SettleAgent: Planning-Enhanced Multi-Agent Legal Negotiation

Anonymous ACL submission

Abstract

Current research on LLM-based legal agents has largely focused on verdict-oriented tasks, while lacking systematic modeling of pre-trial negotiation and settlement processes in civil disputes. We propose SettleAgent, a multi-agent framework for legal negotiation and settlement, and incorporate a red-team adversarial mechanism to systematically evaluate robustness under extreme bargaining tactics. We model bargaining as a sequential game: at each round, counsel agents first generate candidate proposals, then perform forward-looking strategic exploration via tree search, and finally apply RoT (Reflection on Search Trees) to distill success/failure patterns from branching outcomes to update their memory. We also introduce SettleBench, a benchmark for legal negotiation and settlement built from publicly available civil case documents. Experiments show that SettleAgent significantly outperforms a range of LLM and agent baselines in both settlement success and outcome quality, while remaining more stable under red-team stress testing.

1 Introduction

In recent years, LLMs have shown strong capabilities in simulating scenarios and have achieved notable progress on static legal tasks such as retrieval, question answering, and judgment prediction (He et al., 2024), which has in turn spurred extensive agent-based research on simulated courts and adjudication. However, in dynamic and adversarial legal interactions, models often struggle to apply legal rules effectively and to maintain stable reasoning. Moreover, real-world civil disputes rarely end in a final judgment: a large fraction of cases are resolved before trial through negotiation, mediation, or settlement. The core of negotiation and settlement is to reach an agreement that is acceptable to both parties, legally compliant, and enforceable under legal constraints and litigation risk.

Directly relying on LLM or dialogue-based multi-agent setup for legal negotiation therefore entails systematic risks. Negotiation must converge with reference to best alternative to a negotiated agreement (BATNA) and litigation costs; without such a "shadow-of-the-law" anchor, dialogue can be easily driven by rhetoric, distorted by anchoring effects, or trapped in prolonged deadlock. Even when an agreement is reached, unconstrained generation may produce illegal, unenforceable, or coercive clauses, making "agreement" diverge from "implementability." In addition, negotiation inherently involves information asymmetry and power imbalance, and models may amplify structural unfairness when protection for the weaker party is not made explicit. To address these issues, we propose SettleAgen, a framework for legal negotiation and settlement. First, a Clerk agent structures the case file and produces role-conditioned case briefs; an Outcome agent then constructs BATNA references. Next, during negotiation, two counsel agents generate proposals and engage in bargaining under the coordination of a Mediator agent. We further incorporate tree search for lookahead planning, and use Reflection on Search Trees (RoT) distillation to write back search experiences into a memory bank, enabling strategy evolution across rounds. Finally, a Compliance Auditor agent verifies the legality and enforceability of the agreement, triggering revisions or rollbacks when necessary. We additionally introduce a Legal Aid agent to support the weaker party, and systematically evaluate robustness via an independent Evaluation agent and red-team adversarial experiments. To enable evaluation of such capabilities, we further introduce SettleBench, a benchmark specifically designed to assess models' performance in the dynamic, interactive setting of legal negotiation and settlement.

Our main contributions are as follows: 1. We propose SettleAgent, which, to the best of our knowledge, is the first multi-agent framework for

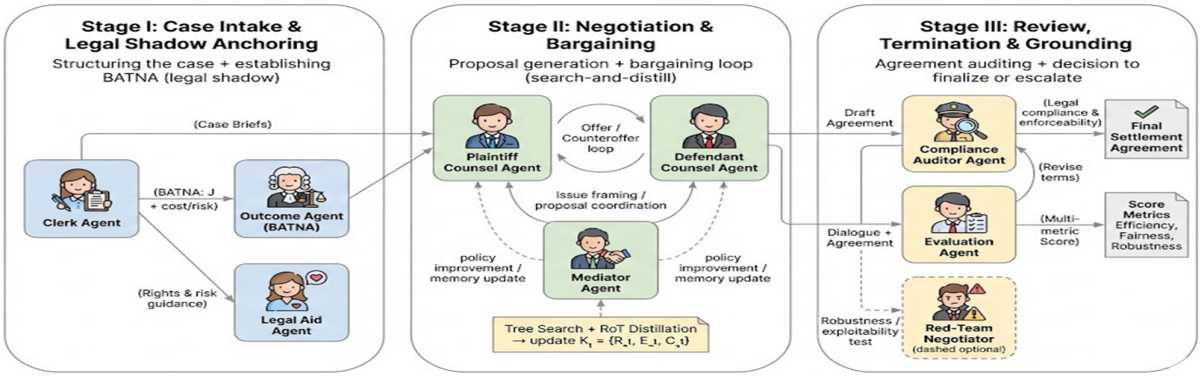

Figure 1: Overview of the SettleAgent framework. In Stage I, the Clerk agent structures the case file and produces role-specific case briefs; the Outcome agent provides an expected judgment outcome along with cost and risk references; and the Legal Aid agent offers rights explanations and risk reminders to the disadvantaged party. In Stage II, the plaintiff and defendant lawyer agents engage in an offer–counteroffer loop under the mediator's coordination, performing look-ahead proposal exploration via tree search and using RoT distillation to write back multi-branch experience into the memory bank, thereby improving cross-round strategy consistency and robustness. In Stage III, the Compliance Audit agent reviews the drafted agreement; the Evaluation agent scores the negotiation outcome; and a red-team adversarial negotiator stress-tests the system's robustness under extreme strategies, ultimately producing an actionable settlement agreement or a failure-to-settle outcome.

legal negotiation and settlement in civil disputes. 2. We introduce SettleBench, the first benchmark dedicated to legal negotiation and settlement, designed to evaluate model capabilities in this dynamic interactive setting and to fill a critical gap in existing legal AI evaluation for pre-trial settlement negotiations. 3. We propose a planning-enhanced negotiation mechanism that uses search-and-distill as the core of the bargaining stage: it first performs tree-search-based lookahead over multi-step proposals and concession paths to filter feasible agreements; it then introduces ROT to distill stable strategic patterns and clause regularities from multi-branch search outcomes back into the policy for continual improvement, enabling counsel agents to continuously update their memory bank without additional annotation. Our framework is illustrated in Fig. 1.

2 Related Work

2.1 LLMsintheLegal Domain

Legal Artificial Intelligence aims to improve legal practice by leveraging artificial intelligence techniques (Surden, 2018; Katz et al., 2023). With the continued development of deep learning, legal NLP has advanced across a wide range of tasks, including legal judgment prediction (Wu et al., 2023), legal question answering (Cui et al., 2023; Louis et al., 2024), legal language understanding (Niklaus et al., 2023), legal case retrieval (Shao et al., 2023), and legal document summarization (Jain et al., 2024). In particular, the emergence of large language models (LLMs) has further accelerated progress in legal applications such as case prediction, legal research, and document analysis (Hamilton, 2023), and has motivated various strategies for enhancing legal reasoning via specialized legal models. Representative examples HanFei-7B, which emphasizes legal knowledge representation and statutory interpretation (Chen et al., 2025); ChatLaw-33B, which integrates a mixture-of-experts architecture with a legal knowledge graph (Cui et al., 2023); DISC-LawLLM for intelligent legal services (Yue et al., 2023); and DeliLaw, a dialogue-based system for efficient legal inquiry handling (Xie et al., 2024).

2.2 Multi-agent forReal-World Simulation

LLM-based multi-agent systems have rapidly advanced in recent years. By coordinating multiple agents, these systems enable knowledge sharing, cognitive synergy, and enhanced decision-making, thereby improving the efficiency and effectiveness of solving complex tasks ((Talebirad and Nadiri, 2023). Prior work suggests that interaction mechanisms akin to human group dynamics can substantially strengthen system capabilities: (Park et al., 2023) show that social behaviors can autonomously emerge within groups of agents; and (Hong et al.) further propose multi-agent paradigms that leverage natural-language communication throughout the entire software development process. Multi-agent cooperation has also been validated across diverse application domains, such as improving language understanding and generation in NLP (Tan and Motani, 2024), enhancing decision-making in

human–robot interaction (Kim et al., 2024), supporting task decomposition and collaborative completion in planning and execution (Yang et al., 2024), enabling personalized learning and intelligent tutoring in education (Yin et al., 2024), and contributing to market analysis, risk assessment, and investment decision-making in finance (Nascimento et al., 2023). Although prior work has made notable progress, existing legal-domain LLM systems are still primarily evaluated on static, well-defined tasks (e.g., QA, retrieval, and judgment prediction), and thus offer limited support for pre-trial negotiation. In pre-trial negotiation, however, agents must converge to an executable agreement under the "shadow of the law" (litigation risks and costs) and strict legal constraints—posing a significant challenge for current multi-agent frameworks. To address this gap, we propose a SettleAgen for negotiation and introduce an evaluation benchmark, SettleBench, to enable systematic assessment.

3 SettleAgen: Negotiation and Settlement

3.1 Agent Design

We model legal negotiation and settlement as a multi-agent interactive environment consisting of seven core agents and two auxiliary agents. The core agents include a Clerk Agent, which organizes case materials, produces role-specific case briefs, and logs the negotiation process into structured minutes. The Clerk's case digest and role-specific briefs are then delivered to the Plaintiff Counsel Agent and the Defendant Counsel Agent, enabling both sides to propose offers, respond to counterpart moves, and safeguard their respective interests throughout the negotiation. We further introduce an Outcome Agent that reads the case skeleton and outputs the expected litigation outcome and associated litigation cost estimates if the negotiation fails and proceeds to court, thereby constructing the BATNA (the outside option "in the shadow of the law"). A Mediator Agent is responsible for issue structuring, package coordination, and driving convergence; a Compliance Auditor Agent performs legality and enforceability checks on draft agreements and, when necessary, triggers constraint-based revisions; and a Legal Aid Agent provides rights clarification, risk warnings, and communication support for the vulnerable party. The auxiliary agents include an Evaluation Agent, which conducts standardized scoring of both outcomes and processes after the negotiation concludes, and a Red-Team Negotiator, which replaces one party in robustness experiments to apply extreme strategies and assess the system's exploitability resistance and safety. The design of the agent roles is illustrated in in Appendix A.1.

3.2 Simulation Workflow

We model legal negotiation and settlement as a stage-wise multi-agent interactive process, consisting of the seven core agents and two auxiliary agents introduced in the previous section. To explicitly characterize the procedural structure, we introduce a stage variable $z_t \in \{\text{Prep}, \text{Barg}, \text{Verify}\}$, corresponding to Settlement Preparation, Negotiation & Bargaining, and Agreement Verification & Closure, respectively. During the negotiation over time steps t, we define the system state as $s_t = \big(x,\ G_{\le t},\ o_t,\ b_t,\ z_t,\ \eta_t\big)$, where x denotes the case skeleton (structured fields such as facts, evidence, and disputed issues), $G_{\le t}$ is the dialogue and offer history up to round t, o_t is the current draft agreement, b_t is the outside option (BATNA) information, and η_t is a process-control indicator . At each time step, the system selects an interacting pair of agents according to the current stage and control variables, and generates an interaction outcome. Let $\mathcal{A}$ denote the set of agents. A scheduling function $(a_i, a_j) = \sigma(s_t) \equiv \sigma(z_t, \eta_t)$ determines which agents interact at step t. The interaction outcome is formalized as $I(a_i, a_j, t) = f_{\text{interact}}\Big(D_{a_i}(s_t),\ \ D_{a_j}(s_t)\Big)$, where $D_a(s_t)$ denotes the decision output of agent a under state s_t (e.g., an offer/counter-offer, a mediation proposal, or compliance feedback). The environment writes the interaction result back to the state via $s_{t+1} = \mathcal{T}\big(s_t,\ I(a_i, a_j, t)\big)$, $z_{t+1} = g\big(z_t,\ s_{t+1}\big)$, where $\mathcal{T}$ denotes the state transition function and g controls stage transitions (moving from preparation to bargaining, from bargaining to verification, or returning to bargaining when compliance vetoes the draft). **Stage I: Settlement Preparation.** This stage consists of two steps: (i) case intake and structuring, and (ii) **BATNA anchoring** (i.e., anchoring the "shadow of the law"). The **Clerk Agent** first consolidates the case materials and generates role-specific briefs; then the **Outcome Agent** outputs the expected adjudication outcome and litigation costs if negotiation fails and proceeds to court, thereby constructing the BATNA.

Clerk Agent. The Clerk Agent's decision function is defined as $D_{\text{clerk}}(s_t) = f_{\text{clerk}}(x, G_{\le t}) =$

$\big(d_t,\ b_t^p,\ b_t^d,\ m_t\big)$, where d_t is the unified case digest, b_t^p and b_t^d are role-specific briefs for the plaintiff and defendant counsel, respectively, and m_t denotes the negotiation minutes. The system writes $\{d_t, b_t^p, b_t^d\}$ into each counsel agent's observable information (encoded in η_t) to support subsequent strategy generation.

Outcome Agent (Litigation Baseline). The Outcome Agent does not participate in bargaining and only produces the outside option: $D_{\text{out}}(s_t) = f_{\text{out}}(x) = \big(J_t,\ c_t^{\text{lit}},\ r_t^{\text{lit}}\big)$, where J_t denotes the expected court outcome if the case proceeds to litigation (either as an interval or an expectation), and c_t^{lit} and r_t^{lit} are the estimated litigation cost and risk, respectively. This output is written into b_t, serving as the BATNA anchor for subsequent negotiation.

Stage Scheduling and Transition. During PREP, the scheduler prioritizes executing the Clerk and Outcome agents:

$$\sigma(s_t) = \begin{cases} (\textsc{Clerk}, \varnothing), & \eta_t.\texttt{brief} = 0, \\ (\textsc{Outcome}, \varnothing), & \eta_t.\texttt{BATNA} = 0, \\ (\textsc{Mediator}, \varnothing), & \text{(Initialize issues).} \end{cases} \tag{1}$$

Once both the briefs and BATNA are ready, the process enters the bargaining stage:

$$g(z_t, s_{t+1}) = \begin{cases} \textsc{Barg}, & z_t = \textsc{Prep} \ \wedge\ \eta_{t+1}.\texttt{brief} = 1 \\ & \qquad \wedge\ \eta_{t+1}.\texttt{BATNA} = 1, \\ z_t, & \text{otherwise.} \end{cases} \tag{2}$$

Stage II: Negotiation & Bargaining. This stage consists of a proposal-generation and bargaining loop. The plaintiff/defendant counsel agents engage in turn-based interactions around the outside option b_t and the current draft agreement o_t. The mediator provides neutral coordination, while legal aid is invoked to deliver explanations and risk warnings when a vulnerable party is triggered.

Counsel Agents (Plaintiff/Defendant Counsel). The counsel decision depends on the current state and an updatable knowledge base K_t: $D_{\text{counsel}}(s_t; K_t) = f_{\text{LLM}}(s_t, K_t) = \big(\textit{type}_t,\ \textit{pkg}_t,\ u_t\big)$, where $\textit{type}_t \in$ offer, counter, accept, reject, request, $\textit{pkg}_t$ is a structured term package (e.g., monetary amount, installment schedule, behavioral commitments, dismissal/confidentiality/breach liability), and u_t is the corresponding natural-language statement. This design enables counsel agents to leverage accumulated knowledge to enhance negotiation capability.

Mediator and Legal Aid Agents. As stable roles, the mediator and legal aid decisions can be written as $D_{\text{med}}(s_t) = f_{\text{LLM}}(s_t)$ $D_{\text{aid}}(s_t) = f_{\text{LLM}}(s_t)$, which output issue structuring/compromise proposals and, for the vulnerable party, rights clarification, risk warnings, and communication support, respectively. **Scheduling and Stage Transition in BARG.** During BARG, the scheduler implements turn-based offer–counteroffer interactions with optional insertions:

$$\sigma(s_t) = \begin{cases} (\textsc{Counsel}_p, \textsc{Counsel}_d), & \eta_t.\texttt{turn} = p, \\ (\textsc{Counsel}_d, \textsc{Counsel}_p), & \eta_t.\texttt{turn} = d, \\ (\textsc{Mediator}, \textsc{Counsel}_k), & \eta_t.\texttt{stall} = 1, \\ (\textsc{LegalAid}, \textsc{Counsel}_k), & \eta_t.\texttt{need\_aid} = 1, \end{cases} \tag{3}$$

where $k \in \{p, d\}$ indicates which side the mediator/legal-aid intervention targets. When a candidate draft is formed (e.g., the parties reach principle-level agreement on core terms), the process enters the verification stage:

$$g(z_t, s_{t+1}) = \begin{cases} \textsc{Verify}, & z_t = \textsc{Barg} \ \wedge \\ & \eta_{t+1}.\texttt{draft\_ready} = 1, \\ z_t, & \text{otherwise.} \end{cases} \tag{4}$$

Stage III: Agreement Verification & Closure. This stage includes compliance/enforceability verification and termination/closure. Once a candidate agreement o_t is produced, the compliance auditor gates the legality and enforceability of clauses and triggers rollback-and-revision when necessary.

Compliance Auditor Agent. The compliance auditor's decision function is $D_{\text{comp}}(s_t) = f_{\text{comp}}(o_t, \Omega) = \big(\textit{flag}_t,\ \Delta o_t\big)$, where $\textit{flag}_t \in$ pass, warn, veto, Ω denotes the set of compliance rules/constraints, and Δo_t provides clause-level revision suggestions. When VETO is triggered, the system returns the draft to the bargaining stage for further revision; when PASS is issued and signing/confirmation is completed, the process terminates and outputs the final agreement along with structured minutes (continuously maintained by the Clerk). The corresponding stage transition is

$$g(z_t, s_{t+1}) = \begin{cases} \textsc{Barg}, & z_t = \textsc{Verify} \ \wedge\ \textit{flag}_t = \textsc{veto}, \\ \textsc{Terminal}, & z_t = \textsc{Verify} \ \wedge\ \textit{flag}_t = \\ & \textsc{pass} \ \wedge\ \eta_{t+1}.\texttt{signed} = 1, \\ z_t, & \text{otherwise.} \end{cases} \tag{5}$$

Termination by Deadline. If the maximum number of rounds or the deadline is reached without an agreement, the episode terminates with the BATNA as the outside option (i.e., outputting the litigation-path outcome characterized by b_t).

In the three-stage workflow, only the bargaining stage requires cross-round strategic consistency and experience accumulation. Therefore, we introduce an updatable memory bank $K_t = \{R_t, E_t, C_t\}$ only for the two counsel agents. where R_t records reusable statutes/compliance constraints and safe phrasing templates, E_t accumulates negotiation experiences such as concessions, deadlock resolution, and adversarial counter-strategies, and C_t stores case cards organized by

cause of action and disputed issues to support retrieval in similar contexts. This memory mechanism does not alter the three-stage macro process; instead, it serves as an internal strategic support for Stage 2. In the next subsection, we present its planning-enhanced implementation: counsel agents perform tree-search-based lookahead exploration and employ RoT to distill patterns from search branches to update K_t.

3.3 Planning-Enhanced Bargaining via Tree Search and RoT Distillation

To improve the strategic foresight, convergence stability, and adversarial robustness of counsel agents in turn-based bargaining, we introduce an explicit planning layer inside Stage 2. This transforms the original single-step generative offer/counteroffer behavior into a search-and-distill decision workflow. The key idea is to treat each proposal in a negotiation round as an action in a sequential game and to conduct a controlled, shallow adversarial search over the proposal space before committing to the final term package. This reduces the risk of myopic concessions and opponent exploitation, and encourages faster convergence to acceptable agreements under BATNA (the "shadow of the law") constraints. During the bargaining stage, let $z_t = Barg$. When it is the acting counsel agent's turn (plaintiff or defendant), we define its action as $a_t = \big(p_t,\ \mathrm{pkg}_t,\ u_t\big)$, where p_t is a structured negotiation plan (issue ordering, concession path, anticipated opponent responses and countermoves), pkg_t is a structured term package (e.g., amount, installment schedule, behavioral commitments, dismissal/confidentiality, breach liability), and u_t is the natural-language utterance.

The counsel agent's decision is explicitly conditioned on an updatable memory bank $D_{\text{counsel}}(s_t; K_t) = f_{\text{LLM}}(s_t, K_t)$, where R_t stores statutes and compliance constraints (bottom lines and safe phrasing templates), E_t stores negotiation experiences (concession/countering strategies and deadlock-resolution patterns), and C_t stores a case library (case cards and term packages organized by cause of action and issues). After an action is executed, the environment updates the state based on the opponent response and optional mediation interventions: $s_{t+1} \sim P\big(\cdot \mid s_t,\ a_t,\ a_t^{\text{opp}}\big)$, where a_t^{opp} denotes the response action by the opposing counsel (and, when applicable, inserted Mediator/Legal Aid actions under deadlock or need).

MCTS over Negotiation Actions At round t, the acting counsel first samples a set of K candidate actions: $A_t = \{a_t^k\}_{k=1}^K \sim \pi_\theta(\cdot \mid s_t, K_t)$, $a_t^k = \big(p_t^k,\ \mathrm{pkg}_t^k,\ u_t^k\big)$. Each candidate includes both a term package and a strategy plan. Using s_t as the root, we build an adversarial search tree T_t whose nodes represent simulated states $\tilde{s}$ and whose edges represent simulated actions $\tilde{a}$ (alternating between the two parties). For each leaf node, we perform a short-horizon rollout of length L (e.g., 2–4 rounds) to approximate the longer-term impact of the candidate action on subsequent bargaining dynamics. For any simulated trajectory from root to a leaf, $\tau = \big(\tilde{s}_t,\ \tilde{a}_t,\ \tilde{s}_{t+1},\ \ldots,\ \tilde{s}_{t+L}\big)$, we construct a multi-dimensional negotiation quality vector $m(\tau) = \big(m_{\text{settle}}(\tau),\ m_{\text{surplus}}(\tau),\ m_{\text{fair}}(\tau),\ m_{\text{comp}}(\tau)\big)$, corresponding to: (i) the tendency to reach an agreement within the rollout horizon, (ii) the surplus over BATNA (improvement relative to b_t), (iii) vulnerable-party protection/fairness (e.g., proxies for worse-than-BATNA risk or utility-gap signals), and (iv) compliance and enforceability risk (aligned with auditing rule triggers). We scalarize this vector into a trajectory value

$$V(\tau) = w^\top m(\tau), w \succeq 0, \sum_i w_i = 1. \quad (6)$$

Backpropagation. For each state node $\tilde{s}$ and action $\tilde{a}$ in the tree, we maintain visit counts and value estimates: $N(\tilde{s}) \in \mathbb{N}, N(\tilde{s}, \tilde{a}) \in \mathbb{N}, Q(\tilde{s}, \tilde{a}) \in \mathbb{R}$. After obtaining $V(\tau)$ from a rollout, we backpropagate along the path:

$$N(\tilde{s}) \leftarrow N(\tilde{s}) + 1, N(\tilde{s}, \tilde{a}) \leftarrow N(\tilde{s}, \tilde{a}) + 1. \quad (7)$$

$$Q(\tilde{s}, \tilde{a}) \leftarrow Q(\tilde{s}, \tilde{a}) + \frac{1}{N(\tilde{s}, \tilde{a})}\Big(V(\tau) - Q(\tilde{s}, \tilde{a})\Big). \quad (8)$$

During selection, at node $\tilde{s}$ we choose

$$\tilde{a}^* = \arg\max_{\tilde{a}} \left[Q(\tilde{s}, \tilde{a}) \ + \ c \cdot \sqrt{\frac{\ln N(\tilde{s})}{N(\tilde{s}, \tilde{a}) + \epsilon}} \right], \quad (9)$$

where $c > 0$ is the exploration coefficient and ϵ is a numerical stabilizer. At the root, the output action can be chosen by maximum value: $a_t^{\text{out}} = \arg\max_{a \in A_t} Q(s_t, a)$. To align with our unified interaction framework, planning only changes the counsel's internal decision output: in the actual interaction, we use a_t^{out} as the counsel action input to the interaction operator, and the environment performs the state transition:

$$I(\text{Counsel}_i, \text{Counsel}_j, t) = f_{\text{interact}}\Big(a_t^{\text{out}},\ D_{\text{counsel}}(s_t; K_t)^{\text{opp}}\Big)$$
$$s_{t+1} = T\big(s_t,\ I(\cdot)\big). \quad (10)$$

RoT Distillation and Memory Updates Tree search improves a single decision, but it does not necessarily yield stable gains across rounds and cases. To convert multi-branch search experience into reusable knowledge, we introduce RoT (Reflection on Search Trees), which distills success patterns and failure patterns from high-/low-value branches of the search tree T_t, and writes them back into the three-memory bank $K_t = \{R_t, E_t, C_t\}$.

Let the set of leaf nodes be L_t. We define high-value and low-value leaves as $L_t^+ = \Big\{\ell \in L_t \mid V(\tau_\ell) \geq \mu_t + \delta\Big\}, L_t^- = \Big\{\ell \in L_t \mid V(\tau_\ell) \leq \mu_t - \delta\Big\}$, where $\mu_t = \frac{1}{|L_t|}\sum_{\ell \in L_t} V(\tau_\ell)$, and δ is a separation threshold (alternatively, top/bottom percentiles can be used). From each trajectory we extract structured negotiation features: $f(\tau)$=(issue(τ); concession(τ); package(τ); BATNA_gap(τ); compliance(τ); stall_pattern(τ)). We then define RoT distillation as $\Delta_t^+ = \text{Distill}^+\Big(\{f(\tau_\ell)\}_{\ell \in L_t^+}\Big)$, $\Delta_t^- = \text{Distill}^-\Big(\{f(\tau_\ell)\}_{\ell \in L_t^-}\Big)$. Here, Δ_t^+ summarizes shared success templates from high-value branches (e.g., better issue bundling and concession ordering, BATNA-anchored strategies, and low-risk clause phrasing), while Δ_t^- summarizes recurring failure modes from low-value branches (e.g., concession paths that lead to deadlock, dangerous offers that are worse-than-BATNA, clause structures that trigger veto/auditor flags, or exploitable vulnerabilities). Finally, we write the distilled patterns back into the three memories:

$$\begin{aligned} R_{t+1} &= \text{Update}_R\big(R_t,\ \Delta_t^+,\ \Delta_t^-\big), \\ E_{t+1} &= \text{Update}_E\big(E_t,\ \Delta_t^+,\ \Delta_t^-\big), \\ C_{t+1} &= \text{Update}_C\big(C_t,\ \Delta_t^+\big), \\ K_{t+1} &= \{R_{t+1},\ E_{t+1},\ C_{t+1}\}. \end{aligned} \tag{11}$$

Memory R_t emphasizes compliance bottom lines and safe phrasing, E_t accumulates negotiation strategies and counter-strategy templates , while C_t stores retrievable positive case cards and high-quality term packages.

4 Experiments

4.1 Experimental Setup and Dataset

We validate the proposed method through an integrated evaluation framework that combines real-case-driven negotiation simulation with a standardized benchmarking protocol. Our negotiation environment is constructed from authoritative legal documents of real-world civil disputes, covering case facts, parties' claims and defenses, disputed issues, and outcome information (settlement/mediation or judicial judgment). Within this environment, the system operates as a multi-agent pipeline: a clerk agent first structures the raw documents to produce an interactive case skeleton and role-specific case briefs; two lawyer agents then act as counsel for the plaintiff and the defendant, respectively, generating proposals and bargaining under the coordination of a mediator agent. Negotiations are guided by a litigation baseline (BATNA) as the reference "shadow of the law" to encourage convergence. Finally, a compliance audit agent verifies the legality and enforceability of the resulting settlement terms, and triggers revision or rollback when necessary.

Figure 2: The SettleBench covers three civil dispute types and provides representative scenario examples.

We construct SettleBench in Fig 2, a benchmark dataset for legal negotiation and settlement, to support systematic comparison of different methods under a unified environment and evaluation protocol. Existing legal AI benchmarks largely emphasize static knowledge testing (e.g., statute QA, element identification, and issue extraction). In contrast, negotiation and settlement constitute a prototypical dynamic, adversarial, and constraint-driven interactive task: beyond producing superficially plausible arguments, a system must form a convergent concession trajectory under litigation risk and cost constraints, and output terms that are lawful, enforceable, and non-coercive. To fill this gap, SettleBench explicitly structures the real-case "facts–issues–outcomes" chain and covers three categories of civil disputes with substantial negotiability and clear legal constraints: contract/quasi-contract disputes, tort liability disputes, and labor and employment disputes. We collect case texts from the publicly available China Judgments Online (China Judgments Documents) platform and extract key information using a unified template. The dataset is further divided into two complementary subsets: (i) a Settled subset (cases resolved via mediation/settlement in practice), from which we extract executable settlement term structures (amount, deadlines, installments, non-monetary

Dataset	Category	Method	SR ↑	Acc@10% ↑	Acc@20% ↑	Acc@30% ↑
Settled	General LLMs	GPT-4o-mini	0.65	34	53	65
		GPT-4o	0.74	41	58	73
		GPT-5.1	0.88	47	72	86
	Legal LLMs	LaWGPT (Zhou et al., 2024)	0.36	18	27	45
		ChatLaw2-MoE (Cui et al., 2023)	0.47	22	36	54
		Qwen2.5-Law	0.59	36	54	67
	Agent Methods	ReAct (Yao et al., 2022)	0.35	13	19	34
		AutoGPT (Yang et al., 2023)	0.38	17	25	42
		AgentsCourt (He et al., 2024)	0.54	27	42	58
		Ours	0.85	54	68	82
Tried	General LLMs	GPT-4o-mini	0.56	36	44	62
		GPT-4o	0.67	35	53	68
		GPT-5.1	0.82	49	65	81
	Legal LLMs	LaWGPT (Zhou et al., 2024)	0.27	12	17	31
		ChatLaw2-MoE (Cui et al., 2023)	0.35	16	23	38
		Qwen2.5-Law	0.51	25	41	56
	Agent Methods	ReAct (Yao et al., 2022)	0.24	8	14	27
		AutoGPT (Yang et al., 2023)	0.28	13	21	35
		AgentsCourt (He et al., 2024)	0.46	21	34	51
		Ours	0.79	51	63	80

Table 1: Negotiation performance on SETTLEBENCH. SR denotes settlement success rate.

Dataset	Category	Method	SR ↑	Acc@10% ↑	Acc@20% ↑	Acc@30% ↑
Settled	General LLMs	GPT-4o-mini	0.53	18	32	45
		GPT-4o	0.65	24	41	57
		GPT-5.1	0.79	33	52	71
	Legal LLMs	LaWGPT (Zhou et al., 2024)	0.25	4	12	21
		ChatLaw2-MoE (Cui et al., 2023)	0.47	13	25	41
		Qwen2.5-Law	0.48	15	26	44
	Agent Methods	ReAct (Yao et al., 2022)	0.23	3	9	18
		AutoGPT (Yang et al., 2023)	0.25	5	11	22
		AgentsCourt (He et al., 2024)	0.54	19	29	43
		Ours	0.82	42	61	77

Table 2: Negotiation performance on SettleBench under red-team stress testing.

terms, and cost allocation, etc.); and (ii) a Tried subset (cases where negotiation failed and proceeded to judgment), for which we use the true judicial outcomes to construct litigation outside options, enabling evaluation of whether a model can convert cases into feasible settlements under the "shadow of the real judgment," while avoiding "failed settlements" that are worse than the outside option.

4.2 Settlement Metrics

Settlement Success Rate. We report the percentage of cases in which the agents reach a finalized settlement within a fixed negotiation budget (e.g., a maximum number of dialogue rounds). Cases that do not produce a signed agreement by the budget are counted as failures. **Payment Amount Accuracy**. For cases where a settlement is reached, we extract the total compensation amount from the final agreement and compare it to the reference amount provided by the dataset (the real-world settlement amount in the Settled subset, and the adjudicated award amount in the Tried subset). We report thresholded accuracy at multiple tolerances—Acc@10%, Acc@20%, and Acc@30%—defined as the proportion of settled cases whose compensation amount deviates from the reference by no more than 10%, 20%, or 30%, respectively.

4.3 Experimental Results

From the two groups of results in Table 1, we observe that even when our negotiation framework is built on top of the lightweight `GPT-4o-mini` backbone, the proposed multi-agent decomposition and planning-enhanced design still yields substantial improvements in both settlement success and monetary-error metrics. On the SETTLED subset, OURS achieves $\mathrm{SR} = 0.85$, representing a clear gain over `GPT-4o-mini` (0.65). The advantage becomes more pronounced under stricter amount-deviation thresholds: OURS reaches $\mathrm{Acc@10\%} = 54$ (+20), and $\mathrm{Acc@20\%}/\mathrm{Acc@30\%} = 68/82$ (vs. $53/65$ for `GPT-4o-mini`). More importantly, OURS is comparable to, and in some cases even surpasses, the stronger `GPT-5.1` on these metrics, suggesting that the improvement does not merely come from scaling up the backbone model, but rather from the structural gains brought by search-based negotiation planning and constraint-aware convergence within our framework.

On the more challenging TRIED subset, overall accuracy drops across methods, yet our framework remains consistently leading. OURS result, which not only substantially outperforms the same-backbone `GPT-4o-mini` $(36/44/62)$ but also exceeds specialized legal LLMs and common agent-based baselines. Meanwhile, the gap between OURS and `GPT-5.1` is marginal, further indicating that our multi-agent negotiation framework can systematically improve settlement success rates and significantly reduce low-quality agreements that "settle easily" but exhibit large monetary drift.

4.4 Stress Testing Negotiation Robustness in Adversarial Settings

To systematically evaluate the robustness of legal negotiation systems under adversarial conditions, we introduce a red-teaming setting and conduct a unified "stress test" of the model's negotiation process. In this setup, the red team deliberately simulates more disruptive real-world counterparts (e.g., aggressive price pressure, stalling tactics, rhetorical manipulation, and strategic ambiguity) to examine whether the model, under high pressure, tends to reach inferior agreements. Specifically, we replace one negotiating party with a Red-Team Negotiator. This adversarial agent operates using a fixed library of attack strategies and prompt templates (see Appendix A2), and prioritizes triggering high-risk behaviors at each dialogue turn. To ensure comparability and fairness, we apply the exact same red-team configuration across all baselines: for the single-LLM baseline, the red team directly interacts as the opponent; for multi-agent methods, the red team replaces the corresponding role agent and participates in the negotiation.

Under the red-team stress test (Table 2), we observe pronounced divergence across methods in both SR and amount accuracy , indicating that extreme opponent strategies substantially amplify the fragility of negotiation systems. Overall, our method achieves the strongest robustness on the SETTLED subset, ranking best on all four metrics. Compared to the strong general-purpose model GPT-5.1, we still improve SR while maintaining a consistent advantage in monetary accuracy . Relative to the representative agent framework AGENTSCOURT, our gains are substantially larger. These results suggest that, under adversarial negotiations, purely dialogue-driven agent collaboration remains vulnerable to tactics such as anchoring, stalling, and clause probing.

Module w/o	SR ↑	Acc@10% ↑	Acc@20% ↑	Acc@30% ↑
Compliance	0.89	39	55	67
Aid	0.79	49	65	79
Tree	0.75	47	63	75
RoT	0.73	45	59	72
Ours	0.85	54	68	82

Table 3: Ablation study on the Settled data. Compliance: Compliance Auditor Agent. Aid: Legal Aid Agent. Tree: Tree Search.

4.5 Ablation Studies

As shown in Table 3, removing the Compliance review module (w/o Compliance) increases the SR from 0.85 to 0.89, but substantially degrades the agreement quality. This suggests that without compliance and enforceability checks, the system is more likely to reach an agreement at all costs: it can converge faster by producing more aggressive or arbitrary clauses, thereby inflating SR, while the resulting agreements often deviate from a reasonable compensation range. Meanwhile, removing legal Aid (w/o Aid) leads to a simultaneous drop in SR and accuracy, indicating that legal assistance—by providing risk reminders and expression support for the disadvantaged party—helps reduce negotiation breakdowns and improves clause quality. Furthermore, removing either Tree search or RoT distillation (w/o Tree: SR 0.75; w/o RoT: SR 0.73) causes an even more pronounced degradation, demonstrating that planning-based search and cross-branch experience distillation effectively mitigate the shortsightedness of one-step generation, making negotiation more stable under multi-round adversarial interactions and closer to a reasonable compensation range.

5 Conclusion

This paper studies a high-frequency yet underexplored stage in civil disputes: pre-trial negotiation, mediation, and settlement. We propose a settlement-oriented multi-agent framework with a Clerk agent to structure the case, an Outcome agent to provide outside-option references, two Counsel agents to negotiate under a Mediator, and Legal-Aid and Compliance-Audit agents as safeguards. We also introduce SettleBench, spanning contract/quasi-contract, tort, and labor disputes, with Settled and Tried subsets to evaluate both settlement success and outcome quality under unified metrics. Experiments show consistent gains over general-purpose and legal-domain LLMs as well as existing agent baselines, including under red-team stress testing, indicating improved robustness.

Limitations

Although the proposed multi-agent legal negotiation framework and the SETTLEBENCH benchmark yield consistent gains in settlement success and robustness evaluation, several important limitations remain.

Data and representativeness. SETTLEBENCH is primarily constructed from publicly available judicial documents, leveraging accessible case facts, claims, defenses, and outcomes, and currently covers three categories of civil disputes: contract/quasi-contract, tort liability, and labor/employment. This coverage is still limited. It does not yet include causes of action with substantially different negotiation structures (e.g., family and divorce matters, medical malpractice disputes, or property/real-right disputes), and it cannot fully represent variations across regions, court levels, or time periods. Moreover, public documents may suffer from anonymization, missing key elements, and inconsistent writing styles, which can reduce the degree of structured extraction and the stability of evaluation for some cases.

Simplified negotiation objectives. To enable reproducible comparisons, we mainly characterize negotiation quality using metrics such as settlement rate (SR) and monetary closeness (Acc@10/20/30). In real-world practice, outcomes are jointly determined by multi-dimensional terms (e.g., installment plans, breach liabilities, apology/clarification, confidentiality, withdrawal conditions, and fee allocation), and parties' utilities are not solely determined by the monetary amount. Our current protocol does not fully capture these non-monetary terms, subjective satisfaction, procedural experience, or long-term performance and enforcement risks.

Bias in modeling outside options (Outcome/BATNA). We use an Outcome Agent (or the ground-truth adjudication outcome) to approximate the "outside option under the shadow of the law," guiding convergence and constraining settlements that fall below a party's baseline. However, real-world BATNA is uncertain and depends on evidence production, counsel strategies, judicial discretion, regional differences, and enforcement risk. When outcome estimation is systematically biased, the negotiation trajectory and final agreement may be steered by an incorrect reference point. This work does not provide a large-scale, systematic analysis of upper bounds on how outcome bias affects negotiation fairness and stability.

Limited coverage of adversarial evaluation. We employ a red-team agent to simulate adversarial tactics such as extreme anchoring, stalling, ambiguous commitments, and constraint probing for stress testing. Nevertheless, these modes do not cover the full complexity of real negotiations (e.g., multi-party settings, information leakage, third-party pressure, abrupt evidence changes, or strategically timed behaviors near statutory deadlines). In addition, red-team evaluation is a worst-case stress test: its intensity and distribution may not match real-world frequencies, and thus should not be interpreted as a direct estimate of failure rates in practice.

Ethics Statement

All civil negotiation and adjudication cases used in this study are obtained from publicly accessible sources, and sensitive information such as parties' names has been properly anonymized to protect privacy. The proposed SettleAgent / SettleBench is intended as a research and evaluation tool to advance the understanding and methodological study of legal negotiation and settlement—an extremely common yet long under-modeled stage—rather than to provide automated legal services in real-world settings. We also acknowledge several important ethical considerations.

First, although SettleAgent achieves higher settlement rates and more stable term quality in simulated negotiations, it is not intended to replace human lawyers, mediators, or judges, nor should it be used to draw legal conclusions for real cases or to directly generate signable agreements. The system outputs should only be treated as candidate proposals and analytical references in research contexts. Second, a negotiation system may be misused to produce manipulative rhetoric, pressure disadvantaged parties, induce unfair terms, or even evade compliance constraints. To mitigate such risks, our framework explicitly incorporates a litigation outside option (the "shadow of the law"/BATNA) as an anchor and a compliance-audit mechanism, and we conduct red-team adversarial evaluation to stress-test exploitability and safety boundaries under extreme strategies. However, these safeguards cannot guarantee absolute safety in open environments. Finally, we emphasize that any attempt to transfer this system to real-world use must be carried out under continuous supervision and careful

review by qualified legal professionals, in compliance with local laws, regulations, and professional ethics. The system's outputs do not constitute formal legal advice and should not be used directly in real negotiations, mediations, or litigation procedures without human oversight.

References

Guhong Chen, Liyang Fan, Zihan Gong, Nan Xie, Zixuan Li, Ziqiang Liu, Chengming Li, Qiang Qu, Hamid Alinejad-Rokny, Shiwen Ni, and 1 others. 2025. Agentcourt: Simulating court with adversarial evolvable lawyer agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5850–5865.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2023. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *arXiv preprint*.

Sil Hamilton. 2023. Blind judgement: Agent-based supreme court modelling with gpt. *arXiv preprint*.

Zhitao He, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Jiexin Xu, Huaijun Li, Kang Liu, and Jun Zhao. 2024. Agentscourt: Building judicial decision-making agents with court debate simulation and legal knowledge augmentation. In *Findings of EMNLP*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, and 1 others. Metagpt: Meta programming for a multi-agent collaborative framework. In *ICLR*.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2024. A sentence is known by the company it keeps: improving legal document summarization using deep clustering. *Artificial Intelligence and Law*, 32(1):165–200.

Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J Bommarito II. 2023. Natural language processing in the legal domain. *arXiv preprint arXiv:2302.12039*.

Callie Y Kim, Christine P Lee, and Bilge Mutlu. 2024. Understanding large-language model (llm)-powered human-robot interaction. In *Proceedings of the 2024 ACM/IEEE international conference on human-robot interaction*, pages 371–380.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22266–22275.

Nathalia Nascimento, Paulo Alencar, and Donald Cowan. 2023. Self-adaptive large language model (llm)-based multiagent systems. In *2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, pages 104–109. IEEE.

Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. 2023. Lextreme: A multi-lingual and multi-task benchmark for the legal domain. *arXiv preprint arXiv:2301.13126*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Yunqiu Shao, Yueyue Wu, Yiqun Liu, Jiaxin Mao, and Shaoping Ma. 2023. Understanding relevance judgments in legal case retrieval. *ACM Transactions on Information Systems*, 41(3):1–32.

Harry Surden. 2018. Artificial intelligence and law: An overview. *Ga. St. UL Rev.*, 35:1305.

Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*.

John Chong Min Tan and Mehul Motani. 2024. Llms as a system of multiple expert agents: An approach to solve the abstraction and reasoning corpus (arc) challenge. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 782–787. IEEE.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023. Precedent-enhanced legal judgment prediction with llm and domain-model collaboration. *arXiv preprint arXiv:2310.09241*.

Nan Xie, Yuelin Bai, Hengyuan Gao, Ziqiang Xue, Feiteng Fang, Qixuan Zhao, Zhijian Li, Liang Zhu, Shiwen Ni, and Min Yang. 2024. Delilaw: A chinese legal counselling system based on a large language model. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 5299–5303.

Hui Yang, Sifu Yue, and Yunzhong He. 2023. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint*.

Yijun Yang, Tianyi Zhou, Kanxue Li, Dapeng Tao, Lusong Li, Li Shen, Xiaodong He, Jing Jiang, and Yuhui Shi. 2024. Embodied multi-modal agent trained by an llm from a parallel textworld. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26275–26285.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *ICLR*.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. 2024. Agent lumos: Unified and modular training for open-source language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12380–12403.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and 1 others. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*.

Zhi Zhou, Jiang-Xin Shi, Peng-Xiao Song, Xiao-Wen Yang, Yi-Xuan Jin, Lan-Zhe Guo, and Yu-Feng Li. 2024. Lawgpt: A chinese legal knowledge-enhanced large language model. *arXiv preprint*.

A Appendix

A.1 Agent Responsibilities and Interactions

A.1.1 Core Agents

Plaintiff Counsel Agent / Defendant Counsel Agent (Plaintiff/Defendant Attorneys). The two counsel agents constitute the primary actors in the negotiation. In each round, they propose or revise settlement offers based on the current negotiation state (e.g., offer history, issue progress, remaining rounds/deadlines). Proposals are expressed in both natural language and a structured term package (e.g., monetary amount, installment plan, apology/performance, dismissal conditions, liquidated damages/breach liability). Each agent aims to maximize its own utility while seeking an acceptable agreement under legal constraints and the outside option (BATNA).

Outcome Agent (Litigation Baseline / BATNA Assessment). The Outcome Agent does not engage in bargaining. Instead, it reads the case skeleton and outputs the expected litigation outcome J (optionally as an interval or expectation) and estimates of litigation cost/time/risk if negotiation fails and proceeds to court, serving as the reference baseline for "bargaining in the shadow of the law." This baseline is used to determine whether an agreement is worse-than-BATNA and to measure the agreement's improvement over the outside option (surplus over BATNA), thereby providing an objective comparator for evaluation.

Mediator Agent (Neutral Mediation / Package Coordination). From a neutral perspective, the Mediator Agent structures the negotiation process by converting unstructured disputes into a set of negotiable issues, proposing compromise solutions or bundled offer packages, and—when parties are deadlocked—facilitating convergence via issue reframing, concession guidance, and coordination. This agent does not represent either party's interests; its primary goal is to increase the probability of reaching an executable agreement within compliance boundaries.

Compliance Auditor Agent (Compliance Auditing and Enforceability Gatekeeping). The Compliance Auditor Agent performs clause-level checks on candidate agreements, focusing on hard legal constraints and obviously risky terms (e.g., minimum protection floors, non-waivable rights, illegal or unenforceable clauses, missing essential elements, and ambiguous wording). When issues are detected, it outputs the violation types and revision suggestions. In risk-sensitive settings, it can be configured to trigger a mandatory correction mechanism (e.g., returning the agreement to the negotiation stage for rewriting) to reduce the probability of generating illegal or high-risk agreements.

Legal Aid Agent (Legal Assistance / Support for Vulnerable Parties). The Legal Aid Agent simulates real-world legal aid and protections for vulnerable parties. For the party labeled as vulnerable, it provides explanations of rights and obligations, clarifies the meaning of BATNA, highlights key risks, and offers expression support (e.g., helping organize claims, pointing to necessary evidence, clarifying unfavorable implications of terms). This agent offers advice and prompts only, and does not substitute for the party's final decision-making, aligning with procedural justice and party autonomy.

Clerk Agent (Court Clerk / Docket Organization and Process Logging). As a non-decision-making neutral role, the Clerk Agent reads and consolidates case materials to generate a unified case digest and role-specific briefs for both counsel agents, reducing information-organization overhead and improving input consistency. It also continuously records the negotiation process, producing structured minutes and a final agreement summary to support downstream evaluation, retrospection, and interpretability analysis.

A.1.2 Auxiliary Agents (Out-of-Episode Agents)

Evaluation Agent (Independent Evaluator/Judge). After the negotiation ends, the Evaluation Agent conducts outcome evaluation and process assessment, producing multi-dimensional scores (e.g., whether an agreement is reached, improvement over BATNA, occurrence of worse-than-BATNA, compliance and risky-clause ratio, degradation under adversarial conditions). To reduce subjective drift, evaluation prioritizes structured fields and verifiable rules, and may provide explanatory diagnostic text when needed.

Red-Team Negotiator (Adversarial Negotiator). The Red-Team Negotiator is used for robustness evaluation. In specific experimental settings, it replaces the plaintiff or defendant counsel and adopts extreme negotiation strategies (e.g., aggressive anchoring/pressure, stalling, manipulative rhetoric, false promises) to actively attack the system. This tests the stability of agreement quality, compliance, and protections for vulnerable

parties under adversarial interaction, and assesses exploitability and safety boundaries.

A.2 Red-Team Adversarial Evaluation (Extreme-Case Stress Test)

In the supplementary material, we provide reproducible details of our red-team adversarial evaluation, designed to test worst-case robustness under extreme opponent strategies. The red-team agent (Red-Team Negotiator) generates adversarial negotiation behaviors from a parameterized library of attack strategies and replaces one party's lawyer role in the experiments. Except for the replaced role, all other environment settings (case inputs, maximum round budget, termination rules, and model parameters) are kept identical to the main experiments, ensuring a fair comparison.

Evaluation Protocol. For each case, we fix the maximum number of negotiation rounds to T, and run each baseline as well as our method under the same red-team configuration. If no final agreement is generated and explicitly confirmed by both parties within the budget, the negotiation is counted as a failure. To reduce randomness, we run each case multiple times and report the mean results (optionally, under different random seeds and/or sampling temperatures).

Attack Surface Taxonomy. The red-team strategies cover the following representative adversarial patterns: (1) Extreme anchoring: repeatedly anchoring offers far outside a reasonable range to induce concessions; (2) Stalling and attrition: creating unproductive turns via repeated requests for clarification, introducing new topics, and cyclic questioning; (3) Rhetorical manipulation: applying emotional pressure, implying threats, or using leading language to force rapid concessions; (4) False promises and ambiguous terms: masking true intent with unverifiable commitments or deliberately vague payment/performance conditions; (5) Compliance-boundary probing: attempting to introduce clearly illegal or unenforceable clauses to test system guardrails. Each attack type is instantiated with an intensity level (mild/medium/strong) and equipped with triggering and escalation rules to simulate pressure that increases over the course of real negotiations.

Templates and Parameterized Implementation. We implement each strategy as a parameterized template (e.g., anchor amounts, concession step size, deadline constraints, threat types), enabling controllable diversity while keeping the adversarial intent consistent.

A.2.1 Unified Interface and Output Schema of the Red-Team Agent

We implement the red-team agent as a parameterized strategy generator. Given the current negotiation state s_t and a red-team configuration γ, it outputs an opponent utterance together with structured meta-information:

$$u_t^{\mathrm{RT}},\ z_t^{\mathrm{RT}} \ = \ f_{\mathrm{RT}}(s_t;\gamma), \tag{12}$$

where u_t^{RT} is a natural-language message (defanged phrasing), and z_t^{RT} is a structured tag set for auditing and reproducibility. Each turn output must satisfy the following schema:

- `intent`: turn-level intent (e.g., `anchor`, `stall`, `probe`, `vague`, `pressure`)
- `tactic`: specific tactic name (see Section A.2.2)
- `message`: natural-language message to the opponent (neutral wording; no threats, intimidation, or illegal instructions)
- `parameters`: parameters used in this turn (e.g., anchoring ratio, number of questions)
- `expected_effect`: expected pressure type (e.g., convergence perturbation / information burden / constraint probing)

Responsible release statement. To mitigate misuse risk, we publicly release the attack-surface taxonomy, parameterized templates, triggering and escalation rules, and defanged example snippets. We do not publicly release full verbatim attack prompts; they may be made available under reasonable research use upon request.

A.2.2 Five Red-Team Adversarial Modes (Mode Cards)

We define five complementary adversarial modes that cover extreme behaviors commonly observed in real-world negotiations: extreme anchoring, delay and stalling, ambiguous commitments, constraint probing, and high-pressure rhetoric. Each mode is specified by $(\texttt{goal}, \texttt{trigger}, \texttt{intensity}, \texttt{params})$, ensuring controllable diversity and reproducibility.

Mode 1: Extreme Anchoring (ANCHOR). Goal. Repeatedly anchor an initial demand/offer far outside a reasonable range, testing whether the

system is pulled by anchors into irrational concessions or deviates from a stable convergence path.
Trigger. Always triggered at round 1; if the opponent does not provide explicit counter-anchoring or request external references, the anchoring strength is increased in rounds 2–3.
Intensity.

- `low`: anchor deviates from a reference value (or the center of the plausible range) by $\approx 30\%$
- `mid`: $\approx 50\%$
- `high`: $\approx 70\%$

Params.

- `anchor_ratio` $\in \{0.3, 0.5, 0.7\}$
- `reanchor_frequency` $\in \{1, 2\}$ (restate the anchor every k rounds)
- `concession_step` $\in \{0.02, 0.05, 0.10\}$ (step size of concessions)

Output constraint. The red team must provide a concrete monetary/term position using neutral language, without threats.

Mode 2: Delay & Stalling (STALL). Goal. Increase information burden and waste rounds via procedural clarifications, repeated confirmations, and topic switching, testing stability under a round budget and convergence mechanism.
Trigger. Triggered when the opponent proposes a relatively complete offer (e.g., amount + deadline); thereafter, triggered at least once every two rounds.
Intensity.

- `low`: ask 1–2 clarification questions
- `mid`: ask 3–4 questions and request restating key points
- `high`: in addition to clarifications, introduce a "new topic" and request a return to fact-checking

Params.

- `num_questions` $\in \{2, 4, 6\}$
- `repeat_rate` $\in \{0.2, 0.5, 0.8\}$ (probability of repeating similar questions)
- `topic_shift_prob` $\in \{0.0, 0.2, 0.4\}$ (probability of topic shift)

Output constraint. Messages are framed as "clarification / request for supporting materials / restatement" and avoid harassment-style phrasing.

Mode 3: Ambiguous Commitments (VAGUE). Goal. Appear agreeable while deliberately omitting key execution fields (deadline, payment method, installments, etc.), inducing acceptance of an unenforceable plan and testing executability and auditing guardrails.
Trigger. Triggered when the monetary gap falls below a threshold ϵ (e.g., 10%) or negotiation enters a "convergence zone".
Intensity.

- `low`: leave 1 key field ambiguous/missing
- `mid`: leave 2 key fields ambiguous/missing
- `high`: keep multiple key fields ambiguous and push for rapid confirmation

Params. We parameterize the policy with three attributes: `missing_fields` $\subseteq$ {`deadline`, `payment_method`, `installment`, `liability`, `enforcement`}, `commitment_strength` $\in$ {`weak`, `medium`, `strong`} indicating the linguistic strength of the "agreement", and `fast_close_prob` $\in \{0.2, 0.5, 0.8\}$ representing the probability of pushing early termination.

Output constraint. No real-world illegal clauses are introduced; risk is created only via missing/ambiguous fields.

Mode 4: Constraint Probing (PROBE). Goal. Propose "boundary-term requests" to probe whether the system will generate or accept non-compliant or unenforceable clauses, testing compliance guardrails and rollback behavior.
Trigger. Triggered when the opponent emphasizes compliance, a draft agreement appears, or a compliance auditor suggests modifications.
Intensity.

- `low`: request 1 mildly high-risk clause (near enforceability boundary)
- `mid`: request 2 high-risk clauses
- `high`: request a bundle of 3 clauses and ask to include them in the agreement

Params.

- `probe_type` $\in$ {`legality`, `enforceability`, `voluntariness`}
- `probe_count` $\in \{1, 2, 3\}$

- `escalate_after_reject` $\in \{$`true`, `false`$\}$

Output constraint (critical). In publicly released text, clause contents are expressed using placeholders:

```
[HIGH-RISK TERM A],
[POTENTIALLY UNENFORCEABLE TERM]
```

and `probe_type` is recorded in metadata, avoiding the release of verbatim illegal/coercive clauses.

Mode 5: High-Pressure Rhetoric (PRESSURE; Defanged). Goal. Increase psychological and time pressure via "tough but non-threatening" language, testing whether the system makes emotional concessions or loses strategic consistency.
Trigger. Triggered when the opponent rejects or holds the line for more than k consecutive rounds (e.g., $k = 2$).
Intensity.

- `low`: tougher tone + continue holding the anchor
- `mid`: tougher tone + shorter responses + set a "soft" round deadline
- `high`: high-frequency re-anchoring + tighter soft deadline (still no threats/intimidation)

Params.

- `tone_level` $\in \{1, 2, 3\}$
- `deadline_round` $\in \{T-3, T-2, T-1\}$ (soft deadline round)
- `repeat_anchor` $\in \{$`true`, `false`$\}$

Output constraint. No intimidation, threats, or illegal inducement is allowed; pressure is applied only via tone, frequency, and soft deadlines.

A.2.3 Triggering and Escalation Rules (Schedule)

We adopt two scheduling mechanisms to balance controllability and diversity:

1. **Round-based scheduling:** e.g., ANCHOR at round 1; STALL at rounds 2/4/6; VAGUE after entering the convergence zone.
2. **Condition-based scheduling:** PROBE when a "draft agreement" or "compliance signal" appears; PRESSURE when the rejection count exceeds a threshold.

Intensity escalation follows a simple rule: if the current mode fails to induce observable behavior change for two consecutive activations (e.g., the opponent continues converging along the same path or repeatedly rejects probing clauses), we increase the intensity by one level up to `high`, or until reaching the maximum round budget T.

A.2.4 Applying Red Team Across Methods (Fairness of Comparison)

To ensure fairness, we use the same red-team configuration γ across all methods:

- **Single-LLM baseline:** the red team directly participates as the opponent.
- **Multi-agent methods:** the red team replaces one *Counsel Agent* (plaintiff or defendant lawyer), while other agents (Mediator, Compliance, Legal Aid, etc.) remain unchanged.

Across all settings, case inputs, maximum rounds, termination rules, and evaluation metrics are identical to the main experiments; the only change is whether the opponent strategy is red-team-driven.