

# ToC: Tree-of-Claims Search with Multi-Agent Language Models

Shuyang Yu<sup>1\*</sup>, Jianan Liang<sup>2</sup>, Hui Hu<sup>2</sup>

<sup>1</sup>Columbia University

<sup>2</sup>AIGROW

sy3309@columbia.edu, liangjianan@aigrow.cn, huhui@aigrow.cn

## Abstract

Optimizing patent claims is a critical yet challenging task, demanding careful balance between maximizing novelty and preserving legal scope. Manual claim drafting is labor-intensive, costly, and inherently inconsistent, while conventional Large Language Models (LLMs) often lack the structured, iterative reasoning essential for precise claim refinement. To address these challenges, we introduce Tree of Claims (ToC), an innovative framework that redefines claim editing as a guided search problem. ToC synergistically integrates Monte Carlo Tree Search (MCTS) with a collaborative multi-agent system, comprising an LLM-based EditorAgent that proposes contextually grounded edits, and an ExaminerAgent that mimics patent examiner critiques through structured, chain-of-thought analyses of novelty and prior art disclosure. Driven by a carefully designed multi-objective reward function, ToC jointly optimizes novelty, scope retention, and semantic coherence. Experimental evaluation on a benchmark of 1145 claims demonstrates that ToC significantly outperforms standard LLMs in zero-shot and few-shot scenarios, achieving an average composite score improvement of 8%, and up to 9% in certain cases. Extensive experiments, including detailed ablation studies, validate ToC’s efficacy in generating superior, legally robust claim revisions. Overall, ToC establishes a transparent, controllable, and interpretable methodology that effectively bridges advanced LLM reasoning capabilities with strategic MCTS planning for structured patent claim optimization.

**Code** — <https://github.com/ysy2003/ToC>

## 1 Introduction

Drafting and revising patent claims is crucial, significantly influencing the legal scope, technical breadth, and commercial potential of intellectual property (Son et al. 2022). When addressing examiner rejections or seeking to optimize claim language for stronger protection, practitioners must navigate complex demands: ensuring precise legal wording, maintaining technical coverage, and maximizing novelty over prior art (Paul 2024). Conventionally, these tasks are executed by experienced patent professionals who rely heavily on domain expertise, iterative refinement, and sophis-

ticated understanding of examiner criteria. However, manual processes are resource-intensive, costly, and inherently inconsistent (Wang, Mudhiganti, and Sharma 2024). Consequently, a critical question emerges: can we construct a systematic, controllable, and transparent approach that effectively integrates the generative capabilities of large language models (LLMs) with structured planning methods to automate and enhance patent claim drafting?

Building on notable progress in pre-trained models, including foundation models (Fang et al. 2025b) and LLMs (Fang et al. 2025a), across domains such as medicine, law, and education, recent advancements have increasingly enabled the automation of patent-related tasks. Nevertheless, existing LLM-driven solutions predominantly operate under single-shot or few-shot paradigms, lacking the iterative and structured reasoning crucial for high-quality claim engineering (Ren and Ma 2024). These approaches often inadequately balance the intricate interplay among legal scope, technical specificity, and linguistic precision, thus restricting their effectiveness in practical scenarios.

Addressing these critical shortcomings, we introduce the Tree of Claims (ToC), an innovative framework that reconceptualizes patent claim editing as a structured search process. ToC integrates Monte Carlo Tree Search (MCTS) with a sophisticated multi-agent collaboration mechanism. Specifically, an LLM-based EditorAgent proposes contextually accurate, legally sound edits, while an ExaminerAgent simulates detailed patent examiner scrutiny, providing structured assessments of novelty and disclosure.

The ToC framework employs an uncertainty-aware MCTS architecture, wherein each node represents an ordered sequence of discrete editing operations. Optimization is guided by a carefully crafted multi-objective reward function, considering novelty enhancement, scope preservation, and semantic coherence. Additional techniques such as uncertainty gating and progressive widening ensure claim improvements remain both interpretable and verifiable. The core contributions of this paper are:

- A novel formulation of patent claim optimization as a structured search problem within the innovative ToC framework, integrating multi-agent collaborative reasoning with MCTS.
- A collaborative multi-agent architecture where an EditorAgent generates legally robust edit operations, and an

\*Work done during an internship at AIGROW.

ExaminerAgent provides systematic assessments, jointly guiding a targeted, multi-objective search process.

- Extensive experiments demonstrating that ToC significantly outperforms state-of-the-art LLM-based baselines (including zero-shot and few-shot settings) in novelty, scope, and legal robustness across real-world patent revision tasks.

## 2 Related Work

Patent claim optimization lies at the confluence of structured text editing, legal-domain reasoning, and sequential decision-making. We review related work across three core areas: LLMs for patent drafting and revision, Multi-agent systems for structured reasoning and Search-based text optimization.

**LLMs for Patent Drafting and Revision.** LLMs like GPT-4 are increasingly used in patent tasks, from claim drafting to responding to office actions. However, these black-box models offer limited control over output structure, scope, or legal consistency (Li et al. 2024). Their one-shot drafts often require substantial human revision, failing to bridge the gap between technical details and legal coverage. While domain-specific fine-tuning improves fluency, it doesn’t solve the core controllability issue (Bai et al. 2024b). Even top LLMs necessitate iterative human refinement for full legal robustness, and recent benchmarks highlight persistent gaps in claim quality (Jiang et al. 2024). Legal-theory work also warns that generative AI can inadvertently expand claim scope without proper safeguards (Wang 2024).

Researchers are now pursuing controllable, feedback-driven approaches. ClaimBrush (Kawano, Nonaka, and Yoshino 2024) incorporates examiner feedback but remains largely a one-pass revision. Other iterative methods, like AutoPatent (Wang et al. 2024b) and EvoPat (Wang et al. 2024c), use multi-agent frameworks, but their editing operations are often opaque and non-deterministic. This lack of transparency hinders real-world integration, where practitioners need to verify every change. To address this, the ToC framework provides systematic, stepwise claim edits with precise control and multi-criteria feedback. Each modification in ToC is explicit and justified, ensuring traceable revisions and verifiable legal scope at every step—crucial for high-stakes patent applications.

**Multi-Agent Systems for Structured Reasoning.** Multi-agent LLM frameworks are powerful for complex tasks, enabling specialized agents to collaborate and critique each other for more coherent results. Examples include MetaGPT (Hong et al. 2023), AutoGen (Wu et al. 2024), OpenAgents (Xie et al. 2023), LongWriter (Bai et al. 2024a), and PARIS-style collaborative systems for patent responses (Chu et al. 2024). These structured approaches enhance LLM reasoning through iterative self-correction.

However, most existing multi-agent systems are for open-ended tasks without strict domain constraints. Patent claim editing, in contrast, demands fine-grained control and strict adherence to legal criteria (Bui 2025). ToC excels here by embedding these constraints directly into its agent roles: an

EditorAgent proposes controlled, atomic, legally-grounded edits, while an ExaminerAgent provides structured, fine-grained feedback on each edit. This interactive critique loop ensures a transparent, justified revision history. Crucially, ToC’s agents operate under strict domain-specific objectives, explicitly checking compliance with patent rules. Unlike prior systems that yield only a final text, ToC generates a complete, reasoned edit history, providing an interpretable and auditable chain of revisions vital for legal practitioners.

**Search-Based Text Optimization.** Search algorithms offer a powerful way to enhance generative LLMs, especially for tasks requiring detailed planning or adherence to multiple constraints. MCTS has been revitalized for text generation through speculative parallelization (Cheng, Kandemir, and Hong 2024) and dynamic-memory guidance (Shi, Fang, and Chen 2025). MCTS also underpins frameworks like Tree-of-Thoughts (Yao et al. 2023), which explore multiple reasoning paths for complex problem-solving. These advances complement earlier evolutionary or RL-based strategies (Wu et al. 2025), yielding more controllable and reliable text generation.

Applying search to legal text, particularly patent claims, presents unique challenges due to strict requirements for coherence, legal compliance, and scope preservation. ToC addresses this by integrating an uncertainty-aware MCTS with specialized LLM agents (Hu et al. 2024). It defines atomic edit operations as moves in the search tree, ensuring every revision path is plausible (Agrawal, Xu, and Carpuat 2021). A multi-objective reward function guides the MCTS, balancing novelty gain against scope and legal coherence (Yuan et al. 2024). A novel  $\sigma$ -gating mechanism further filters candidate edits: the ExaminerAgent reports a variance-based confidence score, and moves with  $\sigma$  above a preset threshold are pruned, reducing erroneous commitments (Ling et al. 2024). These innovations make ToC’s search process controllable and interpretable, providing a traceable and auditable chain of revisions critical for high-stakes patent claim editing.

## 3 Methodology

Recent advances in LLMs have significantly impacted legal text generation but often fall short in high-stakes editing tasks like patent claim revision. Single-shot or few-shot models generally fail to incorporate essential components such as legal constraints, strategic reasoning, and iterative refinement required for expert-level patent drafting.

Observing real-world workflows, we identify two primary roles employed by patent professionals: generating legally valid edits and evaluating these edits for novelty, clarity, and scope. To systematically manage their interaction and effectively control the extensive editing space, we model the claim optimization problem as a structured search task.

### 3.1 Problem Formulation

Given an initial claim  $C_0$  and a set of prior art documents  $P = \{P_1, \dots, P_m\}$ , the objective is to generate a revised claim  $C_T$  that maximizes novelty over  $P$  while preserving

the technical scope of  $C_0$ . The process is modeled as a sequential decision-making problem where each state  $s_t$  represents a claim configuration at time step  $t$ , and each action  $a_t$  represents an atomic editing operation. Additionally, the prior art documents may include images that provide contextual information relevant to the claim.

The action space  $\mathcal{A}$  includes ten atomic operations. Each operation  $a_t$  is characterized by a tuple  $(o_t, e_t, r_t, c_t)$ , where  $o_t$  is the operation type,  $e_t$  is the target element identifier,  $r_t$  is the reasoning chain, and  $c_t$  is the confidence score. See Table 1 for a summary of atomic operations.

To account for editing dependencies and ensure semantic validity, we define the precedence relations among actions. For instance, ADDNOVELFEATURE must precede REPLACESYNONYM to ensure semantic coherence.

The goal is to find an optimal sequence  $\mathcal{A}^* = (a_0, \dots, a_{T-1})$  that maximizes the expected cumulative reward:

$$\mathcal{A}^* = \arg \max_{\mathcal{A}} \mathbb{E} \left[ \sum_{t=0}^{T-1} R(s_t, a_t) \right] \quad (1)$$

where  $R(s_t, a_t)$  is the reward function that evaluates the quality of the modification at each step.

| Operation          | Description                              |
|--------------------|--|
| AddNovelFeature    | Introduce a new technical feature        |
| ReplaceSynonym     | Replace with a synonym                   |
| ReframeViaFigure   | Reframe using a figure                   |
| DropElement        | Remove an element                        |
| MergeElements      | Merge two or more elements               |
| SplitElement       | Split an element into parts              |
| AddLimitation      | Add a limiting condition                 |
| ModifyRelationship | Modify the relationship between elements |
| ChangeOrder        | Change the order of elements             |
| AddDependency      | Add a dependency between elements        |

Table 1: Summary of Atomic Edit Operations in ToC.

### 3.2 ToC Search Architecture

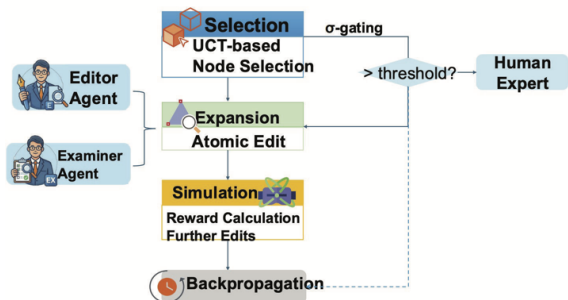


Figure 1: Detailed workflow of the ToC methodology.

In addressing the problem of patent claim optimization, we propose a new search architecture to effectively gen-

erate revised claims. Figure 1 presents the comprehensive ToC workflow, integrating MCTS, multi-agent collaboration, and uncertainty-aware controls. The search traverses four phases: Selection, Expansion, Simulation, and Back-propagation, leveraging uncertainty gating with a threshold ( $\sigma_{epi}^{max} = 0.2$ ) determined through rigorous tuning experiments optimizing performance and consistency.

The search process is organized as follows: Each node in the search tree represents a claim state  $n = (C, O, R, \sigma)$ , where  $C$  is the claim text,  $O$  is the edit operation sequence,  $R$  is cumulative reward, and  $\sigma = (\sigma^{epi}, \sigma^{ale})$  represents epistemic and aleatoric uncertainty components. The tree is explored through four enhanced phases:

1. **Selection.** Starting at the root, the search descends by repeatedly choosing the child with the highest UCT score

$$UCT(n) = \frac{Q(n)}{N(n)} + c \sqrt{\frac{\ln N(p)}{N(n)}}, \quad (2)$$

where  $Q(n)$  and  $N(n)$  are the cumulative reward and visit count of node  $n$ ,  $N(p)$  is the visit count of its parent, and  $c$  controls exploration.  **$\sigma$ -Gating.** For each visited node we estimate epistemic variance  $\sigma_{epi}(n)$ . Paths with  $\sigma_{epi}(n) > \sigma_{max}^{epi} = 0.2$  are pruned or flagged for human review, steering the search away from highly uncertain regions.

2. **Expansion.** Once an expandable node is reached, EXAMINERAGENT pinpoints candidate claim elements; EDITORAGENT then proposes concrete atomic edits. To keep the branching factor manageable we apply *progressive widening*

$$K(n) = \lceil \alpha N(n)^\delta \rceil, \quad \alpha > 0, 0 < \delta < 1, \quad (3)$$

so that only the first  $K(n)$  high-value edits become children, ensuring semantic validity while controlling tree growth.

3. **Simulation.** From the newly added child we roll out a complete claim using a rule-guided policy with three selectable modes: *entropy-based* (favor edits with high uncertainty), *confidence-based* (favor edits with high agent confidence), and *hybrid* (weighted mix 0.6/0.4).

4. **Backpropagation.** The reward  $R(C_t)$  of the simulated claim  $C_t$  is propagated up the path:

$$Q(n) \leftarrow Q(n) + R(C_t), \quad N(n) \leftarrow N(n) + 1. \quad (4)$$

The search terminates when any of the following conditions are met: maximum iterations  $T_{max} = 800$  reached, reward improvement stalls ( $|R_t - R_{t-1}| < \epsilon = 0.01$ ), maximum search time  $T_{search} = 3600$  seconds exceeded, or consecutive failures exceed threshold  $N_{fail} = 20$ .

### 3.3 Multi-Agent Collaboration

The ToC framework employs a collaborative pair of specialized LLM-driven agents that work in concert to achieve optimal claim modification:

**Examiner Agent** The EXAMINERAGENT performs disclosure analysis by examining each claim element against prior art documents, generating structured reasoning chains  $\mathcal{R} = \{r_1, \dots, r_n\}$ . Each reasoning chain  $r_i$  contains:

---

**Algorithm 1: ToC-MCTS with Uncertainty-Aware Search**


---

```

1: Initialize root node  $N_0 \leftarrow (C_0, \emptyset, 0, \sigma_0)$ 
2: Initialize uncertainty decomposer and progressive widening
3: for  $t = 1$  to  $T_{\max}$  do
4:    $N_{\text{sel}} \leftarrow \text{Select}(N_0)$ 
5:   if  $\sigma_{\text{epi}}(N_{\text{sel}}) > \sigma_{\text{max}}^{\text{epi}}$  then
6:     Trigger human intervention or strategy switch
7:   end if
8:    $N_{\text{exp}} \leftarrow \text{Expand}(N_{\text{sel}})$ 
9:    $R \leftarrow \text{Simulate}(N_{\text{exp}})$ 
10:   $\text{Backprop}(N_{\text{exp}}, R)$ 
11:  if  $\text{ShouldTerminate}()$  then
12:    break
13:  end if
14: end for
15: return best  $C^*$  on highest reward path

```

---

- **Status:** Disclosed, NotDisclosed, or PartiallyDisclosed
- **Evidence text:** Direct quotes from prior art supporting the judgment
- **Confidence score:**  $c_i \in [0, 1]$  indicating the examiner’s confidence
- **Uncertainty measure:**  $\sigma_i \in [0, 1]$  quantifying decision uncertainty
- **Human intervention flag:**  $h_i \in \{0, 1\}$  indicating need for human review

The examiner agent uses a specialized prompt template that requires strict JSON output format to ensure consistent parsing and evaluation. When uncertainty exceeds the threshold  $\sigma_{\text{max}}^{\text{epi}} = 0.2$ , the system flags the case for human intervention.

**Editor Agent** The EDITORAGENT generates atomic edit operations based on the examiner’s feedback, applying operations that transform disclosed elements while preserving claim scope. The agent maintains operation history and evaluates modification quality through multiple metrics:

- **Text similarity:** Measures semantic preservation between original and modified claims
- **Scope preservation:** Quantifies the degree to which the original claim scope is maintained
- **Legal readability:** Assesses whether the modified claim maintains patent-legal language standards
- **Technical coherence:** Evaluates the logical consistency of the modified claim

The editor agent employs a sophisticated operation selection strategy that considers both the disclosure status of elements and the potential impact of each operation on claim scope and novelty.

This collaborative loop ensures that each edit is both legally robust and technically meaningful. The concise prompt templates for both agents are summarized in Table 2.

| Agent Role            | Core Task and Key Instructions  |
|-----------------------|---|
| <b>Examiner Agent</b> | Analyzes a claim element against prior art. <ul style="list-style-type: none"> <li>• Decompose into sub-points.</li> <li>• Check each for disclosure (synonym/functional/structural equivalence).</li> <li>• Output strict JSON: <code>claim_points</code>, <code>matched_prior_art</code>, <code>judgement</code>, <code>reasoning</code>.</li> <li>• No natural language outside JSON.</li> </ul> Inputs: <code>&lt;claim_element&gt;</code> , <code>&lt;prior_art&gt;</code> |
| <b>Editor Agent</b>   | Revises claims based on examiner feedback to enhance novelty. <ul style="list-style-type: none"> <li>• Focus on “disclosed” sub-points; preserve “not disclosed” parts.</li> <li>• Avoid trivial paraphrasing; introduce technical novelty.</li> <li>• Maintain grammar and logical coherence.</li> <li>• Output only the revised claim text.</li> </ul> Inputs: <code>Original claim</code> , <code>Judgement chain</code>   |

---

Table 2: Summarized prompt templates for Examiner Agent and Editor Agent.

### 3.4 Reward Function Design

The Monte Carlo tree must balance five often-competing objectives: novelty coverage, scope retention, technical novelty, logical consistency, and risk control within a single scalar score. We therefore combine them linearly:

$$\begin{aligned}
 R(C_t) = & w_1 R_{\text{coverage}}(C_t) - w_2 R_{\text{scope}}(C_t) \\
 & + w_3 R_{\text{novelty}}(C_t) + w_4 R_{\text{consistency}}(C_t) \\
 & - w_5 R_{\text{uncertainty}}(C_t),
 \end{aligned} \quad (5)$$

where  $(w_1, \dots, w_5) = (1.0, 0.5, 1.5, 0.8, 0.3)$  are normalized weights empirically derived from a development set to reflect the practical importance of maximizing coverage and novelty while keeping scope narrowing and uncertainty in check. Each component is computed at the element level and then averaged: coverage rewards turning *Disclosed* points into *NotDisclosed*; scope penalises unnecessary narrowing; novelty counts only those changes the examiner rates as innovative; consistency multiplies legal readability with technical coherence; and uncertainty, derived from epistemic variance, discourages speculative edits.

To isolate genuine model doubt from data noise, total variance is decomposed into epistemic and aleatoric terms,  $\sigma^{\text{total}} = \sigma^{\text{epi}} + \sigma^{\text{ale}}$  with  $\sigma^{\text{ale}} = 1 - \text{confidence}$ . Only  $\sigma^{\text{epi}}$  enters  $R_{\text{uncertainty}}$ , so stochastic prior-art phrasing does not unduly penalise promising revisions.

Finally, progressive widening controls the branching factor as the search deepens:

$$K(n) = \lceil \alpha N(n)^\delta \rceil, \quad (\alpha, \delta) = (2.0, 0.5) \quad (6)$$

which encourages broad exploration early on and

| Model                      | ToC | $R_{cov}$   | $R_{scope}$ | $R_{novelty}$ | $R_{cons}$  | $R_{uncert}$ | Overall           |
|----------------------------|-----|-------------|-------------|---------------|-------------|--------------|-------------------|
| <b>Closed-Source MLLMs</b> |     |             |             |               |             |              |                   |
| OpenAI O1                  | ✓   | 0.560±0.048 | 0.374±0.051 | 0.712±0.050   | 0.942±0.012 | 0.071±0.014  | 0.680±0.03        |
| + zero-shot                | ✗   | 0.492±0.044 | 0.401±0.053 | 0.643±0.048   | 0.931±0.019 | 0.079±0.020  | 0.631±0.03        |
| + few-shot                 | ✗   | 0.525±0.037 | 0.388±0.056 | 0.685±0.041   | 0.937±0.018 | 0.075±0.020  | 0.658±0.03        |
| GPT-4o                     | ✓   | 0.582±0.050 | 0.389±0.050 | 0.732±0.050   | 0.956±0.017 | 0.068±0.018  | <b>0.701±0.03</b> |
| + zero-shot                | ✗   | 0.520±0.046 | 0.417±0.052 | 0.659±0.049   | 0.947±0.018 | 0.071±0.019  | 0.647±0.03        |
| + few-shot                 | ✗   | 0.555±0.045 | 0.405±0.051 | 0.698±0.047   | 0.951±0.014 | 0.070±0.019  | 0.678±0.03        |
| Claude-3.5 Sonnet          | ✓   | 0.548±0.049 | 0.370±0.057 | 0.703±0.051   | 0.945±0.018 | 0.075±0.020  | 0.675±0.03        |
| + zero-shot                | ✗   | 0.472±0.045 | 0.408±0.053 | 0.633±0.047   | 0.934±0.019 | 0.082±0.021  | 0.626±0.03        |
| + few-shot                 | ✗   | 0.510±0.044 | 0.388±0.052 | 0.652±0.046   | 0.940±0.018 | 0.080±0.022  | 0.653±0.03        |
| <b>Open-Source MLLMs</b>   |     |             |             |               |             |              |                   |
| Qwen2.5-VL-32B             | ✓   | 0.507±0.047 | 0.351±0.053 | 0.665±0.050   | 0.924±0.027 | 0.083±0.025  | 0.639±0.03        |
| + zero-shot                | ✗   | 0.424±0.043 | 0.382±0.057 | 0.598±0.049   | 0.913±0.028 | 0.090±0.026  | 0.592±0.03        |
| + few-shot                 | ✗   | 0.468±0.044 | 0.370±0.051 | 0.635±0.048   | 0.918±0.028 | 0.087±0.025  | 0.615±0.03        |
| Qwen2.5-VL-72B             | ✓   | 0.534±0.048 | 0.361±0.056 | 0.682±0.051   | 0.930±0.026 | 0.080±0.024  | 0.658±0.03        |
| + zero-shot                | ✗   | 0.452±0.044 | 0.392±0.054 | 0.624±0.048   | 0.921±0.020 | 0.088±0.025  | 0.611±0.03        |
| + few-shot                 | ✗   | 0.495±0.045 | 0.378±0.051 | 0.655±0.047   | 0.926±0.027 | 0.084±0.024  | 0.636±0.03        |

Table 3: Primary evaluation metrics for ToC and zero-/few-shot baselines on five MLLMs (mean ± SD over 3 seeds,  $N = 500$ ).

fine-grained exploitation later without letting the tree explode.

## 4 Experiments

We conduct comprehensive experiments to evaluate the effectiveness, robustness, and interpretability of the proposed ToC framework in real-world patent claim editing scenarios. Our evaluation focuses on three core questions:

- **Effectiveness:** Does ToC generate higher-quality claims compared to strong LLM-based baselines, including zero-shot and few-shot prompting setups?
- **Component Contribution:** How do modules such as multi-agent collaboration, uncertainty gating, and progressive widening affect performance?
- **Control and Stability:** How sensitive to key hyperparameters like reward weights and uncertainty thresholds?

To answer these questions, we perform comparative evaluations, ablation studies, sensitivity analyses, and expert assessments. Results are measured across multiple objective and subjective dimensions, including novelty, scope preservation, legal readability, and human preference. The following subsections detail the experimental setup, dataset construction, evaluation metrics, and a performance analysis.

### 4.1 Experimental Setup

All experiments are conducted on a high-performance computing cluster equipped with NVIDIA A100 GPUs (80GB). To ensure reproducibility, we fix all random seeds and repeat each experiment three times, reporting the mean and standard deviation. The system is implemented in Python, and all hyperparameters including search depth, reward weights, and LLM decoding settings are managed through a unified configuration interface.

We evaluate the ToC framework using a diverse set of multimodal LLMs as reasoning agents. For closed-source

models, we include OpenAI O1 (Jaech et al. 2024), GPT-4O (Hurst et al. 2024), and Claude 3.5 Sonnet (Anthropic 2024), accessed via their official APIs with deterministic decoding settings (temperature = 0.0). For open-source models, such as Qwen2.5-VL (32B/72B) (Wang et al. 2024a), all inference is performed locally using the HuggingFace Transformers library. Each model is deployed in both the examiner and editor roles within the ToC multi-agent loop, following consistent prompt structures and unified output schemas to ensure a fair and controlled comparison. All evaluations are conducted on standardized data splits with identical inputs and evaluation protocols across models.

### 4.2 Dataset Preprocessing

We construct our benchmark from the USPTO Office Actions Research Dataset <sup>1</sup>, focusing on wireless communications patents and 2016 office actions. The dataset contains 1,145 unique patents (106 allowed, 1,039 rejected), 28,261 claims, and 8,418 prior art references. The data preprocessing pipeline includes parsing structured patent and office action records, aligning examiner-cited prior art with rejected claims, mapping full claim and prior art texts, and decomposing claims into semantic sub-elements. Prior art documents are retrieved through automated web scraping from Google Patents using publication IDs extracted from office action records. Prior art documents are further processed to extract relevant evidence using embedding-based similarity filtering. Our dataset is multimodal in nature, incorporating both textual descriptions and supporting figures/diagrams from patent documents, enabling comprehensive analysis of technical content across different modalities. Each data instance is represented as a quadruple  $\langle c_i, y_i, e_i, r_i \rangle$ , where  $c_i$  is a claim element,  $y_i$  is a binary disclosure label,  $e_i$  is the matched evidence (including both text and visual elements),

<sup>1</sup><https://data.uspto.gov/bulkdata/datasets/ptofact?fileDataFromDate=2017-11-29&fileDataToDate=2017-11-29>

| Model                      | ToC | JSON               | PPL              | ROUGE-L            | BLEU               |
|----------------------------|-----|--------------------|------------------|--------------------|--------------------|
| <b>Closed-Source MLLMs</b> |     |                    |                  |                    |                    |
| OpenAI O1                  | ✓   | 0.994±0.003        | 9.10±1.10        | 0.602±0.050        | 0.540±0.051        |
| + zero-shot                | ✗   | 0.992±0.004        | 9.42±1.15        | 0.567±0.050        | 0.497±0.049        |
| + few-shot                 | ✗   | 0.993±0.004        | 9.25±1.14        | 0.587±0.049        | 0.518±0.050        |
| GPT-4o                     | ✓   | <b>0.996±0.001</b> | <b>8.72±1.00</b> | <b>0.624±0.049</b> | <b>0.554±0.052</b> |
| + zero-shot                | ✗   | 0.994±0.003        | 8.93±1.05        | 0.590±0.046        | 0.522±0.050        |
| + few-shot                 | ✗   | 0.995±0.003        | 8.85±1.03        | 0.610±0.048        | 0.537±0.051        |
| Claude-3.5 Sonnet          | ✓   | 0.995±0.003        | 8.98±1.08        | 0.611±0.050        | 0.530±0.051        |
| + zero-shot                | ✗   | 0.993±0.004        | 9.30±1.14        | 0.582±0.049        | 0.501±0.050        |
| + few-shot                 | ✗   | 0.994±0.002        | 9.18±1.10        | 0.595±0.049        | 0.513±0.050        |
| <b>Open-Source MLLMs</b>   |     |                    |                  |                    |                    |
| Qwen2.5-VL-32B             | ✓   | 0.992±0.004        | 9.80±1.20        | 0.582±0.050        | 0.510±0.051        |
| + zero-shot                | ✗   | 0.990±0.005        | 10.08±1.25       | 0.551±0.050        | 0.478±0.050        |
| + few-shot                 | ✗   | 0.991±0.005        | 9.96±1.23        | 0.567±0.050        | 0.490±0.050        |
| Qwen2.5-VL-72B             | ✓   | 0.993±0.004        | 9.52±1.17        | 0.596±0.050        | 0.525±0.051        |
| + zero-shot                | ✗   | 0.991±0.003        | 9.76±1.22        | 0.563±0.050        | 0.492±0.050        |
| + few-shot                 | ✗   | 0.992±0.004        | 9.65±1.18        | 0.582±0.050        | 0.508±0.051        |

Table 4: Auxiliary generation metrics for the same zero-/few-shot and ToC runs.

| Metric                    | Value  | Percentage |
|---------------------------|--------|------------|
| Total Patents             | 1,145  | 100%       |
| - Allowed Patents         | 106    | 9.3%       |
| - Rejected Patents        | 1,039  | 90.7%      |
| Total Claims              | 28,261 | 100%       |
| - Allowed Claims          | 4,272  | 15.1%      |
| - Rejected Claims         | 23,989 | 84.9%      |
| Average Claims per Patent | 24.7   | -          |
| Patents with Prior Art    | 574    | 50.1%      |
| Patents without Prior Art | 571    | 49.9%      |
| Claims with Prior Art     | 23,260 | 82.3%      |
| Claims without Prior Art  | 5,001  | 17.7%      |
| Prior Art References      | 8,418  | -          |
| Overlapping Applications  | 27     | 2.4%       |

Table 5: Detailed dataset description by category.

and  $r_i$  is a justification. All annotations are reviewed by technical experts to ensure quality and consistency.

The dataset (in Table 5) is imbalanced (9.3% allowed, 90.7% rejected), with allowed patents having more claims on average. 82.3% of claims have prior art references.

### 4.3 Baselines and Evaluation Metrics

To evaluate the effectiveness of the proposed ToC framework, we compare it against two baseline settings for each LLM: a zero-shot setting without any reasoning guidance, and a few-shot (2-shot) setting with exemplars and Chain-of-Thought reasoning prompts. All setups adopt identical input-output formats, prompts, and search budgets to ensure fair comparison across models and configurations.

We evaluate every model on a 500-sample hold-out set of granted and rejected claims. Core metrics follow our reward terms: coverage F1 and  $\Delta R_{\text{coverage}}$  for disclosure

| Error Category          | Count | Proportion (%) |
|-------------------------|-------|----------------|
| Input Misalignment      | 78    | 6.8            |
| Invalid Modifications   | 120   | 10.5           |
| Unsupported Novelty     | 113   | 9.9            |
| Legal or Style Issues   | 31    | 2.7            |
| System Control Failures | 139   | 12.2           |

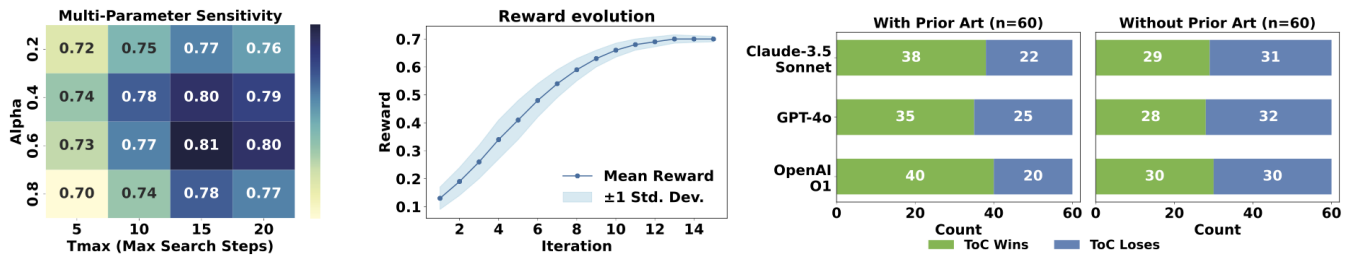
Table 6: Error analysis: categorized failure types in ToC-generated revisions.

avoidance, scope similarity/loss for scope retention, novelty, consistency, and an uncertainty score. To probe general robustness we additionally report JSON-parsing completeness, chain entropy, image–text consistency, legal readability, token-level perplexity (PPL), as well as ROUGE-L, BLEU, and expert preference.

## 4.4 Experimental Results

**Performance Comparison.** Table 3 and table 4 show that integrating ToC markedly boosts every model, with the closed-source GPT-4o leading overall. It posts the best coverage (0.582), novelty (0.732), consistency (0.956), and lowest perplexity (8.72), giving it the highest aggregate score (0.701). Among open-source options, Qwen2.5-VL-72B ranks first, outperforming its 32B variant on all reward-aligned metrics while remaining within 5% of the GPT-4o benchmark. The strong gains of both Qwen models over their non-ToC baselines confirm that the framework transfers reliably across parameter scales.

**Error Analysis.** Expert evaluations involved five patent specialists with at least five years of industry experience, evaluating revisions against a detailed rubric focusing on novelty, legal robustness, and technical coherence. Additionally, these experts conducted comparative evaluations between ToC-revised claims and original drafts, revealing a



(a) Sensitivity heat-map: accuracy vs. uncertainty threshold and MCTS search depth. (b) Reward evolution during the search.

(c) Expert preference scores for final claims.

Figure 2: Visual analysis of ToC optimisation. (a) Sensitivity to hyper-parameters. (b) Reward dynamics across search steps. (c) Human expert evaluation of generated claims.

clear preference for ToC revisions in contexts where prior art was available. Specifically, ToC was preferred in approximately two-thirds of evaluations (see Figure 2c). The detailed error analysis (Table 6) identified prevalent *System Control Failures*, notably including excessive branching due to uncertainty, zero-confidence decisions, missed pruning opportunities, and invalid modifications leading to semantic or legal inconsistencies. Further frequently encountered errors involved unsupported claims of novelty, highlighting areas for future improvement in uncertainty calibration, validation mechanisms, and alignment with prior-art evidence.

**Ablation Study.** To understand the contribution of each module, we conduct a comprehensive ablation analysis. As shown in Figure 3, disabling components like uncertainty gating, progressive widening, or multi-agent collaboration results in noticeable performance degradation. And the absence of uncertainty control and agent interaction significantly impacts novelty, coverage, and overall legal quality.

**Sensitivity Analysis.** Figure 2a sweeps two key hyper-parameters: the progressive-widening coefficient  $\alpha$  (rows) and the maximum search depth  $T_{\max}$  (columns). Accuracy is remarkably stable (0.72–0.81) across the grid, peaking at  $\alpha = 0.6$  and  $T_{\max} = 15$ , a setting that balances the breadth of candidate edits with sufficient roll-out depth. Very small trees ( $T_{\max} \leq 5$ ) or overly aggressive widening ( $\alpha = 0.8$ ) shave off 3–5 points, confirming that extreme exploration or shallow search can dilute reward optimisation. Figure 2b complements the heat-map by tracking the mean reward over search iterations. Roughly 70% of the eventual gain is achieved within the first six iterations; the curve then plateaus after iteration 10, indicating diminishing returns and validating our depth budget.

## 5 Conclusion

We introduced ToC, a novel framework combining structured search and multi-agent collaboration for patent claim optimization. By modeling editing as a sequential decision-making task via MCTS, ToC provides a transparent and controlled approach that significantly improves novelty, legal quality, and expert preference compared to existing

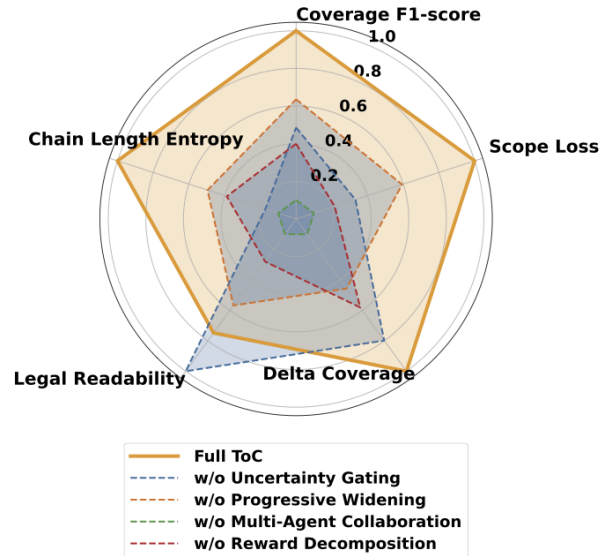


Figure 3: Component-level performance under module ablations. Full ToC includes all modules; others have one component removed.

methods. And ToC represents a paradigm shift from passive generation to strategic collaboration, enhancing correctness, transparency, and interpretability in high-stakes legal writing tasks and facilitating effective human-AI interaction. Future work will focus on improving search efficiency through distributed computing, and generalizing the framework to other structured editing domains such as legal contracts, medical protocols, and scientific methods. ToC offers stronger control, traceability, and multi-turn reasoning, making it particularly suited for domains requiring high reliability and domain-specific consistency.

## References

Agrawal, S.; Xu, W.; and Carpuat, M. 2021. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3757–3769.

- Anthropic. 2024. Claude 3.5 Sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>. Accessed: July 2025.
- Bai, Y.; Zhang, J.; Lv, X.; Zheng, L.; Zhu, S.; Hou, L.; Dong, Y.; Tang, J.; and Li, J. 2024a. Longwriter: Unleashing 10,000+ word generation from long context llms. *arXiv preprint arXiv:2408.07055*.
- Bai, Z.; Zhang, R.; Chen, L.; Cai, Q.; Zhong, Y.; Wang, C.; Fang, Y.; Fang, J.; Sun, J.; Wang, W.; et al. 2024b. Patentgpt: A large language model for intellectual property. *arXiv preprint arXiv:2404.18255*.
- Bui, L. V. 2025. Advancing patent law with generative AI: Human-in-the-loop systems for AI-assisted drafting, prior art search, and multimodal IP protection. *World Patent Information*, 80: 102341.
- Cheng, S.; Kandemir, M. T.; and Hong, D.-Y. 2024. Speculative monte-carlo tree search. *Advances in Neural Information Processing Systems*, 37: 88664–88683.
- Chu, J.-M.; Lo, H.-C.; Hsiang, J.; and Cho, C.-C. 2024. From PARIS to LE-PARIS: toward patent response automation with recommender systems and collaborative large language models. *Artificial Intelligence and Law*, 1–27.
- Fang, Z.; Du, G.; Yu, S.; Guo, Y.; Zhang, Y.; Cao, Y.; Li, J.; Tang, H.-K.; and Goh, S. K. 2025a. To See a World in a Spark of Neuron: Disentangling Multi-task Interference for Training-free Model Merging. *arXiv preprint arXiv:2503.05320*.
- Fang, Z.; Li, C.; Zhou, H.; Yu, S.; Du, G.; Qasem, A.; Lu, Y.; Li, J.; Zhang, J.; and Goh, S. K. 2025b. NeuIPT: Foundation Model for Neural Interfaces. *arXiv preprint arXiv:2510.16548*.
- Hong, S.; Zheng, X.; Chen, J.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S. K. S.; Lin, Z.; Zhou, L.; et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4): 6.
- Hu, Z.; Liu, C.; Feng, X.; Zhao, Y.; Ng, S.-K.; Luu, A. T.; He, J.; Koh, P. W.; and Hooi, B. 2024. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in large language models. *arXiv preprint arXiv:2402.03271*.
- Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Jiang, L.; Zhang, C.; Scherz, P. A.; and Goetz, S. 2024. Can Large Language Models Generate High-quality Patent Claims? *arXiv preprint arXiv:2406.19465*.
- Kawano, S.; Nonaka, H.; and Yoshino, K. 2024. ClaimBrush: A Novel Framework for Automated Patent Claim Refinement Based on Large Language Models. In *2024 IEEE International Conference on Big Data (BigData)*, 6594–6603. IEEE.
- Li, B.; Wang, Y.; Meng, T.; Chang, K.-W.; and Peng, N. 2024. Control large language models via divide and conquer. *arXiv preprint arXiv:2410.04628*.
- Ling, C.; Zhao, X.; Zhang, X.; Cheng, W.; Liu, Y.; Sun, Y.; Oishi, M.; Osaki, T.; Matsuda, K.; Ji, J.; et al. 2024. Uncertainty quantification for in-context learning of large language models. *arXiv preprint arXiv:2402.10189*.
- Paul, J. 2024. A Revolutionary Solution for Automating Patent Application Development.
- Ren, R.; and Ma, J. 2024. PatentGPT: A Large Language Model for Patent Drafting Using Knowledge-based Fine-tuning Method. *arXiv preprint arXiv:2409.00092*.
- Shi, Z.; Fang, M.; and Chen, L. 2025. Monte carlo planning with large language model for text-based game agents. *arXiv preprint arXiv:2504.16855*.
- Son, J.; Moon, H.; Lee, J.; Lee, S.; Park, C.; Jung, W.; and Lim, H. 2022. Ai for patents: a novel yet effective and efficient framework for patent analysis. *IEEE Access*, 10: 59205–59218.
- Wang, B. T. 2024. Prompts and large language models: a new tool for drafting, reviewing and interpreting contracts? *Law, Technology and Humans*, 6(2): 88–106.
- Wang, J.; Mudhiganti, S. K. R.; and Sharma, M. 2024. Patentformer: a novel method to automate the generation of patent applications. In *Proceedings of the 2024 conference on empirical methods in natural language processing: industry track*, 1361–1380.
- Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024a. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Wang, Q.; Ni, S.; Liu, H.; Lu, S.; Chen, G.; Feng, X.; Wei, C.; Qu, Q.; Alinejad-Rokny, H.; Lin, Y.; et al. 2024b. Autopatent: a multi-agent framework for automatic patent generation. *arXiv preprint arXiv:2412.09796*.
- Wang, S.; Yin, X.; Wang, M.; Guo, R.; and Nan, K. 2024c. Evopat: A multi-llm-based patents summarization and analysis agent. *arXiv preprint arXiv:2412.18100*.
- Wu, J.; Feng, M.; Zhang, S.; Jin, R.; Che, F.; Wen, Z.; and Tao, J. 2025. Boosting multimodal reasoning with mcts-automated structured thinking. *arXiv e-prints*, arXiv:2502.
- Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.
- Xie, T.; Zhou, F.; Cheng, Z.; Shi, P.; Weng, L.; Liu, Y.; Hua, T. J.; Zhao, J.; Liu, Q.; Liu, C.; et al. 2023. Openagents: An open platform for language agents in the wild. *arXiv preprint arXiv:2310.10634*.
- Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.
- Yuan, W.; Pang, R. Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 3.