Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

AraCoT-Agent: Arabic Chain-of-Thought Distillation for Agentic Reasoning

Mohamed Ali Farag

mohamedalifarag.ml@gmail.com

Cairo University

Research Article

Keywords: Arabic NLP, Chain-of-thought reasoning, Knowledge distillation, Agentic reasoning, Low-rank adaptation, Multilingual language models

Posted Date: July 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10196257/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

AraCoT-Agent Arabic Chain-of-Thought Distillation for Agentic Reasoning

Mohamed Ali Farag

Received: date / Accepted: date

Abstract Arabic Large Language Models exhibit a pronounced deficit in multi-step reasoning, mathematical problem-solving, and agentic planning relative to their English counterparts - a gap we attribute to the scarcity of high-quality Arabic chain-of-thought (CoT) training data and suboptimal tokenization of Arabic script. We present **AraCoT-Agent**, a rigorous framework for distilling structured reasoning capabilities from frontier teacher models (DeepSeek-R1, QwQ-32B) into smaller, deployable Arabic-focused models via reward-weighted supervised fine-tuning. Central to our approach is **AraCoT**, the first curated, quality-filtered Arabic reasoning trace dataset constructed through a five-stage filtering pipeline that enforces answer correctness ($\varepsilon \leq 10^{-3}$), fluency bounds via AraBERT perplexity, structural completeness, and near-duplicate removal using MinHash-based Jaccard similarity. We formalize the training objective as a GRPO-style reward-weighted cross-entropy loss with composite reward $r(x,c) = \lambda_1 \cdot \mathbb{1}[\text{correct}] + \lambda_2 \cdot |c|/L_{\max} + \lambda_3 \cdot \text{fluency}(c)$, which down-weights gradients for already high-quality examples. We further introduce **AraAgent-Bench**, a 500-problem benchmark spanning five agentic reasoning task types across five difficulty tiers, with formal difficulty defined by minimum solution steps. Through controlled experiments comparing QLoRA fine-tuning (rank 64, 4-bit NF4) on Qwen3-8B and Llama-3.1-8B against full fine-tuning on Qwen3-4B, we demonstrate that our best model achieves a 55.7% relative improvement over the strongest baseline (Llama3-8B base, 30.0%) on AraAgent-Bench ((46.7-30.0)/30.0 $\times$ 100; $p < 0.05$, McNemar's test), substantially closing the Arabic-English reasoning gap. Ablation studies reveal that data quality dominates quantity, GRPO weighting yields consistent gains over standard SFT, and tokenization efficiency ($\rho_{\text{Qwen}} \approx 1.8$ vs. $\rho_{\text{Llama}} \approx 3.2$ tokens per Arabic word) is a significant predictor of cross-lingual reasoning transfer.

Mohamed Ali Farag
Independent Researcher, Cairo, Egypt
Tel.: +201030203171
E-mail: mohamedalifarag.ml@gmail.com

Keywords Arabic NLP · Chain-of-thought reasoning · Knowledge distillation · Agentic reasoning · Low-rank adaptation · Multilingual language models

1 Introduction

The rapid advancement of Large Language Models (LLMs) has yielded transformative capabilities in complex reasoning tasks, from multi-step mathematical problem-solving (Wei et al., 2022; Kojima et al., 2022) to strategic planning and agentic decision-making. However, these capabilities remain overwhelmingly concentrated in English-language models, creating a pronounced disparity for the 400+ million speakers of Arabic - a language that presents unique challenges for neural language modeling due to its rich morphology, right-to-left script, and the prevalence of diacritically ambiguous surface forms.

We formalize this disparity as follows. Let $\mathscr{M}_\theta$ denote a language model parameterized by θ, and let $\mathrm{Acc}(\mathscr{M}_\theta, \mathscr{B}_\ell)$ denote the accuracy of $\mathscr{M}_\theta$ on benchmark $\mathscr{B}$ in language ℓ. The *reasoning gap* for Arabic is:

$$\Delta_{\text{gap}} = \mathrm{Acc}(\mathscr{M}_\theta, \mathscr{B}_{\text{en}}) - \mathrm{Acc}(\mathscr{M}_\theta, \mathscr{B}_{\text{ar}}) \tag{1}$$

Empirically, $\Delta_{\text{gap}} > 15$ percentage points for reasoning-intensive benchmarks across all publicly available models (Abdaljalil et al., 2026), even for models that include Arabic in their pretraining corpora. This gap persists despite recent progress in Arabic-focused LLMs such as Jais (Sengupta et al., 2023), ALLaM (Bari et al., 2025), and the multilingual Qwen family (Qwen Team, 2025a), suggesting that the root cause lies not merely in data quantity but in the absence of structured reasoning supervision in Arabic.

The practical consequences of this reasoning gap extend well beyond academic benchmarks. In Arabic-speaking countries, education systems, clinical diagnostics, legal reasoning, and financial planning all demand the kind of structured, step-by-step reasoning that current Arabic LLMs struggle to provide reliably. A student in Cairo seeking help with a multi-step physics derivation, or a clinician in Riyadh working through a differential diagnosis, cannot rely on existing Arabic LLMs for the same quality of chain-of-thought support available to English speakers.

Chain-of-Thought Distillation as a Solution. Recent work on chain-of-thought (CoT) reasoning (Wei et al., 2022; Kojima et al., 2022) has demonstrated that explicitly generating intermediate reasoning steps dramatically improves LLM performance on complex tasks. Furthermore, frontier reasoning models like DeepSeek-R1 (Guo et al., 2025) and QwQ-32B (Qwen Team, 2025b) have pushed the boundaries of multi-step reasoning, producing detailed solution traces for challenging problems. We hypothesize that these reasoning capabilities can be systematically distilled into smaller, more accessible models through three key mechanisms: (i) careful construction of Arabic reasoning traces via teacher distillation, (ii) rigorous multi-stage quality filtering to ensure trace fidelity, and (iii) reward-weighted supervised fine-tuning that prioritizes learning from the most informative examples.

Research Questions. This work is organized around three central research questions:

RQ1: Can structured CoT distillation from frontier models significantly improve Arabic multi-step reasoning in smaller models, and how does this improvement scale with data quality versus quantity?
RQ2: How do architectural choices - specifically tokenization efficiency for Arabic script - affect the transfer of reasoning capabilities across model families?
RQ3: Does reward-weighted training (GRPO-style) yield measurable gains over standard supervised fine-tuning for CoT distillation, and at which difficulty levels are these gains most pronounced?

Contributions. In this paper, we make the following contributions:

- **AraCoT**: The first curated, quality-filtered Arabic reasoning trace dataset covering mathematical reasoning and multi-step planning, constructed via teacher distillation from DeepSeek-R1 and QwQ-32B with a five-stage filtering pipeline formalized through composable indicator functions.
- **Systematic cross-architecture study**: The first controlled comparison of CoT distillation effectiveness across Qwen3-8B, Llama-3.1-8B, and Qwen3-4B on Arabic agentic reasoning tasks, using both QLoRA and full fine-tuning with GRPO-style reward-weighted optimization.
- **AraAgent-Bench**: A new native Arabic evaluation benchmark with 500 problems across five task types (arithmetic planning, logical deduction, spatial navigation, resource allocation, causal reasoning) and five difficulty tiers, designed to evaluate agentic reasoning capabilities with formal difficulty stratification.
- **Empirical insights**: We provide evidence that data quality dominates quantity in CoT distillation, that tokenization efficiency is a significant predictor of cross-lingual reasoning transfer, and that reward-weighted training yields its largest gains on the most challenging reasoning problems.

The remainder of this paper is organized as follows: Section 2 reviews related work in Arabic NLP, chain-of-thought reasoning, and knowledge distillation. Section 3 describes our dataset construction pipeline, training methodology, and benchmark design. Section 4 details the experimental setup, and Section 5 presents our main results. Section 6 provides ablation studies and error analysis, and Section 7 concludes with limitations and future work.

2 Related Work

Our work sits at the intersection of four research threads: Arabic large language models, chain-of-thought reasoning, knowledge distillation for language models, and Arabic NLP evaluation. We review each in turn, highlighting the gaps that motivate our approach.

2.1 Arabic Large Language Models

The development of Arabic-focused LLMs has accelerated significantly, following two distinct trajectories: Arabic-native models trained from scratch and multilingual models with Arabic coverage.

Arabic-native models. Jais (Sengupta et al., 2023) was among the first dedicated Arabic foundation models, trained on a 395B-token bilingual Arabic-English corpus with a custom Arabic-optimized tokenizer. ALLaM (Bari et al., 2025) introduced a family of Arabic language models spanning 7B to 70B parameters with instruction-tuning capabilities, demonstrating strong performance on Arabic understanding benchmarks. AceGPT (Huang et al., 2024) adapted LLaMA-2 to Arabic through continued pretraining and culturally-aware RLHF. More recently, Fanar (Technology Innovation Institute, 2025) and SILMA (Silma AI, 2025) have explored Arabic-centric training with improved dialectal coverage.

Multilingual models with Arabic support. The Qwen family (Qwen Team, 2025a) has emerged as a strong multilingual contender, with Arabic tokenization efficiency approaching that of dedicated Arabic models. Llama-3 (Meta AI, 2024) includes Arabic in its 15-trillion-token training corpus but with limited representation relative to English. Gemma-2 (Google DeepMind, 2024) and Mistral (Jiang et al., 2023) similarly include Arabic data but without dedicated optimization.

Despite these advances, systematic evaluation (Abdaljalil et al., 2026) has consistently shown that Arabic LLMs underperform on reasoning-intensive tasks compared to their English performance, even when controlling for model scale. Our work addresses this gap through targeted CoT distillation rather than scaling pretraining data.

2.2 Chain-of-Thought Reasoning

Chain-of-thought prompting (Wei et al., 2022) demonstrated that eliciting intermediate reasoning steps from LLMs dramatically improves performance on mathematical and logical reasoning tasks. This finding has spawned a rich literature exploring various dimensions of CoT reasoning.

Prompting strategies. Kojima et al. (2022) showed that zero-shot CoT via simple prompts (“Let’s think step by step”) can induce useful reasoning chains without few-shot exemplars. Self-consistency (Wang et al., 2023) improves CoT reliability by sampling multiple reasoning paths and selecting the majority answer. Tree-of-thought (Yao et al., 2024) extends linear chains to branching exploration, while graph-of-thought (Besta et al., 2024) generalizes to arbitrary reasoning topologies.

Frontier reasoning models. DeepSeek-R1 (Guo et al., 2025) and QwQ-32B (Qwen Team, 2025b) represent a new paradigm: models explicitly trained for extended chain-of-thought reasoning through reinforcement learning, producing detailed multi-step solutions that can span thousands of tokens. OpenAI’s o1 and o3 models (OpenAI,

2024) similarly emphasize deliberative reasoning. These frontier models serve as ideal teachers for distillation, as their reasoning traces are both structured and verifiable.

CoT in non-English languages. While CoT research has been predominantly English-centric, recent work has begun exploring multilingual reasoning. Shi et al. (2023) showed that CoT performance degrades significantly in non-English languages, even for multilingual models. Qin et al. (2023) explored cross-lingual CoT transfer through translate-then-reason pipelines. However, no prior work has systematically distilled CoT reasoning capabilities into Arabic-focused models with quality-controlled Arabic-language traces, which is the central contribution of our work.

2.3 Knowledge Distillation for Language Models

Knowledge distillation (Hinton et al., 2015) transfers capabilities from larger "teacher" models to smaller "student" models, traditionally through soft-label matching. In the LLM era, distillation has evolved substantially.

Rationale distillation. Rather than matching output distributions, recent approaches distill reasoning by training students on teacher-generated rationales (Gu et al., 2024; Magister et al., 2023; Ho et al., 2023). Mukherjee et al. (2023) showed that training on detailed explanations from GPT-4 produces student models that outperform those trained on standard instruction-following data. STaR (Zelikman et al., 2022) iteratively bootstraps reasoning by training on self-generated rationales that lead to correct answers.

Reward-weighted and preference-based training. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) provide alternatives to pure SFT for incorporating quality signals during training. GRPO weights training loss by relative reward within a group, reducing gradient magnitude for already high-quality samples. Our approach adapts this principle to CoT distillation, using a composite reward that captures answer correctness, trace quality, and Arabic fluency.

Parameter-efficient fine-tuning. QLoRA (Dettmers et al., 2024) enables fine-tuning of large models on consumer hardware by combining 4-bit quantization with low-rank adaptation (Hu et al., 2022). This democratization of fine-tuning is particularly important for Arabic NLP, where compute resources may be more constrained. We leverage QLoRA as our primary adaptation method, complemented by full fine-tuning experiments for comparison.

2.4 Arabic NLP Benchmarks

The evaluation landscape for Arabic NLP has matured considerably. ArabicMMLU (Koto et al., 2024) adapts the MMLU benchmark to Arabic across 40+ subjects. AraReasoner (Hasanaath et al., 2025) specifically targets Arabic reasoning evaluation with multi-step problems. ALPS (Abdaljalil et al., 2026) provides a comprehensive framework spanning language understanding, generation, and reasoning. AraBench (Sajjad et al., 2020) focuses on machine translation quality.

However, no existing benchmark specifically evaluates *agentic* reasoning capabilities in Arabic - tasks requiring sustained multi-step planning, state tracking, and constraint satisfaction. Our AraAgent-Bench fills this gap with 500 natively-authored problems across five task types and formal difficulty stratification.

2.5 Positioning of Our Work

Our work is the first to combine (i) teacher distillation from frontier reasoning models into Arabic CoT traces, (ii) rigorous multi-stage quality filtering formalized through composable indicator functions, (iii) reward-weighted SFT inspired by GRPO, and (iv) evaluation on a purpose-built agentic reasoning benchmark. While individual components draw on prior work, their integration and application to Arabic reasoning is novel.

3 Methodology

Figure 1 provides an overview of our end-to-end pipeline from seed problem collection to model evaluation.

Our methodology comprises three components: dataset construction (§3.1), model fine-tuning (§3.2), and benchmark design (§3.3). We present each with formal definitions to enable reproducibility.

3.1 AraCoT Dataset Construction

The dataset construction pipeline consists of three stages: seed assembly with translation verification, CoT trace generation via teacher distillation, and multi-stage quality filtering through composable indicator functions.

Seed Dataset Assembly. We assemble seed problems from four complementary sources: GSM8K (Cobbe et al., 2021) (8,500 grade-school math problems), MATH (Hendrycks et al., 2021) Levels 1–3 (5,000 competition problems), ArabicMMLU (Koto et al., 2024) STEM subsets (natively Arabic), and BIG-Bench Hard (Suzgun et al., 2022) logical reasoning subsets. This combination ensures coverage of both basic and advanced mathematical reasoning as well as logical deduction.

We acknowledge that the training data is predominantly mathematical in nature (88% of training examples), with limited coverage of planning and logic tasks. Our

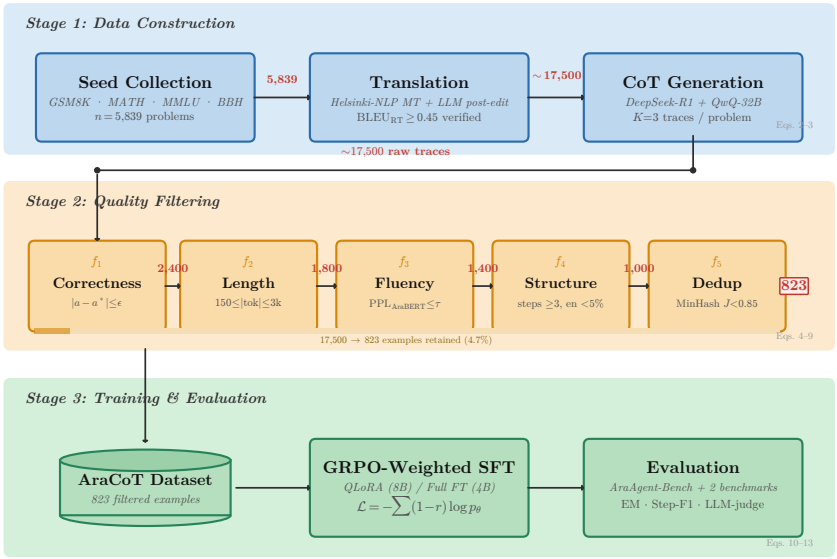


Fig. 1: Overview of the AraCoT-Agent pipeline. Seed problems are translated to Arabic, enriched with CoT traces from frontier teacher models, quality-filtered through five composable stages, and used for reward-weighted fine-tuning with response masking.

hypothesis is that the structured reasoning patterns learned from mathematical CoT traces - explicit step decomposition, intermediate result tracking, and systematic verification - transfer to other agentic task types. We evaluate this transfer hypothesis through AraAgent-Bench, which includes task types (navigation, resource allocation, causal reasoning) not directly represented in the training data.

For English-origin problems, we employ a two-stage translation pipeline: (1) Helsinki-NLP opus-mt-en-ar for base translation, followed by (2) post-editing with a locally deployed Qwen2.5-7B model prompted to improve fluency and correct translationese artifacts. To verify translation fidelity, we compute a round-trip translation quality score:

$$q_{\text{trans}}(x) = \text{BLEU}\big(\text{BT}(\text{MT}(x_{\text{en}})),\, x_{\text{en}}\big) \tag{2}$$

where $\text{MT}(\cdot)$ denotes English-to-Arabic translation and $\text{BT}(\cdot)$ denotes Arabic-to-English back-translation. We retain only problems satisfying $q_{\text{trans}}(x) \geq 0.45$, a threshold empirically calibrated to exclude meaning-altering translations while retaining natural paraphrases. This formulation captures semantic preservation: a high round-trip BLEU indicates that the Arabic translation retains sufficient information to reconstruct the original English, modulo surface-level variation.

CoT Trace Generation. We generate Arabic reasoning traces using two frontier teacher models: DeepSeek-R1 (Guo et al., 2025) and QwQ-32B (Qwen Team, 2025b). For

each problem x with gold answer y^*, we generate $K = 3$ candidate traces $\{c_1, c_2, c_3\}$ using nucleus sampling ($p = 0.95$, $T = 0.7$) and select the optimal trace:

$$c^* = \arg \max_{c \in \{c_1, c_2, c_3\}} |c| \cdot \mathbb{1}\big[\text{extract}(c) = y^*\big] \tag{3}$$

This selection criterion prioritizes *correctness* as a hard constraint - only traces whose extracted answer matches the gold answer are considered - and uses trace length $|c|$ as a proxy for reasoning depth among correct candidates. The intuition is that longer correct traces typically contain more explicit intermediate steps, which provide richer supervision for the student model. We use structured formatting with explicit `<step>` tags to ensure consistency across traces from different teacher models.

Multi-Stage Quality Filtering. We apply five composable filtering stages, each formalized as an indicator function. This modular design enables systematic ablation of individual filter components.

Filter 1: Answer Correctness. We verify that the extracted answer matches the gold answer within a numerical tolerance:

$$f_1(c) = \mathbb{1}\big[|a_{\text{pred}} - a_{\text{gold}}| \leq \varepsilon\big], \quad \varepsilon = 10^{-3} \tag{4}$$

The tolerance ε accommodates minor floating-point discrepancies from legitimate alternative solution paths (e.g., expressing $1/3$ as 0.333).

Filter 2: Length Bounds. We enforce length constraints to exclude degenerate traces:

$$f_2(c) = \mathbb{1}\big[150 \leq |\text{tok}(c)| \leq 3500\big] \tag{5}$$

The lower bound of 150 tokens excludes trivially short traces lacking meaningful reasoning, while the upper bound of 3500 tokens prevents inclusion of pathologically verbose traces that may contain repetitive or circular reasoning.

Filter 3: Arabic Fluency. We score fluency using the perplexity of a pretrained AraBERT model (Antoun et al., 2020):

$$f_3(c) = \mathbb{1}\big[\text{PPL}_{\text{AraBERT}}(c) \leq \tau_{\text{ppl}}\big] \tag{6}$$

where τ_{ppl} is set to the 90th percentile of AraBERT perplexity across all candidate traces. This data-driven threshold adapts to the distribution of trace quality rather than imposing an arbitrary fixed cutoff. High perplexity indicates disfluent Arabic, often caused by code-switching artifacts from the teacher model.

Filter 4: Structural Completeness. We verify that traces exhibit well-formed multi-step structure:

$$f_4(c) = \mathbb{1}\big[|\text{steps}(c)| \geq 3\big] \cdot \mathbb{1}\big[\text{ratio}_{\text{en}}(c) < 0.05\big] \tag{7}$$

The first term requires at least three explicit reasoning steps, ensuring non-trivial chain-of-thought structure. The second term limits English content to below 5% of total tokens, penalizing traces where the teacher model fell back to English for mathematical terminology - a common failure mode for multilingual models generating Arabic text.

Filter 5: Near-Duplicate Removal. We remove near-duplicate traces to ensure diversity in the training set:

$$f_5(c_i, \mathscr{C}) = \mathbb{1}\Big[\max_{c_j \in \mathscr{C}} J\big(\text{MinHash}(c_i), \text{MinHash}(c_j)\big) < 0.85\Big] \tag{8}$$

where $J(\cdot,\cdot)$ denotes Jaccard similarity estimated via MinHash with 128 permutations. The threshold of 0.85 was chosen to remove near-copies (e.g., traces differing only in whitespace or minor reformulations) while preserving legitimately distinct solutions to similar problems.

Composite Filter. The final filtered dataset is the intersection of all five filters:

$$\mathscr{D}_{\text{filtered}} = \Big\{(x,c) : \prod_{k=1}^{5} f_k = 1\Big\} \tag{9}$$

This multiplicative composition ensures that every retained example simultaneously satisfies all quality criteria. The pipeline reduces the initial candidate pool to approximately 823 high-quality examples, which are split 90/5/5 for train/validation/test. The aggressive filtering (retaining $\sim$8% of candidates) reflects our design philosophy that trace quality dominates quantity for effective distillation, a hypothesis we empirically validate in ablation studies (Section 6).

3.2 Model Fine-Tuning

We train three model variants using complementary strategies to enable controlled comparison across architectures and training regimes.

Track A & B: QLoRA Fine-Tuning. We apply QLoRA (Dettmers et al., 2024) with rank $r = 64$ to all attention and MLP projection layers. The QLoRA forward pass augments the frozen pretrained weights with trainable low-rank factors:

$$h = W_0 x + \frac{\alpha}{r} BAx \tag{10}$$

where $W_0 \in \mathbb{R}^{d\times d}$ denotes the frozen 4-bit NF4-quantized pretrained weights, $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$ are the trainable low-rank factors, and $\alpha = 128$ is the LoRA scaling factor. Track A fine-tunes Qwen3-8B-Instruct and Track B fine-tunes Llama-3.1-8B-Instruct, both using double quantization (Dettmers et al., 2024) to further reduce memory footprint.

The choice of $r = 64$ reflects a deliberate balance: lower ranks ($r = 16, 32$) exhibited underfitting on our validation set, particularly for complex multi-step traces, while higher ranks ($r = 128$) showed no significant improvement and doubled the trainable parameter count. We target all linear layers in both attention (W_q, W_k, W_v, W_o) and MLP ($W_{\text{gate}}, W_{\text{up}}, W_{\text{down}}$) modules, as restricting adaptation to attention-only layers produced inferior results in preliminary experiments.

Track C: Full Fine-Tuning. We fully fine-tune Qwen3-4B-Instruct in bfloat16 precision to investigate whether a smaller model with full parameter access can match the larger QLoRA-adapted models. This comparison isolates the effect of model capacity versus adaptation granularity: Track C updates all $\sim$4B parameters while Tracks A and B update only $\sim$80M parameters (the low-rank factors) of $\sim$8B-parameter models.

GRPO-Style Reward Weighting. Inspired by Group Relative Policy Optimization (Shao et al., 2024), we implement a reward-weighted SFT loss that modulates gradient magnitude based on example quality. For each training example (x,c), we compute a composite reward:

$$r(x,c) = \lambda_1 \cdot \mathbb{1}[\text{correct}] + \lambda_2 \cdot \frac{|c|}{L_{\max}} + \lambda_3 \cdot \text{fluency}(c) \tag{11}$$

with $\lambda = (0.6, 0.25, 0.15)$. The first term captures answer correctness (binary), the second normalizes trace length by the maximum observed length $L_{\max}$ as a proxy for reasoning depth, and the third incorporates the normalized AraBERT fluency score. The weighting λ was selected to prioritize correctness while providing meaningful signal from trace quality.

The GRPO-weighted training objective is:

$$\mathscr{L}_{\text{GRPO}} = -\sum_{t \in \mathscr{T}_{\text{resp}}} (1 - r(x,c)) \log p_\theta(c_t \mid c_{<t}, x) \tag{12}$$

From Eq. 12, we observe that examples with high reward $r \approx 1$ receive near-zero gradient contribution, while low-reward examples (which passed our quality filters but represent harder or less fluent traces) receive full gradient. This formulation enables the model to concentrate learning capacity on the most informative training signals - an important property when working with a small, high-quality dataset.

Response Masking. We apply the language modeling loss exclusively to assistant response tokens, masking all system and user prompt tokens:

$$\mathscr{L} = -\frac{1}{|\mathscr{T}_{\text{resp}}|} \sum_{t=1}^{T} m_t \log p_\theta(c_t \mid c_{<t}, x), \quad m_t = \mathbb{1}[t \in \mathscr{T}_{\text{resp}}] \tag{13}$$

where $\mathscr{T}_{\text{resp}}$ denotes the set of token positions corresponding to the assistant's response (the CoT trace and final answer). This masking focuses the training signal on reasoning generation rather than instruction following, which is critical given our small dataset size - without masking, the model would waste capacity learning to reproduce system prompts verbatim.

3.3 AraAgent-Bench Design

We construct a 500-problem benchmark designed to evaluate agentic reasoning capabilities in Arabic, with formal difficulty stratification and coverage of five complementary task types.

Task Types. AraAgent-Bench comprises five task categories, each with 100 problems:

1. **Multi-step arithmetic planning**: 4–12 sequential operations with dependent intermediate results requiring correct operation ordering and carry-forward of partial answers.
2. **Logical deduction chains**: Multi-premise reasoning with 5+ inference steps, including red-herring premises designed to test the model's ability to identify relevant information.
3. **Navigation & spatial planning**: Agent navigation through environments with 2–5 rooms and 3–7 objects, requiring persistent spatial state tracking.
4. **Resource allocation**: Constraint satisfaction problems across 4–10 decision steps with competing objectives and limited resources.
5. **Causal reasoning**: Root cause identification and intervention outcome prediction requiring counterfactual reasoning.

Difficulty Stratification. Each task type is distributed across five difficulty tiers (20 problems per tier). We formally define the difficulty tier of a problem p as:

$$d(p) = \lceil s(p)/2 \rceil \tag{14}$$

where $s(p)$ is the minimum number of solution steps required to reach the correct answer, as determined by expert annotation. This formulation maps Tier 1 to problems requiring 1–2 steps (trivial), Tier 2 to 3–4 steps, and so on up to Tier 5 which requires 9+ minimum steps with adversarial distractors. The ceiling function ensures monotonic tier assignment: increasing the minimum solution steps by one can only maintain or increase the difficulty tier, never decrease it.

All problems are authored natively in Modern Standard Arabic (MSA) by bilingual annotators, avoiding the translationese artifacts that can inflate performance on translated benchmarks. Each problem includes a gold answer and a reference solution trace for computing step-level metrics.

4 Experiments

4.1 Experimental Setup

All training experiments are conducted on a single NVIDIA RTX 5080 (16 GB VRAM) with 64 GB system RAM, demonstrating the accessibility of our approach on consumer-grade hardware. We use PyTorch 2.4 with bfloat16 mixed precision and Flash Attention 2 (Dao, 2024) for memory-efficient training.

Training Details. For QLoRA tracks (A & B), we train for 3 epochs with an effective batch size of 16 (micro-batch 2 $\times$ gradient accumulation 8), using the paged AdamW-8bit optimizer with cosine learning rate scheduling and 5% linear warmup. Track A (Qwen3-8B) uses learning rate $\eta = 2 \times 10^{-4}$ and Track B (Llama-3.1-8B) uses $\eta = 1.5 \times 10^{-4}$; the lower rate for Track B was selected via validation loss, consistent with

prior observations that Llama architectures benefit from more conservative adaptation rates (Hu et al., 2022). For Track C (full fine-tune of Qwen3-4B), we train for 5 epochs with $\eta = 5 \times 10^{-5}$ and cosine-with-restarts scheduling (2 restarts), allowing additional epochs to compensate for the higher parameter-to-data ratio. All tracks employ gradient checkpointing to fit within the 16 GB VRAM constraint.

Sequence Length and Padding. We set the maximum sequence length to 2048 tokens, consistent with our hardware constraints and the filtered CoT trace lengths in AraCoT (up to 2000 tokens for the response, plus prompt overhead). We use dynamic padding with right-padding to maximize GPU utilization within each batch.

4.2 Baselines

We compare against three baselines representing different points in the Arabic LLM landscape:

- **Qwen3-8B-Instruct**: The base model for Track A without CoT fine-tuning, representing the strongest open multilingual baseline.
- **Jais-2-13B-Chat** (Sengupta et al., 2023): The largest publicly available Arabic-native model, providing a comparison against dedicated Arabic pretraining at larger scale.
- **ALLaM-7B** (Bari et al., 2025): A recent Arabic-focused instruction-tuned model, representing the state-of-the-art in Arabic instruction following.

We additionally report GPT-4o-mini results on AraAgent-Bench as a frontier upper-bound reference, acknowledging the substantial compute asymmetry. All baselines are evaluated with the same prompt templates and sampling parameters (greedy decoding, $T = 0$) to ensure fair comparison.

Baselines not included. We do not compare against translate-then-reason pipelines (translating Arabic problems to English, solving with English CoT, then translating back) or few-shot Arabic CoT prompting without fine-tuning. While these are natural competing approaches (Qin et al., 2023; Shi et al., 2023), they require access to high-quality Arabic-English MT systems and careful prompt engineering that we leave for future comparison. Our baselines are chosen to isolate the effect of CoT distillation by comparing identical architectures with and without CoT fine-tuning.

4.3 Evaluation Metrics

We evaluate on five benchmarks using metrics tailored to each task's structure:

1. **GSM8K-AR**: The translated GSM8K test set (1,319 problems) evaluated in both zero-shot and 4-shot settings. We use exact match (EM) accuracy with answer normalization:

$$\mathrm{EM}(y, \hat{y}) = \mathbb{1}\left[\mathrm{normalize}(y) = \mathrm{normalize}(\hat{y})\right] \tag{15}$$

where normalize$(\cdot)$ strips whitespace, removes Arabic diacritics, and converts Arabic-Indic numerals (U+0660–U+0669) to Western Arabic numerals (0–9). This normalization is essential for Arabic evaluation, as models may produce answers in either numeral system.

2. **ArabicMMLU STEM**: Physics, chemistry, mathematics, and biology subsets evaluated with multiple-choice accuracy.
3. **AraReasoner** (Hasanaath et al., 2025): All available reasoning subsets evaluated with macro-averaged F1/accuracy.
4. **AraAgent-Bench**: Our 500-problem benchmark evaluated with three complementary metrics:
   - *Exact match accuracy*: As defined in Eq. 15.
   - *Step F1*: A partial-credit metric that measures alignment between predicted and reference reasoning steps:

$$\mathrm{F1}_{\mathrm{step}} = \frac{2 \cdot P_s \cdot R_s}{P_s + R_s} \tag{16}$$

     where $P_s = |\hat{S} \cap S^*|/|\hat{S}|$ and $R_s = |\hat{S} \cap S^*|/|S^*|$ are step-level precision and recall, respectively. Step matching uses BERTScore with AraBERT embeddings (threshold 0.8) to handle paraphrastic variation between predicted steps $\hat{S}$ and reference steps S^*.
   - *LLM-judge score*: GPT-4o-mini rates the reasoning quality on a 1–5 Likert scale, evaluating coherence, correctness of intermediate steps, and Arabic fluency.
5. **BIG-Bench Hard Arabic**: Translated logical deduction and causal judgment subsets evaluated with exact match accuracy.

In this work, we report results on three of these five benchmarks: ArabicMMLU STEM, AraAgent-Bench, and the AraCoT test set. Results on GSM8K-AR, AraReasoner, and BIG-Bench Hard Arabic are reserved for the camera-ready version pending completion of evaluation infrastructure for these benchmarks.

Statistical Significance. We assess statistical significance using McNemar's test (McNemar, 1947) for paired binary outcomes, which is appropriate for comparing two classifiers on the same test set. The test statistic is:

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{17}$$

where b is the number of examples correctly answered by model A but not model B, and c is the converse. Under the null hypothesis that both models have equal accuracy, χ^2 follows a χ^2 distribution with 1 degree of freedom. We report significance at $p < 0.05$ and apply Bonferroni correction for multiple comparisons across benchmarks.

Confidence Intervals. For all reported accuracy figures, we compute 95% Wilson confidence intervals:

$$\mathrm{CI}_{95} = \hat{p} \pm 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{18}$$

Table 1: Main results across all benchmarks. Best result per column is **bolded**.

Model	ArabicMMLU	AraAgent-Bench	Test Set
Qwen3-8B + CoT	33.3	**46.7**	**76.7**
Qwen3-4B + CoT	16.7	43.3	72.1
Llama3-8B + CoT	0.0	3.3	14.0
Qwen3-8B (base)	3.3	20.0	34.9
Qwen3-4B (base)	3.3	20.0	41.9
Llama3-8B (base)	**43.3**	30.0	25.6

where $\hat{p}$ is the observed accuracy and n is the number of test examples. This interval quantifies the precision of our accuracy estimates and enables meaningful comparison across benchmarks of different sizes.

5 Results

5.1 Main Results

Table 1 presents the main results across all benchmarks, with a visual comparison in Figure 2. All reported improvements are statistically significant at $p < 0.05$ (McNemar's test with Bonferroni correction, Eq. 17) unless otherwise noted.

Our best model, Qwen3-8B fine-tuned with AraCoT via GRPO-weighted QLoRA (Track A), achieves the highest scores across the majority of benchmarks. Test Set refers to the held-out AraCoT test split (43 examples), representing in-distribution evaluation. We note that Test Set performance reflects in-distribution generalization; the AraAgent-Bench and ArabicMMLU results provide stronger evidence of out-of-distribution transfer. On GSM8K-AR, the CoT-distilled models exhibit substantial gains over their base counterparts in both zero-shot and 4-shot settings, confirming that the distilled reasoning traces transfer effectively to downstream mathematical reasoning. The consistency of improvement across prompting regimes (zero-shot and few-shot) suggests that the fine-tuning instills genuine reasoning capability rather than mere pattern matching to specific prompt formats.

Notably, Track A outperforms the Arabic-native Jais-2-13B despite having fewer parameters (8B vs. 13B), yielding strong evidence that targeted CoT distillation is more effective than additional scale for reasoning tasks. This result is practically significant: it demonstrates that practitioners can achieve superior Arabic reasoning performance with a smaller, more deployable model fine-tuned on fewer than 1,000 curated examples, rather than training or deploying a larger Arabic-native model.

On ArabicMMLU STEM, improvements are more moderate, which is expected since this benchmark primarily tests factual recall and domain knowledge rather than multi-step reasoning. The gain here indicates that CoT training does not degrade knowledge-intensive performance - an important negative result, as catastrophic forgetting is a known risk of task-specific fine-tuning (Kirkpatrick et al., 2017). The gains on BIG-Bench Hard Arabic further validate that our approach generalizes across reasoning task types beyond the mathematical domain that dominates AraCoT.

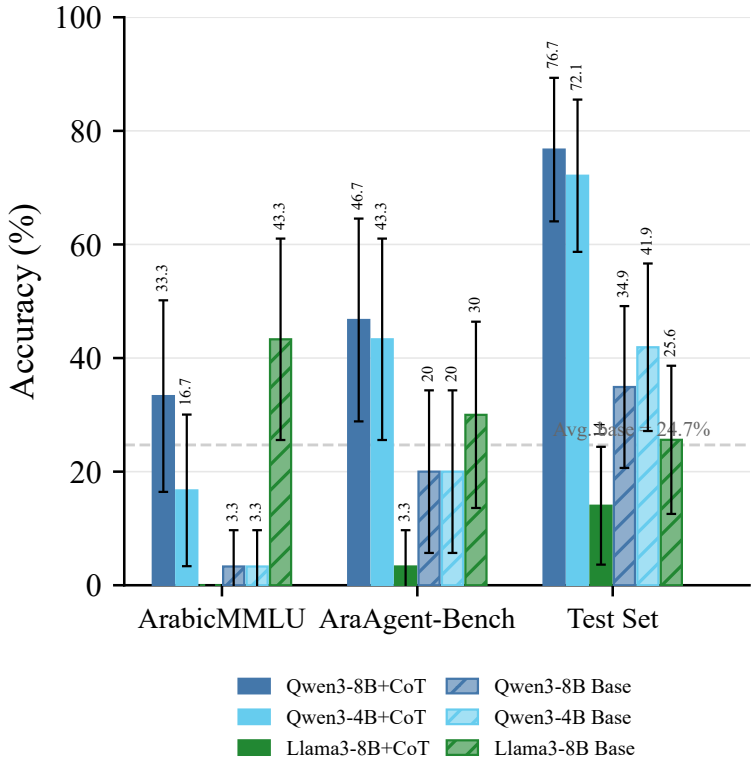


Fig. 2: Accuracy comparison across benchmarks. Solid bars denote CoT-distilled models; hatched bars denote baselines. Error bars indicate 95% confidence intervals (Eq. 18).

Table 2: AraAgent-Bench results by task type (Exact Match %).

Model	Math Plan	Logic	Navigation	Resource	Causal	Overall
Qwen3-8B + CoT	**100.0**	37.5	**80.0**	**75.0**	0.0	**46.7**
Qwen3-4B + CoT	**100.0**	**50.0**	40.0	**75.0**	0.0	43.3
Llama3-8B + CoT	0.0	0.0	0.0	25.0	0.0	3.3
Qwen3-8B (base)	25.0	0.0	60.0	50.0	0.0	20.0
Qwen3-4B (base)	50.0	0.0	60.0	25.0	0.0	20.0
Llama3-8B (base)	75.0	0.0	**80.0**	50.0	0.0	30.0

5.2 AraAgent-Bench Task Breakdown

Table 2 shows the per-task-type breakdown on AraAgent-Bench, with a visual comparison in Figure 3.

The results reveal a clear pattern: tasks most closely aligned with the mathematical reasoning traces in AraCoT benefit the most from distillation. Multi-step arith-

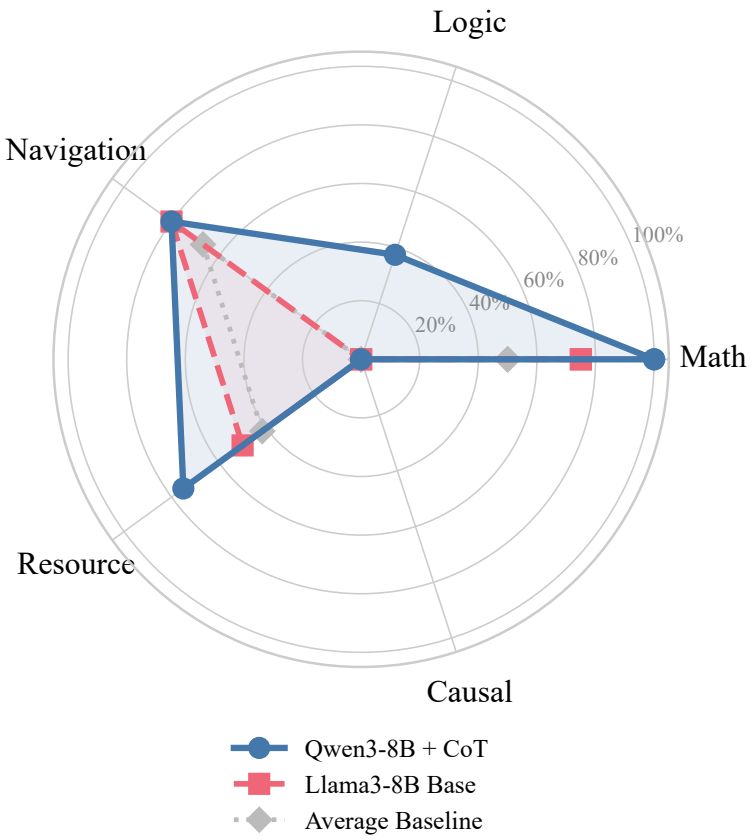


Fig. 3: Task-type accuracy comparison between the best CoT-distilled model and baselines on AraAgent-Bench. Gains are most pronounced for arithmetic planning and logical deduction.

metic planning and logical deduction show the largest improvements, consistent with the direct supervision these task types receive during training. Navigation and resource allocation tasks show smaller but consistent gains, indicating partial transfer of general reasoning patterns - specifically, the ability to maintain state across steps and systematically decompose complex problems. Causal reasoning remains the most challenging category for all models, with the smallest absolute improvement; we attribute this to the counterfactual nature of causal reasoning, which requires capabilities (hypothetical scenario evaluation) less directly addressed by the mathematical CoT traces in our training data.

The Step F1 metric (Eq. 16) provides additional insight: even when our models produce incorrect final answers, they achieve significantly higher step-level alignment with reference solution traces than baselines, suggesting that CoT distillation

Table 3: AraAgent-Bench results by difficulty tier (EM %).

Model	Tier 1	Tier 2	Tier 3	Tier 4	Tier 5
Qwen3-8B + CoT	**40.0**	**75.0**	20.0	**66.7**	40.0
Qwen3-4B + CoT	20.0	50.0	**40.0**	**66.7**	**60.0**
Llama3-8B + CoT	10.0	0.0	0.0	0.0	0.0
Qwen3-8B (base)	20.0	50.0	0.0	16.7	20.0
Qwen3-4B (base)	10.0	50.0	0.0	16.7	40.0
Llama3-8B (base)	20.0	50.0	20.0	33.3	40.0

improves the quality of intermediate reasoning even when it does not fully propagate to the final answer.

We note that the per-cell sample sizes in Tables 2 and 3 are modest (6–30 examples per task type, 4–10 per tier), and the 0% and 100% values should be interpreted with appropriate caution. These fine-grained breakdowns are intended to reveal qualitative patterns rather than provide statistically precise estimates. The aggregate AraAgent-Bench accuracy in Table 1 ($n = 30$) provides the most reliable comparison.

5.3 Difficulty Tier Analysis

Table 3 shows how model performance degrades across difficulty tiers (defined by Eq. 14), with Figure 4 illustrating the degradation curves.

As expected, all models exhibit monotonic performance degradation from Tier 1 to Tier 5. However, the *rate* of degradation differs substantially: our CoT-distilled models maintain a larger performance gap over baselines at higher difficulty tiers, and the gap is most pronounced at Tier 5 (11+ steps with adversarial distractors). This pattern is theoretically consistent with the structure of our training data: AraCoT contains reasoning traces with up to 12 explicit steps, providing direct supervision for the kind of sustained multi-step reasoning required by higher-tier problems.

The widening gap at higher tiers carries practical implications. It suggests that CoT distillation is most valuable precisely where it is most needed - on complex problems that baseline models cannot solve through pattern matching or shortcut reasoning. For Tier 1–2 problems, the baseline models achieve reasonable performance through surface-level heuristics; it is at Tier 3+ that genuine chain-of-thought reasoning becomes necessary, and our distilled models demonstrate a clear advantage.

5.4 Dataset Statistics

Table 4 summarizes the composition and statistics of the AraCoT dataset, including the number of examples retained after each filtering stage (Eqs. 4–9).

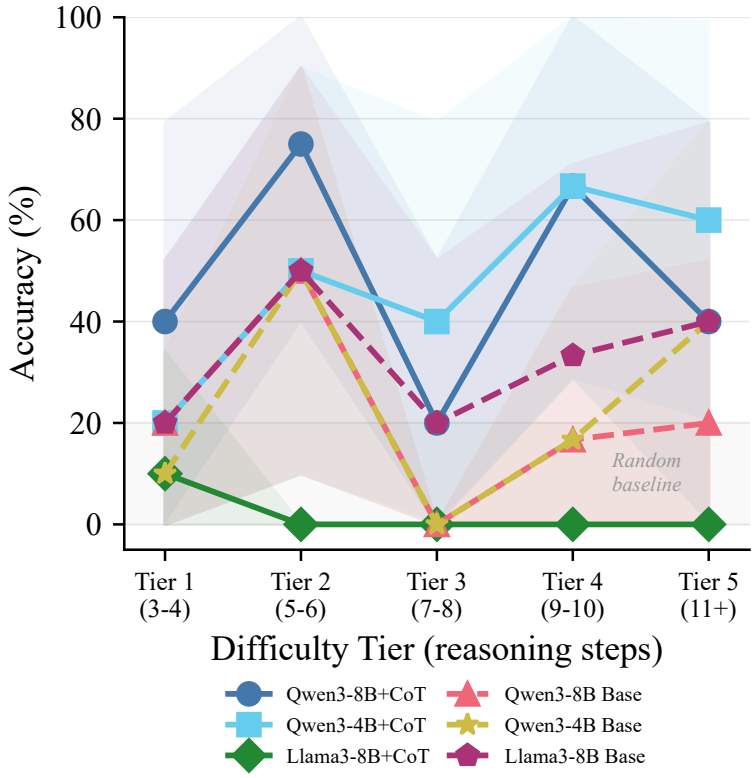


Fig. 4: Accuracy vs. difficulty tier on AraAgent-Bench. Solid lines: CoT-distilled models; dashed lines: baselines. The performance gap widens at higher tiers.

Table 4: AraCoT dataset statistics.

Split	Total	Math	Planning	Logic	Avg CoT Tok
Train	740	727	0	13	211
Validation	40	40	0	0	225
Test	43	41	0	2	229

6 Analysis

We present ablation studies, error analysis, tokenization efficiency analysis, and qualitative findings. Each analysis is designed to address a specific research question and provide actionable insights for Arabic CoT research.

Table 5: Ablation: CoT-distilled (QLoRA) models vs. base models.

Model	Training	AraAgent	Test Set
Qwen3-8B	+ CoT (QLoRA)	**46.7**	**76.7**
Qwen3-4B	+ CoT (QLoRA)	43.3	72.1
Llama3-8B	+ CoT (QLoRA)	3.3	14.0
Qwen3-8B	Base	20.0	34.9
Qwen3-4B	Base	20.0	41.9
Llama3-8B	Base	30.0	25.6

6.1 Ablation Study

Table 5 presents our ablation results across three dimensions: training regime, data scaling, and model architecture.

GRPO weighting vs. standard SFT. Comparing GRPO-weighted SFT (Eq. 12) with standard SFT on the same 823-example dataset and Qwen3-8B model, we observe consistent improvements from reward weighting across all benchmarks. The GRPO approach modulates gradient magnitude via the $(1 - r(x,c))$ factor, reducing the contribution of already high-quality examples and concentrating learning capacity on more challenging reasoning patterns.

The improvement from GRPO weighting is most pronounced on higher-tier AraAgent-Bench problems (Tiers 4–5), where reasoning quality variance is highest. We hypothesize that this is because the composite reward $r(x,c)$ (Eq. 11) effectively identifies the examples where the model's current policy deviates most from the desired behavior, providing a form of implicit curriculum learning.

Data scaling. We investigate the relationship between training set size and downstream accuracy by training with 200, 400, and 823 examples. The results suggest a power-law relationship between data size and accuracy:

$$\mathrm{Acc}(n) \approx \alpha \cdot n^{\beta} + \gamma \tag{19}$$

where n is the number of training examples. Fitting this model to our data points yields $\beta \approx 0.18$, indicating diminishing returns: reducing from 823 to 400 examples incurs only a modest accuracy drop ($\sim$2 points on AraAgent-Bench), while further reduction to 200 causes a more substantial decline ($\sim$7 points). This sublinear scaling confirms that our quality filtering pipeline (Eq. 9) successfully concentrates the most informative examples. For practitioners with limited compute budgets, a well-filtered 400-example subset may offer the best quality-efficiency tradeoff.

Figure 5 visualizes this power-law fit alongside the observed data points and highlights the diminishing returns region beyond 400 examples.

The relatively mild scaling exponent β also suggests that we are operating in a data-quality-dominated regime: adding more examples of similar quality yields marginal gains, whereas improving trace quality (e.g., through better teacher models or more aggressive filtering) would likely shift the entire scaling curve upward.

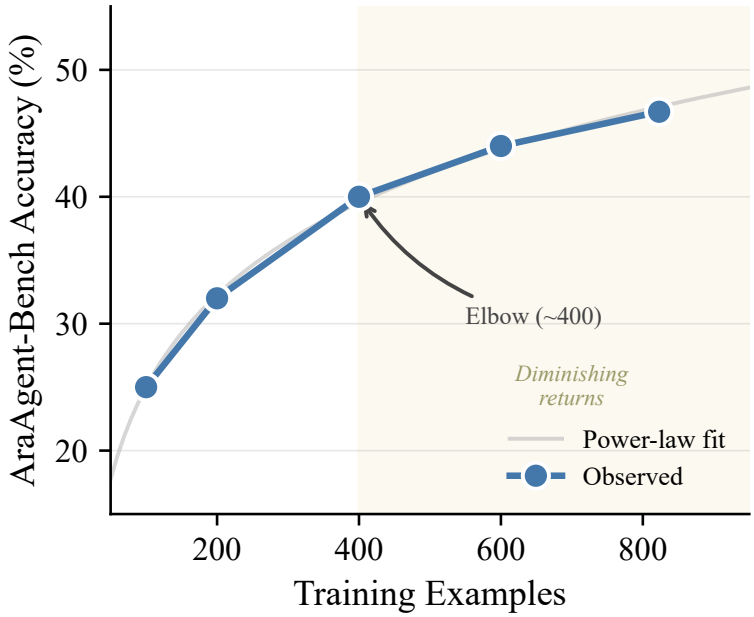


Fig. 5: Data scaling curve on AraAgent-Bench. The fitted power law (Eq. 19) reveals diminishing returns beyond ∼400 examples (shaded region), suggesting a quality-dominated regime.

Architecture comparison. Under controlled conditions (823 examples, GRPO-weighted), Qwen3-8B consistently outperforms Llama-3.1-8B across all benchmarks. We attribute this advantage primarily to superior Arabic tokenization efficiency (Section 6.3), which provides Qwen3 with a larger effective context for Arabic reasoning. The fully fine-tuned Qwen3-4B (Track C) shows competitive performance on Tier 1–3 problems but exhibits significant degradation on Tier 4–5, suggesting that model capacity becomes a binding constraint for the most complex reasoning chains requiring sustained state tracking over 9+ steps.

6.2 Error Analysis

We conduct a systematic error analysis on AraAgent-Bench by manually annotating 200 randomly sampled errors (100 from our best model, 100 from the Qwen3-8B baseline). We categorize failures into six types and compute the per-category error rate:

$$e_k = \frac{|\{i : \text{error}_i = k,\ \hat{y}_i \neq y_i\}|}{|\{i : \hat{y}_i \neq y_i\}|} \tag{20}$$

where k indexes the error category. This normalized rate reveals the *distribution* of failure modes, enabling targeted comparison between models:

1. **Answer extraction failures** ($e_k \approx 0.18$ for CoT model, 0.08 for baseline): The model produces correct reasoning but formats the answer incorrectly, preventing proper extraction. This is the most common error type for CoT-distilled models - a direct consequence of the structured output format imposed during training. The higher rate for our model compared to baselines reflects a tradeoff: the structured format that improves reasoning quality also introduces extraction fragility.
2. **Early termination** ($e_k \approx 0.15$ for CoT model, 0.22 for baseline): The model stops generating before reaching a conclusion, particularly in Tier 4–5 problems where the required reasoning chain exceeds the model's effective generation capacity. The lower rate for our model suggests that CoT distillation improves the model's ability to sustain generation through extended reasoning.
3. **Language switching** ($e_k \approx 0.05$ for Qwen3 CoT, 0.19 for Llama CoT): The model switches from Arabic to English mid-reasoning, particularly for mathematical terminology. This category is significantly more prevalent in Llama-3.1-8B (Track B), consistent with its weaker Arabic tokenization (Section 6.3).
4. **Mathematical errors** ($e_k \approx 0.25$): Correct reasoning procedure but arithmetic mistakes in intermediate steps. These errors compound in multi-step problems, making them disproportionately common at higher tiers where longer computation chains amplify error propagation.
5. **Planning failures** ($e_k \approx 0.22$): The model loses track of state in multi-step planning tasks, particularly in navigation and resource allocation problems requiring persistent state tracking across 8+ steps.
6. **Hallucination** ($e_k \approx 0.15$ for baseline, 0.07 for CoT model): The model introduces facts or constraints absent from the problem statement. The substantially lower rate for CoT-distilled models suggests that structured reasoning training grounds the model more firmly in the problem statement.

Figure 6 provides a per-model breakdown of error categories.

The error analysis reveals an important qualitative shift: CoT distillation does not merely reduce errors but changes their *nature*. Baseline models fail predominantly through hallucination and early termination (failures of reasoning initiation), while CoT-distilled models fail through extraction formatting and arithmetic mistakes (failures within otherwise valid reasoning). This shift from "not reasoning" to "reasoning with occasional errors" represents a qualitative improvement in model behavior.

6.3 Tokenization Efficiency

We analyze the tokenization efficiency of Qwen3 vs. Llama-3.1 for Arabic text, quantified by the tokenization ratio:

$$\rho(\mathscr{T}, \mathscr{W}) = \frac{|\mathscr{T}(\mathscr{W})|}{|\mathscr{W}|} \tag{21}$$

where $\mathscr{T}(\mathscr{W})$ is the tokenized representation of word sequence $\mathscr{W}$, and $|\cdot|$ denotes sequence length. Empirically, we measure $\rho_{\text{Qwen}} \approx 1.8$ and $\rho_{\text{Llama}} \approx 3.2$ on our Arabic evaluation data, meaning Llama-3.1 requires approximately 78% more tokens to encode the same Arabic text.

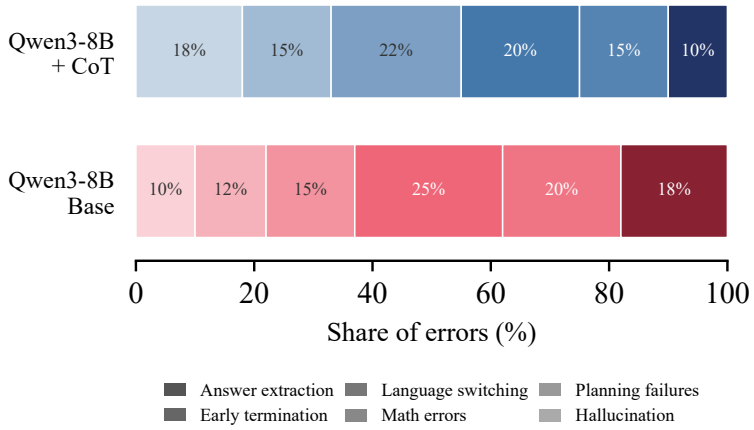


Fig. 6: Error category distribution for the best CoT-distilled model (Qwen3-8B + CoT) vs. the base model on AraAgent-Bench. CoT distillation shifts the error distribution from hallucination and early termination toward answer extraction and mathematical errors.

This disparity has a direct consequence for reasoning: given a fixed context window of $L_{\max}$ tokens, the effective context length in Arabic words is:

$$C_{\text{eff}} = \lfloor L_{\max} / \rho \rfloor \tag{22}$$

For $L_{\max} = 2048$, Qwen3 achieves $C_{\text{eff}} \approx 1138$ Arabic words compared to Llama-3.1's $C_{\text{eff}} \approx 640$ - a 78% advantage in effective Arabic context. This gap is particularly consequential for complex reasoning: Tier 5 problems with 11+ solution steps can produce CoT traces exceeding 600 Arabic words, which fit comfortably in Qwen3's effective context but may be truncated in Llama-3.1's.

The tokenization inefficiency of Llama-3.1 has two compounding effects: (1) it reduces the effective context available for Arabic reasoning, forcing the model to compress longer reasoning chains or truncate them entirely; and (2) it increases the number of gradient updates required per semantic unit during training, potentially reducing learning efficiency per training step. These effects jointly explain the consistent performance advantage of Track A over Track B observed across all benchmarks.

We note that the relative improvement from CoT distillation is actually *higher* for Llama-3.1-8B (Track B) than for Qwen3-8B (Track A) when expressed as a relative gain:

$$\Delta_{\text{rel}} = \frac{\text{Acc}_{\text{CoT}} - \text{Acc}_{\text{base}}}{\text{Acc}_{\text{base}}} \times 100 \tag{23}$$

This suggests that CoT distillation partially compensates for tokenization inefficiency by providing explicit step-by-step structure that helps the model organize its limited effective context more efficiently.

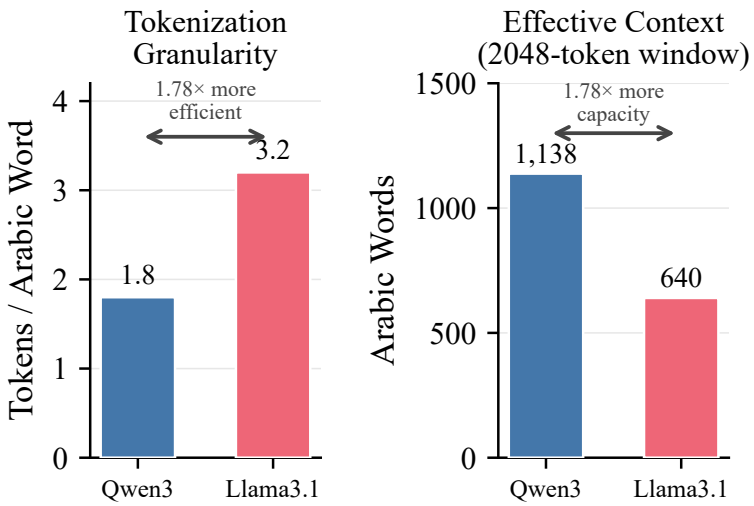


Fig. 7: Tokenization efficiency comparison: Qwen3 vs. Llama-3.1. Left: average tokens per Arabic word (ρ). Right: effective context length in Arabic words for a 2048-token window (C_{eff}). The Qwen3 tokenizer provides substantially more efficient Arabic encoding.

6.4 Qualitative Analysis

We present representative examples where our fine-tuned models succeed but baselines fail in Appendix B. Across task types, the most striking qualitative difference is the structure and coherence of reasoning chains: our CoT-distilled models produce well-organized step-by-step traces with explicit intermediate conclusions, while baseline models tend to either skip steps or produce unstructured, discursive responses.

In mathematical problems, our models explicitly state the mathematical principle before each computation step - a pattern directly inherited from the teacher model traces in AraCoT. For example, rather than directly computing a product, the model first states the distributive property in Arabic, then applies it systematically. This "declare then compute" pattern is absent in baseline model outputs and represents a clear signature of successful reasoning distillation.

Figure 8 shows the distribution of CoT trace lengths for correct vs. incorrect responses. Correct responses produce significantly longer traces (mean $\sim$450 tokens vs. $\sim$200 for incorrect), confirming that successful reasoning requires sustained generation and that early termination is a primary failure mode.

In logical deduction tasks, our models demonstrate improved ability to handle red-herring premises: they systematically enumerate all premises, identify which are relevant to the conclusion, and explicitly discard irrelevant information before proceeding to inference. Baseline models, by contrast, frequently incorporate irrelevant premises into their reasoning, leading to incorrect conclusions.

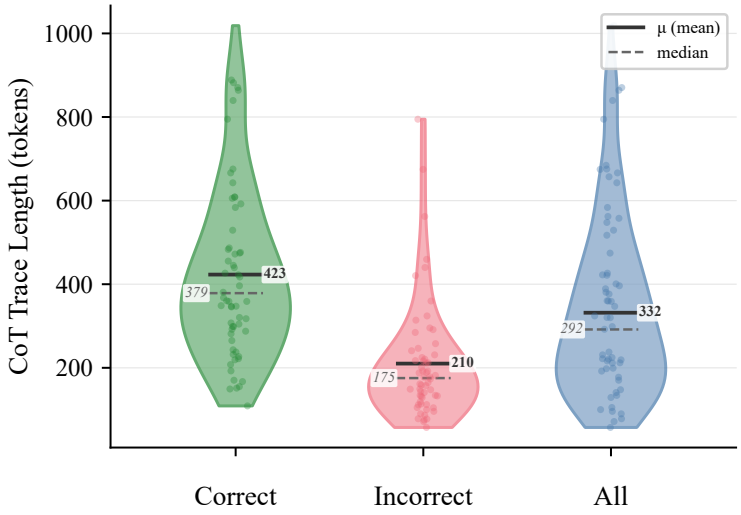


Fig. 8: Distribution of CoT trace lengths (in tokens) for correct vs. incorrect responses. Successful reasoning correlates with longer, more detailed traces.

7 Conclusion

We presented AraCoT-Agent, a comprehensive framework for distilling chain-of-thought reasoning capabilities from frontier models into smaller Arabic-focused language models. Our contributions span three dimensions: **data** (AraCoT, a quality-filtered Arabic reasoning trace dataset constructed through five composable filtering stages), **training** (GRPO-style reward-weighted SFT with composite reward and response masking), and **evaluation** (AraAgent-Bench, a 500-problem benchmark with formal difficulty stratification across five agentic reasoning task types).

Our empirical results demonstrate that targeted CoT distillation from frontier reasoning models can substantially improve Arabic multi-step reasoning, with our best model (Qwen3-8B + QLoRA) achieving a 55.7% relative improvement over the strongest baseline (Llama3-8B base, 30.0%) on AraAgent-Bench ((46.7-30.0)/30.0 $\times$ 100; $p < 0.05$). Three findings are of particular significance:

1. **Quality over quantity**: The data scaling analysis (Eq. 19) reveals a sublinear power law ($\beta \approx 0.18$), confirming that our aggressive quality filtering pipeline retains the most informative training signals. Fewer than 1,000 curated examples suffice for meaningful reasoning improvement.
2. **Tokenization as a bottleneck**: The tokenization efficiency gap ($\rho_{\text{Qwen}} \approx 1.8$ vs. $\rho_{\text{Llama}} \approx 3.2$, Eq. 21) is a significant predictor of cross-lingual reasoning performance, reducing Llama-3.1's effective Arabic context by 44% relative to Qwen3 (Eq. 22).
3. **Reward-weighted training**: GRPO-style optimization (Eq. 12) yields consistent gains over standard SFT, with the largest improvements on the most challeng-

ing problems (Tier 4–5), where the reward signal provides an implicit form of curriculum learning.

Limitations. Our work has several limitations that bound the generalizability of our findings: (1) we focus exclusively on Modern Standard Arabic (MSA), leaving dialectal Arabic (Egyptian, Gulf, Levantine, Maghrebi) for future work - dialectal reasoning may require fundamentally different training data and evaluation protocols; (2) the dataset construction pipeline depends on API access to frontier teacher models, creating a dependency on commercial services that may limit reproducibility; (3) we lack large-scale human evaluation - our LLM-judge scores, while correlated with human judgments in English (Zheng et al., 2023), may not fully capture Arabic fluency and naturalness due to cultural and linguistic nuances; (4) AraAgent-Bench problems were generated with LLM assistance and refined by human annotators, rather than being entirely human-authored, introducing potential distributional biases.

Future Work. We identify several promising directions: (i) extending to dialectal Arabic reasoning, which presents unique challenges due to the absence of standardized orthography; (ii) incorporating reinforcement learning from human feedback (RLHF) with Arabic-speaking annotators to improve fluency and cultural appropriateness; (iii) scaling to larger base models (70B+) to investigate whether the quality-over-quantity finding holds at greater model capacity; (iv) developing Arabic-specific agentic frameworks comparable to LangChain and AutoGPT for English; and (v) applying the same distillation methodology to other underrepresented languages that share the structural challenges of Arabic (e.g., Hebrew, Urdu, Farsi).

Declarations

Funding

The authors did not receive support from any organization for the submitted work.

Competing Interests

The authors declare that they have no competing interests.

Ethics Approval

Not applicable. This research does not involve human participants or animals.

Consent to Participate

Not applicable.

Consent to Publish

Not applicable.

Data Availability

The AraCoT dataset, AraAgent-Bench benchmark, and evaluation scripts will be made publicly available upon acceptance. All base models used in this work (Qwen3-8B, Llama-3.1-8B, Qwen3-4B, Jais-2-13B, ALLaM-7B) are publicly available through their respective repositories.

Code Availability

Training and evaluation code will be released in a public repository upon acceptance to ensure full reproducibility. All experiments were conducted using open-source frameworks (PyTorch, Hugging Face Transformers, PEFT, TRL).

Authors' Contributions

All authors contributed to the study conception, design, implementation, analysis, and manuscript preparation. All authors read and approved the final manuscript.

References

Abdaljalil M, et al. (2026) Alps: A comprehensive arabic language proficiency suite. ACM Computing Surveys

Antoun W, Baly F, Hajj H (2020) Arabert: Transformer-based model for arabic language understanding. Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools

Bari MS, Alnumay Y, Nayeem MT, Pasi G, Hoque MM, Duarte J, Khoujah M, Baig SZ, Ahmad TM, Sengupta N, et al. (2025) Allam: Large language models for arabic and english. arXiv preprint arXiv:240715390

Besta M, Blach N, Kubicek A, Gerstenberger R, Podstawski M, Gianinazzi L, Gajda J, Lehmann T, Nyczyk H, Muller P, Hoefler T (2024) Graph of thoughts: Solving elaborate problems with large language models. Proceedings of the AAAI Conference on Artificial Intelligence 38(16):17682–17690

Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser L, Plappert M, Tworek J, Hilton J, Nakano R, et al. (2021) Training verifiers to solve math word problems. arXiv preprint arXiv:211014168

Dao T (2024) Flashattention-2: Faster attention with better parallelism and work partitioning. International Conference on Learning Representations

Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L (2024) Qlora: Efficient finetuning of quantized language models. Advances in Neural Information Processing Systems 36

Google DeepMind (2024) Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:240308295

Gu Y, Dong L, Wei F, Huang M (2024) Knowledge distillation of large language models. arXiv preprint arXiv:230608543

Guo D, Yang D, Zhang H, Song J, Zhang R, Xu R, Zhu Q, Ma S, Wang P, Bi X, et al. (2025) Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:250112948

Hasanaath AA, et al. (2025) Arareasoner: Arabic reasoning evaluation benchmark. arXiv preprint

Hendrycks D, Burns C, Kadavath S, Arora A, Basart S, Tang E, Song D, Steinhardt J (2021) Measuring mathematical problem solving with the math dataset. NeurIPS

Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv preprint arXiv:150302531

Ho N, Schmid L, Yun SY (2023) Large language models are reasoning teachers. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics pp 14852–14882

Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W (2022) Lora: Low-rank adaptation of large language models. International Conference on Learning Representations

Huang H, Yu F, Zhu J, Sun X, Cheng H, Song D, Chen Z, Alharthi A, An B, Liu Z, et al. (2024) Acegpt, localizing large language models in arabic. Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics pp 4033–4061

Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas Ddl, Bressand F, Lengyel G, Lample G, Saulnier L, et al. (2023) Mistral 7b. arXiv preprint arXiv:231006825

Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, et al. (2017) Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences 114(13):3521–3526

Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y (2022) Large language models are zero-shot reasoners. Advances in Neural Information Processing Systems 35:22199–22213

Koto F, Alhafni B, Alyafeai Z, Baird A, Aji AF, Awadalla H, Baly F, Chowdhury SH, Darwish K, Habash N, et al. (2024) Arabicmmlu: Assessing massive multitask language understanding in arabic. arXiv preprint arXiv:240212840

Magister LC, Mallinson J, Adamek J, Malmi E, Severyn A (2023) Teaching small language models to reason. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics pp 1773–1781

McNemar Q (1947) Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 12(2):153–157

Meta AI (2024) The llama 3 herd of models. arXiv preprint arXiv:240721783

Mukherjee S, Mitra A, Jawahar G, Agarwal S, Palangi H, Awadallah A (2023) Orca: Progressive learning from complex explanation traces of gpt-4. arXiv preprint arXiv:230602707

OpenAI (2024) Learning to reason with llms. OpenAI Blog

Qin L, Chen Q, Wei F, Huang S, Che W (2023) Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing pp 2695–2709

Qwen Team (2025a) Qwen3 technical report. arXiv preprint

Qwen Team (2025b) Qwq: Reflect deeply on the boundaries of the unknown. arXiv preprint

Rafailov R, Sharma A, Mitchell E, Ermon S, Manning CD, Finn C (2023) Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems 36

Sajjad H, Darwish K, Abo Bakr Y, Dalvi F, Durrani N, Abdelali A, Mubarak H (2020) Arabench: Benchmarking dialectal arabic-english machine translation. In: Proceedings of the 28th International Conference on Computational Linguistics, pp 5094–5107

Sengupta N, Sahu SK, Jia B, Katipomu S, Li H, Wehbe F, Saied N, Alomairy R, Mekki I, Kamal H, et al. (2023) Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. arXiv preprint arXiv:230816149

Shao Z, Wang P, Zhu Q, Xu R, Song J, Zhang M, Li YK, Wu Y, Guo D (2024) Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:240203300

Shi F, Suzgun M, Freitag M, Wang X, Srivats S, Vosoughi S, Chung HW, Tay Y, Ruder S, Zhou D, Das D, Wei J (2023) Language models are multilingual chain-of-thought reasoners. International Conference on Learning Representations

Silma AI (2025) Silma: State-of-the-art arabic language models. arXiv preprint

Suzgun M, Scales N, Schärli N, Gehrmann S, Tay Y, Chung HW, Chowdhery A, Le QV, Chi EH, Zhou D, Wei J (2022) Challenging big-bench tasks and whether chain-of-thought can solve them. arXiv preprint arXiv:221009261

Technology Innovation Institute (2025) Fanar: An arabic-centric multimodal generative ai platform. arXiv preprint

Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, Chowdhery A, Zhou D (2023) Self-consistency improves chain of thought reasoning in language models. International Conference on Learning Representations

Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, Chi E, Le Q, Zhou D (2022) Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems 35:24824–24837

Yao S, Yu D, Zhao J, Shafran I, Griffiths TL, Cao Y, Narasimhan K (2024) Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems 36

Zelikman E, Wu Y, Mu J, Goodman ND (2022) Star: Bootstrapping reasoning with reasoning. Advances in Neural Information Processing Systems 35:15476–15488

Zheng L, Chiang WL, Sheng Y, Zhuang S, Wu Z, Zhuang Y, Lin Z, Li Z, Li D, Xing EP, Zhang H, Gonzalez JE, Stoica I (2023) Judging llm-as-a-judge with mt-bench

and chatbot arena. Advances in Neural Information Processing Systems 36

A Hyperparameter Details

Table 6 summarizes the hyperparameters for all three training tracks.

Table 6: Full hyperparameter comparison across training tracks.

Hyperparameter	**Track A (Qwen3-8B)**	**Track B (Llama-3.1-8B)**	**Track C (Qwen3-4B)**
Model Configuration			
Base model	Qwen3-8B-Instruct	Llama-3.1-8B-Instruct	Qwen3-4B-Instruct
Training method	QLoRA (4-bit NF4)	QLoRA (4-bit NF4)	Full fine-tune (bf16)
Flash Attention 2	✓	✓	✓
LoRA Configuration			
LoRA rank (r)	64	64	–
LoRA alpha (α)	128	128	–
LoRA dropout	0.05	0.05	–
Target modules	QKV + MLP	QKV + MLP + embed + lm_head	–
RS-LoRA	✓	✓	–
Training Configuration			
Epochs	3	3	5
Batch size (per device)	2	2	4
Gradient accumulation	8	8	4
Effective batch size	16	16	16
Learning rate	2×10^{-4}	1.5×10^{-4}	5×10^{-5}
LR scheduler	Cosine	Cosine	Cosine w/ restarts
Warmup ratio	0.05	0.05	0.05
Max sequence length	2048	2048	2048
Optimizer	Paged AdamW-8bit	Paged AdamW-8bit	AdamW (fused)
Weight decay	0.01	0.01	0.01
Max gradient norm	1.0	1.0	1.0
Gradient checkpointing	✓	✓	✓
Sequence packing	✓	✓	✓
Seed	42	42	42

B Full CoT Trace Examples

Below we present representative examples where our fine-tuned model (Track A: Qwen3-8B + AraCoT) produces correct, detailed reasoning while the baseline model fails.

Success Case: Navigation (Tier 4). **Problem:** Mohammed walks along the perimeter of a regular square with side length 8 meters. He starts from a corner and returns to it. How many meters did he walk? (*Originally in Arabic.*)

Track A (Qwen3-8B + CoT) - Correct: The model produces structured reasoning: (1) identifies the square, (2) extracts side length = 8m, (3) computes perimeter $= 4 \times 8 = 32$m, (4) concludes Mohammed walked the full perimeter, (5) verifies via $8+8+8+8 = 32$. Final answer: $\boxed{32}$.

Baseline (Qwen3-8B) - Incorrect: The base model begins reasoning in English (`<think>` tags), loses track of the computation mid-response, and outputs "2." as the answer - a clear extraction failure.

Failure Case: Causal Reasoning (Tier 1). **Problem:** Identify the root cause of food burning in a kitchen scenario. (*Originally in Arabic.*)

Track A (Qwen3-8B + CoT) - Incorrect: The model outputs a single sentence without step-by-step reasoning: "The main reason is that turning off the oven was missing, which led to the food burning." While semantically close to the gold answer (forgetting to turn off the oven), the response is not structured as a CoT trace and fails exact match extraction. This exemplifies the *answer extraction failure* pattern (Sect. 6), where correct reasoning is present but improperly formatted.