Silah at QIAS 2026: Fine-Tuning vs. Retrieval-Augmented Generation for Islamic Inheritance Reasoning

Ghader Kurdi, Hanan Justanieah, and Hala Justanieah

Department of Data Science, College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia
Department of Jurisprudence, College of Shari'ah and Islamic Studies Umm Al-Qura University, Makkah, Saudi Arabia
The Higher Division, Institute of the Holy Grand Mosque in Makkah, Makkah, Saudi Arabia
{grkurdi, hmjustaniah}@uqu.edu.sa, dr.hala.just@gmail.com

Abstract

Islamic inheritance is a highly structured, rule-intensive domain that requires precise reasoning. The QIAS 2026 shared task introduces a benchmark for evaluating generative artificial intelligence on end-to-end inheritance problem solving. In this paper, we present our team Silah's participation, comparing three approaches: 1) a multi-stage retrieval-augmented, rule-guided pipeline, 2) supervised fine-tuning of generative large language models, and 3) retrieval-augmented fine-tuning. We evaluate several open-source models, including Qwen2.5, LLaMA, DeepSeek, and Fanar. Our results show that supervised fine-tuning consistently outperforms retrieval-based approaches, with the fine-tuned Fanar-1-9B-Instruct model achieving the best performance (MIR-E = 0.83). These findings suggest that learning implicit reasoning patterns through fine-tuning is more effective than explicit rule injection under current retrieval setups, highlighting the need for more precise and selective rule retrieval in future approaches.

Keywords: Islamic Inheritance, Large Language Models, Generative AI, RAG, Fine-Tuning

1. Introduction

The Islamic law of inheritance, known as Ilm al-Fara'id or al-Mawarith, is a fundamental branch of Islamic jurisprudence. It is characterised by a highly structured and requires precise reasoning to determine heirs, assess eligibility, and allocate shares. In addition, the distribution of shares may involve mathematical adjustments, increasing complexity. As such, it provides a well-defined yet challenging domain for evaluating advanced reasoning capabilities in artificial intelligence (AI).

The Question-and-Answer in Islamic Studies Assessment (QIAS 2026) shared task provides a benchmark for evaluating generative AI models in Islamic inheritance problem solving. The task requires interpreting natural-language scenarios and producing structured outputs that include identified heirs, blocked heirs, assigned shares, adjustment mechanisms such as awl and radd, and the final distribution of the estate. This highlights the dual challenge of linguistic understanding and adherence to domain-specific rules.

In this work, team Silah (صِلة) explores three generative AI approaches for the QIAS 2026 shared task: (1) a multi-stage retrieval-augmented, rule-guided pipeline (RAG-P) based on retrieval-augmented generation (RAG) (Lewis et al., 2020), where relevant rules are dynamically retrieved and injected into the prompt; (2) supervised fine-tuning (SFT) on the QIAS dataset; and (3) retrieval-augmented fine-tuning (RAG-FT), which follows the same training procedure as SFT but incorporates domain-specific rules into the prompt only at inference time to guide predictions.

The main contributions of this paper are twofold: a comparative evaluation of RAG, SFT, and RAG-FT for Islamic inheritance reasoning, and an empirical study across multiple open-source LLMs on the QIAS 2026 benchmark.

2. Background

2.1. Task Definition

The QIAS 2026 shared task focuses on Islamic inheritance reasoning, where the goal is to automatically solve inheritance scenarios described in natural language. Each scenario is presented in Arabic, specifying heirs and their relationships to the deceased (e.g., مات وترك: بنت ابن وأب الأب وأم الأب وأب أب الأب وابنان وأم وأربعة إخوة لأم. ما هو نصيب كل وريث؟).

Given such input, the system must interpret the scenario, apply Islamic inheritance rules, and generate a structured JSON output. The expected output comprises multiple components: eligible heirs and their counts, blocked heirs, legal shares, whether awl or radd applies, the estate origin (asl al-mas'ala), and the final distribution of shares following ta'sil and adjustment when necessary.

Overall, the task combines natural language understanding, rule-based reasoning, and numerical computation, requiring models to perform multi-step decision-making in accordance with Islamic inheritance law.

Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks, pages 213–219

2.2. Dataset Description

The dataset provided for the shared task (Bouchekif et al., 2026) consists of 12,000 training instances and 500 test instances, with 200 training instances reserved for development. Each instance includes a unique identifier, a natural-language question describing an inheritance scenario, a reasoning trace, a category label, and a structured JSON output.

The dataset exhibits a clear class imbalance, with awl and radd cases underrepresented compared to simple cases. As reported in (Bouchekif et al., 2026), awl and radd account for 4.81% (n=577) and 2.87% (n=344), respectively, in the training set, and 7.80% (n=39) and 1.00% (n=5) in the test set. In this work, we use the dataset as provided, without applying resampling or rebalancing strategies.

2.3. Related Work

Early work on Islamic inheritance focused on rule-based and expert systems for automating share calculations (Kurdi, 2026). With the advent of generative AI, there has been growing interest in evaluating its understanding of Islamic inheritance. Initial studies (Bouchekif et al., 2025b; Justanieah and Kurdi, 2025) have explored this capability using multiple-choice questions (MCQs). For example, (Justanieah and Kurdi, 2025) evaluated ChatGPT on a limited set of undergraduate-level exam MCQs using a zero-shot prompting setup. The questions included both general knowledge of Islamic inheritance and scenario-based cases; however, due to the nature of MCQs, they typically focused on isolated aspects of inheritance problems, such as determining the share of a specific heir or identifying blocked heirs, rather than requiring complete end-to-end solutions. The study reported an accuracy of 65.35%, concluding that although ChatGPT demonstrated useful explanatory and reasoning abilities, it lacked the precision required for problems that demand both correct juridical interpretation and accurate mathematical computation.

Recent work has also emphasised the need for robust evaluation frameworks. (Kurdi and Justanieah, 2025) introduced a structured benchmark for inheritance calculation systems, showing that previously reported high-performing systems may perform significantly worse on more comprehensive cases. This emphasis on systematic evaluation was further extended in the QIAS 2025 shared task on Islamic Inheritance Reasoning and Islamic Knowledge Assessment (Bouchekif et al., 2025a), which provided a large-scale benchmark for evaluating language models on inheritance and broader Islamic knowledge. Inheritance reasoning in QIAS 2025 was also formulated as an MCQ answering task, enabling systematic comparison across participating systems. The task demonstrated that strong performance can be achieved using a variety of strategies, including fine-tuning (Elrefai et al., 2025), RAG (Hamed et al., 2025), combining fine-tuning with RAG (Mohammad, 2025), and ensemble-based approaches (AlDahoul and Zaki, 2025). Although QIAS 2025 enabled systematic comparison, it remained limited to answer selection rather than full end-to-end inheritance reasoning. To address this, QIAS 2026 introduces a more challenging setting in which systems are required to generate structured outputs that reflect the complete reasoning process.

3. System Overview

We explore three approaches for Islamic inheritance reasoning: 1) RAG-P, 2) SFT, and 3) RAG-FT. All approaches are built on top of pre-trained generative large language models. These approaches are described in the following sections, while implementation details, including prompts and the curated rule base, are publicly available at: `https://github.com/grkurdi/silah-inheritance-reasoning-llm`.

3.1. Base Models

We experimented with multiple open-source models, varying in size, architecture, and training paradigms, to assess their effectiveness for Islamic inheritance reasoning. Specifically, we evaluated four model families: Qwen: *Qwen2.5-3B-Instruct* (Yang et al., 2024), LLaMA: *Meta-Llama-3-8B-Instruct* (Grattafiori et al., 2024), DeepSeek: *DeepSeek-R1-Distill-Qwen-7B* (DeepSeek-AI, 2025), and Fanar: *Fanar-1-9B-Instruct* (Abbas et al., 2025). These models provide a balance between general-purpose reasoning and Arabic-focused modeling, with *Fanar* optimised for Arabic, while Qwen2.5, LLaMA, and DeepSeek offer strong general reasoning performance.

3.2. Multi-Stage Retrieval-Augmented Rule-Guided Pipeline

We propose a multi-stage, rule-guided approach for Islamic inheritance problem solving. Rather than generating the final answer in a single step, the pipeline decomposes the task into five stages: 1) special-case detection, 2) identification of heirs, blocked heirs, heir types, and initial shares, 3) computation of the estate origin and detection of broken groups in which shares cannot be evenly distributed among group members without fractions, 4) correction and determination of awl or radd, and 5) final adjustment as needed.

First, the question is preprocessed by extracting and normalising heir mentions. Second, relevant rules are retrieved from a manually curated rule base. Third, a stage-specific prompt is constructed and sent to the model, incorporating task instructions, the retrieved rules, and, when applicable, the structured output of the previous stage. This sequential design improves control, interpretability, and alignment with the procedural nature of Islamic inheritance reasoning.

The rule base is derived from educational materials, primarily undergraduate lecture slides prepared by the authors. The rules are organised into JSON files by category, including special cases (22), fixed shares (26), asaba (11), blocking (22), asl al-mas'ala (7), adjustment (4), and awl/radd (13). Examples are provided in the Appendix.

Rule retrieval is performed in a stage-specific manner, where each stage selects a subset of rules tailored to its objective. Retrieval is based on the fields *heirs*, *required_heirs*, *optional_heirs_any*, and *conditions*. For example, in the special-case detection stage, a rule is selected if all *required_heirs* are present in the question. If *optional_heirs_any* is non-empty, at least one of its heirs must also be present. If both fields are empty, the rule is selected if any heir listed in *heirs* appears in the question.

3.3. Supervised Fine-tuning

We fine-tune a pretrained language model using parameter-efficient fine-tuning (PEFT) with Low-Rank Adaptation (LoRA) (Hu et al., 2022). LoRA adapters are applied to key projection layers (e.g., query, key, value, and feed-forward projections), allowing efficient adaptation with a reduced number of trainable parameters. Fine-tuning configuration and training settings are provided in the Appendix.

At inference, the model generates textual outputs that are parsed into structured JSON. Post-processing ensures schema consistency through field normalisation and structure validation. Specifically, normalisation functions enforce correct data types (e.g., integers, floats), standardise list formats (e.g., heirs and shares), and handle malformed or partial outputs. A final validation step ensures that each prediction conforms to the required schema before inclusion in the results.

3.4. Retrieval-Augmented Fine-Tuning

In addition to baseline fine-tuning, we incorporate a lightweight retrieval component that provides explicit inheritance rules at inference time. This approach follows the same supervised fine-tuning procedure as SFT and does not include retrieved rules during training; instead, retrieval is applied only at inference to guide predictions. We reuse the same curated rule set as in RAG-P for consistency.

The approach introduces three components. First, a preprocessing layer extracts and normalises heir mentions, decomposing compound expressions and standardising lexical variants into canonical singular forms. Second, a retrieval module selects relevant rules based on detected heirs by matching against rule metadata (e.g., *heirs*, *required_heirs*, *optional_heirs_any*) and returning a deduplicated set of rules. Third, instead of prompting the model with only task instructions and the question, the RAG-FT variant prepends the retrieved rules to the prompt.

4. Experimental Setup

4.1. Data Splits

We follow the official data splits. The models are trained on the training set, with the validation set used for parameter tuning and preliminary evaluation. The final results are reported on the test set, which is used solely for benchmarking.

4.2. Evaluation Metrics

The official evaluation metric is the MIR-E score, which ranges from 0 to 1 and is calculated as a weighted sum of stage-level scores. The weights are defined as $\lambda_1 = 0.30$ for eligible and blocked heirs, $\lambda_2 = 0.20$ for legal shares, $\lambda_3 = 0.10$ for the awl/radd decision, and $\lambda_4 = 0.20$ for the final distribution:

$$\text{MIR-E} = \lambda_1 S_{heirs} + \lambda_2 S_{shares} + \lambda_3 S_{awl} + \lambda_4 S_{final}$$

Each stage is evaluated according to its output type. Set-valued stages (eligible and blocked heirs) are assessed using set overlap, independent of order. The awl/radd stage, where the model determines whether the case involves awl, radd, or neither, is considered correct only if the predicted category matches the gold label and previous stages are correct. The numerical stages (legal shares and final distribution) are evaluated using a tolerance-based criterion, where the predictions are considered correct if they fall within 0.1 of the gold values.

5. Results

Our best result achieved seventh place on the leaderboard[1] with an MIR-E score of 0.83, demonstrating competitive performance compared to other submissions, where scores ranged from 0.74 to 0.94. Table 1 presents the performance of different models and approaches on the test set.

RAG-P achieved low performance across all models, with scores ranging from 0.17 to 0.38, highlighting limitations in retrieval precision and its im-

[1] https://sites.google.com/view/qias2026/leaderboards

pact on reasoning. In particular, the retrieval step often returns multiple applicable rules for the same heir without sufficiently disambiguating the context. For example, a daughter may receive a fixed share of 1/2 or 2/3 of the estate, or, when combined with a son or a sister of the deceased, become a residuary heir and inherit the remainder after fixed shares are allocated, depending on the case. However, rules corresponding to all these scenarios are often retrieved simultaneously, introducing ambiguity and potentially confusing the model during inference.

Within this multi-stage RAG setting, performance varied across models. Fanar achieved the best results, while DeepSeek obtained the lowest score. A closer analysis revealed that DeepSeek often extracted only heirs, while other key components such as blocked heirs and the post-tasil distribution in the JSON output were frequently missing. This behaviour can be attributed to inconsistent adherence to the prompt instructions, which require Arabic-only responses and a strict output format. The model occasionally generated non-Arabic content (e.g., Chinese tokens) and deviated from the required structure, negatively affecting its results.

Based on these findings, we selected the best-performing models for fine-tuning. Fine-tuning significantly improved performance, with LLaMA and Fanar achieving MIR-E scores of 0.81 and 0.83, respectively. For the RAG-FT configuration, we used the best-performing base model, Fanar, which achieved a score of 0.74. These results highlight the effectiveness of fine-tuning over RAG-based approaches for Islamic inheritance reasoning, particularly in capturing structured rule application and producing accurate final distributions.

The best results were achieved using supervised fine-tuning, while RAG-P performed significantly worse, and RAG-FT yielded intermediate performance. Both RAG-P and RAG-FT introduced noise due to limitations in retrieval precision, as the retrieved rules were sometimes excessive and, in other cases, incomplete. Therefore, careful design of the retrieval mechanism is essential to ensure the selection of minimal, precise, and contextually relevant rules that capture interactions among combinations of heirs, rather than rules triggered solely by the presence of individual heirs.

Base Model	Approach	MIR-E
Qwen2.5	RAG-P	0.28
LLaMA	RAG-P	0.33
DeepSeek	RAG-P	0.17
Fanar	RAG-P	0.38
LLaMA	SFT	0.81
Fanar	SFT	0.83
Fanar	RAG-FT	0.74

Table 1: Model performance on the test set.

6. Limitations

This study has several limitations. First, although we consider multiple open-source models, we do not systematically examine the impact of model scale. Larger variants may exhibit different capabilities and achieve better performance. For example, we evaluate only *Qwen2.5-3B-Instruct*, as larger variants were excluded due to computational constraints. Future work should explore larger variants to enable a more comprehensive comparison.

Second, the absence of a pre–fine-tuning baseline limits our ability to disentangle the contribution of fine-tuning from the inherent capabilities of the models. However, results from team PSL (Mouhoub and BOUCHEKIF, 2026), who evaluate general-purpose models under controlled prompting without task-specific adaptation, provide a useful reference for estimating baseline capabilities.

Third, the comparison between fine-tuning and RAG-based approaches should be interpreted with caution, as our findings depend on the specific retrieval strategy and rules used. More advanced retrieval pipelines have achieved stronger results (Swaileh et al., 2026), suggesting that our RAG performance is constrained by retrieval quality and rule design. Moreover, we did not conduct a dedicated retrieval-level evaluation (e.g., precision and recall of retrieved rules), so attributing errors to retrieval noise remains a hypothesis requiring further investigation. A systematic analysis of retrieval quality and its impact on downstream generation will be explored in future work.

Finally, the analysis is limited to aggregate MIR-E scores, which do not fully capture the strengths and weaknesses of the proposed approaches. As future work, we plan to incorporate stage-level MIR-E breakdowns and conduct a detailed error analysis, thereby providing deeper insights for improving inheritance reasoning systems.

7. Conclusion

In this paper, we present our submission to the QIAS 2026 shared task. We evaluate three approaches based on open-source large language models: RAG-P, SFT, and RAG-FT. The best performance is achieved by SFT, which outperforms both RAG-based variants, demonstrating the effectiveness of learning inheritance reasoning patterns directly from data. In future work, we plan to explore training strategies that explicitly model rule identification and application as intermediate steps, enabling more accurate and interpretable reasoning. We also aim to improve adherence to instructions and output structure by investigating techniques such as constrained decoding.

8. Acknowledgments

We would like to thank the QIAS 2026 organisers for providing the datasets and organising this challenge.

9. Bibliographical References

Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, Mus'ab Husaini, Soon-Gyo Jung, Ji Kim Lucas, Walid Magdy, Safa Messaoud, Abubakr Mohamed, Tasnim Mohiuddin, Basel Mousi, Hamdy Mubarak, Ahmad Musleh, Zan Naeem, Mourad Ouzzani, Dorde Popovic, Amin Sadeghi, Husrev Taha Sencar, Mohammed Shinoy, Omar Sinan, Yifan Zhang, Ahmed Ali, Yassine El Kheir, Xiaosong Ma, and Chaoyi Ruan. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Nouar AlDahoul and Yasir Zaki. 2025. Nyuad at qias shared task: Benchmarking the legal reasoning of llms in arabic islamic inheritance cases. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 861–866.

Abdessalam Bouchekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026. MAWARITH: A dataset and benchmark for legal inheritance reasoning with LLMs. *arXiv preprint arXiv:2603.07539*.

Abdessalam Bouchekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025a. 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.

Abdessalam Bouchekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al-Khatib, and Mohammed Ghaly. 2025b. Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 246–257.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Eman Elrefai, Mohamed Lotfy Elrefai, and Aml Hassan Esmail. 2025. Gumball at qias 2025: Arabic llm automated reasoning in islamic inheritance. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 953–959.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mohamed Motasim Hamed, Nada Ghneim, and Riad Sonbol. 2025. Hiast at qias 2025: Retrieval-augmented llms with top-hit web evidence for arabic islamic reasoning qa. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 883–891.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Hala Justanieah and Ghader Kurdi. 2025. An evaluation of chatgpt's responses to islamic inheritance law questions: Insights and challenges. In *Proceedings of the Conference on Artificial Intelligence and Driving Change in Education*, Saudi Arabia.

Ghader Reda Kurdi. 2026. A systematic mapping study: Exploring islamic inheritance in computing research. *International Journal of Informatics and Communication Technology (IJ-ICT)*. Accepted for publication (in press).

Ghader Reda Kurdi and Hala Mohammad Justanieah. 2025. Enhancing evaluation practices for islamic inheritance calculation systems: Toward a standardized benchmark. *Indonesian Journal of Electrical Engineering and Computer Science*, 40(3):1549–1566.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

AL-Smadi Mohammad. 2025. Qu-nlp at qias 2025 shared task: A two-phase llm fine-tuning and retrieval-augmented generation approach for islamic inheritance reasoning. In *Proceedings of*

The Third Arabic Natural Language Processing Conference: Shared Tasks, pages 892–898.

Mohamed Lamine Mouhoub and Chahinez BOUCHEKIF. 2026. Psl at qias 2026: Which models perform better in arabic inheritance reasoning? In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*, Palma, Spain.

Wassim Swaileh, Mohammed-En-Nadhir Zighem, Hichem Telli, Salah Eddine Bekhouche, Abdellah Zakaria Sellam, and Fadi Dornaika. 2026. Cvpd at qias 2026: Rag-guided llm reasoning for al-mawarith share calculation and heir allocation. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*, Palma, Spain.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

10. Appendices

10.1. Examples of Inheritance Rules

Example 1:

```
{
  "id":"share_01",
  "title":
"أصحاب الفروض (الزوجة)",
  "text":
"الزوجة من أصحاب الفروض.",
  "category":"fixed_share",
  "madhhab":"general",
  "heirs":[
"زوجة"
],
  "conditions":[],
  "lang":"ar"
}
```

Example 2:

```
{
  "id": "special_case_solution_03",
  "title":
"حل المسألة الأكدرية",
  "text":
"في هذه المسألة: ١) الزوج يرث النصف عند عدم وجود فرع وارث. ٢) الأم ترث الثلث بشرط عدم الفرع الوارث وعدم تعدد الإخوة وألا تكون المسألة من العمريتين. ٣) عند اجتماع الجد مع الأخت الشقيقة مع وجود صاحب فرض: يُنظر للجد في أفضل الأحوال من (المقاسمة، ثلث الباقي، سدس الكل)، وفي هذه المسألة سُدس الكل أفضل له فيأخذ السدس. ٤) تُضم الأخت الشقيقة أو الأخت لأب إلى الجد فيما بقي ويُقسم بينهما للذكر مثل حظ الأنثيين.",
  "category": "special_case_solution",
  "madhhab": "general",
  "heirs":
[ "زوج", "ام", "جد", "اخت شقيقة", "اخت لأب"],
"required_heirs":
[ "زوج", "ام", "جد"],
"optional_heirs_any":
[ "اخت شقيقة", "اخت لأب"],
  "conditions": {},
  "lang": "ar"
}
```

10.2. Fine-Tuning Configuration and Training Details

All experiments were implemented using the HuggingFace Transformers library and PyTorch, and executed with GPU acceleration. The training configurations for LoRA and model optimisation are summarised in Tables 2 and 3.

Parameter	Value
Task type	CAUSAL_LM
LoRA rank (r)	16
LoRA alpha	32
LoRA dropout	0.05
Bias	none
Target modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj

Table 2: LoRA configuration.

Parameter	Value
Tokeniser	Same as base model tokeniser
Max sequence length	2048
Optimiser	AdamW
Learning rate	2×10^{-4}
Weight decay	0.01
Learning rate scheduler	Linear (no warmup)
Number of epochs	1
Train batch size per device	4
Eval batch size per device	1
Gradient accumulation steps	16
Effective train batch size	64
Selection metric	eval_loss
Greater is better	False
Mixed precision	FP16 when CUDA is available
Gradient checkpointing	Enabled
Use cache	Disabled
Seed	42

Table 3: Training configuration.