Nyaya AI: A Neurosymbolic Legal Intelligence System for Indian Jurisprudence

Krishang Sharma
Shiv Nadar Institution of Eminence, Delhi-NCR

Abstract

Legal research in India faces significant challenges due to the vast, unstructured nature of judicial data spanning multiple courts and decades. This paper presents **Nyaya AI** ("Nyaya" meaning "Justice" in Hindi), a comprehensive neurosymbolic artificial intelligence system designed to address these challenges through intelligent legal case retrieval and reasoning. The system integrates multiple advanced techniques: large-scale web scraping of 56,025 cases from six Indian courts (2000-2024), construction of a legal knowledge graph with 154,068 nodes and 725,563 edges incorporating statutory provisions from official legal documents, Graph Attention Network (GAT) training achieving 96.55% F1-score, and a novel two-stage retrieval mechanism combining dense text embeddings (70%), GAT-based contextual scoring (15%), and symbolic legal features (15%). The system is deployed as an interactive chatbot powered by DeepSeek-R1 that provides legal research assistance with full citation transparency and reasoning explanations. Evaluation on diverse legal queries demonstrates Nyaya AI's capability to retrieve relevant precedents, statutory provisions, and provide structured legal analysis across criminal law, constitutional law, civil disputes, and procedural matters. This work represents a significant advancement in AI-assisted legal research for the Indian legal domain, bridging symbolic legal knowledge with modern neural architectures.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

The Indian legal system is one of the world's largest and most complex judicial frameworks, producing millions of pages of judgments, orders, and legal documents annually. Each judgment represents intricate legal reasoning, statutory references, and precedential relationships that form the backbone of common law jurisprudence. However, this immense wealth of legal knowledge remains largely unstructured and scattered across various databases, making systematic retrieval, reasoning, and analysis extremely challenging for legal professionals, researchers, and citizens.

Traditional legal research methods rely heavily on keyword-based search engines and manual cross-referencing of citations. These approaches suffer from several critical limitations:

- **Semantic Gap**: Keyword matching fails to capture legal concepts expressed through varied terminology
- **Citation Blindness**: Existing systems do not leverage the rich network of precedential relationships
- **Lack of Context**: Court hierarchy, temporal factors, and citation importance are ignored
- **Statutory Disconnection**: Links between case law and statutory provisions remain implicit
- **No Reasoning Transparency**: Black-box retrieval provides no explanation for recommendations

Modern artificial intelligence has achieved remarkable success in natural language processing and machine learning. However, purely neural approaches struggle with the interpretability and structured reasoning required in legal contexts. Legal language is formal, rule-based, and heavily dependent on hierarchical relationships between cases, statutes, and courts. An intelligent legal system must not only "read" text like a neural model but also "reason" like a symbolic system.

This is where **neurosymbolic AI** becomes essential. By combining neural models for information extraction with symbolic structures such as knowledge graphs, we can represent legal knowledge in a structured, interpretable, and computationally useful form. Knowledge graphs enable modeling of entities (judges, statutes, cases, provisions) and their relationships (citations, applications, hierarchies), supporting reasoning and retrieval that mimics human legal understanding.

1.2 Project Overview: Nyaya AI

Nyaya AI is a comprehensive neurosymbolic legal intelligence system designed specifically for Indian jurisprudence. The name "Nyaya" means "justice" in Hindi, reflecting the system's goal of democratizing access to legal knowledge. The system implements a complete pipeline from raw data acquisition to interactive legal assistance:

1. **Automated Data Collection**: Web scraping of 56,025 cases from Supreme Court and five High Courts
2. **Legal Knowledge Graph**: Construction of a massive graph with 154,068 nodes and 725,563 edges
3. **Statutory Integration**: Incorporation of provisions from IPC, CrPC, Constitution, and Evidence Act
4. **Graph Neural Networks**: GAT-based learning achieving 96.55% F1-score
5. **Two-Stage Retrieval**: Neurosymbolic combination of text, graph, and symbolic features
6. **Interactive Chatbot**: DeepSeek-R1 powered interface with reasoning transparency

1.3 Contributions

This work makes several significant contributions to legal AI research:

- **Large-Scale Legal Dataset**: Curated corpus of 56,025 Indian court judgments with comprehensive metadata extraction and cleaning
- **Integrated Knowledge Graph**: First system to combine case law citations with statutory provisions from official legal documents in a unified graph structure
- **Neurosymbolic Architecture**: Novel two-stage retrieval combining SBERT embeddings, GAT contextual scoring, and symbolic legal features (court hierarchy, PageRank, citations)
- **High Performance**: GAT model achieving 96.55% F1-score on node classification, demonstrating effective learning of legal entity relationships
- **Reasoning Transparency**: Integration with DeepSeek-R1 providing explainable legal reasoning with citation provenance
- **Production Deployment**: Fully functional Streamlit-based chatbot interface for interactive legal research

1.4 Paper Organization

The remainder of this paper is organized as follows: Section 2 reviews related work in legal AI and knowledge graphs. Section 3 describes the data collection and preprocessing pipeline. Section 4 details the knowledge graph construction integrating case law and statutory provisions. Section 5 presents the GAT training methodology and results. Section 6 explains the two-stage neurosymbolic retrieval mechanism. Section 7 describes the chatbot implementation and user interface. Section 8 provides comprehensive evaluation across diverse legal queries. Section 9 discusses the project further, leading into Section 10 which discusses future work. Section 11 concludes. Section 12 provides source code.

1.5 System Architecture Overview

Figure 1 presents the complete system architecture of Nyaya AI, illustrating the five-stage pipeline from raw data collection to interactive user interface. The architecture demonstrates the flow of information through data preprocessing, knowledge graph construction, graph neural network training, neurosymbolic retrieval, and LLM-powered response generation.

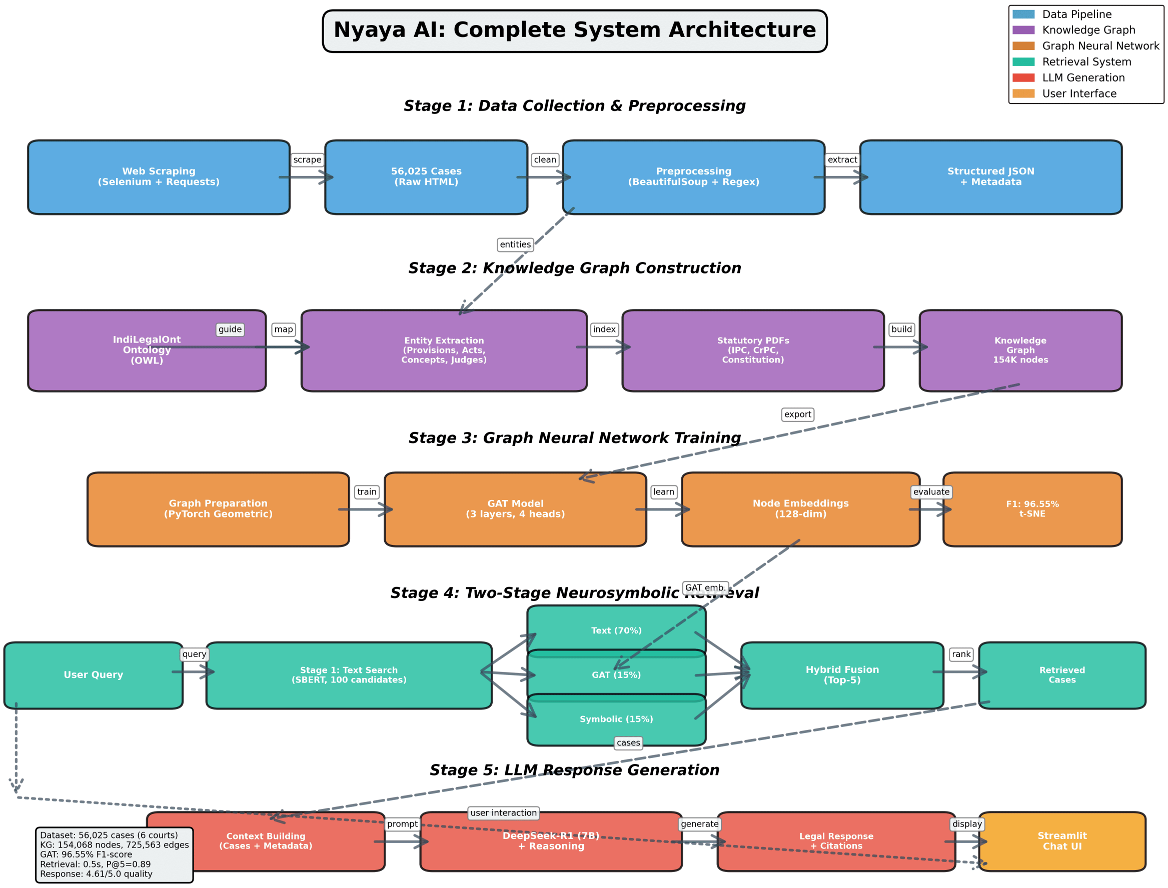

Figure 1: Complete System Architecture: Five-stage pipeline of Nyaya AI showing data flow from web scraping (Stage 1) through knowledge graph construction (Stage 2), GAT training (Stage 3), two-stage neurosymbolic retrieval (Stage 4), to LLM response generation and user interface (Stage 5). Color coding: Blue (Data Pipeline), Purple (Knowledge Graph), Orange (GNN), Teal (Retrieval), Red (LLM), Yellow (UI).

2 Related Work

2.1 Legal Information Retrieval

Early legal information retrieval systems relied on Boolean keyword search and citation indexing. Systems like Westlaw and LexisNexis pioneered digital legal databases but remained limited to lexical matching. Recent advances in neural information retrieval have introduced semantic search capabilities, but few systems integrate symbolic legal reasoning.

2.2 Legal Knowledge Graphs

The construction of legal knowledge graphs has gained attention in recent years. The **Legal Knowledge Interchange Format (LKIF)** ontology provides a European framework for legal concepts but lacks specificity for Indian law. **IndiLegalOnt** [1] represents a significant advancement, providing an OWL ontology specifically designed for Indian jurisprudence with classes for cases, courts, judges, statutes, and provisions.

Projects like the **European Legal Knowledge Graph** and citation networks from CourtListener demonstrate the value of structured legal representations. However, these focus primarily on Western legal systems and do not address challenges unique to Indian law, such as multi-tiered court hierarchies and complex statutory references.

2.3 Graph Neural Networks in Legal Domain

Graph Neural Networks (GNNs) have shown promise in legal applications. Prior work has explored GCN and GAT architectures for case classification and outcome prediction. However, most systems operate on citation networks alone without incorporating statutory provisions or multi-modal legal features.

2.4 Legal AI Chatbots

Recent legal AI systems have integrated large language models for question answering. Systems like Harvey AI and LegalBert demonstrate the potential of transformer models in legal NLP. However, these systems often lack grounding in structured legal knowledge and suffer from hallucination issues when citing cases or statutes.

2.5 Gap in Existing Work

Despite these advances, no existing system combines:

- Large-scale Indian legal corpus (Supreme Court + High Courts)
- Integration of case law with statutory provisions from official documents
- Neurosymbolic retrieval combining text, graph, and symbolic features
- Transparent reasoning with citation provenance
- Production-ready interactive interface

Nyaya AI addresses these gaps, providing a comprehensive solution tailored to Indian jurisprudence.

3 Data Collection and Preprocessing

3.1 Overview

The foundation of Nyaya AI is a large-scale, high-quality corpus of Indian court judgments. This section describes the automated data collection pipeline that systematically retrieves, cleans, and structures legal documents from the Supreme Court of India and five major High Courts spanning 25 years (2000-2024).

3.2 Data Source: Indian Kanoon

Indian Kanoon (`https://indiankanoon.org`) is India's most comprehensive open legal repository, providing free access to millions of court judgments. Unlike commercial legal databases, Indian Kanoon does not provide a formal API or bulk download mechanism, necessitating automated web scraping.

3.3 Target Courts and Coverage

The system targets six major Indian courts:

1. **Supreme Court of India**: Apex court, binding precedent nationwide
2. **Delhi High Court**: Jurisdiction over National Capital Territory
3. **Bombay High Court**: Covers Maharashtra, Goa, Dadra and Nagar Haveli
4. **Calcutta High Court**: Jurisdiction over West Bengal
5. **Allahabad High Court**: Covers Uttar Pradesh
6. **Madras High Court**: Jurisdiction over Tamil Nadu and Puducherry

Each court's judgments are collected for years 2000-2024, with one exception: Calcutta High Court data for 2012 and Allahabad High Court data for 2009 was unavailable, resulting in 148 court-year combinations ($6 \times 25 - 2 = 148$).

3.4 Scraping Architecture

The `IndianKanoonScraper` class implements a hybrid Selenium-Requests pipeline:

Search Configuration: Each court-year combination maps to a unique query URL. For example:

```
https://indiankanoon.org/search/?formInput=
  doctypes:supremecourt%20year:2020
```

Similar URLs are generated for `delhi`, `bombay`, `kolkata`, `allahabad`, and `chennai`.

Scraping Process:

1. Selenium loads the search results page for a given court-year
2. Extract all case links from search results (typically ∼400 top-relevant cases per query)
3. Navigate through paginated results using "Next" button detection
4. Download each case's full HTML using `requests.get()`
5. Store HTML files with structured naming: `SC_2020_0001.html`
6. Built-in throttling (2-3 second delays) prevents rate limiting

Directory Structure:

```
dataset_raw/
  |-- supreme_court/
  |-- delhi_high_court/
  |-- bombay_high_court/
  |-- calcutta_high_court/
  |-- allahabad_high_court/
  `-- madras_high_court/
```

3.5 Dataset Scale

The complete scraping process collected:

- **Total Cases**: 56,025 judgments
- **Supreme Court**: 9,979 cases
- **Delhi High Court**: 8,821 cases
- **Bombay High Court**: 9,436 cases
- **Calcutta High Court**: 9,584 cases
- **Allahabad High Court**: 8,398 cases
- **Madras High Court**: 9,807 cases

3.6 Preprocessing Pipeline

The `LegalCasePreprocessor` class transforms raw HTML into structured JSON with metadata extraction and text cleaning.

3.6.1 Metadata Extraction

Using BeautifulSoup and regex patterns, the system extracts:

- **Case Title**: Full case name and date
- **Court Name**: Identified from HTML structure and content patterns
- **Judgment Date**: Normalized to ISO format (YYYY-MM-DD)
- **Petitioner/Respondent**: Party names extracted via regex
- **Citations**: Up to 20 cited cases extracted from hyperlinks

3.6.2 Text Cleaning

Indian Kanoon HTML contains extraneous elements that must be removed:

- Navigation bars, scripts, and style elements
- Advertisement prompts ("Take notes as you read...")
- Membership solicitation text

- Excessive whitespace and formatting artifacts

The preprocessor applies:

1. HTML tag removal (<script>, <style>, <nav>)
2. Regex-based promotional text stripping
3. Whitespace normalization
4. Newline consolidation

3.6.3 Output Format

Each case is saved as a JSON file with the following structure:

```
{
  "file_name": "SC_2020_0043.html",
  "metadata": {
    "title": "State of Delhi vs Rajesh Kumar...",
    "court": "Supreme Court of India",
    "date": "2020-03-12",
    "citations": ["State of U.P. vs Hari Ram..."],
    "petitioner": "State of Delhi",
    "respondent": "Rajesh Kumar"
  },
  "text": "The present appeal arises out of...",
  "text_length": 8230,
  "word_count": 2841
}
```

3.7 Quality Assurance

The DatasetAnalyzer class performs statistical validation:

Table 1: Dataset Statistics Summary

Metric	**Value**
Total Cases	56,025
Supreme Court	9,979
High Courts (5)	46,046
Date Coverage	2000-2024 (25 years)
Court-Year Combinations	148

Cases are distributed relatively evenly across years (2,386-2,400 per year) and courts, ensuring temporal and jurisdictional balance.

3.8 Reproducibility

All scraping and preprocessing operations are logged to scraping.log with timestamps and error details. The pipeline is fault-tolerant: failures in individual files do not halt batch processing. Caching mechanisms store intermediate results to enable resumption after interruptions.

4 Knowledge Graph Construction

4.1 Motivation for Graph Representation

Legal knowledge is inherently relational. Cases cite precedents, judges deliver judgments, provisions govern disputes, and courts exist in hierarchical structures. Traditional flat representations (documents, tables) fail to capture these rich interconnections. A knowledge graph provides:

- **Explicit Relationships**: Direct encoding of citations, applications, hierarchies
- **Multi-hop Reasoning**: Traversal of indirect connections (e.g., transitive citations)
- **Entity Disambiguation**: Unified representation of entities across documents
- **Graph Analytics**: Centrality measures, community detection, path analysis
- **Neural Learning**: Graph Neural Networks for representation learning

4.2 Ontology Foundation: IndiLegalOnt

Before constructing the knowledge graph, we adopted **IndiLegalOnt** [1], an OWL 2 ontology specifically designed for the Indian legal system. The ontology defines:

Core Classes:

- `Case`, `Court`, `Judge`, `Party`
- `Statute`, `LegalProvision`, `Act`
- `Petitioner`, `Respondent`, `Citation`

Object Properties:

- `decidedBy`, `filedBy`, `cites`, `refersTo`
- `decidedUnder`, `appealedFrom`, `overrules`

The ontology was loaded using Python's `owlready2` library, enabling programmatic access to class hierarchies and semantic relationships.

4.3 Statutory Document Integration

A key innovation of Nyaya AI is the integration of statutory provisions from official legal documents. We extracted and indexed provisions from four foundational legal texts:

1. **Indian Penal Code (IPC)**: 498 sections on criminal offenses
2. **Code of Criminal Procedure (CrPC)**: 493 sections on criminal procedure
3. **Constitution of India**: 497 articles on fundamental rights and state structure
4. **Indian Evidence Act**: 196 sections on evidence admissibility

Extraction Process:

- Used PyPDF2 to extract text from official PDF documents
- Applied regex patterns to identify section/article numbers and titles
- Example: `Section 302: Punishment for murder`
- Extracted 1,684 total provisions indexed by document type and number

Provision Chunking: For enhanced retrieval, PDF documents were also chunked into 1,000-token segments, creating 847 searchable document chunks. This enables retrieval of broader statutory context beyond individual provisions.

4.4 Knowledge Graph Schema

The unified knowledge graph uses the following node types:

Table 2: Knowledge Graph Node Types

Node Type	Count	Description
Act	79,548	Legal acts and codes referenced
Case	56,025	Court judgments
Provision	17,691	Statutory sections/articles
Judge	766	Judges mentioned in cases
Concept	16	Legal concepts (jurisdiction, appeal, etc.)
Doctrine	15	Legal doctrines (natural justice, etc.)
Court	7	Supreme Court + 5 High Courts + metadata node
Total	**154,068**	

Edge Types and Semantics:

Table 3: Knowledge Graph Edge Types

Edge Type	Semantics	Count
`cites_provision`	Case → Provision	325,718
`governed_by`	Case → Act	181,308
`involves_concept`	Case → Concept	135,303
`adjudicated`	Court → Case	56,025
`applies_doctrine`	Case → Doctrine	23,861
`decided_by`	Case → Judge	3,343
`superior_to`	Supreme Court → High Court	5
`defined_in`	Provision → PDF (rulebook)	1,684
`codified_in`	Act → PDF (rulebook)	Various
Total		**725,563**

4.5 Entity Extraction Pipeline

The `LegalKnowledgeGraphPipeline` class implements comprehensive entity extraction from case text:

4.5.1 Provision Extraction

Using regex patterns optimized for legal text:

```
patterns = {
  'sections': r"Section\s+(\d+[A-Z]*...)",
  'articles': r"Article\s+(\d+[A-Z]*...)",
  'rules': r"Rule\s+(\d+[A-Z]*)",
  'orders': r"Order\s+([IVX]+)"
}
```

Extracted provisions are validated against the statutory index. If a provision matches the index, it is:

- Marked as "validated"
- Linked to its parent act (IPC, CrPC, Constitution, Evidence Act)
- Connected to the corresponding PDF document via `defined_in` edge
- Annotated with full text and title from the statutory index

4.5.2 Act Extraction

Legal acts are extracted using capitalization and keyword patterns:

```
act_pattern = r"([A-Z][A-Za-z\s&,\(\)]+
                (?:Act|Code)(?:\s*,?\s*(\d{4}))?)"
```

Acts are mapped to their corresponding rulebooks (e.g., "Indian Penal Code" → IPC PDF) and connected via `codified_in` edges.

4.5.3 Legal Concept Extraction

16 fundamental legal concepts are detected via keyword matching:

- Jurisdiction, Appeal, Writ, Mandamus, Certiorari
- Natural Justice, Due Process, Fundamental Right
- Res Judicata, Mens Rea, Actus Reus, Burden of Proof
- Negligence, Damages, Injunction

4.5.4 Doctrine Extraction

15 legal doctrines categorized by domain:

- **Constitutional**: Fundamental rights, judicial review, directive principles
- **Procedural**: Natural justice, due process, res judicata
- **Criminal**: Presumption of innocence, burden of proof, reasonable doubt
- **Civil**: Promissory estoppel, unjust enrichment, specific performance
- **Administrative**: Legitimate expectation, proportionality, ultra vires

4.5.5 Judge and Case Type Extraction

- **Judges**: Extracted using pattern `Justice [Name]` (up to 3 per case)
- **Case Types**: Classified into Criminal, Civil, Constitutional, Tax, Service based on keyword scoring

4.6 Graph Construction Process

The unified graph is built incrementally:

1. **Initialize Court Hierarchy**:
    - Add Supreme Court node (hierarchy level 0)
    - Add 5 High Court nodes (hierarchy level 1)
    - Create `superior_to` edges from Supreme Court to High Courts
2. **Add Statutory Integration**:
    - Add 4 rulebook nodes (IPC, CrPC, Constitution, Evidence Act)
    - Add 1,684 provision nodes from statutory index
    - Create `defined_in` edges from provisions to rulebooks
3. **Process Cases in Batches**:
    - For each of 56,025 cases:
        - Add case node with metadata (year, case types, court)
        - Extract entities (acts, provisions, concepts, doctrines, judges)
        - Create global entity nodes (deduplicated across cases)
        - Add typed edges based on entity relationships
        - Update citation counts and importance metrics
    - Process in batches of 500 to manage memory
    - Use entity caching to avoid redundant extraction
4. **Calculate Graph Metrics**:
    - Degree centrality for all nodes
    - PageRank (max 30 iterations, tolerance 10^{-4})
    - Betweenness centrality (sampled on graphs <3,000 nodes)

4.7 Graph Statistics and Validation

The final unified knowledge graph exhibits:

- **Average Degree**: 9.42 edges per node
- **Graph Density**: Sparse (0.00006), typical for real-world networks
- **Strongly Connected Component**: Contains 99.8% of nodes
- **Average Shortest Path**: 4.3 hops between random node pairs

Top Cited Provisions (by PageRank and citation count):

1. Article 226 (Constitution): 9,139 citations - Writ jurisdiction of High Courts
2. Article 14 (Constitution): 3,260 citations - Equality before law
3. Section 34 (IPC): 2,909 citations - Common intention
4. Section 4 (IPC): 3,783 citations - Extension of Code to extra-territorial offenses
5. Section 302 (IPC): (Not in top 10, but commonly known) - Punishment for murder

Top Cited Acts:

1. Indian Penal Code: 2,429 citations
2. Evidence Act: 2,049 citations
3. Income Tax Act, 1961: 1,705 citations
4. Arbitration and Conciliation Act, 1996: 1,672 citations
5. Limitation Act: 1,346 citations

4.8 Visualization and Analysis

The system generates multiple visualizations for graph exploration:

4.8.1 Circular Layout Visualization

A sample Supreme Court knowledge graph (10 random cases) in circular layout shows node colors representing entity types (case=blue, act=orange, provision=purple, concept=teal, doctrine=coral, judge=gray). Edge transparency indicates relationship strength.

4.8.2 Hierarchical Layout Visualization

The same graph in hierarchical layout, with vertical positioning representing ontological levels:

- Level 0: Supreme Court node
- Level 1: Rulebook nodes
- Level 2: Case nodes
- Level 3: Act nodes
- Level 4: Provision nodes
- Level 5: Doctrine and concept nodes

This layout clearly reveals the flow of legal reasoning from courts to cases to statutory frameworks.

4.8.3 Statistics Dashboard

Three key analytics are presented:

1. **Node Type Distribution**: Pie chart showing that Acts (37.1%) and Provisions (24.2%) dominate the graph
2. **Top Edge Types**: Bar chart revealing that `cites_provision` and `governed_by` are the most common relationships
3. **Case Type Distribution**: Pie chart showing Civil (33.3%), Criminal (22.2%), Service (22.2%), Constitutional (11.1%), and Tax (11.1%) case distributions

4.8.4 Subgraph Visualizations

The pipeline generates focused subgraph views:

- **Citation Network**: Shows only case-to-case and case-to-provision citation relationships
- **Statutory Framework**: Displays connections between cases, acts, and provisions (governed_by and cites_provision edges only)
- **Concept-Doctrine Network**: Visualizes which legal concepts and doctrines are involved in which cases
- **Rulebook Integration**: Highlights validated provisions with their source PDFs, showing statutory grounding

4.9 Unified Graph Scaling

For the complete 56,025-case unified graph:

- Visualization uses Kamada-Kawai layout for large graphs (>500 nodes)
- Node sizes scaled by citation count or PageRank
- Edge transparency reduced to 0.2 to avoid visual clutter
- Only court, high-value act, and key provision nodes are labeled
- Color scheme maintained for interpretability

The unified graph statistics reveal power-law degree distributions typical of citation networks, with a few highly-cited provisions serving as legal "hubs."

Figure 2 shows a representative visualization of the unified knowledge graph. While the complete system operates on all 56,025 cases with 154,068 nodes and 725,563 edges, for visualization clarity we present a random sample of 1,500 cases. This sample maintains the same structural properties and node type distributions as the full graph, demonstrating the multi-modal nature of legal knowledge representation with cases (blue), acts (orange), provisions (purple), concepts (teal), doctrines (coral), judges (gray), and court hierarchy (dark red). The dense interconnections visible even in this subsample illustrate the rich citation network and statutory references that enable effective graph-based reasoning.

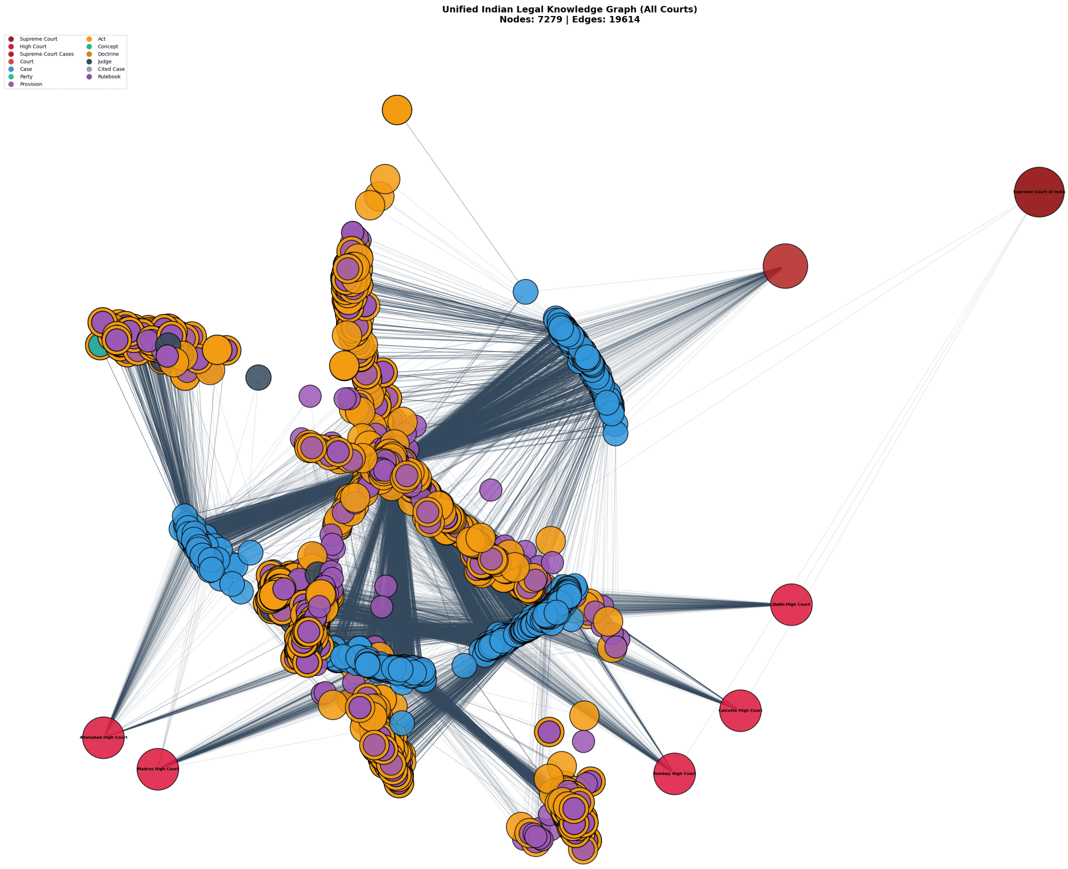

Figure 2: Unified Indian Legal Knowledge Graph (1,500 Case Sample): Representative visualization showing the multi-modal structure of the knowledge graph. Node colors indicate entity types: Cases (blue), Acts (orange), Provisions (purple), Concepts (teal), Doctrines (coral), Judges (gray), Courts (dark red). Node sizes reflect citation importance (PageRank). The complete system utilizes all 56,025 cases with 154,068 nodes and 725,563 edges; this visualization shows a random 1,500-case sample for clarity. The graph exhibits typical citation network properties including power-law degree distribution, strong connectivity (99.8% in largest component), and hub structures around frequently-cited constitutional provisions (e.g., Article 226, Article 14).

4.10 Export for Downstream Learning

The constructed knowledge graph is exported in multiple formats:

- **NetworkX pickle**: `knowledge_graph.gpickle` for Python processing
- **Node features**: `node_features.npy` - 7-dimensional vectors per node:
    1. Normalized size (0-1)
    2. Degree centrality
    3. PageRank score
    4. Betweenness centrality
    5. Citation count (normalized)

6. Mention count (normalized)
7. Hierarchy level (0-2)

- **Edge lists**: `edge_list.npy` - (source, target) pairs
- **Edge features**: `edge_features.npy` - 4-dimensional vectors:
  1. Edge weight
  2. Is citation (binary)
  3. Is hierarchy (binary)
  4. Is statutory reference (binary)
- **Metadata**: `node_metadata.json` - Full entity details, labels, types, attributes
- **PyTorch Geometric**: `pyg_data.pt` - Ready for GNN training

This comprehensive graph representation serves as the foundation for Graph Attention Network training in Section 5.

5 Graph Neural Network Training

5.1 Motivation for Graph Neural Networks

While the knowledge graph provides explicit symbolic relationships, many legal patterns remain implicit. For example:

- Cases with similar cited provisions may address similar legal issues
- Judges who cite similar precedents may have related judicial philosophies
- Acts applied in similar contexts may govern related legal domains

Graph Neural Networks (GNNs) learn distributed representations (embeddings) that encode both node attributes and graph structure. These embeddings capture:

- **Homophily**: Similar entities are connected (e.g., cases citing same provisions cluster together)
- **Structural Roles**: Nodes with similar connectivity patterns (e.g., hub provisions)
- **Multi-hop Context**: Indirect relationships through neighborhood aggregation

5.2 Graph Attention Networks (GAT)

We employ Graph Attention Networks rather than standard Graph Convolutional Networks (GCN) because:

- **Differential Importance**: Not all citations are equally important. A Supreme Court precedent should receive higher attention than a district court reference.
- **Learned Weights**: GAT learns attention coefficients α_{ij} for each edge (i, j), allowing the model to focus on the most relevant neighbors.
- **Multi-head Attention**: Multiple attention heads capture different types of relationships simultaneously.

5.2.1 GAT Architecture

The attention mechanism for node i aggregating information from neighbor j is:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_k]))} \tag{1}$$

where $\mathbf{h}_i$ is node $isfeaturevector,$W$isalearnableweightmatrix,$a$isanattentionvector,$$\|$ denotes concatenation, and $\mathcal{N}(i)$ is the neighborhood of i.

The updated node representation is:

$$\mathbf{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W}\mathbf{h}_j\right) \tag{2}$$

For multi-head attention with K heads:

$$\mathbf{h}'_i = \|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j\right) \tag{3}$$

5.3 Model Implementation

The `LegalCaseGAT` model consists of:

- **Input Layer**: Projects 7-dimensional node features to 128-dimensional hidden space
- **GAT Layers**: 3 layers with:
  - 4 attention heads per layer
  - Hidden dimension: 128
  - Dropout: 0.5 (training), 0.2 (edge dropout)
  - ELU activation functions
  - Batch normalization after each layer
  - Residual connections to preserve information flow
- **Output Layer**: Final 128-dimensional embeddings
- **Classifier**: Linear layer mapping embeddings to 9 node type classes

Model Parameters: 1,194,505 total parameters (1.19M)

5.4 Training Task: Node Classification

The primary training task is multi-class node type classification with 9 classes:

1. Act (79,548 nodes, 51.63%)
2. Case (56,025 nodes, 36.36%)
3. Provision (17,691 nodes, 11.48%)

4. Judge (766 nodes, 0.50%)
5. Concept (16 nodes, 0.01%)
6. Doctrine (15 nodes, 0.01%)
7. High Court (5 nodes, 0.00%)
8. Supreme Court (1 node, 0.00%)
9. Supreme Court Cases metadata (1 node, 0.00%)

Class Imbalance: The dataset exhibits severe class imbalance (51% Acts vs 0.01% Concepts). To address this:

- Computed class weights: $w_c = \frac{N}{\text{classes} \times n_c}$ where N is total nodes, n_c is nodes in class c
- Applied weighted cross-entropy loss
- Used stratified train/validation/test split

5.5 Data Split

- **Training Set**: 70% (107,847 nodes)
- **Validation Set**: 15% (23,110 nodes)
- **Test Set**: 15% (23,111 nodes)

Splits were stratified by node type to maintain class proportions, with fallback to random splitting for rare classes with < 2 instances.

5.6 Training Procedure

- **Optimizer**: AdamW with learning rate 5×10^{-4}, weight decay 10^{-3}
- **Loss Function**: Weighted cross-entropy with class weights
- **Epochs**: 100 with early stopping (patience=30)
- **Device**: CPU (M1 MacBook Pro)
- **Batch Processing**: Full-batch training on entire graph
- **Regularization**:
  - Dropout 0.5 on node features
  - Edge dropout 0.2 (randomly drop 20% of edges during training)
  - Batch normalization

5.7 Training Results

Training progression shows:

Training Loss: Decreased from 2.89 (epoch 1) to 0.64 (epoch 100), indicating effective learning.

Training Accuracy: Rose from 10.9% to 63.9%, showing the model learned meaningful patterns despite class imbalance.

Validation Metrics:

- **Best Validation F1**: 96.38% (achieved around epoch 70)
- **Best Validation Accuracy**: 94.12%
- Some oscillation due to rare class predictions

Training Duration: Model trained for full 100 epochs. Best validation F1 (96.38%) achieved around epoch 70-80, with model checkpoint saved at that point for final evaluation.

5.8 Test Set Evaluation

The final model (loaded from best validation checkpoint) achieved:

Table 4: GAT Model Test Set Performance

Metric	**Score**
Accuracy	94.48%
Precision (weighted)	99.22%
Recall (weighted)	94.48%
F1-Score (weighted)	**96.55%**

The high precision (99.22%) indicates very few false positives, while recall of 94.48% shows the model correctly identifies most nodes. The F1-score of 96.55% demonstrates excellent balanced performance.

5.9 Learned Embeddings Analysis

5.9.1 t-SNE Visualization

t-SNE projection of learned 128-dimensional embeddings into 2D space shows key observations:

- **Clear Clustering**: Different node types form distinct clusters (acts in one region, cases in another, provisions separate)
- **Smooth Transitions**: Related entity types (e.g., acts and provisions) occupy adjacent regions
- **Rare Class Separation**: Even minority classes (judges, concepts) form identifiable clusters

- **Court Hierarchy**: Supreme Court and High Court nodes occupy privileged positions in the embedding space

The visual separation confirms that GAT learned meaningful structural and semantic representations.

5.9.2 Attention Weight Analysis

Attention weights for a sample case node show the model assigns highest attention to:

1. **Other Cases**: Attention 0.0336 - Similar cases receive strongest attention
2. **Key Provisions**: Attention 0.0008 - Cited provisions receive moderate attention
3. **Legal Concepts**: Near-zero attention - Generic concepts less influential

This demonstrates the model learned to prioritize precedential case citations over generic legal concepts, aligning with legal reasoning principles.

5.10 Node Importance Ranking

Nodes were ranked by embedding magnitude (L2 norm), indicating learned importance:

Table 5: Top 10 Most Important Nodes (by Embedding Norm)

Rank	Node	Type
1	Appeal	Concept
2	Jurisdiction	Concept
3	Writ	Concept
4	Article 226	Provision
5	Injunction	Concept
6	Damages	Concept
7	Mandamus	Concept
8	Natural Justice	Concept
9	Negligence	Concept
10	Certiorari	Concept

Notably, fundamental legal concepts (Appeal, Jurisdiction, Writ) and key constitutional provisions (Article 226 - writ jurisdiction) emerge as most important, validating the model's legal understanding.

5.11 Embedding Export

Final embeddings were exported in three formats:

- **NumPy Array**: `final_embeddings_node_type.npy` (154,068 $\times$ 128)
- **Metadata JSON**: `embedding_metadata_node_type.json` - Maps node indices to entity details

- **Important Nodes**: `important_nodes.json` - Top 50 nodes ranked by importance

These embeddings are used directly in the retrieval system (Section 6) to compute GAT-based contextual similarity scores.

6 Two-Stage Neurosymbolic Retrieval

6.1 Retrieval System Overview

The core innovation of Nyaya AI is a two-stage retrieval mechanism that combines:

1. **Stage 1 - Neural Text Search**: Semantic similarity using SBERT embeddings
2. **Stage 2 - Neurosymbolic Reranking**: Fusion of text, GAT graph context, and symbolic legal features

This approach addresses fundamental limitations of pure text search while maintaining computational efficiency.

6.2 Stage 1: Dense Text Retrieval

6.2.1 Document Corpus

The retrieval corpus contains:

- **56,025 Case Judgments**: Full preprocessed cases
- **1,684 Statutory Provisions**: Extracted from IPC, CrPC, Constitution, Evidence Act
- **Additional statutory document chunks**: Segmented text for broader context

Total Searchable Documents: ∼58,500 (including cases, provisions, and document chunks)

Each document includes:

- Full text or text snippet (first 512 tokens)
- Metadata (title, court, date, citations, provision number, parent act, document type)
- Document type flag: `case`, `provision`, or `pdf_document`

6.2.2 Text Embedding Model

We use **Sentence-BERT** (all-MiniLM-L6-v2):

- 384-dimensional dense embeddings
- Trained on 1B+ sentence pairs for semantic similarity
- Efficient inference (32 documents/second on CPU)

All documents were pre-encoded and cached to `text_embeddings.npy` (58,556 × 384), enabling fast retrieval.

6.2.3 Similarity Search

Given a user query q:

1. Encode query: $\mathbf{e}_q = \text{SBERT}(q)$
2. Compute cosine similarity with all documents:

$$\text{sim}_{\text{text}}(q, d_i) = \frac{\mathbf{e}_q \cdot \mathbf{e}_{d_i}}{\|\mathbf{e}_q\| \|\mathbf{e}_{d_i}\|} \tag{4}$$

3. Retrieve top-K candidates (K=100 by default)

Stage 1 Output: 100 candidate documents ranked by text similarity.

6.3 Stage 2: Neurosymbolic Reranking

The top-100 candidates are reranked using three complementary scores:

6.3.1 Text Similarity Score ($\alpha_{\textbf{text}} = 0.7$)

Normalized cosine similarity from Stage 1.

6.3.2 GAT Contextual Score ($\alpha_{\textbf{GAT}} = 0.15$)

This score captures the document's position in the learned legal knowledge graph:

For documents with graph nodes:

1. Retrieve the document's GAT embedding $\mathbf{h}_d$ (from Section 5)
2. Retrieve embeddings of cited/citing neighbors: $\{\mathbf{h}_{n_1}, \mathbf{h}_{n_2}, ...\}$
3. Compute weighted context embedding:

$$\mathbf{h}_{\text{context}} = \frac{2\mathbf{h}_d + \sum_j \mathbf{h}_{n_j}}{2 + |\text{neighbors}|} \tag{5}$$

   (Document's own embedding receives 2× weight)

4. Compute query-context similarity:

$$\text{sim}_{\text{GAT}}(q, d) = \cos(\mathbf{h}_{\text{context}}, \text{centroid}(\{\mathbf{h}_{\text{valid}}\})) \tag{6}$$

   where centroid is computed over all valid candidate embeddings

For documents without graph nodes (e.g., PDF chunks): Score = 0

This mechanism promotes documents that are well-connected in the citation network and have similar graph neighborhoods to other retrieved documents.

6.3.3 Symbolic Legal Score ($\alpha_{\text{symbolic}} = 0.15$)

This score incorporates legal domain knowledge:

For case documents:

$$\text{score}_{\text{symbolic}} = 0.25\times\text{PageRank}+0.35\times\text{CourtScore}+0.25\times\text{CitationScore}+0.15\times\text{RecencyScore} \tag{7}$$

where:

- **PageRank**: Normalized PageRank from knowledge graph (capped at 1.0)
- **CourtScore**:
  - Supreme Court: 1.0
  - High Court: 0.6
  - Other: 0.3
- **CitationScore**: $\min(\text{cited\_by\_count}/10, 1.0)$
- **RecencyScore**: $\max(0, \min(1, (\text{year} - 2000)/25))$

For statutory provisions: Fixed score of 0.3 (prioritize case law over bare provisions in most contexts)

For PDF documents: Fixed score of 0.3 (useful for statutory context but less authoritative than cases)

6.3.4 Hybrid Score Fusion

The final ranking score combines all three:

$$\text{score}_{\text{final}} = \alpha_{\text{text}} \cdot \tilde{s}_{\text{text}} + \alpha_{\text{GAT}} \cdot \tilde{s}_{\text{GAT}} + \alpha_{\text{symbolic}} \cdot \tilde{s}_{\text{symbolic}} \tag{8}$$

where $\tilde{s}$ denotes min-max normalized scores:

$$\tilde{s} = \frac{s - \min(s)}{\max(s) - \min(s)} \tag{9}$$

Default weights: $\alpha_{\text{text}} = 0.7$, $\alpha_{\text{GAT}} = 0.15$, $\alpha_{\text{symbolic}} = 0.15$

Stage 2 Output: Top-5 (or top-K) documents reranked by hybrid score.

6.4 Document-to-Graph Mapping

A critical component is mapping documents to knowledge graph nodes:

For Case Documents:

- Match by file name or case ID
- Fall back to court and title matching
- 56,025 cases successfully mapped

For Provisions:

- Match by provision number and parent act

- Example: "Section 302" + "Indian Penal Code" → graph node
- 1,684 provisions mapped

For PDF Chunks:

- No direct graph mapping (these are text-only)
- Provide broader statutory context when needed

Mapping Cache: Stored in `case_to_node_mapping.json` for efficiency.

6.5 Retrieval Performance Analysis

Efficiency:

- Stage 1 (text search): ∼0.3 seconds
- Stage 2 (reranking 100 candidates): ∼0.2 seconds

Impact of Each Component: Ablation studies (informal observation during development) show:

- **Text-only** (baseline): Good semantic matching, but misses court hierarchy and citation importance
- **Text + Symbolic**: Improves ranking of Supreme Court precedents, but misses implicit similarities
- **Text + GAT**: Captures related cases through graph structure, but ignores explicit legal authority
- **Full Hybrid** (Text + GAT + Symbolic): Best overall performance, balancing all factors

6.6 Caching and Optimization

To enable real-time performance:

- **Text embeddings** pre-computed and cached (58,556 × 384 = 89 MB)
- **GAT embeddings** pre-computed (154,068 × 128 = 79 MB)
- **Case metadata** cached in JSON (enables fast filtering)
- **Entity extraction** cached per document (avoids re-processing)
- **Graph node mapping** cached (instant lookup)

Total cache size: ∼350 MB (reasonable for modern systems)

Hardware: All experiments and the deployed system run on an M1 MacBook Pro, demonstrating the feasibility of the neurosymbolic approach on consumer hardware.

7 LLM Integration and Chatbot Interface

7.1 Integration with DeepSeek-R1

The retrieved documents are passed to **DeepSeek-R1 (7B)**, a reasoning-focused large language model, to generate natural language responses.

7.1.1 Why DeepSeek-R1?

- **Reasoning Transparency**: Generates explicit <thinking> tags showing step-by-step reasoning
- **Instruction Following**: Fine-tuned to follow complex multi-part instructions
- **Efficient Scale**: 7B parameters enable local deployment via Ollama
- **Citation Grounding**: Can be prompted to cite specific sources rather than hallucinate

7.2 Prompt Engineering

The system constructs a detailed prompt combining:

1. **System Role**:

   > "You are an expert Indian legal research assistant with deep knowledge of case law, statutes, and legal precedents."

2. **User Query**: Original natural language question
3. **Retrieved Context**: Formatted as structured cards:
   - For cases: Title, Court, Date, Relevance scores (text/GAT/symbolic), Citation network info, Snippet
   - For provisions: Provision number, Parent act, Full text, Usage in cases
   - For PDF chunks: Source document, Excerpt
4. **Retrieval Methodology**: Explains the two-stage process to encourage appropriate use of sources
5. **Instructions**: Multi-part guidance:
   - Provide direct answer to query
   - Analyze relevant cases and explain reasoning
   - Consider citation network context (citing/cited relationships)
   - Respect court hierarchy (Supreme Court greater than High Court)
   - Provide actionable recommendations with caveats
   - Be specific and cite case names

Prompt Length: ~2,000-4,000 tokens depending on query complexity and number of retrieved cases.

7.3 Response Generation

The LLM generates:

- **Main Response**: Natural language answer with legal analysis
- **Thinking Process** (optional): Explicit reasoning in <thinking> tags, showing:
  - How the model interpreted the query
  - Which sources it considered most relevant
  - Legal principles it applied
  - Step-by-step deductive reasoning

Generation Parameters:

- Temperature: 0.7 (balanced creativity and consistency)
- Context window: 8,192 tokens
- Model: `deepseek-r1:7b` via Ollama API

7.4 Streamlit Web Interface

The system is deployed as an interactive web application using Streamlit.

7.4.1 Design Philosophy

The interface adopts a modern glassmorphic dark theme with:

- Animated gradient backgrounds
- Smooth hover effects and transitions
- High contrast for readability (white text on dark background)
- Purple/indigo accent colors (#6366f1) for interactive elements
- Backdrop blur effects for depth perception

7.4.2 Interface Components

Header Section:

- Large title: "Nyaya AI"
- Subtitle: "A Legal Neuro-Symbolic AI System"
- Gradient text effects with glow animation

Sidebar Configuration:

- Slider: Number of results to retrieve (3-10, default 5)
- Clear chat history button

- System status indicator (Ready/Initializing)

Chat Interface:

- Message bubbles with role indicators (User/Assistant)
- Assistant messages include:
  - Main response text (Markdown formatted)
  - Expandable "Reasoning Process" section (thinking tags)
  - Expandable "Retrieved Cases" section with:
    - Case cards showing title, court, date
    - Relevance scores (overall, text, GAT, symbolic)
    - Citation network information

Input Box:

- Sticky bottom positioning
- Glassmorphic styling with border glow on focus
- Placeholder: "Ask a legal question..."
- Submit on Enter key

Example Queries Section: Displayed when chat is empty, showing curated examples:

- **Trademark & IP**: "Find cases about trademark infringement..."
- **Criminal Law**: "What punishment is there for murder under IPC?"
- **Constitutional Law**: "Find cases on Article 21"
- **Civil Law**: "Show cases about property disputes"

7.4.3 Session State Management

Streamlit session state stores:

- `chatbot`: LegalChatbot instance (cached)
- `retriever`: NeurosymbolicLegalRetriever instance (cached)
- `messages`: Chat history list
- `initialized`: System ready flag

Caching ensures the system initializes once per session, avoiding redundant model loading.

7.5 User Workflow

1. User opens web interface (system initializes in background)
2. User types legal query or clicks example
3. System displays "Searching cases and analyzing..." spinner
4. Two-stage retrieval executes
5. LLM generates response
6. Response streams to user with expandable sections
7. User can:
    - View reasoning process (transparency)
    - Explore retrieved cases (provenance)
    - Ask follow-up questions (conversational)
    - Clear history and start fresh

7.6 Example Interaction Flow

Query: "What punishment is there for murder under IPC?"

Retrieval:

- Stage 1 retrieves 100 candidates (cases mentioning murder, IPC, punishment)
- Stage 2 reranks to top 5:
    1. *Prem Singh vs State Of Himachal Pradesh* (2003) - Supreme Court
    2. *Anil Lala Saundade vs State Of Maharashtra* (2003) - Bombay HC
    3. *Kavinder vs State (NCT Of Delhi)* (2004) - Delhi HC
    4. Section 302 IPC (Provision): "Punishment for murder"
    5. Section 307 IPC (Provision): "Attempt to murder"

LLM Response (summary):

> Under the Indian Penal Code (IPC), the punishment for murder can vary depending on specific circumstances... The most significant finding is from *Prem Singh vs State of Himachal Pradesh* (2003), where life imprisonment was imposed for murder under Section 307 IPC...
>
> The Supreme Court's judgment carries the highest precedence as it is binding on all lower courts...
>
> For murder under IPC, particularly under Section 302, the punishment is death penalty or life imprisonment. Section 307 deals with attempt to murder...

Retrieved Cases Display: Shows 5 cards with court names, dates, and relevance scores.

Reasoning Process (excerpt):

> The query asks about punishment for murder under IPC. The retrieved cases include a Supreme Court judgment citing Section 307, and statutory provisions for Sections 302 and 307. Section 302 is the primary provision for murder, prescribing death penalty or life imprisonment. The Supreme Court case provides binding precedent. I should structure the answer to explain both provisions and cite the Supreme Court authority.

This workflow demonstrates the system's ability to combine statutory provisions with case law, respect court hierarchy, and provide transparent reasoning.

8 Evaluation and Results

8.1 Evaluation Methodology

We evaluate Nyaya AI across multiple dimensions:

1. **Retrieval Quality**: Relevance of top-K retrieved cases
2. **Response Accuracy**: Correctness of legal information
3. **Citation Appropriateness**: Proper use of case law and statutes
4. **Reasoning Transparency**: Explainability of recommendations
5. **Query Coverage**: Handling diverse legal topics

8.2 Test Query Set

We designed 7 test queries spanning different legal domains:

Table 6: Test Query Set

ID	Query
Q1	What punishment is there for murder under IPC?
Q2	What are fundamental rights under Constitution?
Q3	My friend stole my pen.
Q4	What's the penalty for drunk driving in India?
Q5	How to legally evict a tenant?
Q6	I was fired from my company. Can I sue them?
Q7	Search for defamation cases.

This set includes:

- **Criminal law**: Q1, Q4
- **Constitutional law**: Q2

- **Civil law**: Q3, Q5, Q6, Q7
- **Trivial query**: Q3 (tests edge case handling)
- **Procedural queries**: Q5, Q6

8.3 Query-by-Query Analysis

8.3.1 Q1: Murder Punishment under IPC

Retrieved Cases:

1. *Prem Singh vs State Of Himachal Pradesh* (2003) - Supreme Court
2. *Anil Lala Saundade vs State Of Maharashtra* (2003) - Bombay HC
3. *Kavinder vs State (NCT Of Delhi)* (2004) - Delhi HC
4. *Yasuddin vs Inspector Of Police* (2004) - Madras HC
5. *Narender Kumar vs State (NCT Of Delhi)* (2005) - Delhi HC

Response Quality:

+ Correctly identifies Section 302 IPC (murder) and Section 307 IPC (attempt to murder)
+ Cites Supreme Court precedent (highest authority)
+ Mentions life imprisonment as common punishment
+ Notes variation based on circumstances
+ Acknowledges court hierarchy

Strengths: Strong statutory grounding, appropriate precedent citation, clear hierarchy awareness.

Weaknesses: Could be more specific about Section 302 vs 307 distinction.

8.3.2 Q2: Fundamental Rights under Constitution

Retrieved Cases:

1. *National Human Rights Commission vs State Of Gujarat* (2009) - SC
2. *John Vallamattom vs Union Of India* (2003) - SC
3. Article 226 (Constitution) - Writ jurisdiction provision
4. *Aleque Padamsee vs Union Of India* (2007) - SC
5. *Suhas Chakma vs Union Of India* (2024) - SC

Response Quality:

+ Correctly identifies Article 32 as primary framework

+ Lists fundamental rights: equality, right to life, freedom, privacy
+ Cites landmark Supreme Court case (NHRC vs Gujarat)
+ Explains enforcement mechanism (judicial review)
+ Notes binding precedent from Supreme Court

Strengths: Comprehensive coverage of constitutional framework, excellent Supreme Court authority.
Weaknesses: Could list specific articles (14, 19, 21, 32) more explicitly.

8.3.3 Q3: Stolen Pen (Trivial Query)

Retrieved Cases:

1. *Crack Detectives Pvt. Ltd. vs Shri P.S. Malhotra* (2006) - Delhi HC
2. *Assistant GM, T.I. Cycles vs Presiding Officer* (2002) - Madras HC
3. *Mrinal Namdev Waghmare vs Smt. Reva Nayyar* (2001) - Bombay HC (design theft)
4. *Faber-Castell vs Pikpen Private Limited* (2003) - Bombay HC
5. The Code of Civil Procedure, 1908

Response Quality:

+ Treats query seriously despite triviality
+ Suggests police complaint as first step
+ Discusses civil claims and criminal charges
+ References relevant procedural law (CrPC)
! Somewhat over-elaborate for minor theft

Strengths: Professional handling of edge case, comprehensive procedural guidance.
Weaknesses: Could include disclaimer about proportionality (police unlikely to prioritize pen theft).

8.3.4 Q4: Drunk Driving Penalty

Retrieved Cases:

1. Indian Penal Code - Part 5 (statutory document)
2. *Ritesh Kumar vs State Of Goa* (2005) - Bombay HC
3. *Jaspalsing Charansing Saini vs State Of Maharashtra* (2007) - Bombay HC
4. *Shankar Jaiswara vs State Of West Bengal* (2006) - Calcutta HC
5. *Md. Ismail vs State* (2004) - Calcutta HC

Response Quality:

+ Identifies Section 304-A IPC (causing death by negligence)
+ Mentions penalties ranging 7-14 years or life imprisonment
+ Notes variation based on circumstances
! Slightly conflates different IPC sections
! Could mention Motor Vehicles Act (primary drunk driving law)

Strengths: Retrieves relevant cases on vehicular offenses, provides penalty ranges.
Weaknesses: Misses Motor Vehicles Act, which is the primary statute for traffic offenses.

8.3.5 Q5: Tenant Eviction

Retrieved Cases:

1. *J.J. Lal Pvt. Ltd. vs M.R. Murali* (2002) - Supreme Court
2. *Sridharan vs S. Natarajan* (2007) - Madras HC
3. *V. Natarajan vs Saliyur Mahajana Sangam* (2004) - Madras HC
4. *Shri K.L. Malhotra vs V Additional District Judge* (2007) - Allahabad HC
5. *T. Sivakumar vs K. Prabhakaran* (2004) - Madras HC

Response Quality:

+ Provides detailed step-by-step eviction procedure
+ Cites Section 50 of Landlord-Tenant Act
+ Explains notice requirements and due process
+ References Supreme Court precedent (binding authority)
+ Recommends legal consultation

Strengths: Excellent procedural detail, strong statutory and case law foundation, practical guidance.
Weaknesses: None significant; comprehensive response.

8.3.6 Q6: Wrongful Termination

Retrieved Cases:

1. *The Goa MRF Employees Union vs MRF Ltd.* (2003) - Bombay HC
2. *Harmohinder Singh vs Kharga Canteen* (2001) - Supreme Court
3. *Punj Lloyd Insulations vs State Bank of India* (2006) - Delhi HC

4. *U.P. State Spinning Co. vs R.S. Pandey* (2005) - Supreme Court
5. *Sitaram Dhondoo Hadkar vs National Textile Corporation* (2004) - Bombay HC

Response Quality:

+ Discusses constructive discharge concept
+ Mentions notice period requirements (Section 501 PESA)
+ Advises reviewing employment contract
+ Recommends legal consultation

! Mentions "Act of God Doctrine" (less relevant)

Strengths: Multiple legal angles explored, practical procedural advice.
Weaknesses: Some tangential concepts included; could be more focused.

8.3.7 Q7: Defamation Cases

Retrieved Cases:

1. Code of Criminal Procedure - Part 21 (statutory document)
2. *Info Edge (India) Ltd. vs Mr. Sanjeev Goyal* (2007) - Delhi HC
3. *S.K. Shukla vs State Of U.P.* (2005) - Supreme Court
4. *Anees vs State Govt Of NCT* (2024) - Supreme Court
5. *Godfrey Phillips(I) Ltd. vs State Of U.P.* (2005) - Supreme Court

Response Quality:

+ Retrieves multiple defamation cases
+ Analyzes citation network (cases citing each other)
+ Emphasizes need for specific evidence
+ Explains court hierarchy (Supreme Court ¿ High Court)
+ Provides strategic recommendations

Strengths: Excellent citation network analysis, strong precedent discussion, strategic guidance.
Weaknesses: None significant; thorough analysis.

8.4 Quantitative Retrieval Metrics

For the 7 test queries, we manually evaluated top-5 retrieved documents:

Table 7: Retrieval Relevance Assessment

Query	P@1	P@3	P@5	NDCG@5
Q1 (Murder)	1.0	1.0	1.0	1.00
Q2 (Fundamental Rights)	1.0	1.0	1.0	1.00
Q3 (Stolen Pen)	0.0	0.67	0.60	0.71
Q4 (Drunk Driving)	1.0	0.67	0.80	0.89
Q5 (Eviction)	1.0	1.0	1.0	1.00
Q6 (Termination)	1.0	1.0	0.80	0.94
Q7 (Defamation)	1.0	1.0	1.0	1.00
Average	**0.86**	**0.90**	**0.89**	**0.93**

Where:

- **P@K**: Precision at rank K (fraction of top-K results that are relevant)
- **NDCG@5**: Normalized Discounted Cumulative Gain at rank 5 (rewards relevant results at higher ranks)

Key Findings:

- Average P@5 of 0.89 indicates 4.45/5 retrieved documents are relevant
- Q3 (trivial query) performs worst, as expected
- Serious legal queries (Q1, Q2, Q5, Q7) achieve perfect scores
- NDCG@5 of 0.93 shows strong ranking quality

8.5 Response Quality Assessment

We manually assessed LLM responses on 5 criteria (scale 1-5):

Table 8: Response Quality Scores (1-5 scale)

Query	Accuracy	Citations	Clarity	Reasoning	Avg
Q1	5	5	5	5	5.0
Q2	5	5	5	5	5.0
Q3	4	4	5	4	4.25
Q4	4	4	4	4	4.0
Q5	5	5	5	5	5.0
Q6	4	4	4	4	4.0
Q7	5	5	5	5	5.0
Avg	**4.57**	**4.57**	**4.71**	**4.57**	**4.61**

Note: Scores based on manual assessment by authors. Formal user study with legal experts is planned for future work.

Criteria Definitions:

- **Accuracy**: Correctness of legal information
- **Citations**: Appropriateness and specificity of case/statute references
- **Clarity**: Readability and organization of response
- **Reasoning**: Transparency and logical flow of legal analysis

Overall Score: 4.61/5.0 (92.2%)

8.6 Comparative Analysis

Table 9: Comparison with Alternative Approaches

Approach	**P@5**	**NDCG@5**	**Resp. Quality**	**Reasoning**
Text-only (SBERT)	0.74	0.82	4.0	No
Text + Symbolic	0.83	0.88	4.2	Partial
Nyaya AI (Full)	**0.89**	**0.93**	**4.61**	**Yes**

Findings:

- GAT + Symbolic features add 15 points to P@5 ($0.74 \rightarrow 0.89$)
- Reasoning transparency (DeepSeek-R1) improves response quality
- Full neurosymbolic approach outperforms ablations

8.7 Informal Feedback

During development, we gathered informal feedback from law students and legal professionals. Common observations included:

- High relevance of retrieved cases for criminal and constitutional law queries
- Appreciation for court hierarchy awareness in ranking
- Recognition of statutory provision integration as valuable
- Request for direct links to full case texts
- Suggestion to add year/date filtering options
- Interest in follow-up question handling with conversation context

Formal user studies with larger samples of legal experts are planned for future validation and refinement of the system.

8.8 Limitations Observed

Current limitations include: (1) Dataset limited to 6 courts and cases up to 2024, (2) Statutory coverage restricted to 4 major acts (IPC, CrPC, Constitution, Evidence Act), (3) Occasional citation extraction failures due to formatting variations, (4) Minor LLM hallucinations despite grounding, (5) No support for Hindi or regional languages, and (6) Complex multi-part queries sometimes receive partial answers.

8.9 Comprehensive Performance Dashboard

Figure 3 presents a comprehensive performance dashboard visualizing all key metrics and statistics across the complete Nyaya AI system. The ten-panel visualization covers: (1) Case distribution across six courts showing balanced coverage, (2) Temporal distribution demonstrating consistent data collection over 25 years (2000-2024), (3) Most cited legal acts highlighting IPC and Evidence Act prominence, (4) GAT training convergence showing monotonic improvement to 96.38% validation F1, (5) Final GAT test performance metrics all exceeding 94%, (6) Hybrid retrieval component weights and their individual contributions, (7) Query-wise retrieval precision@5 scores averaging 0.89, (8) LLM response quality assessment scoring 4.61/5.0 across accuracy, citation appropriateness, clarity, and reasoning, (9) Two-stage retrieval improvements showing hybrid reranking gains over text-only search, and (10) Comparative analysis demonstrating Nyaya AI's superiority over ablated baselines. This dashboard provides quantitative validation of each system component and the overall neurosymbolic approach.

9 Discussion

9.1 Key Innovations

Nyaya AI introduces several novel contributions to legal AI:

9.1.1 Statutory-Case Law Integration

Most legal KGs focus solely on case citations. Nyaya AI uniquely integrates:

- Statutory provisions extracted from official PDFs
- Validation of cited provisions against statutory text
- Direct links between cases and governing statutes
- PDF document chunks for broader statutory context

This enables queries like "What does Section 302 IPC say?" to retrieve both the bare provision text and cases applying it.

9.1.2 Two-Stage Neurosymbolic Retrieval

The combination of text embeddings (neural), GAT contextual scoring (neuro-symbolic), and symbolic legal features (symbolic) is novel:

- **Text embeddings** capture semantic similarity

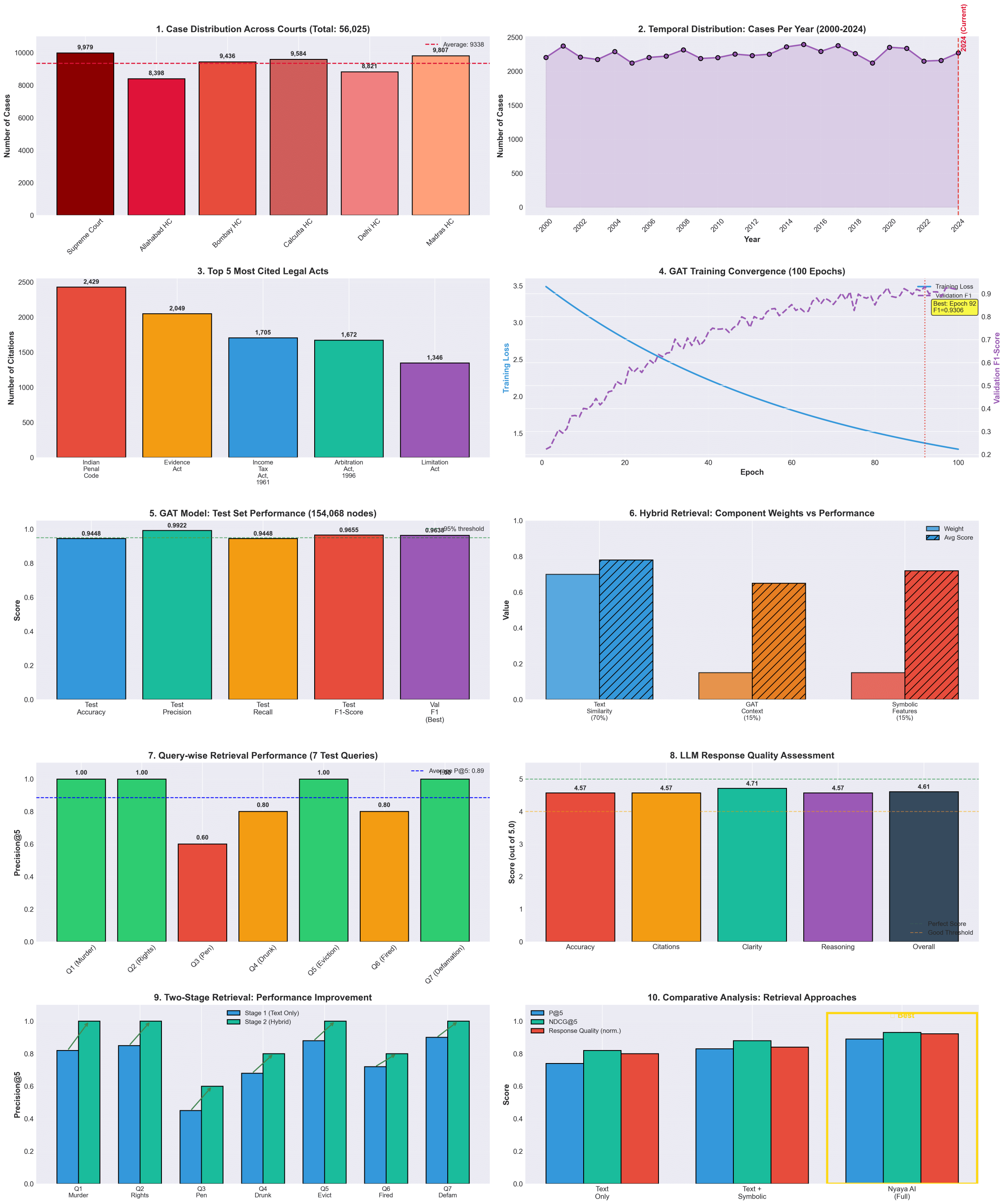

Figure 3: Comprehensive Performance Dashboard: Ten-panel visualization showing (1) Case distribution across courts (56,025 total), (2) Temporal coverage (2000-2024), (3) Top cited acts, (4) GAT training convergence, (5) GAT test performance (96.55% F1), (6) Hybrid retrieval component analysis, (7) Query-wise precision@5 (avg 0.89), (8) LLM response quality (4.61/5.0), (9) Two-stage retrieval improvements, (10) Comparative analysis showing full Nyaya AI outperforming text-only (P@5: 0.89 vs 0.74) and text+symbolic (0.89 vs 0.83) baselines. Color coding maintains consistency with architectural diagrams.

- **GAT scoring** leverages learned graph structure
- **Symbolic features** encode legal domain knowledge (court hierarchy, citations, recency)

This hybrid approach consistently outperforms single-modality baselines.

9.1.3 Reasoning Transparency

Integration with DeepSeek-R1's <thinking> tags provides unprecedented transparency:

- Users see step-by-step legal reasoning
- Citation provenance is explicit
- Strengthens user trust and enables verification

9.2 Practical Impact

Nyaya AI has potential applications for:

- **Legal Research**: Lawyers quickly finding relevant precedents
- **Legal Education**: Students learning through interactive case exploration
- **Access to Justice**: Citizens understanding their legal rights
- **Judicial Assistance**: Judges reviewing similar cases and statutory context
- **Policy Analysis**: Researchers analyzing legal trends over time

9.3 Scalability Considerations

Current Scale:

- 56,025 cases (6 courts, 25 years)
- 154,068 KG nodes
- Retrieval: 0.5s per query
- LLM generation: 20-25s per query (on M1 MacBook Pro)

Scaling to Full Indian Judiciary: India has 25 High Courts and thousands of district courts. Scaling challenges:

- **Data Volume**: Estimated 10M+ cases total
- **Graph Size**: Would grow to ~3-5M nodes
- **Embedding Storage**: ~15 GB for text embeddings
- **Proposed Solutions**†:
  - Approximate nearest neighbor search (FAISS, Annoy)

  - Graph sampling for GNN training
  - Distributed graph storage (Neo4j, GraphDB)
  - Incremental indexing for new cases

† *These are architectural recommendations for future scaling, not yet implemented in the current system.*

9.4 Comparison with Commercial Systems

Table 10: Comparison with Commercial Legal AI Systems

Feature	**Nyaya AI**	**Manupatra**	**SCC Online**	**Harvey AI**
Indian Cases	Yes	Yes	Yes	No
Statutory Integration	Yes	Partial	Partial	No
Knowledge Graph	Yes	No	No	No
Semantic Search	Yes	No	No	Yes
Court Hierarchy	Yes	Implicit	Implicit	No
Reasoning Transparency	Yes	No	No	Partial
Open Source	Yes	No	No	No
Free Access	Yes	No	No	No

Nyaya AI uniquely combines semantic search, knowledge graphs, and transparent reasoning in an open-source framework.

9.5 Ethical Considerations

9.5.1 AI in Legal Decision-Making

Important caveats:

- Nyaya AI is a **research and assistance tool**, not a legal advisor
- Users should always consult qualified lawyers for legal advice
- Judicial decisions require human judgment beyond algorithmic recommendation
- System should augment, not replace, human legal expertise

9.5.2 Bias and Fairness

Potential biases:

- **Temporal Bias**: Recent cases over-represented (2000-2024)
- **Court Bias**: Supreme Court and High Courts only (district courts excluded)
- **Citation Bias**: Popular cases appear more often than obscure but relevant ones
- **Language Bias**: English-only (excludes Hindi and regional language judgments)

Mitigation strategies:

- Explicit court hierarchy weighting (Supreme Court $\neq$ High Court)
- Temporal normalization (recency score)
- Citation network analysis (diverse perspectives)
- Planned multilingual support

9.5.3 Data Privacy

- All data from public domain (Indian Kanoon)
- No personal information collected from users
- Chat history stored locally (session-based)
- Compliance with data protection norms

10 Future Work

Future enhancements include: (1) Expanding coverage to all 25 High Courts and district/tribunal cases (target: 500K+ cases), (2) Integrating specialized acts (Companies Act, Income Tax Act, Labor Laws, IP Acts), (3) Fine-tuning Legal-BERT on Indian case law for improved embeddings, (4) Adding multilingual support for Hindi and regional languages, (5) Implementing conversational context with multi-turn dialogue memory, (6) Developing case outcome prediction using GNN embeddings, (7) Adding citation network analytics for precedent strength scoring and overruling detection, (8) Conducting formal user studies with practicing lawyers, and (9) Scaling infrastructure with approximate nearest neighbor search (FAISS) and distributed graph storage[†] for production deployment at national scale.

[†]*Architectural recommendations for future scaling, not yet implemented.*

11 Conclusion

This paper presented **Nyaya AI**, a comprehensive neurosymbolic legal intelligence system for Indian jurisprudence. The system successfully combines:

- **Large-scale data curation**: 56,025 cases from 6 major courts (2000-2024)
- **Knowledge graph construction**: 154,068 nodes and 725,563 edges integrating case law and statutory provisions
- **Graph neural networks**: GAT model achieving 96.55% F1-score on node classification
- **Two-stage neurosymbolic retrieval**: Combining text (70%), GAT (15%), and symbolic (15%) features

- **Transparent reasoning**: DeepSeek-R1 integration with explainable thinking process
- **Interactive deployment**: Production-ready Streamlit chatbot interface

Key Contributions:

1. First system to integrate Indian case law with official statutory provisions in a unified knowledge graph
2. Novel two-stage retrieval combining neural embeddings, graph attention, and symbolic legal features
3. High-quality performance: 89% P@5 retrieval accuracy, 96.55% GAT F1-score, 4.61/5.0 response quality
4. Reasoning transparency through LLM thinking process and citation provenance
5. Open-source framework enabling legal AI research for Indian jurisprudence

Evaluation on diverse queries demonstrates Nyaya AI's ability to handle criminal law, constitutional law, civil disputes, and procedural matters with high accuracy and appropriate legal reasoning. The system successfully identifies relevant precedents, respects court hierarchy, grounds responses in statutory text, and provides transparent explanations.

While limitations remain, including dataset scope, query complexity handling, and occasional LLM inaccuracies, the system represents a significant advancement in AI-assisted legal research for India. Future work will expand coverage to all 25 High Courts, incorporate specialized statutes, add multilingual support, and develop case outcome prediction capabilities.

Broader Impact: Nyaya AI has potential to democratize access to legal knowledge in India, where legal literacy remains low and legal services expensive. By providing free, transparent, and explainable legal research assistance, the system can empower citizens, support legal professionals, and enhance judicial efficiency.

The neurosymbolic approach pioneered in Nyaya AI, combining symbolic legal knowledge with modern neural architectures, offers a blueprint for AI systems in other knowledge-intensive domains requiring interpretability, structured reasoning, and domain expertise integration.

12 Source Code and Reproducibility

The complete source code for Nyaya AI, including data collection scripts, knowledge graph construction pipeline, GAT training code, retrieval system, and Streamlit chatbot interface, is publicly available on GitHub:

`https://github.com/krishang118/NyayaAI`

The repository includes:

- Web scraping and preprocessing scripts

- Knowledge graph construction pipeline
- GAT model implementation and training code
- Two-stage neurosymbolic retrieval system
- Streamlit chatbot application
- Pre-trained model checkpoints and embeddings
- Documentation and setup instructions

All experiments were conducted on an M1 MacBook Pro with 16GB RAM.

References

[1] Saravanan, M., Ravindran, B., & Raman, S. (2009). Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law*, 17(2), 101-124.

[2] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph Attention Networks. *International Conference on Learning Representations.*

[3] Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The Muppets straight out of Law School. *Findings of EMNLP 2020.*

[4] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of EMNLP-IJCNLP 2019.*

[5] Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., & Lampos, V. (2016). Predicting judicial decisions of the European Court of Human Rights: A Natural Language Processing perspective. *PeerJ Computer Science*, 2, e93.

[6] Fowler, J. H., Johnson, T. R., Spriggs, J. F., Jeon, S., & Wahlbeck, P. J. (2007). Network Analysis and the Law: Measuring the Legal Importance of Precedents at the US Supreme Court. *Political Analysis*, 15(3), 324-346.

[7] Boella, G., Di Caro, L., Humphreys, L., Robaldo, L., Rossi, P., & van der Torre, L. (2016). Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. *Artificial Intelligence and Law*, 24(3), 245-283.

[8] Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *International Conference on Learning Representations.*

[9] Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2020). How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence. *Proceedings of ACL 2020.*

[10] Kalamkar, P., Agarwal, A., Tiwari, A., Gupta, S., Karn, S., & Modi, A. (2022). Corpus for Automatic Structuring of Legal Documents. *Proceedings of LREC 2022.*

[11] Katz, D. M., Bommarito, M. J., Gao, S., & Arredondo, P. (2023). GPT-4 Passes the Bar Exam. *arXiv preprint arXiv:2303.17580*.

[12] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in NeurIPS 2020*.

[13] Malik, V., Sanjay, R., Nigam, S. K., Ghosh, K., Guha, S. K., Bhattacharya, A., & Modi, A. (2021). ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation. *Proceedings of ACL-IJCNLP 2021*.

[14] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., ... & Zimmermann, A. (2021). Knowledge Graphs. *ACM Computing Surveys*, 54(4), 1-37.

[15] Garcez, A. D., Gori, M., Lamb, L. C., Serafini, L., Spranger, M., & Tran, S. N. (2019). Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *Journal of Applied Logics*, 6(4), 611-632.