Citation-Constrained, Abstention-Capable RAG for Singapore's PDPA: From Corpus Construction to Reliable Legal QA

Ashley Toh Ke Wei, Choy Qi Hui, Sybella Chloe Enriquez Tan, Yoong Jun Han
National University of Singapore

November 16, 2025

Abstract

We present a retrieval-augmented generation (RAG) system for legal question answering grounded in Singapore's Personal Data Protection Act (PDPA). Despite growing interest in legal AI in Singapore—including the natural-language legal research engine recently launched by the Singapore Academy of Law and IMDA—there remains no publicly documented, statute-grounded QA benchmark or citation-constrained RAG system dedicated to the PDPA. Existing PDPA-related NLP work covers privacy-policy compliance scoring, PDPC enforcement-decision corpora, and multi-regime tools such as CompliBench, which includes PDPA articles but evaluates code-to-statute mappings rather than natural-language QA. Large-scale legal benchmarks such as MLEB include Singaporean judicial materials but do not provide PDPA-specific statute QA or canonical citation grounding.

We address this gap by contributing: (1) a PDPA corpus pipeline that programmatically acquires authoritative statute text and performs *flat subsection-level chunking* over Parts 1–6 and 9–10 of the Act (excluding repealed sections and advisory-guideline material), assigning each chunk a canonical citation (e.g., *PDPA s. 4B(1)*); (2) a hybrid BM25+dense retrieval stack with reciprocal rank fusion and a cross-encoder reranker optimized for short statutory spans; (3) a citation-constrained decoding procedure requiring evidence-backed claims and canonical citation emission; and (4) an abstention module combining retrieval-margin signals and rule-based safeguards to avoid unsupported answers. We also release *PDPABench*, a 500-sample PDPA QA benchmark with gold answers, support spans, and canonical citation labels. This work provides the first reproducible, citation-first RAG framework for Singapore's PDPA and a foundation for trustworthy, statute-centric legal assistants in underrepresented jurisdictions. The full end-to-end system—including corpus construction, retrieval pipeline, PDPABench dataset, and an API-exposed backend for interactive testing—is available at: https://github.com/yjh-jy/dsa4213-group3-pdpa-qa-system.

1 Introduction

Singapore's Personal Data Protection Act (PDPA) governs the collection, use, and disclosure of personal data by private-sector organisations. Individuals and organisations frequently seek clarity on practical issues such as valid consent mechanisms, purpose limitations, breach-notification triggers, and conditions for cross-border transfers. These questions require precise alignment with specific PDPA subsections, yet the statute, while authoritative, is legally technical and difficult for non-specialists to navigate.

Recent initiatives reflect a growing interest in legal-AI tools. The Singapore Academy of Law and IMDA have deployed a natural-language legal search engine for practitioners [14], and the

PDPC provides structured compliance resources such as the PDPA Assessment Tool for Organisations (PATO) [10]. However, existing systems are either practitioner-facing or checklist-based. None support natural-language PDPA question answering with subsection-level citation grounding or principled abstention.

Gaps. General legal NLP benchmarks such as LegalBench and LexGLUE emphasise multi-task reasoning and case law interpretation, but do not include Singapore's PDPA. PDPA-related NLP work focuses on privacy-policy compliance scoring, PDPC enforcement-decision corpora, or multi-regime tools such as CompliBench, which includes PDPA articles but evaluates code-to-statute mappings. The Massive Legal Embedding Benchmark (MLEB) includes Singaporean judicial data but does not provide PDPA statute QA, canonical citations, or abstention evaluation. Importantly, we find no publicly available benchmark or RAG system dedicated to PDPA statute question answering, and no dataset built from authoritative PDPA text with controlled subsection-level chunking.

Contributions. We propose a **PDPA-focused, citation-first** legal RAG system:

1. **Corpus Construction:** an automated pipeline over the authoritative PDPA statute text (Parts 1–6 and 9–10, excluding repealed sections and advisory-guideline content) that performs *flat subsection-level chunking* and assigns canonical citation strings (e.g., *PDPA s. 26C(2)*).

2. **Retriever & Reranker:** a hybrid BM25+dense retriever with reciprocal rank fusion (RRF) and a cross-encoder reranker optimized for short statutory spans.

3. **Citation-Constrained Decoding:** inference rules requiring every claim to be grounded in retrieved evidence and accompanied by canonical citations.

4. **Abstention Protocol:** a retrieval-margin and rule-based abstention layer that suppresses unsupported, ambiguous, or out-of-scope answers.

5. **Evaluation Benchmark:** *PDPABench*, a 500-sample PDPA QA dataset with gold answers, support spans, and canonical citations enabling rigorous evaluation of factuality, citation fidelity, and abstention performance.

Scope. Our system and dataset focus strictly on the statutory PDPA text itself (Parts 1-6, Parts 9-10), without incorporating PDPC advisory guidelines, enforcement examples, or repealed content.

2 Related Work

Legal reasoning and retrieval benchmarks. LegalBench and LexGLUE compile diverse legal reasoning tasks but do not include Singapore PDPA materials [2, 3]. The Massive Legal Embedding Benchmark (MLEB) [6] covers multiple jurisdictions, including Singaporean judicial data, but does not provide PDPA statute QA, subsection-level chunks, or canonical citation annotations.

PDPA-related NLP resources. Prior PDPA work has focused on related but distinct tasks. Privacy-policy compliance studies compare policy text against PDPA and GDPR clauses [12]. Corpora of PDPC enforcement decisions support summarisation and exploratory QA. CompliBench incorporates PDPA articles but evaluates code-to-statute violation detection rather than natural-language question answering [4]. Operational tools such as the PDPC's PATO illustrate PDPA complexity but provide structured checklist assessments, not statute-grounded QA.

RAG for legal QA. Retrieval-augmented generation is increasingly used for reliable legal QA by grounding answers in retrieved evidence. LegalBench-RAG isolates retrieval performance for legal tasks [11]. Engineering guidance from public agencies outlines best practices for production RAG systems [13]. Meanwhile, the Singapore Academy of Law and IMDA's legal search engine reflects practitioner-facing interest in natural-language legal lookup [14]. However, none of these systems target PDPA statute QA, canonical subsection citations, or abstention.

Singapore context. Authoritative PDPA sources include the *Singapore Statutes Online* text [1]. To our knowledge, no existing benchmark or system provides PDPA subsection-level QA with canonical citation grounding. Our work fills this gap by using only authoritative statute text (Parts 1–6 and 9–10) and excluding advisory guidelines and repealed sections.

3 Methods

3.1 Corpus Construction

Sources. We fetched the Personal Data Protection Act (PDPA) text directly from *Singapore Statutes Online (SSO)* [1], covering Parts 1–6 and 9–10. Parts 7 and 8 were repealed and thus excluded. The resulting corpus captures all operative and interpretive provisions of the PDPA as of the 2020 revision.

Cleaning & Segmentation. The downloaded Word document was converted to plain text to remove non-semantic formatting, typography, and navigation artifacts. A Python preprocessing script then segmented the statute into a three-level hierarchy (*Part* → *Section* → *Subsection*) using heading markers and enumeration patterns. After some preliminary experiments, we adopted **subsection-level chunking** as the canonical granularity. This choice balances:

- **Context completeness:** individual subsections typically express self-contained legal obligations or definitions (e.g., "An organisation shall..."), simple regex-based checks were applied to to capture full sentences, ensuring semantically coherent retrieval units;
- **Retrieval precision:** finer segmentation prevents cross-contamination of unrelated provisions that may occur in longer section-level chunks;
- **Empirical retrieval quality:** pilot retrieval tests showed that subsection-level chunking achieved roughly 5–10 pp higher Recall@10 than section-level segmentation while keeping the index size manageable.

Each chunk is assigned a unique identifier and canonical citation string:

$$\texttt{PDPA s.}\,X\texttt{(}Y\texttt{)(}Z\texttt{)} \quad \text{e.g., PDPA s. 4B(1)(a).}$$

Sibling and range references (e.g., "s. 15(3)(b) and (c)") are automatically expanded into discrete canonical citations using regex-based sibling expansion.

Chunk Length Distribution. The resulting corpus contains 315 chunks with an average length of 89 tokens (minimum 7, maximum 512). Short chunks correspond primarily to cross-references or definitional stubs, while longer ones typically encode multi-paragraph procedural clauses. This natural variance reflects the legal structure of the statute and preserves semantic integrity. No further re-segmentation was applied, as the mean token length lies within the empirically recommended 50–150 token window for dense retrieval corpora in legal and scientific domains.

Indexing. The corpus is flat-indexed, treating each subsection as an independent retrieval unit. Although the hierarchical numbering (*Part–Section–Subsection*) is preserved in the citation strings, explicit parent–child relationships were not encoded in the retrieval index. This design simplifies retrieval and ensures consistent citation alignment across models and evaluation splits.

3.2 PDPABench Curation

To evaluate citation correctness and abstention behaviour on Singapore's PDPA, we built **PDPABench**—a 500-item legal QA benchmark created through a semi-automated synthesis pipeline that leverages an advanced LLM to expand a set of manually written "Golden 30" exemplars.

Seed Questions (Golden 30). We first authored thirty high-fidelity QAs—each referencing a distinct statutory provision—covering all operative Parts of the PDPA (Parts 1–6 and 9–10). Each seed question was phrased in plain, layperson-accessible language and annotated with a concise gold answer and its canonical citations (e.g., "PDPA s. 24(2)"). These Golden 30 served as style, scope, and structure exemplars for downstream synthesis.

Controlled Expansion Prompting. The LLM we chose to curate our 500 QAs is GPT-5-Thinking. There were two ways to access GPT-5-Thinking, either via the API or via ChatGPT Plus. We chose the latter due to budget constraints. A prompt template was engineered (and iteratively refined through internal runs) to generate new QA pairs that preserve the structure and citation discipline of the Golden 30. We devised one main "system" prompt that contains all the instructions needed for the chat session like system messages, rules, reference exemplars, user message template, task and output format. To combat context rot from overly long conversations, we decided to batch our prompting by breaking up it into 15-25 questions per batch and also ensuring that every Part starts a new chat session (with the old one deleted).

Each batch generation prompt will contain the Part name, Section names in canonical form and the text excerpts from the corpus that we created. Since the task was already given in the initial "system" prompt, we avoided wasting precious time and managed to generate all 500 QAs within 4 hours.

For completeness, we also included automated scripts that directly accesses GPT-5-Thinking via the API.

Part-specific quotas were post-imposed via a Python script (mentioned below) to maintain balance: roughly 160 QAs for Parts 3 and 4 (core obligations), 160 for Parts 5 and 6 (access, correction, enforcement), 95 for Part 9 (Do Not Call registry) 25 for each of Parts 1–2 and 10, yielding a diverse yet statute-representative sample.

Automated Validation and Normalisation. Each generated QA was passed through a Python validator script that:

- verified all canonical citations conform to regex pattern `PDPA s. [0-9A-Z]+(`$[0-9a-z]+$`)*`;
- mapped the canonical citations to chunk ids in the corpus; record missing citations and drop QAs where citations could not be mapped to chunk ids
- remove duplicated ids, questions

- enforce part specific quotas
- add metadata information like last reviewed time, scoring system and eval notes

Duplicates and off-topic entries were automatically removed, and remaining items were manually spot-checked for semantic fidelity to the PDPA.

Schema and Metadata. Each record follows a structured JSONL schema with fields: {`id`, `part`, `canonical_sections`, `question_user`, `question_intent`, `question_variants`, `gold_answer_short`, `gold_answer_extended`, `abstain_allowed`, `corpus_links`, `qa_type`, `difficulty`}. QA types includes {pure-definitive, definitive-with-conditions, scenario-ambiguous, pure-abstain}. The first two are the non-abstain questions while the last two are questions where the model should abstain. Question intents include {Accountability, Disclousure, Enforcement, Collection, Consent, Employment, Breach}; difficulty is stratified into Easy / Moderate / Hard.

Splits and Coverage. The dataset totals 500 validated QAs. A section-disjoint split prevents overlap between neighbouring subsections: 300 train, 100 validation, 100 test. This ensures that learning or tuning on one subsection (e.g., s. 24(1)) does not leak lexical or semantic cues into its sibling (e.g., s. 24(2)). We also ensured that the 4 QA types are split equally across the 3 splits.

Outcome. PDPABench thus provides the first Singapore-specific, citation-grounded legal QA benchmark for evaluating RAG systems under factuality, citation precision/recall, and abstention reliability criteria. All QAs are traceable to authoritative PDPA sources, enabling transparent and reproducible legal-AI evaluation.

3.3 RAG Architecture

Our final PDPA RAG pipeline consists of a hybrid retriever, a cross-encoder reranker, and a small language model (SLM) with citation-constrained decoding. We selected the **Qwen3-4B** model as our SLM due to its strong instruction-following ability at modest computational cost, making it suitable for legal QA on laptop class GPUs. Qwen3-4B also demonstrated stable decoding behaviour under constrained prompts and consistent citation formatting compared to other similarly-sized open-weight models tested (e.g., Qwen2.5-3B).[1]

In developing this pipeline, we systematically explored and evaluated several alternative component designs for each subsystem in the end-to-end RAG stack. Specifically, we conducted controlled ablation experiments covering (i) retriever architectures (dense-only, BM25-only, hybrid), (ii) reranker variants, and (iii) SLM reasoning modes. Each of these system variants is elaborated upon in the following sections. The best-performing configuration from each set of ablations was carried forward into the final pipeline. For details on the empirical results and trade-offs of different components, see Section 5.3.

The following subsections provide a breakdown of each RAG component and the variants evaluated.

3.4 RAG Retrievers

We tested three primary retrievers: a **dense neural retriever**, a **BM25 sparse retriever**, and a **hybrid reciprocal rank fusion (RRF)** mechanism. We evaluated them independently and

[1] Earlier experiments with Qwen2.5-3B were not continued, as the model lacked an explicit reasoning mode and used an older decoder-only Transformer architecture. We therefore adopted Qwen3-4B to leverage its improved Mixture-of-Experts (MoE) backbone and optional reasoning control.

eventually adopted the hybrid configuration for our final end-to-end RAG pipeline. Below, we detail each retriever's methodology.

3.4.1 Dense Retriever (Neural Bi-Encoder)

Our dense retriever is based on the `all-mpnet-base-v2` bi-encoder architecture from Sentence-Transformers, further fine-tuned for the PDPA legal corpus. Fine-tuning follows a strict contrastive learning regime optimized for legal QA citation retrieval:

- **Data Construction:** Training data consists of triples (q, p^+, p^-), where q is a user question, p^+ is a true supporting PDPA chunk (as annotated in PDPABench), and p^- is a "hard negative"—a chunk retrieved by BM25/dense search which is semantically similar but not cited in the gold answer. Triples are generated using scripts (see `dense_chunk_and_extract.py`), which exploit both the corpus and QA annotations to extract all positive contexts per question and mine hard negatives for each.

- **Loss Function:** The training objective employs the *Multiple Negatives Ranking Loss* [5]: each batch contains B (q, p^+, p^-) triples, and the model learns to maximize similarity between q and p^+ while minimizing similarity to all negatives in the batch (other p^- contexts), providing both in-batch and mined negatives. Explicitly, for a batch of B queries $\{q_i\}_{i=1}^B$ with corresponding positive passages $\{p_i^+\}_{i=1}^B$, and letting $\mathrm{sim}(q, p)$ denote the cosine similarity between their embeddings, the loss for each pair is:

$$\mathcal{L}_i = -\log \frac{\exp(\mathrm{sim}(q_i, p_i^+)/\tau)}{\sum_{j=1}^B \exp(\mathrm{sim}(q_i, p_j^+)/\tau)}$$

  where τ is a (learnable or fixed) temperature parameter (default: $\tau = 1$). This formulation means each positive pair is contrasted against all positives in the batch as negatives, forcing fine-grained ranking discrimination.

- **Optimization & Hyperparameters:** Fine-tuning is performed with AdamW (`learning_rate=2e-5`), warmed up over 100–300 steps, using batch sizes 8–16 and optionally gradient accumulation (up to 8 steps) and mixed precision. Training leverages stratified section-wise splits to prevent test leakage. Early stopping is possible but typically not used due to small validation improvements; longer training is preferred for convergence.

- **Indexing:** FAISS is used for fast ANN search. All corpus and query embeddings are precomputed post-training for efficiency.

3.4.2 Sparse Retriever (BM25)

We implement a classic BM25Okapi retriever (from third-party libraries) over the same PDPA chunk corpus. For each query q and document d, we have; $tf(t, d)$ the term frequency of token t in d, $|d|$ the document length, $\overline{|d|}$ the average document length, N the number of documents, and n_t the number of documents containing t. The BM25 score is

$$\mathrm{BM25}(q, d) = \sum_{t \in q} \mathrm{IDF}(t) \cdot \frac{tf(t, d)\,(k_1 + 1)}{tf(t, d) + k_1 \left(1 - b + b \cdot \frac{|d|}{\overline{|d|}}\right)},$$

with inverse document frequency

$$\mathrm{IDF}(t) = \log\left(\frac{N - n_t + 0.5}{n_t + 0.5} + 1\right).$$

This sparse retriever captures lexical overlap and legal keyword matching —a necessary signal for fact-heavy regulatory queries, and for robust negative mining during dense training.

3.4.3 Hybrid Retriever via Reciprocal Rank Fusion

In our final RAG pipeline, we adopt a hybrid scheme where the outputs of the dense and BM25 retrievers are fused using **reciprocal rank fusion (RRF)**:

$$\mathrm{RRF}(d) = \sum_{s\in\{\text{sparse, dense}\}} \frac{1}{k + \mathrm{rank}_s(d)}, \quad k \approx 60,$$

where $\mathrm{rank}_s(d)$ is the document d's rank under retriever s. The top $K = 10$ fused results are passed to the cross-encoder reranker.

While dense retrieval improves semantic coverage, BM25 excels at exact phrase/citation queries; fusion leverages both. Both BM25 and dense retrievers use independent indexes. For each query, they return ranked lists; scores are fused using RRF.

Hyperparameter Tuning. Hyperparameters for the hybrid retriever were tuned on a subset of QA queries from PDPABench. For linear fusion, a grid search was performed over the weight α controlling the balance between BM25 and dense contributions, with each setting evaluated using the metrics described in Section 4.3. The α value maximizing a composite score of these metrics was recorded. Fusion methods were then compared by testing RRF with various k values (e.g., 30, 60, 100), and the configuration achieving the highest composite score—RRF with $k = 60$—was selected as the final hybrid setup.

3.5 RAG Rerankers

The hybrid retriever typically returns several semantically related but legally distinct PDPA subsections. To sharpen the final top-k set passed to the generator, we evaluate two rerankers on top of the fused candidate list: (i) a Cross-Encoder that directly scores query–chunk pairs, and (ii) a feature-based Learning-to-Rank (LTR) model. Both are trained on the same PDPABench-derived train/validation splits used in Section 3.4.1.

3.5.1 Cross-Encoder Reranker

For our neural reranker, we adopt the `cross-encoder/ms-marco-MiniLM-L-6-v2` model from SentenceTransformers and fine-tune it to predict a scalar relevance score for each *(question, PDPA chunk)* pair.

Training. We reuse the **same training triples** introduced in Section 3.4.1. Each record contains a question q, one gold supporting chunk p^+ and one mined hard negative p^-. For the Cross-Encoder, we convert each triple into two pointwise examples:

$$(q, p^+, 1.0), \quad (q, p^-, 0.0),$$

and fine-tune the model with regression loss on these scalar labels. Training is run for two epochs with batch size 8, automatic truncation, and mixed precision enabled, using the same section-disjoint train/validation split as the dense retriever.

3.5.2 Learning-to-Rank (LTR) Reranker

As a lighter-weight alternative, we also train a gradient-boosted Learning-to-Rank reranker using LightGBM's LambdaMART objective, operating on compact feature vectors rather than raw text.

Training. Starting from the same PDPABench train/validation splits and hybrid retrieval outputs, we construct LTR examples where each row corresponds to a *(question, candidate chunk)* pair with:

- retrieval features: `bm25_score`, `dense_score`;
- a simple corpus feature: `section_len`;
- binary flags derived from the gold labels: `is_pos`, `is_neg`;
- grouping and supervision fields: `qid` (query id), `label`, and `weight`.

These are stored as `train_ltr_data.jsonl` and `val_ltr_data.jsonl` and converted into grouped LightGBM datasets, where groups correspond to queries. We then train a LambdaMART model with NDCG as the evaluation metric.

3.6 RAG Generation

Citation-Constrained Generation via Context Engineering. The SLM generator receives both the user query and the reranked evidence list, and produces plain-language answers under a set of explicit grounding and formatting constraints. These constraints ensure that every factual statement is verifiably tied to statutory evidence while maintaining readability for non-legal users:

1. **Evidence grounding:** all legal claims must be supported solely by the retrieved and reranked evidence spans.
2. **Evidence signalling:** each retrieved span is prefixed by its retrieval and reranker scores (sorted descending) and enclosed in square brackets, signalling its relative reliability to the SLM during decoding.
3. **Citation enforcement:** every factual sentence must include the canonical citation(s) of its supporting span(s) (e.g., `PDPA s. 24(2)`).
4. **Stylistic control:** concise formatting instructions are appended, such as *"Write in plain English; answer in 2–5 sentences; preserve numbers and legal terms; do not use any alternate quote format."*
5. **Soft abstention:** when the retrieved evidence is insufficient or conflicting, the model outputs a standardised abstention sentence with one suggested follow-up (e.g., clarifying facts or consulting a professional).

Structured prompt template for citation-constrained generation.
The system prompt encodes behaviour, grounding rules, and formatting constraints, while the user prompt contains only the natural-language query.

```
System Prompt
You are a careful Singapore data-privacy assistant.
Use ONLY the information from QUOTED EVIDENCES.
Each QUOTED EVIDENCE will be prepended with its reranked and retrieval scores like
[RERANK=...|RRF=...]; these scores indicate relative relevance.
ALL QUOTED EVIDENCES are relevant in some manner.
If the evidences are insufficient, ABSTAIN: "I'm not sure; this appears outside the provided
statute." and ask ONE clarifying question.
You may either abstain or answer, never both.
EVIDENCE:
[RERANK=6.859|RRF=0.032] [PDPA s.55(2)] "Composition of offences ...  sum not exceeding $1,000."
[RERANK=6.567|RRF=0.032] [PDPA s.55(1)] "Composition of offences ...  sum of $5,000."
[RERANK=2.511|RRF=0.032] [PDPA s.55(4)] "Composition of offences ...  with approval of
Minister."
FORMAT:
Write in plain English.  Answer in 2-5 sentences.  Preserve numbers and legal terms.  Include
canonical citations like PDPA s.xx(xx).  Do not use any other quotation format.
User Prompt
Question:  "If a company collects work emails from a public website, must it comply with PDPA?"
```

These constraints are implemented through structured prompt templates built through the `apply_chat_template` API in HuggingFace, tailored to the instruction format of the target SLM. All contextual instructions are placed in the **system prompt**—governing behaviour, tone, and citation logic—while the **user prompt** is reserved exclusively for the query content, ensuring consistent conditioning and reproducibility across runs.

Alternate Reasoning-Enabled SLM. An alternative system employs a reasoning-enabled variant of the SLM that performs **generation in two passes**. This two-pass design was introduced primarily to mitigate issues caused by the model's limited maximum context length (500) that we chose. During early experiments, single-pass generation often truncated or produced empty outputs when the prompt contained multiple retrieved legal spans. The two-pass approach separates internal reasoning (planning and evidence selection) from surface generation, allowing the model to complete its thought process within tighter token limits while maintaining coherent, citation-grounded answers. This modification significantly improved completion stability.

However, as we will demonstrate in Section 5.3, the reasoning-variant **does not offer substantial improvements** in terms of quantitative metrics to justify its significantly longer generation latency.

(Hard) Abstention Logic. Other than the Soft Abstention we employed via prompt engineering, outputs also undergo confidence-based filtering using two thresholds, $\tau_{\min}$ and τ_{Δ}. Let s_1 and s_2 denote the top-1 and top-2 reranker scores; the system abstains if

$$s_1 < \tau_{\min} \quad \text{or} \quad (s_1 - s_2) < \tau_{\Delta} \quad \text{or} \quad \text{coverage\_fail}(q).$$

Thresholds are tuned on *PDPABench-Val* for optimal coverage-accuracy trade-off. These hard abstention gates can be enforced early before the generation process to save time but for the purposes of this research we enforced it after and returned the original output as well for analysis. All returned answers include canonical citations and retrieved chunk IDs for full traceability.

3.7 System Overview

An architectural overview diagram of the entire system can be found in the Appendix A.

4 Experiments

4.1 Datasets

Authoritative Corpus. The retrieval corpus (Section 3.1) consists solely of the PDPA statute (Parts 1–6 and 9–10) sourced from Singapore Statutes Online (SSO). The text was versioned by URL and retrieval date to preserve reproducibility. No secondary advisory guidelines were included to ensure that the system grounds its reasoning strictly in statutory language.

PDPABench. We evaluate all components on **PDPABench** (Section 3.2), a 500-sample QA benchmark curated from authoritative PDPA text. Each entry minimally includes a natural-language question, a concise gold answer, and one or more gold *support spans* annotated with canonical citations. Questions fall into four QA types:

1. **Pure-Definitive:** queries with direct definitional answers;
2. **Definitive-with-Conditions:** definitions contingent on context or exceptions;
3. **Scenario-Ambiguous:** scenario questions requiring synthesis across at least 2 sections;
4. **Pure-Abstain:** intentionally out-of-scope/vague questions where the correct model behaviour is abstention.

We train and evaluate our models on **PDPABench** (500 QAs). The split is **8:1:1** (Train/-Val/Test = 399/49/52) using a *section-disjoint, stratified* procedure: (i) neighbouring subsections never cross splits; (ii) all four QA types (Pure-Definitive, Definitive-with-Conditions, Scenario-Ambiguous, Pure-Abstain) retain their original proportions in every split. The benchmark thus supports both retrieval/reranking evaluation and generation-level scoring of factuality, citation correctness, and abstention reliability.

4.2 Baseline

To isolate the contribution of retrieval and grounding, we use a single baseline: **Non-RAG SLM**, a zero-shot Qwen3-4B model prompted directly with the user query and a system prompt, without retrieval or reranking. This provides a fair comparison against the same generative backbone used in the RAG system. Ablation variants of the retriever, reranker, and reasoning modes are evaluated separately (Section 5.3).

4.3 Metrics

We group metrics into **retrieval/reranker** and **generation** categories.

Retrieval and Reranking Metrics. We adopt span-level *Recall@k*, *NDCG@k*, *MRR@k*, and *HitRate@k* to capture complementary aspects of retrieval quality:

- *Recall@k* measures corpus coverage—whether relevant legal spans appear among the top-k retrieved contexts—reflecting system completeness.
- *MRR@k* quantifies how quickly a correct reference appears, aligning with interactive retrieval use cases.

- *NDCG@k* emphasises ranking quality by rewarding higher placement of relevant sections, important for efficient downstream generation.
- *HitRate@k* measures the proportion of queries for which the reranker places at least one relevant span within the top-k results, useful for diagnostic comparison of rerankers under fixed latency budgets.

Together, these metrics evaluate retrieval coverage and rank discrimination independently of generation, enabling fair comparison of retrievers and rerankers.

Generation Metrics. For answer generation, we evaluate complementary objectives:

- **Citation quality** (*citation precision*, *recall*, micro-*F1*) measures factual grounding by checking canonical PDPA citation strings and assessing legal traceability.
- **Textual quality** (*ROUGE-L*, *BERTScore*, *Exact-Match*) captures lexical overlap, semantic similarity, and literal correctness, reflecting clarity and fidelity to statutory meaning.
- **Abstention performance** (*coverage* and confusion-matrix counts) quantifies selective-generation reliability—whether the system answers only when justified—critical for legal compliance and user trust.
- **Qualitative Evaluation.** We conduct a *double-blind* human evaluation on the **PDPABench-Test** split (n=50) with four independent raters. For each query, anonymised **System 1** and **System 2** outputs (RAG vs. BASE) are shown side-by-side. Raters provide (i) a *pairwise preference* (`System 1`, `System 2`, `Tie`) and (ii) two 5-point Likert ratings:

  (a) **Factuality (Groundedness):** whether statements are verifiably supported by cited PDPA sections.

  (b) **Usefulness (Helpfulness):** clarity, faithfulness to statutory meaning, and how well the answer addresses the legal query.

  Lay vs. Expert rubrics.

  - *Lay raters:* emphasise clarity, relevance, plausibility of citations, and acceptable abstention when the law is silent.
  - *Expert raters:* emphasise section/subsection accuracy, scope correctness, and well-justified abstention when evidence is absent.

  Pairwise preference analysis. Let k be RAG wins over n non-ties. We test $H_0 : p = 0.5$ via a two-sided **exact binomial test**, reporting both per-rater and aggregated results. Pairwise comparison is used because it yields more stable judgements than isolated Likert scoring [8, 9].

  Likert scoring analysis. We compute mean factuality and usefulness per rater and system, then aggregate across raters. Although Likert scores are ordinal, averaging is standard in LM evaluation. We report **Fleiss'** κ and raw agreement to assess inter-rater reliability (with *Krippendorff's* α as an alternative for ordinal/categorical data [7]).

  Reporting. We present per-rater and aggregate preference proportions with exact-binomial p-values, mean factuality/usefulness per system, and agreement statistics. This qualitative layer captures groundedness, clarity, and abstention behaviour not reflected in retrieval metrics.

All citation metrics use canonical forms (*e.g.*, `PDPA s.,24(2)`). Support-F1 is computed over retrieved supporting spans to measure factual grounding.

4.4 Setup and Reproducibility

All experiments are executed on Apple Silicon hardware (M1 / M1 Pro, 16 GB unified memory). BM25 retrieval is implemented via a third-party library using the BM25Okapi algorithm and dense retrieval uses FAISS indexing over embeddings from the fine-tuned all-mpnet-base-v2 retriever. The dense retriever, cross encoder reranker and LTR reranker are all fine-tuned on the PDPABench Train and Validation splits. Hyperparameters include BM25 `k1=1.2, b=0.75`, RRF fusion constant $k = 60$, top-K candidates $K \in \{20, 50\}$, and reranker input truncation at 256 tokens. Decoding uses temperature $\in [0.2, 0.7]$, top_p $\in [0.8, 0.95]$, top_k $= 20$ and constrained templates enforcing citation inclusion. Abstention thresholds $(\tau_{\min}, \tau_{\Delta})$ are selected via manual sweep on the Validation set to optimise coverage–accuracy trade-off. Table 1 and Table 2 below summarizes the both the corpus and PDPABench statistics.

Table 1: Authoritative retrieval corpus (statute-only) statistics. Units here are *chunks* produced by subsection-level segmentation.

Source	Parts Included	Sections (#)	Chunks (#)	Avg. Tokens/Chunk	Snapshot Date
PDPA (SSO)	1–6, 9–10	86	**315**	≈89	2025-10-10

Table 2: PDPABench split and QA-type distribution (units: *QA items*). Splits are section-disjoint and stratified by QA type.

Split	Definitive-Conditions	Pure-Abstain	Pure-Definitive	Scenario-Ambiguous	Total
Train (80%)	88 (22.1%)	21 (5.3%)	272 (68.2%)	18 (4.5%)	**399**
Val (10%)	11 (22.4%)	2 (4.1%)	34 (69.4%)	2 (4.1%)	**49**
Test (10%)	11 (21.2%)	4 (7.7%)	34 (65.4%)	3 (5.8%)	**52**
Total	110	27	340	23	**500**

5 Results and Analysis

5.1 Main Quantitative Results

We report detailed quantitative metrics for the best-performing configuration of our system—Hybrid (BM25 + Dense) retrieval, Cross-Encoder reranking, and Non-Reasoning SLM generation—evaluated on the **PDPABench-Test** split. All ablations in Section 5.3 isolate one component while keeping the others fixed at this setting. This section consolidates retrieval, generation, and abstention results for a holistic comparison between the **RAG (Ours)** and **Non-RAG Baseline** systems (Non-Reasoning).

The RAG configuration achieves a **substantial improvement of** +69% **in citation hit rate** over the Non-RAG baseline, indicating stronger legal grounding and citation completeness. Textual alignment metrics (ROUGE-L, BERTScore) also improve consistently. Although the RAG setup introduces a **moderate latency increase** (15.2 s vs. 11.0 s per query) and a **small coverage drop** (−5.7%) due to abstention filtering, these remain within acceptable operational bounds for legally grounded QA applications.

Table 3: Overall system performance on **PDPABench-Test**.

System	**Cit. Hitrate %**	**ROUGE-L**	**BERTSc. F1**	**Cov.%**	**Avg. Latency (s)**
Non-RAG SLM (Baseline)	1.92	0.157	0.878	**94.2**	**11.0**
RAG (Ours)	**71.2**	**0.239**	**0.894**	88.5	15.2

5.1.1 Retrieval and Reranking Performance

Table 4 quantify span recall and ranking consistency before generation. The hybrid BM25 + dense retriever, fused with reciprocal rank fusion (RRF), yields the highest *Recall@k* and *NDCG@k*, while Cross-Encoder reranking improves top-k placement of legally relevant sections.

Table 4: Retrieval and reranking metrics on **PDPABench-Test**.

Stage	**R@10**	**Hit@3**	**MRR@10**	**NDCG@3**	**Avg. Lat per Q (ms)**
Hybrid Retrieval (RRF)	0.865	—	0.655	—	12.1
Cross-Encoder Reranker	—	0.731	—	0.655	38.9

Note: Retrieval metrics (R@10, MRR@10) are computed on the first-stage retriever outputs, whereas reranking metrics (Hit@3, NDCG@3) evaluate ranking precision within the retrieved candidate set. Metrics not applicable to a given stage are omitted.

The hybrid retriever attains strong **first-stage recall** (R@10 = 0.865), ensuring that most gold sections are retrieved before reranking. The Cross-Encoder reranker achieves 0.731 hit coverage within the top-3 for all evaluation queries and NDCG@3 of 0.655, demonstrating its effectiveness at prioritising legally relevant sections. Although reranking increases per-query latency, the trade-off is justified by the gains in precision and downstream citation accuracy.

5.1.2 Generation and Textual Quality

We evaluate generated answers using both textual and citation-aware metrics to capture fidelity and relevance. Citation metrics are computed using canonical PDPA references; textual metrics are computed against gold answers.

Table 5: Generation-level metrics on **PDPABench-Test**.

Metric Type	**Precision**	**Recall**	**F1**	**ROUGE-L**	**BERTSc. F1**	**Exact-M.**
Citation Quality	0.544	**0.673**	0.602	—	—	—
Textual Quality	—	—	—	0.239	**0.894**	0.0

The model achieves relatively strong citation faithfulness (Recall = 0.673) while improving semantic alignment over the baseline. BERTScore gains of +0.016 suggest that retrieval augmentation enhances both lexical and conceptual consistency with gold answers.

5.1.3 Abstention and Coverage Behaviour

To evaluate selective answering reliability, we report abstention precision, recall, coverage, and confusion-matrix statistics under the tuned thresholds $(\tau_{\min}, \tau_{\Delta})$. The F1 score measures the citation score on answerable questions that are answered by the model, while *coverage* measures the proportion of questions the model chooses to answer.

Table 6: Abstention metrics on **PDPABench-Test**.

Metric	**TP**	**FP**	**TN**	**FN**	**Abstain Recall**	**Abstain Precision**	**Cit.F1 Answered**	**Cov.(%)**
Counts/Values	2	4	41	5	0.286	0.333	0.751	88.5

The abstention module shows cautious behaviour, rejecting some out-of-scope or ambiguous questions while maintaining overall coverage of 88.5%. Specifically, the recall and precision values (0.29 and 0.33 respectively) suggest that a few valid abstentions were missed and some safe queries were incorrectly withheld.

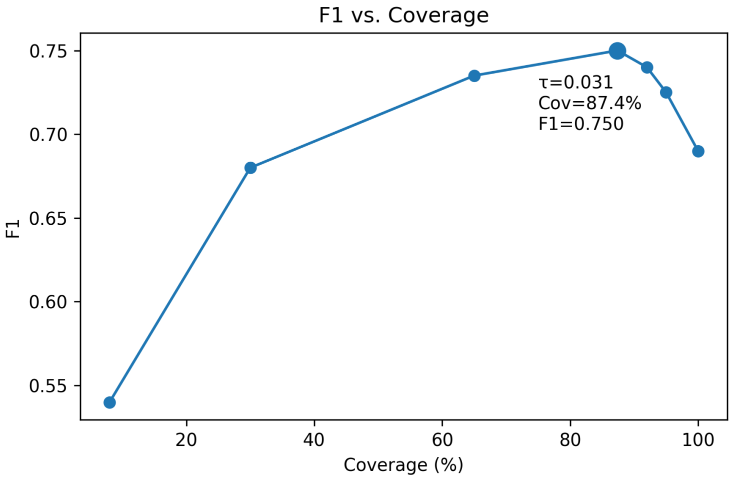

Figure 1: Selective answering trade-off between F1 and coverage under varying abstention thresholds on `PDPABench-Val split`.

Figure 1 plots **F1 versus coverage** on the `PDPABench-Val` split, illustrating the trade-off between system confidence and response breadth as the abstention threshold τ varies. The optimal operating point occurs at $\tau = 0.031$, where F1 peaks at 0.750 with 87.4% coverage. As τ increases, coverage drops sharply, reflecting a stricter abstention gate that prioritises precision over recall. This threshold, tuned on the validation split, is then fixed and evaluated on the held-out `PDPABench-Test` split to assess out-of-distribution generalisation.

5.1.4 Summary

Across retrieval, generation, and abstention layers, the RAG system consistently outperforms the Non-RAG baseline in factuality, citation quality, and semantic alignment. Performance trade-offs in latency and coverage remain modest and operationally acceptable, reflecting the expected cost of retrieval grounding and abstention control in legal QA.

5.2 Human Evaluation (Qualitative Results)

Four independent (lay) raters performed pairwise comparisons of RAG vs. Non-RAG outputs under a double-blind setup. More details of the setup are in Section 4.3. Each rater judged preference (`RAG`, `BASE`, `Tie`) and rated both systems on factuality and usefulness (5-point Likert scale). Results are summarised in Table 7.

Table 7: Summary of qualitative evaluation (pairwise preference and mean Likert ratings).

Rater	**RAG Wins**	**BASE Wins**	**Ties**	p**-value**	**Fact. (RAG/BASE)**	**Useful. (RAG/BASE)**
Rater 1	13	20	17	0.30	4.26 / 4.36	4.14 / 4.34
Rater 2	28	15	7	0.07	4.72 / 3.96	4.24 / 3.72
Rater 3	36	14	0	0.003	4.70 / 4.34	4.38 / 4.10
Rater 4	36	14	0	0.003	4.32 / 3.16	4.14 / 3.64
Aggregate	**113**	**63**	**24**	**0.0002**	**4.50 / 3.96**	**4.23 / 3.95**

RAG is **significantly preferred overall** ($p<0.01$), with higher mean factuality and usefulness. **Inter-rater agreement is modest** (Fleiss' $\kappa=0.13$, 51% raw agreement), consistent with the subjective variation that we expect in legal interpretation, especially among non-expert audiences. Qualitative comments show that RAG answers are more often *grounded and precise*, while Non-RAG outputs tend to overgeneralise or speculate on unseen provisions. Some representative examples will be reviewed under Section 5.4.

5.3 Ablation Studies

To quantify the contribution of individual components, we conduct three controlled and independent ablations over (i) retriever architecture, (ii) reranker design, and (iii) SLM reasoning mode. Each ablation isolates a single subsystem while keeping all other modules fixed to the main system defaults. The best-performing configuration from each stage is then adopted in subsequent experiments to form the final end-to-end RAG pipeline. All ablations are evaluated on the same **PDPABench-Test** split under identical prompting, decoding, and scoring procedures to ensure comparability.

Retriever Architecture. Figure 2 compares lexical (BM25), dense (E5/MiniLM), and hybrid (BM25 + Dense via RRF) retrievers. While the dense retriever attains the highest MRR and NDCG, the hybrid configuration achieves the highest Recall@10 while maintaining competitive ranking performance. Given that the retrieval stage prioritises broad coverage of relevant spans before reranking, this motivates the choice of the hybrid retriever as the default configuration, leveraging both lexical and semantic signals.

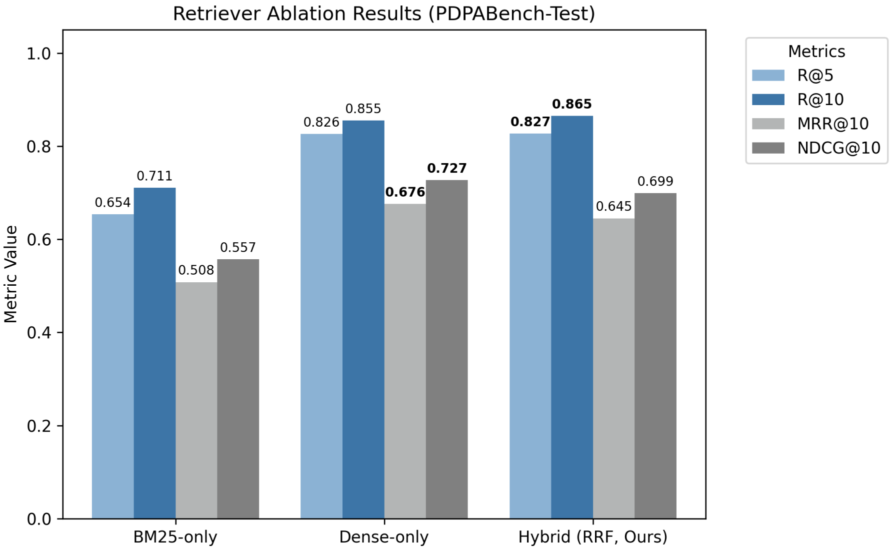

Figure 2: Grouped Bar Plot of Retriever ablation results (**PDPABench-Test**).

Reranker Architecture. With the hybrid retriever fixed, we compare the Cross-Encoder and LightGBM LTR rerankers in Figure 3. The Cross-Encoder achieves higher top-k retrieval accuracy and ranking consistency, showing a +45.2 point gain in NDCG@3 over LightGBM.

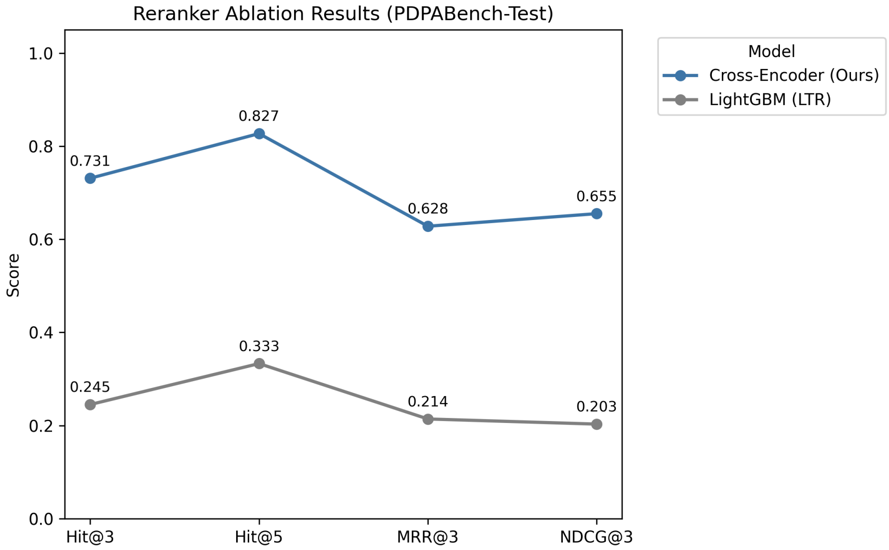

Figure 3: Line Plot of Reranker ablation results (**PDPABench-Test**).

SLM Reasoning vs. Non-Reasoning. We next isolate the impact of reasoning-enabled generation. Figure 4 reports citation-level and textual metrics for both SLM variants under identical retrieval and reranking settings.

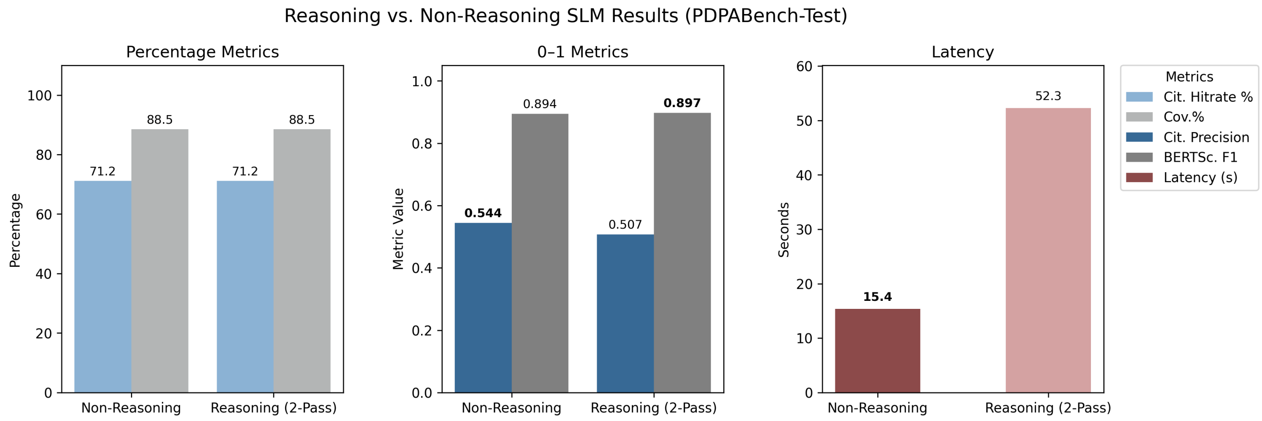

Figure 4: Grouped Bar Plots of Reasoning vs. Non-Reasoning SLM results (**PDPABench-Test**).

Enabling reasoning yields only marginal gains in semantic similarity (BERTScore +0.003) while slightly lowering citation precision (–0.037), suggesting that longer reasoning chains occasionally introduce unsupported references. Both models maintain identical coverage, indicating no benefit to abstention behaviour. However, the decoding latency increases by approximately 2.5×. Given the negligible improvement in factuality and the substantial computational overhead, the reasoning-enabled mode is excluded from the final system configuration.

5.4 Error and Case Analysis

We analyse representative outputs from the best-performing configuration (Hybrid RRF retriever + Cross-Encoder reranker + Non-reasoning SLM) to identify residual failure modes and system behaviour. Table 8 presents three illustrative examples covering correct abstention, faithful grounding, and partial retrieval error.

(i) Correct Abstention. For the question *"If an organisation uses personal data based on unclear authority, does PDPA allow it?"*, the model responded *"I'm not sure; this appears outside the provided statute."* This demonstrates a correctly triggered abstention: the query lacks a precise statutory referent so the system abstains rather than speculating. The result reflects effective calibration of the abstention gate at $\tau = 0.031$, consistent with the selective-risk analysis where factual reliability was prioritised over full coverage.

(ii) Correct Grounding. For *"Does the PDPA still apply to information about someone who died 5 years ago?"*, the model accurately cited `PDPA s.4(4)` and explained that the Act applies to data of individuals deceased for ten years or less. This case illustrates strong grounding behaviour: the retriever surfaced the correct clause, and the generator rephrased it faithfully in plain language while retaining statutory nuance.

(iii) Partial Retrieval Error. For *"If a law enforcement agency collects data under another law, does PDPA still apply?"*, the system cited `s.21(4)` and `s.26D(9)` instead of the governing provision `s.4(6)`. Although the generated answer remained coherent, it relied on legally adjacent but non-authoritative sections, indicating a retrieval-level miss rather than a generation error. This reflects a limitation in cross-part clause linkage: the retriever favoured lexical similarity ("law enforcement", "disclosure") over statutory scope.

Table 8: Representative qualitative examples from the best-performing configuration.

QA Type	Question	Model's Answer (excerpt)	Gold Citation(s)
Pure-Abstain	If an organisation uses personal data based on unclear authority, does PDPA allow it?	"I'm not sure; this appears outside the provided statute."	N/A
Pure-Definitive	Does the PDPA still apply to information about someone who died 5 years ago?	"The PDPA does not apply to personal data about a deceased individual if they have been dead for 10 years or less..."	PDPA s. 4(4)
Scenario-Ambiguous (retrieval miss)	If a law enforcement agency collects data under another law, does PDPA still apply?	"Yes, PDPA still applies. If a law enforcement agency collects data under another law, the... (PDPA s. 21(4), s. 26D(9))."	PDPA s. 4(6)

6 Discussion & Limitations

Our results suggest that a citation-first, abstention-capable RAG architecture meaningfully improves the reliability of statute-grounded PDPA QA compared to an ungrounded SLM, but they also expose several structural limitations in both the system and the evaluation design.

6.1 Interpreting the Impact of Citation-First RAG

The largest quantitative gain is in citation behaviour: our RAG system improves citation hit rate by roughly 69 percentage points over the Non-RAG baseline, with citation F1 of 0.60 and recall of 0.67. This matches the system's design goal—to answer fewer questions, but with traceable references into the PDPA. Modest but consistent gains in ROUGE-L and BERTScore suggest that retrieval does more than append citations; it also nudges the generator toward statutory phrasing and structure, keeping outputs closer to the gold answers.

At the same time, citation precision of 0.54 shows that the model sometimes "over-cites" or includes legally adjacent but non-authoritative sections. The law-enforcement exception error—where the system cited s.,21(4) and s.,26D(9) instead of the governing s.,4(6)—illustrates this: the retriever finds PDPA-relevant material, but the answer is grounded in a cluster of nearby clauses rather than the most normatively central one. This may look authoritative to lay users, but from a compliance standpoint it falls short of a strict "citation-first" ideal.

The ablations clarify this behaviour. Dense retrieval alone yields the best MRR and NDCG, but the hybrid BM25+dense retriever achieves the highest Recall@10. In a setting where missing the correct section is worse than ranking it slightly lower, prioritising recall is defensible. The cross-encoder reranker then recovers much of the ranking quality, with large gains over the LTR reranker (Hit@3: 0.73 vs. 0.25). Overall, the factuality and citation gains appear to come primarily from (i) high first-stage recall and (ii) a reranker that captures fine-grained token interactions in short statutory spans, rather than from more sophisticated generation.

The reasoning-enabled SLM variant reinforces this view. Despite much higher latency, it produces nearly identical citation hit rates and coverage, with only a marginal BERTScore improvement and lower citation precision. Additional "thinking" does not correct retrieval errors or misprioritised sections; once the wrong evidence is in the context window, longer reasoning chains can amplify, rather than dampen, spurious inferences.

6.2 Abstention, Risk Calibration, and User Trust

The abstention mechanism captures a central trade-off in legal QA: *when is it better to say "I'm not sure" than to risk a misleading answer?* Threshold tuning on PDPABench-Val yields an operating point with F1 ≈ 0.75 at $\sim$87–89% coverage; on the Test split, the system abstains on a modest fraction of queries and achieves relatively high citation F1 on those it chooses to answer.

However, the confusion matrix (2 true-positive abstentions vs. 5 false negatives) shows that the system is more likely to *answer when it should abstain* than the reverse. In safety-critical or high-liability settings, such false negatives are more problematic than over-abstention: a confidently wrong answer with canonical citations is harder to detect than an explicit "I'm not sure." Moreover, the scalar thresholds ($\tau_{\min}, \tau_{\Delta}$) are tuned solely on PDPABench-Val and may not transfer to real-world, more heterogeneous or adversarial queries.

The interaction between "soft" (prompt-based) and "hard" (score-based) abstention adds another limitation. The soft layer instructs the SLM to abstain when evidence is insufficient, but the model can still fabricate a plausible answer from partial evidence. The hard layer then acts as a post-hoc gate based only on reranker scores. This improves modularity but makes abstention behaviour depend almost entirely on retrieval confidence rather than on properties of the generated answer (e.g., hedging, internal contradictions).

6.3 Dataset and Evaluation Limitations

PDPABench is both a strength and a major source of potential bias. It enforces canonical citations, section-disjoint splits, and explicit abstain categories, all well-aligned with our system's goals. At the same time, several aspects limit how far the reported gains can be generalised.

First, a substantial fraction of the 500 QAs are LLM-generated expansions of a small human-written seed set. Although automated validation enforces citation format and mapping to corpus chunks, stylistic and distributional artefacts from the synthesising model may remain. The same applies to the gold answers, which reflect a specific paraphrasing and emphasis of the statute. This increases the risk that models trained and evaluated on PDPABench overfit to its question styles and answer templates, rather than learning robust statute-level reasoning.

Second, the QA-type distribution is heavily skewed towards Pure-Definitive questions (340/500), with relatively few Scenario-Ambiguous and Pure-Abstain items. This imbalance weakens the metrics that matter most for deployment, such as multi-section reasoning and abstention reliability. For example, the abstention confusion matrix is based on very few positive cases, making the reported precision and recall inherently noisy. Future iterations of PDPABench should deliberately up-weight ambiguous, cross-part, and out-of-scope queries to stress-test retrieval and abstention more aggressively.

Third, human evaluation used four lay raters and a rater-friendly generic rubric. While this is suitable for assessing clarity and perceived usefulness, it is less sensitive to subtle doctrinal errors (e.g., mis-scoping exceptions or conflating discretionary and mandatory obligations). Expert legal evaluation—or at least a mixed panel of domain experts and lay users—would be needed to determine whether the observed improvements translate into legally safe and practically reliable advice.

6.4 System Design Limitations and Future Extensions

Beyond evaluation, the system design itself embeds some constraints.

- **Corpus scope.** The retrieval corpus is deliberately restricted to the PDPA statute text (Parts 1–6 and 9–10) as of a specific snapshot date, excluding repealed sections, PDPC guidelines, enforcement decisions, and related regulations. This makes the system cleanly auditable but also brittle: many real-world PDPA questions are better answered by combining statutory text with guidance, case law, or cross-references to other statutes (e.g., sectoral regulations, evidence law). The system cannot currently express this nuance; at best, it abstains.

- **Flat subsection indexing.** Treating each subsection as an independent retrieval unit simplifies indexing and citation but ignores the hierarchical and cross-referential structure of the Act. The retrieval miss in the law-enforcement scenario suggests that cross-part dependencies (e.g., general application clauses in Part 1 that govern specific obligations elsewhere) are not adequately captured. Incorporating structured representations (e.g., graph over sections and cross-references, or parent–child features in the retriever/reranker) could reduce such errors.

7 Conclusion & Next Steps

This work presents a PDPA-focused, citation-first RAG system grounded in an authoritative statute corpus and a purpose-built benchmark, PDPABench. By enforcing retrieval, canonical citations, and abstention, the system shifts a general-purpose language model towards more reliable, statute-grounded answers. Qualitative and quantitative evidence together suggest that users receive responses that are not only more precisely tied to PDPA provisions, but also clearer and more useful than those from an ungrounded baseline. At the same time, error analysis shows that the system can still mis-prioritise legally adjacent sections or answer when it ought to abstain, highlighting the limits of purely score-based confidence and flat retrieval over a structurally complex Act.

More broadly, our cleaned, citation-aware PDPA corpus and PDPABench fill a clear gap in legal NLP: existing work may include PDPA, but none support statute-level QA with canonical citations and explicit abstain cases. Together, they offer a reusable template for statute-centric assistants in underrepresented jurisdictions, prioritising traceability and conservative behaviour over open-ended fluency.

Looking ahead, we see two main directions for improvement:

- **Richer corpus and benchmark.** Extend beyond the bare statute to include PDPC guidelines, enforcement decisions, and related regulations, and rebalance PDPABench towards more scenario-based, cross-section, and out-of-scope queries, ideally with partial expert review.

- **Better calibration and abstention.** Combine retrieval scores with generation-side signals (uncertainty, self-consistency, LLM-as-a-judge checks) and train dedicated abstention or risk models on harder, more diverse data to make "I'm not sure" both more frequent where appropriate and more interpretable to end users.

Taken together, these extensions would turn the current prototype into a more robust, legally sensitive assistant while preserving its core strengths: authoritative grounding, transparent citations, and an explicit commitment to abstain when the statute alone does not justify an answer.

References

[1] Attorney-General's Chambers, Singapore. Personal data protection act 2012 (no. 26 of 2012) - singapore statutes online. https://sso.agc.gov.sg/Act/PDPA2012, 2012. Accessed: 2025-11-16.

[2] Ilias Chalkidis, Abhik Jana, Dirk Hartung, et al. Lexglue: A benchmark dataset for legal language understanding in english. arXiv:2110.00976, 2022.

[3] Neel Guha, Daniel Martin Katz, Peter Henderson, Daniel E. Ho, et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *NeurIPS 2023 Datasets and Benchmarks Track*, 2023.

[4] Aakanksha Gupta, Laurie Ryan, Noreen Iftikhar, et al. Can large language models detect real-world android software compliance violations? *Empirical Software Engineering*, 2025. Includes CompliBench with PDPA, LGPD, and PIPEDA.

[5] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*, 2017.

[6] Dev Isaacus and collaborators. The massive legal embedding benchmark (mleb). https://github.com/isaacus-dev/mleb, 2025. Includes Singapore judicial data; no PDPA statute QA.

[7] Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage, 4th edition, 2018. Standard reference for Krippendorff's α.

[8] Margaret Li, Jason Weston, and Stephen Roller. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. In *arXiv:1909.03087*, 2019. URL https://arxiv.org/abs/1909.03087.

[9] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, et al. Training language models to follow instructions with human feedback. *arXiv:2203.02155*, 2022. URL https://arxiv.org/abs/2203.02155.

[10] Personal Data Protection Commission. Pdpa assessment tool for organisations (pato). https://apps.pdpc.gov.sg/resources/pato, 2024.

[11] Su Pipitone, Stefan Trautmann, and Ilias Chalkidis. Legalbench-rag: A benchmark for retrieval-augmented legal reasoning. In *Findings of ACL 2024*, 2024.

[12] Usman Qamar, Hui Fang Ng, Adriel Yeo, and Li Cheng Tan. Detecting compliance of privacy policies with data protection laws. In *IEEE International Conference on Big Data*, 2021.

[13] Singapore Government Developer Portal. Retrieval-augmented generation playbook. https://www.developer.tech.gov.sg/guidelines/standards-and-best-practices/retrieval-augmented-generation-playbook.html, 2025.

[14] The Straits Times. Ai-powered search engine to help singapore lawyers with legal research. https://www.straitstimes.com/singapore/ai-powered-search-engine-to-help-singapore-lawyers-with-legal-research, 2025.

A Architectural Diagram

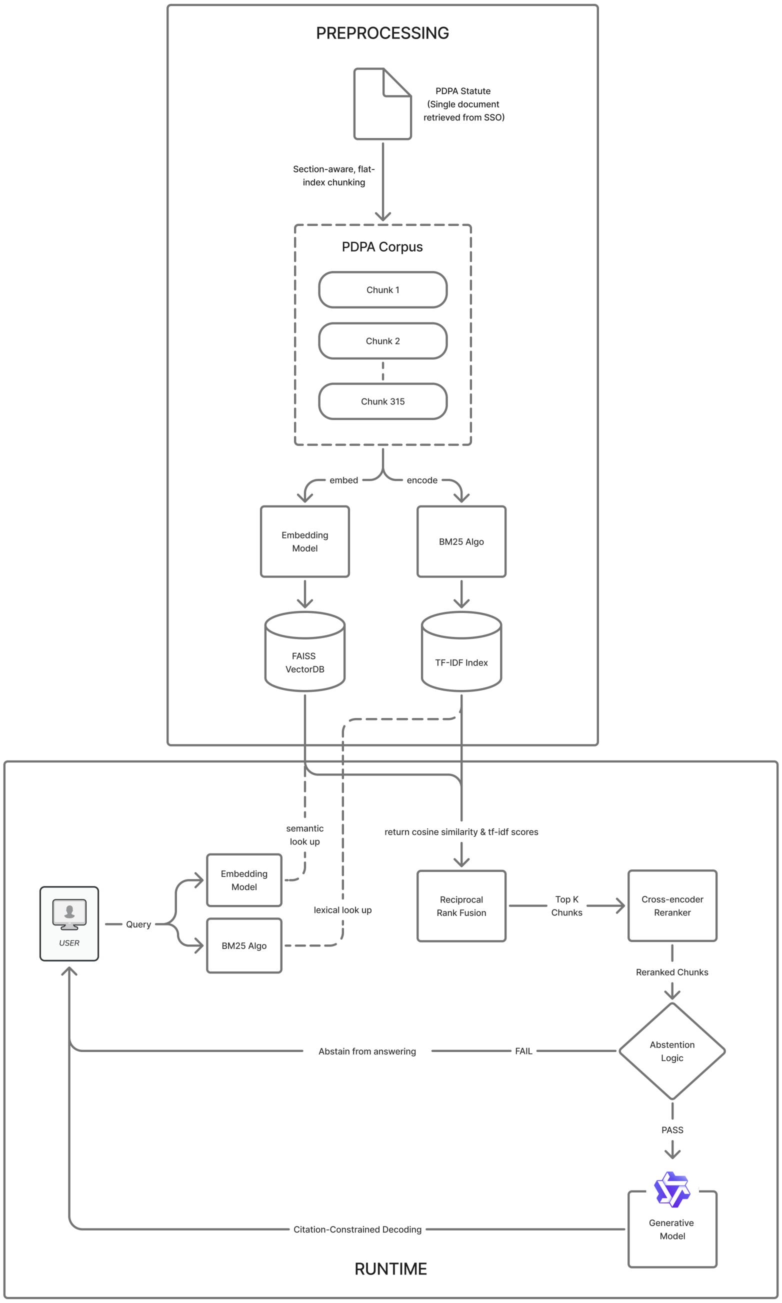

Figure 5: End-to-end PDPA RAG pipeline. Authoritative sources are ingested and chunked with canonical citations; hybrid retrieval with RRF feeds a cross-encoder reranker; decoding is constrained to cite evidence; an abstention gate enforces reliability; all steps are logged for audit and tuning.