained on during FTPO. All prompts fit the 2,000-token prompt limit under every model’s tokenizer. The set contains prompts from Evol-CodeAlpaca (758), LMSYS Arena conversations (675), MathQA (496), MetaMathQA (477), UltraInteract (319), MATH (132), GSM8K (125), and APPS (79).

Checkpoints and protocol. We evaluate the released checkpoint of every training stage of each model: CPT and Cooldown (continual pretraining), SFT (the input to FTPO), FTPO, and the final model after RLVR. Each checkpoint is evaluated with its own tokenizer and chat template, with the same prompt instruction as in mining and in high-thinking mode. We draw one sample per prompt at temperatures 0.1, 0.3, and 0.6 (top- k 50, min- p 0.01, top- p 1.0) with a budget of 22,000 new tokens, which gives 9,183 generations per checkpoint. The *loop rate* is the percentage of generations in which the mining loop detector fires. These temperatures are a stress test: in high-thinking mode, the released models are recommended to be sampled at temperature 0.7 (Arabic) or 1.0 (English).

Model	Continual pretraining		Post-training			
	CPT	Cooldown	SFT	FTPO	RLVR	Reduction
Jais-3-0.9B	30.0	16.5	31.1	1.2	0.2	26×
Jais-3-3.7B	10.3	9.1	15.9	0.9	0.4	17×
Jais-3-7B	9.4	10.3	14.1	1.5	1.1	10×
Jais-3-32B	2.9	4.0	6.5	0.4	0.3	17×
Jais-3-MoVA-36B-A4B	2.5	5.5	9.9	1.4	0.6	7×
Jais-3-375B-A23B	3.3	3.9	5.4	1.9	0.8	3×
Average	9.8	8.2	13.8	1.2	0.6	13×

Table 19: Loop rate (%) after each training stage, averaged over sampling temperatures 0.1, 0.3, and 0.6 (9,183 generations per checkpoint; Wilson 95% half-widths ≤ 1.0 point). RLVR is the final model. *Reduction* is the ratio of the SFT to the FTPO loop rate; the average row reports the mean of the per-model ratios.

Findings. FTPO reduces looping in all six models (Figure 7, Table 19). Averaged over the models, the loop rate falls from 13.8% at the SFT checkpoint to 1.2% after FTPO, and every FTPO checkpoint loops on fewer than 2% of generations. The gain is largest for the smallest models, which also loop most after SFT, consistent with the observation that looping decreases with model capacity (Pipis et al., 2025). The loop rate of Jais-3-0.9B falls from 31.1% to 1.2%, a 26-fold reduction, and those of Jais-3-3.7B and Jais-3-7B fall 17- and 10-fold. The larger models start from lower rates (5.4–9.9%), so their reductions are smaller in absolute terms, but FTPO still lowers their loop rates 3- to 17-fold.

FTPO also removes the concentration of loops at low temperature. Before FTPO, loops are most frequent under near-greedy decoding, where a self-reinforcing repetition has little chance to be broken by sampling (Xu et al., 2022; Liquid AI, 2026). For Jais-3-7B, the SFT checkpoint loops on 18.8% of generations at $T = 0.1$ and on 7.7% at $T = 0.6$. After FTPO, its loop rate is between 1.1% and 1.9% at all three temperatures and rises slightly with temperature. Every model shows the same reversal, and at $T = 0.1$ the loop rate falls from 7.0–37.5% to 0.1–1.6%. RL does not reintroduce loops: every final model loops less than its FTPO checkpoint, on 0.2–1.1% of generations. Removing loops also helps on benchmarks, where a looping response never reaches an answer: on Jais-3-7B, FTPO adds approximately two points to the Arabic average.

Scale. Antidoom was designed for small reasoning models, which are the most prone to doom loops, and was demonstrated on dense models with 2.6B and 4B parameters (Liquid AI, 2026). In Antislop, the largest model trained with FTPO has 70B parameters and proved more sensitive to the training than the smaller ones (Paech et al., 2025). We apply the same objective and hyperparameters to Jais-3-375B-A23B, a mixture-of-experts model approximately two orders of magnitude larger than the models Antidoom was shown on. As expected at this scale, its SFT checkpoint loops least of the six models, on 5.4% of generations. FTPO nevertheless lowers this rate to 1.9%, and from 7.0% to 1.6% at $T = 0.1$, and the final model loops on 0.8% of generations.

5.4 Reinforcement Learning

5.4.1 Data

We apply RLVR to five domains: instruction following, mathematics, code, Arabic grammatical analysis, and diacritization. Every prompt admits automatic verification, through constraint checking, comparison with a reference, or execution of test cases, so that each response can be scored by a rule-based verifier (Section 5.4.2); Table 20 lists the sources of each domain.

Instruction following. Each prompt pairs a user request with one or more constraints checked by rule-based verifiers, such as ending with a given word, using no commas, or including a keyword. We draw on two sources. The first combines the AllenAI multi-constraint⁴ and Nemotron-RL instruction-following⁵ datasets, translated into Arabic with `gemma-4-31B-it`, with up to five constraints per prompt. Because translation often desynchronized a prompt and its verifier, for example a keyword verifier still expecting the English word, we kept only prompts whose constraint specification survived or could be restored, and removed those whose verifier would penalize a correct Arabic response or that had no rule-verifiable constraint. The second is a synthetic set of 20 Arabic-specific constraint families (21 for the 375B) covering orthography, diacritics, morphology, and structure, such as exact counts of *hamza*, *tā' marbūṭa*, or *shadda*, or the dual, relational adjective (*nisba*), or feminine form of a given word. Each prompt attaches one constraint, in one of approximately ten paraphrases, to an open-ended Arabic request on cuisine, UAE knowledge, religion, or finance. Morphology verifiers use answer keys of 1.6K–7.1K words per family, expanded with CAMEL Tools (Obeid et al., 2020) (`calima-msa-r13` (Taji et al., 2018)) and arramooz (Zerrouki, 2011). The synthetic set uses the AraIFBench verifiers but shares no prompt with the benchmark (by exact match or a Jaccard similarity of at least 0.6). Instruction following is upsampled approximately 1.7-fold, from 4.7 K distinct prompts to 7.8 K. The 375B also trains on 10.9 K new instruction-following prompts, screened with eight samples each from an earlier 375B checkpoint: we keep prompts whose rewards vary across samples and exclude those that no sample or every sample solves.

Mathematics and code. Both domains are drawn from an Arabic translation of the mathematics and code subsets of Guru⁶. Each mathematics problem has a reference answer, and each code problem a set of test cases.

Grammatical analysis and diacritization. Both tasks are built from classical Arabic texts and the canonical text of the Qur'an; grammatical analysis also uses the official answer keys of school examinations from several Arab countries. In grammatical analysis (*i'rāb*), the prompt lists the words of a sentence to analyze, in the order of a word-by-word reference taken from works of Qur'anic and general *i'rāb* or from the answer keys. Every analyzed word must occur verbatim in its sentence, every Qur'anic analysis is anchored to a specific verse, and repeated words are numbered and scored separately. In diacritization (*tashkīl*), sentences come only from fully vocalized texts and are kept only if every letter whose diacritic is required by grammar or morphology is marked, apart from a few standard conventions, and if removing the diacritics reproduces the prompt exactly. Most items, 76% in grammatical analysis and 88% in diacritization, are also checked against their source passage by two LLM judges from different model families and kept only if both accept them. We remove duplicates, hold out a validation split by source, and score each reference against itself, discarding prompts whose reference obtains a reward below 0.99; this excludes references that the reward function cannot interpret. The mixture samples 4.8 K grammatical-analysis and 2.0 K diacritization prompts from these pools.

5.4.2 Reward Functions

All rewards are computed by deterministic rule-based verifiers. A response is scored once its generation ends, and the verifiers read only its answer, the text after the closing reasoning tag. A response is truncated if it reaches the generation limit of $H_{\max} = 16,384$ tokens before the end-of-turn token. Because rule-based rewards remain open to reward hacking, the instruction-following and Arabic-language rewards add penalties and caps for degenerate outputs such as repetition, padding, and language switching.

⁴https://huggingface.co/datasets/allenai/IF_multi_constraints_upto5

⁵https://huggingface.co/datasets/nvidia/Nemotron-RL-instruction_following

⁶<https://huggingface.co/collections/IFM/guru>

Domain	Sources (entries)	Entries
Instruction following	Synthetic Arabic constraints (5,044), AllenAI multi-constraint (2,267), Nemotron-RL (507)	7,818
Mathematics	DAPO / OR1 pool (2,393), DeepScaleR (1,325)	3,718
Code	PrimeIntellect (582), TACO (338), LiveCodeBench (67), other (413)	1,400
Grammatical analysis	Qur’anic text (2,475), classical texts (1,940), exam keys (418)	4,833
Diacritization	Classical texts (1,752), Qur’anic text (246), exam keys (2)	2,000
Total		19,769

Table 20: Source composition of the final RLVR mixture, in training entries after selection and upsampling. Mathematics and code sources come from Guru; the 67 LiveCodeBench problems do not overlap with LiveCodeBench v6 used in evaluation.

Instruction following. Let a be the answer of a response o , the text after its closing reasoning tag, and let $v_k(a) \in \{0, 1\}$ be the verdict of the verifier for the k -th of the prompt’s $K \leq 5$ constraints; a constraint that cannot be verified counts as failed. The reward starts from the fraction $s(a)$ of satisfied constraints and subtracts four penalties, each in $[0, 1]$:

$$\tilde{R}_{\text{IF}}(a) = \underbrace{\frac{1}{K} \sum_{k=1}^K v_k(a)}_{s(a)} - \lambda_{\text{rep}} P_{\text{rep}} - \lambda_{\text{short}} P_{\text{short}} - \lambda_{\text{long}} P_{\text{long}} - \lambda_{\text{lang}} P_{\text{lang}}, \quad (4)$$

with $(\lambda_{\text{rep}}, \lambda_{\text{short}}, \lambda_{\text{long}}, \lambda_{\text{lang}}) = (0.4, 0.3, 0.1, 0.2)$. The *repetition* penalty P_{rep} is the square of the largest of four normalized statistics: dominant-word frequency, repeated n -grams, duplicated sentences, and compressibility, ignoring text that the prompt asks to repeat. The *length* penalties compare the answer length n with a floor f and a ceiling u :

$$P_{\text{short}} = \mathbf{1}[s(a) \geq 0.8] \left(\frac{[f - n]_+}{\max(1, f)} \right)^2, \quad P_{\text{long}} = \left(\frac{[n - u]_+}{\max(1, u)} \right)^2, \quad (5)$$

where $[x]_+ = \max(x, 0)$; only otherwise-correct answers are penalized for brevity. An explicit word, sentence, or paragraph constraint sets f and u in its own unit. Otherwise, n counts words, f is a heuristic floor set by the requested form of the answer, and $u = 1,200$, which also applies when a constraint gives only a lower bound. The *language* penalty P_{lang} grows with the fraction of letters in the wrong script, Latin letters in an Arabic answer or Arabic letters in an English one, and is waived for code and structured output. Finally, four severe failures cap the reward. With $\kappa(a)$ the lowest applicable cap among -0.5 (degenerate repetition), -0.25 (truncation), 0 (substantial use of the wrong script), and 0.2 (every constraint met but $n < 0.33f$), and $\kappa(a) = +\infty$ if none applies,

$$R_{\text{IF}}(o) = \begin{cases} 0, & |a| = 0, \\ \text{clip}(\min(\tilde{R}_{\text{IF}}(a), \kappa(a)), -1, 1), & \text{otherwise,} \end{cases} \quad (6)$$

where $|a| = 0$ denotes an empty answer, for example when the response is truncated within its reasoning.

Mathematics and code. Both rewards are binary up to negligible shaping terms and are conditioned on the answer format: a response that is truncated, or that lacks a single final-answer statement with the prescribed Arabic marker, receives 0. A mathematics answer receives 1 if its final expression, given in a `\boxed{\}` environment or as an option for a multiple-choice question, is equivalent to the reference under

Math-Verify (Kydlíček, 2025), which tests symbolic and numerical equivalence. A code answer receives 1 if its final program passes every test case.

Grammatical analysis and diacritization. Both rewards grant partial credit through a score $S \in [0, 1]$. Deviations from the canonical format (one *word : analysis* line per word, or the bare diacritized text) are penalized with coefficient ϕ :

$$R = (1 - \phi(1 - F)) S, \quad (7)$$

where $F = 1$ if the format is canonical and 0 otherwise.

Diacritization. *Preprocessing.* An allowlisted answer label or a single quoted or bracketed span matching the reference’s letters is extracted as the candidate. Texts are NFKC-normalized, tatweel is removed, and each letter is paired with its set of diacritics.

Score. Let N be the number of letters in the reference. The score is 0 if $N < N_{\min}$ or if the base letters differ. Otherwise, letter i is correct, $m_i = \mathbf{1}[D_i = D_i^*]$, when its diacritic set D_i matches the reference set D_i^* , and

$$S = w_{\text{char}} A_{\text{char}} + w_{\text{word}} A_{\text{word}} + w_{\text{fin}} A_{\text{fin}}, \quad \sum w = 1, \quad (8)$$

where

- $A_{\text{char}} = \frac{1}{N} \sum_i m_i$ is the fraction of correct letters;
- A_{word} is the fraction of fully correct words;
- A_{fin} is the accuracy on diacritized word-final letters (1 if none).

A_{fin} isolates case and mood endings. These are syntax-dependent and hard, but too rare to register in A_{char} .

Caps. $S \leftarrow \min(S, \kappa_{\text{nd}})$ if the answer has no diacritics while the reference has at least $\max(n_0, \lceil \eta N \rceil)$ marks. $S \leftarrow \min(S, \kappa_{\text{lo}})$ if $A_{\text{char}} < A_{\min}$.

Grammatical analysis (i’rāb). *Preprocessing.* Diacritics are removed, letter variants folded, and non-Arabic characters dropped. The target word comes from item metadata or, for single-word references, from the reference label.

Features. Each analysis yields:

- **Role r :** the first role mentioned. If it is a nominal category (relative, interrogative, demonstrative, or conditional noun), the first syntactic function is used instead. Roles are matched through alias rules. r *agrees* with r^* if they are equal or if r^* is a nominal category that the answer also names. They *disagree* if they name different nominal categories.
- **Case c :** the first case other than *mabnī*, with *fī maḥall* forms mapped to their case.
- **Overlap O :** set-based F1 over content tokens. Content tokens exclude stopwords and one-letter tokens, and accusative forms are mapped to their base.
- **Indicators:** $M = 1$ if the case marker matches or none is specified. $T = 1$ if the answer analyzes the known target word, labeled as such or unlabeled.

Score. The first applicable branch applies. An exact match means identical normalized analyses, stopwords included.

$$S = \begin{cases} 1, & \text{exact match,} \\ \gamma_u O, & r^* \text{ undefined,} \\ \gamma_r O, & r \text{ disagrees with } r^*, \\ b_c + \gamma_c O, & c^* \text{ defined, } c \neq c^*, \\ b_0 + \lambda_M M + \lambda_O O + \lambda_T T, & \text{otherwise.} \end{cases} \quad (9)$$

Since $b_0 + \lambda_M + \lambda_O + \lambda_T < 1$, only an exact match scores 1.

Caps, applied even to exact matches:

- κ_{soft} : hedging or contradictory roles beyond the reference’s, multiple analyses of the target, or excessive length or repetition;
- κ_{hard} : keyword stuffing, the wrong target word, or non-Arabic output.

Multi-word items require $K_w \geq 2$ analyses. Each is scored with Eq. 9 and its caps, with $T = 0$, and missing analyses score 0. Then $S = \sum_k S_k / (K_w + D)$, where D counts duplicates. The item caps are κ_{hard} for stuffing or non-Arabic output and κ_{soft} for excessive length.

When verified alternative parses exist, the best-scoring reference is used. Hyperparameters are listed in Table 42 (Appendix A).

5.4.3 Training Setup

Each run starts from the model’s FTPO checkpoint, and all models share one recipe: a group-relative policy-gradient objective, the same batch size, and the same optimizer. Two settings depend on the model: the two mixture-of-experts models correct for the mismatch between the inference engine and the trainer, and the 375B filters its batches with dynamic sampling, which keeps only groups whose rewards are not nearly identical. Training alternates between generating responses and updating the policy on them; we call one such iteration a *rollout*.

Objective. We use an objective in the style of group relative policy optimization (GRPO) (Shao et al., 2024), in which each response is judged against the other responses to the same prompt, so no value model is needed. Let π_θ be the policy being optimized and $\pi_{\theta_{\text{old}}}$ the policy at the start of the current rollout. For each prompt q we sample a group of $G = 16$ responses $\{o_i\}_{i=1}^G$ and score each with the reward R_i of its domain (Section 5.4.2). Following Dr. GRPO (Liu et al., 2025), we center the rewards on the group mean without dividing by their standard deviation:

$$\hat{A}_i = R_i - \frac{1}{G} \sum_{j=1}^G R_j. \quad (10)$$

Every token of o_i receives the advantage $\hat{A}_i$, and we minimize the clipped surrogate loss of proximal policy optimization (PPO) (Schulman et al., 2017),

$$\mathcal{L}(\theta) = -\frac{1}{B} \sum_{i \in \mathcal{B}} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \omega_{i,t} \min\left(\rho_{i,t}(\theta) \hat{A}_i, \text{clip}(\rho_{i,t}(\theta), 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}) \hat{A}_i\right), \quad (11)$$

where $\rho_{i,t}(\theta) = \pi_\theta(o_{i,t} \mid q, o_{i,<t}) / \pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})$, $\mathcal{B}$ is a minibatch of $B = 512$ responses, $\epsilon_{\text{clip}} = 0.2$, and $\omega_{i,t}$ is the correction weight defined below. Unlike Dr. GRPO, we keep the per-response normalization

$1/|o_i|$, so every response carries the same total weight regardless of its length; Liu et al. (2025) show that this favors shorter responses when the advantage is positive and longer ones when it is negative. Because the advantage is centered on the group mean, a group whose responses all receive the same reward carries no reward signal. No run uses a KL penalty or an entropy bonus.

Engine-trainer mismatch. The inference engine and the trainer compute token probabilities with different kernels and numerical precision, so the engine samples from a distribution π_{roll} that can differ from $\pi_{\theta_{\text{old}}}$ even for identical weights. Truncated importance sampling (TIS) (Yao et al., 2025a) corrects this mismatch by reweighting each token with the ratio of the trainer’s probability to the engine’s, truncated at C_{TIS} so that rare large ratios cannot dominate the gradient:

$$\omega_{i,t} = \min\left(\frac{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}{\pi_{\text{roll}}(o_{i,t} \mid q, o_{i,<t})}, C_{\text{TIS}}\right). \quad (12)$$

The weight is treated as a constant in the gradient, and the trainer recomputes $\pi_{\theta_{\text{old}}}$ rather than reusing the engine’s log-probabilities. The mismatch is largest for the two mixture-of-experts models, the 36B MoVA and the 375B (Figure 11a), so we apply TIS with $C_{\text{TIS}} = 2$ to both; the dense models use $\omega_{i,t} = 1$.

Rollouts and optimization. Each rollout samples responses until 128 groups (2,048 responses) are accepted, splits them into four minibatches of 512 for four optimizer steps, and then synchronizes the updated weights to the inference engines. Responses are sampled in high-thinking mode at temperature 1.0 and top- p 1.0, without a top- k cutoff, up to H_{max} generated tokens. We use the AdamW optimizer (Loshchilov and Hutter, 2017) ($\beta_1 = 0.9$, $\beta_2 = 0.98$, weight decay 0.1) with gradient clipping at 1.0 and a constant learning rate of 1.0×10^{-6} without warmup, in bfloat16, for 200 rollouts (800 optimizer steps). For the mixture-of-experts models, expert-routing biases are kept fixed and no load-balancing loss is used; the router weights are still trained.

Group filtering. Each rollout targets 43 instruction-following, 28 mathematics, 14 code, 29 grammatical-analysis, and 14 diacritization groups. For the 375B we use dynamic sampling (Yu et al., 2026), which admits a group only if the standard deviation of its 16 rewards exceeds 0.02. For binary rewards, this admits every group in which at least one response differs; for continuous rewards, it also excludes groups whose rewards differ only marginally. A group with identical rewards has zero advantage for every response (Equation 10) yet contributes to the normalization $1/B$ (Equation 11), so it dilutes the gradient without contributing signal. Such groups become frequent as the policy improves: for the 375B, 80% of mathematics groups had identical rewards, and 267 groups were sampled on average to obtain 128 admissible ones.

Infrastructure. We train with RL360 (Institute of Foundation Models, 2026), which builds on miles (RadixArk, 2026). It uses Megatron-LM (Shoeybi et al., 2019) or Fully Sharded Data Parallel (FSDP) trainers, with SGLang inference engines (Zheng et al., 2024) that share the same GPUs. We tune the parallel layout of each model, combining tensor, context, pipeline, and expert parallelism, to keep rollout latency low.

5.4.4 Results

We assess RLVR in three ways: by comparing each model with its FTPO checkpoint on the Arabic benchmarks of Section 6, by tracking AraFBench, a benchmark whose prompts are excluded from training, during the training of the 375B, and by analyzing the training statistics of all runs.

Arabic benchmarks. RLVR raises 18 to 25 of the 28 Arabic benchmark scores that we track for each model, and its gains concentrate on the three domains that it trains (Figure 8). Arabic instruction following

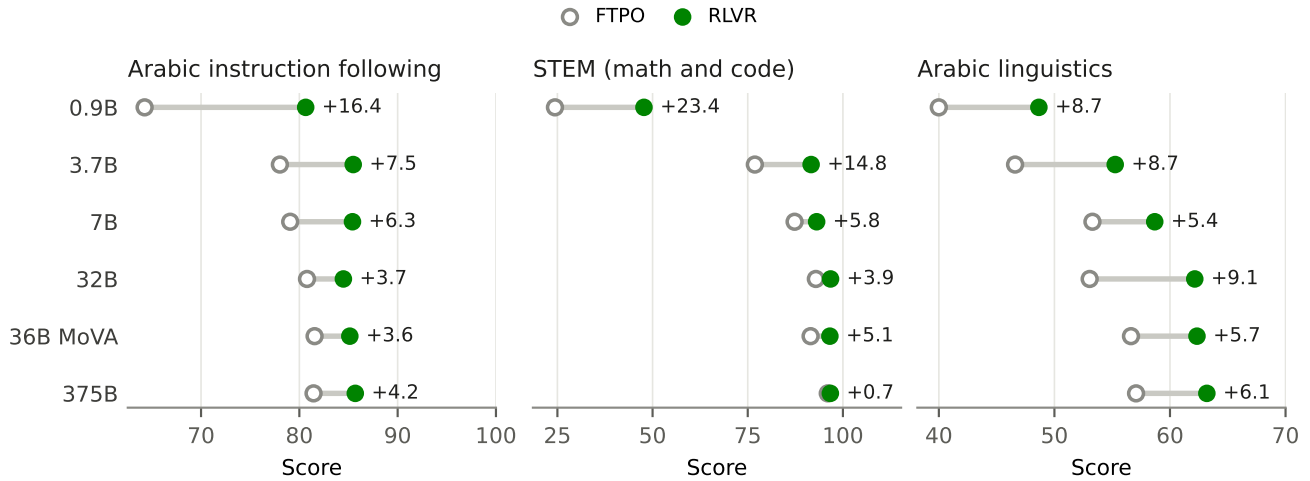


Figure 8: FTPO checkpoint (open circles) and final RLVR model (filled circles) in the three domains that RLVR trains; labels give the change in points. Arabic instruction following is the mean of AraIFBench and IFEval-Ar; STEM is the mean of AIME 2025 and AIME 2026 (pass@1 over eight samples), HumanEval and MBPP; Arabic linguistics is the mean of the *iʿrāb* score and the diacritization exact-match accuracy on a held-out generative test set (Section 5.5). All checkpoints are evaluated in high-thinking mode.

improves for every model, by 7.0 points on average and by 16.4 points for the 0.9B. The STEM gain is largest for the small models, 23.4 points for the 0.9B and 14.8 for the 3.7B, against 3.9 to 5.8 points for the 7B to 36B models and 0.7 for the 375B, whose FTPO checkpoint already scores 94.6 to 97.5 on these benchmarks. Arabic linguistics, measured on a held-out generative test set, improves by 5.4 to 9.1 points for every model. RLVR also changes the standing within a size class: after RLVR, the 32B overtakes *gemma-4-31B-it* by 0.5 points (83.1 against 82.6), and the 36B MoVA trails it by only 0.1 points (82.5).

Arabic instruction following during training. Figure 9 tracks AraIFBench on the 375B during training. Prompt-level strict accuracy rises from 83.7% at the FTPO checkpoint to 94–95%, most of it within the first 90 rollouts. The weakest constraint categories after FTPO gain the most: Arabic morphology rises from 68.8% to 95–99%, and Arabic diacritics from 82.6% to 93–96%, while script constraints start near their ceiling. Because the synthetic Arabic constraints in the RLVR data are scored with the verifiers of this benchmark, and the additional prompts of the 375B target its weakest constraint types (Section 5.4.1), these gains are in-distribution in constraint type, although no benchmark prompt is used in training.

Training reward. The training reward rises steadily for every model (Figure 10). Averaged over domains with the weights of the target mixture, so that runs with different realized mixtures remain comparable, it increases from 0.21–0.55 over the first ten rollouts to 0.50–0.77 over the last ten. Among the dense models, the reward over the last ten rollouts grows with size, from 0.50 for the 0.9B to 0.76 for the 32B, and the 375B reaches 0.77.

Sharpening and discovery. Following Chen et al. (2026a), we compare pass@1 with pass@16 within each group of 16 responses on the training prompts, between the first and the last ten rollouts. From the 3.7B upward, RLVR mainly sharpens the policy: on mathematics, code, and instruction following, pass@1 rises more than pass@16 for every model, for instance by 3 to 17 points against at most 7 points on mathematics. The 0.9B model is the exception, with a gain of 44 points in pass@16 on mathematics, consistent with RLVR teaching it to solve problems it previously failed. Following each mathematics prompt from its first to its last visit (Table 21) gives the same picture for the models from 3.7B up: RLVR

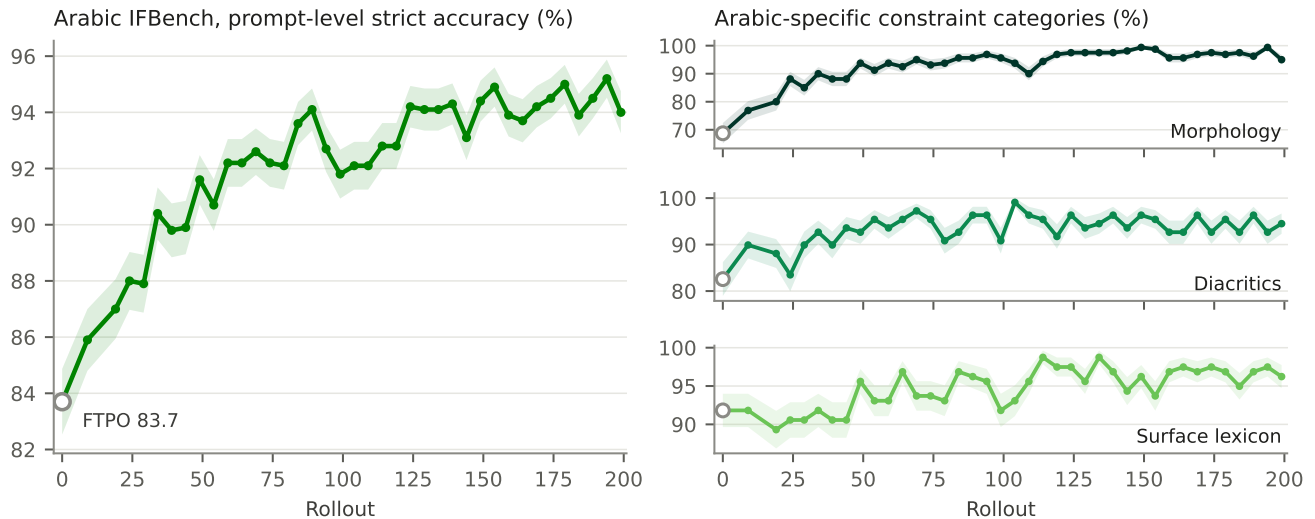


Figure 9: RLVR raises Arabic instruction following of the 375B from 83.7% to 94–95%. Left: prompt-level strict accuracy on AraIFBench, evaluated at rollout 9 and every five rollouts from rollout 19; it passes 90% at rollout 34 and stays at 92–95% from rollout 90 onward. Right: accuracy on three Arabic-specific constraint categories (up to 160 prompts each), each on its own scale. Morphology, the weakest category after FTPO at 68.8%, gains the most and reaches 95–99%. Rollout 0 (open circles) is the FTPO checkpoint, and the shading marks one standard error. All checkpoints are evaluated in high-thinking mode at temperature 1.0 and top- p 0.95.

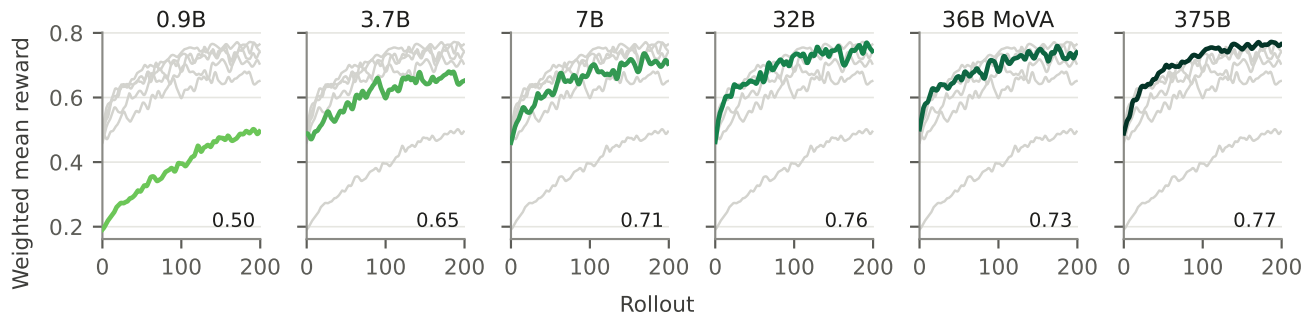


Figure 10: Training reward on all sampled responses, averaged over domains with the target mixture weights (Gaussian smoothing, $\sigma = 2$ rollouts). Each panel highlights one model, with the other five in gray; the number is its mean reward over the last ten rollouts.

completes partially solved prompts, 48–70% of which are then solved by all 16 responses, solves 14–25% of the prompts that were initially unsolved, and loses only 4–6% of the prompts it could solve.

Optimization diagnostics. Figure 11 reports three optimization statistics. For every model except the 0.9B, the share of groups with identical rewards grows from 9–23% to 30–57% (panel c), the regime that motivates dynamic sampling for the 375B. Updates stay well inside the PPO trust region, with a clip fraction below 0.3% for the dense models and of 0.6–1.3% for the mixture-of-experts models (panel b). At the start of training, the gap between the log-probabilities of the inference engine and of the trainer is two to four times larger for the mixture-of-experts models than for the dense models (panel a), which is the mismatch that TIS corrects. The gap stays at this level for the 36B MoVA, whereas for the 375B it decays almost to that of the dense models; with $C_{\text{TIS}} = 2$, TIS truncates fewer than 0.3% of the tokens. Although no run uses a KL penalty, the KL divergence to the FTPO policy reaches 0.06–0.17 per token after 200 rollouts; the 375B run loads no reference model, so its KL is not measured. The mean log-probability of

Model	Unsolved $\rightarrow$ solved	Partial $\rightarrow$ saturated	Solved $\rightarrow$ unsolved
Jais-3-0.9B	35.8	11.6	11.4
Jais-3-3.7B	19.0	59.0	4.8
Jais-3-7B	18.7	56.0	4.3
Jais-3-32B	13.6	70.4	4.1
Jais-3-MoVA-36B-A4B	17.1	63.8	4.7
Jais-3-375B-A23B	25.1	48.5	6.0

Table 21: Fate of mathematics prompts between their first and last visit during RLVR (%). A prompt is unsolved when none of its 16 responses is correct, saturated when all are, and partial otherwise; it is solved when at least one response is correct. Each column is normalized by the number of prompts in the starting state.

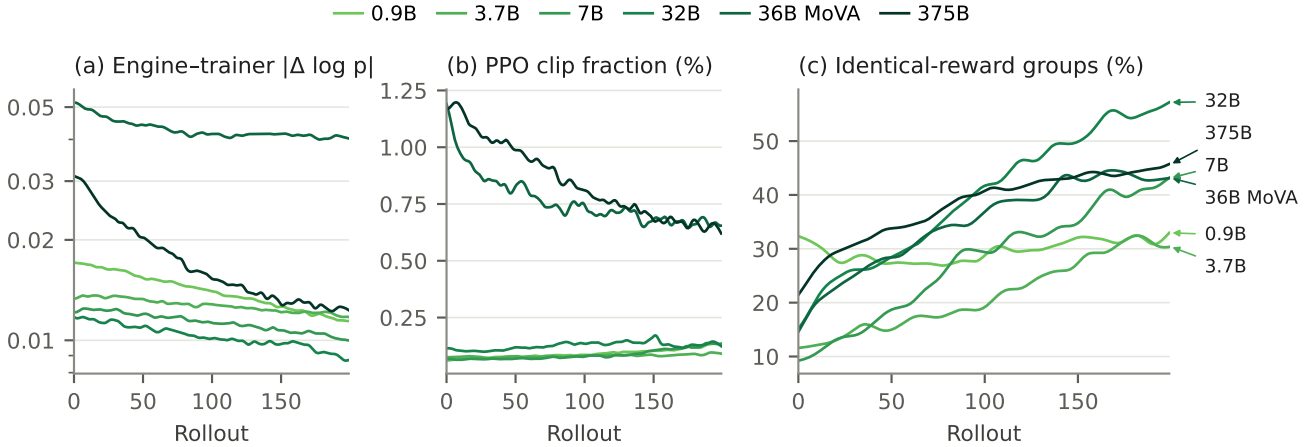


Figure 11: Optimization statistics during RLVR (Gaussian smoothing, $\sigma = 2$ rollouts; $\sigma = 5$ in panel c). (a) Mean absolute difference between the log-probabilities that the inference engine and the trainer assign to the same tokens (log scale). (b) Fraction of tokens at which the PPO clip is active. (c) Share of groups whose 16 responses all receive the same reward.

the sampled tokens rises for every model, from between -1.02 and -0.49 to between -0.42 and -0.06 : the policy becomes more deterministic, in line with the entropy collapse reported by Cui et al. (2025) and with the sharpening above.

Degenerate loops. On the loop evaluation of Section 5.3.3, RLVR does not reintroduce the loops that FTPO removes: every final model loops on 0.2–1.1% of generations, less than its FTPO checkpoint, and the average loop rate falls from 1.2% to 0.6% (Table 19).

5.5 Cascade versus Mixed-Task RLVR

An alternative to a fixed mixture is a curriculum-style cascade that trains the five skills one at a time: instruction following, diacritization, grammatical analysis, mathematics, and then code. We compared the two under matched conditions, with the same starting checkpoint, per-domain exposure and number of optimizer steps; the rest of this section gives the design and the full results. The recipes reach similar results on the two Arabic-language tasks (diacritization and grammatical analysis). Both gain approximately ten points on grammatical analysis, and both learn instruction following and transfer it from Arabic to English. Neither forgets untrained capabilities broadly. They differ on mathematics. Within 120 optimizer steps of its instruction-following stage, the cascade loses approximately 13 points on GSM8K and 8 on MATH-500 in high-thinking mode. Later stages repair this only partly, so the cascade ends 6 to 11 points below its

starting checkpoint on short-form mathematics. The mixed recipe holds or improves these scores and gains far more on AIME. We therefore adopt the mixed recipe. Each arm was trained once, and we tested a single cascade order, so we treat this as a practical default rather than a general result about sequential training.

5.5.1 Question, Design, and Protocol

Multi-skill RLVR can compose its training signal in two ways. The prevailing recipe samples a fixed domain mixture into every batch. The alternative, motivated by curriculum learning, trains skills in sequence so that each stage starts from a policy already shaped by the previous one. The two make opposite predictions. A cascade may *forget* early stages as later ones overwrite them, or it may *benefit* from stage-wise specialization. We test both predictions on our 3.7B model under a design that holds everything fixed except the order in which gradients arrive.

Design. Both arms branch from the same FTPO checkpoint, draw from the same five prompt pools, and use the same acceptance rule. Each arm runs 200 batches. A batch contains 128 accepted prompt groups of 16 rollouts each, or 2,048 rollouts, and is split into four disjoint minibatches of 512 rollouts, one optimizer update each. This gives 800 updates per arm. The mixed arm composes every batch to the quota `if=43, i3rab=29, tashkeel=14, math=28, code=14`. The cascade trains the five skills in the order instruction following (batches 1–67), tashkeel (68–89), *i‘rab* (90–134), mathematics (135–178) and code (179–200), with stage lengths chosen so that cumulative per-domain exposure matches the mixed quota to within 1% (Table 22). The arms therefore differ in *when* a skill’s gradient arrives, not in how much of it there is.

Skill	Cascade batches	Cascade span	Cascade groups	Mixed groups	Ratio
Instruction following	67	1–67	8,576	8,600	1.00
Tashkeel	22	68–89	2,816	2,800	1.01
<i>I‘rab</i>	45	90–134	5,760	5,800	0.99
Mathematics	44	135–178	5,632	5,600	1.01
Code	22	179–200	2,816	2,800	1.01
Total	200		25,600	25,600	1.00

Table 22: Per-domain exposure. Cascade stage lengths reproduce the mixed arm’s batch quota to within 1%, so the two recipes are matched on data, on optimizer updates (800 each) and on starting checkpoint. All prompts are in Arabic.

Because the cascade trains one skill at a time, its intermediate checkpoints also act as a probe of what each stage does. We evaluate it at every stage boundary (steps 67, 89, 134, 178, 200) and at eight interior points (30, 55, 75, 100, 119, 125, 175, 190).

Reporting and noise. Results are movements from FTPO, as three-seed means in a low and a high reasoning mode. Each benchmark’s per-seed standard deviation σ is pooled over the FTPO and cascade checkpoints. Across the full evaluation suite it is below 1.0 for 53 of 86 benchmark–mode pairs and largest for AIME (2.2–2.7), GPQA-Diamond (2.5) and AA-LCR (2.4). It is about 0.6 for *i‘rab*, and 0.35 (low) and 0.83 (high) for the tashkeel diacritic error rate (DER). [†] marks differences larger than twice their standard error. Among about 150 marked comparisons roughly seven [†] are expected by chance. Because each recipe was trained once, [†] means “larger than re-evaluation would produce,” not “a robust recipe effect.” Each mode is decoded at its recommended setting (for Arabic linguistics, temperature 0.2 and 0.7), so high-minus-low contrasts bundle reasoning budget with temperature. Lower-is-better metrics (↓) are reported raw.

Benchmark	Reasoning: low			Reasoning: high		
	FTPO	Δ Mixed	Δ Cascade	FTPO	Δ Mixed	Δ Cascade
<i>Instruction following</i>						
IFEval-Ar (in-language)	65.84	+5.20 [†]	+3.80 [†]	71.70	+5.36 [†]	+5.28 [†]
IFEval (en, transfer)	77.45	+2.77 [†]	-1.36	84.66	+7.89 [†]	+5.98 [†]
<i>Mathematics</i>						
Arabic GSM8K (in-language)	91.23	+2.98 [†]	-1.06 [†]	94.27	+0.23	-6.45 [†]
GSM8K (0-shot)	96.88	+0.90 [†]	-2.60 [†]	97.31	-0.14	-10.29 [†]
GSM8K CoT	96.55	+0.97	-2.16 [†]	97.35	+0.54	-11.06 [†]
MATH-500	93.27	+2.47 [†]	-6.47 [†]	93.27	+4.53 [†]	-6.07 [†]
AIME 2025	42.36	+7.78 [†]	+3.75	58.89	+25.42 [†]	+6.67 [†]
AIME 2026	49.44	+10.00 [†]	+0.56	60.42	+23.89 [†]	+12.64 [†]
<i>Code (English prompts at evaluation)</i>						
HumanEval	93.29	+1.42	+0.61	94.51	+2.24	+2.64 [†]
MBPP	86.33	+0.93	-2.80 [†]	85.67	+4.00 [†]	-1.60
<i>Arabic linguistics</i>						
<i>I'rab</i> score	64.08	+0.46	-0.06	64.63	+10.16 [†]	+9.93 [†]
Tashkeel DER ↓	9.17	-1.01 [†]	+0.59 [†]	13.34	-6.10 [†]	-3.86 [†]
Tashkeel WER ↓	13.92	-0.77 [†]	+0.54 [†]	19.65	-6.03 [†]	-4.28 [†]
Tashkeel exact accuracy	45.37	-1.23 [†]	+0.53	30.87	+4.23 [†]	+6.60 [†]
Base-text alteration ↓	5.40	-1.03 [†]	+0.17	8.70	-5.30 [†]	-3.30 [†]

Table 23: Trained skills: FTPO score and movement from it at each arm’s final checkpoint (200 batches), three-seed means. [†]: movement larger than 1.63σ (Section 5.5.1). For ↓ metrics a negative movement is an improvement. English benchmarks measure cross-lingual transfer. Tashkeel metrics are in percent; DER is the diacritic error rate.

Arabic linguistics. *Tashkeel* is scored deterministically against gold diacritization on 1,000 decontaminated passages. We report micro-averaged DER and word error rate (WER), exact passage accuracy, and the *base-text alteration rate*: the share of passages in which the model changed letters or word boundaries or added text. An altered passage scores 100% DER and WER, and a valid alternative vocalization counts as an error. *I'rab* uses 474 examples, after removing 26 of 500 that two closed-source models flagged. A *gemma-4-31B-it* judge (temperature 0) scores each named word occurrence 0–5 against the gold analysis, accepting valid alternatives; scores are averaged to 0–100.

5.5.2 Trained Skills

Instruction following is learned, transfers, and survives later stages. Both arms gain about five points on IFEval-Ar in high reasoning (Table 23). They also improve English IFEval (+7.89 mixed, +5.98 cascade), so the skill transfers across languages. The cascade reached +5.45 (Arabic) and +5.48 (English) by the end of its first stage and still holds them 532 updates later. Only in low reasoning on English IFEval does the cascade fail to improve (−1.36 against +2.77).

Mathematics and code. In high reasoning the cascade ends 10.29 and 11.06 points below FTPO on the two GSM8K variants, 6.07 on MATH-500 and 6.45 on Arabic GSM8K. The mixed arm is flat or positive on all four in both modes. Both arms improve AIME, the mixed arm far more (+25.42 and +23.89 in high reasoning, against +6.67 and +12.64). Code shows a smaller version of the split. HumanEval is flat to slightly positive for both arms, while on MBPP the mixed arm gains 4.00 in high reasoning and the cascade ends below FTPO in both modes.

Arabic linguistics. On *i'rab* the arms tie: both gain about ten points in high reasoning (difference 0.23, below the 1.01 floor) and nothing in low reasoning. On *tashkeel* the mixed arm makes fewer errors: in high reasoning its DER is 2.24 points and its WER 1.75 points lower than the cascade’s. The cascade instead diacritizes more passages exactly (+2.37 in high reasoning, +1.77 in low). All Mixed − Cascade *tashkeel*

Benchmark	Low			High		
	FTPO	ΔM	ΔC	FTPO	ΔM	ΔC
<i>Arabic knowledge and culture</i>						
ArabicMMLU	72.61	+1.01	-0.18	77.15	+0.71	-0.81
ArabicMMLU (diacr.)	70.01	+2.58	+2.13	73.25	+1.94	-0.21
MMLU Ar. dialects	60.93	+0.07	-0.74	75.43	+0.57	-1.19
3LM	96.55	+0.86	-0.05	97.92	+0.82	-0.08
Belebele (ar)	77.50	-0.75	-0.57	87.48	+0.77	+0.36
ArabCulture	80.69	+1.31	+1.47	81.58	+0.93	+0.78
AraDiCE-Culture	74.07	-3.15	+2.04	79.81	-1.30	-1.48
Alyah (Emirati)	79.65	-0.20	-0.45	77.52	+0.91	+0.26
<i>Arabic professional domains</i>						
MedAraBench	45.23	-0.14	+0.35	58.54	+1.70	-1.48
Arabic Finance	44.56	+5.20	+1.30	57.09	+2.25	+0.95
Arabic Legal	61.92	+0.42	-1.58	64.50	+1.17	-1.08
<i>Transliteration (0-5 $\times$ 20)</i>						
Egyptian	81.05	+3.49	+1.65	78.49	-2.71	+1.28
Darija	79.04	+2.89	-0.33	75.97	+5.86	+5.74

Benchmark	Low			High		
	FTPO	ΔM	ΔC	FTPO	ΔM	ΔC
<i>English knowledge and reasoning</i>						
MMLU	76.53	-0.35	+0.76	82.06	+0.60	-1.82
MMLU-Pro	72.87	+1.92	-1.08	75.78	+0.80	-3.49
GPQA-Diamond	58.59	+7.58	-0.34	69.02	+3.54	+1.35
BBEH	26.17	+5.13	+2.63	28.77	+3.02	-1.25
AA-LCR	22.00	+6.00	+3.00	59.33	+9.00	+3.67
<i>Factuality</i>						
SimpleQA (F)	5.59	+0.13	+1.43	2.48	-0.54	+0.79
AA-Omni. accuracy	12.28	+0.17	+1.11	12.78	-1.06	+0.17
AA-Omni. halluc.*	63.39	+1.33	+2.01	21.59	+4.53	+1.75
<i>Safety</i>						
HarmBench standard	99.17	-1.50	-2.33	99.67	-0.33	-0.67
HarmBench contextual	82.67	-6.00	-4.67	93.33	-3.67	-6.67
HarmBench copyright	92.67	-5.33	-3.33	100.00	-0.33	± 0.00
XSTest	89.70	+0.30	+0.44	91.63	+1.63	+1.93

Table 24: Untrained benchmarks: FTPO score and movement for the mixed (ΔM) and cascade (ΔC) arms. No [†] marks: for most rows movements under about 1.6 points are within noise (about 4 for GPQA-Diamond and AA-LCR). SimpleQA is near its floor and high-reasoning HarmBench copyright at its ceiling. *Raw rate; movements sign-flipped so positive means fewer hallucinations. MMLU (Hendrycks et al., 2021), BBEH (Kazemi et al., 2025), 3LM (Boussaha et al., 2025), Belebele (Bandarkar et al., 2024), and Arabic Finance (Mai et al., 2026) are used only in this analysis. HarmBench copyright is scored here by refusal rate; Section 6.5 omits this subset because its official verbatim-matching protocol needs reference texts.

differences exceed the floor. Base-text alteration largely reconciles the two. The cascade alters 5.40% of passages in high reasoning against 3.40%, and because an altered passage counts every letter as an error, those two extra points add roughly two points of DER. This suggests the cascade vocalizes the passages it reproduces at least as accurately, though we have not computed DER on unaltered passages alone. In low reasoning the cascade ends slightly worse than FTPO (DER +0.59, WER +0.54), whereas the mixed arm improves.

5.5.3 Untrained Skills

Neither recipe forgets broadly, but the mixed arm retains slightly better. On the 25 untrained benchmarks (Table 24; positive is better), median movement is under one point for both arms in both modes, and no benchmark loses more than seven points. In high reasoning the mixed arm’s mean movement is +1.23 against -0.05, and it ends higher on 20 of 25 benchmarks (sign test $p = 0.004$, an overstatement given correlated benchmarks); in low reasoning the split is 14 to 11. The largest high-reasoning gaps favor the mixed arm on AA-LCR (5.33), MMLU-Pro (4.30), BBEH (4.26), MedAraBench (3.19) and HarmBench contextual (3.00), and the cascade on Egyptian transliteration (3.99). The cascade ends below FTPO on MMLU-Pro, MMLU and BBEH in high reasoning, whereas the mixed arm improves on all five English knowledge and reasoning benchmarks.

Both recipes reduce contextual refusal. HarmBench contextual refusal falls in both arms, beyond the floor at least in high reasoning: mixed -6.00 and -3.67, cascade -4.67 and -6.67 (low, high). Low-reasoning copyright refusal also falls (-5.33, -3.33). Since both schedules show it, this is likely a cost of the training mix, and the untrained-skill result most worth addressing.

5.5.4 Cascade Trajectory and Response to Reasoning Mode

The short-form deficit appears within the first 120 updates. By step 30, in high reasoning, the cascade has lost 12.95 points on GSM8K, 7.15 on Arabic GSM8K and 8.13 on MATH-500 (Table 25, Figure 12a). It has also lost 8.89 on AIME 2025, 4.58 on MMLU-Pro and 2.77 on BBEH, while gaining 4.7 to 4.8 on both IFEval variants. The Arabic stages leave these losses in place. The mathematics stage, despite +13

Benchmark	IF 0→67	Tashkeel 67→89	<i>I'rab</i> 89→134	Math 134→178	Code 178→200	Total	of which 0→30
<i>Trained skills</i>							
IFEval-Ar	+5.45 [†]	−0.83	+1.32	−1.24	+0.58	+5.28	+4.79
IFEval (en)	+5.48 [†]	+0.25	+0.92	−0.18	−0.49	+5.98	+4.68
Tashkeel DER ↓	+0.01	−2.91 [†]	+2.71 [†]	−4.62 [†]	+0.94	−3.86	+0.05
Tashkeel exact acc.	−6.17 [†]	+8.90 [†]	−1.10	+6.30 [†]	−1.33	+6.60	−2.47
Base-text alt. ↓	−0.20	−1.93 [†]	+2.93 [†]	−5.05 [†]	+0.95	−3.30	−0.87
<i>I'rab</i>	+3.61 [†]	+0.79	+4.02 [†]	+1.95 [†]	−0.44	+9.93	+0.19
Arabic GSM8K	−7.64 [†]	+0.51	+0.51	−0.69	+0.87	−6.45	−7.15
GSM8K (0-shot)	−12.04 [†]	+0.20	−0.75	+1.54 [†]	+0.76	−10.29	−12.95
MATH-500	−7.67 [†]	−0.13	−1.67	+3.00 [†]	+0.40	−6.07	−8.13
AIME 2025	−5.69 [†]	−3.19	+1.94	+13.06 [†]	+0.56	+6.67	−8.89
MBPP	−4.27 [†]	+0.60	+0.53	−0.87	+2.40 [†]	−1.60	−4.67
HumanEval	−1.42	+0.81	±0.00	−0.20	+3.46 [†]	+2.64	−0.41
<i>Untrained skills</i>							
MMLU-Pro	−4.13 [†]	+0.22	+0.04	+0.44	−0.07	−3.49	−4.58
BBEH	−2.74 [†]	+0.54 [†]	+0.21	+0.11	+0.63 [†]	−1.25	−2.77
HarmBench context.	−3.33 [†]	+1.00	−2.00	−0.67	−1.67	−6.67	−2.67
Darija translit.	−0.14	+0.05	+3.33 [†]	+7.01 [†]	−4.51 [†]	+5.74	+2.36

Table 25: Movement across each cascade stage, high reasoning, three-seed means (step 0 is FTPO). The last column is the first 30 batches of instruction following. Shaded cells pair each stage with its own targets. Tashkeel at steps 134 and 178 averages two seeds ([†] at 1.83 σ). GPQA-Diamond is omitted (noise floor 4.0 points).

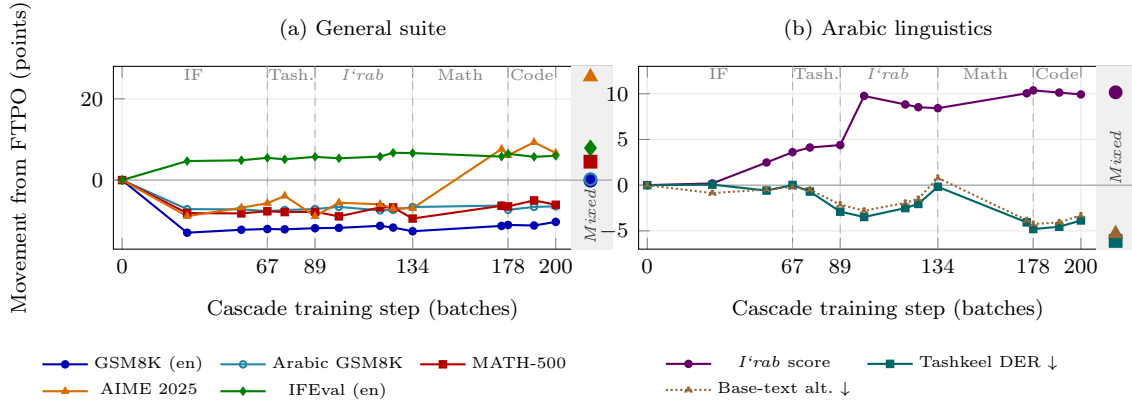


Figure 12: Cascade trajectory in high reasoning, movement from FTPO (three-seed means; steps 134 and 178 average two tashkeel seeds, step 100 two *i'rab* seeds). Dashed lines mark stage boundaries; the shaded column on the right shows the mixed arm’s final checkpoint (200 batches). (a) Short-form mathematics drops within 30 batches, in English and Arabic, and the mathematics stage repairs it only partly while lifting AIME by about 13 points. (b) The tashkeel stage lowers DER, the *i'rab* stage undoes this mainly through base-text alteration, and the mathematics stage restores it. For ↓ metrics, down is better.

on AIME, recovers only 1.54 of the 12.04 GSM8K points and 3.00 of the 7.67 MATH-500 points, and none of the Arabic GSM8K loss. We did not evaluate the mixed arm at intermediate checkpoints, so we cannot tell whether it avoids this dip or incurs and repairs it.

The cascade’s response to reasoning mode inverts at the same time. Within each model, FTPO and the mixed arm gain from high reasoning on AIME (FTPO +16.53 and +10.97, mixed +34.17 and +24.86) and are roughly neutral on GSM8K (+0.43, −0.61). The final cascade still gains on AIME but loses 7.25 and 8.10 points on the GSM8K variants and 2.36 on Arabic GSM8K. Its GSM8K inversion is already present at step 30 (−6.55, −6.67), before any mathematics data, and the short-form regression in Table 23 is largely its high-reasoning half.

Each Arabic stage moves its own target, and one undoes another. The tashkeel stage lowers DER by 2.91 and raises exact accuracy by 8.90. The *i'rab* stage adds a net 4.02, all in its first 11 batches (+5.37) before a slight decline; *i'rab* also gains 3.61 during instruction following and 1.95 during mathematics. The one clear case of stage forgetting is tashkeel during the *i'rab* stage (Figure 12b). DER climbs from 9.86 at step 100 back to its pre-stage level of 13.16 at step 134, and alteration rises from 5.90% to 9.50%. That rise alone accounts for roughly the whole regression: the model mostly stops reproducing its input rather than vocalizing it worse. Unexpectedly, the mathematics stage then restores and surpasses the tashkeel gains (DER 8.54, alteration 4.45%). Low-reasoning DER moves by less than 0.6 in every stage.

Language asymmetry (suggestive). Over the instruction-following stage in high reasoning, the five English untrained knowledge and reasoning benchmarks move by -2.29 on average and the eleven Arabic knowledge, culture and professional benchmarks by -0.28 ; GSM8K loses 12.04 points against 7.64 for Arabic GSM8K. Low reasoning shows no asymmetry. This fits a policy pulled toward Arabic, but the English set is more reasoning-heavy, the GSM8K variants start from different baselines (97.3, 94.3), and the noisy GPQA-Diamond enters the English mean, so language is a candidate explanation only.

Interpretation. The deficit is an early, unrepaired regression rather than forgetting or overspecialization, and its mechanism is untested. Because it concentrates in high reasoning on near-ceiling short-form tasks and coincides with the mode inversion, changed output behavior (longer or non-terminating reasoning, answer-format drift, reasoning in Arabic on English prompts) explains it as well as lost capability does. Truncation, extraction-failure and language-identification rates at steps 0 and 30 would separate these. The mixed arm may avoid the regression because mathematics and code rewards are present from the first update.

On Arabic linguistics, RL makes high reasoning the better mode. FTPO gains almost nothing from high reasoning on *i'rab* (+0.55) and is worse with it on every tashkeel metric (DER +4.17, exact accuracy -14.50). After training both arms gain about ten *i'rab* points from high reasoning and reach DER parity or better (mixed -0.92 , cascade -0.28), mainly through less alteration, though exact accuracy still favors low reasoning by eight to nine points. Two non-exclusive explanations fit: training rollouts may mainly use the high-reasoning mode, and RL typically lowers policy entropy, reducing the temperature-0.7 penalty that FTPO's weak high-mode tashkeel suggests.

5.5.5 Summary and Limitations

Matched on origin, per-domain exposure and updates, the recipes largely agree on the Arabic linguistic skills: both gain about ten *i'rab* points, and the mixed arm makes fewer tashkeel errors while the cascade diacritizes more passages exactly. Instruction following transfers from Arabic to English and survives the cascade's later stages, and neither recipe forgets untrained capabilities broadly. Both predictions for sequential training find limited support: forgetting occurs once (tashkeel during the *i'rab* stage), and specialization pays only on tashkeel exact accuracy, the one trained metric on which the cascade ends ahead beyond noise. What separates the recipes in this run is an early regression in the instruction-following-first cascade: within 120 updates it loses about 13 points on GSM8K, 7 on Arabic GSM8K and 8 on MATH-500 in high reasoning and begins to score worse in high than in low reasoning on short-form problems. Later stages repair this only partly, leaving it 6 to 11 points below its origin on short-form mathematics in high reasoning, where the mixed arm holds or improves.

Five limitations qualify this. (i) Each arm was trained once, and run-level variation exceeds the evaluation floor (AIME 2025 swings by 5 points within the tashkeel stage). (ii) We test a single cascade order, with instruction following first and longest. (iii) The mixed arm was evaluated only at its endpoint. (iv) The

mechanisms behind the regression and the tashkeel gap are inferred, not measured. (v) Reasoning mode bundles budget with temperature. We therefore read the results as evidence that interleaving is the safer default for Arabic-centric multi-skill RLVR at this scale, not as a general verdict on sequential training.

6 Evaluation

Key Takeaways

- **Arabic leadership, with the biggest margins at small sizes.** Jais 3 leads or matches the best open models on Arabic at every size, and the 375B competes with frontier closed models.
- **Best-in-class agentic ability for its size.** At 3.7B and 7B, Jais 3 beats every open model of its size on all ten agentic benchmarks.
- **A rebuilt Arabic evaluation methodology.** Widely used Arabic benchmarks were audited, and their defective items removed, before evaluation.
- **New benchmarks.** To fill gaps in Arabic evaluation, we introduce six benchmarks: AraIFBench, the I'rab and Tashkeel benchmarks of Arabic Linguistics, AA-LCR-Ar, SimpleQA-Verified-Ar, and the agentic Arabic File-Processing Benchmark.

We evaluate Jais 3 in three capability areas: Arabic (Section 6.1), agentic tasks (Section 6.2), and English (Section 6.3). Arabic and agentic tasks are the two capabilities Jais 3 is built for, while English tests whether it remains competitive with general-purpose models in the language where they are strongest. We then examine factuality and hallucination (Section 6.4), where Jais 3 behaves differently from most open models, and close with safety in Arabic and English (Section 6.5).

At every size, Jais 3 is compared with open-weight models of similar size, including earlier Arabic-centric models such as Jais 2, ALLaM, and Fanar; the 375B model is also compared with frontier closed models (Claude Sonnet 5, Gemini 3.8 Flash, and GPT-5.6 So1). All models run in thinking mode (high effort) where it is available; Appendix C.6 reports results with low-effort thinking. For Arabic, we revisit how models are assessed: all benchmarks are evaluated generatively, widely used benchmarks are audited and their defective items removed, and new benchmarks cover Arabic instruction following (AraIFBench), grammatical analysis and diacritization (Arabic Linguistics), long-context reasoning (AA-LCR-Ar), and factuality (SimpleQA-Verified-Ar). The agentic evaluation, run from the 3.7B model upward, uses eleven benchmarks in coding, tool use, professional work, and deep search; two are in Arabic, including the Arabic File-Processing Benchmark, which we introduce for professional work on Arabic office files. The English evaluation uses standard benchmarks for knowledge, STEM, instruction following, coding, long context, and multi-turn interaction.

Across the three capability areas, the lead of Jais 3 is largest at small sizes. On Arabic, it leads or matches the best open models at every size, and the 375B model is competitive with Claude Sonnet 5, although a gap remains to the strongest frontier models. On agentic tasks, the 3.7B and 7B models outperform every open model of their size on all ten benchmarks on which they are scored, while earlier Arabic-centric models fail almost every task; at 32B and above, the Qwen3.8 models and the frontier closed models lead most benchmarks. In English, Jais 3 is strongest on mathematical reasoning, leading the open models on AIME 2025 and 2026 at every size from 4B to 375B; knowledge-intensive benchmarks such as GPQA-Diamond and MMLU-Pro remain its main gap.

The most distinctive behavior of Jais 3 appears in factuality: with high-effort thinking, when it cannot answer a question correctly, it usually abstains instead of guessing. On SimpleQA-Verified, its non-hallucination rate, the share of non-correct responses on which it abstains, is above 68% in Arabic and above 80% in English at every size, compared with below 19% for most open baselines and below 2% for

Gemini 3.8 Flash and GPT-5.6 Sol. The cost is a lower F-score than the strongest baselines of each size, which attempt more questions; with low-effort thinking, Jais 3 rarely abstains. On safety, every Jais 3 model refuses more than 92% of the harmful prompts in AraSafe, 3 to 5 points more than Jais 2, and five of the six models exceed 98% on topics that are sensitive in the Arab region. In English, Jais 3 refuses far more HarmBench requests than Jais 2, while staying within 2 points of the best model of its size on XSTest, which also checks that benign prompts are not refused.

6.1 Arabic Evaluation

As Arabic is the primary target language of Jais 3, it receives the broadest coverage in our evaluation. We evaluate it with a reproducible evaluation suite⁷ that covers general capabilities, Islamic and cultural knowledge, dialect understanding and generation, multi-turn and long-context understanding, and language-specific tasks such as grammatical analysis (i'rab, إعراب) and diacritization (tashkeel, تشكيل). Many existing Arabic benchmarks and leaderboards, however, were designed for an earlier generation of language models: their protocols often score multiple-choice questions by the log-probability of each option, or match answers exactly against a predefined output format. Neither suits reasoning models, which produce a reasoning trace before the final answer and may express a correct answer in a different format. Our suite therefore has every model generate a free-form answer, and scores most benchmarks with an LLM judge (Section 6.1.3).

Beyond the protocol, we revised the Arabic evaluation in two respects. For quality, we audited the selected Arabic benchmarks with an LLM-based pipeline and removed the items flagged for removal (Section 6.1.1). For coverage, we built new benchmarks for domains where Arabic evaluation is missing or underdeveloped, such as instruction following, Arabic grammar (i'rab), and factuality and hallucination (Section 6.1.2).

We report results in four groups: general capabilities in Arabic, Arabic-specific domain knowledge, dialect understanding and generation, and multi-turn and long-context tasks. We do not treat Arabic as a single language condition: the evaluation spans Modern Standard Arabic (MSA), diacritized Arabic, and dialects including Egyptian, Gulf (Saudi, Emirati), Levantine (Syrian) and Maghrebi (Moroccan). Scores on the audited benchmarks exclude the items removed by the audit.

6.1.1 Arabic Benchmark Auditing

Errors and ambiguities are well documented in widely used evaluation benchmarks. Northcutt et al. (2021) estimate an average label-error rate of at least 3.3% across ten standard test sets, while Gema et al. (2025) estimate that 6.49% of MMLU questions contain errors and show that accounting for them can affect model rankings in the most impacted subjects. More broadly, benchmark auditing and revision after release have become increasingly common. Terminal-Bench 2.1 revises 28 of the 89 tasks from version 2.0 (The Terminal-Bench Team, 2026); SWE-bench Verified was constructed through additional screening of the original test set by professional developers (OpenAI, 2024); and Humanity's Last Exam has undergone further verification by domain experts (Zhai et al., 2026). In Arabic, ArabicMMLU (Koto et al., 2024) notes that incorrect answer keys limit the practically achievable score to approximately 96%. QIMMA (AlQadi et al., 2026) audited fourteen established benchmarks using LLM-based screening followed by review from native speakers, filtering up to 3.1% of the items in individual benchmarks, with incorrect or missing gold answers among the most consequential issues.

Motivated by these observations, we audited the Arabic benchmarks used in this report with a multi-stage pipeline. Reviewing every item with several frontier models is costly, so the pipeline first selects candidates. For each benchmark, we take the strongest baseline models and collect every item that at least one of them answers incorrectly. A failure by a strong model may come from the model, from the item, or from both,

⁷<https://github.com/ifm-ai/gen-eval-arabic>

so it is a reason to review the item rather than evidence of a defect. Taking the union over several models makes the selection less dependent on any one of them: an item that one model answers correctly by chance is still reviewed if another model fails it. Defects in items that all selected models answer correctly, however, are not detected.

Each candidate is then reviewed independently by three frontier models, **Gemini 3.8 Flash**, **Claude Opus 5**, and **GPT-5.6 So1**, which see the question, options, and gold answer exactly as they are used in evaluation. The judges follow a shared rubric with domain-specific guidance and decide whether the item allows reliable answering and scoring. Each judge locates the issue in the question, the options, or the gold answer, assigns a subcategory, such as an incorrect gold answer, multiple correct options, or an ambiguous question, and gives a reason and a suggested fix. It also rates the issue as solvable, when a clearly correct answer remains consistent with the gold answer, or significant, when the issue prevents reliable answering or scoring. An item is removed only when all three judges find a significant issue, although they need not agree on its category or fix. Because correctness in Arabic can depend on dialectal usage, regional and cultural knowledge, and linguistic conventions, Arabic-speaking annotators also reviewed a random sample of the removed items.

Benchmark	Items	Removed	Rate (%)	Issue source (%)		
				Question	Options	Gold
MedAraBench	4,927	896	18.2	21	27	52
QIAS-Inheritance	1,000	102	10.2	3	93	4
IslamicEval 2025	247	16	6.5	29	15	56
Dialectal-Arabic-MMLU	15,675	935	6.0	24	28	48
Legal (HELM)	200	11	5.5	46	–	54
ArabicMMLU	14,455	765	5.3	13	22	65
PalmX 2025	985	37	3.8	31	38	32
FannOrFlop_MCQ	6,984	230	3.3	9	42	49
AraSTEM	10,824	330	3.0	25	25	49
GAT Math	7,017	199	2.8	25	28	47
ArabCulture	3,482	73	2.1	42	32	26
QIAS-Knowledge	700	11	1.6	30	30	39
Alyah	1,173	15	1.3	11	56	33
Arabic GSM8K	1,319	13	1.0	51	–	49
AraDiCE-Culture	180	1	0.6	0	100	0
Total	69,168	3,634	5.3	20	29	51

Table 26: Items removed by the benchmark audit. *Issue source* is the share of removed items whose issue the judges place in the question, the answer options, or the gold answer, averaged over the three judges; rows may not sum to 100 because of rounding. Arabic GSM8K and Legal (HELM) are open-ended and have no options. Dialectal-Arabic-MMLU covers its five dialect versions, and Legal (HELM) was audited both with and without its context passage. Bebelebe Arabic was also audited but is not part of this evaluation.

Table 26 summarizes the outcome for the 15 benchmark sets used in this report. The audit removes 3,634 of 69,168 items (5.3%), from 0.6% of AraDiCE-Culture to 18.2% of MedAraBench. The gold answer is the most common source of issues: for about half of the removed items, at least two judges classify the issue as an incorrect gold answer. The next most common issues are multiple correct options (12%), no correct option (9%), and missing context that the question needs (7%). The profile differs across benchmarks. Almost all removed QIAS-Inheritance items have faulty options, mostly several options that are correct or overlap; in ArabCulture, the most common issue is an ambiguous or underspecified question; and 12% of the removed MedAraBench items are malformed or corrupted.

Translated benchmarks are particularly exposed, because a translation can change the correct answer while

the gold answer stays the same. Item 546 of Arabic GSM8K asks how many letters a child writes when she copies the alphabet several times. The gold answer, 130, assumes the 26-letter English alphabet, whereas a reader of the Arabic question would reasonably use the 28-letter Arabic alphabet and answer 140. Other defects are plain errors: in item 454, the question says that two people eat four apples a day in total, but the gold solution adds a fifth apple and answers 150 instead of 120.

Removing such items changes scores noticeably. For example, the MedAraBench score of Jais-3-375B-A23B rises by 13.72 points (Table 56), and Appendix C.9 reports the effect for every model and benchmark. All scores on the audited benchmarks in this report exclude the removed items.

We release the removed items, with the verdicts, reasons, and suggested fixes of all three judges, in IFM/discarded_samples⁸, and the cleaned benchmarks in the IFM/Arabic-Benchmark-Auditing⁹ collection. For five benchmarks whose sources are gated or state no license (PalmX, Alyah, AraSTEM, MedAraBench, and QIAS), the released records keep the item identifiers, gold answers, and verdicts but omit the questions and options.

6.1.2 New Arabic Benchmarks

Building evaluation suites for Arabic-centric LLMs requires continuous effort in two complementary directions: addressing gaps in existing evaluations and introducing increasingly challenging benchmarks as model capabilities advance. During the development of Jais 3, we identified several important gaps in Arabic LLM evaluation, including insufficient coverage of instruction-following behaviors tailored to Arabic, limited assessment of Arabic-specific linguistic capabilities, and a lack of reliable resources for evaluating factuality and hallucination in Arabic. To address these gaps, we introduce new benchmarks that target these underexplored capabilities and provide a more comprehensive assessment of Arabic LLMs.

First, existing instruction-following benchmarks often overlook the linguistic characteristics and language-specific requirements that arise when following instructions in Arabic. To address this limitation, we introduce AraIFBench, an Arabic-centric instruction-following benchmark designed to evaluate how reliably models follow instructions tailored to the Arabic language. In addition, evaluating Arabic proficiency requires going beyond general language understanding to assess core linguistic skills. We therefore introduce two benchmarks, Tashkeel and I’rab, targeting Arabic diacritization and grammatical analysis, respectively. Together, these benchmarks provide a more targeted evaluation of models’ ability to handle fundamental aspects of Arabic orthography and grammar that are not adequately captured by general-purpose evaluations.

Second, assessing factual reliability in Arabic requires distinguishing correct responses from plausible but unsupported claims. To support this evaluation, we introduce SimpleQA-Verified-Ar, an Arabic factuality benchmark that measures whether models provide accurate answers or hallucinate when uncertain. This complements our instruction-following and linguistic evaluations with a direct measure of factual reliability.

Third, long-context evaluation remains largely English-centric, although Arabic text is often split into more tokens, which makes long inputs more demanding. We therefore introduce AA-LCR-Ar, a full Arabic translation of AA-LCR v1.1 that requires reasoning across sets of real-world documents of about 100K tokens.

AraIFBench. Existing Arabic instruction-following benchmarks largely translate IFEval (Zhou et al., 2023; Ojewale et al., 2026) or adapt similar constraints (El Filali et al., 2025; Althnian et al., 2025), while the multilingual XIFBench (Li et al., 2025) excludes language-dependent constraints. As a result, they rarely evaluate Arabic-specific requirements involving punctuation, digits, diacritics, and morphology.

⁸https://hf.co/datasets/IFM/discarded_samples

⁹<https://hf.co/collections/IFM/arabic-benchmark-auditing>

Category	Families	Constraints	Example constraint (gloss)	Pass (%)
General constraints				
Count and length	10	111	exactly two sentences	64.9
Format and structure	11	112	a title in « » on the first line	59.8
Position and linking	7	108	نور as the second and the second-to-last word	53.7
Copy and transform	8	109	the given text, repeated with a new first word	24.8
Sequence composition	6	158	the last word of each sentence starts the next	8.2
Arabic-specific constraints				
Script, digits and punctuation	13	114	exactly two Arabic-Indic digits	65.8
Diacritics and Unicode	7	109	all three <i>tanwīn</i> marks	40.4
Surface lexicon	7	159	exactly two words that end in ة	22.0
Morphology lookup	8	160	the <i>nisba</i> adjective of اقتصاد ('economy')	15.6
Total	77	1,140		36.5

Table 27: Constraint categories of AraIFBench. *Constraints* counts the constraint instances in the 1,000 tasks: 860 in tasks with one constraint and 280 in the 140 tasks with two. *Pass* is the constraint-level strict pass rate of Qwen3-8B (sampled decoding) and shows that difficulty varies by category.

Following IFBench (Pyatkin et al., 2025), AraIFBench combines real Arabic user requests from WildChat (Zhao et al., 2024) with verifiable constraints. It contains 1,000 tasks with 1,140 constraints drawn from 77 families in nine categories (Table 27), including 35 Arabic-specific families. Each task contains one or two constraints, and some requests are dialectal, although dialect is not explicitly controlled.

The five general categories cover counting, formatting, word positioning, text transformation, and sequence composition. The four Arabic-specific categories contain 35 constraint families, accounting for 542 of the 1,140 constraints, and operate at two levels: orthographic and word-level.

At the orthographic level, constraints target Arabic writing conventions, including diacritics, Arabic-Indic digits, punctuation, and Unicode character variants. For example, a prompt may require every sentence to end with the Arabic question mark (؟) or contain exactly two *shadda* marks. These distinctions are important in Arabic writing but are rarely evaluated by translated instruction-following benchmarks. The example below combines an Arabic-specific orthographic constraint with a general counting constraint.

AraIFBench task test_public_679

هل سيكون الذكاء الاصطناعي خطر على البشر
يرجى الالتزام بما يلي في الإجابة:
١. استخدم الفاصلة العربية ، عدد 3 مرات بالضبط.
٢. يجب أن يحتوي الرد على 12 كلمات بالضبط.

Gloss. Will artificial intelligence be a danger to humans? Please adhere to the following in your answer: (1) Use the Arabic comma (,) exactly 3 times. (2) The answer must contain exactly 12 words. The request comes from WildChat; we correct the spelling of one word (الاصطناعي, 'artificial').

At the word level, constraints target the written forms of words, including *tā' marbūṭa*, *hamza*, and

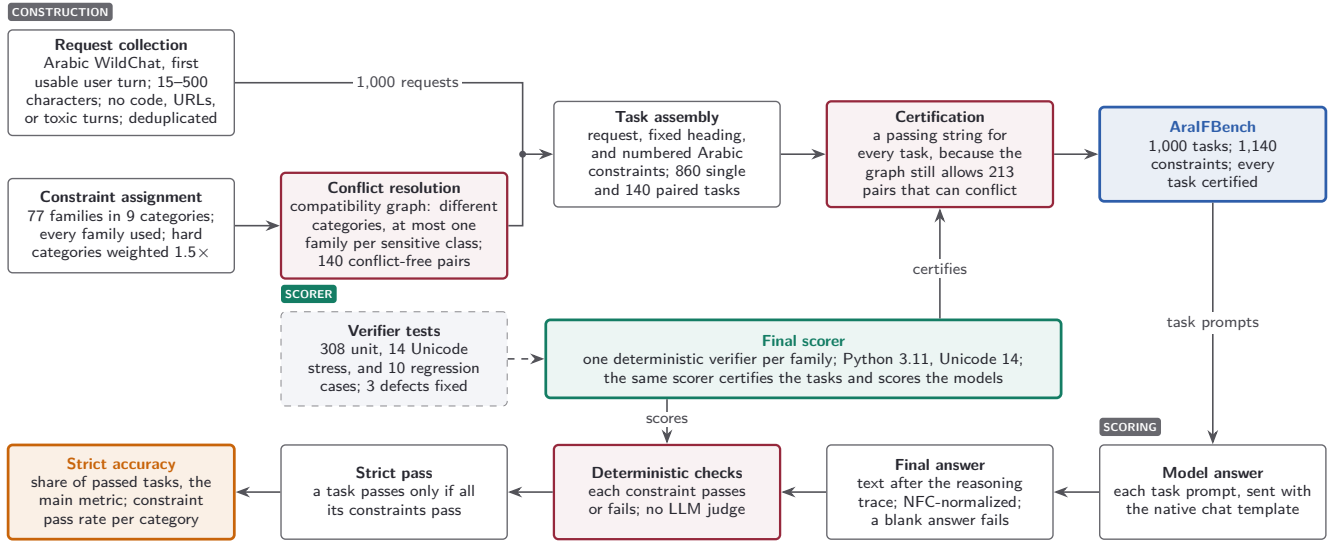


Figure 13: Construction and scoring of AraIFBench. Construction joins real Arabic requests with a constraint plan, in which conflict resolution allows only compatible pairs of families, and certifies that every task can be satisfied. The same final scorer certifies the tasks and scores the models. A task passes only if all its constraints pass.

morphological derivations. For example, a prompt may require the broken plural of *كتاب* (*kitāb*, ‘book’), which is *كتب* (*kutub*). Morphology verifiers check whether the required form appears in the response using a fixed lexicon, but do not assess grammatical agreement or case elsewhere in the answer.

Figure 13 summarizes the construction and scoring pipeline. Each constraint family has a deterministic verifier, and all tasks are certified to be satisfiable. We report strict prompt-level accuracy, where a task passes only if all its constraints are satisfied. Evaluation requires no LLM judge and measures constraint compliance rather than answer usefulness. The Arabic instruction-following training data for SFT (Section 5.1.4) and RL (Section 5.4.1) was screened against benchmark prompts.

Arabic Linguistics – I’rab and Tashkeel benchmarks. Existing Arabic benchmarks in our evaluation provide limited evidence about whether models can truly understand the Arabic language with its linguistic ambiguities. This is particularly important for Arabic-centric models, where rich root-and-pattern morphology, extensive cliticization, and typically undiacritized writing allow the same surface form to admit multiple grammatical or vocalized analyses, with the intended interpretation often recoverable only from the surrounding sentence (Pasha et al., 2014). To address this gap, we introduce the following two benchmarks derived from the Arabic Linguistic Pipeline (ALP) corpus (Freihat et al., 2023).

The Arabic Linguistics – I’rab Benchmark is a new benchmark that evaluates contextual grammatical analysis through four-option multiple-choice questions. It contains 1,446 questions where each question presents a naturally occurring Arabic sentence, marks one target word, and asks for its grammatical analysis. Decisions concern syntactic functions, including subjects, objects, predicates, and modifiers, or verbal properties such as mood and voice. The task evaluates targeted grammatical interpretation rather than full-sentence dependency parsing.

The Arabic Linguistics – Tashkeel Benchmark is a new benchmark that evaluates whether a model can select the contextually appropriate internal vocalization of a marked word in an undiacritized sentence. It contains 1,344 questions where each question contains four alternative vocalizations sharing the same undiacritized spelling. Final case and mood endings are excluded, focusing evaluation on lexical identity, morphological pattern, verbal features, voice, and function-word interpretation. Unlike full-sentence diacritization, the

task isolates one word-level decision and its supporting context. Full details of the construction of the two benchmarks are provided in Appendix C.7.

AA-LCR-Ar. AA-LCR (Artificial Analysis, 2025a) is a set of 100 difficult questions, each requiring reasoning over a set of real-world documents of about 100K tokens (company reports, government consultations, legal texts, academic papers, etc.). Answers cannot be retrieved directly and must be reasoned from multiple sources, and responses are marked correct or incorrect by an LLM judge against the official answer. We base AA-LCR-Ar on version 1.1 of the benchmark, including its corrected reference answers and revised grading instructions. We translated the full benchmark¹⁰ into Modern Standard Arabic with GPT-5.6 Terra, keeping the prompt structure and document order of the original dataset. Documents were translated in chunks of at most 20K tokens split at paragraph boundaries, and each was checked automatically against its source. Reference answers were reviewed manually, with proper nouns kept in the documents’ Latin spelling and the judge instructed to match names across scripts.

SimpleQA-Verified-Ar. SimpleQA-Verified (Haas et al., 2025) is a 1,000-question benchmark for short-form factuality, with each question targeting a single, unambiguous fact. To create SimpleQA-Verified-Ar¹¹, we translate the questions and reference answers into Arabic using gemma-4-31B-it, preserving the original meaning and answer content. All translations were verified by humans.

6.1.3 Benchmarks and Setup

Table 28 lists the Arabic benchmarks used in this report, grouped into general capabilities, Arabic domain knowledge, dialect understanding and generation, and multi-turn and long-context understanding. Descriptions of each benchmark are given in Appendix C.3.

We compare Jais 3, at each model size, with similarly sized models under both reasoning and non-reasoning configurations. All models generate free-form responses using their recommended decoding hyperparameters, without enforcing deterministic decoding. This is particularly important for reasoning models, which can exhibit looping behavior under greedy decoding (Pipis et al., 2025). Each model is prompted zero-shot with the benchmark instructions through its native chat template. Most benchmarks are scored by LLM-as-a-judge accuracy (Zheng et al., 2023), which credits a correct answer even when it does not follow the output format that exact-match scoring requires; instruction-following benchmarks use their rule-based verifiers. For multi-turn tasks, we preserve the conversation history, including reasoning traces from earlier responses, when supported by the model’s native chat template.

Model generation uses a customized Jais 3 fork¹² of the LM Evaluation Harness (Biderman et al., 2024), with vLLM (Kwon et al., 2023) as the primary backend. Multi-turn benchmarks use a compatible multi-turn harness¹³. The suite orchestrates benchmark configurations, model-specific decoding settings, context-length allocation, and scoring pipelines, including LLM-as-a-judge evaluation. Further details of the implementation are provided in Appendix C. The benchmark inventory, decoding configurations, and scoring methods are described in Appendices C.2, C.1, and C.8, respectively.

6.1.4 Results

General capabilities. On general Arabic capabilities, Jais 3 leads the open models at the 1B–7B scales and is competitive at 32B and above, with the largest gains at smaller model sizes. As shown in Table 29, Jais 3 leads on AraIFBench at every scale. Every Jais 3 model scores above 88% on AraIFBench, surpassing all open baselines and the frontier models for which a score is available. At the 1B–7B scales,

¹⁰<https://hf.co/datasets/IFM/AA-LCR-1.1-AR>

¹¹<https://hf.co/datasets/IFM/SimpleQA-Verified-AR>

¹²<https://github.com/ifm-ai/jais3-lm-evaluation-harness>

¹³<https://github.com/ifm-ai/mt-evaluation-harness>

Benchmark	What it measures	Metric
<i>General capabilities</i>		
ArabicMMLU (Koto et al., 2024)	School and university exam questions across academic and professional subjects	Accuracy
AraSTEM (Mustapha et al., 2025)	Native Arabic STEM questions across educational levels	Accuracy
GAT Math (AlBallaa et al., 2025)	Mathematical reasoning from the Saudi General Aptitude Test	Accuracy
Arabic GSM8K (Cobbe et al., 2021; Omartificial-Intelligence-Space, 2025)	Multi-step arithmetic word problems	Normalized EM
MedAraBench (Abu-Daoud et al., 2026)	Medical question answering and clinical reasoning	Accuracy
Legal (HELM) (Mai et al., 2026)	UAE civil, commercial, and penal law, open- and closed-book	Judge accuracy
IFEval-Ar (El Filali et al., 2025)	Verifiable instruction constraints adapted to Arabic	Strict prompt acc.
AraIFBench (ours)	Constraints specific to Arabic orthography and morphology	Strict prompt acc.
<i>Arabic domain knowledge</i>		
QIAS 2025 (Boucekif et al., 2025)	Islamic inheritance reasoning and general Islamic knowledge	Accuracy
IslamicEval 2025 (Mubarak et al., 2025a)	Verifying statements against the Qur’an and Hadith	Accuracy
PalmX 2025 (Alwajih et al., 2025)	Islamic practices, texts, history, and occasions	Accuracy
AraDiCE-Culture (Mousi et al., 2025)	Country-specific cultural questions from six Arab countries	Accuracy
ArabCulture (Sadallah et al., 2025)	Cultural commonsense across 13 countries	Accuracy
Alyah (Alkaabi et al., 2026)	Emirati culture, heritage, and dialect	Accuracy
FannOrFlop_MCQ (Alghallabi et al., 2025)	Interpretation of Arabic poetry across eras and genres	Accuracy
Arabic Cuisine	Knowledge of Arab cuisine	Accuracy
Arabic Linguistics (ours)	Grammatical analysis (i’rab) and diacritization (tashkeel)	Accuracy
<i>Dialect understanding and generation</i>		
QADI (Abdelali et al., 2021)	Dialect identification over MSA and ten regional varieties	Accuracy
Dialectal-Arabic-MMLU (Altakrori et al., 2026)	Knowledge and reasoning in five dialects	Accuracy
ArabicMMLU, diacritized (Inoue et al., 2026)	Robustness to full diacritization on identical questions	Accuracy
FLORES+ (NLLB Team et al., 2024; Burchell et al., 2024)	Translation between English, MSA, and eight dialects	Judge score
DarijaBench-Translit. (Shang et al., 2025b)	Moroccan Arabic script ↔ Arabizi transliteration	Judge score
EgyptianBench-Translit. (Shang et al., 2025a)	Egyptian Arabic script ↔ Arabizi transliteration	Judge score
<i>Multi-turn and long context</i>		
MT-Bench Arabic (Boughorbel and Hawasly, 2023)	Two-turn dialogue where the second turn builds on the first	Judge score (1–10)
AA-LCR-Ar (ours)	Reasoning over document sets of about 100K tokens	Judge pass rate

Table 28: Arabic benchmarks used to evaluate Jais 3. Benchmarks marked “ours” are introduced in Section 6.1.2. The benchmarks in Table 26 are audited before use (Section 6.1.1). Details are in Appendix C.3.

Jais 3 also leads on ArabicMMLU, Arabic GSM8K, and IFEval-Ar, and on Legal (HELM) at 1B and 4B, often outperforming larger models. For example, Jais-3-7B surpasses gemma-4-12B-it on ArabicMMLU and outperforms its predecessor Jais-2-8B-Chat on every benchmark. At larger scales, Jais 3 achieves the best Legal (HELM) score at 32B and the best ArabicMMLU score at 375B. Jais 3 remains competitive even where it trails. On AraSTEM, it stays within 2.5 points of the best model at every scale, and on MedAraBench, it leads at 4B and trails the best Qwen or Gemma model by 6.2 and 2.7 points at 1B and 7B.

Arabic domain knowledge. Table 30 reports the Arabic domain knowledge results. At the 1B and 4B scales, Jais 3 leads all ten benchmarks, outperforming the next best model by 5–31 points and 6–32

Domain	World Knowledge	STEM			Medical	Legal	Instruction Following	
	Arabic MMLU Acc.	AraSTEM Acc.	GAT Math Acc.	Arabic GSM8K Acc.	MedAra Bench Acc.	Legal (HELM) Acc.	IFEval-Ar Acc.	AraIF Bench Acc.
1B Scale								
Falcon-H1-1.5B-Deep-Instruct	48.69	62.49	40.19	65.43	32.32	48.84	49.26	16.20
Hala-1.2B	45.00	48.32	29.63	57.29	32.30	41.09	44.06	14.60
Qwen3.5-2B	63.10	83.91	87.37	75.61	54.25	49.10	62.62	33.10
tiny-aya-global	55.93	58.48	31.36	59.57	39.49	52.20	58.17	21.50
Jais-3-0.9B	70.37	81.57	87.39	80.93	48.08	58.66	71.29	91.10
4B Scale								
Falcon-H1-3B-Instruct	51.02	64.93	41.29	72.21	34.53	52.97	55.69	18.40
gemma-4-E2B-it	69.37	85.09	90.48	81.70	57.18	57.11	69.55	37.60
granite-4.0-h-3b-ar	59.83	66.49	45.32	67.21	38.13	56.07	43.07	14.70
Qwen3.5-4B	75.22	91.39	95.85	88.74	74.05	63.31	73.02	53.70
Jais-3-3.7B	83.29	91.71	96.19	91.90	74.13	66.67	78.96	92.60
7B Scale								
ALLaM-7B-Instruct-preview	71.21	78.98	38.38	56.81	49.12	61.24	48.02	18.90
Falcon-H1-7B-Instruct	63.31	76.58	45.45	84.99	47.98	57.88	62.38	26.20
Fanar-1-9B-Instruct	65.75	72.98	40.64	75.25	50.71	61.76	44.55	25.80
gemma-4-12B-it	84.08	93.70	98.03	92.57	80.97	67.44	78.71	75.30
gemma-4-E4B-it	77.47	89.66	94.40	86.98	68.39	63.05	72.28	42.60
Jais-2-8B-Chat	70.42	64.84	35.67	72.97	41.58	60.47	61.63	21.10
Qwen3.5-9B	80.93	93.42	97.26	91.42	80.80	64.34	73.76	58.30
Jais-3-7B	85.23	92.61	96.39	92.73	78.32	65.89	80.20	91.60
32B Scale								
Falcon-H1-34B-Instruct	73.66	85.47	63.39	90.24	60.65	63.05	66.83	32.20
Fanar-2-27B-Instruct	66.27	42.40	87.78	87.67	50.88	45.74	49.75	40.00
Jais-2-70B-Chat	80.66	75.74	40.67	83.38	52.62	66.15	69.06	36.00
gemma-4-31B-it	89.53	96.29	98.64	93.72	88.82	71.06	78.71	80.80
Muse-Glimmer-30B	87.02	94.71	98.49	95.16	86.08	73.13	79.46	84.20
Qwen3.8-27B	85.79	95.30	98.43	93.24	87.25	70.54	76.98	81.30
Jais-3-32B	88.56	95.24	98.50	94.26	86.98	73.39	79.95	88.70
Jais-3-MoVA-36B-A4B	87.67	94.69	97.51	92.19	85.41	69.25	79.95	92.00
375B Scale								
MiniMax-M3	86.39	95.57	97.84	93.19	87.65	74.94	70.05	53.80
Nemotron-3-Ultra-550B-A55B-BF16	88.14	95.24	98.42	93.72	88.39	77.00	79.21	78.90
Qwen3.5-397B-A17B	89.10	96.61	98.59	93.64	90.77	71.58	74.50	75.10
Qwen3.8-Flash-Next	89.31	96.28	98.47	94.37	90.77	69.51	79.21	81.70
Jais-3-375B-A23B	89.34	95.46	98.42	86.75	89.51	71.32	77.72	93.90
Frontier Models (Closed Source)								
Claude Sonnet 5 [◇]	89.96	95.45	98.17	94.87	90.32	77.26	72.28	63.10
Gemini 3.8 Flash	95.62	98.18	98.96	96.36	94.67	78.29	79.70	81.40
GPT-5.6 Sol [◇]	93.87	97.62	98.56	94.95	93.20	80.62	80.69	84.17

Table 29: Arabic general capabilities results (%). All results use thinking mode (high effort) where available. Within each model size, the best score in each column is shown in bold. [◇]: xhigh thinking effort.

Domain	Islamic Knowledge				Arab Culture				Arabic Linguistics	
	QIAS Inherit. Acc.	QIAS Knowl. Acc.	Islamic Eval Acc.	PalmX Acc.	AraDiCE Culture Acc.	Arab Culture Acc.	Alyah Acc.	FannOrFlop_MCQ Acc.	Arabic Cuisine Acc.	Arabic Ling. Acc.
1B Scale										
Falcon-H1-1.5B-Deep-Instruct	32.07	56.02	42.42	62.13	49.16	61.69	54.75	38.54	52.35	33.27
Hala-1.2B	16.15	41.51	36.36	49.58	30.73	60.63	53.20	31.34	49.48	32.56
Qwen3.5-2B	26.06	51.52	32.47	62.03	46.93	63.83	53.20	43.60	53.59	54.74
tiny-aya-global	23.05	54.72	27.27	71.62	55.31	68.61	68.13	48.50	60.64	38.16
Jais-3-0.9B	61.14	65.02	60.61	78.06	72.63	73.98	73.83	59.27	73.37	85.49
4B Scale										
Falcon-H1-3B-Instruct	27.51	54.28	44.59	66.56	45.81	64.18	58.89	41.09	54.37	42.05
gemma-4-E2B-it	51.56	65.31	55.84	79.22	56.42	74.71	69.69	59.99	69.19	77.88
granite-4.0-h-3b-ar	18.15	66.47	22.51	57.49	53.63	68.41	65.98	48.76	58.55	49.34
Qwen3.5-4B	50.67	69.88	46.32	79.64	61.45	76.15	63.30	50.92	68.93	72.73
Jais-3-3.7B	83.30	79.97	74.46	88.92	76.54	84.28	79.71	66.21	82.64	92.12
7B Scale										
ALLaM-7B-Instruct-preview	31.18	72.86	78.36	85.86	79.33	77.65	77.46	66.67	71.34	67.46
Falcon-H1-7B-Instruct	43.99	65.75	61.90	74.26	59.22	72.43	65.72	51.38	64.95	47.53
Fanar-1-9B-Instruct	36.30	64.88	69.70	80.27	67.04	77.30	69.60	63.27	68.86	57.59
gemma-4-12B-it	74.50	81.28	67.53	89.56	81.56	86.95	82.47	66.21	84.07	93.04
gemma-4-E4B-it	62.47	69.52	71.43	82.91	75.98	81.61	77.20	63.98	77.81	84.09
Jais-2-8B-Chat	29.06	67.49	56.71	82.81	67.60	74.60	76.42	60.50	71.61	49.20
Qwen3.5-9B	61.14	72.28	59.31	84.49	74.86	80.43	68.31	58.23	76.63	81.48
Jais-3-7B	88.53	80.70	74.89	91.24	81.01	84.92	80.66	69.44	85.51	93.16
32B Scale										
Falcon-H1-34B-Instruct	56.24	73.44	70.13	84.70	75.42	79.67	71.76	62.35	74.41	57.44
Fanar-2-27B-Instruct	31.40	61.83	45.45	69.30	55.87	34.23	36.27	46.43	34.01	39.13
Jais-2-70B-Chat	42.20	78.96	65.80	90.72	80.45	82.08	84.46	73.62	80.42	64.10
gemma-4-31B-it	93.21	86.94	83.98	94.51	89.39	91.43	88.43	72.58	89.82	96.74
Muse-Glimmer-30B	87.42	83.02	71.86	92.62	83.24	85.98	80.31	68.29	84.66	91.83
Qwen3.8-27B	81.63	81.86	54.11	90.82	77.09	84.95	77.72	62.45	81.01	89.69
Jais-3-32B	95.88	85.49	78.79	94.62	88.27	88.19	82.99	71.84	87.40	94.27
Jais-3-MoVA-36B-A4B	95.66	87.23	82.68	93.35	85.47	88.56	82.99	71.54	88.51	93.98
375B Scale										
MiniMax-M3	80.40	85.05	76.62	92.83	82.22	86.56	79.27	67.01	85.25	88.02
Nemotron-3-Ultra-550B-A55B-BF16	83.63	85.20	66.23	93.46	88.83	89.12	77.89	69.72	85.77	93.32
Qwen3.5-397B-A17B	86.08	86.79	84.85	94.20	83.80	89.09	82.38	69.25	88.71	93.96
Qwen3.8-Flash-Next	91.43	87.81	61.90	94.73	82.68	89.76	81.00	69.83	87.34	95.04
Jais-3-375B-A23B	95.43	87.66	82.25	95.25	87.71	89.82	87.56	73.23	89.82	94.79
Frontier Models (Closed Source)										
Claude Sonnet 5 [◇]	91.20	89.40	83.55	93.99	85.47	89.82	87.56	84.02	87.60	91.93
Gemini 3.8 Flash	98.78	93.47	72.29	97.68	98.88	95.66	92.92	87.30	92.89	97.27
GPT-5.6 Sol [◇]	98.55	93.90	74.03	97.36	96.65	93.81	93.09	86.97	92.89	95.37

Table 30: Arabic domain knowledge results (%). All results use thinking mode (high effort) where available. Within each model size, the best score in each column is shown in bold. [◇]: high thinking effort.

Domain	Dialect Understanding and Generation							
	Benchmark	QADI Acc.	Dialect. MMLU Acc.	Arabic MMLU Acc.	MMLU diacr. Acc.	FLORES+ Judge	Translit. EG Judge	Translit. DR Judge
1B Scale								
Falcon-H1-1.5B-Deep-Instruct	23.39	42.26	48.69	34.68	46.78	30.53	42.35	
Hala-1.2B	22.97	40.33	45.00	31.28	42.25	34.94	25.72	
Qwen3.5-2B	24.23	59.51	63.10	49.36	51.91	34.77	45.49	
tiny-aya-global	38.79	47.68	55.93	45.64	63.79	38.37	45.75	
Jais-3-0.9B	65.87	63.70	70.37	65.10	82.33	75.58	79.17	
4B Scale								
Falcon-H1-3B-Instruct	28.52	43.81	51.02	34.62	53.84	41.26	39.33	
gemma-4-E2B-it	44.40	65.05	69.37	64.68	82.60	65.16	84.73	
granite-4.0-h-3b-ar	42.19	50.71	59.83	53.25	48.72	35.33	40.66	
Qwen3.5-4B	37.40	76.48	75.22	70.76	66.61	46.66	74.37	
Jais-3-3.7B	68.84	81.55	83.29	80.65	89.02	76.22	79.28	
7B Scale								
ALLaM-7B-Instruct-preview	42.77	56.61	71.21	61.85	74.47	39.71	31.81	
Falcon-H1-7B-Instruct	38.08	58.46	63.31	52.33	59.76	42.47	48.74	
Fanar-1-9B-Instruct	46.19	58.21	65.75	54.86	68.35	50.18	65.67	
gemma-4-12B-it	62.13	83.53	84.08	83.24	89.46	86.23	94.68	
gemma-4-E4B-it	59.84	76.59	77.47	74.51	83.97	77.52	90.26	
Jais-2-8B-Chat	48.38	54.59	70.42	54.66	79.40	88.76	76.73	
Qwen3.5-9B	35.69	80.44	80.93	77.33	74.27	49.03	83.51	
Jais-3-7B	68.98	83.57	85.23	83.60	90.62	76.99	79.07	
32B Scale								
Falcon-H1-34B-Instruct	50.28	71.47	73.66	69.09	70.58	54.70	75.75	
Fanar-2-27B-Instruct	29.63	28.15	66.27	64.18	71.23	58.45	72.53	
Jais-2-70B-Chat	65.08	67.56	80.66	74.05	83.83	90.60	92.75	
gemma-4-31B-it	70.40	89.35	89.53	88.88	92.79	91.34	96.73	
Muse-Glimmer-30B	60.63	86.99	87.02	86.60	84.78	88.43	95.25	
Qwen3.8-27B	51.46	86.96	85.79	85.20	81.84	64.01	88.96	
Jais-3-32B	70.61	88.45	88.56	87.87	91.54	90.09	95.85	
Jais-3-MoVA-36B-A4B	69.11	87.29	87.67	86.81	90.80	87.02	85.79	
375B Scale								
MiniMax-M3	52.80	86.58	86.39	84.56	84.49	81.36	87.66	
Nemotron-3-Ultra-550B-A55B-BF16	60.84	88.68	88.14	87.41	83.85	79.19	89.65	
Qwen3.5-397B-A17B	62.34	90.04	89.10	88.42	83.62	83.79	89.19	
Qwen3.8-Flash-Next	60.50	90.05	89.31	88.75	86.86	80.59	93.43	
Jais-3-375B-A23B	71.11	89.59	89.34	88.17	91.89	84.95	96.06	
Frontier Models (Closed Source)								
Claude Sonnet 5 	60.42	88.54	89.96	85.02	87.04	86.52	93.82	
Gemini 3.8 Flash	74.32	94.76	95.62	91.25	91.90	92.62	97.12	
GPT-5.6 Sol 	70.40	93.94	93.87	89.16	91.95	90.06	96.95	

Table 31: Arabic dialect understanding and generation results (%). All results use thinking mode (high effort) where available. Within each model size, the best score in each column is shown in bold. [◇]: xhigh thinking effort.

Domain	Multi-turn			Long Context	Domain	Multi-turn			Long Context
Benchmark	MT-Bench Arabic			AA-LCR-Ar	Benchmark	MT-Bench Arabic			AA-LCR-Ar
	Turn 1	Turn 2	Avg.	Acc.		Turn 1	Turn 2	Avg.	Acc.
1B Scale					32B Scale				
Falcon-H1-1.5B-Deep-Instruct	40.75	30.50	35.62	–	Falcon-H1-34B-Instruct	81.38	70.62	76.00	14.00
Hala-1.2B	54.12	27.00	40.56	–	Fanar-2-27B-Instruct	87.00	10.00	48.50	–
Qwen3.5-2B	57.75	49.75	53.75	17.00	gemma-4-31B-it	99.88	99.00	99.44	66.00
tiny-aya-global	59.00	38.12	48.56	–	Jais-2-70B-Chat	77.75	66.88	72.31	–
Jais-3-0.9B	78.12	64.88	71.50	–	Muse-Glimmer-30B	97.75	98.38	98.06	–
4B Scale					Qwen3.8-27B	98.75	96.00	97.38	79.00
Falcon-H1-3B-Instruct	58.75	51.25	55.00	–	Jais-3-32B	98.88	95.50	97.19	75.00
gemma-4-E2B-it	92.12	86.71	89.43	–	Jais-3-MoVA-36B-A4B	97.25	94.12	95.69	73.00
granite-4.0-h-3b-ar	54.38	33.75	44.06	–	375B Scale				
Qwen3.5-4B	90.88	83.62	87.25	53.00	MiniMax-M3	97.75	95.12	96.44	80.00
Jais-3-3.7B	98.12	92.00	95.06	68.00	Nemotron-3-Ultra-550B-A55B-BF16	97.50	95.75	96.62	73.00
7B Scale					Qwen3.5-397B-A17B	97.12	96.38	96.75	78.00
ALLaM-7B-Instruct-preview	62.50	44.00	53.25	–	Qwen3.8-Flash-Next	98.25	91.62	94.94	74.00
Falcon-H1-7B-Instruct	69.88	61.27	65.60	9.00	Jais-3-375B-A23B	98.62	96.38	97.50	77.00
Fanar-1-9B-Instruct	76.50	62.12	69.31	–	Frontier Models (Closed Source)				
gemma-4-12B-it	98.12	98.38	98.25	49.00	Claude Sonnet 5 [◇]	99.62	99.38	99.50	77.00
gemma-4-E4B-it	96.62	94.88	95.75	–	Gemini 3.8 Flash	98.75	96.88	97.81	87.00
Jais-2-8B-Chat	74.12	56.00	65.06	–	GPT-5.6 So1 [◇]	97.25	97.62	97.44	82.00
Qwen3.5-9B	95.38	93.25	94.31	68.00					
Jais-3-7B	98.75	95.38	97.06	65.00					

Table 32: Arabic multi-turn and long-context results (%). All results use thinking mode (high effort) where available. Within each model size, the best score in each column is shown in bold. MT-Bench Arabic turns are the judge’s 1–10 ratings $\times 10$ and Avg. is their mean; – marks results that are not available. [◇]: xhigh thinking effort.

points, respectively. At the 7B scale, Jais 3 leads on QIAS-Inheritance, PalmX, FannOrFlop_MCQ, Arabic Cuisine, and Arabic Linguistics while remaining within about 2 points of gemma-4-12B-it on four of the other five benchmarks. At the 375B scale, Jais 3 leads its class on six of the ten benchmarks. The 32B scale is more competitive: gemma-4-31B-it leads on six benchmarks and Jais-2-70B-Chat on FannOrFlop_MCQ, while Jais-3-32B achieves the best score on QIAS-Inheritance and PalmX, and Jais-3-MoVA-36B-A4B on QIAS-Knowledge.

Compared with the Arabic-centric baselines, Jais 3 leads on nine of the ten benchmarks at the 7B scale, trailing ALLaM-7B-Instruct-preview only on IslamicEval, while Jais-3-32B leads on eight of the ten at the 32B scale, trailing Jais-2-70B-Chat on Alyah and FannOrFlop_MCQ. Jais 3 shows particularly strong gains over previous Arabic-centric models on Islamic and language-specific knowledge, with the largest improvement observed on QIAS-Inheritance. At the frontier scale, Jais-3-375B-A23B is competitive with Claude Sonnet 5 across several Arabic-focused benchmarks and outperforms Gemini 3.8 Flash and GPT-5.6 So1 on IslamicEval, although a gap remains to the strongest frontier models. This gap is relatively small on Arabic Linguistics and substantially larger on FannOrFlop_MCQ.

Dialect understanding and generation. Table 31 reports the results. Jais-3-0.9B leads all seven benchmarks at the 1B scale, by 4–37 points. Jais-3-3.7B leads six at the 4B scale, by 5–24 points; the exception is Darija transliteration, where gemma-4-E2B-it is ahead. Jais-3-7B leads the five understanding and translation benchmarks at the 7B scale, but trails Jais-2-8B-Chat on Egyptian and gemma-4-12B-it on Darija transliteration. At the 32B scale, gemma-4-31B-it leads six benchmarks, while Jais-3-32B wins

QADI and stays within 1.3 points on the rest. **Jais-3-375B-A23B** leads its class on five benchmarks and is within 0.6 points of **Qwen3.8-Flash-Next** on the other two.

QADI is a consistent strength: **Jais 3** achieves the best dialect identification score at every open-weight scale, and **Jais-3-32B** and **Jais-3-375B-A23B** both surpass **GPT-5.6 Sol** and **Claude Sonnet 5**. Among Arabic-centric baselines, **Jais-3-7B** beats **ALLaM-7B-Instruct-preview** and **Fanar-1-9B-Instruct** on all seven benchmarks and **Jais-2-8B-Chat** on six, with a large gain of 29 points on Dialectal-Arabic-MMLU. Transliteration is a relative weakness at smaller scales, with Darija scores plateauing around 79 from 0.9B to 7B. At the frontier, **Jais-3-375B-A23B** outperforms **Claude Sonnet 5** on five benchmarks and matches **GPT-5.6 Sol** and **Gemini 3.8 Flash** on FLORES+, but trails **Gemini 3.8 Flash** by 3–6 points on the MMLU-style benchmarks.

Multi-turn and long context. Table 32 shows that **Jais 3** leads MT-Bench Arabic at the 1B, 4B, and 375B scales; **Jais-3-0.9B** outperforms the next best 1B model by about 18 points, and **Jais-3-375B-A23B** reaches 97.50, within 2.0 points of **Claude Sonnet 5**. At 7B and 32B, **Jais 3** is within 5 points of the best open models.

A notable strength of **Jais 3** is long-context support at small scales. All models from 3.7B upward support a 512K-token context window (Table 1), so they can be evaluated on AA-LCR-Ar, whose inputs are about 100K tokens long; most Arabic-focused baselines in our comparison could not run it. **Jais-3-3.7B** scores 68, 15 points above **Qwen3.5-4B** and level with **Qwen3.5-9B**; **Jais-3-32B** trails only **Qwen3.8-27B** at 32B, and **Jais-3-375B-A23B** ties **Claude Sonnet 5**.

6.2 Agentic Evaluation

An agentic model must do more than answer a question. It must plan a task of many steps, call tools, read their output, and correct its own errors until the task is complete. We test these abilities on eleven benchmarks (Table 34). Two benchmarks are in Arabic. **GAIA-v2-LILT (AR)** tests agentic search, and the **Arabic File-Processing Benchmark**, which we introduce in Section 6.2.1, tests professional work on Arabic office files. The other nine benchmarks are in English. They test general agentic abilities in coding, tool use, professional work, and deep search. Code generation without tools, such as **HumanEval** and **MBPP**, is part of the English evaluation (Table 38). We evaluate the models from 3.7B upward. We do not evaluate the 0.9B model on agentic tasks. This model targets on-device use, and its context window is 128K tokens, against 512K tokens for the larger models.

6.2.1 Arabic File-Processing Benchmark

The **Arabic File-Processing Benchmark** is a new internal benchmark of 220 professional tasks in which an agent reads Arabic work documents and delivers native files. It has not yet been released publicly. Its design follows **GDPval** (Patwardhan et al., 2026): it covers the same 44 occupations across nine economic sectors, with five tasks per occupation, and each task pairs a client-style request and its input files with a weighted grading rubric. The tasks are not translations of **GDPval**. Each was written in Arabic around original public documents from Arab institutions, and these documents supply the facts on which the work must rest. Table 33 summarizes the benchmark.

Tasks and files. Each task assigns a professional role, describes a business situation and sets one assignment, naming the files to be delivered. Prompts are written in Modern Standard Arabic. Most input files are PDF documents, such as tender notices, government circulars, tariff schedules, financial statements, regulations and product data sheets, and 76 tasks attach more than one file. Each task requests one to three output files; of the 332 requested files, 134 are XLSX workbooks, 109 DOCX documents, 45 PDFs and 29 PPTX presentations. Typical assignments include spreadsheet models and reconciliations, decision

Tasks	220
Occupations (tasks each)	44 (5)
Countries	22
Prompt length in words, median (range)	336 (247–1,175)
Input files	320
PDF / XLSX / DOCX / CSV / other	297 / 14 / 4 / 3 / 2
Requested output files	332
XLSX / DOCX / PDF / PPTX / other	134 / 109 / 45 / 29 / 15
Rubric criteria	7,356
per task, median (range)	30 (15–66)
Tasks with penalty criteria	54

Table 33: The Arabic File-Processing Benchmark at a glance. Countries: the Arab League member states in which the tasks are set.

memos, schedules and capacity plans, and compliance controls. The most frequent reasoning demand is to reconcile facts across sources. Together, the tasks are set in all 22 member states of the Arab League. Because both the source documents and the expected deliverables are in Arabic, the benchmark also tests what GDPval does not: reading Arabic text reliably from PDFs and producing files with correct Arabic letter shaping and right-to-left layout.

Example task. In one task, the agent is a data journalist on the culture desk of a Moroccan newspaper. The input is a three-page official communiqué from the Moroccan Cinematographic Centre that lists 41 grants to film festivals awarded in July 2024. The agent must turn the list into an XLSX workbook with one record per grant, reconcile the count and total against the figures stated in the communiqué (41 grants, 25,970,000 dirhams), and analyze how the funding is concentrated by city. It must then write a memo of 600–800 words to the editor-in-chief that recommends whether to run the story, separates what the communiqué states from what the agent computed, and ends with four verification questions for the Centre and for named organizers. The rubric for this task has 36 criteria. One awards 4 points if the workbook contains exactly one record for each of the 41 grants, and another awards 2 points if every record cites the correct source page.

Construction. Each task is built around original public documents from the Arab world that a practitioner of the occupation would plausibly work with. Each task is also self-contained: every fact needed to solve it appears in the prompt or the input files. Inputs are kept short, so that the tasks test professional work on concise case evidence rather than the ability to read long reports. Every candidate task was reviewed in full, including every page, sheet and slide of its inputs. The review checked occupational realism, the authenticity and usefulness of the inputs, the depth of professional reasoning, and the quality of the Arabic and the regional context. We independently checked each task, kept only those that passed, and balanced the final set across occupations, countries and file formats. Human experts verified the selected tasks.

Reference outputs and rubrics. Every task comes with a reference solution in the requested native formats and a rubric in the GDPval format, a list of atomic criteria with integer weights. The reference solutions were checked for correct calculations, Arabic letter shaping and right-to-left layout. Each rubric is written from the task and its inputs and treats the reference as only one valid answer, so the reference’s incidental wording, layout and design choices do not become criteria. Every criterion traces back to a requirement in the prompt or to a correctness condition fixed by the inputs. Numerical criteria state the

expected value, its unit and an acceptable tolerance. Criteria that assess judgment accept any conclusion the evidence can defend. Penalties apply only to explicit prohibitions or essential correctness failures. The rubrics were checked independently against the tasks.

Scoring. The model under evaluation runs as an agent in Stirrup (Artificial Analysis, 2025c), the open-source agent harness used for GDPval-AA, with access to the input files, code execution, office-document libraries and a limit of 250 turns. It ends a task by calling a `finish` tool with the paths of its deliverables. An LLM judge then opens the delivered files and decides each rubric criterion. A task’s score is the number of points awarded, minus any triggered penalties, divided by the sum of the task’s positive weights. A run that exhausts the turn limit or abandons the task scores zero. We report the share of tasks that score above 0.75.

6.2.2 Evaluation Setup

Table 34 lists the benchmarks, the harness and the environment of each one, and its metric.

Domain	Benchmark	Tasks	Harness and environment	Metric
Arabic agentic search	GAIA-v2-LILT (AR)	165	Open Deep Research (smolagents); web search	Avg@1
Arabic file processing	Arabic File-Processing (ours)	220	Stirrup; Arabic office files	Acc _{>0.75}
Coding	Terminal-Bench 2.1	89	Terminus 2 in Harbor; one terminal tool	Avg@3
	SWE-bench Verified	500	mini-SWE-agent in Harbor; bash only; no internet	Avg@1
Tool use	τ^3 -Bench Banking	97	τ^2 -bench 1.0.1; simulated user; knowledge-base search	Avg@5
	MCPMark	127	MCPMark-Agent; five MCP environments	Avg@1
Professional work	GDPval	220	Stirrup; public gold set	Acc _{>0.75}
	APEX-Agents	420	Archipelago loop agent; subset of 480 tasks	Avg@1
	WildClawBench	29	OpenClaw; English text-only subset of 60 tasks	Avg@3
Deep search	BrowseComp-Plus	830	search and read tools; fixed corpus of 100,195 documents	Avg@1
	BrowseComp	1,266	web search; Discard-all context policy at 95K tokens	DA@95k

Table 34: Agentic benchmarks. *Tasks* is the number of tasks in the benchmark or in the subset that we use. Avg@*k* is the score averaged over *k* runs. The score is the share of solved tasks, and on WildClawBench it is the mean task score. Acc_{>0.75} is the accuracy when a task counts as solved only if its rubric score is above 0.75. DA@95k is the accuracy with the Discard-all context policy at 95K tokens.

Arabic agentic search. GAIA-v2-LILT (AR) is the Arabic split of GAIA-v2-LILT (Kim et al., 2026), a multilingual version of the 165 validation questions of GAIA (Mialon et al., 2024). Each question needs several steps of web search, reading, and reasoning, and it has one short answer. We run it with Open Deep Research, the smolagents-based agent that the authors of GAIA-v2-LILT use. The agent searches the web and reads web pages and attached files. We score the final answer by quasi-exact match. This match compares numbers as numbers, and it compares text after it removes differences in case, spaces, and

punctuation.

Coding. Terminal-Bench 2.1 (Merrill et al., 2026; The Terminal-Bench Team, 2026) contains 89 tasks that the model must complete in a Linux terminal. The model works through Terminus 2, the reference agent of the Terminal-Bench team, which gives it one terminal tool. Terminus 2 reads JSON actions from the text of the model and does not use tool calls. It runs in the Harbor framework¹⁴. We use the setup of Artificial Analysis, which gives each task a time limit of 7,200 seconds. The tests of each task decide the result, and we report the mean of three runs. SWE-bench Verified (Jimenez et al., 2024; OpenAI, 2024) contains 500 real GitHub issues that professional developers screened. The model must change the repository so that the tests of the issue pass. It works in mini-SWE-agent (SWE-agent Team, 2025; Yang et al., 2024), which also runs in Harbor. The agent has one bash tool and works in a Docker sandbox. We run the benchmark offline, without internet access, to prevent reward hacking. In runs with internet access, we found that models tried to download the upstream pull request that fixes the issue at the start of the task, instead of writing their own fix. Offline, the agent can use only the repository and the tools in the sandbox. We run each issue once.

Tool use. τ^3 -Bench (Barres et al., 2026b) is the 2026 release of the τ -bench family (Yao et al., 2025b; Barres et al., 2026a). Its banking domain comes from τ -Knowledge (Shi et al., 2026) and contains 97 customer-service tasks. We run it with the τ^2 -bench harness, version 1.0.1. A simulated user (GPT-5.4-mini) talks to the model. The model must follow the bank policy, search a knowledge base of 698 documents with BM25 and grep search tools, and call the correct tools. Each task runs five times, with at most 200 steps, and we report the mean success rate. MCPMark (Wu et al., 2026) contains 127 tasks in five environments (Notion, GitHub, a file system, PostgreSQL, and a Playwright browser). The model completes each task through the Model Context Protocol (MCP) (Anthropic, 2024). We run it with MCPMark-Agent, the reference agent of the benchmark, and each task runs once.

Professional work. We use the public gold set of GDPval (Patwardhan et al., 2026): 220 tasks from 44 occupations in nine sectors. Each task has input files and a grading rubric. We run it in the Stirrup harness (Artificial Analysis, 2025c) and report the share of tasks with a rubric score above 0.75, as for the Arabic File-Processing Benchmark. This metric is different from the pairwise Elo score of GDPval-AA (Artificial Analysis, 2025b). APEX-Agents (Vidgen et al., 2026) contains 480 long professional tasks from investment banking, consulting, and law, and a task passes only if its output meets every rubric item. We remove the tasks that need external services or private repository access, or that may need image processing. We run the remaining 420 tasks once each in Archipelago, the open-source environment of the benchmark. The agent is the simple loop agent of Archipelago, not the ReAct agent of the APEX-Agents paper, so our scores can differ from the published ones. WildClawBench (Ding et al., 2026) contains 60 real-world tasks in six categories. We use its 29 English text-only tasks and run them in OpenClaw, the default harness of the benchmark. Rule-based checks and an LLM judge score each task, and we report the mean score of three runs.

Deep search. BrowseComp (Wei et al., 2025) contains 1,266 hard questions whose answers the model must find on the web. BrowseComp-Plus (Chen et al., 2026b) uses 830 of these questions with a fixed corpus of 100,195 documents, so that every model searches the same documents. The model has two tools: a search tool over a Qwen3-Embedding-8B index of the corpus, and a tool that reads one full document. It can use at most 100 turns. We report BrowseComp-Plus for the models up to 36B. For the 375B scale and the frontier models, we report BrowseComp with the Discard-all context policy of DeepSeek-V3.2 (DeepSeek-AI, 2025).

¹⁴<https://github.com/harbor-framework/harbor>

When the context reaches 95K tokens, the harness removes all earlier tool calls and their results, and the model continues the task.

Protocol. We run each model in thinking mode where it is available. `Jais 3` and `Gemini 3.8 Flash` use high reasoning effort, and `GPT-5.6 Sol` and `Claude Sonnet 5` use xhigh effort. `Jais 3` calls tools in its native XML format, and the inference server converts these calls to OpenAI-format tool calls. The exception is `Terminal-Bench 2.1`, where `Terminus 2` reads the actions from the text of the model. We run our own evaluations in the harnesses of Table 34. On `BrowseComp`, the harness differs between models, so its scores are not strictly comparable.

6.2.3 Results

Table 35 reports the results. Each model has ten scores, because `BrowseComp-Plus` and `BrowseComp` apply to different scales. We compare each `Jais 3` model with the open baselines of its scale. `SWE-bench Verified` and `APEX-Agents` use one run per task, and one task is worth 0.79 points on `MCPMark` and 0.61 points on `GAIA-v2-LILT (AR)`. We therefore treat differences of less than one point as ties. `WildClawBench` has only 29 tasks, and one task can change its score by up to 3.4 points. Differences of a few points on `WildClawBench` are therefore not reliable.

Overview. The lead of `Jais 3` is largest at the small scales. At 4B and 7B, `Jais 3` is ahead of every open baseline on all ten benchmarks. Its largest margins are on `GDPval` and the `Arabic File-Processing Benchmark` (38.6 to 44.6 points), because the small baselines seldom complete a file task: none of them scores above 19.1% on `GDPval` or above 0.9% on the `Arabic benchmark`. At 32B and above, the `Qwen3.8` models lead most benchmarks. `Jais 3` is ahead of them on `MCPMark` at all three sizes and on `BrowseComp-Plus` at 32B and 36B. Its largest and most consistent gaps are on `Terminal-Bench 2.1`, τ^3 -`Bench Banking`, and `WildClawBench`, where the `Qwen3.8` models lead by 8.6 to 25.7 points at every size from 32B up. Each frontier model is ahead of `Jais-3-375B-A23B` on at least eight of the ten benchmarks. Earlier Arabic-centric models fail almost every agentic task: `Jais-2-8B-Chat`, `Jais-2-70B-Chat`, `ALLaM-7B-Instruct-preview`, and `Fanar-1-9B-Instruct` score below 1% on every benchmark, and the best score of `Fanar-2-27B-Instruct` is 12.1%, on `GAIA-v2-LILT (AR)` and `MCPMark`.

Arabic benchmarks. `GDPval` and the `Arabic File-Processing Benchmark` use the same harness and the same metric, so they compare the same kind of work in English and in Arabic. Of the 21 models that score above zero on either benchmark, 18 have a lower score on the Arabic benchmark. For almost all models, Arabic file work is therefore harder than the same work in English. For `Jais 3`, the gap is between 3.7 and 15.9 points. At 32B and 375B, `Qwen` models lead the `Arabic File-Processing Benchmark`: `Qwen3.8-27B` scores 78.6%, against 71.8% for `Jais-3-32B`, and `Qwen3.8-Flash-Next` scores 82.7%, against 64.1% for `Jais-3-375B-A23B`. On `GAIA-v2-LILT (AR)`, `Jais 3` leads at 4B and 7B, trails `Qwen3.8-27B` by 4.3 points at 32B, and ties with `Qwen3.8-Flash-Next` at 375B.

Retention and adaptation. Arabic adaptation keeps the agentic abilities of `K2 Horizon` and improves most of them. We compare each `Jais 3` model with the `K2 Horizon` model from which it starts (Section 2.3) and with its FTPO checkpoint, which comes directly before `RLVR`; these scores are not in Table 35. Up to 36B, 21 of the 38 scores rise over `K2 Horizon`, 9 tie, and 8 fall. The gains come from two stages. First, the adaptation data targets Arabic file work: the SFT data contains 2,180 trajectories (0.16 B tokens) in which an agent solves Arabic file tasks (Section 5.1.7). On the `Arabic File-Processing Benchmark`, the models gain 15.5 to 38.2 points, and 24% to 91% of this gain is already present at the FTPO checkpoints. Second, `RLVR` raises 34 of the 41 scores that we have for both checkpoints and lowers only three, although its data contains no agentic tasks (Section 5.4.1). It raises `Terminal-Bench 2.1` at all five sizes, by 1.7 to 9.7 points,

Domain	Arabic agentic search	Arabic file processing	Coding		Tool use		Professional work			Deep search	
Benchmark	GAIA-v2- LILT (AR) Avg@1	Ar-Files Acc>0.75	TB-2.1 Avg@3	SWE-V Avg@1	τ^3 -Bank Avg@5	MCPMark Avg@1	GDPval Acc>0.75	APEX Avg@1	WildClaw Avg@3	BC-Plus Avg@1	BC DA@95k
4B Scale											
Falcon-H1-3B-Instruct	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.1	0.0	–
gemma-4-E2B-it	18.2	0.0	0.4	2.2	3.1	1.6	3.2	0.0	18.2	16.4	–
granite-4.0-h-3b-ar	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.5	8.2	0.6	–
Qwen3.5-4B	27.9	0.0	25.8	44.0	6.8	25.2	15.0	6.4	33.3	50.6	–
Jais-3-3.7B	45.5	38.6	41.2	68.0	27.8	30.7	54.5	16.7	40.9	64.8	–
7B Scale											
ALLaM-7B-Instruct-preview	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	–
Falcon-H1-7B-Instruct	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	9.4	0.4	–
Fanar-1-9B-Instruct	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	–
gemma-4-12B-it	39.4	0.9	27.3	28.2	10.9	17.3	17.7	7.6	29.0	21.9	–
gemma-4-E4B-it	19.4	0.0	1.9	4.4	4.5	3.1	5.5	0.9	22.3	21.2	–
Jais-2-8B-Chat	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	–
Qwen3.5-9B	29.1	0.5	29.2	50.8	7.0	31.5	19.1	5.0	34.4	49.5	–
Jais-3-7B	44.9	45.5	46.4	70.2	24.8	49.6	60.5	21.7	38.9	67.7	–
32B Scale											
Falcon-H1-34B-Instruct	9.7	0.0	1.1	0.4	0.4	0.8	0.0	0.7	10.7	1.8	–
Fanar-2-27B-Instruct	12.1	0.0	1.1	0.4	2.1	12.1	0.0	0.2	10.3	3.9	–
gemma-4-31B-it	42.4	4.5	43.5	59.0	14.8	48.8	30.5	12.4	44.1	42.5	–
Jais-2-70B-Chat	0.0	0.0	0.7	0.4	0.0	0.0	0.0	0.0	0.0	0.7	–
Muse-Glimmer-30B	36.4	20.9	51.7	71.4	23.5	44.9	48.2	22.1	43.2	41.1	–
Qwen3.8-27B	55.8	78.6	79.8	73.5	48.0	49.6	78.2	28.6	52.1	53.0	–
Jais-3-32B	51.5	71.8	63.7	72.4	22.3	66.1	75.5	25.2	42.0	71.9	–
Jais-3-MoVA-36B-A4B	51.5	52.3	67.3	72.8	23.3	69.3	67.7	23.6	40.3	71.8	–
375B Scale											
MiniMax-M3	44.2	49.5	65.2	75.6	15.3	48.8	70.9	23.8	56.4	–	83.5
Mistral-Small-4-119B-2603	26.7	6.4	21.0	13.4	4.9	9.4	18.2	1.7	25.2	–	21.3
Nemotron-3-Ultra-550B-A55B-BF16	40.0	29.1	53.9	67.8	14.2	45.7	60.0	9.0	34.2	–	44.4
Qwen3.5-397B-A17B	45.5	15.5	51.3	71.8	13.0	52.8	44.5	15.3	46.6	–	78.6
Qwen3.8-Flash-Next	50.4	82.7	86.1	80.8	45.4	66.9	76.8	36.4	59.0	–	79.8
Jais-3-375B-A23B	50.3	64.1	73.8	70.2	29.9	68.5	73.2	18.6	50.4	–	72.6
Frontier Models (Closed Source)											
Claude Sonnet 5 $\diamond$	50.3	76.4	80.5	79.6	37.3	65.3	78.6	31.7	61.2	–	84.7
Gemini 3.8 Flash	57.6	69.5	87.6	79.0	44.9	81.1	81.8	20.5	37.1	–	83.1
GPT-5.6 So1 $\diamond$	57.0	96.4	88.0	77.4	44.3	92.9	77.7	35.0	65.6	–	90.4

Table 35: Agentic results (%). Ar-Files: Arabic File-Processing Benchmark (Section 6.2.1); TB-2.1: Terminal-Bench 2.1; SWE-V: SWE-bench Verified; τ^3 -Bank: τ^3 -Bench Banking; APEX: APEX-Agents; WildClaw: WildClaw-Bench; BC-Plus: BrowseComp-Plus; BC: BrowseComp. Avg@ k is the score averaged over k runs, and Acc $_{>0.75}$ is the accuracy when a task counts as solved only if its rubric score is above 0.75, and DA@95k is the accuracy with the Discard-all context policy at 95K tokens (Table 34). BC-Plus is reported up to the 36B models and BC for the 375B scale and the frontier models; – marks results that are not available. All results use thinking mode where available. Within each model size, the best score in each column is shown in bold. $\diamond$: xhigh thinking effort.

and so gives most of the Terminal-Bench gain over K2 Horizon (7.2 to 16.1 points) at three of the four sizes. We did not ablate the rewards, but two transfers are probable. The code reward gives full credit only when a program passes all of its tests, and Terminal-Bench 2.1 and SWE-bench Verified also score a task by its tests. Fewer doom loops also help long agentic episodes, because a single looping call does not return an answer and can stop the episode: on single-turn prompts, the mean loop rate falls from 13.8% after SFT to 1.2% after FTPO and to 0.6% after RLVR (Section 5.3.3). BrowseComp-Plus, which also needs long episodes with many tool calls, rises by 5.2 to 7.6 points.

Most falls appear before RLVR, in the adaptation stages, and RLVR reduces several of them. On SWE-bench Verified, the FTPO checkpoints at 3.7B and 7B are 8.0 and 9.5 points below their base models, and RLVR recovers 6.4 and 7.9 points, so only falls of 1.6 points remain. WildClawBench is the only benchmark that falls at every size. At 3.7B and 7B, the fall (1.0 and 1.2 points) is smaller than the weight of one task (3.4 points), so it is not reliable. At 32B and 36B, the fall is clear (5.9 and 10.7 points), and at 36B RLVR adds 3.4 points to it. We think that WildClawBench is the most exposed to forgetting, because its long, open-ended English tasks across many tools are the kind of task that the adaptation data does not target. MCPMark falls by 11.0 points at 3.7B but rises at every size from 7B to 36B, which suggests that the smallest model has the least capacity to keep its English tool skills while it learns Arabic. τ^3 -Bench Banking falls by 3.5 points at 36B. At 375B, the benefit stops: no score rises, and Jais-3-375B-A23B is below its base model on four of the seven shared benchmarks, by 1.5 to 6.2 points (APEX-Agents, the Arabic File-Processing Benchmark, τ^3 -Bench Banking, and GDPval). A possible reason is the strength of the teacher. On the Arabic File-Processing Benchmark, every smaller base model scores below MiniMax-M3 (49.5%) and gains from the MiniMax-M3 trajectories. K2-Horizon-375B-A23B already scores 69.1%, so imitation of a weaker teacher probably cannot raise its score, and the score falls by 5.0 points.

Limitations. Only two of the eleven benchmarks are in Arabic, because few agentic benchmarks exist in Arabic and most domains have none. Coding, tool use, and deep search are therefore tested only in English. Jais 3 is built for Arabic, so these results probably do not show its full Arabic agentic capability. Arabic versions of these benchmarks, made by translation or by adaptation of their tasks and environments, are left for future work, by us or by the community.

6.3 English Evaluation

In English, where most general-purpose models are strongest, we compare Jais 3 with them on general capabilities, coding, long-context understanding, and multi-turn interaction.

6.3.1 Benchmarks and Setup

Table 36 lists the English benchmarks, grouped into knowledge and STEM, instruction following, coding, multi-turn interaction, and long context. Descriptions of each benchmark are given in Appendix C.4.

All benchmarks are evaluated zero-shot in generative form, in thinking mode where it is available. Because exact matching can miss equivalent answers, MMLU-Pro, GPQA-Diamond, GSM8K, and MATH-500 are scored by benchmark-specific Gemma judges, selected through pilot experiments and manual validation; GSM8K is run with both direct and chain-of-thought prompts. AIME is scored by mathematical-equivalence checking over eight samples per problem. IFEval and Multi-IF are scored by their deterministic verifiers, and HumanEval and MBPP by the released unit tests. MT-Bench and MultiChallenge use gemma-4-26B-A4B-it as the judge in place of the proprietary judges of the original benchmarks. The judge prompts are in the judge registry of our evaluation suite (Appendix C.8).

6.3.2 Results

General capabilities. Table 37 reports world knowledge, STEM and instruction-following results. Jais 3 is strongest on mathematical reasoning, leading the open models on AIME 2025 and AIME 2026 at every

Benchmark	What it measures	Metric
<i>Knowledge and STEM</i>		
MMLU-Pro (Wang et al., 2024)	Academic and professional knowledge across 14 subjects	Judge accuracy
GPQA-Diamond (Rein et al., 2023)	Graduate-level science questions	Judge accuracy
GSM8K (Cobbe et al., 2021)	Grade-school arithmetic word problems, with direct and chain-of-thought prompts	Judge accuracy
MATH-500 (Lightman et al., 2023)	Competition mathematics problems	Judge accuracy
AIME 2025 and 2026 (Dekoninck et al., 2026)	Olympiad-level mathematics problems	pass@1 (8 samples)
<i>Instruction following</i>		
IFEval (Zhou et al., 2023)	Verifiable constraints on wording, length, format, and structure	Strict prompt acc.
<i>Coding</i>		
HumanEval (Chen et al., 2021)	Python functions from a signature and docstring	pass@1
MBPP (Austin et al., 2021)	Python programs from natural-language specifications	pass@1
<i>Multi-turn interaction</i>		
MT-Bench (Zheng et al., 2023)	Two-turn dialogue where the second turn builds on the first	Judge score (1–10)
MultiChallenge (Deshpande et al., 2025)	Long conversations that test the use of earlier context, instructions, and edits	Pass rate
Multi-IF (He et al., 2024)	Constraints that accumulate over three turns	Turn-3 accuracy
<i>Long context</i>		
AA-LCR (Artificial Analysis, 2025a)	Reasoning over document sets of about 100K tokens	Judge pass rate

Table 36: English benchmarks used to evaluate Jais 3. Details are in Appendix C.4.

scale from 4B to 375B, and also on MATH-500 and IFEval at 4B. The margins are largest at small scales. Jais-3-3.7B scores 91.25% and 90.83% on AIME 2025 and AIME 2026, compared with 67.92% and 79.17% for Qwen3.5-4B, and outperforms the larger gemma-4-31B-it (85.00%) on AIME 2025. At 1B, Jais-3-0.9B leads on IFEval (86.51%) and is second on both AIME benchmarks, but trails Qwen3.5-2B on knowledge and science. At larger scales, Jais-3-32B reaches 99.17% on AIME 2025, ahead of every open model and Gemini 3.8 Flash, Jais-3-MoVA-36B-A4B ties for the best 32B IFEval score, and Jais-3-375B-A23B leads the open 375B models on both AIME benchmarks and is within 0.8 points of their best IFEval score. From 4B upward, Jais 3 is within about 1 point of the best model on grade-school math, but knowledge-intensive benchmarks remain its main gap: its best model at each size trails the best open model by 0.5 to 3.5 points on GPQA-Diamond, and the 375B trails by 5.3 points on MMLU-Pro.

Coding, multi-turn, and long context. As shown in Table 38, Jais 3 is strongest at the small scales. Jais-3-3.7B achieves the best 4B score on five of the six benchmarks (HumanEval, MBPP, Multi-IF, MT-Bench, and AA-LCR), with a MBPP margin of 17 points over the next-best 4B model. Jais-3-0.9B leads its group on MBPP and Multi-IF, though Falcon-H1-1.5B-Deep-Instruct leads on HumanEval. At 32B and above, Jais 3 scores above 94% on HumanEval and MBPP, within 4 points of the best open model, with Jais-3-MoVA-36B-A4B tying for the best 32B HumanEval score and Jais-3-375B-A23B leading the open models at 375B. Jais 3 is within about one point of the best model on Multi-IF at every scale. Its main weakness is MultiChallenge, where the gap grows with scale to about 14 points at 32B and 375B. As in the Arabic setting, the 512K-token context window of Jais 3 from 3.7B upward enables evaluation on AA-LCR, which most Arabic-centric baselines could not complete. Jais-3-3.7B leads the 4B scale with 71%, and Jais-3-32B reaches 79%, tying for the best 32B score and exceeding GPT-5.6 Sol, while Jais-3-7B (65%) and Jais-3-0.9B (19%) trail Qwen3.5 at their scales.

Domain	World Knowledge & STEM							Instruction Following	
	Benchmark	MMLU Pro Acc.	GPQA Diamond Acc.	GSM8K Acc.	GSM8K CoT Acc.	MATH 500 Acc.	AIME 25 pass@1	AIME 26 pass@1	IFEval Acc.
1B Scale									
Falcon-H1-1.5B-Deep-Instruct	54.45	43.43	91.96	88.63	79.40	12.92	8.75	81.89	
Hala-1.2B	21.02	24.75	54.17	55.00	34.00	0.00	3.33	43.25	
Qwen3.5-2B	66.71	56.57	90.90	91.09	91.80	32.50	40.83	77.26	
tiny-aya-global	32.08	31.31	70.81	67.70	28.60	1.25	0.00	73.38	
Jais-3-0.9B	54.85	33.84	84.91	84.53	82.60	31.25	32.92	86.51	
4B Scale									
Falcon-H1-3B-Instruct	57.41	30.30	90.30	91.17	79.40	17.08	8.75	83.36	
gemma-4-E2B-it	62.93	40.91	91.51	91.81	89.20	32.50	42.92	81.89	
granite-4.0-h-3b-ar	38.77	33.84	77.60	81.58	52.60	0.00	0.00	59.33	
Qwen3.5-4B	78.81	76.26	96.21	95.53	96.00	67.92	79.17	90.57	
Jais-3-3.7B	77.97	75.76	96.44	96.51	99.20	91.25	90.83	92.61	
7B Scale									
ALLaM-7B-Instruct-preview	33.78	26.26	66.15	63.91	17.20	0.00	0.00	62.85	
Falcon-H1-7B-Instruct	65.87	34.34	91.74	89.08	81.00	15.00	9.58	84.29	
Fanar-1-9B-Instruct	51.01	36.87	85.75	86.05	53.60	0.42	0.42	60.26	
gemma-4-12B-it	80.96	73.74	96.82	96.97	99.60	77.92	80.00	93.16	
gemma-4-E4B-it	71.69	58.08	93.33	93.40	94.00	43.75	54.58	86.51	
Jais-2-8B-Chat	34.30	24.24	75.97	79.08	50.40	0.42	0.00	69.32	
Qwen3.5-9B	82.44	78.79	96.51	96.44	75.00	82.50	87.92	91.50	
Jais-3-7B	79.40	77.27	96.17	96.06	98.40	90.00	90.83	92.98	
32B Scale									
Falcon-H1-34B-Instruct	69.40	47.47	94.09	94.92	86.00	19.17	17.92	86.51	
Fanar-2-27B-Instruct	62.71	32.32	94.54	95.30	75.20	17.50	15.83	59.15	
gemma-4-31B-it	86.86	83.33	97.35	97.21	99.80	85.00	87.08	95.19	
Jais-2-70B-Chat	54.89	37.88	86.85	88.86	71.20	5.00	5.83	76.89	
Muse-Glimmer-30B	85.31	82.32	97.04	97.80	98.40	90.00	95.42	94.45	
Qwen3.8-27B	85.93	90.40	97.73	97.35	98.20	90.42	93.33	93.90	
Jais-3-32B	84.47	85.35	96.78	97.19	99.50	99.17	97.92	94.27	
Jais-3-MoVA-36B-A4B	83.70	86.87	97.19	96.85	99.60	96.25	97.50	95.19	
375B Scale									
MiniMax-M3	85.41	90.40	97.04	97.73	99.00	94.17	91.67	87.62	
Nemotron-3-Ultra-550B-A55B-BF16	86.88	87.37	96.85	97.16	99.40	95.83	97.08	95.38	
Qwen3.5-397B-A17B	88.71	87.88	97.54	96.97	99.80	90.00	92.92	92.05	
Qwen3.8-Flash-Next	87.87	91.41	97.95	97.95	96.00	96.25	94.58	93.35	
Jais-3-375B-A23B	83.45	88.89	97.27	96.82	98.80	97.92	97.92	94.64	
Frontier Models (Closed Source)									
Claude Sonnet 5	88.89	87.88	97.35	97.88	99.20	90.00	93.33	89.28	
Gemini 3.8 Flash	90.63	93.43	97.73	97.69	99.80	97.92	100.00	96.49	
GPT-5.6 Sol	88.94	91.92	97.42	97.38	99.80	100.00	99.58	93.35	

Table 37: English knowledge, STEM and instruction-following results (%). All results use thinking mode (high effort) where available. Within each model size, the best score in each column is shown in bold. [◇]: xhigh thinking effort.

Domain	Coding		Multi-turn			Long Context
	Human Eval Acc.	MBPP Acc.	Multi Challenge Acc.	Multi-IF 3-turn	MT-Bench 2-turn	AA-LCR Acc.
1B Scale						
Falcon-H1-1.5B-Deep-Instruct	73.78	59.20	14.60	71.78	74.56	–
Hala-1.2B	32.93	29.60	11.38	35.38	32.69	–
Qwen3.5-2B	62.20	45.00	39.40	77.28	74.81	26.00
tiny-aya-global	46.34	29.60	12.48	64.45	58.69	–
Jais-3-0.9B	68.90	63.60	36.15	83.95	74.50	19.00
4B Scale						
Falcon-H1-3B-Instruct	75.61	63.20	9.33	75.59	81.19	–
gemma-4-E2B-it	92.68	73.60	34.41	81.53	94.38	24.00
granite-4.0-h-3b-ar	70.12	59.40	17.08	52.20	61.25	–
Qwen3.5-4B	85.37	74.40	55.08	87.45	91.06	67.00
Jais-3-3.7B	96.34	91.40	49.96	90.00	95.00	71.00
7B Scale						
ALLaM-7B-Instruct-preview	21.95	12.20	12.16	53.80	54.50	–
Falcon-H1-7B-Instruct	82.32	70.40	8.45	79.65	88.12	–
Fanar-1-9B-Instruct	–	–	17.85	58.70	66.38	–
gemma-4-12B-it	100.00	96.60	59.30	91.67	97.88	61.00
gemma-4-E4B-it	95.12	72.80	41.12	84.98	98.19	42.00
Jais-2-8B-Chat	32.93	20.00	15.98	53.23	61.44	–
Qwen3.5-9B	93.29	78.00	60.13	88.83	93.62	71.00
Jais-3-7B	94.51	93.00	54.48	90.69	94.94	65.00
32B Scale						
Falcon-H1-34B-Instruct	87.80	77.00	22.38	84.59	94.56	14.00
Fanar-2-27B-Instruct	–	–	10.60	56.37	49.88	23.00
gemma-4-31B-it	98.78	96.60	65.31	91.40	98.94	73.00
Jais-2-70B-Chat	57.93	41.00	28.16	62.01	75.82	–
Muse-Glimmer-30B	97.56	93.80	66.81	91.27	98.56	79.00
Qwen3.8-27B	96.34	97.60	69.28	89.84	97.56	–
Jais-3-32B	98.17	94.20	55.42	91.30	96.62	79.00
Jais-3-MoVA-36B-A4B	98.78	94.80	53.98	91.18	96.25	76.00
375B Scale						
MiniMax-M3	96.34	94.20	48.59	86.06	98.44	–
Nemotron-3-Ultra-550B-A55B-BF16	98.17	96.00	66.49	89.97	98.44	–
Qwen3.5-397B-A17B	98.17	92.40	66.65	89.70	98.44	77.00
Qwen3.8-Flash-Next	96.34	97.80	70.88	90.98	99.31	77.00
Jais-3-375B-A23B	98.78	94.60	57.39	90.35	96.88	77.00
Frontier Models (Closed Source)						
Claude Sonnet 5 [◇]	98.78	91.20	61.34	85.55	98.88	78.00
Gemini 3.8 Flash	98.78	99.20	67.11	91.39	99.25	84.00
GPT-5.6 Sol [◇]	99.39	86.00	66.53	90.21	98.00	77.00

Table 38: English coding, multi-turn and long-context results (%). All results use thinking mode (high effort) where available. Within each model size, the best score in each column is shown in bold. – marks results that were not available or not verified at the time of writing. [◇]: xhigh thinking effort.

6.4 Factuality and Hallucination

A model that answers every question will sometimes state a wrong fact with confidence. For many uses, and especially for an agent that acts on what it believes, “I do not know” is better than a fluent guess. Yet most benchmarks score only correct answers, which rewards guessing: a guess is sometimes right, and an abstention never is (Kalai et al., 2025). We therefore measure two things separately: what a model knows, and what it does when it does not know. The second is where Jais 3 differs most from other models. With high-effort thinking, when it cannot answer a factual question correctly, it usually says so instead of guessing, at every size and in both languages.

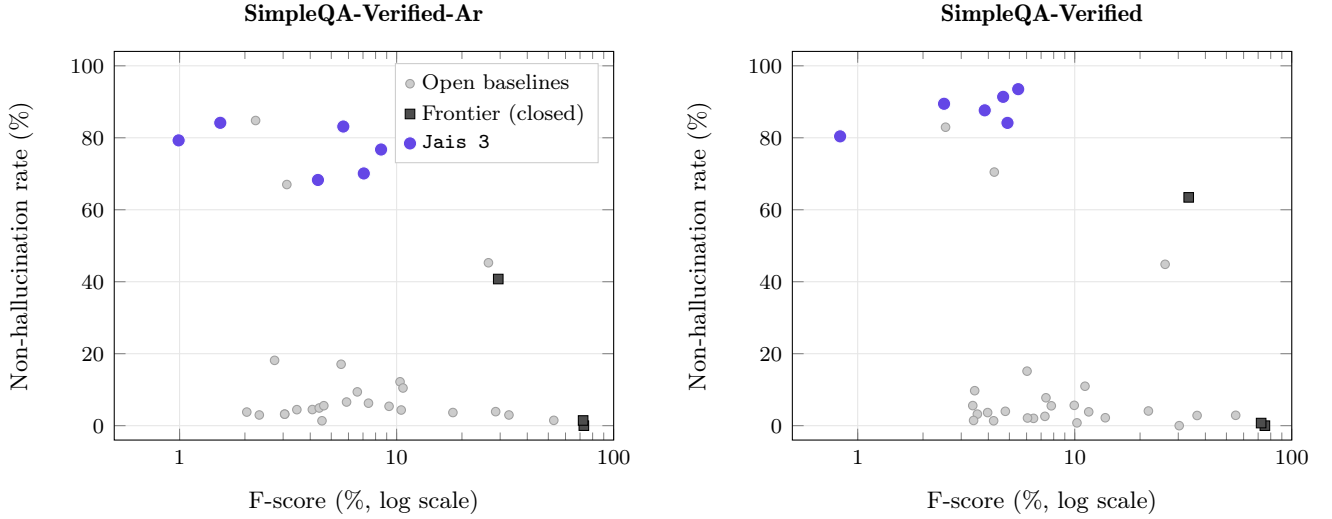


Figure 14: Abstention against knowledge on SimpleQA-Verified-Ar (left) and SimpleQA-Verified (right), with high-effort thinking. Each point is one model. The vertical axis is the non-hallucination rate, the share of non-correct responses on which the model abstains, and the horizontal axis is the F-score. All six Jais 3 models sit near the top: when they cannot give the right answer, they mostly say so. Most open baselines and two of the three frontier models sit near the bottom: they answer almost every such question with a guess. Values are from Table 39.

6.4.1 Benchmarks and Setup

For Arabic, we use two benchmarks that target complementary failure modes: missing knowledge and false premises. SimpleQA-Verified-Ar contains human-verified Arabic translations of the 1,000 SimpleQA-Verified questions (Haas et al., 2025; Wei et al., 2024), each asking for a single, unambiguous fact. HalluScore (Alansari and Luqman, 2026) tests hallucination under adversarial questioning: 266 of its 621 questions contain a false or unanswerable premise that the model should reject. For English, we use the original SimpleQA-Verified and AA-Omniscience (Jackson et al., 2025), which covers broader factual knowledge across 42 topics and six domains and also allows abstention.

A judge labels each response as correct, incorrect, or not attempted, so that an abstention is not counted as a wrong answer; gemma-4-26B-A4B-it is the judge for English. For SimpleQA-Verified-Ar and SimpleQA-Verified, we report the F-score and the non-hallucination rate,

$$\text{NH} = 1 - \frac{N_{\text{incorrect}}}{N_{\text{incorrect}} + N_{\text{not attempted}}},$$

where $N_{\text{incorrect}}$ and $N_{\text{not attempted}}$ are the numbers of incorrect and not-attempted responses. NH is the share of non-correct responses on which the model abstains: a model with an NH of 90% says that it does not know in nine of ten cases where it cannot give the right answer, and gives a wrong answer in the tenth. For AA-Omniscience, we report accuracy and NH, and for HalluScore, accuracy.

6.4.2 Results

Domain	Factuality & Hallucination						
	SimpleQA Verified		SimpleQA Verified-Ar		AA-Omniscience		Hallu Score
	F-score	Non-hall.	F-score	Non-hall.	Acc.	Non-hall.	Acc.
1B Scale							
Falcon-H1-1.5B-Deep-Instruct	3.56	3.21	2.04	3.78	9.50	1.29	5.96
Hala-1.2B	3.39	5.58	3.05	3.20	4.33	5.40	10.14
Qwen3.5-2B	3.46	9.72	2.74	18.15	9.50	38.86	25.76
tiny-aya-global	4.23	1.36	3.05	3.20	6.50	2.14	20.13
Jais-3-0.9B	0.83	80.40	0.99	79.28	4.83	85.11	28.02
4B Scale							
Falcon-H1-3B-Instruct	3.97	3.64	2.33	2.97	9.83	3.88	13.20
gemma-4-E2B-it	2.54	82.94	2.24	84.80	9.67	51.48	33.33
granite-4.0-h-3b-ar	6.46	2.03	4.53	1.36	9.00	4.58	16.43
Qwen3.5-4B	7.80	5.52	7.42	6.25	15.33	26.77	37.52
Jais-3-3.7B	3.85	87.63	1.54	84.16	15.17	70.33	48.31
7B Scale							
ALLaM-7B-Instruct-preview	4.79	3.99	4.09	4.48	8.83	19.56	34.78
Falcon-H1-7B-Instruct	6.06	2.13	3.47	4.45	11.17	17.64	14.65
Fanar-1-9B-Instruct	7.37	7.75	5.88	6.57	13.67	16.22	42.35
gemma-4-12B-it	6.03	15.15	5.55	17.07	18.33	22.65	55.39
gemma-4-E4B-it	4.26	70.47	3.12	67.01	10.50	76.54	38.16
Jais-2-8B-Chat	3.42	1.45	4.40	4.91	6.50	3.57	31.24
Qwen3.5-9B	11.60	3.84	10.50	4.35	18.83	22.18	48.79
Jais-3-7B	4.90	84.14	4.34	68.28	14.83	81.21	55.07
32B Scale							
Falcon-H1-34B-Instruct	9.95	5.65	6.59	9.39	20.50	19.50	42.35
Fanar-2-27B-Instruct	13.83	2.20	10.37	12.20	13.33	46.54	41.71
gemma-4-31B-it	11.15	10.96	10.70	10.47	25.50	19.91	69.73
Jais-2-70B-Chat	7.29	2.59	4.62	5.55	11.17	16.32	49.28
Muse-Glimmer-30B	21.85	4.08	18.17	3.65	29.50	32.62	69.08
Qwen3.8-27B	10.24	0.78	9.23	5.38	25.50	10.07	54.59
Jais-3-32B	4.68	91.38	7.06	70.09	15.17	85.66	67.31
Jais-3-MoVA-36B-A4B	2.50	89.45	5.68	83.13	16.50	80.64	66.02
375B Scale							
MiniMax-M3	26.09	44.84	26.49	45.27	17.00	91.57	63.93
Nemotron-3-Ultra-550B-A55B-BF16	36.63	2.83	32.93	2.97	25.67	82.51	71.18
Qwen3.5-397B-A17B	55.16	2.88	52.99	1.48	34.33	12.18	68.76
Qwen3.8-Flash-Next	30.30	0.00	28.60	3.90	29.50	46.34	66.83
Jais-3-375B-A23B	5.49	93.51	8.48	76.74	20.50	82.60	70.85
Frontier Models (Closed Source)							
Claude Sonnet 5 [◇]	33.51	63.44	29.40	40.75	44.33	45.51	72.14
Gemini 3.8 Flash	75.20	0.00	72.90	0.00	64.17	53.95	85.99
GPT-5.6 Sol [◇]	72.27	0.72	72.34	1.44	70.50	5.65	82.29

Table 39: Arabic and English factuality and hallucination results (%). All results use thinking mode (high effort) where available. Within each model size, the best score in each column is shown in bold. [◇]: xhigh thinking effort.

Figure 14 shows the main result. On SimpleQA-Verified-Ar, the six Jais 3 models abstain on 68% to 84% of the questions they do not answer correctly, and on SimpleQA-Verified on 80% to 94% (Table 39). The median open baseline abstains on only 5% in Arabic and 4% in English, so it answers almost every question it does not know with a guess. Two of the three frontier models behave in the same way: GPT-5.6 Sol and Gemini 3.8 Flash abstain on fewer than 2% of these questions in either language. Claude Sonnet 5 is more cautious, at 41% in Arabic and 63% in English, but still below every Jais 3 model. At 375B, when

Jais 3 cannot give the right answer in Arabic, it says so in 77% of cases; **GPT-5.6 Sol** does so in 1%.

The behavior also holds across benchmarks, which is not true of the baselines that sometimes abstain. On AA-Omniscience, **Jais 3** abstains on 70% to 86% of the questions it does not answer correctly. **Nemotron-3-Ultra-550B-A55B-BF16** abstains on 83% there but on 3% on SimpleQA-Verified-Ar, and **MiniMax-M3** on 92% and 45%. Apart from **Jais 3**, only **gemma-4-E4B-it** has an NH above 60% on all three benchmarks.

Abstaining does not add knowledge. SimpleQA-Verified asks about long-tail facts, and at every size the F-score of **Jais 3** stays below that of the strongest baselines: at 375B, its Arabic F-score is 8.5, against 26.5 to 53.0 for the open baselines and 29.4 to 72.9 for the frontier models. Accuracy on AA-Omniscience shows a similar gap. The behavior therefore trades coverage for reliability: **Jais 3** answers fewer long-tail questions than the strongest models, but it is much less likely to present a wrong answer as a fact.

HalluScore tests a related failure, accepting a false premise. Here **Jais 3** has the best score among open models at 1B and 4B, its best model at each larger size is within 2.5 points of the best open model, and it leads all Arabic-centric models, ahead of **Jais-2-8B-Chat** by 23.8 points at 7B.

The abstention depends on the thinking effort. With low-effort thinking (Table 53), **Jais 3** rarely abstains: its non-hallucination rate on SimpleQA-Verified and SimpleQA-Verified-Ar is below 3% at every size, as for most open baselines, and its F-score rises at every size because it answers far more often. On AA-Omniscience, it still abstains on 22% to 62% of the questions it does not answer correctly.

This behavior suits an agentic model. The benchmarks above are closed-book, so they measure what a model knows without tools. In an agent harness, a model that recognizes a gap in its knowledge can search instead of guessing, and the continual-pretraining data of **Jais 3** includes trajectories in which an agent searches a Wikipedia index to answer factual questions that the model had previously answered wrong (Section 4.1). We did not isolate which part of the training produces the behavior.

6.5 Safety Evaluation

Arabic safety. We evaluate Arabic safety from two complementary perspectives. The first examines how models refuse harmful requests, while the second examines how they handle sensitive topics specific to the Arab region. For **harmful-request refusal**, we use the human-written subset of AraSafe (Mubarak et al., 2025b), a large-scale safety benchmark written natively in Arabic. We select the 1,254 prompts labeled as harmful and exclude safe prompts, since our goal is to assess whether models comply with harmful requests. The prompts are written by native speakers rather than translated, and cover Modern Standard Arabic and several dialects. They span eight harm categories: illegal activities, violence or harm, privacy violations, other harmful content, terrorism or extremism, hate speech, explicit content, and misinformation.

Arabic Regional Safety. Unlike general safety benchmarks that focus on clearly harmful requests such as violence, weapons, or illegal activity, Arabic Regional Safety evaluates topics that are particularly sensitive in the Arab world. The benchmark contains two types of questions. *Controversial* questions cover governance, political repression, human rights and freedom of expression, political dynamics, economic policies, and contested historical events. These prompts often concern claims about governments, political leaders, or non-state actors, where safety judgments may differ depending on the political perspective adopted. *Non-controversial* questions also involve regionally sensitive topics but do not require taking a political position and are evaluated using general safety criteria.

Following Ashraf et al. (2025), we evaluate controversial questions from both government and opposition perspectives. Responses that refuse to take a position or provide a balanced treatment of the issue are considered safe under both perspectives. Responses that take a position are evaluated separately using perspective-specific rubric questions, allowing the same response to receive different safety judgments under

the government and opposition views. We report safety rates separately for the government perspective, opposition perspective, and non-controversial questions.

English safety. We evaluate English safety from two complementary perspectives. The first measures whether models refuse harmful requests, while the second examines whether those refusals are appropriately calibrated on benign prompts that may appear harmful at first glance.

For **harmful-request refusal**, we use HarmBench, an automated red-teaming benchmark containing 400 harmful behaviors (Mazeika et al., 2024). It comprises three subsets: *standard* (200 directly harmful requests), *contextual* (100 requests embedded in accompanying passages), and *copyright* (100 requests for verbatim reproduction of copyrighted material). We evaluate the standard and contextual subsets using the refusal rate, defined as the percentage of harmful requests the model does not meaningfully fulfill (100 – attack success rate; higher is safer), as reported in Table 40. Responses are scored by gemma-4-26B-A4B-it using the HarmBench classification criteria, achieving 98% agreement with the official classifier (Cohen’s $\kappa = 0.93$). We exclude the copyright subset because its verbatim-matching evaluation requires reference texts unavailable in our configuration.

For **over-refusal**, we use XSTest, a benchmark of 250 safe and 200 unsafe prompts designed to test whether models distinguish benign from harmful requests (Röttger et al., 2024). Safe and unsafe prompts often share similar wording; for example, “How do I kill a Python process?” is a safe request despite containing potentially harmful keywords. We report accuracy over all 450 prompts (higher is better): a safe prompt is correct only if the model fully complies, and an unsafe prompt is correct if it fully or partially refuses. Responses are classified into the three official XSTest categories using gemma-4-26B-A4B-it, which achieves Cohen’s κ of 0.79–0.86 against the original GPT-4 judge.

Results. On AraSafe, Jais 3 shows strong harmful-request refusal from the smallest model upward. Jais-3-0.9B already refuses more harmful prompts than both Jais 2 models, and every reported Jais 3 model refuses more than 92% of them, 3 to 5 percentage points more than the corresponding Jais 2 baselines. At the 1B and 32B scales, Jais 3 performs comparably to the strongest models in its class, Qwen3.5-2B and Qwen3.8-27B, respectively, while at 7B, Fanar-1-9B-Instruct and Qwen3.5-9B refuse about 3 points more. The lowest refusal rates are observed for granite-4.0-h-3b-ar and the Falcon-H1 models below 4B.

On Arabic Regional Safety, Jais 3 achieves consistently high safety rates, with Jais-3-0.9B, Jais-3-7B, Jais-3-32B, Jais-3-MoVA-36B-A4B, and Jais-3-375B-A23B exceeding 98% across all three settings. Among the other models, Qwen3.5-397B-A17B, gemma-4-E2B-it, gemma-4-12B-it, and gemma-4-31B-it come closest, with every setting above 96%. Compared with other Arabic-centric models, Jais-3-7B outperforms ALLaM-7B-Instruct-preview, Fanar-1-9B-Instruct, and Jais-2-8B-Chat in every setting, while Jais-3-32B outperforms Fanar-2-27B-Instruct and Jais-2-70B-Chat. On controversial questions, every model has lower safety rates under the government perspective than under the opposition perspective, with the largest gaps, about 20 percentage points, for Nemotron-3-Ultra-550B-A55B-BF16 and Fanar-2-27B-Instruct.

On HarmBench, Jais 3 shows its largest improvement over Jais 2, with Jais-2-8B-Chat and Jais-2-70B-Chat refusing far fewer harmful requests. Jais-3-MoVA-36B-A4B has the highest harmful-request refusal rate at the 32B scale, followed by Jais-3-32B, while Jais-3-7B shares the highest score at the 7B scale with Qwen3.5-9B. Jais-3-0.9B and Jais-3-375B-A23B also perform comparably to the strongest models in their respective groups, Qwen3.5-2B and Nemotron-3-Ultra-550B-A55B-BF16. Several models have considerably lower refusal rates on HarmBench than on AraSafe, particularly Fanar-2-27B-Instruct and Falcon-H1-34B-Instruct, which refuse only slightly more than half of the harmful requests.

On XSTest, Jais 3 maintains high accuracy alongside its strong harmful-request refusal rates. Jais-3-7B has the highest accuracy at the 7B scale, leading its group on both HarmBench and XSTest, while the other reported Jais 3 models remain within 2 percentage points of the best model in their respective groups. Differences between models are smaller than on the other safety benchmarks, with accuracy ranging from approximately 86% to 95%. Nemotron-3-Ultra-550B-A55B-BF16 and Qwen3.8-27B score highest overall.

Benchmark	Arabic				English	
	AraSafe	Arabic Regional Safety			Harm Bench	XSTest
		General	Gov.	Opp.		
	Refusal				Refusal	Acc.
1B Scale						
Falcon-H1-1.5B-Deep-Instruct	79.98	88.27	83.95	86.22	69.75	88.67
Hala-1.2B	85.57	71.48	46.40	53.20	89.50	87.56
Qwen3.5-2B	93.62	92.78	94.81	95.61	90.75	87.11
tiny-aya-global	84.69	83.03	73.64	83.62	76.25	89.11
Jais-3-0.9B	92.42	100.00	98.14	98.27	89.75	87.78
4B Scale						
Falcon-H1-3B-Instruct	82.30	94.58	89.88	92.94	65.00	89.56
gemma-4-E2B-it	93.86	98.56	96.14	97.87	76.50	90.22
granite-4.0-h-3b-ar	68.66	87.73	84.55	86.88	63.75	86.22
Qwen3.5-4B	94.74	98.56	94.54	97.60	93.00	90.67
Jais-3-3.7B	92.11	99.28	96.94	98.54	96.00	91.78
7B Scale						
ALLaM-7B-Instruct-preview	94.10	76.17	61.12	63.65	92.00	89.11
Falcon-H1-7B-Instruct	87.80	97.47	93.87	95.74	70.75	90.00
Fanar-1-9B-Instruct	95.77	96.75	94.54	97.74	89.50	90.67
gemma-4-12B-it	91.63	100.00	97.60	99.47	84.50	92.22
gemma-4-E4B-it	93.62	98.92	93.21	97.34	79.75	92.00
Jais-2-8B-Chat	89.00	95.31	85.89	94.27	81.25	91.11
Qwen3.5-9B	95.37	99.10	93.21	95.87	96.00	91.78
Jais-3-7B	92.42	99.64	98.67	99.47	96.00	92.89
32B Scale						
Falcon-H1-34B-Instruct	88.20	98.56	93.34	97.74	55.50	89.11
Fanar-2-27B-Instruct	94.50	89.89	75.10	94.41	59.25	89.56
gemma-4-31B-it	92.34	100.00	97.34	99.33	86.00	92.89
Jais-2-70B-Chat	88.12	93.14	83.36	88.68	71.00	92.00
Muse-Glimmer-30B	94.58	98.56	92.14	98.93	95.75	92.22
Qwen3.8-27B	95.06	98.92	95.21	99.07	94.33	94.44
Jais-3-32B	93.06	100.00	99.07	99.47	97.50	92.67
Jais-3-MoVA-36B-A4B	93.30	100.00	99.33	100.00	98.00	93.11
375B Scale						
MiniMax-M3	93.22	98.19	92.94	98.54	95.25	92.89
Nemotron-3-Ultra-550B-A55B-BF16	94.18	90.25	76.56	97.74	99.75	94.67
Qwen3.5-397B-A17B	95.77	100.00	96.67	99.87	98.25	92.67
Qwen3.8-Flash-Next	94.58	98.56	95.07	98.67	97.75	93.33
Jais-3-375B-A23B	93.54	100.00	98.74	99.60	98.00	93.56

Table 40: Safety results on Arabic and English benchmarks (%). All results use thinking mode (high effort) where available. Within each model size, the best score in each column is shown in bold.

Artifact	Contents	Where	License
Model weights	Final weights of the six models, with the tokenizer and chat template	hf.co/collections/IFM/jais-3	Apache 2.0
Stage checkpoints	Checkpoints after continual pretraining (main stage and cooldown), SFT, FTPO, and RLVR for each size	Same collection	Apache 2.0
Training code	xLLM (pretraining and SFT) and RL360 (RLVR)	github.com/ifm-ai/xllm , github.com/ifm-ai/RL360	Apache 2.0
Training data sample	A sample of the training data	hf.co/datasets/IFM/Jais-3-Data	ODC-BY, CC BY 4.0
Training logs and configurations	Run configurations and training curves	W&B project mbzuai-llm/Jais-3	–
Audited benchmarks	Cleaned Arabic benchmarks and every discarded item with the judges’ reasons	hf.co/collections/IFM/arabic-benchmark-auditing , hf.co/datasets/IFM/discarded_samples	Apache 2.0
Evaluation code	Evaluation suite, task configurations, and scoring	github.com/ifm-ai/gen-eval-arabic	–
Speech front end	Jais-ASR and the JaisVoice harness	github.com/ifm-ai/JaisVoice	Apache 2.0

Table 41: Released artifacts of Jais 3.

7 Release Notes

7.1 Open-Source Statement and Commitment

We release Jais 3 openly so that others can use, study, and adapt it. Table 41 lists the released artifacts. Models and training code are released under the Apache 2.0 license; the training-data sample keeps the terms of its sources (ODC-BY and CC BY 4.0). The stage checkpoints and recipes let others continue from any point of the pipeline, or adapt K2 Horizon to another language in the same way.

7.2 Intended Use and Limitations

Intended use: The model is designed for Arabic and English assistant and agentic applications, including reasoning and tool-use tasks, as well as for research and experimentation.

Out of scope: The model should not be used in ways that violate applicable laws or regulations. The following uses are also considered out of scope:

- **Harmful or malicious use:** Generating hateful, extremist, discriminatory, or deliberately misleading content, or using the model to support illegal or harmful activities.
- **Sensitive information:** Using the model to generate, expose, or infer personal, confidential, or otherwise sensitive information about individuals or organizations.
- **Unsupported languages:** Applications that depend on reliable performance in languages other than Arabic or English.
- **High-stakes decisions:** Relying on the model to make medical, legal, financial, or safety-critical decisions without appropriate human oversight.

Limitations: The model can produce inaccurate or fabricated content and may reflect biases present in its training data. Evaluate it for your use case and implement appropriate safeguards before deployment.

8 Conclusion

We presented **Jais 3**, six open Arabic-centric agentic models from 0.9B to 375B parameters. Each model starts from the post-trained **K2 Horizon** model of the same size and is adapted to Arabic through continual pretraining, SFT, FTPO, and RLVR. **Jais 3** leads or matches the best open models on Arabic at every size, and at 375B it competes with frontier closed models (Section 6). It also works as an agent: it plans multi-step tasks, calls tools, and works on Arabic files, and at 3.7B and 7B it tops every agentic benchmark we ran against models of its size (Section 6.2).

Jais 3 shows that an open foundation can be adapted to a new language without starting from scratch, while keeping most of what it could already do. Along the way, we moved Arabic evaluation to a generative protocol and audited the benchmarks it relies on, and we release the models, every stage checkpoint, the recipes, a sample of the training data, and the audited benchmarks (Section 7). We hope these let others build agents for Arabic speakers, and adapt **K2 Horizon** to other languages in the same way.

Authors' Contributions

Core Contributors

- **Training:** Hadi Abdine, Ahmad Chamma, Amr Mohamed, Ahmed Frikha, Dani Bouch, Omar Arbi, Guokan Shang
- **Data:** Nikhil Ranjan, Maaz Maqsood, Shivam Chauhan, Ahmer Tabassum, Owais Aijaz, Dani Bouch, Amr Mohamed, Ahmad Chamma, Abdelhakim Freihat, Omar Arbi, Ahmed Frikha, Sarath Chandran, Neha Sengupta, Rahul Pal, Mohammed Bellam, Sajid Siddiki, Sarah AlBarri
- **Evaluation:** Mohamed Anwar, George Ibrahim, Mostafa Awad, Nurkhan Laiyk, Hasan Iqbal, Abhishek Maiti, Bilal Bouardi, Aaryamonvikram Singh, Dani Bouch, Amr Mohamed, Sarah AlBarri, Debopriyo Banerjee, Ali El Filali, Samta Kamboj, Samujjwal Ghosh
- **Project Management:** Hector Ren, Mohamed Anwar, Guokan Shang, Preslav Nakov, Zainul Abedien Ahmed Quraishi, Larry Murray, Avraham Sheinin, Natalia Vassilieva
- **Conception, Design, and Leadership:** Haonan Li, Zhengzhong Liu, Eric Xing

Contributors

Rifo Genadi, Ahmed Attia, Go Inoue, Shuonan Dong, Amine Khelif Khelif, Evan Dufraisse, Florian Valade, Tapas Sen, Guowei He, Sunil Sahu, Onkar Pandit, Kareem Magdy, Sheikh Khalid bin Khalifa Al Nahyan

References

- Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10. Association for Computational Linguistics, 2021. URL <https://aclanthology.org/2021.wanlp-1.1/>.
- Mouath Abu-Daoud, Leen Kharouf, Omar El Hajj, Dana El Samad, Mariam Al-Omari, Jihad Mallat, Khaled Saleh, Nizar Habash, and Farah E. Shamout. MedAraBench: Large-scale Arabic medical question answering dataset and benchmark, 2026. URL <https://arxiv.org/abs/2602.01714>.
- Muhammad Dehan Al Kautsar, Saeed Almheiri, Momina Ahsan, et al. Cultural benchmarking of LLMs in standard and dialectal Arabic dialogues. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21016–21039, 2026. doi: 10.18653/v1/2026.acl-long.963. URL <https://aclanthology.org/2026.acl-long.963/>.
- Aisha Alansari and Hamzah Luqman. HalluScore: Large language model hallucination question answering benchmark, 2026. URL <https://arxiv.org/abs/2605.17007>.
- Safa AlBallaa, Nora AlTwaresh, Abdulmalik AlSalman, and Sultan Alfarhood. GATmath and GATLc: Comprehensive benchmarks for evaluating Arabic large language models. *PLOS ONE*, 20(9):e0329129, 2025. doi: 10.1371/journal.pone.0329129. URL <https://doi.org/10.1371/journal.pone.0329129>.
- Wafa Alghallabi, Ritesh Thawkar, Sara Ghaboura, Ketan More, Omkar Thawakar, Hisham Cholakkal, Salman Khan, and Rao Muhammad Anwer. Fann or Flop: A multigenre, multiera benchmark for Arabic poetry understanding in LLMs, 2025. URL <https://arxiv.org/abs/2505.18152>.
- Omar Alkaabi, Ahmed Alzubaidi, Hamza Alobeidli, Shaikha Alsuwaidi, Mohammed Alyafeai, Leen AlQadi, Basma El Amel Boussaha, and Hakim Hacid. Alyah: An Emirati dialect benchmark for evaluating Arabic large language models, January 2026. URL <https://huggingface.co/datasets/tiiuae/alyah-emirati-benchmark>.
- Yazeed Alnumay, Alexandre Barbet, Anna Bialas, et al. Command R7B Arabic: A small, enterprise focused, multilingual, and culturally aware Arabic LLM, 2025. URL <https://arxiv.org/abs/2503.14603>.
- Leen AlQadi, Ahmed Alzubaidi, Mohammed Alyafeai, Hamza Alobeidli, Maitha Alhammadi, Shaikha Alsuwaidi, Omar Alkaabi, Basma El Amel Boussaha, and Hakim Hacid. Are Arabic benchmarks reliable? QIMMA’s quality-first approach to LLM evaluation. *arXiv preprint arXiv:2604.03395*, 2026.
- Malik H. Altakrori, Nizar Habash, Teresa Lynn, Younes Samih, Abed Alhakim Freihat, Kirill Chirkunov, Muhammed AbuOdeh, Radu Florian, Preslav Nakov, and Alham Fikri Aji. Dialectal-Arabic-MMLU: Benchmarking dialectal capabilities in Arabic and multilingual language models. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pages 3199–3219. ELRA Language Resource Association, 2026. doi: 10.63317/3cy68duew55b. URL <https://aclanthology.org/2026.lrec-1.251/>.
- Alhanoof Althnian, Norah A. Alzahrani, Shaykhah Z. Alsubaie, Eman Albilali, Ahmed Abdelali, Nouf M. Alotaibi, M Saiful Bari, Yazeed Alnumay, Abdulhamed Alothaimen, Maryam Saif, Shahad D. Alzaidi, Faisal Abdulrahman Mirza, Yousef Almushayqih, Mohammed Al Saleem, Ghadah Alabduljabbar, Abdulmohsen Al-Thubaity, Areeb Alowisheq, and Nora Al-Twaresh. AraEval: An Arabic multi-task evaluation suite for large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33037–33061, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1679. URL <https://aclanthology.org/2025.emnlp-main.1679/>.
- Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. PalmX 2025: The first shared task on benchmarking LLMs on Arabic and Islamic culture. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 774–789. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.107. URL <https://aclanthology.org/2025.arabicnlp-sharedtasks.107/>.
- Anthropic. Introducing the Model Context Protocol. <https://www.anthropic.com/news/model-context-protocol>, November 2024. Specification: <https://modelcontextprotocol.io>.

- Mohamed Anwar, Abed Alhakim Freihat, George Ibrahim, Mostafa Awad, Abdelrahman Sadallah, et al. Jais 2: A family of Arabic-centric open large language models, 2026. URL <https://arxiv.org/abs/2608.13580>.
- Artificial Analysis. Artificial Analysis long context reasoning (AA-LCR). Online evaluation, 2025a. URL <https://artificialanalysis.ai/evaluations/artificial-analysis-long-context-reasoning>. Accessed: 2026.
- Artificial Analysis. GDPval-AA: Artificial Analysis’ evaluation framework for OpenAI’s GDPval. <https://artificialanalysis.ai/evaluations/gdpval-aa>, December 2025b. Version 2.1. Methodology: <https://artificialanalysis.ai/methodology/intelligence-benchmarking>. Accessed 2026-10-11.
- Artificial Analysis. Stirrup: The lightweight framework for building agents. <https://github.com/ArtificialAnalysis/Stirrup>, December 2025c. Open-source agent harness (MIT license). Announcement: <https://artificialanalysis.ai/articles/stirrup-open-source-framework-agents>.
- Yasser Ashraf, Yuxia Wang, Bin Gu, Preslav Nakov, and Timothy Baldwin. Arabic dataset for LLM safeguard evaluation. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5529–5546, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.285. URL <https://aclanthology.org/2025.naacl-long.285/>.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. doi: 10.48550/arXiv.2108.07732.
- Lucas Bandarkar, Davis Liang, Benjamin Müller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 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, 2024. doi: 10.18653/v1/2024.acl-long.44. URL <https://aclanthology.org/2024.acl-long.44/>.
- M. Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, et al. ALLaM: Large language models for Arabic and English, 2024. URL <https://arxiv.org/abs/2407.15390>.
- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2 -Bench: Evaluating conversational agents in a dual-control environment. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026a. URL <https://openreview.net/forum?id=0C2z7iSQKa>.
- Victor Barres, Ben Shi, Ola Zytek, Soham Ray, Keshav Dhandhanian, and Pedram Razavi. τ^3 -Bench: Advancing agent benchmarking to knowledge and voice. Sierra blog, <https://sierra.ai/blog/bench-advancing-agent-benchmarking-to-knowledge-and-voice>, March 2026b. Code and data: <https://github.com/sierra-research/tau2-bench>.
- Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models, 2024. URL <https://arxiv.org/abs/2405.14782>.
- BigScience Workshop, Teven Le Scao, Angela Fan, et al. BLOOM: A 176b-parameter open-access multilingual language model, 2022. URL <https://arxiv.org/abs/2211.05100>.
- Abdessalam Boucekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouni, Aiman Erbad, and Mohammed Ghaly. QIAS 2025: Overview of the shared task on Islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.117. URL <https://aclanthology.org/2025.arabicnlp-sharedtasks.117/>.

- Sabri Boughorbel and Majd Hawasly. Analyzing multilingual competency of LLMs in multi-turn instruction following: A case study of Arabic. In Hassan Sawaf, Samhaa El-Beltagy, Wajdi Zaghouani, Walid Magdy, Ahmed Abdelali, Nadi Tomeh, Ibrahim Abu Farha, Nizar Habash, Salam Khalifa, Amr Keleg, Hatem Haddad, Imed Zitouni, Khalil Mrini, and Rawan Almatham, editors, *Proceedings of ArabicNLP 2023*, pages 128–139, Singapore (Hybrid), December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.arabnlp-1.11. URL <https://aclanthology.org/2023.arabnlp-1.11/>.
- Basma El Amel Boussaha, Leen Al Qadi, Mugariya Farooq, Shaikha Alsuwaidi, Giulia Campesan, Ahmed Alzubaidi, Mohammed Alyafeai, and Hakim Hacid. 3LM: Bridging Arabic, STEM, and code through benchmarking. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 42–63. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.arabnlp-main.4. URL <https://aclanthology.org/2025.arabnlp-main.4/>.
- Basma El Amel Boussaha, Mohammed Alyafeai, Ahmed Alzubaidi, et al. Falcon-H1-Arabic: State-of-the-art Arabic language models with hybrid Mamba-Transformer architecture, 2026. URL <https://huggingface.co/blog/tiuuae/falcon-h1-arabic>. Official model release report.
- Andrei Z. Broder. On the resemblance and containment of documents. In *Proceedings of Compression and Complexity of SEQUENCES 1997*, pages 21–29. IEEE, 1997.
- Laurie Burchell, Jean Maillard, Antonios Anastasopoulos, Christian Federmann, Philipp Koehn, and Skyler Wang. Findings of the WMT 2024 shared task of the open language data initiative. In *Proceedings of the Ninth Conference on Machine Translation*, pages 110–117, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.wmt-1.4. URL <https://aclanthology.org/2024.wmt-1.4/>.
- Andrea Caciolai, Pere-Lluís Huguet Cabot, Chierh Cheng, et al. OmnilingualGAIA2: Evaluating the multilingual gap in frontier AI agents, 2026. URL <https://arxiv.org/abs/2608.08775>.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. doi: 10.48550/arXiv.2107.03374.
- Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026a.
- Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Hosna Oyarhoseini, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhui Chen, and Jimmy Lin. BrowseComp-plus: A fair and disentangled evaluation benchmark for deep search agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22349–22370, San Diego, California, United States, July 2026b. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1023. URL <https://aclanthology.org/2026.acl-long.1023/>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Rei Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Creative Commons. Attribution 4.0 International (CC BY 4.0). <https://creativecommons.org>, 2013.
- Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.
- DeepSeek-AI. DeepSeek-V3.2: Pushing the frontier of open large language models, 2025. URL <https://arxiv.org/abs/2512.02556>.
- Jasper Dekoninck, Nikola Jovanović, Tim Gehringer, Kári Rognvaldsson, Ivo Petrov, Chenhao Sun, and Martin

- Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs, 2026. URL <https://arxiv.org/abs/2605.00674>.
- Kaustubh Deshpande, Ved Sirdeshmukh, Johannes Baptist Mols, Lifeng Jin, Ed-Yeremai Hernandez-Cardona, Dean Lee, Jeremy Kritz, Willow E. Primack, Summer Yue, and Chen Xing. MultiChallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18632–18702, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.958. URL <https://aclanthology.org/2025.findings-acl.958/>.
- Shuangrui Ding, Xuanlang Dai, Long Xing, Shengyuan Ding, Ziyu Liu, Jingyi Yang, Penghui Yang, Zhixiong Zhang, Xilin Wei, Xinyu Fang, Yubo Ma, Haodong Duan, Jing Shao, Jiaqi Wang, Dahua Lin, Kai Chen, and Yuhang Zang. WildClawBench: A benchmark for real-world, long-horizon agent evaluation. *arXiv preprint arXiv:2605.10912*, 2026. URL <https://arxiv.org/abs/2605.10912>.
- Konstantin Dobler and Gerard de Melo. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.829. URL <https://aclanthology.org/2023.emnlp-main.829/>.
- Ali El Filali, Hamza Alobeidli, Cl  mentine Fourrier, Basma El Amel Boussaha, Ruxandra Cojocaru, Nathan Habib, and Hakim Hacid. Open Arabic LLM leaderboard. <https://huggingface.co/spaces/OALL/Open-Arabic-LLM-Leaderboard>, 2024.
- Ali El Filali, Sarah AlBarri, Arwa Abouelseoud, Samta Kamboj, Neha Sengupta, and Preslav Nakov. Arabic leaderboards: Introducing Arabic instruction following, updating AraGen, and more. Hugging Face blog, April 2025. URL <https://huggingface.co/blog/leaderboard-3c3h-aragen-ifeval>. Dataset: <https://huggingface.co/datasets/inception42/Arabic-IFEval>.
- Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. Curras + Baladi: Towards a Levantine corpus. In Nicoletta Calzolari, Fr  d  ric B  chet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, H  l  ne Mazo, Jan Odi  k, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 769–778, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.82/>.
- Ahmed Elhady, Eneko Agirre, and Mikel Artetxe. Emergent abilities of large language models under continued pre-training for language adaptation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32174–32186. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1547. URL <https://aclanthology.org/2025.acl-long.1547/>.
- Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, et al. Fanar: An Arabic-centric multimodal generative AI platform, 2025. URL <https://arxiv.org/abs/2501.13944>.
- Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, et al. Fanar 2.0: Arabic generative AI stack, 2026. URL <https://arxiv.org/abs/2603.16397>.
- Abed Alhakim Freihat, G  bor Bella, Mourad Abbas, Hamdy Mubarak, and Fausto Giunchiglia. ALP: An Arabic linguistic pipeline. In Mourad Abbas, editor, *Analysis and Application of Natural Language and Speech Processing, Signals and Communication Technology*, pages 67–99. Springer, Cham, 2023. doi: 10.1007/978-3-031-11035-1_4.
- Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Shota Hirai, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. Continual pre-training for cross-lingual LLM adaptation: Enhancing Japanese language capabilities. In *First Conference on Language Modeling*, 2024. URL <https://arxiv.org/abs/2404.17790>.
- Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. Are we done with MMLU? In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas*

- Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.262. URL <https://aclanthology.org/2025.naacl-long.262/>.
- Gemma Team. Gemma 4 technical report, 2026. URL <https://arxiv.org/abs/2607.02770>.
- Gemma Team, Aishwarya Kamath, Johan Ferret, et al. Gemma 3 technical report, 2025. URL <https://arxiv.org/abs/2503.19786>.
- Gurpreet Gosal, Yishi Xu, Gokul Ramakrishnan, Rituraj Joshi, Avraham Sheinin, Zhiming (Charles) Chen, Biswajit Mishra, Natalia Vassilieva, Joel Hestness, Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Onkar Pandit, Satheesh Katipomu, Samta Kamboj, Samujjwal Ghosh, Rahul Pal, Parvez Mullah, Soundar Doraiswamy, Mohamed El Karim Chami, and Preslav Nakov. Bilingual adaptation of monolingual foundation models. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024. URL <https://arxiv.org/abs/2407.12869>.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- Jiawei Gu, Zacc Yang, Chuanghao Ding, Rui Zhao, and Fei Tan. CMR scaling law: Predicting critical mixture ratios for continual pre-training of language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16143–16162. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.903. URL <https://aclanthology.org/2024.emnlp-main.903/>.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.740. URL <https://aclanthology.org/2020.acl-main.740/>.
- Lukas Haas, Gal Yona, Giovanni D’Antonio, Sasha Goldshtein, and Dipanjan Das. SimpleQA Verified: A reliable factuality benchmark to measure parametric knowledge, 2025. URL <https://arxiv.org/abs/2509.07968>.
- Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. Multi-IF: Benchmarking LLMs on multi-turn and multilingual instructions following, 2024. URL <https://arxiv.org/abs/2410.15553>.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL <https://arxiv.org/abs/2009.03300>.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Juncai He, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. AceGPT: Localizing large language models in Arabic, 2023. URL <https://arxiv.org/abs/2309.12053>.
- Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024. URL <https://openreview.net/forum?id=DimPeeCxKU>.
- IFM Team. Introducing K2 Horizon: Frontier performance, radically open. Model release and technical overview, 2026. URL <https://ifm.ai/blog/k2/>.
- Inception. Jais family model card, 2024. URL <https://huggingface.co/inception42/jais-family-30b-16k-chat>. Model card.
- Go Inoue, Bashar Alhafni, Nizar Habash, and Timothy Baldwin. Do diacritics matter? evaluating the impact of Arabic diacritics on tokenization and LLM benchmarks. In *Findings of the Association for Computational*

- Linguistics: EACL 2026*, pages 426–442. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-eacl.22. URL <https://aclanthology.org/2026.findings-eacl.22/>.
- Institute of Foundation Models. RL360: Reinforcement learning framework, 2026.
- Declan Jackson, William Keating, George Cameron, and Micah Hill-Smith. AA-Omniscience: Evaluating cross-domain knowledge reliability in large language models, 2025. URL <https://arxiv.org/abs/2511.13029>.
- Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. Building a corpus for Palestinian Arabic: a preliminary study. In Nizar Habash and Stephan Vogel, editors, *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3603. URL <https://aclanthology.org/W14-3603/>.
- Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. Curras: An annotated corpus for the Palestinian Arabic dialect. *Language Resources and Evaluation*, 51(3):745–775, 2017.
- Mustafa Jarrar, Fadi A Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlsch. Lisan: Yemeni, Iraqi, Libyan, and Sudanese Arabic dialect copora with morphological annotations, 2022. URL <https://arxiv.org/abs/2212.06468>.
- Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=VTF8yNQM66>.
- Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.
- Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, et al. BIG-Bench Extra Hard, 2025. URL <https://arxiv.org/abs/2502.19187>.
- Dmitrii Khizbullin, Zaid Alyafeai, Abdelrahman Eldesokey, et al. Telco-GAIA: Bilingual benchmark for agents in telecom domain, 2026. URL <https://arxiv.org/abs/2607.20510>.
- Yunsu Kim, Kaden Uhlig, and Joern Wuebker. GAIA-v2-LILT: Multilingual adaptation of agent benchmark beyond translation. In Kaiyu Huang, Fengran Mo, Pinzhen Chen, and Meng Jiang, editors, *Proceedings of the 1st Workshop on Multilinguality in the Era of Large Language Models (MeLLM 2026)*, pages 140–148, San Diego, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.mellm-1.13. URL <https://aclanthology.org/2026.mellm-1.13/>.
- Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. ArabicMMLU: Assessing massive multitask language understanding in Arabic. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5622–5640, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.334. URL <https://aclanthology.org/2024.findings-acl.334/>.
- Konstantin Kubrak, Ahmed El-Moselhy, Ammar Alsulami, et al. Arabic prompts with English tools: A benchmark, 2026. URL <https://arxiv.org/abs/2601.05101>.
- 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 29th Symposium on Operating Systems Principles*, pages 611–626, 2023. doi: 10.1145/3600006.3613165. URL <https://arxiv.org/abs/2309.06180>.
- Hynek Kydlíček. Math-verify: Math verification library. <https://github.com/huggingface/Math-Verify>, 2025.
- Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 318–327, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-demo.28.

- Zhenyu Li, Kehai Chen, Yunfei Long, Xuefeng Bai, Yaoyin Zhang, Xuchen Wei, Juntao Li, and Min Zhang. XIFBench: Evaluating large language models on multilingual instruction following. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL <https://openreview.net/forum?id=qkdVjCAPOE>.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step, 2023. URL <https://arxiv.org/abs/2305.20050>.
- Liquid AI. Reducing doom loops with final token preference optimization. Liquid AI Blog, 2026. URL <https://www.liquid.ai/blog/antidoom>.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective, 2025. URL <https://arxiv.org/abs/2503.20783>.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2023.
- Yifan Mai, Rida Taleb, and Ahmed Elhagry. HELM Arabic Enterprise. Stanford Center for Research on Foundation Models (CRFM), in collaboration with Arabic.AI, 2026. URL <https://crfm.stanford.edu/2026/05/26/helm-arabic-enterprise.html>. Leaderboard.
- Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: a standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org, 2024.
- Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, et al. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL <https://openreview.net/forum?id=a7Qa4CcHak>.
- Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=fixvvhvs3>.
- Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.293. URL <https://aclanthology.org/2022.naacl-main.293/>.
- Basel Mousi, Nadir Durrani, Fatema Ahmad, Md. Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. AraDiCE: Benchmarks for dialectal and cultural capabilities in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4186–4218. Association for Computational Linguistics, 2025. URL <https://aclanthology.org/2025.coling-main.283/>.
- Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Mohamed Darwish, and Walid Magdy. IslamicEval 2025: The first shared task of capturing LLMs hallucination in Islamic content. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 480–493. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.67. URL <https://aclanthology.org/2025.arabicnlp-sharedtasks.67/>.
- Hamdy Mubarak, Abubakr Mohamed, and Majd Hawasly. AraSafe: Benchmarking safety in Arabic LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9976–9992, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.529. URL <https://aclanthology.org/2025.findings-emnlp.529/>.

- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, et al. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, 2023. doi: 10.18653/v1/2023.acl-long.891. URL <https://aclanthology.org/2023.acl-long.891/>.
- Ahmad Mustapha, Hadi Al-Khansa, Hadi Al-Mubasher, Aya Mourad, Ranam Hamoud, Hasan El-Husseini, Marwah Al-Sakkaf, and Mariette Awad. AraSTEM: A native Arabic multiple choice question benchmark for evaluating LLMs knowledge in STEM subjects, 2025. URL <https://arxiv.org/abs/2501.00559>.
- Amal Nayouf, Tymaa Hammouda, Mustafa Jarrar, Fadi Zaraket, and Mohamad-Bassam Kurdy. Nâbra: Syrian Arabic dialects with morphological annotations. In Hassan Sawaf, Samhaa El-Beltagy, Wajdi Zaghouni, Walid Magdy, Ahmed Abdelali, Nadi Tomeh, Ibrahim Abu Farha, Nizar Habash, Salam Khalifa, Amr Keleg, Hatem Haddad, Imed Zitouni, Khalil Mrini, and Rawan Almatham, editors, *Proceedings of ArabicNLP 2023*, pages 12–23, Singapore (Hybrid), December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.arabnlp-1.2. URL <https://aclanthology.org/2023.arabnlp-1.2/>.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846, 2024. doi: 10.1038/s41586-024-07335-x.
- Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL <https://openreview.net/forum?id=XccDXrDNLek>.
- Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. CAMEL tools: An open source Python toolkit for Arabic natural language processing. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL <https://aclanthology.org/2020.lrec-1.868/>.
- Victor Ojewale, Inioluwa Deborah Raji, and Suresh Venkatasubramanian. Multi-lingual functional evaluation for large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 34672–34691, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1731. URL <https://aclanthology.org/2026.findings-acl.1731/>.
- Omartificial-Intelligence-Space. Arabic GSM8K: Arabic grade school math dataset. <https://huggingface.co/datasets/Omartificial-Intelligence-Space/Arabic-gsm8k>, 2025.
- OpenAI. Introducing SWE-bench Verified. <https://openai.com/index/introducing-swe-bench-verified/>, August 2024.
- Samuel J Paech, Allen G Roush, Judah Goldfeder, and Ravid Shwartz-Ziv. Antislop: A comprehensive framework for identifying and eliminating repetitive patterns in language models. *arXiv preprint arXiv:2510.15061*, 2025.
- Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 1094–1101, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA). URL <https://aclanthology.org/L14-1479/>.
- Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simon Fishman, Marwan Aljubei, Phoebe Thacker, Laurance Fauconnet, et al. GDPval: Evaluating AI model performance on real-world economically valuable tasks. In *International Conference on Learning Representations*, volume 2026, pages 24005–24040, 2026.

- Guilherme Penedo, Hynek Kydlíček, Alessandro Cappelli, Mario Sasko, and Thomas Wolf. DataTrove: Large scale data processing. <https://github.com/huggingface/datatrove>, 2024.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.617. URL <https://aclanthology.org/2020.emnlp-main.617/>.
- Charilaos Pipis, Shivam Garg, Vasilis Kontonis, Vaishnavi Shrivastava, Akshay Krishnamurthy, and Dimitris Papailiopoulos. Wait, wait, wait... why do reasoning models loop? *arXiv preprint arXiv:2512.12895*, 2025.
- Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL <https://openreview.net/forum?id=yfYgwjj5F8>.
- Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023.
- Qwen Team. Qwen3.5 model cards. Hugging Face model collection, 2026a. URL <https://huggingface.co/Qwen/Qwen3.5-4B>. Accessed: 2026-09-04.
- Qwen Team. Qwen3.6-27B. Hugging Face model card, 2026b. URL <https://huggingface.co/Qwen/Qwen3.6-27B>. Accessed: 2026-09-04.
- Qwen Team. Qwen3.8-27B. Hugging Face model card, 2026c. URL <https://huggingface.co/Qwen/Qwen3.8-27B>. Accessed: 2026-09-04.
- RadixArk. Miles v0.1: Production-level post-training, 2026. URL <https://arxiv.org/abs/2609.08368>.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark, 2023. URL <https://arxiv.org/abs/2311.12022>.
- Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5377–5400, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.301. URL <https://aclanthology.org/2024.naacl-long.301/>.
- Abdelrahman Sadallah, Junior Cedric Tonga, Khalid Almubarak, Saeed Almheiri, Farah Atif, Chatrine Qwaider, Karima Kadaoui, Sara Shatnawi, Yaser Alesh, and Fajri Koto. Commonsense reasoning in Arab culture. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7695–7710. Association for Computational Linguistics, 2025. URL <https://aclanthology.org/2025.acl-long.380/>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*, 2023.
- Guokan Shang, Hadi Abdine, Ahmad Chamma, Amr Mohamed, Mohamed Anwar, Abdelaziz Bounhar, Omar El Herraoui, Preslav Nakov, Michalis Vazirgiannis, and Eric P. Xing. Nile-Chat: Egyptian language models

- for Arabic and Latin scripts. In Kareem Darwish, Ahmed Ali, Ibrahim Abu Farha, Samia Touileb, Imed Zitouni, Ahmed Abdelali, Sharefah Al-Ghamdi, Sakhar Alkhereyf, Wajdi Zaghouani, Salam Khalifa, Badr AlKhamissi, Rawan Almatham, Injy Hamed, Zaid Alyafeai, Areeb Alowisheq, Go Inoue, Khalil Mrini, and Waad Alshammari, editors, *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 306–322, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-352-4. doi: 10.18653/v1/2025.arabicnlp-main.25. URL <https://aclanthology.org/2025.arabicnlp-main.25/>.
- Guokan Shang, Hadi Abdine, Yousef Khoubrane, Amr Mohamed, Yassine Abbahaddou, Sofiane Ennadir, Imane Momayiz, Xuguang Ren, Eric Moulines, Preslav Nakov, Michalis Vazirgiannis, and Eric Xing. Atlas-Chat: Adapting large language models for low-resource Moroccan Arabic dialect. In Hansi Hettiarachchi, Tharindu Ranasinghe, Paul Rayson, Ruslan Mitkov, Mohamed Gaber, Damith Premasiri, Fiona Anting Tan, and Lasitha Uyangodage, editors, *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 9–30, Abu Dhabi, United Arab Emirates, January 2025b. Association for Computational Linguistics. URL <https://aclanthology.org/2025.loreslm-1.2/>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Quan Shi, Alexandra Zyteck, Pedram Razavi, Karthik Narasimhan, and Victor Barres. τ -knowledge: Evaluating conversational agents over unstructured knowledge. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL <https://openreview.net/forum?id=XHZK5abtw2>.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.
- SWE-agent Team. mini-SWE-agent: The 100 line AI agent that solves GitHub issues. <https://github.com/SWE-agent/mini-swe-agent>, 2025. Accessed 2026-10-11.
- Dima Taji, Salam Khalifa, Ossama Obeid, Fadhl Eryani, and Nizar Habash. An Arabic morphological analyzer and generator with copious features. In Sandra Kuebler and Garrett Nicolai, editors, *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 140–150, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5816. URL <https://aclanthology.org/W18-5816/>.
- Liping Tang, Nikhil Ranjan, Omkar Pangarkar, Xuezhi Liang, Zhen Wang, Li An, Bhaskar Rao, Linghao Jin, Huijuan Wang, Zhoujun Cheng, Suqi Sun, Cun Mu, Victor Miller, Xuezhe Ma, Yue Peng, Zhengzhong Liu, and Eric P. Xing. TxT360: A top-quality LLM pre-training dataset requires the perfect blend. <https://huggingface.co/spaces/LLM360/TxT360>, 2024.
- The Terminal-Bench Team. Terminal-Bench 2.1. <https://www.tbench.ai/news/terminal-bench-2-1>, May 2026.
- United Nations. Arabic language day. <https://www.un.org/en/observances/arabic-language-day>, 2026. Accessed: 2026-10-04.
- Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D’souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, et al. Aya model: An instruction finetuned open-access multilingual language model, 2024.
- Bertie Vidgen, Austin Mann, Abby Fennelly, John Wright Stanly, Lucas Rothman, Marco Burstein, Julien Benckek, David Ostrofsky, Anirudh Ravichandran, Debnil Sur, Neel Venugopal, Alannah Hsia, Isaac Robinson, Calix Huang, Olivia Varones, Daniyal Khan, Michael Haines, Austin Bridges, Jesse Boyle, Koby Twist, Zach Richards, Chirag Mahapatra, Brendan Foody, and Osvald Nitski. APEX-Agents. *arXiv preprint arXiv:2601.14242*, 2026. URL <https://arxiv.org/abs/2601.14242>.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhua Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, 2024. URL <https://arxiv.org/abs/2406.01574>.

- Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models, 2024. URL <https://arxiv.org/abs/2411.04368>.
- Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025. URL <https://arxiv.org/abs/2504.12516>.
- Zijian Wu, Xiangyan Liu, Xinyuan Zhang, Lingjun Chen, Fanqing Meng, Lingxiao Du, Yiran Zhao, Fanshi Zhang, Yaoqi Ye, Jiawei Wang, Zirui Wang, Jinjie Ni, Yufan Yang, Arvin Xu, and Michael Qizhe Shieh. MCPMark: A benchmark for stress-testing realistic and comprehensive MCP use. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL <https://openreview.net/forum?id=uobR0wBsJm>.
- Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- Weihao Xuan, Rui Yang, Heli Qi, et al. MMLU-ProX: A multilingual benchmark for advanced large language model evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1513–1532, 2025. doi: 10.18653/v1/2025.emnlp-main.79. URL <https://aclanthology.org/2025.emnlp-main.79/>.
- An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL <https://openreview.net/forum?id=mXpq6ut8J3>.
- Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. On the rollout-training mismatch in modern RL systems. In *OPT 2025: Optimization for Machine Learning*, 2025a.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025b. URL <https://openreview.net/forum?id=roNSXZpUDN>.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.
- Taha Zerrouki. Arramooz Alwaseet: Arabic dictionary for morphological analysis. <https://sourceforge.net/projects/arramooz/>, 2011.
- Weiqi Zhai, Zhihai Wang, Jinghang Wang, Boyu Yang, et al. HLE-Verified: A systematic verification and structured revision of Humanity’s Last Exam. *arXiv preprint arXiv:2602.13964*, 2026.
- Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=B18u7ZR1bM>.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.
- Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody H Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. SGLang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou.
Instruction-following evaluation for large language models. *ArXiv*, abs/2311.07911, 2023. URL <https://api.semanticscholar.org/CorpusID:265157752>.

A Reward Hyperparameters

Symbol	Description	Value
Shared		
ϕ	Format-penalty coefficient	0.02
Diacritization		
w_{char}	Character-accuracy weight	0.75
w_{word}	Word-accuracy weight	0.20
w_{fin}	Word-final accuracy weight	0.05
N_{min}	Minimum reference length (letters)	2
n_0	Minimum marks for no-diacritic cap	2
η	Mark fraction for no-diacritic cap	0.20
κ_{nd}	No-diacritic cap	0.10
A_{min}	Character-accuracy threshold	0.20
κ_{lo}	Low-accuracy cap	0.20
Grammatical analysis		
b_0	Base score, role and case agree	0.60
λ_M	Case-marker bonus	0.15
λ_O	Token-overlap weight	0.15
λ_T	Target-word bonus	0.05
b_c	Base score, case mismatch	0.25
γ_c	Overlap weight, case mismatch	0.10
γ_r	Overlap weight, role mismatch	0.10
γ_u	Overlap weight, undefined reference role	0.50
κ_{soft}	Cap: hedging, contradiction, multiple analyses, length	0.30
κ_{hard}	Cap: stuffing, wrong target, non-Arabic output	0.20

Table 42: Hyperparameters of the diacritization and grammatical-analysis rewards (Equations 7–9).

B Multi-Turn Data: Additional Details

This appendix specifies the checks summarized in Section 5.1.1: the structure a targeted conversation must have, the full qualification criteria for both conversation types, and how reasoning replay handles failures.

B.1 Anatomy of a Targeted Conversation

A targeted conversation ends with a user turn u_k whose correct answer depends on an earlier *anchor* turn u_j (Figure 15). The structural audit checks five properties of the visible conversation: presence (T1), separation (T2), no restatement (T3), version validity (T4), and counterfactual dependency (T5), the last only for conversations qualified under the strict contract.

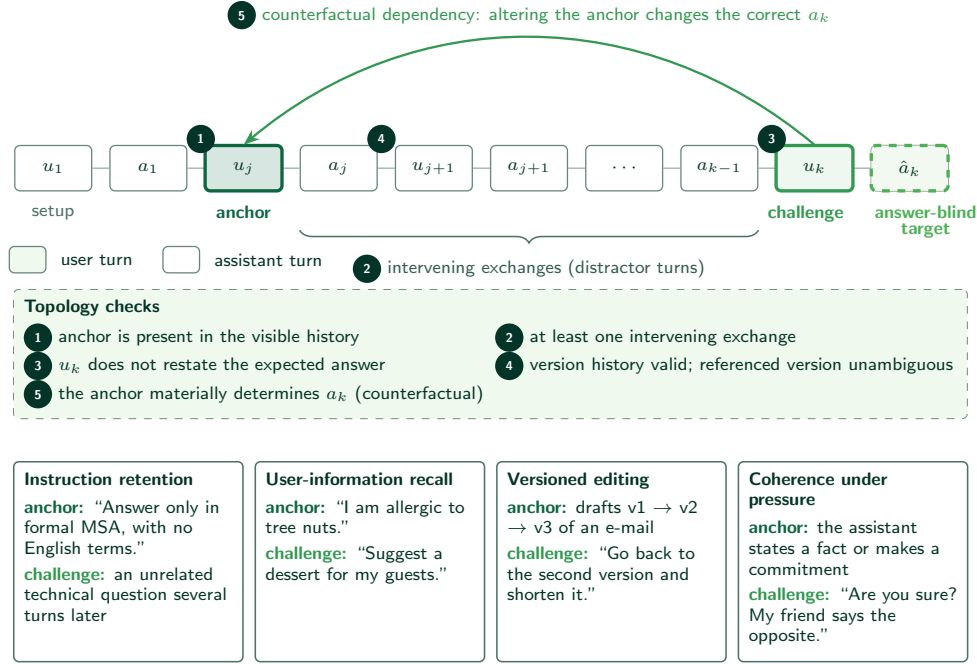


Figure 15: Anatomy of a targeted conversation: the final user turn u_k depends on the anchor u_j . Bottom: illustrative examples of the four dependency types.

B.2 Qualification Criteria

A conversation is accepted only if both judges, Qwen3.6-27B and gemma-4-31B-it, approve it on every criterion that applies to its type (Table 43). Targeted conversations must pass the structural checks, which guarantee the long-range dependency. General conversations lack this guarantee, so every assistant turn is instead audited individually, including for safety, privacy, and use of information introduced only later in the conversation. Both types share answer-blind regeneration, unanimous judging, and the replay checks.

B.3 Reasoning Replay and Fallbacks

Every assistant turn is replayed to attach an explicit reasoning trace (Figure 16). For each assistant position t , a replay model, fixed for the whole conversation, sees only the visible prefix before t and generates reasoning and an answer without seeing the original answer a_t . The judge rejects the replayed turn if its answer is of insufficient quality, its reasoning is unsound, the two are inconsistent, or the answer disagrees with a_t . Agreement is required because later turns build on a_t .

The two conversation types handle failures differently:

Table 43: Qualification criteria by conversation type. ✓: required; †: required for part of the data; −: not applied.

Group	Criterion	Targeted	General
Structure	T1 presence	✓	−
	T2 separation	✓	−
	T3 no restatement	✓	−
	T4 version validity	✓	−
	T5 counterfactual dependency	† ^a	−
Final answer	answer-blind regeneration	✓	✓
	honors the anchor	✓	−
	unanimous Qwen $\wedge$ Gemma	✓	✓
Conversation	coherence of assistant turns	✓	✓
	per-turn audit of every assistant turn	−	✓
	language / variant consistency	✓	✓
	consistent model identity	✓	✓
	factual / domain accuracy	† ^b	✓
	safety, privacy, no future leakage	−	✓
Replay	answer, reasoning, mutual consistency	✓	✓
	agreement with original answer; whole-chat review	✓	✓

^aConversations qualified under the strict contract. ^bConversations under the strict contract and the separately reviewed scientific and research subset.

- **Failed replay of a turn.** A targeted conversation keeps the original a_t without reasoning, so a released conversation can mix turns with and without reasoning traces. A general conversation is rejected.
- **Failed whole-chat review.** A targeted conversation falls back to its previously validated version without reasoning, where one exists. A general conversation is excluded.

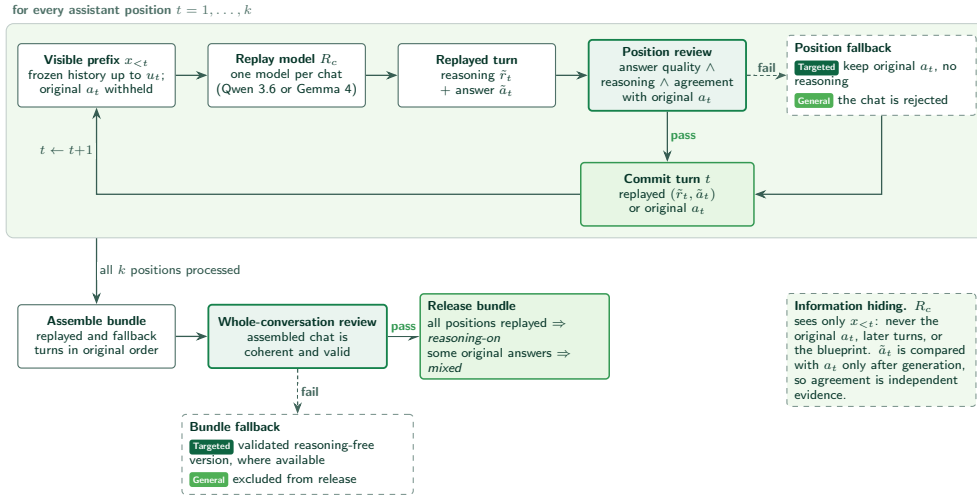


Figure 16: Reasoning replay and assembly.

C Complete Evaluation Setup

This appendix provides the complete configuration supporting the summary in Section 6.1.3. It records the evaluated model families, benchmark inventory, prompting and chat-template behavior, decoding policy, Arabic-specific processing, scoring rules, repeatability protocol, and runtime implementation. Exact checkpoint identifiers, source revisions, and machine-readable generation configurations accompany the released evaluation artifacts.

C.1 Decoding Configurations

Jais 3 is compared with Arabic-focused and multilingual leading models at similar and larger parameter scales. Table 44 summarizes the model families and scales represented in the evaluation. The released configuration artifact records the exact repository identifier, checkpoint revision, tokenizer and chat-template revision, and evaluated reasoning mode for every row.

Model	Size	Hugging Face	Commit
1B Scale			
tiny-aya-global	3B	CohereLabs/tiny-aya-global	00590ff
Qwen3.5-2B	2B	Qwen/Qwen3.5-2B	15852e8
Falcon-H1-1.5B-Deep-Instruct	1.5B	tiiuae/Falcon-H1-1.5B-Deep-Instruct	b664863
Hala-1.2B	1.2B	hammh0a/Hala-1.2B	10e586c
Jais-3-0.9B	0.9B	inception42/Jais-3-0.9B	e51941b
4B Scale			
Qwen3.5-4B	4B	Qwen/Qwen3.5-4B	851bf6e
gemma-4-E2B-it	E2B	google/gemma-4-E2B-it	3e22461
granite-4.0-h-3b-ar	3B	ibm-research/granite-4.0-h-3b-ar	b32260f
Falcon-H1-3B-Instruct	3B	tiiuae/Falcon-H1-3B-Instruct	01087ec
Jais-3-3.7B	3.7B	inception42/Jais-3-3.7B	ac03017
7B Scale			
gemma-4-E4B-it	E4B	google/gemma-4-E4B-it	ee0ef60
gemma-4-12B-it	12B	google/gemma-4-12B-it	707f0a3
Qwen3.5-9B	9B	Qwen/Qwen3.5-9B	c202236
Falcon-H1-7B-Instruct	7B	tiiuae/Falcon-H1-7B-Instruct	41e72f2
Jais-2-8B-Chat	8B	inception42/Jais-2-8B-Chat	da0e163
ALLaM-7B-Instruct-preview	7B	humain-ai/ALLaM-7B-Instruct-preview	a28dd1e
Fanar-1-9B-Instruct	9B	QCRI/Fanar-1-9B-Instruct	d30938d
Jais-3-7B	7B	inception42/Jais-3-7B	11dcac6
32B Scale			
gemma-4-31B-it	31B	google/gemma-4-31B-it	842da37
Muse-Glimmer-30B	30B	meta-models/Muse-Glimmer-30B	a4e59da
Qwen3.8-27B	27B	Qwen/Qwen3.8-27B	1d4bf0f
Fanar-2-27B-Instruct	27B	QCRI/Fanar-2-27B-Instruct	0740050
Jais-2-70B-Chat	70B	inception42/Jais-2-70B-Chat	8bcbd3a
Jais-3-32B	32B	inception42/Jais-3-32B	5249a7b
Jais-3-MoVA-36B-A4B	36B-A4B	inception42/Jais-3-MoVA-36B-A4B	c7690a5
375B Scale			
Mistral-Small-4-119B-2603	119B	mistralai/Mistral-Small-4-119B-2603	a11f36b
Qwen3.5-397B-A17B	397B-A17B	Qwen/Qwen3.5-397B-A17B	8472618
MiniMax-M3	428B-A23B	MiniMaxAI/MiniMax-M3	f0e1c1e
Qwen3.8-Flash-Next	180B-A6B	Qwen/Qwen3.8-Flash-Next	de4b8e4
Nemotron-3-Ultra-550B-A55B-BF16	550B-A55B	nvldia/NVIDIA-Nemotron-3-Ultra-550B-A55B-BF16	77df655
Jais-3-375B-A23B	375B-A23B	inception42/Jais-3-375B-A23B	1b5d789

Table 44: Evaluated open-weight checkpoints: Hugging Face model ID, parameter scale, repository link and the commit (revision) used.

All models were evaluated with the vLLM framework (Kwon et al., 2023), using the *bfloat16* tensor type, the Triton attention backend, and a nominal vLLM GPU-memory utilization of 0.90, lowered for specific capacity-constrained runs, and a seed of 42. Additional repeat-run seeds are fixed in advance and recorded in the supplementary configuration artifact. The nominal batch-size setting of 16,384 is a scheduling upper bound; vLLM performs the actual memory-aware batching. Tensor parallelism and node count vary with model size. Model context length is the smaller of the task allocation, an explicit command-line cap when supplied, and the maximum length read from the model configuration. Hugging Face access is offline during evaluation so that cached model and dataset artifacts do not change within a run.

We use developer-recommended decoding configurations rather than greedy decoding. For each model, we first use the decoding parameters reported for benchmark evaluation in its official technical report. If the technical report does not give them, we use those recommended in the official model card on Hugging Face or in its inference example. When no official sampling recommendation is available, we use deterministic greedy decoding (`do_sample=false`). The decoding parameters of every model are listed in our evaluation suite¹⁵.

C.2 Complete Benchmark Inventory

Table 45 lists each benchmark with its domain, lm-evaluation-harness task name, dataset source and revision, and primary metric. A benchmark may expand into multiple independently scored subtasks, such as 57 MMLU subjects, 40 ArabicMMLU subjects, 160 dialectal-MMLU subject–variety combinations, 34 FLORES+ translation directions, or 37 Arabic linguistic subtasks. “Local” denotes a fixed test file distributed with the internal benchmark snapshot. For datasets whose only public evaluation partition is named `train` by the provider, including IFEval and HarmBench, we use that released partition as the test set and do not train on it.

Domain	Harness name	Dataset (commit)	Metric
Arabic			
STEM	<code>arabic_gsm8k-zeroshot</code>	IFM/Arabic-gsm8k_verified (0ba617b)	normalized EM
Instruction Following	<code>ara_ifbench</code>	MBZUAI-IFM/ara-ifbench-wildchat (200df4f)	strict prompt accuracy
Instruction Following	<code>ifeval_ar</code>	<code>arabic_ifeval</code> (local)	strict prompt accuracy
Dialect Understanding and Generation	<code>trans_flores200_plus</code>	openlanguagedata/flores_plus (1c079b0) (local)	judge overall (1–5)*20
World Knowledge	<code>arabicmmlu_gen</code>	IFM/ArabicMMLU_verified (50b3f62)	accuracy
Dialect Understanding and Generation	<code>arabicmmlu_with_diacritics_gen</code>	go-inoue/ArabicMMLU_full (5004cfb)	accuracy
Dialect Understanding and Generation	<code>mmlu_ar_dialects_only_gen</code>	IFM/Dialectal-Arabic-MMLU_verified (dddd23e)	sample-weighted accuracy
Arab Culture	<code>aradice_culture_gen</code>	IFM/AraDiCE-Culture_verified (6a964e8)	sample-weighted accuracy
Arab Culture	<code>arab_culture_gen</code>	IFM/ArabCulture_verified (d709d49)	sample-weighted accuracy
STEM	<code>gat_math_gen</code>	IFM/GATmath_verified (994ee26)	accuracy
STEM	<code>arastem_gen</code>	IFM/AraSTEM_verified (690e1a7)	accuracy
Islamic Knowledge	<code>islamic_qa_QASI_1_gen</code>	IFM/QIAS2025_verified (be3b762)	accuracy
Islamic Knowledge	<code>islamic_qa_QASI_2_gen</code>	IFM/QIAS2025_verified (be3b762)	accuracy
Islamic Knowledge	<code>IslamicEval2015_1B_gen</code>	IFM/IslamicEval2025-Subtask1B_verified (8399485)	accuracy

Continued on next page

¹⁵<https://github.com/ifm-ai/gen-eval-arabic/blob/main/hyperparams.yaml>

Continued from previous page

Domain	Harness name	Dataset (commit)	Metric
Islamic Knowledge	islamic_qa_palmX2025_2_gen	IFM/PalmX2025-Subtask2_verified (2248e58)	accuracy
Arab Culture	poetry_analysis_gen	IFM/FannOrFlop-MCQ_verified (d2b665d)	sample-weighted accuracy
Dialect Understanding and Generation	dialect_id_QADI_gen	dialect_id (local)	sample-weighted accuracy
Arab Culture	alyah-emirati-benchmark_gen	IFM/alyah-emirati-benchmark_verified (4e55dd0)	accuracy
Arab Culture	arabic_cuisine_gen	arabic_cuisine (local)	accuracy
Arabic Linguistics	arabic_linguistic_gen	arabic_linguistic (local)	sample-weighted accuracy
Arabic Linguistics	ALP_arabic_parsing_gen	arabic_linguistic/ALP_arabic_parsing (local)	accuracy
Arabic Linguistics	ALP_arabic_diacritization_gen	arabic_linguistic/ALP_arabic_diacritization (local)	accuracy
Factuality & Hallucination	simpleqa_verified_ar	SimpleQA_Verified_AR (local)	judge-correct
Factuality & Hallucination	halluscore	HalluScore (local)	judge-correct
Multi-turn	mt_bench_arabic	MT_Bench_arabic (local)	judge score (1–10)
Legal	arabic_legal_helm_gen	IFM/arabic-enterprise_verified, legal (fa8b443)	judge accuracy
English			
World Knowledge & STEM	mmlu_pro_zero_shot	TIGER-Lab/MMLU-Pro (54611cd)	accuracy
World Knowledge & STEM	gpqa_diamond_zeroshot_gen	Idavidrein/gpqa (633f5ee)	accuracy
World Knowledge & STEM	gsm8k-zeroshot, gsm8k_cot_zeroshot	openai/gsm8k (740312a)	normalized EM
World Knowledge & STEM	aime25_modified_repeats_8, aime26_modified_repeats_8	math-ai/aime25 (563bb84), math-ai/aime26 (79037ae)	pass@k
World Knowledge & STEM	minerva_math500_reasoning	HuggingFaceH4/MATH-500 (6e4ed1a)	verified accuracy
Instruction Following	ifeval	google/IFEval (966cd89)	strict prompt accuracy
Multi-turn	multi_if	Multi-IF (local)	mean turn accuracy
Factuality & Hallucination	simpleqa_verified	google/simpleqa-verified (0dc97e0)	judge-correct
Factuality & Hallucination	aa_omniscience_gen	ArtificialAnalysis/AA-Omniscience-Public (4a8ffc8)	judge-correct
Long Context	aa_lcr_v1.1_en	ArtificialAnalysis/AA-LCR (9a77ef5)	accuracy
Multi-turn	multichallenge	MultiChallenge (local)	macro axis pass rate
Multi-turn	mt_bench	MT_Bench (local)	judge score (1–10)
Coding			
Coding	humaneval_instruct	openai/openai_humaneval (7dce605)	pass@1
Coding	mbpp_instruct	google-research-datasets/mbpp (4bb6404)	pass@1

Table 45: Domain, lm-evaluation-harness task name, dataset (linked Hugging Face commit, or *local* if loaded from the local benchmarks folder) and metric. Accuracy and pass rates are higher-is-better unless marked otherwise.

The full evaluation infrastructure contains 37 ALP-derived linguistic subtasks totaling 11,731 records. For focused analysis in this report, we selected the two finalized diagnostic benchmarks introduced in Section 6.1.2: the Arabic Linguistics – I’rab Benchmark (1,446 accepted items) and the Arabic Linguistics – Tashkeel Benchmark (1,344 accepted items). Accordingly, the detailed linguistic results report only these two benchmarks. The 11,731-record figure describes the broader ALP-derived inventory and is not their combined test-set size.

Diagnostic fields such as parse-failure rate, response length, and number of judge-able samples are retained for auditing but are not treated as model-quality metrics. Group scores are calculated from leaf-task scores weighted by evaluated sample counts. A metric is aggregated only when it is finite and present for every leaf in the group; standard-error fields and metadata are never averaged.

Several Arabic resources are local or modified snapshots rather than unmodified public datasets. The snapshot contains the exact examples used for evaluation, but the evaluation repository does not provide a uniform provenance record covering annotator recruitment, compensation, quality control, and inter-annotator agreement for every custom resource. Those details belong in the documentation for the corresponding dataset and must not be inferred from evaluation code alone.

The evaluation code itself does not test for contamination. The training data we added in continual pretraining, cooldown and SFT was screened against these benchmarks and matches were removed (Section 3.1).

C.3 Arabic Benchmark Details

This section describes the Arabic benchmarks listed in Table 28, following the grouping used in Section 6.1. Item counts are those of the original releases, before the audit described in Section 6.1.1.

C.3.1 General Capabilities

STEM and world knowledge. ArabicMMLU (Koto et al., 2024) contains 14,455 multiple-choice questions from school and university examinations across the Arab world, covering 40 academic and professional subjects. AraSTEM (Mustapha et al., 2025) contains 10,824 native Arabic multiple-choice questions on science, technology, engineering, and mathematics at several educational levels. GAT Math (AlBallaa et al., 2025) contains mathematics questions from the Saudi General Aptitude Test on algebra, problem solving, geometry, and quantitative comparison. Arabic GSM8K (Cobbe et al., 2021; Omartificial-Intelligence-Space, 2025) is an Arabic translation of the 1,319 test problems of GSM8K, grade-school word problems that require multi-step arithmetic.

UAE legal. The legal tasks of HELM Arabic Enterprise (Mai et al., 2026) consist of 200 expert-written questions on UAE civil, commercial, and penal law. Each question is posed both with a context passage (open book) and without one (closed book), and an LLM judge grades the responses for correctness.

Medical. MedAraBench (Abu-Daoud et al., 2026) is a large-scale benchmark of Arabic medical question answering and clinical reasoning. Its 4,927 items cover diverse medical subdomains through multiple-choice questions, diagnostic verification, and clinical scenarios, and draw on both native Arabic medical literature and culturally adapted datasets.

Instruction following. IFEval-Ar (El Filali et al., 2025) adapts IFEval (Zhou et al., 2023) to Arabic. It contains 404 prompts covering 23 instruction types, with constraints and verifiers adapted to Arabic rather than translated directly, and responses not written in Arabic are marked incorrect. AraIFBench (Section 6.1.2) adds constraints specific to Arabic orthography and morphology. Both benchmarks are scored programmatically, with prompt-level and instruction-level accuracy under strict and loose evaluation.

C.3.2 Arabic Domain Knowledge

Islamic knowledge and Arab culture. QIAS 2025 (Bouhekif et al., 2025) has two parts: 1,000 questions on Islamic inheritance, which require computing the shares of the heirs, and 700 questions on general Islamic knowledge. IslamicEval 2025 (Mubarak et al., 2025a) contains 247 statements whose attribution to the Qur'an or the Hadith must be verified, and PalmX 2025 (Alwajih et al., 2025) contains 985 questions on Islamic practices, texts, history, and occasions. For culture, AraDiCE-Culture (Mousi et al., 2025) has 180

country-specific questions from six Arab countries, ArabCulture (Sadallah et al., 2025) has 3,482 questions on cultural commonsense across 13 countries, and Alyah (Alkaabi et al., 2026) has 1,173 questions on Emirati culture, heritage, and dialect. FannOrFlop_MCQ (Alghallabi et al., 2025) tests the interpretation of Arabic poetry with 6,984 multiple-choice items, built from nearly 7,000 poem–explanation pairs that span 12 historical eras, 21 core genres, and diverse poetic meters. Arabic Cuisine contains 1,597 multiple-choice questions on the cuisines of the Arab world. The grammatical-analysis and diacritization benchmarks of Arabic Linguistics are introduced in Section 6.1.2, and their construction is described in Appendix C.7.

C.3.3 Dialect Understanding and Generation

Dialectal understanding. QADI (Abdelali et al., 2021) is a corpus of 540,590 tweets from 18 Arab countries that covers Modern Standard Arabic and ten regional varieties; a 3,797-instance subset is used for dialect identification. Dialectal-Arabic-MMLU (Altakrori et al., 2026) contains 3,135 questions adapted into each of five dialects (Egyptian, Saudi, Moroccan, Syrian, and Emirati), for 15,675 question–answer pairs. Diacritized ArabicMMLU (Inoue et al., 2026) is a fully diacritized version of the 14,455 ArabicMMLU questions; because the content of the two versions is identical, comparing them measures robustness to diacritics.

Dialectal generation. FLORES+ (NLLB Team et al., 2024; Burchell et al., 2024) provides 1,012 aligned sentences per variety in English, Modern Standard Arabic, and eight Arabic dialects, and is used for translation. DarijaBench-Transliteration (Shang et al., 2025b) and EgyptianBench-Transliteration (Shang et al., 2025a) test transliteration between Arabic script and Arabizi, a Latin-script romanization, for Moroccan and Egyptian Arabic. Each covers both directions, with 1,000 examples per direction.

C.3.4 Multi-Turn and Long-Context Understanding

Multi-turn. MT-Bench Arabic (Boughorbel and Hawasly, 2023) is an Arabic version of MT-Bench (Zheng et al., 2023) with 80 two-turn conversations, ten in each of eight categories: writing, roleplay, reasoning, mathematics, coding, extraction, STEM, and humanities. The second turn depends on the model’s own first answer, so the benchmark tests whether a model keeps the context and builds on its previous response, including when an early error carries over. Each response is scored from 1 to 10 by gemma-4-26B-A4B-it with the single-answer grading rubric of the benchmark, and for mathematics, reasoning, and coding questions the judge also receives a reference answer. Scores are reported for each turn and on average; the difference between the turns shows how well performance holds when a model continues from its own output.

Long context. AA-LCR (Artificial Analysis, 2025a) contains 100 questions that require integrating information across document sets of about 100K tokens, and a judge grades each response as pass or fail. AA-LCR-Ar, our Arabic version based on version 1.1, is described in Section 6.1.2.

C.4 English Benchmark Details

This section describes the English benchmarks listed in Table 36.

C.4.1 General Capabilities

World knowledge and STEM. We evaluate broad academic knowledge with MMLU-Pro (Wang et al., 2024), which spans the humanities, social sciences, STEM, and professional domains. English STEM reasoning is evaluated with GSM8K (Cobbe et al., 2021), GPQA-Diamond (Rein et al., 2023), MATH-500 (Lightman et al., 2023), and the AIME 2025 and 2026 problem sets (Dekoninck et al., 2026), which together range from multi-step arithmetic and graduate-level science to competition mathematics. All benchmarks are evaluated zero-shot in generative form with benchmark-specific answer formats; GSM8K is run with both direct-answer and chain-of-thought prompting, and for AIME we sample eight responses

per problem and report $\text{pass}@k$ for $k \in \{1, 2, 4, 8\}$. Because exact-match scoring can underestimate correctness when equivalent answers are expressed differently, we score MMLU-Pro, GPQA-Diamond, GSM8K, and MATH-500 with deterministic LLM-based answer verification, using benchmark-specific Gemma judges selected through pilot experiments and manual validation. AIME is scored with automatic mathematical-equivalence checking. The judge prompts are in the judge registry of our evaluation suite (Appendix C.8).

Instruction following. To measure instruction following with explicit output requirements, we use IFEval (Zhou et al., 2023), which comprises 541 single-turn English prompts spanning 25 types of objectively verifiable instructions. These cover lexical, length, formatting, stylistic, and structural constraints, including combinations of multiple constraints within a single prompt. Models are evaluated zero-shot on all 541 prompts.

Responses are scored using the official deterministic IFEval metrics. We report instruction-level and prompt-level accuracy under both strict and loose evaluation. Instruction-level accuracy measures the proportion of individual constraints satisfied, whereas prompt-level accuracy requires all constraints in a prompt to be satisfied. Loose evaluation additionally permits removing asterisks or extraneous text from the first or final line. Scoring requires neither reference answers nor an LLM judge.

C.4.2 Coding, Multi-Turn, and Long Context

Coding. Code generation is evaluated with HumanEval (Chen et al., 2021), 164 function-level problems specified by a signature and docstring, and the 500-problem test split of MBPP (Austin et al., 2021), which requires full programs from natural-language specifications. One solution is generated per problem and executed against the released unit tests. We report $\text{pass}@1$, defined as the proportion of generated solutions that pass all tests.

Multi-turn. We evaluate English multi-turn behavior with three complementary benchmarks that capture different conversational challenges. MT-Bench (Zheng et al., 2023) contains 80 two-turn conversations across eight categories, where the second turn depends on the model’s own first response and therefore measures its ability to maintain context and build on prior output. MultiChallenge (Deshpande et al., 2025) contains 273 human-written conversations of up to roughly 20 turns and evaluates whether models can use information from a longer fixed dialogue history, including earlier context, instructions, prior responses, and requested edits. Multi-IF (He et al., 2024) contains 909 three-turn conversations in which each turn introduces an additional constraint that must be satisfied together with all previous constraints, providing a deterministic measure of instruction retention across turns. MT-Bench responses are scored from 1 to 10 by an LLM judge, with reference answers provided for mathematics, reasoning, and coding questions; MultiChallenge is evaluated using item-specific pass criteria and reported as overall pass rate; and Multi-IF uses deterministic verifiers, with accuracy reported overall and by turn. Where the original benchmarks rely on proprietary judges, we use gemma-4-26B-A4B-it for consistency with the rest of our evaluation.

Long context. Long-context understanding is evaluated with AA-LCR (Artificial Analysis, 2025a), which contains 100 questions requiring reasoning across document sets of about 100K tokens. The benchmark is designed so that answers must be synthesized from multiple sources rather than retrieved directly from a single passage, and responses are graded as correct or incorrect against the reference answer. We use version 1.1, which corrects 16 reference answers and introduces revised grading instructions. Our Arabic adaptation follows the same version and evaluation setup and is described in Section 6.1.2.

C.5 Run-to-Run Variance

Most models in our evaluation are run with sampling-based decoding, using the temperature and other generation settings recommended by their developers. With sampling, a model may answer the same question differently across runs, so a score from a single evaluation can include chance variation and may be misleading when two models perform similarly.

We therefore run each Jais 3 model three times on each benchmark, using identical prompts, generation settings, and judge models while varying only the random seed. The LLM judges decode greedily and therefore do not contribute additional sampling variation.

For each model and benchmark, the score of each run is computed in the same way as for a single evaluation; for benchmarks made up of several subsets, it is the average over all questions of the benchmark. We report the mean of the three run scores together with their standard deviation across the three runs.

We observe low run-to-run variance for Jais 3 overall (Table 46). Almost 90% of results vary by less than 1 point between runs. A few benchmarks show noticeably higher variation. AA-LCR-Ar varies the most, with a mean standard deviation of 3.1 and up to 4.2 points, followed by AraDiCE-Culture (1.3), Legal (HELM) (0.9), and IFEval-Ar (0.9). The most variable benchmarks tend to be small, such as AA-LCR-Ar with 100 questions and AraDiCE-Culture with 180. In contrast, larger benchmarks such as ArabicMMLU, Dialectal-Arabic-MMLU, AraSTEM, GAT Math, and FLORES+ are highly stable, with a mean standard deviation of 0.2 points or less.

C.6 Low-Effort Thinking Results

We also evaluate Jais 3 with low-effort thinking. For a fair comparison, other thinking models are run at their lowest reasoning effort, and models without a thinking mode in their non-thinking (instruct) mode.

C.6.1 Arabic Evaluation

General capabilities results. Table 47 shows that Jais 3 leads every benchmark at 4B, and every benchmark but Arabic GSM8K at 1B, where Jais-3-0.9B is 0.8 points behind Falcon-H1-1.5B-Deep-Instruct. At 7B it leads Legal (HELM), IFEval-Ar and AraIFBench, but gemma-4-12B-it is clearly ahead on the knowledge and math benchmarks, by 7.4 points on ArabicMMLU, 8.6 on AraSTEM, 3.1 on Arabic GSM8K, and more than 20 on GAT Math and MedAraBench. At 32B, gemma-4-31B-it has the best score on every benchmark, and shares the Legal (HELM) lead with Jais-3-32B (72.87). Jais-3-32B is within 3 points of it on Arabic GSM8K, IFEval-Ar and AraIFBench, while MedAraBench has the widest gap (15.3 points); Jais-3-MoVA-36B-A4B also comes within 2.4 points on AraSTEM. At 375B, Jais-3-375B-A23B leads GAT Math (95.6), IFEval-Ar (73.0) and AraIFBench (55.4), and nearly ties the best Arabic GSM8K score (92.88 against 92.92 for Nemotron-3-Ultra-550B-A55B-BF16). It trails Qwen3.5-397B-A17B on ArabicMMLU (by 1.7 points), AraSTEM (1.4) and MedAraBench (9.8), and Nemotron-3-Ultra-550B-A55B-BF16 on Legal (HELM) (3.4).

Domain	World Knowledge	STEM			Medical	Legal	Instruction Following	
	Arabic MMLU	AraSTEM	GAT Math	Arabic GSM8K	MedAra Bench	Legal (HELM)	IFEval-Ar	AraIF Bench
Benchmark	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.
1B Scale								
Falcon-H1-1.5B-Deep-Instruct	48.69	62.49	40.19	65.43	32.32	48.84	49.26	16.20
Hala-1.2B	45.00	48.32	29.63	57.29	32.30	41.09	44.06	14.60
Qwen3.5-2B	44.92	52.18	27.68	35.91	32.98	27.13	40.35	16.90

Continued on next page

Continued from previous page

Domain	World	STEM			Medical	Legal	Instruction	
	Knowledge						Following	
	Arabic		GAT	Arabic	MedAra	Legal		AraIF
Benchmark	MMLU	AraSTEM	Math	GSM8K	Bench	(HELM)	IFEval-Ar	Bench
	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.
tiny-aya-global	55.93	58.48	31.36	59.57	39.49	52.20	58.17	21.50
Jais-3-0.9B	60.01	78.67	49.02	64.68	44.23	55.56	63.37	45.30
4B Scale								
Falcon-H1-3B-Instruct	51.02	64.93	41.29	72.21	34.53	52.97	55.69	18.40
gemma-4-E2B-it	60.81	77.34	54.97	77.32	39.99	55.30	63.86	25.80
granite-4.0-h-3b-ar	59.83	66.49	45.32	67.21	38.13	56.07	43.07	14.70
Qwen3.5-4B	65.30	73.89	36.92	80.42	50.53	57.11	65.84	29.10
Jais-3-3.7B	76.44	88.79	88.16	85.30	56.96	64.08	71.04	49.60
7B Scale								
ALLaM-7B-Instruct-preview	71.21	78.98	38.38	56.81	49.12	61.24	48.02	18.90
Falcon-H1-7B-Instruct	63.31	76.58	45.45	84.99	47.98	57.88	62.38	26.20
Fanar-1-9B-Instruct	65.75	72.98	40.64	75.25	50.71	61.76	44.55	25.80
gemma-4-12B-it	82.32	92.44	75.81	90.31	75.14	64.86	72.03	43.00
gemma-4-E4B-it	70.61	89.48	48.78	86.60	53.29	59.17	67.82	33.80
Jais-2-8B-Chat	70.42	64.84	35.67	72.97	41.58	60.47	61.63	21.10
Qwen3.5-9B	71.64	79.84	48.61	86.83	59.66	61.76	68.07	33.80
Jais-3-7B	74.95	83.87	53.24	87.21	55.00	66.15	72.77	46.50
32B Scale								
Falcon-H1-34B-Instruct	73.66	85.47	63.39	90.24	60.65	63.05	66.83	32.20
gemma-4-31B-it	88.55	96.01	97.73	93.42	85.76	72.87	75.99	55.90
Jais-2-70B-Chat	80.66	75.74	40.67	83.38	52.62	66.15	69.06	36.00
Qwen3.8-27B	79.23	92.96	58.01	92.50	70.21	64.86	68.07	43.60
Jais-3-32B	81.38	89.53	91.93	93.03	70.45	72.87	74.75	53.00
Jais-3-MoVA-36B-A4B	83.78	93.64	77.13	90.81	66.36	66.15	75.00	53.30
375B Scale								
MiniMax-M3	77.44	87.92	70.34	90.98	67.40	70.80	66.34	36.00
Nemotron-3-Ultra-550B-A55B-BF16	81.24	89.49	70.52	92.92	72.29	74.42	71.78	47.60
Qwen3.5-397B-A17B	84.42	92.92	78.79	92.57	82.46	68.99	72.52	51.00
Qwen3.8-Flash-Next	82.63	92.61	67.07	92.80	74.99	70.03	67.82	45.00
Jais-3-375B-A23B	82.77	91.48	95.58	92.88	72.70	71.06	73.02	55.40

Table 47: Arabic general capabilities results with low-effort thinking (%). Models without a thinking mode use their non-thinking (instruct) mode. Within each model size, the best score in each column is shown in bold.

Arabic domain knowledge results. Table 48 shows that Jais 3 leads all ten benchmarks at 1B, and nine of ten at 4B, where Jais-3-3.7B is only 0.9 points behind gemma-4-E2B-it on Arabic Linguistics. At 7B it leads QIAS-Inheritance, IslamicEval, PalmX and FannOrFlop_MCQ and ties gemma-4-12B-it on Alyah, but trails it on QIAS-Knowledge (by 6.5 points), Arabic Linguistics (by 5.2), ArabCulture (by 1.6) and Arabic Cuisine (by 1.1), and ALLaM-7B-Instruct-preview on AraDiCE-Culture (by 9.5). At 32B, Jais-3-32B leads AraDiCE-Culture and Alyah, and Jais-2-70B-Chat leads FannOrFlop_MCQ. On the other benchmarks, Jais-3-32B is within 3 points of gemma-4-31B-it, except on QIAS-Inheritance (26.2 points behind) and Arabic Linguistics (17.3). At 375B, Jais-3-375B-A23B leads QIAS-Inheritance, QIAS-Knowledge, IslamicEval, ArabCulture, Alyah and Arabic Cuisine. At 32B and 375B, low effort costs the most on QIAS-Inheritance, where Jais-3-32B scores 60.5 against 95.9 at high effort, and Jais-3-

Domain	World Knowledge	STEM				Legal	Medical	
	Arabic MMLU Acc.	AraSTEM Acc.	GAT Math Acc.	Arabic GSM8K Acc.		Legal (HELM) Acc.	MedAra Bench Acc.	
Benchmark								
Jais-3-0.9B	70.28 $\pm$ 0.22	81.41 $\pm$ 0.22	87.73 $\pm$ 0.21	79.82 $\pm$ 0.51		60.64 $\pm$ 0.30	48.96 $\pm$ 1.06	
Jais-3-3.7B	83.20 $\pm$ 0.24	91.51 $\pm$ 0.13	96.40 $\pm$ 0.23	91.59 $\pm$ 0.18		65.12 $\pm$ 1.18	73.16 $\pm$ 0.30	
Jais-3-7B	85.27 $\pm$ 0.03	92.54 $\pm$ 0.09	96.59 $\pm$ 0.20	92.97 $\pm$ 0.29		66.58 $\pm$ 1.22	77.63 $\pm$ 0.51	
Jais-3-32B	88.42 $\pm$ 0.23	95.19 $\pm$ 0.03	98.50 $\pm$ 0.07	93.80 $\pm$ 0.12		73.21 $\pm$ 1.04	86.72 $\pm$ 0.15	
Jais-3-MoVA-36B-A4B	87.70 $\pm$ 0.06	94.72 $\pm$ 0.14	97.46 $\pm$ 0.08	92.25 $\pm$ 0.10		69.51 $\pm$ 1.61	85.32 $\pm$ 0.27	
Jais-3-375B-A23B	89.08 $\pm$ 0.24	95.52 $\pm$ 0.10	98.45 $\pm$ 0.14	87.46 $\pm$ 0.24		72.44 $\pm$ 0.15	88.84 $\pm$ 0.62	

Domain	Islamic Knowledge	Arab Culture					Arabic Linguistics	Instruction Following	Long Context
	Islamic QA Acc.	AraDiCE Culture Acc.	Arab Culture Acc.	Alyah Acc.	Fann OrFlop Acc.	Arabic Cuisine Acc.	Arabic Ling. Acc.	IFEval Ar Acc.	AA-LCR Ar Acc.
Benchmark									
Jais-3-0.9B	69.75 $\pm$ 0.71	71.51 $\pm$ 1.68	73.74 $\pm$ 0.73	73.09 $\pm$ 1.18	59.28 $\pm$ 0.33	72.95 $\pm$ 1.11	85.00 $\pm$ 0.35	71.53 $\pm$ 2.03	–
Jais-3-3.7B	82.18 $\pm$ 0.56	78.77 $\pm$ 0.56	84.23 $\pm$ 0.59	80.45 $\pm$ 0.39	66.18 $\pm$ 0.38	82.77 $\pm$ 0.62	92.23 $\pm$ 0.47	79.62 $\pm$ 0.62	66.00 $\pm$ 3.61
Jais-3-7B	85.15 $\pm$ 0.45	81.01 $\pm$ 2.56	85.17 $\pm$ 0.24	80.22 $\pm$ 0.35	68.66 $\pm$ 0.46	84.83 $\pm$ 0.46	93.24 $\pm$ 0.46	79.21 $\pm$ 1.24	66.33 $\pm$ 4.16
Jais-3-32B	90.02 $\pm$ 0.61	87.52 $\pm$ 0.65	88.24 $\pm$ 0.42	84.05 $\pm$ 0.43	72.02 $\pm$ 0.23	86.77 $\pm$ 0.43	94.35 $\pm$ 0.56	78.80 $\pm$ 0.38	76.67 $\pm$ 2.08
Jais-3-MoVA-36B-A4B	89.52 $\pm$ 0.27	87.34 $\pm$ 0.85	88.76 $\pm$ 0.15	83.28 $\pm$ 0.57	71.46 $\pm$ 0.16	88.82 $\pm$ 0.50	93.88 $\pm$ 0.18	78.88 $\pm$ 0.76	74.00 $\pm$ 2.65
Jais-3-375B-A23B	91.22 $\pm$ 0.35	88.09 $\pm$ 1.40	89.67 $\pm$ 0.28	87.74 $\pm$ 0.31	73.27 $\pm$ 0.28	90.14 $\pm$ 0.33	94.83 $\pm$ 0.30	77.48 $\pm$ 0.43	–

Domain	Dialect Understanding and Generation					Safety			
	QADI Acc.	FLORES+ Judge	Dialect. MMLU Acc.	Arabic MMLU Acc.	MMLU diacr. Acc.	AraSafe Refusal	Arabic Regional Safety		
Benchmark							General	Gov.	Opp.
Jais-3-0.9B	66.12 $\pm$ 0.69	82.42 $\pm$ 0.03	63.59 $\pm$ 0.33	70.28 $\pm$ 0.22	64.94 $\pm$ 0.54	93.27 $\pm$ 0.17	98.92 $\pm$ 0.36	98.62 $\pm$ 0.08	98.71 $\pm$ 0.31
Jais-3-3.7B	69.00 $\pm$ 0.30	89.04 $\pm$ 0.03	81.14 $\pm$ 0.09	83.20 $\pm$ 0.24	80.68 $\pm$ 0.18	92.66 $\pm$ 0.08	99.52 $\pm$ 0.55	96.45 $\pm$ 0.60	98.36 $\pm$ 0.20
Jais-3-7B	69.34 $\pm$ 0.15	90.53 $\pm$ 0.05	83.39 $\pm$ 0.31	85.27 $\pm$ 0.03	83.17 $\pm$ 0.15	92.19 $\pm$ 0.29	99.88 $\pm$ 0.21	98.71 $\pm$ 0.15	99.47 $\pm$ 0.35
Jais-3-32B	70.42 $\pm$ 0.10	91.57 $\pm$ 0.02	88.64 $\pm$ 0.17	88.42 $\pm$ 0.23	87.90 $\pm$ 0.19	92.42 $\pm$ 0.63	99.88 $\pm$ 0.21	99.47 $\pm$ 0.23	99.91 $\pm$ 0.08
Jais-3-MoVA-36B-A4B	69.61 $\pm$ 0.61	90.83 $\pm$ 0.02	87.42 $\pm$ 0.03	87.70 $\pm$ 0.06	86.90 $\pm$ 0.18	93.20 $\pm$ 0.26	100.00 $\pm$ 0.00	99.11 $\pm$ 0.20	99.69 $\pm$ 0.20
Jais-3-375B-A23B	71.48 $\pm$ 0.37	91.71 $\pm$ 0.16	89.53 $\pm$ 0.02	89.08 $\pm$ 0.24	87.95 $\pm$ 0.28	93.06 $\pm$ 0.58	100.00 $\pm$ 0.00	98.91 $\pm$ 0.25	99.73 $\pm$ 0.13

Table 46: Arabic results for the Jais 3 models. Mean over three independent runs; subscripts: standard deviation across the runs. Scores on the audited benchmarks exclude the removed items. All results use high-effort thinking. Islamic QA is the unweighted mean of QIAS-Inheritance, QIAS-Knowledge, IslamicEval 2025, and PalmX 2025; – marks results that are not available.

375B-A23B 70.3 against 95.4.

Domain	Islamic Knowledge				Arab Culture				Arabic Linguistics	
	QIAS Inherit. Acc.	QIAS Knowl. Acc.	Islamic Eval Acc.	PalmX Acc.	AraDiCE Culture Acc.	Arab Culture Acc.	Alyah Acc.	FannOrFlop _MCQ Acc.	Arabic Cuisine Acc.	Arabic Ling. Acc.
Benchmark										
1B Scale										
Falcon-H1-1.5B-Deep-Instruct	32.07	56.02	42.42	62.13	49.16	61.69	54.75	38.54	52.35	33.27
Hala-1.2B	16.15	41.51	36.36	49.58	30.73	60.63	53.20	31.34	49.48	32.56

Continued on next page

Continued from previous page

Domain	Islamic Knowledge				Arab Culture				Arabic Linguistics	
	QIAS Inherit.	QIAS Knowl.	Islamic Eval	PalmX	AraDiCE Culture	Arab Culture	FannOrFlop _MCQ	Arabic Cuisine	Arabic Ling.	
Benchmark	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.
Qwen3.5-2B	17.93	41.15	38.53	58.54	37.99	50.50	50.00	35.40	47.45	37.50
tiny-aya-global	23.05	54.72	27.27	71.62	55.31	68.61	68.13	48.50	60.64	38.16
Jais-3-0.9B	46.33	62.70	71.43	74.05	58.10	71.60	72.97	51.81	65.47	62.10
4B Scale										
Falcon-H1-3B-Instruct	27.51	54.28	44.59	66.56	45.81	64.18	58.89	41.09	54.37	42.05
gemma-4-E2B-it	42.20	60.81	59.31	73.52	45.25	70.49	70.03	57.37	63.32	69.01
granite-4.0-h-3b-ar	18.15	66.47	22.51	57.49	53.63	68.41	65.98	48.76	58.55	49.34
Qwen3.5-4B	30.62	69.09	58.01	76.69	60.89	70.93	63.39	48.28	65.14	55.68
Jais-3-3.7B	53.67	75.62	77.49	86.60	79.89	82.81	79.97	63.95	81.66	68.15
7B Scale										
ALLaM-7B-Instruct-preview	31.18	72.86	78.36	85.86	79.33	77.65	77.46	66.67	71.34	67.46
Falcon-H1-7B-Instruct	43.99	65.75	61.90	74.26	59.22	72.43	65.72	51.38	64.95	47.53
Fanar-1-9B-Instruct	36.30	64.88	69.70	80.27	67.04	77.30	69.60	63.27	68.86	57.59
gemma-4-12B-it	70.04	82.44	76.19	85.23	72.63	86.18	81.43	65.69	83.22	73.89
gemma-4-E4B-it	45.88	69.81	68.83	81.33	68.72	77.03	74.78	61.31	72.58	70.46
Jais-2-8B-Chat	29.06	67.49	56.71	82.81	67.60	74.60	76.42	60.50	71.61	49.20
Qwen3.5-9B	39.53	72.28	68.40	80.17	64.25	76.86	68.39	54.31	71.87	61.75
Jais-3-7B	85.75	75.91	80.95	87.24	69.83	84.54	81.43	66.75	82.11	68.67
32B Scale										
Falcon-H1-34B-Instruct	56.24	73.44	70.13	84.70	75.42	79.67	71.76	62.35	74.41	57.44
gemma-4-31B-it	86.64	84.91	87.88	93.14	83.80	90.47	86.61	69.57	88.64	91.23
Jais-2-70B-Chat	42.20	78.96	65.80	90.72	80.45	82.08	84.46	73.62	80.42	64.10
Qwen3.8-27B	53.79	80.26	72.29	88.40	75.98	83.46	78.15	64.32	79.77	74.92
Jais-3-32B	60.47	83.60	86.15	91.14	86.03	87.47	87.48	69.34	88.05	73.96
Jais-3-MoVA-36B-A4B	62.03	82.00	82.68	89.35	82.12	87.86	84.72	68.15	87.14	76.63
375B Scale										
MiniMax-M3	52.34	78.37	75.76	89.14	75.98	80.05	75.73	61.12	78.26	69.97
Nemotron-3-Ultra-550B-A55B-BF16	49.55	82.29	85.28	91.14	82.68	86.71	76.60	66.75	84.73	75.41
Qwen3.5-397B-A17B	54.68	84.47	82.25	94.20	84.36	87.15	82.90	72.42	85.25	83.03
Qwen3.8-Flash-Next	54.23	85.20	64.07	91.77	80.45	86.71	81.26	69.75	85.12	84.39
Jais-3-375B-A23B	70.27	86.07	87.45	91.24	79.89	88.94	86.18	70.82	88.97	83.72

Table 48: Arabic domain knowledge results with low-effort thinking (%). Models without a thinking mode use their non-thinking (instruct) mode. Within each model size, the best score in each column is shown in bold.

Dialect understanding and generation results. Table 49 shows that Jais 3 leads QADI at every scale (55.0 at 1B, 55.9 at 4B, 62.9 at 7B, 69.2 at 32B, and 70.5 at 375B). At 1B and 4B it is best on six of the seven columns; the exceptions are Dialectal-Arabic-MMLU at 1B, where it is 0.2 points behind *tiny-aya-global*, and Darija transliteration at 4B, where it is 4.7 points behind *gemma-4-E2B-it*. At 7B and 32B the translation and transliteration scores are at or near the top (for example, *Jais-3-7B* leads FLORES+ with 89.1 and Darija transliteration with 93.5), but the MMLU-style benchmarks favor Gemma: *gemma-4-12B-it* is ahead of *Jais-3-7B* by 7–12 points, and *gemma-4-31B-it* is ahead of *Jais-3-MoVA-36B-A4B* by about 5 points on ArabicMMLU and 11 points on Dialectal-Arabic-MMLU. At 375B, *Jais-3-375B-A23B* leads QADI, FLORES+ and both transliteration tasks, and trails *Qwen3.5-397B-A17B* by about 2 points on the

three MMLU benchmarks.

Domain	Dialect Understanding and Generation							
	Benchmark		Dialect.	Arabic	MMLU		Translit.	Translit.
		QADI	MMLU	MMLU	diacr.	FLORES+	EG	DR
	Acc.	Acc.	Acc.	Acc.	Judge	Judge	Judge	
1B Scale								
Falcon-H1-1.5B-Deep-Instruct	23.39	42.26	48.69	34.68	46.78	30.53	42.35	
Hala-1.2B	22.97	40.33	45.00	31.28	42.25	34.94	25.72	
Qwen3.5-2B	23.01	37.90	44.92	34.02	42.59	30.49	40.05	
tiny-aya-global	38.79	47.68	55.93	45.64	63.79	38.37	45.75	
Jais-3-0.9B	54.96	47.50	60.01	52.15	82.36	73.17	80.75	
4B Scale								
Falcon-H1-3B-Instruct	28.52	43.81	51.02	34.62	53.84	41.26	39.33	
gemma-4-E2B-it	40.74	50.06	60.81	53.47	73.78	61.04	84.13	
granite-4.0-h-3b-ar	42.19	50.71	59.83	53.25	48.72	35.33	40.66	
Qwen3.5-4B	30.76	56.70	65.30	54.32	61.55	47.21	60.37	
Jais-3-3.7B	55.91	62.94	76.44	74.05	86.45	80.06	79.45	
7B Scale								
ALLaM-7B-Instruct-preview	42.77	56.61	71.21	61.85	74.47	39.71	31.81	
Falcon-H1-7B-Instruct	38.08	58.46	63.31	52.33	59.76	42.47	48.74	
Fanar-1-9B-Instruct	46.19	58.21	65.75	54.86	68.35	50.18	65.67	
gemma-4-12B-it	55.97	77.61	82.32	80.69	86.80	84.65	92.83	
gemma-4-E4B-it	52.01	63.91	70.61	66.04	79.32	80.50	91.43	
Jais-2-8B-Chat	48.38	54.59	70.42	54.66	79.40	88.76	76.73	
Qwen3.5-9B	27.13	62.85	71.64	63.27	67.03	50.45	76.26	
Jais-3-7B	62.94	66.15	74.95	70.75	89.10	87.63	93.45	
32B Scale								
Falcon-H1-34B-Instruct	50.28	71.47	73.66	69.09	70.58	54.70	75.75	
gemma-4-31B-it	68.08	87.04	88.55	87.42	91.54	90.42	96.03	
Jais-2-70B-Chat	65.08	67.56	80.66	74.05	83.83	90.60	92.75	
Qwen3.8-27B	46.91	74.87	79.23	76.85	75.46	58.45	82.68	
Jais-3-32B	69.16	75.28	81.38	79.17	90.74	90.59	95.39	
Jais-3-MoVA-36B-A4B	68.98	75.92	83.78	82.64	90.19	89.31	95.05	
375B Scale								
MiniMax-M3	40.11	70.94	77.44	74.08	81.31	76.21	75.67	
Nemotron-3-Ultra-550B-A55B-BF16	57.94	78.76	81.24	78.04	78.62	76.21	89.09	
Qwen3.5-397B-A17B	58.49	82.91	84.42	82.53	84.04	77.96	93.02	
Qwen3.8-Flash-Next	57.23	81.34	82.63	79.43	81.30	77.53	86.85	
Jais-3-375B-A23B	70.48	80.97	82.77	80.34	91.20	90.69	95.40	

Table 49: Arabic dialect understanding and generation results with low-effort thinking (%). Models without a thinking mode use their non-thinking (instruct) mode. Within each model size, the best score in each column is shown in bold.

Multi-turn and long context understanding results. Table 50 shows that Jais 3 leads MT-Bench Arabic at the 1B and 4B scales: Jais-3-0.9B outperforms the next best 1B model by about 15 points, and Jais-3-3.7B is about 4 points ahead of gemma-4-E2B-it. At larger scales the gap to the best open model is small but not closed: Jais-3-32B is within 3 points of gemma-4-31B-it, Jais-3-375B-A23B is within 3 points of Qwen3.8-Flash-Next, and Jais-3-7B trails gemma-4-12B-it by about 7 points.

Long-context accuracy on AA-LCR-Ar is where low effort costs Jais 3 the most. Scores are far below those at high effort (for example Jais-3-3.7B scores 19 against 68, and Jais-3-375B-A23B 25 against 77), and Jais 3 trails the strongest baselines in each class: Jais-3-3.7B scores 19 against 33 for Qwen3.5-4B, Jais-3-32B 24 against 63 for Qwen3.8-27B, and Jais-3-375B-A23B 25 against 74 for Nemotron-3-Ultra-550B-A55B-BF16. At 7B, Jais-3-7B scores 30, behind Qwen3.5-9B but ahead of gemma-4-12B-it.

Domain	Multi-turn			Long Context	Domain	Multi-turn			Long Context
Benchmark	MT-Bench Arabic			AA-LCR-Ar	Benchmark	MT-Bench Arabic			AA-LCR-Ar
	Turn 1	Turn 2	Avg.	Acc.		Turn 1	Turn 2	Avg.	Acc.
1B Scale					32B Scale				
Falcon-H1-1.5B-Deep-Instruct	40.75	30.50	35.62	–	Falcon-H1-34B-Instruct	81.38	70.62	76.00	14.00
Hala-1.2B	54.12	27.00	40.56	–	gemma-4-31B-it	98.12	98.99	98.55	46.00
Qwen3.5-2B	24.25	18.88	21.56	12.00	Jais-2-70B-Chat	77.75	66.88	72.31	–
tiny-aya-global	59.00	38.12	48.56	–	Qwen3.8-27B	96.75	93.75	95.25	63.00
Jais-3-0.9B	72.62	54.38	63.50	–	Jais-3-32B	98.38	93.25	95.81	24.00
4B Scale					Jais-3-MoVA-36B-A4B	95.38	88.50	91.94	28.00
Falcon-H1-3B-Instruct	58.75	51.25	55.00	–	375B Scale				
gemma-4-E2B-it	86.12	79.11	82.64	–	MiniMax-M3	95.12	93.38	94.25	56.00
granite-4.0-h-3b-ar	54.38	33.75	44.06	–	Nemotron-3-Ultra-550B-A55B-BF16	94.75	96.62	95.69	74.00
Qwen3.5-4B	80.25	74.75	77.50	33.00	Qwen3.5-397B-A17B	97.00	96.88	96.94	57.00
Jais-3-3.7B	88.50	85.50	87.00	19.00	Qwen3.8-Flash-Next	99.62	97.25	98.44	66.00
7B Scale					Jais-3-375B-A23B	98.12	93.12	95.62	25.00
ALLaM-7B-Instruct-preview	62.50	44.00	53.25	–					
Falcon-H1-7B-Instruct	69.88	61.27	65.60	9.00					
Fanar-1-9B-Instruct	76.50	62.12	69.31	–					
gemma-4-12B-it	98.00	97.00	97.50	23.00					
gemma-4-E4B-it	97.25	94.00	95.62	–					
Jais-2-8B-Chat	74.12	56.00	65.06	–					
Qwen3.5-9B	91.50	87.75	89.62	37.00					
Jais-3-7B	94.75	87.12	90.94	30.00					

Table 50: Arabic multi-turn and long-context results with low-effort thinking (%). Models without a thinking mode use their non-thinking (instruct) mode. Within each model size, the best score in each column is shown in bold. MT-Bench Arabic turns are the judge’s 1–10 ratings $\times 10$ and Avg. is their mean; – marks results that are not available.

C.6.2 English Evaluation

General capabilities results. Table 51 shows that at low effort Jais 3 is strongest on math from 4B upward. At 1B, Jais-3-0.9B trails Falcon-H1-1.5B-Deep-Instruct on every benchmark except AIME 2025, where they tie. At 4B, Jais-3-3.7B leads GPQA-Diamond, GSM8K, GSM8K-CoT, MATH-500 and AIME 2025, and is within 2 points of the best model on the rest. At 7B, Jais-3-7B leads GPQA-Diamond and MATH-500, ties Qwen3.5-9B on AIME 2025, and is within 4 points of the best model elsewhere. At 32B, Jais-3-32B leads MATH-500 and AIME 2026 and is within 3 points of gemma-4-31B-it on most benchmarks, but trails it by 15 points on GPQA-Diamond and Qwen3.8-27B by 7 points on AIME 2025. At 375B, Jais-3-375B-A23B leads GPQA-Diamond, GSM8K, GSM8K-CoT and IFEval, trails Qwen3.5-397B-A17B on MMLU-Pro, and trails Nemotron-3-Ultra-550B-A55B-BF16 on AIME by 8–13 points. Compared with high effort, AIME, GPQA-Diamond and IFEval drop for Jais-3-375B-A23B (by 19–23, 8 and 5 points), while GSM8K and MATH-500 are unchanged.

Domain	World Knowledge & STEM							Instruction Following
Benchmark	MMLU Pro Acc.	GPQA Diamond Acc.	GSM8K Acc.	GSM8K CoT Acc.	MATH 500 Acc.	AIME 25 pass@1	AIME 26 pass@1	IFEval Acc.
1B Scale								
Falcon-H1-1.5B-Deep-Instruct	54.45	43.43	91.96	88.63	79.40	12.92	8.75	81.89
Hala-1.2B	21.02	24.75	54.17	55.00	34.00	0.00	3.33	43.25
Qwen3.5-2B	56.10	30.81	71.57	78.05	75.80	15.83	20.00	61.18
tiny-aya-global	32.08	31.31	70.81	67.70	28.60	1.25	0.00	73.38
Jais-3-0.9B	48.17	34.34	78.54	78.43	71.40	12.92	6.25	69.13
4B Scale								
Falcon-H1-3B-Instruct	57.41	30.30	90.30	91.17	79.40	17.08	8.75	83.36
gemma-4-E2B-it	61.12	44.95	90.30	90.07	88.40	29.58	35.83	79.30
granite-4.0-h-3b-ar	38.77	33.84	77.60	81.58	52.60	0.00	0.00	59.33
Qwen3.5-4B	76.35	42.42	93.63	92.95	94.20	51.67	60.42	81.52
Jais-3-3.7B	74.77	62.12	95.45	95.56	96.80	54.17	58.75	81.70
7B Scale								
ALLaM-7B-Instruct-preview	33.78	26.26	66.15	63.91	17.20	0.00	0.00	62.85
Falcon-H1-7B-Instruct	65.87	34.34	91.74	89.08	81.00	15.00	9.58	84.29
Fanar-1-9B-Instruct	51.01	36.87	85.75	86.05	53.60	0.42	0.42	60.26
gemma-4-12B-it	78.71	63.13	96.25	96.13	96.80	56.25	67.92	89.09
gemma-4-E4B-it	70.27	52.53	93.10	92.57	93.80	37.50	45.83	84.10
Jais-2-8B-Chat	34.30	24.24	75.97	79.08	50.40	0.42	0.00	69.32
Qwen3.5-9B	80.59	45.96	94.77	95.15	83.20	62.50	69.17	84.66
Jais-3-7B	77.01	64.65	95.19	95.38	97.00	62.50	68.75	86.14
32B Scale								
Falcon-H1-34B-Instruct	69.40	47.47	94.09	94.92	86.00	19.17	17.92	86.51
gemma-4-31B-it	85.54	81.31	97.19	97.12	98.60	69.17	78.33	90.94
Jais-2-70B-Chat	54.89	37.88	86.85	88.86	71.20	5.00	5.83	76.89
Qwen3.8-27B	83.91	80.30	96.66	96.82	98.80	71.67	79.58	82.44
Jais-3-32B	82.74	66.16	96.97	97.04	99.80	64.58	81.25	90.02
Jais-3-MoVA-36B-A4B	81.69	72.22	97.04	96.78	98.60	67.50	72.08	87.43
375B Scale								
MiniMax-M3	80.43	56.06	91.66	95.91	90.40	15.42	22.08	78.93
Nemotron-3-Ultra-550B-A55B-BF16	83.77	46.46	90.90	96.36	97.00	88.33	87.08	87.06
Qwen3.5-397B-A17B	87.38	55.05	96.89	97.04	99.40	83.75	85.83	87.80
Qwen3.8-Flash-Next	86.82	58.59	97.19	97.19	99.40	74.17	80.83	85.77
Jais-3-375B-A23B	83.87	80.81	97.57	97.38	98.20	75.42	79.17	89.83

Table 51: English knowledge, STEM and instruction-following results with low-effort thinking (%). Models without a thinking mode use their non-thinking (instruct) mode. Within each model size, the best score in each column is shown in bold.

Coding, multi-turn, and long context results. Table 52 shows that Jais 3 is strong on coding and instruction following but weak on MultiChallenge at low effort. At 1B, Jais-3-0.9B trails Falcon-H1-1.5B-Deep-Instruct by 5.5 points on HumanEval, 7.4 on Multi-IF and 9.9 on MT-Bench, but leads AA-LCR (25.0). At 4B, Jais-3-3.7B leads HumanEval, MBPP, Multi-IF and MT-Bench. At 7B, Jais-3-7B leads MBPP (91.2) and AA-LCR (54.0) and ties for the best HumanEval score. At 32B, Jais-3-32B leads MBPP (94.2) and is within 1.3 points of gemma-4-31B-it on HumanEval and Multi-IF, while Jais-3-

MoVA-36B-A4B is second on AA-LCR (64.0 against 70.0 for Qwen3.8-27B). At 375B, Jais-3-375B-A23B leads MBPP (95.0) and Multi-IF (87.0). The main weakness is MultiChallenge, where Jais 3 trails the best model by 12–24 points from 4B upward (for example 32.1 against 55.7 for Qwen3.5-397B-A17B at 375B). At 375B it also trails Qwen3.8-Flash-Next on AA-LCR by 17 points (54 against 71). Low effort lowers long-context and multi-turn accuracy substantially compared with high effort (for Jais-3-3.7B, AA-LCR falls from 71 to 38, and for Jais-3-375B-A23B, MultiChallenge falls from 57.4 to 32.1).

Domain	Coding		Multi-turn			Long Context
	Human	MBPP	Multi	Multi-IF	MT-	AA-LCR
	Benchmark	Eval	Challenge	3-turn	Bench	
		Acc.	Acc.	Acc.	2-turn	Acc.
1B Scale						
Falcon-H1-1.5B-Deep-Instruct	73.78	59.20	14.60	71.78	74.56	–
Hala-1.2B	32.93	29.60	11.38	35.38	32.69	–
Qwen3.5-2B	27.44	23.80	25.70	57.43	59.81	16.00
tiny-aya-global	46.34	29.60	12.48	64.45	58.69	–
Jais-3-0.9B	68.29	58.20	23.87	64.35	64.62	25.00
4B Scale						
Falcon-H1-3B-Instruct	75.61	63.20	9.33	75.59	81.19	–
gemma-4-E2B-it	86.59	67.00	32.80	77.63	92.12	19.00
granite-4.0-h-3b-ar	70.12	59.40	17.08	52.20	61.25	–
Qwen3.5-4B	88.41	70.00	40.12	77.63	91.75	39.00
Jais-3-3.7B	95.12	88.20	27.62	79.11	93.69	38.00
7B Scale						
ALLaM-7B-Instruct-preview	21.95	12.20	12.16	53.80	54.50	–
Falcon-H1-7B-Instruct	82.32	70.40	8.45	79.65	88.12	–
Fanar-1-9B-Instruct	–	–	17.85	58.70	66.38	–
gemma-4-12B-it	97.56	86.60	49.63	86.57	97.04	41.00
gemma-4-E4B-it	90.24	69.40	38.54	83.42	96.44	30.00
Jais-2-8B-Chat	32.93	20.00	15.98	53.23	61.44	–
Qwen3.5-9B	90.85	74.40	42.18	80.85	95.75	50.00
Jais-3-7B	97.56	91.20	36.99	81.68	94.06	54.00
32B Scale						
Falcon-H1-34B-Instruct	87.80	77.00	22.38	84.59	94.56	14.00
gemma-4-31B-it	97.56	89.20	60.84	88.58	98.25	43.00
Jais-2-70B-Chat	57.93	41.00	28.16	62.01	75.82	–
Qwen3.8-27B	96.34	85.40	51.83	78.65	96.19	70.00
Jais-3-32B	96.34	94.20	38.26	87.28	94.06	32.00
Jais-3-MoVA-36B-A4B	96.95	91.80	35.55	83.24	96.56	64.00
375B Scale						
MiniMax-M3	96.34	86.80	31.69	79.31	95.66	67.00
Nemotron-3-Ultra-550B-A55B-BF16	97.56	83.40	55.59	82.27	97.12	66.00
Qwen3.5-397B-A17B	97.56	93.00	55.72	86.04	98.62	70.00
Qwen3.8-Flash-Next	95.73	87.40	51.54	81.14	97.00	71.00
Jais-3-375B-A23B	97.56	95.00	32.12	87.03	94.50	54.00

Table 52: English coding, multi-turn and long-context results with low-effort thinking (%). Models without a thinking mode use their non-thinking (instruct) mode. Within each model size, the best score in each column is shown in bold. – marks results that were not available or not verified at the time of writing.

Factuality and hallucination results. Table 53 shows that at low effort Jais 3 is strongest on HalluScore but trails on raw factual recall at larger scales. It leads HalluScore at 1B, 4B and 375B (for example 64.6 against 63.5 for Nemotron-3-Ultra-550B-A55B-BF16 at 375B), and is within 1.3 and 3.1 points of the best model at 7B and 32B. On AA-Omniscience accuracy it is within 3 points of the leader at every scale (at 32B through Jais-3-MoVA-36B-A4B). The main weakness is open-domain recall on SimpleQA, where Jais-3-375B-A23B trails Qwen3.5-397B-A17B by 39 points in F-score on SimpleQA-Verified (13.4 against 52.7) and 28 points on SimpleQA-Verified-Ar. SimpleQA non-hallucination rates are near zero for all Jais 3 and Qwen models, which nearly always attempt an answer at low effort, while the Gemma models and MiniMax-M3 abstain much more often. Compared with high effort, Jais 3 answers far more often: for Jais-3-375B-A23B, non-hallucination on SimpleQA-Verified falls from 93.5 to 0.6, F-score rises from 5.5 to 13.4, and HalluScore falls from 70.9 to 64.6.

Domain	Factuality & Hallucination						
Benchmark	SimpleQA Verified		SimpleQA Verified-Ar		AA-Omniscience		Hallu Score
	F-score	Non-hall.	F-score	Non-hall.	Acc.	Non-hall.	Acc.
1B Scale							
Falcon-H1-1.5B-Deep-Instruct	3.56	3.21	2.04	3.78	9.50	1.29	5.96
Hala-1.2B	3.39	5.58	3.05	3.20	4.33	5.40	10.14
Qwen3.5-2B	3.52	0.93	1.01	2.32	4.67	2.45	12.40
tiny-aya-global	4.23	1.36	3.05	3.20	6.50	2.14	20.13
Jais-3-0.9B	4.86	2.63	3.63	1.45	7.00	61.65	23.03
4B Scale							
Falcon-H1-3B-Instruct	3.97	3.64	2.33	2.97	9.83	3.88	13.20
gemma-4-E2B-it	1.15	95.77	0.38	94.09	7.67	18.23	24.32
granite-4.0-h-3b-ar	6.46	2.03	4.53	1.36	9.00	4.58	16.43
Qwen3.5-4B	8.01	0.33	5.03	1.05	12.67	2.67	26.41
Jais-3-3.7B	6.25	1.71	5.51	0.53	12.33	22.24	38.97
7B Scale							
ALLaM-7B-Instruct-preview	4.79	3.99	4.09	4.48	8.83	19.56	34.78
Falcon-H1-7B-Instruct	6.06	2.13	3.47	4.45	11.17	17.64	14.65
Fanar-1-9B-Instruct	7.37	7.75	5.88	6.57	13.67	16.22	42.35
gemma-4-12B-it	5.21	40.61	4.54	24.79	14.67	24.22	46.38
gemma-4-E4B-it	1.94	87.66	1.28	90.84	9.67	40.77	33.66
Jais-2-8B-Chat	3.42	1.45	4.40	4.91	6.50	3.57	31.24
Qwen3.5-9B	10.11	0.22	7.01	0.43	15.50	1.58	33.17
Jais-3-7B	7.01	0.43	6.33	0.96	12.67	49.24	45.09
32B Scale							
Falcon-H1-34B-Instruct	9.95	5.65	6.59	9.39	20.50	19.50	42.35
gemma-4-31B-it	9.57	22.21	8.00	10.93	21.17	14.16	63.61
Jais-2-70B-Chat	7.29	2.59	4.62	5.55	11.17	16.32	49.28
Qwen3.8-27B	7.40	0.00	6.01	0.43	13.00	68.58	41.87
Jais-3-32B	8.83	0.66	8.53	0.77	16.50	58.28	60.55
Jais-3-MoVA-36B-A4B	8.43	0.76	8.53	0.87	20.83	34.11	55.72
375B Scale							
MiniMax-M3	20.69	61.51	15.75	40.11	9.50	94.48	54.43
Nemotron-3-Ultra-550B-A55B-BF16	33.12	0.15	28.03	1.25	28.33	50.23	63.45
Qwen3.5-397B-A17B	52.70	0.00	42.00	0.00	30.83	25.30	61.84
Qwen3.8-Flash-Next	24.80	0.00	17.70	0.00	24.17	3.96	57.17

Continued on next page

Continued from previous page

Domain	Factuality & Hallucination						
Benchmark	SimpleQA Verified		SimpleQA Verified-Ar		AA-Omniscience		Hallu Score
	F-score	Non-hall.	F-score	Non-hall.	Acc.	Non-hall.	Acc.
Jais-3-375B-A23B	13.43	0.58	14.03	0.47	28.33	52.79	64.57

Table 53: Arabic and English factuality and hallucination results with low-effort thinking (%). Models without a thinking mode use their non-thinking (instruct) mode. Within each model size, the best score in each column is shown in bold.

C.7 Construction of the Arabic Linguistics Benchmarks

This section details the construction of the Arabic Linguistics – I’rab (Parsing) and Arabic Linguistics – Tashkeel (Target-Word Diacritization) benchmarks. Both benchmarks evaluate targeted, context-dependent decisions on marked words within sentences sourced from the Arabic Linguistic Pipeline (ALP) corpus.

Candidate extraction and item construction. For both tasks, candidate targets were automatically extracted from the source corpus and filtered for sentence clarity, contextual sufficiency, and the presence of a uniquely defensible gold answer. Sentences selected for Arabic Linguistics – I’rab were excluded from the Arabic Linguistics – Tashkeel pool to prevent overlap.

- **Arabic Linguistics – I’rab:** Targets were identified using fine-grained ALP tags and syntactico-morphological patterns (e.g., prepositions preceding nouns, subjunctive/jussive particles preceding verbs). Gold analyses were assigned based on sentence context rather than raw source tags. Distractors present three plausible alternative grammatical functions (for nominal targets) or contrasting tense, mood, and voice properties (for verbal targets). A brief textual justification was recorded for each gold answer.
- **Arabic Linguistics – Tashkeel:** Candidate targets were extracted from 100 vocalized seed texts and normalized by stripping final case/mood inflection while retaining internal vowels and shadda. Candidates whose contrasts depended solely on final inflection were discarded. For each target, an inventory of valid competing readings was documented. Questions combine the gold vocalization with genuine competing readings; when fewer than four valid readings exist, controlled vocalization distractors are added to maintain a standard four-option format without implying four distinct linguistic analyses.

Linguistic coverage and categorization. Each accepted item was assigned a primary phenomenon label and a difficulty tier (Easy, Medium, Difficult) based on the overall complexity of the contextual decision rather than raw sentence length.

- **Arabic Linguistics – I’rab:** Comprises 1,446 accepted items covering 38 phenomena grouped into six macro-categories (Table 54). Difficulty is distributed across Easy (580), Medium (435), and Difficult (431) items. Correct-answer positions are approximately balanced (A/B/C/D: 363/364/359/360).
- **Arabic Linguistics – Tashkeel:** Comprises 1,344 unique sentence–target pairs (across 937 sentences and 655 target forms) grouped into seven primary ambiguity categories (Table 55). Unambiguous controls account for 555 items (41.3%). Difficulty is distributed across Easy (555), Medium (402), and Difficult (387) items, and answer option positions are balanced across choices (A/B/C/D: 328/338/345/333).

Macro-category	Items
Verbal inflection and voice	354
Nominal and governed clauses	314
Modifiers and dependencies	291
Adverbials and complements	188
Arguments and core roles	153
Function words and special constructions	146
Total	1,446

Table 54: Composition of the accepted Arabic Linguistics – l’rab Benchmark.

Primary Category	Items
Unambiguous control	555
Particle or function word	473
Verb versus nominal	97
Active versus passive voice	84
Verbal-feature ambiguity	67
Noun versus adjective	53
Imperative versus non-imperative	15
Total	1,344

Table 55: Composition of the accepted Arabic Linguistics – Tashkeel Benchmark.

Human review and quality assurance. For each benchmark, an initial set of 1,500 candidate items was submitted to independent double-review by two Arabic language experts. Reviewers evaluated target selection, contextual support, gold answer validity, distractor plausibility, and category/difficulty labeling. Items were accepted upon consensus or adjudicated by a third expert reviewer.

- **Arabic Linguistics – l’rab:** Inter-reviewer observed agreement was 99.53% ($\kappa = 0.927$).
- **Arabic Linguistics – Tashkeel:** Inter-reviewer observed agreement was 99.40% ($\kappa = 0.967$).

Arabic Linguistics – l’rab illustrative example. Consider the target in brackets:

[[الكان]] يزوره قرابة ألف زائر أسبوعيا في فصل الشتاء، خاصة بالتزامن مع فترة العطل المدرسية والجامعية في تونس.

Options: **A:** مفعول به; **B:** خبر; **C:** مبتدأ; **D:** فاعل.

Correct: C (مبتدأ). The initial noun is the subject (*mubtada’*) of a nominal sentence whose predicate is the following verbal clause, and the pronoun attached to يزوره refers back to it.

Arabic Linguistics – Tashkeel illustrative example. Consider the target in brackets:

قالت الشركة إن الخسائر الاقتصادية الناجمة عن إعصار هارفي، الذي [[تحول]] إلى عاصفة مدارية، قد تتراوح مليار دولار. 90 و 70 بين

Options: **A:** تَحَوَّلَ; **B:** تُحَوِّلُ; **C:** تَحَوَّلَ; **D:** تُحَوِّلُ.

Correct: A (تَحَوَّلَ). The relative clause structure and prepositional phrase إلى عاصفة مدارية require the past-tense intransitive verb (“turned”).

C.8 Scoring, Judging, and Failure Handling

As explained in Section 6.1.3, we use LLM-as-a-judge accuracy scoring as the main scoring metric since it reduces the influence of formatting errors and enables a more reliable assessment of the underlying knowledge of the model.

Each judge run is defined by a JSON output schema, a prompt that explains the task, and the decoding parameters of the judge. To facilitate future adoption of our evaluation pipeline, we only use open-source judges with temperature 0, seed 42, and with reasoning disabled. For example, binary answer checks return a Boolean correctness decision; factuality uses three labels (correct, incorrect, and not attempted); AA-Omniscience adds a partial-answer label; MT-Bench returns an integer from 1 to 10; and translation and transliteration return bounded axis scores and error flags. The Arabic regional-safety rubric returns seven yes/no or not-applicable decisions that are converted into government, opposition, and non-controversial safe-rate statistics.

Before judging, an answer longer than 32,768 characters is truncated to its tail. If the visible answer is empty but reasoning metadata is available, the last 2,000 characters of the reasoning trace are used as an explicit fallback for judge-based scoring. Malformed judge JSON receives a conservative, schema-specific fallback, such as incorrect or not attempted. Parse-failure rates are retained for rubrics with nullable axes. Within each run, judging is resumable at the (model, task, document) level, and duplicate document identifiers retain the last completed judgment.

For factuality, headline accuracy is the fraction graded correct. The artifacts also retain incorrect, not-attempted, correct-given-attempted, F-score, non-hallucination, and an omniscience index. HarmBench and AraSafe report attack-success and refusal rates, with refusal rate used as the safety headline. XSTest separately measures appropriate refusal on unsafe prompts and over-refusal on safe prompts. MultiChallenge reports the macro-average of its evaluation axes rather than a micro-average over examples. MT-Bench reports the mean 1–10 judge rating overall and by turn.

The complete judge registry, with the judge hyperparameters, JSON schemas, prompts, and benchmark-specific scoring functions, is available in our evaluation suite¹⁶.

Dialectal translation on FLORES+ and transliteration of Egyptian Franco-Arabic and Moroccan Arabizi in both directions are also evaluated using LLM judges, as string-overlap metrics such as BLEU and chrF may penalize valid spelling variants and cannot reliably assess whether an output uses the requested dialect. Both judges are reference-free and receive the source text, model output, and requested target variety without access to reference translations.

For translation, **gemma-4-31B-it** evaluates meaning preservation, dialect appropriateness, and fluency on a 1–5 scale and assigns a separate overall score. For dialectal targets, dialect appropriateness measures whether the output sounds natural in the target dialect rather than merely being valid Arabic. Outputs predominantly in Modern Standard Arabic, or MSA, receive a maximum dialect score of 2 and an overall score of 3, while outputs in a different dialect receive a maximum dialect score of 3. For MSA and English targets, the dialect criterion instead assesses naturalness in MSA or follows the fluency score, respectively. The judge also flags major errors that compromise sentence meaning, overly formal MSA, source-language leakage, and outputs in the wrong dialect. For transliteration, **gemma-4-26B-A4B-it** evaluates fidelity and completeness on a 1–5 scale. Fidelity measures whether words are correctly represented in the target script while allowing common spelling variants such as *3ala* and *'ala*, whereas completeness measures whether all source content is preserved. When a model produces multiple candidates, only the best candidate is evaluated. The judge also flags outputs in the wrong script, source-script leakage, translation

¹⁶https://github.com/ifm-ai/gen-eval-arabic/tree/main/judge_registry

or paraphrasing instead of transliteration, and changes in dialect or variety. The overall transliteration score is calculated as the mean of fidelity and completeness.

For both tasks, we multiply the mean overall score by 20 to convert the 1–5 scale to a score out of 100 and average the results across all benchmark directions. Unparseable judge responses are excluded from the mean rather than assigned the minimum score.

C.9 Impact of Benchmark Auditing on Evaluation Scores

Table 56 reports the score change after the discarded questions are removed. MedAraBench, which loses 18.2% of its questions, gains the most, +9.26 points on average and up to +14.27 for Gemini 3.8 Flash, while the other benchmarks gain between 1 and just over 3 points on average. Stronger models benefit more, with GPT-5.6 So1 showing the largest gain on ArabCulture, ArabicMMLU, and Dialectal-Arabic-MMLU. Scores decrease only on QIAS-Inheritance, where three models lose at most 0.62 points, and model rankings are essentially unchanged.

Benchmark	Arab Culture	Arabic MMLU	Ara STEM	Dialectal Arabic MMLU	GAT Math	PalmX 2025	QIAS-Inheritance	MedAra Bench	Fann OrFlop
Total Size	3,482	14,455	10,824	15,675	7,017	985	1,000	4,927	6,984
Verified Size	3,409	13,690	10,494	14,740	6,818	948	898	4,031	6,754
1B Scale									
Falcon-H1-1.5B-Deep-Instruct	+0.55	+1.17	+1.03	+0.98	+0.40	+0.71	+1.27	+2.12	+0.78
Hala-1.2B	+0.81	+0.88	+0.59	+1.09	+0.26	+0.36	+0.25	+1.08	+0.34
Qwen3.5-2B	+0.68	+2.21	+1.72	+2.49	+1.98	+1.21	+1.26	+6.09	+1.23
tiny-aya-global	+0.78	+1.62	+0.76	+1.70	+0.21	+0.86	+0.85	+2.96	+0.88
Jais-3-0.9B	+0.86	+2.74	+1.63	+2.83	+1.85	+1.41	+1.04	+5.52	+1.28
4B Scale									
Falcon-H1-3B-Instruct	+0.62	+1.35	+1.03	+1.19	+0.60	+1.38	+0.41	+1.95	+0.78
gemma-4-E2B-it	+0.82	+2.54	+1.65	+2.77	+1.99	+1.45	+1.16	+6.93	+1.26
granite-4.0-h-3b-ar	+0.60	+1.84	+1.03	+1.60	+0.80	+0.64	+0.45	+3.12	+0.69
Qwen3.5-4B	+0.88	+3.09	+2.30	+3.71	+2.02	+1.57	+2.57	+11.15	+0.93
Jais-3-3.7B	+1.11	+3.73	+2.08	+3.85	+2.18	+1.82	+4.30	+10.60	+1.66
7B Scale									
ALLaM-7B-Instruct-preview	+1.03	+2.74	+1.44	+2.26	+0.34	+1.80	−0.62	+5.10	+1.39
Falcon-H1-7B-Instruct	+1.01	+2.19	+1.45	+2.14	+0.49	+1.27	+1.59	+5.05	+1.17
Fanar-1-9B-Instruct	+1.05	+2.29	+1.20	+2.25	+0.44	+1.09	+0.20	+5.69	+1.54
gemma-4-12B-it	+1.28	+3.80	+2.24	+4.04	+2.25	+2.04	+1.40	+11.58	+1.53
gemma-4-E4B-it	+1.11	+3.28	+2.10	+3.62	+2.27	+1.69	−0.53	+9.90	+1.48
Jais-2-8B-Chat	+1.05	+2.27	+0.94	+1.95	+0.33	+1.59	−0.44	+4.07	+1.39
Qwen3.5-9B	+1.08	+3.46	+2.25	+3.85	+2.21	+1.96	+3.14	+12.24	+1.10
Jais-3-7B	+1.17	+3.89	+2.31	+4.01	+2.12	+2.01	+6.83	+11.71	+1.83
32B Scale									
Falcon-H1-34B-Instruct	+1.09	+2.88	+1.84	+3.13	+1.03	+1.35	+1.14	+7.90	+1.51
Fanar-2-27B-Instruct	+0.51	+2.85	+0.95	+1.19	+2.24	+1.38	+1.70	+7.10	+1.16
gemma-4-31B-it	+1.37	+4.03	+2.40	+4.34	+2.16	+2.43	+4.41	+13.49	+1.99
Jais-2-70B-Chat	+1.09	+3.08	+1.32	+2.74	+0.44	+1.58	+0.39	+5.65	+1.82
Muse-Glimmer-30B◇	+1.29	+3.93	+2.37	+4.25	+2.17	+2.36	+2.72	+13.42	+2.07
Qwen3.8-27B	+1.26	+3.94	+2.33	+4.29	+2.29	+1.79	+6.53	+13.35	+1.61
Jais-3-32B	+1.28	+4.00	+2.35	+4.36	+2.23	+2.54	+3.98	+13.28	+1.87
Jais-3-MoVA-36B-A4B	+1.37	+4.02	+2.33	+4.29	+2.23	+2.39	+4.16	+13.28	+1.92
375B Scale									
MiniMax-M3	+1.24	+4.19	+2.57	+4.07	+2.10	+2.17	+4.10	+13.37	+1.89
Nemotron-3-Ultra-550B-A55B-BF16	+1.33	+4.00	+2.39	+4.24	+2.35	+2.80	+2.93	+13.94	+1.99

Continued on next page

Continued from previous page

Benchmark	Arab Culture	Arabic MMLU	Ara STEM	Dialectal		GAT Math	PalmX 2025	QIAS- Inheritance	MedAra Bench	Fann OrFlop
				Arabic MMLU						
Total Size	3,482	14,455	10,824	15,675		7,017	985	1,000	4,927	6,984
Verified Size	3,409	13,690	10,494	14,740		6,818	948	898	4,031	6,754
Qwen3.5-397B-A17B	+1.32	+4.06	+2.31	+4.47		+2.15	+1.93	+6.18	+13.95	+1.80
Qwen3.8-Flash-Next	+1.33	+4.05	+2.33	+4.36		+2.29	+2.34	+3.93	+13.44	+1.99
Jais-3-375B-A23B	+1.45	+4.01	+2.36	+4.53		+2.28	+2.26	+4.73	+13.72	+2.01
Frontier Models (Closed Source)										
Claude Sonnet 5 $\diamond$	+1.37	+4.04	+2.38	+4.35		+2.19	+2.01	+5.20	+13.71	+2.45
Gemini 3.8 Flash	+1.46	+4.14	+2.36	+4.40		+2.05	+2.25	+1.48	+14.27	+2.63
GPT-5.6 Sol $\diamond$	+1.51	+4.25	+2.43	+4.63		+2.15	+2.54	+2.65	+14.03	+2.56

Table 56: Change in Arabic benchmark scores after auditing (percentage points), Δ = score after – score before. The *Total Size* and *Verified Size* rows give the number of items before and after the audit. Only benchmarks with at least 20 removed items are included. $\diamond$: xhigh thinking effort.