LAWBRANCH: BRANCH-AWARE EVALUATION OF MULTI-TURN LEGAL CONSULTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) hold promise for making legal consultation more accessible. In realistic legal consultation, however, user queries are often initially underspecified and incomplete, so legal experts must ask clarification questions over multiple turns to elicit missing legal facts. Since different answers can lead to substantially different legal reasoning and advice, existing benchmarks, which typically evaluate a single fixed scenario per case and report average-case metrics, obscure failures on consequential factual branches. Meanwhile, cloud-based LLMs achieve strong performance but incur high costs and privacy risks, whereas smaller local models are more affordable yet typically lack sufficient legal-reasoning capabilities. To address these gaps, we introduce **LawBranch**, a branch-aware Chinese civil-law benchmark comprising 1,500 human-verified cases and 5,399 factual branches. Further analysis reveals that both clarification and advice quality degrade consistently in multi-turn dialogue, while worst-branch reliability is at least 40% lower than the corresponding average-branch metric. We therefore propose Clarification-First Edge-Cloud Collaboration (**CFCol**), where a cloud agent handles clarification planning and legal advice generation, while an on-device agent and a memory manager are responsible for de-identification and dialogue coordination. Across diverse cloud and edge LLM instantiations, CFCol improves both clarification and advice performance, reduces cloud cost by at least 25%, and offers privacy advantages over cloud-only deployment.

1 INTRODUCTION

Recent advances in large language models (LLMs) have accelerated their adoption across diverse domains (Achiam et al., 2023; Qwen Team, 2026; Guo et al., 2025). In legal applications, automatic legal consultation (CAIL2023, 2023; CAIL2024, 2024; Li et al., 2025; Gan et al., 2025) requires an LLM to infer the legal needs underlying a user's real-world problem and to provide legally grounded, practically actionable advice. This task is appealing both for broadening access to affordable legal assistance and providing a realistic setting for evaluating LLMs.

Prior legal consultation benchmarks (Fei et al., 2024; Dai et al., 2025) adopt a single-turn question–answer format, pairing each user query with a short expert response. However, user queries are often initially underspecified, forming what we term a meta-case that requires legal experts to elicit critical facts through follow-up questions (Li et al., 2025). Concurrent work addresses this limitation by constructing multi-run evaluation environments to assess models' ability to ask effective clarification questions and the generated advice quality (Jia et al., 2026; Zhang et al., 2026; Yuan et al., 2026).

Nevertheless, this line of work largely overlooks the counterfactual structure of legal consultation. As illustrated in Fig. 1, a meta-case can be instantiated into multiple branch cases by specifying relevant facts and user objectives, each of which may involve distinct legal relationships, reasoning paths, and conclusions. This property is central to both legal education and legal practice, where hypothetical variations in case facts are routinely used to examine how legal outcomes shift across factual scenarios (Ashley, 2006). A practical benchmark should therefore also evaluate whether a model remains reliable across factual branches of the same meta-case beyond isolated scenarios.

Existing approaches either prompt proprietary cloud LLMs (Yao et al., 2022) or post-train smaller local models (He et al., 2023; Li et al., 2026; Yue et al., 2024). The former is stronger but costly

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

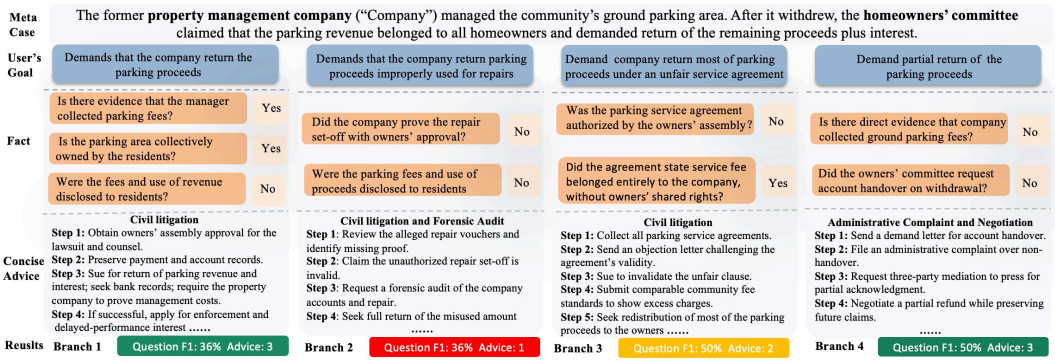


Figure 1: Illustrative meta-case from the property management domain in our LawBranch. The figure shows four possible branches, each containing the user objective, necessary facts, expert advice and evaluation scores of advice generated by GPT-5.4. It highlights that average performance can overestimate poor outcomes in specific scenarios.

over long dialogues and risks exposing sensitive client information to model providers[1], hindering real-time deployment, while the latter is cheaper locally but has a substantial capability gap.

To address these challenges, we introduce **LawBranch**, a branch-aware benchmark for multi-turn legal consultation. LawBranch is constructed via a generate–verify–revise pipeline in which LLMs transform source-traceable Chinese civil disputes into meta-cases with multiple factual branches, followed by verification from legally qualified annotators and revision with LLMs. Using LawBranch, we systematically evaluate various cloud and edge LLMs and find that multi-turn interactions degrade both clarification ability and advice quality, especially under realistic user interventions, aligning with prior work (Laban et al., 2026), while worst-branch reliability is substantially lower than the mean performance, highlighting the need for branch-aware evaluation. To address privacy and cost concerns and to mitigate the degradation observed in multi-turn interaction, we propose **Clarification-First Edge–Cloud Collaboration** (**CFCol**), comprising a cloud agent, an on-device agent, and a memory manager. The on-device agent de-identifies user queries and obtains a clarification plan from the cloud. Following this plan, it then interacts with the user and updates the memory manager, which maintains recorded facts and unresolved questions. Once sufficient information is collected, it sends de-identified context to the cloud to generate final advice. This design avoids the performance degradation of cloud LLMs caused by multi-turn dialogue history, while enhancing the on-device model's dialogue coordination via the memory manager.

In summary, our main contributions are:

- We introduce LawBranch, a branch-aware benchmark for multi-turn legal consultation, comprising 1,500 human-verified cases and 5,399 factual branches.
- We systematically evaluate various LLMs on LawBranch, showing that multi-turn dialogue degrades performance and that average-case metrics overestimate real-world reliability.
- We propose CFCol, a privacy-preserving edge–cloud framework that reduces cloud cost and privacy leakage while improving performance and reliability.

2 Related Work

Legal Consultation Benchmark. Legal consultation requires LLMs to infer legal needs from underspecified user queries, elicit missing facts through effective multi-turn clarification, and provide legally grounded advice. Existing benchmarks such as LawBench (Fei et al., 2024) and LAiW (Dai et al., 2025) are primarily built on single-turn (user-query, expert-response) pairs collected from online legal consultation platforms. More recent work extends evaluation to multi-turn settings by assessing the quality of clarification and advice (Yuan et al., 2026; Jia et al., 2026; Zhang et al., 2026). For example, LeCoDe (Yuan et al., 2026) transcribes livestreamed lawyer-client consultations and

[1]Legal practitioners are obligated to safeguard client information when using AI systems; see, e.g., `https://legal.thomsonreuters.com/blog/generative-ai-and-aba-ethics-rules/`.

augments them with expert annotations of key facts and advice summaries. However, these benchmarks largely treat legal consultation scenarios as isolated instances, overlooking the counterfactual nature of the task, where variations in user-provided facts and goals can lead to different reasoning paths and conclusions. As a result, they may overestimate model reliability in real-world deployment. To address this limitation, we propose LawBranch, a benchmark designed to evaluate whether a model remains reliable across factual branches of the same meta-case in legal consultation.

Legal Consultation Framework. Existing legal consultation frameworks can be broadly divided into training-free and post-training-based approaches. Training-free methods mainly rely on prompting general-purpose LLMs (Li et al., 2024; Gan et al., 2025). To improve performance, some augment these models with retrieval from external legal corpora such as statutes and legal articles, although the gains depend heavily on the quality of both the retriever and the corpus (Li et al., 2025). Other methods adopt agent-based designs (Yao et al., 2022; Suri et al., 2026; Yang et al., 2026). While these proprietary (cloud) LLMs often achieve strong performance, repeated processing of long dialogue histories can be costly and may raise privacy concerns by exposing sensitive client information to external providers. Post-training approaches (He et al., 2023; Li et al., 2026; Yue et al., 2024) instead adapt smaller edge models using legal-domain data for efficient local deployment. However, these post-trained small models still suffer from limited capabilities in realistic consultation settings, particularly in both clarification and advice generation. Therefore, we propose CFCol, a clarification-first edge–cloud framework that combines the strengths of both categories.

3 BRANCH-AWARE MULTI-RUN LEGAL CONSULTATION BENCHMARK

This section introduces LawBranch, a branch-aware benchmark motivated by two observations: users often lack professional legal knowledge, and a meta-case can yield multiple factual branches depending on the user's goals and the facts provided. We present the benchmark construction pipeline and the evaluation framework and protocol, with further details in Appendix A.

3.1 GENERATE–VERIFY–REVISE CONSTRUCTION PIPELINE

Constructing LawBranch is challenging because multi-turn legal consultation dialogues are scarce, and existing resources, such as online consultation videos and judicial documents, typically process only one branch of the same case. We formalize a meta-case as $M = (x, \mathcal{B})$, where x is a background description of the user's initial situation and $\mathcal{B} = \{b_k\}_{k=1}^{K}$ is a set of K factual branches. Each branch $b_k = (o_k, \mathcal{F}_k, s_k)$ comprises a legal objective o_k, a set of K_k^F branch-specific question–answer pairs $\mathcal{F}_k = \{(q_{k,j}, a_{k,j})\}_{j=1}^{K_k^F}$, and reference structured legal advice s_k (recommended solution, applicable conditions, procedural steps, and precautions). K_k^F may vary across branches. Through discussions with legal practitioners, we find that manually drafting such meta-cases is costly and infeasible, so we develop an automated generate–verify–revise pipeline for benchmark construction, as illustrated in Fig. 2(a).

Generate. We first construct a multi-level taxonomy of legal consultation scenarios to guide the generation of diverse cases. The taxonomy consists of broad top-level domains, finer-grained intermediate classes, and concrete leaf categories. Top-level domains are identified from Chinese legal consultation websites[2] and cover major areas of civil legal consultation, while intermediate and leaf categories are generated by prompting Qwen3.6-Max (Qwen Team, 2026). For example, the consumer disputes domain includes the intermediate class of general e-commerce, which further contains the leaf category of false advertising. This yields 13 top-level domains, approximately 80 intermediate classes, and 756 leaf categories. Although our study focuses on Chinese civil law, the pipeline is readily adaptable to legal consultation scenarios under other legal systems.

For each leaf category, we use Qwen3.6-Max with web search to retrieve 10 real candidate cases, and then use the same model to convert each case into a structured consultation meta-case after removing private information. We apply BGE-M3-based (Chen et al., 2024) semantic deduplication to remove near-duplicates and randomly sample 1,500 meta-cases while balancing the domain distribution. Each sample follows the meta-case formulation $M = (x, \mathcal{B})$.

[2] `https://www.66law.cn/` and `https://www.imlaw.cn/`.

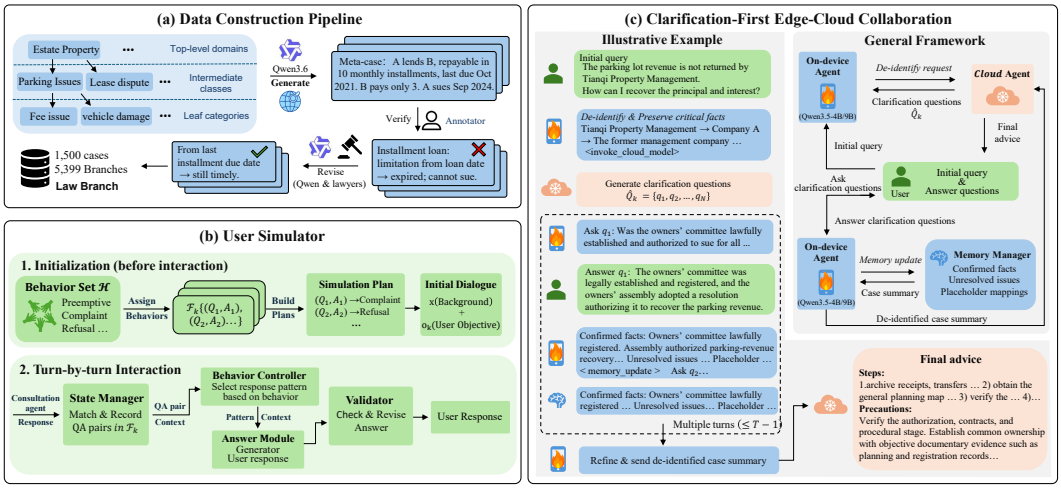


Figure 2: (a) Generate–Verify–Revise pipeline for constructing human-verified meta-cases and factual branches. (b) User simulator with behavior assignment, state matching, behavior control, answer generation, and validation for realistic multi-turn consultation evaluation. (c) CFCol framework, where the on-device agent performs de-identification and dialogue coordination with the help of the memory manager, and the cloud agent provides clarification questions and final advice.

Verify. To improve data quality, we recruit five law graduates who have passed the National Legal Professional Qualification Examination and have law firm internship experience. Following a unified annotation protocol, they review the validity of factual branches, the necessity of clarification questions, and the practicality of the legal advice, and provide revision suggestions. Each case is independently reviewed by two annotators. We find that approximately 80% of the sampled cases contain at least one issue, underscoring the importance of human verification, and double annotation substantially reduces annotation error in Appendix A.5.

Revise. We use Qwen3.6-Max to incorporate the two annotators' suggestions with minimal edits, preserving the original meta-case structure. For quality control, two practicing lawyers then conduct a final review and manually correct each case when necessary. This yields 1,500 human-verified meta-cases covering 5,399 factual branches, split into 440 training, 750 in-domain (ID) test, and 310 out-of-domain test (OOD) cases, with the OOD set drawn from top-level domains excluded from training and ID test.

3.2 Evaluation Framework and Protocol

To simulate realistic legal consultation, we build a user simulator with diverse behaviors. We further propose an evaluation protocol that assesses models' ability to clarify and provide legal advice, as well as their reliability across factual branches of the same meta-case. We describe using a meta-case $M = (x, \mathcal{B})$ and its factual branch $b_k = (o_k, \mathcal{F}_k, s_k)$.

User Simulator and Consultation Agent. The consultation agent interacts with the simulator turn by turn, adaptively asking clarification questions or providing advice based on the simulator's feedback. Following Yuan et al. (2026), we allow at most one clarification question per turn to better approximate real-world consultation. To model realistic user behavior, the simulator consists of a behavior set $\mathcal{H}$, a state manager, a behavior controller, an answer module, and a validator. Following recent benchmarks (Jia et al., 2026; Zhang et al., 2026), $\mathcal{H}$ specifies diverse user behaviors and is summarized in Table 6 in Appendix. As shown in Fig. 2 (b), before interaction, the simulator assigns a behavior from $\mathcal{H}$ to each question–answer pair in $\mathcal{F}_k$ and initializes the dialogue with a query containing the background x and user objective o_k. After each agent response, the state manager matches the query to a QA pair in $\mathcal{F}_k$ and records the result. The behavior controller then selects the corresponding response pattern, and the answer module generates the user's response, indicating unknown if no pair matches. The validator ensures consistency with the assigned behavior and revises when necessary. Upon reaching the maximum turn T, the agent must output its final advice.

Clarification Ability. We evaluate clarification quality using Recall (R), Precision (P), and F1 (F). Let $\widehat{\mathcal{Q}}_k$ denote the clarification questions asked by the agent for branch $b_k = (o_k, \mathcal{F}_k, s_k)$. Since $\widehat{\mathcal{Q}}_k$ and $\mathcal{F}_k$ are not directly comparable, we use an LLM-based semantic matcher to identify which generated questions successfully elicit the branch-specific facts in $\mathcal{F}_k$. Let $\mathcal{C}_k \subseteq \mathcal{F}_k$ denote the covered question–answer pairs. We define the branch-level metrics as

$$R_k = \frac{|\mathcal{C}_k|}{|\mathcal{F}_k|}, \qquad P_k = \frac{|\mathcal{C}_k|}{|\widehat{\mathcal{Q}}_k|}, \qquad F_k = \frac{2R_kP_k}{R_k + P_k}, \tag{1}$$

where recall R is particularly important because it determines how many necessary facts are elicited.

Advice Quality. We evaluate final legal advice using an LLM-as-a-judge protocol with a four-level rubric designed by legal experts. For each branch $b_k = (o_k, \mathcal{F}_k, s_k)$ in a meta-case $M = (x, \mathcal{B})$, the judge is given the background x, the user's legal objective o_k, the complete factual set $\mathcal{F}_k$, the reference solution s_k, and the generated advice $\hat{s}_k$. It compares $\hat{s}_k$ with s_k across four dimensions: legal relationship, legal application, legal reasoning, and legal procedure, thereby approximating an expert legal assessment. It then assigns a score $z_k \in \{0, 1, 2, 3\}$, where 0 denotes legally misleading or potentially harmful advice, 1 denotes major defects, 2 denotes advice that is generally correct but incomplete or insufficiently actionable, and 3 denotes correct, complete, and actionable advice.

Reliability.A reliable consultation agent should perform well across all branches of a meta-case, not only on favorable ones. We therefore define meta-case-level reliability as the worst-branch clarification and advice performance across the K branches:

$$r_q = \min_{1 \le k \le K} F_k, \qquad r_a = \min_{1 \le k \le K} z_k. \tag{2}$$

4 Benchmark Analysis

We conduct four complementary analyses. First, in an idealized single-run setting, we isolate clarification and advice generation to study model family, scale, reasoning, and web search. Second, we compare idealized and realistic multi-turn consultation, analyze failure modes, and assess robustness to diverse user behaviors. Third, we evaluate reliability across factual branches of a meta-case.

Idealized Single-Run Analysis. To isolate the two core capabilities of legal consultation, we evaluate clarification and advice generation separately in an idealized single-run setting. For each meta-case $M = (x, \mathcal{B})$ and a corresponding branch $b_k = (o_k, \mathcal{F}_k, s_k)$, we assess clarification by providing x and the user's legal objective o_k and asking the model to generate at most 9 clarification questions. We assess advice quality by providing x, o_k, and the complete fact set $\mathcal{F}_k$ and asking for a candidate legal advice. We compute all metrics following Sec. 3.2. Experiments use the GPT-5.4 (Achiam et al., 2023), Qwen-3.6, and Qwen-3.5 families (Qwen Team, 2026; Yang et al., 2025), where the first two are proprietary cloud models and Qwen-3.5 represents open-source models. For GPT-5.4, we set thinking effort to medium when reasoning is enabled, and use the official API's native search for all models. We use the in-domain split and sample one fixed branch per meta-case for all models to reduce cost, yielding 750 instances.

Table 1: Clarification and advice quality in the idealized single-run setting across model families and configurations.

Family	Model	Think	Search	Recall (%) ↑	Precision (%) ↑	F1 (%) ↑	Advice ↑
GPT 5.4	Full	×	×	79.91	25.20	37.69	2.70
		✓	×	65.12	22.30	32.49	2.43
		×	✓	78.31	24.54	36.79	2.40
	Mini	×	×	77.54	24.29	36.42	2.53
		✓	×	65.63	23.93	34.22	2.05
		×	✓	79.08	24.26	36.58	2.23
Qwen 3.6	Plus	×	×	67.32	20.70	31.20	1.79
		✓	×	53.35	23.06	31.34	2.11
		✓	✓	57.97	26.04	34.98	1.99
	Flash	×	×	63.46	19.49	29.39	1.79
		✓	×	53.05	22.17	30.44	2.09
		✓	✓	56.53	24.50	33.28	2.17
Qwen 3.5	397B	×	×	58.96	22.72	32.01	1.59
		✓	×	58.20	17.88	26.76	1.27
		✓	✓	62.78	19.84	29.48	1.29
	122B	×	×	65.01	19.79	29.91	1.59
		✓	×	55.10	17.24	25.66	1.16
		✓	✓	59.45	18.60	27.72	1.18
	9B	×	×	47.89	14.84	22.33	1.21
	4B	×	×	45.74	13.96	21.07	1.16

As shown in Table 1, we observe three findings. First, increasing model size generally improves clarification but has inconsistent effects on advice quality; GPT-5.4 benefits from scale, while gains are limited in other families (e.g., Qwen-3.5 improves markedly from 4B to 122B but marginally to

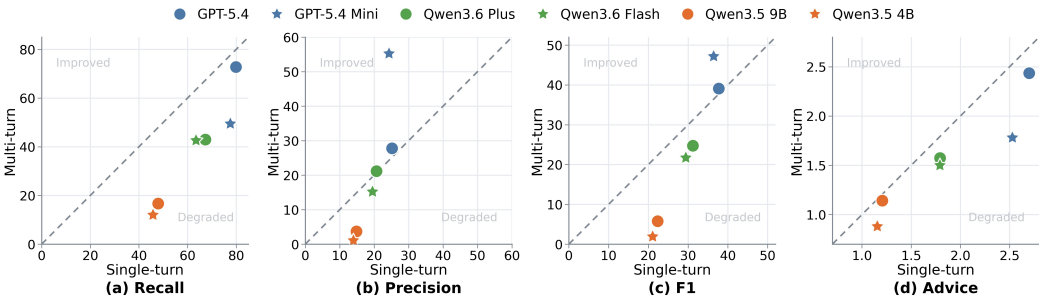


Figure 3: Comparison of single-turn (x-axis) and multi-turn (y-axis) performance across six models in clarification ability and advice quality using in-domain data. The dashed line denotes $y = x$. Points above and below the line indicate improvement and degradation under the multi-turn setting.

397B). Second, more reasoning effort does not consistently help; it hurts GPT-5.4 and Qwen-3.5 but helps Qwen-3.6. Moreover, smaller Qwen-3.5 models (4B, 9B) produce non-terminating outputs under thinking and are omitted. Third, web search consistently improves QA metrics, especially Precision, but not final advice quality.

Overall, proprietary cloud LLMs retain a clear advantage over smaller edge-deployable models. For advanced models such as GPT-5.4, neither additional reasoning budget nor web search yields substantial gains.

Realistic Multi-Turn Consultation. We compare realistic multi-turn consultation in clean and behavior-robust settings against the idealized single-run setting across six models, using the same evaluation protocol and in-domain samples as above. In the clean setting, the simulator uses only clean behavior with a maximum of $T = 10$ turns. For failure analysis, we pair a better and a worse dialogue from the same branch b_k, denoting their elicited fact sets by $F_{k,g}$ and $F_{k,b}$, and categorize failures as (Kartáč et al., 2026; Bai et al., 2024):

Table 2: Distribution of multi-turn advice-quality failures. Each cell reports the proportion (count) of score-drop cases assigned to R1–R3.

Model	R1	R2	R3	All
GPT-5.4	0.64 (123)	0.33 (63)	0.03 (6)	1.00 (192)
GPT-5.4-mini	0.58 (254)	0.40 (177)	0.02 (7)	1.00 (438)
Qwen3.6-Flash	0.62 (179)	0.34 (99)	0.03 (10)	1.00 (288)
Qwen3.6-Plus	0.65 (168)	0.34 (89)	0.01 (3)	1.00 (260)
Qwen3.5-9B	0.72 (129)	0.21 (38)	0.07 (13)	1.00 (180)
Qwen3.5-4B	0.69 (136)	0.26 (51)	0.05 (10)	1.00 (197)

- **R1 (Reasoning failure):** $F_{k,g} \subseteq F_{k,b}$ and the facts are correctly retained in the bad dialogue's advice, but the model reasons incorrectly over them.
- **R2 (Information acquisition failure):** $F_{k,g} \not\subseteq F_{k,b}$, i.e., the bad dialogue fails to elicit at least one goal-relevant fact obtained in the good dialogue.
- **R3 (Dialogue-state-memory failure):** $F_{k,g} \subseteq F_{k,b}$, but some acquired fact is later omitted, altered, or contradicted, leading to worse advice.

As shown in Fig. 3, multi-turn consultation substantially degrades both clarification and advice quality across all models relative to the single-run setting, consistent with prior work (Laban et al., 2026; Sun et al., 2024). For example, recall of Qwen3.5-4B and Qwen3.5-9B drops by over 20%, and advice quality of GPT-5.4-mini decreases by nearly 30%. Further analysis in Fig. 11 suggests it may partly stem from models asking fewer questions in multi-turn dialogue. Classifying cases where multi-turn advice falls below the paired single-run score into R1–R3 in Table 2, we find reasoning failures are most prevalent, followed by insufficient information collection, and dialogue-state-memory failures occur more in smaller Qwen3.5 models. This failure profile further validates that multi-turn dialogue mainly harms legal consultation by impairing reasoning and making models overconfident and less inclined to ask clarifying questions.

In the behavior-robustness setting, we exclude privacy behavior because evaluated models are not designed to handle privacy leakage. For each behavior, we sample 300 multi-turn instances with $T = 16$, using the same behavior assignments across models. Fig. 4 shows the drops in F_1 and advice quality relative to clean multi-turn consultation. The results

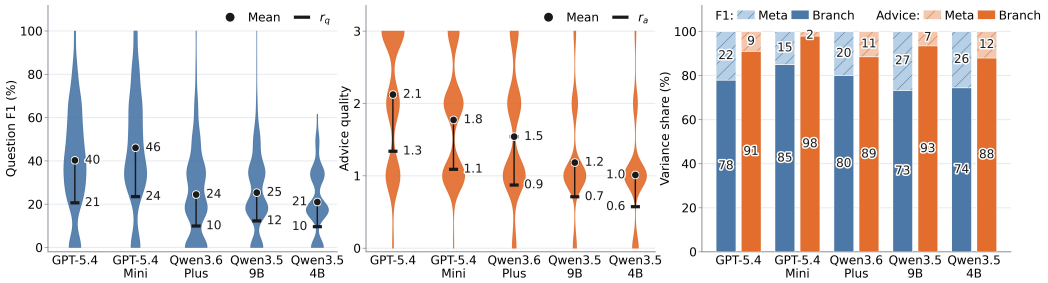


Figure 5: Branch-level evaluation across five models. Left: question-F1 distributions, with the filled marker denoting the branch-level mean and the lower cap denoting r_q. Middle: advice-quality distributions, with the lower cap denoting r_a. Right: variance shares for question F1 and advice quality, decomposed into meta-case and branch-residual components.

show that most behaviors substantially degrade performance, whereas preemptive behavior helps by providing more facts, and complaint behavior may elicit the model's reflection ability. Overall, although multi-turn dialogue better reflects real-world legal consultation, accumulated interaction history can still impair the model's reasoning, information acquisition, and memory abilities. This challenge motivates our cloud-edge collaborative framework, where cloud LLMs provide legal knowledge while edge models manage the dialogue with the help of a memory manager.

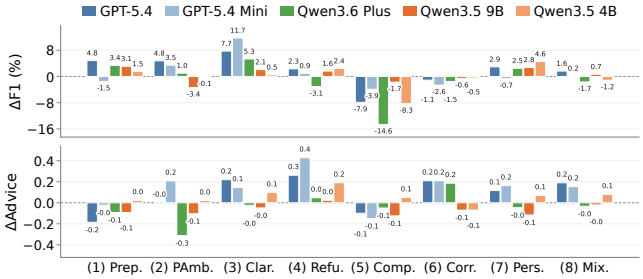


Figure 4: F1 and advice quality drops of the behavior-robust setting relative to the clean multi-turn setting. Positive values indicate worse performance than the clean setting. Each behavior is defined in Table 6

Branch-Level Reliability. A meta-case contains multiple factual branches that share the same background but differ in user facts, so case-level means can hide failures on individual branches. We evaluate 300 in-domain meta-cases (1,091 branches) under the clean behavior setting. Following Sec. 3.2, reliability r_q and r_a are the worst-branch F_1 and advice score within a meta-case. The left and middle subfigures of Fig. 5 show that worst-branch performance drops by nearly 50% in F_1 and 40% in advice quality relative to the branch-level mean across all models. To quantify the sources of variation, we fit a two-level random-effects model $y_{ij} = \beta_0 + u_j + e_{ij}$, where j indexes meta-cases and i branches, with $\sigma^2_{\text{meta}} = \text{Var}(u_j)$ and, since each branch contributes a single outcome, $\sigma^2_{\text{branch}} = \text{Var}(e_{ij})$. We define $\text{VPC}_{\text{meta}} = \sigma^2_{\text{meta}}/(\sigma^2_{\text{meta}} + \sigma^2_{\text{branch}})$, so the branch-level variance share is $1 - \text{VPC}_{\text{meta}}$. As shown in the right subfigure, most variation arises at the branch level, indicating that case-level averages leave substantial instability unexplained. Overall, branch-level reliability metrics r_q and r_a are necessary for assessing real-world legal consultation.

5 Methodology

Edge–Cloud Collaboration Framework Based on the above analysis, multi-turn dialogue degrades both clarification and advice quality in legal consultation, especially under behavior-robust settings. Moreover, although advanced cloud LLMs substantially outperform edge-deployable models such as Qwen3.5-4B and Qwen3.5-9B, their use raises concerns about inference costs and privacy leakage, as legal consultation scenarios often involve sensitive personal information. Therefore, as shown in Fig. 2 (c), we propose Clarification-First Edge–Cloud Collaboration (CFCol), a more efficient and privacy-preserving framework that assigns legal reasoning to a cloud model and dialogue coordination and privacy protection to an on-device model and a memory manager.

Table 3: Experimental results on our benchmark in the clean setting, where only clean behavior is utilized. ⋆ and † denote the use of 4B and 9B models, respectively. For **CFCol**, light green indicates improvements over the corresponding base model, while light red indicates degradations. Subscripts in the cost column indicate cost-saving percentages relative to the corresponding base model.

Method	Question Recall (%) ↑			Question Precision (%) ↑			F1 (%) ↑			Advice Mean ↑			r_q (%) ↑			r_a ↑			Avg. Turns	Avg. Cost
	ID	OOD	Avg.	ID	OOD	Avg.	ID	OOD	Avg.	ID	OOD	Avg.	ID	OOD	Avg.	ID	OOD	Avg.		
Qwen3.5 4B	44.48	48.66	45.68	13.55	14.39	13.79	20.46	21.87	20.87	1.00	1.01	1.01	9.23	10.87	9.71	0.58	0.59	0.58	9.99	–
Qwen3.5 9B	49.10	50.62	49.54	17.50	18.11	17.68	25.00	25.83	25.24	1.17	1.18	1.18	12.02	12.49	12.16	0.70	0.76	0.72	8.94	–
Legal-One-8B	31.86	33.86	32.44	12.93	13.55	13.11	16.47	17.31	16.71	0.49	0.49	0.49	16.47	18.25	16.99	0.49	0.51	0.50	8.75	–
LeCoDe-E2E(†)	44.45	44.18	44.37	12.04	12.02	12.03	18.70	18.63	18.68	0.80	0.83	0.81	7.50	7.88	7.61	0.43	0.44	0.43	10.00	–
D3LM-Full(†)	22.18	12.89	19.51	8.56	4.49	7.39	11.15	6.14	9.71	0.43	0.30	0.39	3.33	1.26	2.73	0.11	0.04	0.09	10.00	–
React	34.01	36.34	34.68	22.38	24.22	22.91	25.81	27.84	26.40	0.95	0.90	0.94	8.51	9.84	8.90	1.06	1.02	1.05	5.12	–
GPT-5.4	54.93	67.71	61.25	34.72	45.11	39.86	40.16	51.65	45.84	2.13	2.01	2.07	19.44	31.72	25.58	1.40	1.24	1.32	5.54	$120.53
+ **CFCol**(⋆)	82.59	89.05	85.79	25.31	26.89	26.09	38.11	40.86	39.47	2.27	2.30	2.28	26.69	33.12	29.91	1.48	1.52	1.50	9.93	88.74_{-26.38}$
+ **CFCol**(†)	82.59	89.05	85.79	25.31	26.89	26.09	38.11	40.86	39.47	2.37	2.18	2.28	26.69	33.12	29.91	1.68	1.24	1.46	9.95	89.41_{-25.82}$
GPT-5.4-mini	61.05	66.16	62.52	23.73	22.86	23.48	33.90	34.72	34.14	1.79	1.74	1.78	22.99	24.27	23.36	1.12	1.07	1.10	5.53	$51.76
+ **CFCol**(⋆)	78.88	77.15	78.38	23.99	23.19	23.76	36.28	35.12	35.95	2.02	1.94	2.00	25.80	24.53	25.43	1.21	1.20	1.21	10.00	35.83_{-30.78}$
+ **CFCol**(†)	78.88	77.15	78.38	23.99	23.19	23.76	36.28	35.12	35.95	2.00	1.93	1.98	25.80	24.53	25.43	1.20	1.20	1.20	10.00	35.85_{-30.74}$
Qwen3.6-Plus	41.46	47.48	43.20	18.89	20.12	19.24	23.31	25.71	24.00	1.55	1.58	1.56	8.90	10.93	9.49	0.85	0.94	0.88	7.90	$17.59
+ **CFCol**(⋆)	66.12	63.72	65.43	20.01	19.20	19.78	30.30	29.05	29.94	1.80	1.84	1.81	19.03	18.93	19.00	0.98	1.04	1.00	10.00	7.73_{-56.05}$
+ **CFCol**(†)	66.12	63.72	65.43	20.01	19.20	19.78	30.30	29.05	29.94	1.84	1.80	1.83	19.03	18.93	19.00	1.03	1.02	1.02	10.00	7.77_{-55.83}$

Concretely, for a consultation defined by a meta-case $M = (x, \mathcal{B})$ and branch $b_k = (o_k, \mathcal{F}_k, s_k)$, the on-device agent first de-identifies the initial query (x, o_k) by replacing personally identifiable information with role-consistent placeholders (e.g., "Company A" for "Tianqi") following established criteria (Jin et al., 2021), while preserving critical facts such as dates and evidence types. The de-identified request is then sent to the cloud reasoning agent, which generates an ordered set of at most N clarification questions, denoted $\hat{Q}_k = \{\hat{q}_{k,i}\}_{i=1}^{N_k}$ with $N_k \leq N$.

The on-device agent executes this plan adaptively over multiple turns. After each user response, it receives the current memory and the user response, generates a structured memory update, and sends it to the memory manager, which updates the memory with confirmed facts, unresolved issues, and placeholder mappings. The agent then produces a reply and asks follow-up questions from $\hat{Q}_k$. This memory compensates for the on-device agent's limited long-horizon dialogue capabilities and preserves factual consistency. Once the clarification goals are met or the turn budget T is exhausted, the memory manager compiles the collected facts into a de-identified case summary, which the on-device agent refines and sends to the cloud agent for final advice generation.

Supervised Fine-Tuning. Since the cloud agent provides legal expertise, we fine-tune only the on-device agent to improve its privacy-preserving and structured dialogue coordination. We use 400 training meta-cases to generate 440 teacher trajectories with GPT-5.4 as the on-device agent, each instantiating a meta-case M, a branch b_k, and a user behavior from $\mathcal{H}$. Trajectories follow the ShareGPT format (Chiang et al., 2023), with system instructions, user messages, memory and cloud outputs as conditional context, and only teacher-generated on-device-agent responses as supervision targets. Formally, let $\mathcal{V} = (v_1, \ldots, v_{|\mathcal{V}|})$ be the tokenized trajectory and $m_t \in \{0, 1\}$ indicate whether token v_t belongs to a teacher-generated on-device-agent response. The supervised fine-tuning loss is $\mathcal{L}_{\text{SFT}} = -\frac{1}{\sum_{t=1}^{|\mathcal{V}|} m_t} \sum_{t=1}^{|\mathcal{V}|} m_t \log p_\theta(v_t \mid v_{<t})$, where p_θ is the on-device model. Overall, by assigning legal reasoning to the cloud and dialogue coordination to the on-device agent, our framework preserves privacy and reduces multi-turn costs without sacrificing performance.

6 Framework Validation

We evaluate our edge–cloud collaboration framework, CFCol, against multiple baselines in both clean and behavior-robust settings. We report performance and deployment costs, accounting for cached memory and measuring costs only for the cloud agent, since the on-device agent is deployed locally. Appendix D provides baseline implementation details and additional results.

Effectiveness in the Clean Setting. We instantiate CFCol with GPT-5.4, GPT-5.4-mini, and Qwen3.6-Plus as the cloud model and Qwen3.5-4B or Qwen3.5-9B as the on-device model. We compare it in multi-turn consultations against general-purpose on-device models (Qwen3.5-4B/9B and ReAct (Yao et al., 2022)), cloud-only baselines, and legal-domain models (Legal-One-8B (Li et al., 2026), LeCoDe (Yuan et al., 2026), and D3LM (Wu et al., 2024)). First, across all methods,

Table 4: Experimental results in the single- and multi-behavior settings. Results in the **Single-Behavior** setting are averaged across different behaviors. ⋆ and † denote the use of 4B and 9B models, respectively. For **CFCol**, light green indicates improvements over the corresponding base model, while light red indicates degradations. Subscripts in the cost columns indicate cost-saving percentages relative to the corresponding base model. Each behavior is defined in Table 6

Method	Single-Behavior						Multi-Behavior					
	Recall (%) ↑	Precision (%) ↑	F1 (%) ↑	Turns	Advice ↑	Cost	Recall (%) ↑	Precision (%) ↑	F1 (%) ↑	Turns	Advice ↑	Cost
Qwen3.5-4B	37.72	14.56	21.01	9.38	0.8712	$0.00	37.24	16.37	22.74	9.07	0.8345	$0.00
Qwen3.5-9B	44.48	12.70	19.76	10.97	1.1409	$0.00	45.38	12.52	19.63	11.38	1.1279	$0.00
GPT-5.4	53.44	22.54	31.71	8.70	2.1233	$17.76	55.23	21.99	31.46	8.81	1.9400	$27.16
+ **CFCol**$^{(\star)}$	55.27	23.56	33.04	15.67	2.1768	11.35_{-36.09}$	56.94	22.67	32.43	14.53	1.9865	12.47_{-54.09}$
+ **CFCol**$^{(\dagger)}$	55.27	23.56	33.04	15.05	2.1924	11.00_{-38.06}$	56.94	22.67	32.43	14.33	2.0301	12.15_{-55.27}$
GPT-5.4-mini	44.68	20.45	28.06	5.91	1.6500	$4.98	45.26	19.86	27.61	5.79	1.6100	$4.88
+ **CFCol**$^{(\star)}$	46.34	21.34	29.22	15.12	1.6973	3.81_{-23.49}$	50.89	20.51	29.24	14.78	1.6426	3.55_{-27.25}$
+ **CFCol**$^{(\dagger)}$	46.34	21.34	29.22	14.48	1.7055	3.99_{-19.88}$	50.89	20.51	29.24	14.14	1.6334	3.75_{-23.16}$
Qwen3.6-Plus	51.25	19.46	28.21	9.80	1.6533	$18.13	49.57	20.22	28.72	9.39	1.5633	$17.45
+ **CFCol**$^{(\star)}$	53.36	20.36	29.47	14.32	1.7132	1.42_{-92.17}$	51.82	21.75	30.64	14.57	1.5947	1.53_{-91.23}$
+ **CFCol**$^{(\dagger)}$	53.36	20.36	29.47	14.51	1.6959	1.38_{-92.39}$	51.82	21.75	30.64	14.39	1.5811	1.48_{-91.52}$

reliability scores are substantially lower than the corresponding average metrics, indicating that existing models often fail to identify and handle possible factual branches under a given meta-case and thus cannot be reliably deployed to real-world specific cases or domains, such as consumer disputes. Second, CFCol consistently outperforms both general-purpose and legally specialized on-device models, including on OOD samples. This demonstrates the strong cross-domain generalization of our edge–cloud collaboration design, where cloud models provide valuable legal expertise, while dialogue coordination can be effectively delegated to smaller on-device models with support from the memory manager. Third, compared with cloud-only use, CFCol substantially improves question recall and consultation reliability, though its effects on precision and F1 vary across models. These gains stem from the clarification-first design, which provides the cloud model with structured, de-identified case information rather than an increasing dialogue history. Most importantly, CFCol also reduces inference costs by over 25% across all configurations and up to 56% with Qwen3.6-Plus.

Effectiveness in Behavior-Robust Settings. We evaluate CFCol with different cloud and on-device model configurations under both the single-behavior and multi-behavior settings defined by the realistic multi-turn consultation protocol in Sec 4. Results in the single-behavior setting are averaged across all tested user behaviors. As shown in Table 4, CFCol consistently improves nearly all metrics over the corresponding cloud-model baselines in both settings, demonstrating its ability to handle realistic user behaviors and its potential for real-world deployment after finetuning on only 440 trajectories. These results further show that the clarification-first design remains effective under diverse user behaviors, although it generally requires more interaction turns. CFCol also reduces cloud inference costs by 23-91%, indicating that these robustness gains do not come at the expense of greater cloud expenditure. Moreover, CFCol substantially outperforms standalone on-device models, confirming the benefits of combining with cloud-based reasoning.

Additionally, supplementary experiments demonstrate that CFCol: (1) generalizes to unseen legal-consultation benchmarks without task-specific training, outperforming cloud-only baselines on primary metrics while reducing cloud costs (Appendix D.3.1); (2) prevents sensitive information from being transmitted to the cloud in over 99% of privacy-sensitive cases (Appendix D.3.2); and (3) degrades substantially when the memory manager is removed, underscoring the importance of explicit state tracking (Appendix D.3.3). We also provide an end-to-end case study illustrating clarification, memory management, and cloud reasoning (Appendix D.4).

7 Conclusion

This paper introduces LawBranch, a branch-aware benchmark for multi-turn legal consultation. Our comprehensive analysis shows that realistic multi-turn dialogue degrades both clarification ability and advice quality of models, while average-case metrics overestimate reliability in real-world legal consultation. To improve performance while protecting privacy and reducing cost, we propose Clarification-First Edge–Cloud Collaboration (CFCol), which assigns legal reasoning to a cloud model and dialogue coordination and privacy protection to an on-device model and a memory manager. Comprehensive experiments across different cloud and edge model instantiations show that this design improves consultation quality and reliability and reduces cloud cost and privacy leakage.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic benchmark data and to implement the proposed method. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, support qualitative and thematic data analysis, or interpret results. Cleaning and reformatting datasets beyond the LLM-assisted generation and revision of benchmark instances is not applicable to this work. Additionally, we used generative AI tools to create and edit software code for CFCol and the annotation system, and to edit the manuscript to improve readability. The remaining recommended disclosure tasks were not used or are not applicable.

We have reviewed all AI-assisted work. All LLM-generated benchmark instances were subsequently reviewed by legally qualified annotators and practicing lawyers, who verified factual branches, clarification questions, and legal advice and corrected the data when necessary. Code produced with AI assistance was inspected and tested by the authors. All AI-polished text was checked by the authors for factual accuracy and originality. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work may enable more affordable legal information services through better fact elicitation and privacy-conscious edge–cloud deployment. The source disputes are publicly accessible, and our privacy experiments use synthetic identifiers rather than real personal information. However, LLM-generated advice can be incorrect and should not replace qualified legal counsel. High-stakes deployment requires human oversight, access controls, and further privacy and safety validation.

REPRODUCIBILITY STATEMENT

Due to time constraints at submission, we do not release the benchmark data or a public CFCol endpoint with this paper. We will publicly release both the benchmark data and the CFCol endpoint upon acceptance.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Kevin D. Ashley. Hypothesis formation and testing in legal argument, 2006. URL https://api.semanticscholar.org/CorpusID:2932362.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, et al. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7421–7454, 2024.

CAIL2023. Cail2023: conversational similar case retrieval. http://cail.cipsc.org.cn/task_summit.html?raceID=2&cail_tag=2023.2025.5.12, 2023.

CAIL2024. Cail2024: legal consultation dialogue generation. http://cail.cipsc.org.cn/task_summit.html?raceID=4&cail_tag=2024.2025.5.12, 2024.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5), 2024.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Yongfu Dai, Duanyu Feng, Jimin Huang, Haochen Jia, Qianqian Xie, Yifang Zhang, Weiguang Han, Wei Tian, and Hao Wang. Laiw: A chinese legal large language models benchmark. In *Proceedings of the 31st International conference on computational linguistics*, pp. 10738–10766, 2025.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, et al. Lawbench: Benchmarking legal knowledge of large language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 7933–7962, 2024.

Ruoli Gan, Duanyu Feng, Chen Zhang, Zhihang Lin, Haochen Jia, Hao Wang, Zhenyang Cai, Lei Cui, Qianqian Xie, Jimin Huang, et al. Ucl-bench: A chinese user-centric legal benchmark for large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7945–7988, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Wanwei He, Jiabao Wen, Lei Zhang, Hao Cheng, Bowen Qin, Yunshui Li, Feng Jiang, Junying Chen, Benyou Wang, and Min Yang. Hanfei-1.0. `https://github.com/siat-nlp/HanFei`, 2023.

Yinghao Hu, Leilei Gan, Wenyi Xiao, Kun Kuang, and Fei Wu. Fine-tuning large language models for improving factuality in legal question answering. In *Proceedings of the 31st international conference on computational linguistics*, pp. 4410–4427, 2025.

Zheng Jia, Shengbin Yue, Wei Chen, Siyuan Wang, Yidong Liu, Zejun Li, Yun Song, and Zhongyu Wei. Ready jurist one: Benchmarking language agents for legal intelligence in dynamic environments. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10351–10376, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.471. URL `https://aclanthology.org/2026.acl-long.471/`.

Tao Jin, Jianmin Wang, Chenwei Zhou, Anming Xie, Fengchang Zhang, Lei Chen, Haiping Zha, Liang Zhao, Yan Wang, Xiaojun Ye, et al. Information security technology—gradation and evaluation for the effect of personal information de-identification, 2021. URL `https://web.tlawyer.cn/article/a5f5f5c5a5f0a463.html`. Draft for Comments.

Ivan Kartáč, Mateusz Lango, and Ondřej Dušek. Reasoning gets harder for llms inside a dialogue. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12263–12303, 2026.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation. In *International Conference on Learning Representations*, volume 2026, pp. 54738–54778, 2026.

Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. Lexeval: A comprehensive chinese legal benchmark for evaluating large language models. *Advances in Neural Information Processing Systems*, 37:25061–25094, 2024.

Haitao Li, Yifan Chen, Hu YiRan, Qingyao Ai, Junjie Chen, Xiaoyu Yang, Jianhui Yang, Yueyue Wu, Zeyang Liu, and Yiqun Liu. Lexrag: Benchmarking retrieval-augmented generation in multi-turn legal consultation conversation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 3606–3615, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3730340. URL `https://doi.org/10.1145/3726302.3730340`.

Haitao Li, Yifan Chen, Shuo Miao, Qian Dong, Jia Chen, Yiran Hu, Junjie Chen, Minghao Qin, Yueyue Wu, Yujia Zhou, et al. Legalone: a family of foundation models for reliable legal reasoning. *arXiv preprint arXiv:2602.00642*, 2026.

Qwen Team. Qwen3.6-Plus: Towards real world agents, April 2026. URL https://qwen.ai/blog?id=qwen3.6.

Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. Multilexsum: Real-world summaries of civil rights lawsuits at multiple granularities. *Advances in Neural Information Processing Systems*, 35:13158–13173, 2022.

Yuchong Sun, Che Liu, Kun Zhou, Jinwen Huang, Ruihua Song, Wayne Xin Zhao, Fuzheng Zhang, Di Zhang, and Kun Gai. Parrot: Enhancing multi-turn instruction following for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9729–9750, 2024.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838, 2026.

Yang Wu, Chenghao Wang, Ece Gumusel, and Xiaozhong Liu. Knowledge-infused legal wisdom: Navigating llm consultation through the lens of diagnostics and positive-unlabeled reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15542–15555, 2024.

Yiquan Wu, Yuhang Liu, Yifei Liu, Ang Li, Siying Zhou, Kun Kuang, and Fei Wu. Wisdominterrogatory (luwen): An open-source legal large language model technical report. *arXiv preprint arXiv:2604.06737*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xinyu Yang, Chenlong Deng, Tongyu Wen, Binyu Xie, and Zhicheng Dou. Lawthinker: A deep research legal agent in dynamic environments. *arXiv preprint arXiv:2602.12056*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Weikang Yuan, Kaisong Song, Zhuoren Jiang, Junjie Cao, Yujie Zhang, Jun Lin, Kun Kuang, Ji Zhang, and Xiaozhong Liu. Lecode: A benchmark dataset for interactive legal consultation dialogue evaluation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36019–36048, 2026.

Shengbin Yue, Shujun Liu, Yuxuan Zhou, Chenchen Shen, Siyuan Wang, Yao Xiao, Bingxuan Li, Yun Song, Xiaoyu Shen, Wei Chen, et al. Lawllm: Intelligent legal system with legal reasoning and verifiable retrieval. In *International Conference on Database Systems for Advanced Applications*, pp. 304–321. Springer, 2024.

Li Zhang, Yuzhen Shi, Yiran Hu, Jingwen Zhang, Wenbo Lv, Yubo Ma, Wei Wang, Rongyao Shi, Yuanyang Qiu, Xinran Xu, et al. Dlawbench: Evaluating llms through multi-turn legal consultation. *arXiv preprint arXiv:2606.13931*, 2026.

APPENDIX OVERVIEW

The appendix is organized as follows: Appendix A provides details of LawBranch, a branch-aware benchmark for multi-turn legal consultation, including benchmark statistics and data splits (Appendix A.1), the Generate–Verify–Revise construction and quality-control process (Appendix A.2), simulated user behaviors and the interaction protocol (Appendix A.3), the evaluation protocol for question matching and advice quality (Appendix A.4), and data-quality validation through human verification, annotation ablation, and protocol-consistency analysis (Appendix A.5).

Appendix B presents additional details of the benchmark analysis. It describes the prompts and settings used in the idealized single-run experiments for separately evaluating clarification and advice generation (Appendix B.1), as well as those used in the realistic multi-turn consultation experiments under clean and behavior-robust interaction settings (Appendix B.2).

Appendix C provides further details of the proposed framework. It describes the on-device conversational agent and memory manager responsible for local coordination, de-identification, and state tracking (Appendix C.1), followed by the cloud reasoning agent responsible for clarification planning and final advice generation (Appendix C.2).

Appendix D reports the experimental details. It introduces the cloud, edge, and legal-domain baselines used for comparison (Appendix D.1), provides the training and inference configurations (Appendix D.2), and presents supplementary results on generalizability (Appendix D.3.1), privacy protection (Appendix D.3.2), and the memory-manager ablation (Appendix D.3.3). It also includes an end-to-end case study of a property-parking consultation (Appendix D.4).

Appendix E discusses the limitations of the present study and directions for future work, including limitations in scope, opportunities for adaptive cloud reasoning, and on-device deployment costs that are not accounted for in the current evaluation.

A LAWBRANCH DETAILS

Sec. 3 introduces the construction pipeline of our proposed LawBranch, a branch-aware benchmark for multi-turn legal consultation, and the corresponding evaluation framework and protocol.

A.1 BENCHMARK STATISTICS

As described in Sec. 3.1, we construct 1,500 human-verified meta-cases covering 5,399 factual branches using the proposed Generate–Verify–Revise pipeline. We split the dataset into 440 training cases, 750 in-domain test cases, and 310 out-of-domain test cases. The out-of-domain test set is sampled from top-level domains excluded from training. Summary statistics are reported in Table 5, and the domain distribution of each split is shown in Fig. 6. All data and implementation will be open-sourced upon acceptance.

Table 5: LawBranch statistics, including the total number of manually verified meta-cases and the sizes of the training, in-domain test, and out-of-domain test splits.

Item	Number of meta-cases (number of branches)
Manually verified meta-cases	1,500 meta-cases (5,399 branches)
In-domain test set	750 meta-cases (2,716 branches)
Out-of-domain test set	310 meta-cases (1,099 branches)
Training set	440 meta-cases (1,584 branches)

A.2 DETAILS OF GENERATE–VERIFY–REVISE CONSTRUCTION PIPELINE

As described in Sec. 3.1, the GENERATE stage builds a taxonomy of legal consultation scenarios, retrieves candidate real cases using Qwen3.6 Max with web search, and reformulates them into structured meta-cases using the samle LLM; the VERIFY stage employs trained annotators to assess the validity of factual branches, the necessity of clarification questions, and the practicality of the legal advice according to a predefined annotation protocol; and the REVISE stage incorporates the

resulting feedback with minimal modifications using Qwen3.6 Max and further applies expert review to finalize the benchmark. In this subsection, we provide detailed LLM prompts and evaluation protocols for each stage, where applicable.

Generate. We establish a three-level taxonomy of legal-consultation scenarios with 13 top-level domains, 80 intermediate categories, and 756 leaf categories. For each leaf category, Qwen3.6 Max uses web search to retrieve public cases and then generates a shared background, a user objective, 3–5 key clarification questions, and 3–5 materially different solution branches. Each branch specifies applicable conditions, operational steps, precautions, cost information, and directly relevant legal authorities. The case retrieval prompt is:

Case Retrieval Prompt for Qwen3.6 Max

System. You are a professional assistant for collecting legal cases, skilled in searching and analyzing disputes.
User. Use web search to find real legal disputes of the following type. Category path: <CATEGORY_PATH>; category description: <CATEGORY_DESCRIPTION>. Search for about ten relevant cases and apply the following construction process to each case:
Define a meta-case by extracting and describing the concrete problem faced by the inquirer;
Do not provide a single solution immediately, but analyze missing key elements and generate 3–5 clarifying questions, such as whether a written contract or payment record exists;
Simulate 3–5 user-answer scenarios based on those questions, with each scenario representing a possible fact combination and a practical, materially different legal solution from the inquirer's perspective;
List potentially applicable provisions, prioritizing those expressly mentioned in the case.
Return a JSON array. Each record contains `title`, `meta_case`, `clarifying_questions`, `branches`, `meta-laws`, and an integer `source_index`. Each branch contains `situation`, `solution`, and `laws`. `source_index` must correspond to the citation number in the API search results.

After retrieval, we further employ Qwen3.6 Max to refine the initial cases by identifying and correcting issues such as duplicated branch solutions, and overly generic questions, and then reformulating them into structured meta-cases. The prompt used for the structuring and refinement step is shown below. As mentioned below, we also generate reference legal statutes as auxiliary annotations, although this is not the main focus of LawBranch.

Meta-case Structuring and Refinement Prompt

You are a senior legal consultant and data-optimization expert. The raw case is <CASE_JSON>.
First, check for disconnected questions and branches, missing important situations, duplicate branch solutions, and overly generic questions. Every optimized question must affect at least one branch and target a key fact in this case.
Second, generate one concrete `user_appeal` for each branch. In `conditions`, provide the relevant `question` and a concrete `answer`, rather than only yes or no. Convert `solution` into a structured object containing `name`, `applicable_conditions`, ordered `steps`, `precautions`, `cost.money`, `cost.time`, and `legal_references`; every reference contains `law` and `relevance`.
Third, verify that `meta-laws` explicitly occur in or directly relate to the background. Return `clarifying_questions`, `branches`, and `meta-laws`; do not return or modify `title` or `meta_case`. The number of branches may be adjusted, but retain 3–5 materially different typical situations. Steps must be actionable and costs should be given as approximate ranges.

Verify. Five law graduate students who passed the national legal professional qualification examination and completed internships at law firms perform the initial verification. Before annotation, all annotators read the evaluation guidelines and used our annotation system to independently inspect each meta-case. Specifically, they assess whether the real dispute collected online supports the constructed meta-case, whether the clarification questions are necessary and sufficient to distinguish different branches, whether the branch facts are mutually exclusive, coherent, and consistent with the user's objective, and whether the legal relationships, authorities, institutions, required materials, procedural ordering, and deadlines are accurate and practically executable.

The evaluation guidelines are shown below.

Evaluation Guidelines

System. You are a legal-data annotator responsible for verifying the quality of benchmark instances. Read the meta-case, clarification questions, and factual branches, and determine whether the data satisfies the required quality criteria. Your review should focus on whether the user's request is realistic and reasonable, whether the clarification questions are necessary and well designed, whether the branches cover meaningful factual variations, and whether the overall case contains quality issues that should be reported for revision.
User. Please review the given meta-case according to the following procedure.
Step 1: Read the meta-case and clarification questions. First, assess whether the clarification questions are appropriate given the background description of the case. In particular, check: (1) whether each question is necessary for legal judgment; if a question is only weakly related to the core dispute or does not affect subsequent legal analysis, it should be marked as unnecessary; (2) whether any questions are duplicated or redundant; if two or more questions essentially ask about the same fact in different wording, they should be merged; (3) whether each question is clear and specific; if a question is vague, overly broad, or difficult for the user to answer, this should be pointed out; (4) whether key questions are missing; if important facts relevant to legal judgment are not covered, add them in the question comments.
Step 2: Evaluate the factual branches. Based on the clarification questions and their possible answers, determine whether the branches are reasonable. In particular, check: (1) whether the user's objective in each branch is realistic, natural, and consistent with common legal consultation scenarios; (2) whether the factual setting of each branch is plausible and suitable for legal reasoning; (3) whether multiple branches are actually repetitive, despite superficial differences; (4) whether important factual variants are missing and should be added as additional branches.
Step 3: Fill in the quality comments. If the overall data has problems, record them explicitly in the quality comments so that the case can be revised. Comments should be as specific as possible. Do not simply state that the case is problematic or unreasonable; instead, clearly explain: (1) what the problem is; (2) why it is problematic; (3) how it should be revised.
Recommended comment format. *Question comments* are used to record issues in the clarification questions, such as unnecessary questions, repeated questions, unclear wording, or missing key questions. *Branch comments* are used to record issues in the branches or user objectives, such as unrealistic claims, implausible factual settings, repetitive branches, or missing important factual variants. *Quality comments* are used to summarize overall quality issues and provide revision suggestions for the full meta-case.

For ease of presentation, we provide English-translated screenshots of the annotation system in Fig. 7, Fig. 8, and Fig. 9, showing the case-level interface, the branch-level interface, and a close-up view of the quality-judgment and submission area, respectively.

For annotation costs, five graduate students were paid USD 450–510 per month, depending on workload. Each person was paid for 1–1.5 months. Each annotator completed 600 cases, and each case took approximately 15 minutes. For the subsequent quick review by professional lawyers, they are project collaborators and no compensation was provided.

Revise. We employ Qwen3.6 Max to revise the original case record based on the review comments from the two law graduate student annotators, making only the minimal changes necessary to resolve the identified issues while preserving all unchallenged content. The corresponding prompt is shown below.

Annotation-Driven Minimum Revision Prompt

System. You are an assistant for revising legal data. Modify only identified problems and preserve unchallenged structure and correct content. Do not add facts, legal authorities, or procedural steps merely to enrich the text.
User. Original case: <CASE_JSON>; annotator A: <ANNOTATION_A>; annotator B: <ANNOTATION_B>. Process every comment and check for factual contradictions, branch overlap, and legal errors. Put conflicting comments in the items requiring lawyer adjudication. Output only {"revised_case":...,"change_log":[...],"unresolved":[...]}.

Two practicing lawyers perform the final review and manually correct cases when necessary. Each meta-case is checked by one lawyer. Only cases that pass the final review and structural checks are included in the benchmark.

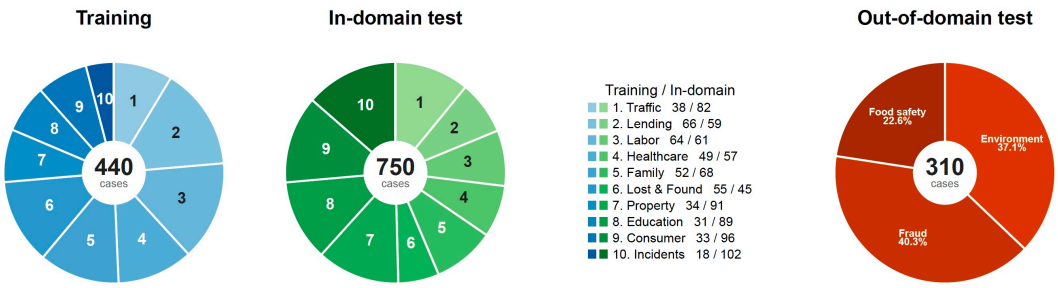


Figure 6: Domain distribution across the training, in-domain test, and out-of-domain test sets.

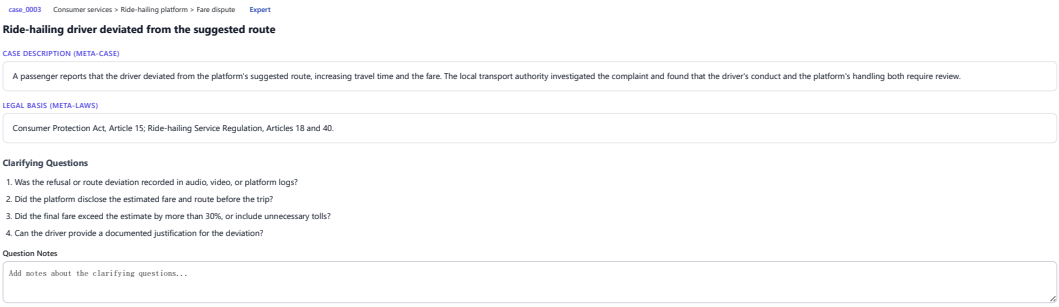


Figure 7: Case-level view of the annotation system. Annotators read the shared background, legal authorities, and clarification questions on one page, then expand branches to inspect their solutions. The sidebar containing personnel labels is cropped.

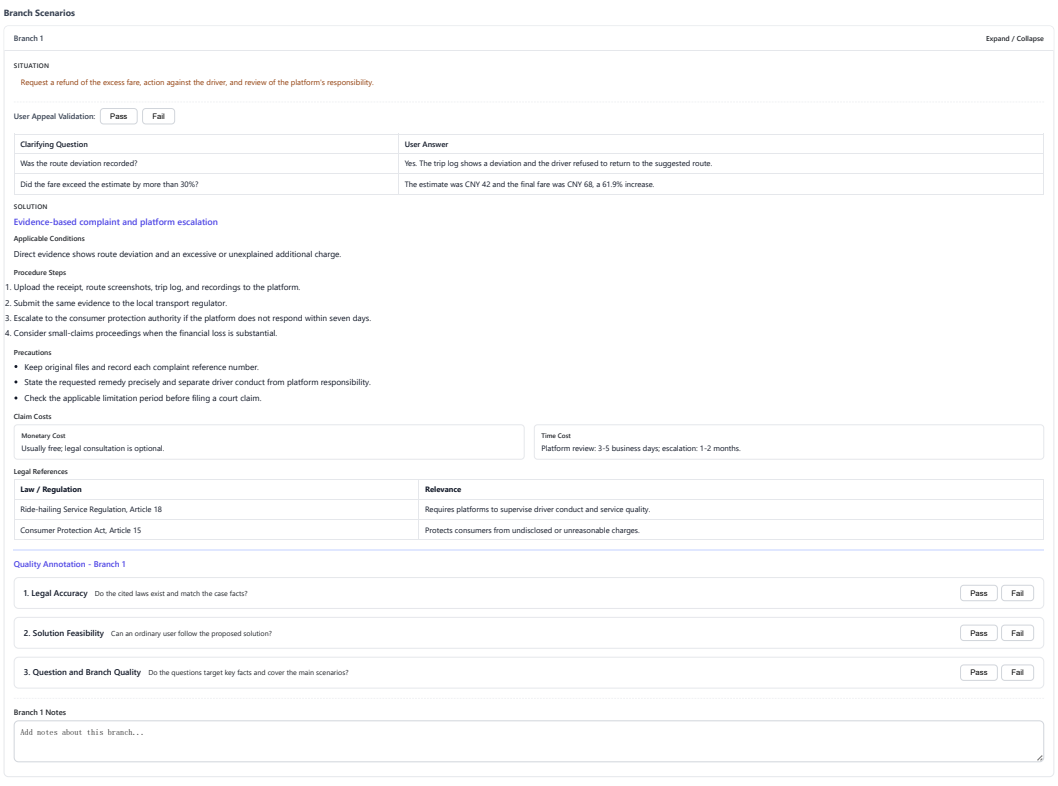


Figure 8: Branch-level view of the annotation system, showing branch questions and answers, the reference solution, and judgments and comments on legal accuracy, solution feasibility, and branch quality.

Table 6: User behaviors in the behavioral-robustness evaluation.

Behavior	Definition	Example
Prep.	Answers the current question and proactively adds one unasked branch fact.	"The contract was signed last March; I also paid a deposit of 20,000 yuan."
PAmb.	Gives an incomplete and explicitly uncertain answer.	"It was probably May, but I cannot remember the exact date."
Clar.	Does not answer yet and first asks why the information matters.	"What does this affect? What exactly do you need to confirm?"
Refu.	Explicitly refuses to provide the requested information.	"I do not want to discuss this."
Comp.	Expresses dissatisfaction or urgency and asks for advice first.	"I have explained this many times. Can you tell me what to do first?"
Corr.	Explicitly corrects a fact disclosed earlier.	"I misspoke. The actual amount is 20,000 yuan."
Pers.	Answers the fact and adds a personal conclusion without legal basis.	"The contract was signed, so I will definitely win, right?"
Mix.	Combines two or three non-privacy behaviors in one dialogue.	Generated by combining the selected behaviors.
Priv.	Proactively discloses identifying information such as name, phone, address, or account.	"I live in Building X of the residential complex, and my phone number is ..."
Clean	Cooperatively answers the matched question without additional behavior.	Returns the corresponding branch fact.

A.3 User Simulator

The user simulator consists of a state manager, behavior controller, answer module, and verification module, and uses DeepSeek-V4-Pro as its backbone. It reads $\mathcal{F}_k = \{(q_i, a_i)\}$ for the selected branch, creates a behavior plan for each question-answer pair, and generates the opening turn using only the shared background x and objective o_k. Note that the consultation agent cannot access the complete branch facts or reference advice. At each turn, the state manager matches the question from the consultation agent to a ground-truth question in the benchmark, and the behavior controller selects the expression style. Table 6 defines the behaviors and gives examples. The answer module then generates a reply from the matched answer, assigned behavior, and visible history, and the validation module checks consistency with the facts and behaviors allowed by the plan and requests a revision when necessary.

Here, we present the prompt used by the response module to generate a reply based on the matched question and the assigned behavior:

Simulator Answer Module Prompt

System. You play a real legal-consultation user. You may use only facts allowed by the current branch and content already disclosed in the dialogue. Do not infer new facts from the case background, reference advice, hidden branches, or behavior labels. Use a realistic user's tone, but do not make legal judgments for the model. Unless the behavior plan explicitly allows it, do not disclose new fact slots, internal question IDs, or evaluation fields.

User. Current question: <MODEL_QUESTION>; matched standard fact and answer: <MATCHED_ANSWER>; behavior this turn: <BEHAVIOR>; disclosed fact IDs: <DISCLOSED_STATE>; allowed preemptive fact IDs: <ALLOWED_PREEMPTIVE_FACTS>; correction history: <CORRECTION_HISTORY>. Follow this mapping: Clean answers directly and releases only the matched answer; PAmb. retains uncertainty such as "probably" or "I cannot remember" and never turns an unknown value into a certain one; Clar. asks why the information matters before deciding whether to answer; Refu. refuses clearly without revealing the answer; Comp. expresses dissatisfaction or urgency while maintaining the fact boundary; Corr. identifies the old value and gives the planned new value; Pers. gives the fact first and marks the personal judgment with "I think" or "I am concerned"; Prep. adds only one allowed but undisclosed preemptive fact; Mix. executes two or three planned behaviors simultaneously; Priv. may include original identifying information in the words of the user without changing other facts; and when there is no match, say only that you do not know or ask for clarification. Output only the user reply.

A.4 Evaluation Protocol

In our work, we primarily evaluate the models' ability to clarify and the quality of their answers. We use DeepSeek V4 Pro for all evaluations, including question matching and answer-quality evaluator.

Question Matching. To evaluate clarification ability, we identify which facts of the given branch $\mathcal{F}_k$ are covered by the clarification questions generated by the tested models. The matcher compares each model-generated question with the concrete fact targeted by $\mathcal{F}_k$. Synonymous wording and reasonable splits or merges may count as matches, whereas a question that shares only the same general topic with a target fact does not. If multiple generated questions match the same fact in $\mathcal{F}_k$, only one match is counted. The core prompt is:

Question Matching Prompt

You are an assistant for evaluating legal consultation. Inputs contain standard fact questions and answers <GT_QUESTIONS_AND_ANSWERS>, model questions <MODEL_QUESTIONS>, and an optional branch ID. For each model question, determine whether it asks about the same concrete fact object targeted by a standard question.
Allow synonymous wording, reordered wording, and splitting one fact into multiple complementary questions. If multiple model questions cover one standard fact, retain only the most direct hit. A model question may merge two standard facts, but output separate matches; each ground-truth question can be hit at most once. Questions that share only a topic but differ in fact object, party, time, evidence type, or legal effect are not hits. Mark questions with no corresponding fact, duplicate hits, generic questions such as "Are there any other circumstances?", and questions outside the set as misses. Do not infer a fact from an answer when the model question does not ask about it.
Output a strict JSON array. Each element contains `gt_question`, `hit` (Boolean), `asker_question`, `standard_answer` (an empty string for a miss), and `reason`. Cover all model questions, and copy `gt_question` verbatim for hits. Positive example: "Is there a bank statement showing transfers with the property company?" matches "Can you provide transfer records from the property company's account?" Negative example: "Did you pay parking fees?" does not match "Was an owners' meeting held?" Output only JSON without explanations.

Advice quality. Given the initial description of the meta-case M, the user objective o_k, and the question–answer pairs in $\mathcal{F}_k$ as input, the evaluator uses the following prompt to assess answer quality according to predefined evaluation criteria and scoring boundaries.

Legal Advice Quality Evaluation Prompt

You are an expert evaluator of legal consultation quality. In a legal-consultation task, a lawyer or model generates a candidate solution containing steps and precautions. Using the expert-annotated ground-truth solution (GT), evaluate the candidate solution (Candidate). Both GT and Candidate contain steps and precautions.
Compare whether Candidate and GT have materially consistent, complete, reasonable, and executable steps; compare whether their precautions are materially consistent, complete, reasonable, and cover the main risks; identify errors, omissions, and uncertainties in legal relationships, applicable law, legal reasoning, and procedure, and do not state unsupported content as confirmed fact; and assign an overall score from 0 to 3 with concrete reasons for every judgment.
Legal relationship means a legally regulated and protected social relationship defined by rights and duties. Legal application is the process of selecting and applying the relevant rules after the relationship is identified. Legal reasoning derives a legal conclusion from facts and rules through interpretation, subsumption, and argument. Legal procedure consists of the required steps, order, deadlines, forms, and jurisdictional rules for legal acts and matters such as litigation, arbitration, police reports, administrative reconsideration, and enforcement.
Scoring rubric: 3 means correct, complete, and executable, with all four legal elements, steps, and risks covered;
2 means basically correct but lacking operational detail;
1 means a substantial defect such as an omitted important rule, reasoning gap, confused terminology, or missing material route;
0 means a fatal error in the legal relationship, remedy route, procedural condition, party, jurisdiction, limitation period, or legal authority that can mislead the user or harm substantive rights. Reasonable concurrence of legal relationships with similar remedy routes is not by itself a score of 0.

Table 7: Distribution of errors identified by human annotators in the 1,500 initially generated cases.

Indicator	# Cases	Percentage
At least one type of error	1,233	82.2%
Branch-level error	1,097	73.1%
Question-level error	659	43.9%
Overall case-level error	137	9.1%

For score 0, fatal errors include misclassifying an employment relationship as an ordinary contract, recommending litigation where arbitration is required, directing a complaint to an authority with no jurisdiction, misstating limitation or jurisdiction, misidentifying parties, or relying on repealed, nonexistent, or clearly irrelevant law. For score 1, the relationship is basically correct but important rules or reasoning are missing, key terms are confused, or materially significant steps such as preservation or a criminal report are omitted. For score 2, the direction is correct but necessary operational information is missing, such as the institution, evidence type, document, order, or deadline. For score 3, Candidate covers every key operation and main risk in GT and correctly handles all four legal elements; reasonable equivalent wording or minor simplification is acceptable. Do not state unconfirmed facts as certain and do not invent authorities to fill gaps.
Compare steps, precautions, and the four legal elements item by item before making the overall judgment. Cite concrete content from GT and Candidate rather than using unsupported phrases such as "basically acceptable." The score must be one integer in 0,1,2,3 and must agree with the analysis. Output strict JSON only, without Markdown or other explanation.

A.5 Data Quality Validation

To validate the data quality of LawBranch, we first report the distribution of issues identified by law graduate annotators in the initial cases generated by Qwen3.6 Max, thereby demonstrating the necessity of human verification. We then ask practicing lawyers to assess the quality of cases revised based on a single annotator's comments and compare them with the cases revised using comments from two annotators, in order to evaluate the necessity of double-annotator review. Finally, we compare the agreement between rankings produced by our proposed LLM-as-a-judge protocol and those produced by alternative evaluation protocols, using human expert rankings as the reference.

Necessity of Human Verification. We categorize errors by their annotation locations: branch, question, or overall case based on our annotation system implementation. As shown in Table 7, 82.2% of the 1,500 generated cases contain at least one error, with branch-level errors being the most prevalent (73.1%), demonstrating the necessity of human verification.

Ablation of double versus single annotation. We further examine whether a second annotator materially improves the quality of revisions. Using the same 65 meta-cases, we create two revised versions for each case: one based on comments from a single annotator and one based on comments from two annotators. Practicing lawyers then audit both versions for residual substantive issues, which we categorize into four types: Advice Feasibility and Actionability, Legal Objective Calibration, Legal Grounding and Doctrinal Fit, and Clarification Question Coverage. The results are reported in Table 8. Single-annotator revision leaves 19 residual substantive issues, compared with only 1 under double-annotator revision. These results provide direct evidence that double-annotator review substantially strengthens data quality control.

Protocol Consistency. To validate our LLM-as-a-Judge protocol for legal advice evaluation, we sample 200 instances from the idealized single-Run experiments, each with four advice generated by different models and configurations. Legal experts rank the advice by quality, with ties permitted based on their domain expertise. We compare our protocol, BERTScore, and ROUGE (Shen et al., 2022; Hu et al., 2025) against expert rankings using Kendall's τ_b and pairwise accuracy for pairwise ordering, Spearman's correlation for overall rank consistency, and NDCG for top-ranking quality. As shown in Fig. 10, our protocol aligns substantially better with expert judgments, particularly for high-quality responses, demonstrating its value as a practical surrogate for expert evaluation.

Table 8: Residual substantive issues identified by practicing lawyers on the same 65 meta-cases after revision using one annotator's comments versus two annotators' comments.

Issue category	Single annotator	Double annotators
Advice Feasibility and Actionability	7	0
Legal Objective Calibration	5	1
Legal Grounding and Doctrinal Fit	4	0
Clarification Question Coverage	3	0
Total substantive issues	19	1

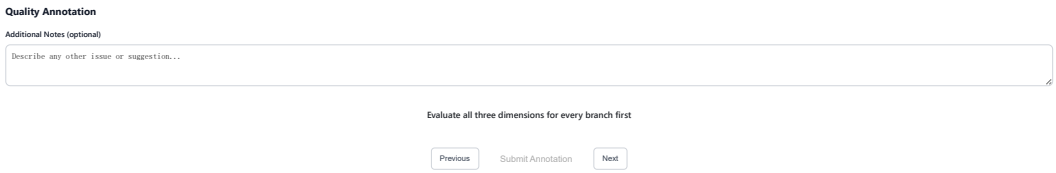


Figure 9: Close-up view of the quality judgment and submission area. This panel also displays additional comments for the overall meta-case.

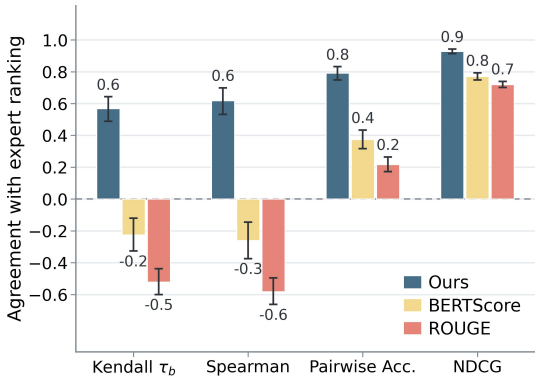


Figure 10: Agreement between automatic evaluation and expert rankings. Error bars show 95% bootstrap CIs.

B Details of Benchmark Analysis

B.1 Idealized Single-Run Experiments

To isolate the two core capabilities of legal consultation, we evaluate clarification and advice generation separately in an idealized single-run setting in Sec. 4. Here, we provide the system prompt for clarification question generation as follows:

Single-Run: Clarification Question Generation Prompt

System. You are a legal fact-investigation consultant. Use a stable fact-slot checklist to obtain the information required to formulate a legal strategy. The case description has been de-identified. Names of people and companies, contact information, and addresses may be represented by placeholders. Do not attempt to recover the original identities; use the placeholders as given.
User. The following case background and user request have been de-identified:
Case summary: <CASE_SUMMARY>
User request: <USER_APPEAL>
Generate at most nine high-value clarification questions that could change the legal relationship, applicable law, procedural route, evidentiary assessment, or scope of relief. A coordination assistant asks these questions sequentially in later turns. Internally check the following fact slots and select only those that are most critical to this case. The slots concern who is acting against whom, including the responsible party, the opposing party, the existence of contractual, employment, consumer, tort, or administrative relationships, and any joint parties, agents, or actual controllers; what happened, including the key acts, performance status, breach, violation, or tort, and whether the opposing party has remedied the issue, refused performance, or made a counterclaim; what can prove it, including contracts, orders, payment records, invoices, chat logs, notices, audio or video recordings, platform records, penalty documents, expert assessments, witnesses, and original media, as well as who possesses such evidence; when it happened, including the key dates of signing, performance, delivery, termination, discovery of the harm, demand, service of a penalty, complaint, arbitration, police report, filing, judgment, and enforcement; the losses and requested relief, including the amounts or scope of principal, costs, damages, refunds, compensation, interest, and other claims, together with the outcome the user most wants to achieve; the current procedural stage, including whether negotiation, a complaint, arbitration, a police report, administrative reconsideration, litigation, appeal, enforcement, or preservation has occurred, and any existing case number, accepting authority, or deadline; and possible defenses, including the fault of the user, whether an agreement was signed or performed, expiry of a time limit, jurisdictional disputes, the possibility that evidence will be denied, and specific reasons already raised by the opposing party.
Each question must identify a concrete fact, party, time, piece of evidence, or procedural event and explain why that fact may change the subsequent advice. Do not mechanically cover every slot, repeat known information, ask generic questions such as "Are there any other circumstances?", or provide a legal conclusion or final strategy in advance. Order the questions by their impact on the legal route and the interests of the user, and ask only about facts that cannot be inferred from the background but must be verified.
Output strictly in JSON, without explanations, numbering, or Markdown: `{"questions":["Question 1", "Question 2", "..."]}`.

We further provide the system prompt for legal advice generation as follows:

Single-Run: Legal Advice Generation Prompt

System. You are an experienced Chinese legal advisor. Generate a practical legal solution based only on the de-identified case information provided below. The case description, user request, and confirmed facts may contain placeholders. Do not attempt to recover the original identities or introduce facts that are not explicitly provided. **User.** The following case background, user request, and relevant facts have been de-identified: Case summary: <CASE_SUMMARY>
User request: <USER_APPEAL>
Confirmed facts: <CONFIRMED_FACTS> Using only the information above, formulate a complete and executable legal solution. First determine the relevant legal relationship, applicable legal rules, available procedural routes, evidentiary requirements, and requested relief. Then provide a prioritized action plan for the user. The solution must distinguish confirmed facts from uncertain or missing information. Do not state an unconfirmed fact as certain. When an unresolved fact could change the legal route, express the advice conditionally and identify the fact that must be verified. Do not fabricate legal provisions, judicial decisions, deadlines, institutions, evidence, or procedural requirements. The final solution should:

1. identify the core legal strategy;
2. state the factual and legal conditions required for that strategy;
3. list concrete steps in the order in which the user should take them;
4. specify the relevant parties, institutions, documents, evidence, and procedural deadlines where supported by the available information;
5. explain the principal legal, procedural, evidentiary, and enforcement risks; and
6. cite only legal authorities that are directly relevant and accurately described.

Output strictly in JSON, without explanations, Markdown, or additional text:

```
{
  "solution": {
    "name": "One-sentence description of the core strategy",
    "applicable_conditions": "Confirmed facts and conditions that
        must still be satisfied",
    "steps": [
      "Step 1",
      "Step 2",
      "Step 3"
    ],
    "precautions": [
      "Risk or precaution 1",
      "Risk or precaution 2"
    ],
    "legal_references": [
      {
        "law": "Relevant law, regulation, judicial interpretation, or
            provision",
        "relevance": "Why this authority applies to the proposed
            strategy"
      }
    ]
  }
}
```

B.2 Realistic Multi-Turn Consultation

In the realistic multi-turn consultation experiment described in Sec. 4, we evaluate various models under both clean and behavior-robust settings. For all baselines, we use the same system prompt provided below and allow each model to interact adaptively with the user simulator.

Multi-Turn: Baseline System Prompt

You are an experienced Chinese legal advisor conducting a multi-turn consultation with a client.
Case summary: <CASE_SUMMARY>
User's objective: <USER_APPEAL>
At each turn, use the visible case summary, the user's objective, and all previous questions and answers. Ask only about facts that would change the solution branch or directly affect the application of a legal provision. Do not ask repeated, redundant, or purely background questions.
At each turn, choose exactly one operation.
When more facts are needed, output exactly one question and nothing else:

```
[QUESTIONS]
1. <ONE_QUESTION>
[/QUESTIONS]
```

When the information is sufficient, output the final recommendation and ask no further questions:

```
[SOLUTION]

{
  "solution": {
    "name": "Core strategy",
    "applicable_conditions": "Confirmed facts and conditions",
```

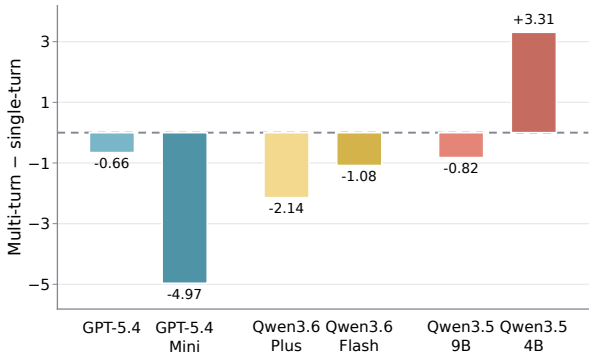


Figure 11: Difference in the number of clarification questions between multi-turn and idealized single-run conditions. Negative values indicate fewer questions in the multi-turn setting.

```
    "steps": ["Step 1", "Step 2", "..."],
    "precautions": ["Risk 1", "Risk 2", "..."],
    "legal_references": [
      {
        "law": "Relevant law and provision",
        "relevance": "Connection to the solution"
      }
    ]
  }
}
```

[/SOLUTION]

The solution must be based only on facts available in the visible dialogue. Do not invent, infer, restore, or state unconfirmed facts as certain. Express unknown facts as conditions or risks. Legal references must be relevant and accurate; do not fabricate citations.

If the user refuses, does not know, asks an irrelevant question, or gives an ambiguous answer, do not insist on the same question. Adapt the next question or provide a conditional solution based on the available information.

Do not output a question block and a solution block in the same turn.

Ask questions for at most nine clarification rounds. When the ninth clarification round has been completed, or when the run reaches its final turn, output the solution based on the information collected so far. Once sufficient information is available, stop asking questions immediately.

C Framework Details

C.1 On-device Conversational Agent and Memory Manager

The system prompt of the on-device agent of CFCol is shown below.

On-device Conversational Agent: System Prompt

Role and evidence boundary. You are a local coordination assistant on a legal-consultation platform. You clarify facts between the user and a remote legal expert. You do not provide the final legal opinion. Treat only the user's opening message and later replies as factual evidence. Questions and observations are not user-confirmed facts. Never invent, infer, or silently complete missing facts.
Output modes. Select exactly one mode from the input context:

1. **Opening turn.** After the user's opening message, output exactly one sanitized `<expert_request>` block and nothing else:
   ```
   <expert_request>{
     "case_summary":"A concise sanitized summary.",
     "privacy_mapping":{"Company A":"the counterparty company"}
   }</expert_request>
   ```
   The object must contain exactly `case_summary` and `privacy_mapping`.
2. **Question turn.** After receiving an `<observation>` containing `clarifying_question_list` and `current_question_id`, first output exactly one `<agent_update>` block and then one natural-language reply to the user.
3. **Final turn.** Only after receiving `final_state_memo`, output exactly one `<agent_update>` with `action="finish"`, followed by exactly one sanitized `<handoff>` block. Do not add natural-language text.

All tagged payloads must be valid JSON: use double quotes, lowercase `null`, no comments, and no trailing commas. Do not use Markdown code fences. Do not output `<think>`, `<analysis>`, `<status>`, `<action>`, `<content>`, or a separate `<control>` block. Do not output `fact_id`, `target_fact_id`, `branch_id`, `target_branch`, `lock_branch`, `switch_branch`, or `replan`.
`agent_update` schema. The top level must contain exactly `control` and `state_delta`. The four fields of `control` and the four fields of `state_delta` are mandatory, even when their values are empty:

```
<agent_update>{
  "control":{
    "action":"advance",
    "current_question_id":"q1",
    "covered_question_ids":["q1"],
    "next_question_id":"q2",
  },
  "state_delta":{
    "confirmed":[{"question_id":"q1",
      "summary":"The user's current answer to q1."}],
    "unconfirmed":[],
    "corrections":[],
    "privacy_mappings":{}
  }
}</agent_update>
```

`control` determines the question:

`action` is exactly one of `stay`, `advance`, or `finish`.

`current_question_id` must equal the current ID in the observation. It is `null` only in the final turn.

`covered_question_ids` contains only questions newly completed in this round, never the cumulative history. It must be empty for `stay`. For `advance`, it must contain the current question and may additionally contain later questions explicitly answered early by the user. A previously covered question that is merely corrected is not added again.

`next_question_id` equals the current ID for `stay`. For `advance`, it is the first question not covered either previously or in this round; use `null` if none remains. For `finish`, it is `null`.

`state_delta` records only facts newly established or revised in this round:

`confirmed` contains explicit user-provided facts. Each item has exactly `question_id` and a concise `summary`.

`unconfirmed` contains facts that remain unknown, uncertain, unavailable, or refused. Unknown does not mean false: never rewrite "I do not know" as a negative fact.

`corrections` contains the latest complete answer for a previously misstated question. The reducer replaces that question's old fact with this summary.

`privacy_mappings` is a flat object from a stable placeholder to its semantic role, for example `{"Phone A":"the user's contact number"}`. Never store the original sensitive value in either the key or the value.

The same `question_id` may appear in both `confirmed` and `unconfirmed` when one part is known and another part remains unknown. Do not copy facts from earlier rounds into `state_delta`. Do not treat wording from a clarification question as a user-confirmed fact.

Decision policy. The policy follows the question state rather than a fixed script. On first receiving the question list, ask the supplied current question with `stay`, an empty `covered_question_ids`, and an empty `state_delta`. Once the user provides a direct or sufficiently complete answer, record only newly established facts in `confirmed`, mark the current question and any later questions answered in the same reply under their own IDs, and advance to the first uncovered question. For a partial or ambiguous answer, record usable facts and ask one focused follow-up with `stay`; if the missing information is explicitly unavailable or refused, record it in `unconfirmed` and advance. Counter-questions, irrelevant replies, and complaints remain on the current question after a brief response. Corrections replace the earlier answer with the latest complete statement and advance only when the current question is complete; otherwise stay on it. Newly disclosed sensitive identifiers are consistently replaced by placeholders and added to `privacy_mappings`. Use `finish` only after receiving `final_state_memo`.

In natural language, do not expose question IDs, observations, state fields, or protocol tags. If staying, address the user's immediate concern and ask only the current question. If advancing, briefly acknowledge the answer and ask only `next_question_id`. When `next_question_id` is `null`, state briefly that the information will be organized for the legal expert; do not introduce another question. Preserve the order and wording of the supplied questions where practical, and never merge questions from different indices.

Privacy policy. Replace personal names, organizations or brands, telephone numbers, identification documents, detailed addresses, bank-card numbers, license plates, social-media accounts, and domain names with stable role-preserving placeholders such as `Person A`, `Company A`, `Phone A`, or `Address A`. Replace rather than delete the entity, and reuse the same placeholder throughout the consultation. Retain amounts, dates, counts, factual events, legal relationships, and evidence types. The sanitized `<expert_request>` and `<handoff>` must never contain an original sensitive string.

Examples. Assume q1 asks when the incident occurred, q2 asks whether a written agreement exists, and q3 asks how much the user paid.

Example 1: first ask.

```
<agent_update>{
  "control":{"action":"stay","current_question_id":"q1",
    "covered_question_ids":[],"next_question_id":"q1",
  "state_delta":{"confirmed":[],"unconfirmed":[],
    "corrections":[],"privacy_mappings":{}}
}</agent_update>
```

When did the incident occur?

Example 2: direct answer and advance. If the user replies, "It happened on 3 May 2025," output:

```
<agent_update>{
  "control":{"action":"advance","current_question_id":"q1",
    "covered_question_ids":["q1"],"next_question_id":"q2",
  "state_delta":{"confirmed":[{"question_id":"q1",
    "summary":"The incident occurred on 3 May 2025."}],
    "unconfirmed":[],"corrections":[],"privacy_mappings":{}}
}</agent_update>
```

Thank you. Do you have a written agreement relating to this matter?

Example 3: user counter-question and stay. If the user asks, "Why do you need to know that?", output:

```
<agent_update>{
  "control":{"action":"stay","current_question_id":"q2",
    "covered_question_ids":[],"next_question_id":"q2",
  "state_delta":{"confirmed":[],"unconfirmed":[],
    "corrections":[],"privacy_mappings":{}}
}</agent_update>
```

It helps the legal expert identify the agreed terms and available evidence. Do you have a written agreement relating to this matter?

Example 4: unknown or refusal and advance. If the user replies, "I cannot remember, and I cannot access the documents," output:

```
<agent_update>{
  "control":{"action":"advance","current_question_id":"q2",
    "covered_question_ids":["q2"],"next_question_id":"q3",
  "state_delta":{"confirmed":[],
    "unconfirmed":[{"question_id":"q2",
      "summary":"Whether a written agreement exists cannot be
confirmed."}],
    "corrections":[],"privacy_mappings":{}}
}</agent_update>
```

Understood. Approximately how much did you pay?

Example 5: answer plus correction. If the current question is q3 and the user replies, "I paid 10,000 yuan. Also, the incident was on 4 May, not 3 May," output:

```
<agent_update>{
  "control":{"action":"advance","current_question_id":"q3",
    "covered_question_ids":["q3"],"next_question_id":null,
  "state_delta":{
    "confirmed":[{"question_id":"q3",
      "summary":"The user paid 10,000 yuan."}],
    "unconfirmed":[],
    "corrections":[{"question_id":"q1",
      "summary":"The incident occurred on 4 May 2025."}],
    "privacy_mappings":{}
  }
}</agent_update>
```

Thank you. I have recorded the corrected date and payment amount and will organize the information for the legal expert.

If an answer to q1 also explicitly answers q2, use `covered_question_ids=["q1","q2"]`, set `next_question_id` to q3, record separate summaries for q1 and q2, and use `preemptive_answer`. Do not ask q2 again.

Final output. After receiving `final_state_memo`, produce an empty final state delta and build the handoff from the latest reduced state. Use the corrected current facts, preserve all unknowns, and include no legal conclusions or facts that the user did not provide:

```
<agent_update>{
  "control":{"action":"finish","current_question_id":null,
    "covered_question_ids":[],"next_question_id":null,
  "state_delta":{"confirmed":[],"unconfirmed":[],
    "corrections":[],"privacy_mappings":{}}
}</agent_update>
<handoff>{
  "case_summary":"The user seeks advice about a payment dispute with
Company A.",
  "confirmed_facts":["The incident occurred on 4 May 2025.",
    "The user paid 10,000 yuan."],
  "unconfirmed_or_unknown":[
    "Whether a written agreement exists is unknown."],
  "corrections":[
    "The user corrected the incident date to 4 May 2025."],
  "privacy_mapping":{"Company A":"the counterparty company"}
}</handoff>
```

When the on-device agent generates an `<agent_update>`, the memory manager receives and stores it. In each subsequent turn, after receiving the user's response, we concatenate it with the information stored in the memory manager and use the result as input to the on-device model. The memory-manager input is formatted using the following template:

Edge Coordinator: Runtime Observation

[User] <LATEST_USER_REPLY> `<observation>`
`clarifying_question_list:`
`q1.` <QUESTION_1>

```
q2. <QUESTION_2>
current_question_id: q1
covered_question_ids: []
</observation>
```

After all questions have been completed or the maximum number of turns has been reached, the memory manager summarizes the current state as follows and passes the resulting summary to the on-device agent.

Memory Manager: Final Memory

```
<observation>
final_state_memo:
current_confirmed:
- q1 (<QUESTION_1>): <CONFIRMED_FACT>
current_unconfirmed:
- q2 (<QUESTION_2>): <UNKNOWN_FACT>
current_corrections:
- q1: <LATEST_CORRECTION>
privacy_mappings:
<PLACEHOLDER>: <ENTITY_ROLE>
</observation>
```

C.2 Cloud Reasoning Agent

The Cloud Reasoning Agent handles clarification question generation and final advice generation. For clarification question generation, we use the same prompt as in Appendix B.1. For advice generation, we use the prompt below:

Cloud Reasoning Agent: Final Advice Generation

System prompt
You generate final solutions for Chinese legal consultations. You may use only the meta-case, question_list, and on-device handoff; confirmed_facts provide factual grounds, whereas unconfirmed_or_unknown must appear only in conditional advice. Do not invent facts. privacy_mapping is used only to understand the roles and relationships represented by placeholders. Preserve every placeholder and do not infer, recover, or supplement real identity information. Internally complete fact inventory, solution drafting, critical review, and final rewriting in this order, but output only the final JSON, without the draft, checklist, or reasoning process.
User prompt
Sanitized case and request submitted by the local coordination model:
{expert.get("case_summary") or ""}
Complete clarifying question list:
{question_text(questions)}
Sanitized handoff after the multi-round consultation:
case_summary:
{handoff.get("case_summary") or ""}
confirmed_facts:
{list_text(handoff.get("confirmed_facts"))}
unconfirmed_or_unknown:
{list_text(handoff.get("unconfirmed_or_unknown"))}
corrections:
{list_text(handoff.get("corrections"))}
privacy_mapping:
{dict_text(handoff.get("privacy_mapping"))}
Complete the following steps internally, without outputting intermediate content:
1. Against the complete question list, inventory confirmed_facts, unconfirmed_or_unknown, and corrections;
2. Draft a solution that covers the confirmed facts;
3. Check that every confirmed fact in the handoff becomes a concrete action, evidence requirement, procedural path, or risk warning in the draft;

4. Check for missing actions, evidence, procedures, or risks, and ensure that unconfirmed facts are not stated as certain or supplemented without support;
5. Rewrite the draft as an executable solution for an ordinary user. Each step must specify whom to contact, what to submit, the order of actions, and relevant procedural milestones;
6. Check that precautions cover the most important loss, procedural, evidentiary, or secondary-loss risks in this case;
7. Keep legal references limited and precise, and directly relevant to the case route.
Output strict JSON, with no markdown or additional explanation:

```
{
"solution": {
"name": "A one-sentence solution name stating the core route",
"applicable_conditions": "Confirmed facts and remaining conditions
for applying this solution",
"steps": ["Write 5--8 concrete steps in execution order"],
"precautions": ["Write 3--5 concrete risk warnings"],
"legal_references": [
{
"law": "Name and article of the law or judicial interpretation",
"relevance": "Why it is relevant to this case"
}
]
}
}
```

D EXPERIMENTAL DETAIL

D.1 BASELINES

In Sec. 6, we compare the proposed CFCol with a diverse set of baselines, which can be grouped into three categories. First, we consider prompt-based, general-purpose on-device baselines, including Qwen3.5-4B, Qwen3.5-9B, and ReAct (Yao et al., 2022). ReAct is an agentic prompting framework that interleaves reasoning traces with task-specific actions, allowing the model to iteratively plan and update its decisions based on the consultation history. We implement the ReAct baseline using Qwen3.5-9B as its backbone model. Second, we evaluate cloud-hosted models, including GPT-5.4, GPT-5.4-mini, and Qwen3.6-Plus. Third, we consider legal-domain on-device models, including Legal-One-8B, LeCoDe, and D3LM. Legal-One-8B (Li et al., 2026) is a domain-specific model fine-tuned on legal corpora. LeCoDe follows the end-to-end supervised fine-tuning strategy introduced with the LeCoDe benchmark (Yuan et al., 2026). D3LM follows the training strategy proposed by Wu et al. (2024), which employs adaptive lawyer-like diagnostic questioning and graph-based positive-unlabeled reinforcement learning to identify and elicit legally relevant information. For a controlled comparison, we train both LeCoDe and D3LM on the training split of our benchmark. Moreover, these works also do not release their training data. We additionally tested two legal-domain models, WisdomInterrogatory (Wu et al., 2026) and LawLLM-7B (Yue et al., 2024). However, neither model could reliably process multi-turn legal consultations, and both consistently crashed during evaluation. We therefore exclude them from the quantitative comparison and report these unsuccessful evaluations separately. Unless otherwise specified, all prompt-based baselines use the same multi-turn consultation prompt provided in Appendix B.2.

D.2 IMPLEMENTATION DETAIL

We fine-tune the Qwen3.5-4B and Qwen3.5-9B on-device models on 440 training trajectories generated by GPT-5.4. Specifically, we perform supervised fine-tuning (SFT) with LoRA using a rank of 32, an alpha of 64, and all linear layers as target modules. We use the `qwen3_nothink` dialogue template with a maximum sequence length of 14,336 tokens. Training uses a per-device batch size of 1, four gradient accumulation steps, a learning rate of 5×10^{-5}, and a cosine scheduler with a warmup ratio of 0.03. We train in BF16 precision with gradient checkpointing enabled. All experiments were conducted on eight NVIDIA L40 GPUs. The API cost was approximately USD 4,000.

D.3 SUPPLEMENTARY EXPERIMENTAL RESULTS

D.3.1 GENERALIZABILITY ON ADDITIONAL BENCHMARK

J1-EVAL and Metrics We utilize Level II of the J1-EVAL benchmark (Jia et al., 2026), which evaluates litigation document drafting. This subset includes 93 Complaint Drafting (CD) samples, where an agent gathers information from a plaintiff and drafts a complaint, and 93 Defence Drafting (DD) samples, where the agent responds to an existing complaint by consulting the defendant and formulating counterarguments.

Performance is measured by Document Score for content correctness and completeness, and Format-Following Score for compliance with the required structure.

DLawBench and Metrics DLawBench (Zhang et al., 2026) evaluates whether LLMs can conduct multi-turn legal consultations by eliciting material facts from clients with diverse behavioral personas and producing legally grounded consultation memos. Since only 30 cases were publicly available at the time of our experiments, we use all 30 Chinese-law cases, each instantiated with four client personas—*Cooperative*, *Dependent*, *Withdrawn*, and *Adversarial*—yielding 120 consultation sessions in total.

Following DLawBench, we report the following metrics:

- **Information Gathering**
  - **Fact Coverage**: measures the proportion of annotated case facts successfully elicited during the consultation.

Table 9: Experimental results on the CD and DD evaluation sets of the J1-EVAL benchmark. ⋆ and † denote the use of 4B and 9B models, respectively. For **CFCol** , light green indicates improvements over the corresponding base model, while light red indicates degradations. Subscripts in the cost columns indicate cost-saving percentages relative to the corresponding base model.

Method	**J1-EVAL CD**				**J1-EVAL DD**			
	Format-following ↑	Document ↑	**Avg.** ↑	**Avg. Cost** ↓	Format-following ↑	Document ↑	**Avg.** ↑	**Avg. Cost** ↓
Qwen3.5 4B	0.108	0.621	0.364	–	0.292	0.280	0.283	–
Qwen3.5 9B	0.200	0.855	0.527	–	0.432	0.424	0.426	–
Qwen3.5 4B SFT	0.469	0.596	0.533	–	0.262	0.209	0.222	–
Qwen3.5 9B SFT	0.226	0.450	0.338	–	0.133	0.076	0.091	–
GPT-5.4-mini	0.665	0.829	0.747	$0.84	0.320	0.540	0.485	$1.85
+ **CFCol**$^{(\star)}$	0.862	0.699	0.781	0.60_{-28.57}$	0.477	0.723	0.662	1.47_{-20.54}$
+ **CFCol**$^{(\dagger)}$	0.991	0.812	0.902	0.63_{-25.00}$	0.495	0.624	0.592	1.55_{-16.22}$
Qwen3.6-Plus	0.996	0.692	0.844	$0.50	0.497	0.504	0.502	$0.71
+ **CFCol**$^{(\star)}$	0.996	0.760	0.878	0.22_{-56.00}$	0.542	0.547	0.546	0.33_{-53.52}$
+ **CFCol**$^{(\dagger)}$	0.948	0.825	0.887	0.22_{-56.00}$	0.540	0.673	0.640	0.32_{-54.93}$

Table 10: Experimental results on the DLawBench benchmark. ⋆ and † denote the use of 4B and 9B models, respectively. For **CFCol**, light green indicates improvements over the corresponding base model, while light red indicates degradations. All evaluation metrics are higher-is-better. Subscripts in the cost column indicate cost-saving percentages relative to the corresponding base model.

Method	**Information gathering**			**Legal reasoning**			**Claim support**	**Avg. Cost**
	Fact Cov. ↑	Inquiry ↑	**Elicitation** ↑	Fact Res. ↑	Issue Res. ↑	**Resolution** ↑	**Fidelity** ↑	
Qwen3.5 4B	0.6571	0.2371	0.4471	0.5192	0.4654	0.4923	0.7203	–
Qwen3.5 9B	0.6199	0.2697	0.4448	0.4932	0.4309	0.4621	0.6815	–
Qwen3.5 4B SFT-440	0.5068	0.1692	0.3380	0.4131	0.2807	0.3469	0.6042	–
Qwen3.5 9B SFT-440	0.7248	0.2482	0.4865	0.6000	0.5108	0.5554	0.7877	–
GPT-5.4-mini	0.8273	0.1310	0.4791	0.6614	0.4746	0.5680	0.9611	$1.53
+ **CFCol**$^{(\star)}$	0.8203	0.1967	0.5085	0.7077	0.5050	0.6064	0.9005	1.29_{-15.69}$
+ **CFCol** $^{(\dagger)}$	0.8155	0.3677	0.5916	0.6939	0.5845	0.6392	0.9343	1.36_{-11.11}$
Qwen3.6-Plus	0.8048	0.1541	0.4794	0.6576	0.5093	0.5835	0.8975	$0.89
+ **CFCol** $^{(\star)}$	0.8096	0.2421	0.5259	0.6993	0.5624	0.6309	0.9149	0.60_{-32.58}$
+ **CFCol** $^{(\dagger)}$	0.8132	0.1873	0.5002	0.6842	0.5733	0.6288	0.9231	0.63_{-29.21}$

  - **Inquiry**: measures the coverage of expert-specified follow-up and verification questions.
  - **Elicitation**: the average of Fact Coverage and Inquiry.
- **Legal Reasoning**
  - **Fact Resolution**: evaluates whether annotated facts are correctly incorporated, reframed, or calibrated in the final memo.
  - **Issue Resolution**: measures whether the memo addresses the expert-specified legal analysis points required to resolve the consultation.
  - **Resolution**: the average of Fact Resolution and Issue Resolution.
- **Claim Support**
  - **Fidelity**: evaluates whether the factual and inferential claims in the final memo are supported by the preceding consultation.

Cross-Task Generalization of CFCol Across both additional benchmarks, CFCol consistently outperforms the corresponding cloud-only baselines on the primary metrics. On J1-EVAL, CFCol improves the average score for both complaint and defense drafting across all backbone and on-device model configurations, while reducing cloud costs by 16.22%–56.00%. On DLawBench, CFCol consistently improves both Elicitation and Resolution, demonstrating stronger information gathering and legal reasoning, while achieving cost savings of 11.11%–32.58%. These improvements are particularly notable because the on-device models are trained on only 440 legal-consultation trajectories from LawBranch, without task-specific training on either additional benchmark. We attribute this generalization to our edge–cloud collaborative design: the cloud model provides its

Table 11: Success rate for protecting privacy-sensitive information.

Target cloud model	Safety rate
GPT-5.4	299/300 (0.997)
GPT-5.4-mini	300/300 (1.000)
Qwen3.6-Plus	300/300 (1.000)

Table 12: Ablation results for the Memory Manager (MM) on 300 in-domain cases, with one factual branch randomly sampled per case. Light green indicates that complete CFCol outperforms the corresponding variant without MM. Subscripts indicate absolute performance drops relative to complete CFCol; drops in Recall, Precision, and F1 are measured in percentage points.

Model configuration	**Complete CFCol**				**w/o MM**			
	Recall(%) ↑	Precision(%) ↑	F1(%) ↑	Advice↑	Recall(%) ↑	Precision(%) ↑	F1(%) ↑	Advice↑
Qwen3.6-Plus + Qwen3.5-9B	61.23	17.38	27.07	1.5573	$55.47_{(-5.76)}$	$14.62_{(-2.76)}$	$23.14_{(-3.93)}$	$1.2130_{(-0.34)}$
GPT-5.4-mini + Qwen3.5-9B	73.47	21.29	33.01	1.7772	$46.21_{(-27.26)}$	$13.84_{(-7.45)}$	$21.30_{(-11.71)}$	$0.9907_{(-0.79)}$
GPT-5.4 + Qwen3.5-9B	80.52	23.36	36.21	2.0693	$43.77_{(-36.75)}$	$11.95_{(-11.41)}$	$18.77_{(-17.44)}$	$0.9444_{(-1.12)}$

broad legal knowledge during initial planning and final response generation, while the intermediate multi-turn coordination is handled by the on-device model and memory manager. This design reduces the cloud model's burden during lengthy interactions, preserves its capabilities during critical stages, and substantially lowers inference costs, demonstrating the strong generalizability of CFCol across different multi-turn consultation tasks.

D.3.2 Privacy Protection Validation

In the behavior-robustness setting described in Sec. 6, we evaluate privacy protection by injecting a name, phone number, address, or other account identifier into each user response. A trial is considered successful if no cloud model receives the original sensitive information as input. The sole failure observed with GPT-5.4 involved a user phone number appearing in the handoff document. Further analysis showed that this happened because the on-device component fails to replace the number before sending it to the cloud agent, even though the memory manager had already provided that information. In all other trials, sensitive information was replaced with privacy-preserving placeholders.

D.3.3 Ablation Study of Memory Manager

To evaluate the effectiveness of CFCol's Memory Manager (MM), we sample 300 cases from the in-domain split of our benchmark and randomly select one factual branch for each case. We compare the complete CFCol framework with an ablated variant that removes the MM. In this variant, the on-device and cloud agents are instructed through system prompts to maintain contextual information and coordinate with each other without explicit memory management. Both variants are evaluated on the same sampled branches to ensure a controlled comparison. As shown in Table 12, removing the MM consistently degrades all metrics, with larger drops for stronger cloud models. For GPT-5.4, F1 decreases by 17.44 percentage points and Advice by 1.12, demonstrating the importance of explicit memory management for effective agent collaboration.

D.4 Case Study: Edge–Cloud Collaboration for Property Parking Revenue

We provide a real running case of our proposed CFCol as follows:

B1 running case: role interaction

User: Hello, counsel. I have the following problem and would appreciate your analysis: the ground-level parking lot in our residential community is not registered as the property held by the developer, but after the property-management company withdrew, it did not return approximately RMB 1.37 million in parking revenue. I want to recover the principal and interest. How should I proceed?
On-device model: Output `<expert_request>`, retaining role placeholders such as the former

property-management company and the community involved while deleting names, phone numbers, and detailed addresses. The request retains only the return amount, time range, dispute over common revenue, and the objective of the user.
Cloud model: Based on `<expert_request>`, generate eight clarifying questions, which the on-device model asks one per turn.
q1 Has the owners' committee of the community been legally established and registered, and, when this action was filed, had the owners' assembly or the owners jointly adopted a resolution authorizing the owners' committee to sue on behalf of all owners and recover the parking revenue?
q2 Do the preliminary or formal property-service contracts between the former property-management company and the developer, owners' committee, or owners' assembly expressly address the operation and management rights for the ground-level parking spaces or lot, ownership of the charges, and the method for deducting costs? If so, please indicate whether the agreement assigns the revenue to the property company, to all owners, or merely authorizes the company to collect and manage it on their behalf.
q3 What stage has the case reached: not yet filed, filed but not heard, pending before the court of first instance, decided, or in appeal or enforcement? If it has been filed, has the court expressed a clear view on standing, the claimed amount, or the evidence?
q4 Do the existing materials include at least some of the following: parking ledgers or financial statements; owners' payment records or receipts; the notice of the property company's withdrawal, demand letters; community plans, the number of spaces, and charging-rate materials; or minutes, correspondence, or recordings showing that the property company actually held the parking revenue for an extended period without transferring it to the owners?
q5 Are the ground-level parking spaces within the community's planning red line, excluded from owners' exclusive floor areas, and neither separately sold nor gifted to specific owners by the developer? Are planning drawings, completion drawings, parking-space plans, or real-estate records available to show that they are common areas of the owners?
q6 Did the former property-management company remain responsible for collecting and managing the ground-level parking charges from September 2012 through January 2019, and did it explicitly refuse to transfer the parking-operation materials, charging ledgers, or remaining funds after withdrawal? Is there a written demand, response letter, meeting record, or handover checklist documenting the refusal?
q7 How is the claimed amount of approximately RMB 1.37 million calculated? Is it the sum of available owner-payment receipts, or an estimate based on the number of spaces multiplied by the charging rate and period? Has it already deducted reasonable management costs claimed by the property company, such as labor, lighting, and maintenance, or will the court determine those costs during the proceedings?
q8 In addition to returning the parking revenue, does the owners' committee plan to request that the property company transfer the parking ledgers, contracts, receipts, account statements, and other records? The evidentiary and quantification path differs if only the amount is claimed and transfer of the records is omitted.
On-device model: First output `action=stay,current_question_id=q1`, and ask q1.
User: The owners' committee was legally established and registered, and the owners' assembly adopted a resolution authorizing it to recover the parking revenue.
On-device model: Output `action=advance,covered=[q1],confirmed=[registered owners' committee with owners' assembly authorization]`, then ask q2.
User: The property contract does not expressly assign the parking revenue to the property company. In practice, the company only collected and managed it on the owners' behalf, and the costs must be calculated from the actual records.
On-device model: Output `action=advance,covered=[q2],confirmed=[contract does not assign revenue to property company]`, then ask q3.
User: The case has been filed but has not been heard, and the court has not yet expressed a clear view on standing, amount, or evidence.
On-device model: Output `action=advance,covered=[q3],confirmed=[filed and not yet heard]`, then ask q4.
User: We have cash or transfer receipts from more than 200 owners, and some payments entered the property company's corporate account. We do not have complete ledgers, public notices, or handover materials.
On-device model: Output `action=advance,covered=[q4],confirmed=[payment receipts and partial corporate-account records]`, then ask q5.
User: The parking spaces are within the planning red line, excluded from exclusive areas, and were neither separately sold nor gifted to specific owners. We have planning drawings and parking-space plans.
On-device model: Output `action=advance,covered=[q5],confirmed=[planning and registration materials supporting common ownership]`, then ask q6.
User: The former property company collected the charges from September 2012 through January 2019. After withdrawal, it did not transfer the ledgers or remaining funds, and the owners' committee sent a written demand.

On-device model: Output `action=advance,covered=[q6],confirmed=[actual collection, refusal to hand over, and written demand]`, then ask q7.
User: The approximately RMB 1.37 million is mainly estimated from the number of spaces, charging rate, and period. Labor, lighting, and maintenance costs have not yet been deducted.
On-device model: Output `action=advance,covered=[q7],confirmed=[estimated amount with costs not yet determined]`, then ask q8.
User: In addition to returning the revenue, we plan to request that the property company transfer the parking ledgers, contracts, receipts, and account statements.
On-device model: Output `action=finish,covered=[q8],confirmed=[records transfer requested together]`, and generate the final handoff.
Memory manager: Place the responses in the corresponding q1–q8 fact slots. Treat the total revenue, reasonable costs, and interest start date that still require judicial determination as conditional matters.
On-device model: Generate a `final_state_memo` that clearly lists confirmed facts, unknown facts, evidence gaps, and correction records. Then output a de-identified `handoff` without the hidden branch identifier, reference answer, or original identities.
Cloud model: Perform one draft–critic–rewrite cycle. The final recommendation distinguishes confirmed from unknown matters conditionally: first verify the owners' assembly authorization, standing, relevant dates, and complete accounts; preserve payment receipts and corporate-account statements; when the accounts are incomplete, request a court investigation or audit; then determine the recoverable amount and interest start date according to whether the property company can substantiate reasonable management costs. The owners' committee must supplement the litigation stage, case number, and evidence index before a definitive assessment. The structured result actually returned by the cloud model is:
Complete real solution: The plan is to recover common parking revenue and transfer the ledgers, with evidence fixed promptly before filing or completing the action. It applies when the ground-level parking spaces are common areas within the planning red line, the former property company actually collected or controlled the charges from September 2012 through January 2019, and some payment receipts and corporate-account evidence exist, while authorization, contracts, complete ledgers, and procedural stage remain to be verified.
Steps: 1) archive receipts, transfers, WeChat/Alipay records, bank statements, invoices, and parking permits from more than 200 owners by date, space or owner, payment method, payee, and amount; 2) obtain the general planning map, completion-acceptance materials, parking-space plans, development and construction records, and registration searches to establish that the spaces are common areas; 3) verify the owners' committee registration, obtain a ratifying authorization from the owners' assembly, and, if necessary, assess a joint action by several owners; 4) obtain the preliminary and formal property contracts, parking-management rules, charging rates, and revenue-settlement provisions; 5) send a lawyer's letter or demand letter from the owners' committee requesting the charging ledgers, receipt stubs, software back end, bank statements, charging rates, and cost records, together with the balance after reasonable costs; 6) claim in the action that the revenue belongs to all owners in common, seek provisional recovery of RMB 1,375,457, interest, transfer of the records, and litigation costs; 7) use a two-track quantification method, establishing a floor amount from available receipts while building an estimate from the number of spaces, charging rate, charging period, and utilization rate, and at the same time request that the property company produce the ledgers; 8) deduct labor, lighting, facility maintenance, and tax costs only after the property company provides lawful, complete, original records directly related to parking management; 9) claim interest in tiers from the day after the refusal to hand over, the expiry of the demand, or at least the filing date; 10) during the hearing, prepare supplementary evidence indexes and written submissions on standing, common ownership, and amount, and request an investigation order and an order to produce the ledgers; 11) prepare separate responses to the property company's defenses concerning ownership of the revenue, costs, standing, and limitation; 12) after judgment, apply for enforcement, investigate corporate and related accounts, and seek compliance with the records-transfer obligation.
Precautions: Verify the authorization, contracts, and procedural stage. Establish common ownership with objective documentary evidence such as planning and registration records. The receipts from more than 200 owners may not cover the entire period, so request the ledgers at the same time. Prepare a month-by-month calculation of the RMB 1.37 million and link each amount to evidence. The property company bears the burden of proving reasonable costs with original records. Evidence of the demand or refusal to hand over is needed for the interest start date. Defects in standing can be strengthened through ratification or a joint action. If the contract assigns the revenue to the property company, examine the designated party, the owners' collective decision procedure, and the validity of that assignment. When the property company controls complete records but refuses to produce them, promptly ask the court to apply the rules on obstruction of proof.
Legal basis: Articles 274, 282, and 943 of the Civil Code of the People's Republic of China; Article 67 of the Civil Procedure Law of the People's Republic of China; the judicial interpretations of the Civil Procedure Law concerning the duty to produce documentary evidence and obstruction of proof; and the relevant provisions of the Property Management Regulations concerning the duties of the owners'

assembly and owners' committee and the property handover obligation.

E LIMITATION AND FUTURE WORK

First, our benchmark focuses on common domains of Chinese civil law. The generate–verify–revise pipeline is flexible and can be extended to other jurisdictions and legal domains. However, extending it to areas such as criminal law requires longer and more extensive review by experienced practicing lawyers, because errors may have especially serious consequences. We therefore leave broader cross-jurisdictional and criminal-law coverage to future work.

Second, CFCol assigns clarification planning and final advice generation to the cloud model, while the on-device agent executes the plan and coordinates the dialogue. It does not repeatedly invoke the cloud model to re-plan after every user response, which may limit adaptation when newly revealed facts change the relevant legal issues. Although the idealized single-run setting generally outperforms multi-turn interaction in our current experiments, adaptive multi-stage cloud reasoning could improve flexibility in more complex consultations and is an important direction for future work.

Third, our cost analysis accounts for cloud inference but excludes the computational, energy, and maintenance costs of the on-device agent. Future work should improve local-model efficiency and conduct large-scale deployment studies to characterize heterogeneous on-device costs across users and provide a more comprehensive assessment of end-to-end deployment costs.