

Pipeline Engineering vs Fine-Tuning: A Pre-Registered Evaluation of a Kenyan Legal RAG Assistant

Mike Keror · 17 October 2026 · Code, data and pre-registration: github.com/mikekeror-rgb/advocate_renumeration_assistant_v1

On a fresh 236-question test fixed before any system under test was run, removing the retrieval and calculation components costs the assistant 4.7 to 11.9 points each (28.0 with all three removed), while QLoRA fine-tuning and DPO of Qwen2.5-7B show no detectable benefit.

Question

A common assumption is that fine-tuning a small model is what makes a domain assistant good. This project tests it against a rival explanation: that most of the gain comes from the engineering around a general-purpose model.

The assistant answers questions about Kenya's Advocates Remuneration Order and the court rulings that apply it. Two kinds of change were compared:

- **Pipeline components**, each removed in turn: case-scoped retrieval, full-length context snippets, and a deterministic fee calculator.
- **Fine-tuning** of Qwen2.5-7B-Instruct in two stages: QLoRA supervised fine-tuning (SFT), then on-policy DPO.

An earlier study on 62 questions found each component worth 13 to 26 points, and no benefit from fine-tuning (SFT +1.6 [-3.2, +6.5], DPO 0.0 [-4.8, +4.8]). Those questions were also used while the pipeline was being tuned, so they flatter it. This report repeats the comparison on a new test set that was built after the pipeline was frozen, and registered before any system was scored on it.

System

The corpus is the Advocates Remuneration Order plus a set of court rulings, split into 3,414 chunks held in a Chroma collection. A question goes through five steps:

1. **Hybrid retrieval.** Dense vector search and BM25 keyword search are combined by reciprocal rank fusion.
2. **Case scoping.** If the question names a case, retrieval is restricted to that case, so the answer cannot be drawn from a different ruling.

3. **Context building.** Retrieved chunks are passed to the model as snippets of up to 2,000 characters each.
4. **Fee calculator.** A router recognises fee-calculation questions and computes the figure in code from the Order's schedule, and a guard answers fee questions the router cannot compute (no amount given, or an unsupported fee type) with a fixed message instead of letting the model guess. The model never does the arithmetic.
5. **Generation.** The model writes an answer that cites the chunks it used. The backend is swappable by environment variable; all runs reported here used one vLLM server.

The fine-tuned systems are LoRA adapters served alongside the base model: an SFT adapter trained with QLoRA on training data written by a teacher model (gpt-oss-20b), and a DPO adapter trained on top of it with on-policy samples.

Eight configurations were run on the same questions:

Name	What differs from the full pipeline on the base model
base	Nothing: Qwen2.5-7B-Instruct with the full pipeline
base_rerun	Nothing, run a second time to measure run-to-run noise
ep1	SFT adapter (epoch 1) in place of the base model
dpo	DPO adapter in place of the base model
nocase	Case scoping off
snip1000	Snippets cut from 2,000 to 1,000 characters
nocalc	Fee router and guard off (one switch, so both are disabled together)
stripped	All three of the above off

Evaluation design

The test set has 236 questions in five strata, written after the pipeline was frozen (git tag `pipeline-frozen-v1`).

Stratum	Questions	What it tests
Ruling, unseen cases	69	Questions on 14 rulings never used in training or earlier evaluation

Stratum	Questions	What it tests
Ruling, seen cases	48	Questions on rulings that were used in training or earlier evaluation
Statute	23	Questions on the Advocates Remuneration Order
No-answer	56	Questions the corpus cannot answer; the correct reply is a refusal
Calc	40	Fee calculations with a single correct figure

How it was built. A question-writing model (qwen/qwen3.8-27b) drafted 276 candidates from corpus passages. Each was then reviewed by hand and marked keep, fix or drop. 194 were kept as written, 42 were rewritten and 40 were dropped. Automatic flags marked 75 rows as likely problems (references to "the passage", case names leaking into questions that should not name the case, duplicates, figures not found in the source passage), and the other 201 rows were read by hand too. Automatic flags were used only to decide what to look at first.

Three models, three roles. The training data came from gpt-oss-20b, the questions from qwen/qwen3.8-27b, and the answers are judged by gpt-5.4-mini. The judge is a different model from both the question writer and the training teacher.

Gold answers for calculations were computed by hand. The router's own figures were kept in a separate file and compared only after the hand values were final. All 40 matched.

Freezing. The set's SHA-256 hash is 95ddd72d65bfffeca (first 16 characters). The set, the analysis plan and the hypotheses were committed and tagged in the repository (prereg-v3 for the dataset, prereg-v3.2 for the final pre-registration) before any system under test was run.

Scoring. Each answer is marked correct or incorrect against the reference by the judge model. Calc answers are also checked against the calculator's output, and no-answer questions require a refusal.

Results

The full pipeline on the base model answers 87.7% of the 236 questions correctly: 78.3% on unseen rulings and 79.2% on seen rulings, against 91% to 100% on the other three strata.

System	All	Unseen rulings	Seen rulings	Statute	No-answer	Calc
base	87.7	78.3	79.2	91.3	96.4	100.0

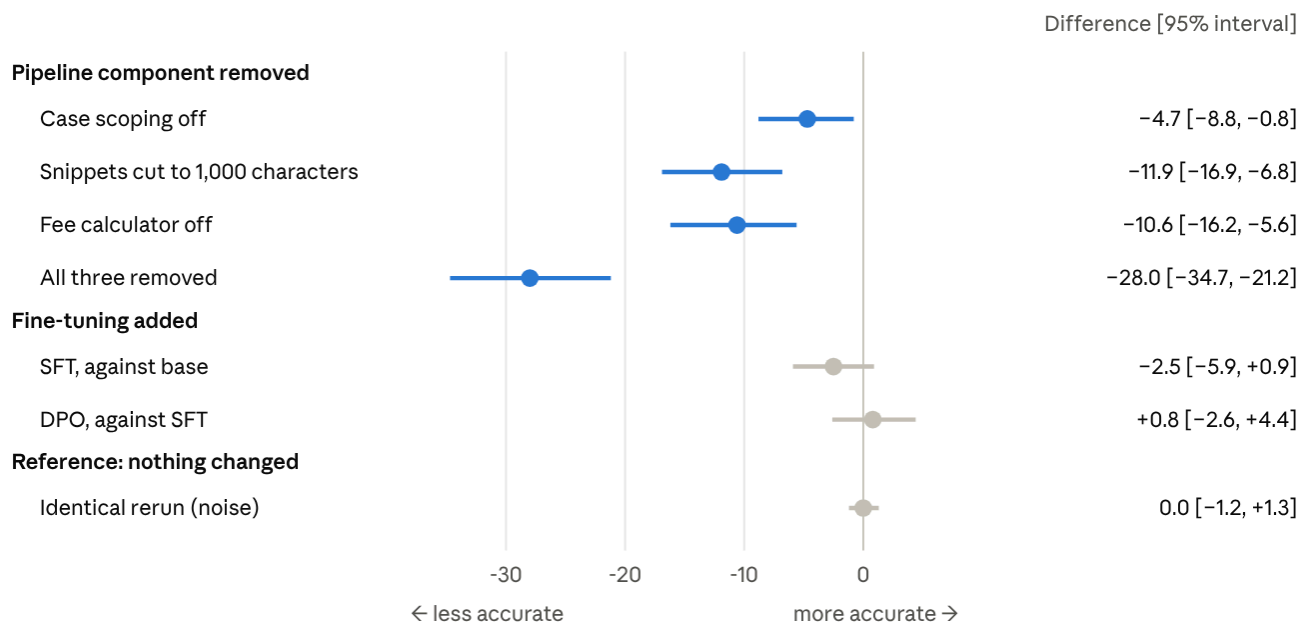
System	All	Unseen rulings	Seen rulings	Statute	No-answer	Calc
base_rerun	87.7	76.8	81.2	91.3	96.4	100.0
ep1 (SFT)	85.2	75.4	72.9	87.0	96.4	100.0
dpo	86.0	72.5	81.2	87.0	96.4	100.0
nocase	83.1	68.1	70.8	91.3	96.4	100.0
snip1000	75.8	52.2	60.4	87.0	96.4	100.0
nocalc	77.1	79.7	85.4	95.7	100.0	20.0
stripped	59.7	42.0	54.2	91.3	100.0	22.5

Accuracy, % of questions correct. Intervals for each cell are in `eval/v3_results.md` in the repository.

Reading the table. Calc, no-answer and statute are at or near the ceiling for most systems, so they cannot separate them. The ruling strata, at about 78%, carry the result. A second run of the identical configuration changed 2 of 236 answers (overall difference +0.0 points, interval [-1.2, +1.3]), so differences of more than a couple of points are not run-to-run noise. The overall column is diluted by the easy strata: removing a ruling-specific component moves rulings by 8 to 26 points but the overall score by only 5 to 12.

Pipeline parts are worth 5 to 28 points; fine-tuning cannot be told from zero

Change in overall accuracy against the system before the change, percentage points, 236 questions



Overall accuracy over all 236 questions; 95% intervals from a cluster bootstrap by case, 10,000 resamples · `eval/v3_results.md`

The pre-registered fine-tuning tests used unseen rulings only, which this overall view mixes with easier strata. Their intervals are given below.

H1: each pipeline component helps (supported)

In every case the difference is the ablation minus the full pipeline, with a 95% interval from a cluster bootstrap by case.

- **Snippets cut to 1,000 characters:** -11.9 points overall [-16.9, -6.8]; -26.1 on unseen rulings [-36.9, -16.2]; -18.8 on seen rulings [-33.3, -6.2].
- **Calculator off:** -10.6 overall [-16.2, -5.6], driven by the calc stratum, which falls from 100% to 20% (-80.0 [-92.5, -67.5]). The other strata do not get worse (unseen rulings +1.4 [-3.1, +6.7]).
- **Case scoping off:** -4.7 overall [-8.8, -0.8]; -10.1 on unseen rulings [-22.1, 0.0]. This is the weakest component, and its interval on unseen rulings touches zero.
- **All three off:** -28.0 overall [-34.7, -21.2], from 87.7% to 59.7%.

The direction matches the earlier 62-question study, where each component was worth 13 to 26 points. The effects here are smaller, which is consistent with the earlier questions having been used while the pipeline was tuned.

H2 and H3: fine-tuning shows no detectable benefit

These are the two confirmatory tests, both measured on unseen rulings.

- **SFT against base:** -2.9 points [-10.1, +4.9]. Overall: -2.5 [-5.9, +0.9].
- **DPO against SFT:** -2.9 points [-12.5, +5.6]. Overall: +0.8 [-2.6, +4.4].

Both intervals include zero and are 15 to 18 points wide, because the 69 unseen questions come from only 14 rulings. The test is able to rule out an SFT gain above about 5 points and a DPO gain above about 6 points on unseen rulings. It cannot rule out small gains or small losses.

What the results support

Supported. Each of the three pipeline components improves accuracy on a test it was not tuned on. The retrieval components (snippet length, case scoping) act on rulings, and the calculator acts on fee calculations. Together they are worth 28 points overall.

Not shown. That fine-tuning is worthless. The result is that no benefit is detectable at this sample size. The SFT point estimates are at or below the base model in every stratum where answers differ, but every interval includes zero.

Limits of this study.

- Unseen rulings come from 14 cases, so the intervals on the fine-tuning comparison are wide.

- The statute stratum has 23 questions and cannot be unseen: the statute text was in the training data. It shows that the questions are held out, not the knowledge.
- Calc, no-answer and statute sit near the ceiling. Some no-answer questions are plainly off-topic (one asks about resting time for a grilled cut of meat), which almost any system refuses.
- Questions were drafted by a model and checked by one person, the author.
- The judge is itself a model. A random sample of 30 judged answers (seed 20261017, drawn from all eight runs) was re-judged by hand and agreed with the judge on all 30. Only 13 of the 30 were scored by the model judge; the other 17 were deterministic checks on fee figures and refusals. The informative result is therefore 13 of 13, which at 95% confidence still allows a disagreement rate of up to about 20%.
- One SFT adapter and one DPO adapter were evaluated. Other training data or settings might behave differently.
- One jurisdiction, one 7B base model, one pipeline.

Exploratory observations

These were not part of the pre-registered plan. They are leads, not findings.

- **The calculator fired on questions it should not have.** In the full-pipeline run it was triggered on four ruling questions (`v3_unseen_named_022` , `v3_seen_named_002` , `v3_seen_named_011` , `v3_seen_named_026`). At least one of them, `v3_seen_named_002` , was answered correctly only when the calculator was off. The router needs a tighter test for when a question is really a fee calculation.
- **Turning the calculator off helped eight non-calc questions and hurt one.** In the identical rerun the comparable counts were one and one. The same switch also disables the guard. Reading the base-run answers, six of the eight were questions that were not fee calculations but received a fixed fee message (five from the guard, one from the router); with both off, the model answered them and was marked correct. The other two flips, and the one question that got worse, are the size of the run-to-run noise. So the fee components misfire on about 2.5% of all questions (6 of 236), and the full pipeline's accuracy is understated by about that much. Tightening the router and guard triggers is the clearest next improvement.
- **The DPO system scores higher on seen rulings (81.2%) than on unseen (72.5%).** This could indicate memorisation, but the intervals overlap widely, the base model shows no such gap (79.2% against 78.3%), and the SFT system shows the opposite one (72.9% against 75.4%). The gap has not been tested.

Reproducibility

Everything is in the [project repository](#).

Item	Value
Frozen pipeline	branch <code>ablation-frozen</code> , tag <code>pipeline-frozen-v1</code>
Eval set	<code>eval/qa_pairs_v3.jsonl</code> , 236 questions, SHA-256 starting <code>95ddd72d65bffecca</code>
Pre-registration	<code>eval/PREREG.md</code> ; tags <code>prereg-v3</code> (dataset) and <code>prereg-v3.2</code> (final text)
Judge	<code>gpt-5.4-mini</code> , unchanged prompt
Run script	<code>run_v3_eval.sh</code> (refuses to run if the set's hash differs)
Raw results	<code>eval/results_ev3_*.csv</code> , one file per configuration
Analysis	<code>python eval/analyze_v3.py --out eval/v3_results.md</code> ; cluster bootstrap by case, 10,000 resamples, seed 20261017

Deviations from the pre-registration, all before any system under test was run.

- Section 4 originally said ablations would run on the best-performing system. It was changed to the base model, matching the earlier study and avoiding a choice made after seeing results.
- A five-question harness check (one question per stratum) was run on `gpt-oss-20b` before the final tag, to confirm the harness reads the new file. Its output was deleted and is not reported.
- The pre-registration mentions a tag `prereg-v3.1` that was never created; `prereg-v3.2` is the final one.

Still to do: the retrieval-hit, faithfulness and latency measurements, which are on the portfolio page, are not yet written up here.