INSCLAIMBENCH: BENCHMARKING INSURANCE CLAIM ADJUDICATION ACROSS THE DECISION CHAIN

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in reasoning-oriented large language models (LLMs) have motivated increasing evaluation of their ability to perform professional decision tasks. Insurance claim adjudication is one such task, requiring models to connect case evidence, insurance rules, intermediate judgments, and payout calculations across a structured decision process. We introduce **InsClaimBench**, an end-to-end benchmark for evaluating insurance claim adjudication across the decision chain. Grounded in real claim materials and structured insurance rules, InsClaimBench contains 3,780 cases in 375 case families across auto, property, and health insurance, comprising 86,656 atomic rule judgments. It evaluates each claim from atomic rules through adjudication modules to payout decisions and amounts, with controlled factual variants testing whether required changes are correctly propagated across levels. Evaluation of six LLMs reveals a progressive loss of reliability along the decision chain. Payout-decision accuracy ranges from 74.23–80.19%, while joint decision–amount accuracy drops to 47.54–73.15%. Strong local performance also fails to ensure case-level correctness: atomic-rule accuracy reaches 95.48%, whereas rule-vector exact match peaks at only 36.90%, and the most frequent module errors are not necessarily those most associated with final-decision failure. Under factual changes, these inconsistencies further become propagation failures: module updates are less reliable than rule updates, correct local judgments can still yield incorrect payouts, and correct payouts can conceal intermediate errors. These results show that reliable claim adjudication requires consistent composition and propagation across the decision chain.

1 INTRODUCTION

Recent advances in reasoning-oriented large language models (LLMs) have substantially improved their performance on complex reasoning tasks (Guo et al., 2025). This progress has motivated increasing attention to LLM reasoning and decision-making in professional domains, including financial reasoning, clinical diagnosis, legal adjudication, and autonomous driving (Xie et al., 2026; Zhu et al., 2026; Chen et al., 2026b; Ma et al., 2026). Such high-stakes professional decisions are rarely single predictions. They are chains of dependent judgments whose intermediate correctness and downstream consequences may diverge.

Insurance claim adjudication is one such professional decision task. It requires insurers to investigate case-specific facts, interpret applicable policy terms, determine coverage and causation, and assess the amount of loss or payment (Lin et al., 2024; Li et al., 2025). These determinations are interdependent: insurance liability depends on whether the loss falls within the insured risk and is not excluded, while the final settlement further depends on loss assessment and applicable payment conditions (Asmat & Tennyson, 2014; Huynh et al., 2015). Whether LLMs can effectively navigate these interconnected judgments and reach appropriate claim decisions and settlement outcomes has therefore emerged as an important question for further investigation.

Existing insurance benchmarks have progressively moved beyond knowledge-oriented question answering toward reasoning tasks grounded in specific insurance scenarios and business workflows (Feng et al., 2015; Chen et al., 2025; Lin et al., 2025; Chen et al., 2026a; Liu et al., 2026). Despite this shift toward scenario-based decision-making, existing benchmarks typically focus on selected subtasks or partial decision processes; for example, Liu et al. (2026) evaluates clause-grounded

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.

reasoning from case facts to claim verdicts, while Lin et al. (2025) evaluates predefined reasoning steps within specific insurance workflows. They therefore do not yet capture claim adjudication as a unified end-to-end decision process that connects underlying judgments, intermediate decisions, final claim outcomes, and their financial consequences, limiting the ability to analyze where LLMs succeed or fail along the complete decision chain.

To address this gap, we introduce **InsClaimBench**, an end-to-end benchmark for structured insurance claim adjudication (see Figure 1). InsClaimBench represents each claim as a connected decision chain spanning atomic rule judgments, adjudication modules, claim-level decisions, and payout calculation, with expert-defined logic linking the reference decisions across these levels. Grounded in real claim materials and structured insurance rules, the benchmark contains 3,780 cases organized into 375 case families across auto, property, and health insurance, comprising 86,656 atomic rule judgments. Beyond evaluating performance at individual stages, InsClaimBench constructs controlled factual variants that modify selected claim facts while holding unrelated evidence fixed, enabling systematic analysis of whether LLMs update affected judgments, preserve unaffected ones, and correctly propagate these changes to downstream claim decisions and payment outcomes.

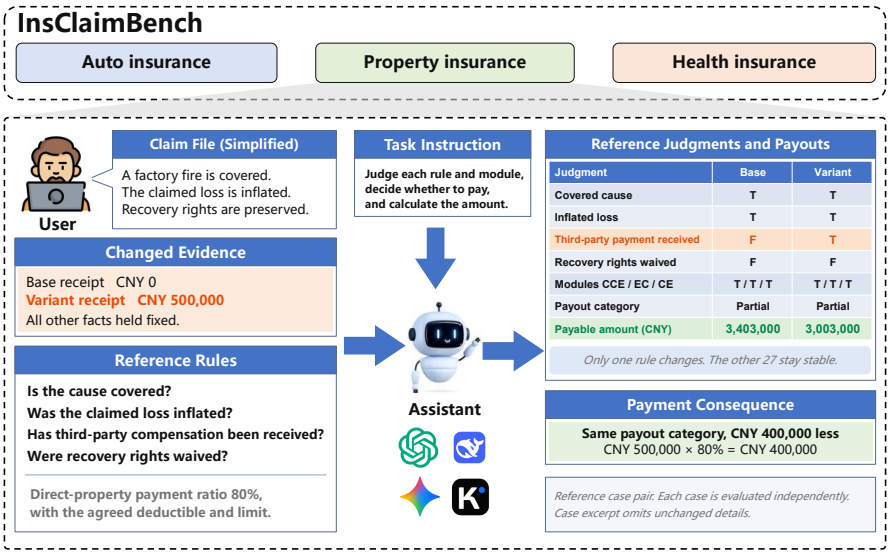

Figure 1: **Overview of InsClaimBench.** An example illustrating the end-to-end claim adjudication process, including atomic rule judgments, adjudication modules, claim decisions, and payout outcomes under controlled factual changes.

Our evaluation of six representative LLMs reveals four main findings. First, **current LLMs do not yet achieve consistently reliable performance on insurance claim adjudication**. Claim-decision accuracy ranges from 74.23% to 80.19%, while final payout amount accuracy varies more substantially from 47.54% to 73.15%. Second, **strong performance on individual judgments does not reliably translate into correct end-to-end adjudication**. Atomic rule accuracy reaches as high as 95.48%, yet the best model correctly resolves all applicable rules in only 36.90% of cases and reaches only 73.15% accuracy on final payout amounts. Third, **the difficulty of claim adjudication is uneven across the decision chain, with Event and Causation emerging as a consistent bottleneck**. It is the least accurate adjudication module for all six models, with accuracy ranging from 70.21% to 75.58%. Finally, **models exhibit substantial failures in propagating judgments through the decision chain**. Among cases requiring a module update, 28.57–39.98% of updates fail even when the original prediction is correct; moreover, correct payment outcomes can still conceal erroneous intermediate judgments.

Our contributions are threefold: (1) **Decision-chain evaluation framework.** We introduce **InsClaimBench**, connecting reference rule judgments, adjudication modules, payout decisions, and amounts through expert-defined logic for end-to-end evaluation and cross-level diagnosis. (2) **Controlled counterfactual evaluation.** We construct controlled case pairs to test required judgment

updates, stability of unaffected judgments, and correctness of downstream decisions and payouts under factual changes. (3) **Empirical diagnosis.** Across six LLMs, we show that high local accuracy does not ensure case-level correctness, the most frequent module errors need not be most associated with final-decision failure, and correct payouts can conceal intermediate errors.

2 Related Work

2.1 Insurance Benchmarks

Insurance benchmarks have gradually expanded from domain knowledge and document-based question answering to scenario-level reasoning and business-oriented tasks (Feng et al., 2015; Chen et al., 2025; Lin et al., 2025; Liu et al., 2026). This shift has introduced richer evaluation settings, including multi-step workflows, policy-grounded reasoning, and claim-level decisions. Nevertheless, most benchmarks still define their evaluation targets around individual capabilities or selected components of an insurance task, rather than the relationships among successive decisions. As a result, how performance and errors evolve from underlying judgments to final decisions and financial outcomes remains insufficiently characterized.

2.2 Process-Level and Counterfactual Evaluation

Beyond domain-specific benchmarks, reasoning evaluation has increasingly moved from outcome-level scoring toward examining the decision process itself. Two complementary directions have emerged: process-level evaluation localizes errors and assesses intermediate steps, rule application, or dependent task stages (Zheng et al., 2025; Song et al., 2025; Xie et al., 2026; Zhou et al., 2025; Chen et al., 2026b), while controlled counterfactual interventions test whether model reasoning and outputs respond consistently to prescribed changes (Roewer-Després et al., 2025; Han et al., 2026). These studies show that final-answer correctness can conceal deficiencies in intermediate reasoning and that errors may accumulate across dependent stages (Xie et al., 2026; Chen et al., 2026b; Han et al., 2026). However, the two perspectives are rarely integrated within a domain-defined professional decision process where changes in evidence have explicit implications for intermediate judgments and downstream consequences.

3 InsClaimBench

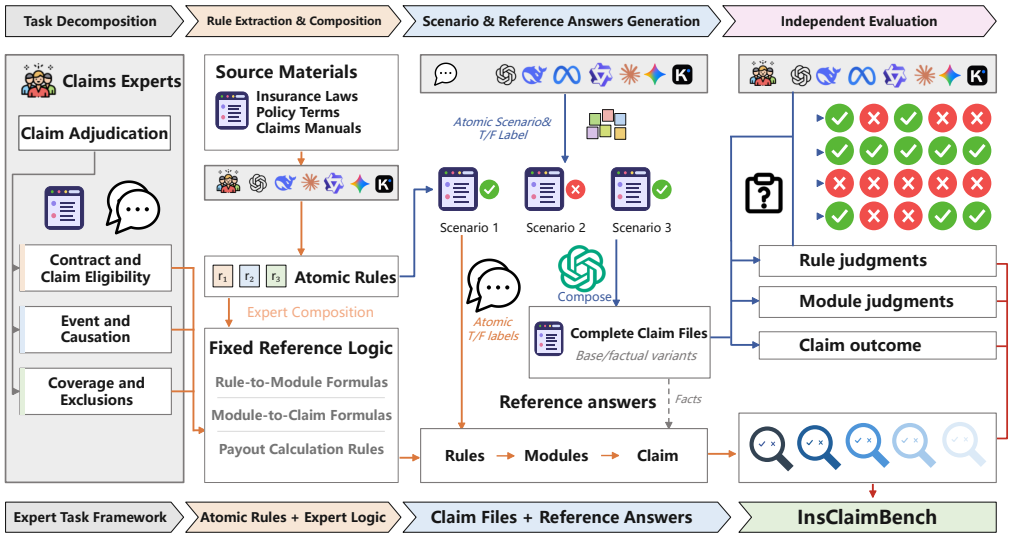

Figure 2: **Construction pipeline of InsClaimBench.** Expert-defined rules and logic guide scenario generation, reference answer construction, and independent evaluation at the rule, module, and claim levels.

Figure 2 presents the overall pipeline of InsClaimBench, covering task decomposition, rule extraction and composition, case and reference-answer generation, and independent evaluation. We first define claim adjudication as a hierarchical decision task and extract atomic rules from domain materials, with expert-defined logic connecting these rules to adjudication modules, claim outcomes, and payout calculations. The resulting rules and logic guide the construction of complete claim files and controlled factual variants, after which models and human evaluators are independently assessed at different levels of the decision chain.

3.1 Task Framework

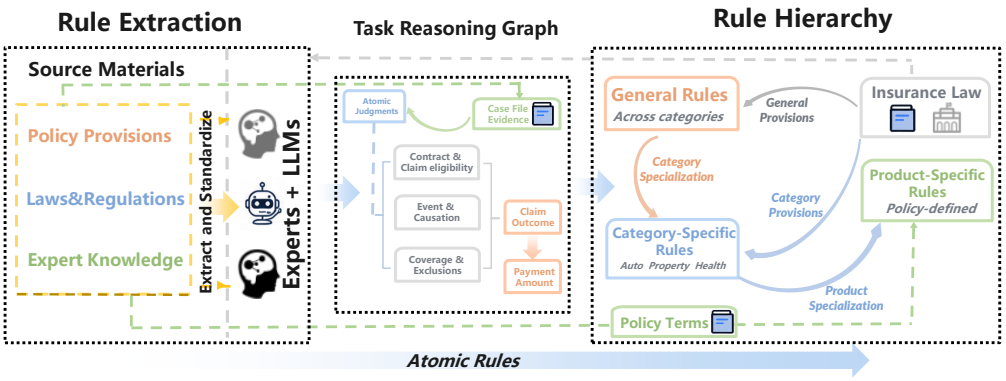

Figure 3: **Rule extraction and organization in InsClaimBench. Left**, claims experts and LLMs extract atomic rules from policy provisions, laws and regulations, and expert knowledge. **Center**, the reference decision structure links case evidence and atomic judgments to adjudication modules, claim outcomes, and payout amounts. **Right**, rules are organized into general, category-specific, and product-specific levels.

We defines insurance claim adjudication as an end-to-end task from a complete claim file to a final payout amount. To ensure that our benchmark closely aligns with real-world insurance applications, we have developed a top-down hierarchical task definition methodology. This framework incorporates a three-layer hierarchy:

Claim outcome level. Given a complete claim file, the overall task is to determine the payout decision and amount owed under the policy. The task covers auto, property, and health insurance. A predicted no-payout claim is assigned a predicted payout amount of zero in all payout-based evaluations.

Module level. An insurance claim is a request for payment following a loss that may be covered under an insurance contract. Claim adjudication therefore requires determining what loss occurred and whether the policy provides coverage for that loss. These determinations can be organized into 3 legally and factually distinct modules. *Contract and Claim Eligibility (CCE)* checks whether the contract and claimant meet the requirements for a claim. *Event and Causation (EC)* identifies the event, the cause of loss, and the causal chain. *Coverage and Exclusions (CE)* applies coverage conditions, exclusions, and exceptions.

Atomic rule level. The adjudication modules are further decomposed into atomic rules that can be judged True or False from the claim evidence. For example, a disclosure provision may require separate judgments about truthful disclosure, intentional or grossly negligent omission, its effect on underwriting, and the insurer's prior knowledge.

We then define a **counterfactual pair** as a base claim file and a variant whose selected facts are modified to reverse a targeted atomic judgment, while unrelated evidence and the governing rules remain fixed. Both cases have reference answers for atomic rules, adjudication modules, payment eligibility, and final amounts. The reference logic and calculation rules determine which downstream judgments and amounts should change and which should remain unchanged.

Table 1: Rules, rule judgments, and cases across insurance domains in InsClaimBench.

Insurance domain		Distinct rules	Total rule judgments	Cases
Motor		86	32,054	1,265
Property		28	35,000	1,250
Health	Critical illness	43	8,701	627
	Medical	48	10,901	638
Total		—	86,656	3,780 (375)

3.2 Dataset Construction

Atomic rule extraction. Claims experts and LLMs extract atomic rules from policy terms, insurance laws and regulations, and expert knowledge (Figure 3). Each rule specifies a proposition to judge from case evidence. Rules are organized into general, category-specific, and product-specific levels according to their applicability.

Expert-defined formulas. We combine the atomic judgments into module decisions and payment eligibility. The resulting judgments and case facts determine the reference payout amount under the applicable calculation rules. To improve transparency and enable limited replication of the reference-answer construction procedure, Appendix D provides a simplified and specified worked example from atomic judgments through module aggregation and payout calculation.

Case generation. LLM (in this paper, GPT-5.6-sol) generates atomic scenarios from the extracted rules and combines them into complete claim files. Each atomic scenario supplies evidence that establishes a reference T/F label for its associated rule.

Case family organization. To construct a counterfactual variant, the generator revises the facts supporting a selected atomic judgment so that its reference label reverses, while preserving unrelated facts in the base case. For each base case and variant, we use the atomic reference labels and case facts to compute module decisions, payment eligibility, and final amounts under fixed expert logic and calculation rules. Comparing the reference answers at the two endpoints identifies the required changes and invariances at each level. We group each base case with its variants into a case family and pair every variant with its base case. The resulting dataset contains 375 families comprising 3,780 cases and 3,405 base–variant pairs. Table 1 summarizes the distribution of rules, rule judgments, and cases across insurance domains.

3.3 Expert Review and Human Validation

Four claims experts, organized into two pairs, independently review a stratified sample of approximately 10% of case families without access to the reference answers. Their atomic judgments match the reference labels with 93.99% accuracy, with module accuracies of 93.47–97.61%, payout decision accuracy of 96.79%, and amount accuracy of 96.32% (Table 2). This correspondence validates, the correctness of the generated atomic scenarios and their associated answers, and the aggregation logic linking atomic judgments to downstream decisions and payouts. Two expert in the same group further achieve agreement rates of 91.54% and 91.30%, with Cohen's κ of 0.822 and 0.805 (Appendix A), supporting consistent interpretation of the generated cases and rules.

Table 2: Human evaluation on the sampled subset (%). Each metric is computed separately for each of the four experts on their assigned subset and then averaged across experts.

Rule		Module			Claim output		
Acc.	Vector EM	CCE	EC	CE	T/F Acc.	Amount Acc.	MAPE
93.99	92.46	93.47	97.61	95.13	96.79	96.32	2.61

4 EXPERIMENT

4.1 MODEL SELECTION

We evaluate six models from different model families. These include GPT-5.6-sol from OpenAI (OpenAI, 2026), Gemini-3.8-Flash from Google DeepMind (Google DeepMind, 2026), Qwen-3.8-Flash from Alibaba (Qwen Team, 2026), Kimi-K2.6 from Moonshot AI (Moonshot AI, 2026), DeepSeek-V4.1-Flash (DeepSeek-AI et al., 2026), and Qwen3.5-27B model. We evaluate all models on the same cases and rules. All inputs are text.

4.2 EVALUATION PROTOCOL

Models and human evaluators complete the adjudication in separate steps. They first judge each atomic rule, then independently assess the three modules, CCE, EC, and CE, and finally decide whether the claim is payable. For rule-level evaluation, each request contains the complete claim file and the original text of the target atomic rule. Module judgments and claim-level payout decisions are obtained through separate requests containing the complete claim file, without supplying the atomic rule texts or predictions from the rule-level evaluation. When a model judges a claim payable, a separate request to the same model uses the claim file and its directly predicted payout decision to compute the payout amount. Human evaluators likewise provide an amount when they judge a claim payable.

4.3 EVALUATION METRICS

Judgment scores. Rule accuracy pools all applicable rule judgments. Module accuracy and final T/F accuracy use all cases. *Rule-vector Exact Match* (*EM*) requires every rule in a case to be correct. R^+ and R^- denote cases with all rules correct and at least one rule error. D^+ and D^- denote correct and incorrect final decisions. The shares R^+D^- and R^-D^+ use all cases as the denominator. *Final fail.* is the percentage of wrong final decisions among cases with an error in the given module, with the error-case count in parentheses. *Top rules* are ranked by error count, and their error rates use occurrences within the same subset.

Payout amounts. Let $\mathcal{A} = \{i \mid \hat{c}_i = 1\}$ contain cases predicted payable and $\mathcal{A}_+ = \{i \in \mathcal{A} \mid q_i > 0\}$. We report mean absolute percentage error (MAPE) as

$$\mathrm{MAPE} = \frac{100}{|\mathcal{A}_+|} \sum_{i \in \mathcal{A}_+} \frac{|\hat{q}_i - q_i|}{q_i}. \tag{1}$$

For each base–variant pair $(x_i, x_{i'})$, we evaluate the two complete claim files independently. The reference answers define the expected changes and invariances in rule judgments, module decisions, and payout amounts.

For end-to-end evaluation, we set $\tilde{q}_i = \hat{q}_i$ for predicted payable claims and $\tilde{q}_i = 0$ otherwise, and report

$$\mathrm{Acc}_{\mathrm{joint}} = \frac{100}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \mathbf{1}[\hat{c}_i = c_i,\ |\tilde{q}_i - q_i| \le \tau_i], \qquad \tau_i = \max\{1,\ 0.0001 q_i\}, \tag{2}$$

where $\mathcal{J}$ contains 3,761 cases, excluding 19 cases with missing payout references.

Local response. Let z, z' be binary reference labels for the same rule or module, with hats denoting predictions. *Sensitivity* measures whether required changes occur, and *Invariance* measures whether judgments remain stable when required. Across case pairs, we define

$$\begin{aligned} \mathrm{Sensitivity} &= \Pr(\hat{z}' \neq \hat{z} \mid z' \neq z), \\ \mathrm{Invariance} &= \Pr(\hat{z}' = \hat{z} \mid z' = z). \end{aligned} \tag{3}$$

Pair correctness. *Sensitivity* and *Invariance* measure whether model predictions change when the reference label changes and remain unchanged when the reference label is unchanged. We then report pair correctness separately for changed and unchanged reference labels:

$$\mathrm{PC}_{\Delta} = \Pr(\hat{z} = z, \hat{z}' = z' \mid z' \neq z), \qquad \mathrm{PC}_{=} = \Pr(\hat{z} = z, \hat{z}' = z' \mid z' = z). \tag{4}$$

Both metrics require the predictions at the two endpoints to match their corresponding references. *Module Update Failure* (*MUF*) counts incorrect variant judgments among pairs where the base module prediction is correct and the reference label changes. *Payment Adjustment Failure* (*PAF*) and *Masked Module Error* (*MME*) use eligible case pairs, whose counts are shown as n. *PAF* requires at least one rule change and correct judgments for all changed rules at both endpoints. See more details in Appendix C. All scores are percentages.

Consequence propagation. We examine whether changes in rule judgments are correctly reflected in module decisions and final amounts. *Payment adjustment failure* occurs when all changed rules are judged correctly and the base amount is correct, but the variant amount is wrong. *Masked module error* occurs when the reference amount stays fixed and both predicted amounts are correct, yet a module whose reference label changes is judged incorrectly in either case. These checks capture errors in payment updates and intermediate judgments hidden by correct payments.

Generator comparison. Differences are final T/F accuracy on DeepSeek-generated cases minus accuracy on GPT-generated cases, in percentage points. We use paired family-bootstrap 95% confidence intervals, resampling matched case families with their base cases and variants kept together.

5 RESULTS

Through experiments we find LLMs can be accurate at individual adjudication judgments, yet reliability degrades when correctness must remain coherent across the decision hierarchy and under factual changes. This degradation is heterogeneous across the adjudication chain, because frequent local errors and errors associated with downstream failures do not necessarily occur at the same stage.

5.1 OVERALL ADJUDICATION PERFORMANCE

Binary payout decisions substantially overstate end-to-end adjudication reliability. Table 3 shows that payout decision accuracy ranges from 74.23% to 80.19%, whereas joint decision–amount accuracy ranges much more widely from 47.54% to 73.15%.

Aggregate performance masks substantial variation across claim types. As shown in Table 3, payout decision accuracy on medical claims ranges from 65.05% to 73.51% and is lower than the corresponding accuracy on the other claim types for all six models. The variation can be substantial within the same model. For example, Kimi-K2.6 reaches 86.80% accuracy on auto claims but 69.91% on medical claims. Overall payout accuracy therefore conceals considerable heterogeneity across adjudication settings.

Table 3: Payout decision and amount performance (%). Decision accuracy is shown overall and by insurance type. Amount metrics are pooled across insurance types.

	Payout decision accuracy					Payout calculation	
Model	Overall	Auto	Property	Illness	Medical	MAPE	Joint Acc.
GPT-5.6-Sol	80.05	82.61	79.52	82.62	**73.51**	**5.63**	69.72
Gemini-3.8-Flash	79.29	76.36	**84.24**	**84.53**	70.22	5.82	**73.15**
DeepSeek-V4.1-Flash	77.46	79.84	74.72	83.89	71.79	7.67	69.21
Qwen3.8-Flash	76.64	80.79	73.52	79.43	71.79	7.73	64.88
Kimi-K2.6	**80.19**	**86.80**	78.40	80.86	69.91	10.45	64.26
Qwen3.5-27B	74.23	80.40	72.16	75.28	65.05	28.76	47.54

5.2 FROM LOCAL JUDGMENTS TO CASE-LEVEL DECISIONS

Strong local judgment accuracy does not translate into reliable case-level adjudication. As shown in Table 4, individual rule accuracy remains high across models, yet the probability that all applicable rules are simultaneously correct drops sharply at the case level. For example, Kimi-K2.6 reaches 89.31% rule accuracy but only 14.68% exact match. High local accuracy therefore still permits substantial error accumulation within complete claims.

Local rule correctness and final payout decisions diverge in both directions. Across models, 1.98%–5.05% of cases have all applicable rules judged correctly but an incorrect final payout decision, while 48.20%–67.49% contain at least one incorrect rule judgment despite a correct final decision (Table 4). Final decision accuracy can therefore conceal substantial inconsistency in the underlying adjudication process.

Table 4: Rule accuracy and agreement with final T/F decisions (%). Metrics follow Section 4.3.

Model	Rule Acc.	**Vector EM**	R^+D^-	R^-D^+
GPT-5.6-Sol	**95.48**	**36.90**	5.05	48.20
Gemini-3.8-Flash	94.58	31.96	3.70	51.03
DeepSeek-V4.1-Flash	87.13	26.51	3.68	54.63
Qwen3.8-Flash	91.85	26.43	3.94	54.15
Kimi-K2.6	89.31	14.68	1.98	67.49
Qwen3.5-27B	91.48	15.03	2.65	61.85

5.3 Error Patterns Across Adjudication Modules

The most frequent module errors are not the errors most strongly associated with final-decision failure. As shown in Table 5, *Event and Causation* is consistently the least accurate module and accounts for the largest number of module errors across all six models, while *Contract and Claim Eligibility* and *Coverage and Exclusions* achieve substantially higher accuracy.

Errors in different adjudication modules have different relationships with final decision failure. Despite being the most frequent, *Event and Causation* errors are generally less often accompanied by an incorrect final decision than errors in the other two modules. For GPT-5.6-Sol, final-decision failure occurs in 42.25% of cases with an *Event and Causation* error, compared with 62.63% for *Contract and Claim Eligibility* and 59.05% for *Coverage and Exclusions*. Module accuracy and downstream failure therefore capture different aspects of adjudication reliability. These static associations do not reveal whether models update the adjudication chain consistently when the underlying evidence changes. We therefore turn to controlled factual variants.

Table 5: Module accuracy and final T/F failure when a module is wrong (%). Error-case counts are in parentheses. Metrics follow Section 4.3.

Model	CCE		EC		CE	
	Acc.	Final fail. (n)	Acc.	Final fail. (n)	Acc.	Final fail. (n)
GPT-5.6-Sol	**92.57**	62.63 (281)	73.39	42.25 (1,006)	91.67	59.05 (315)
Gemini-3.8-Flash	86.06	51.42 (527)	**75.58**	41.60 (923)	**92.22**	48.98 (294)
DeepSeek-V4.1-Flash	78.68	40.94 (806)	70.21	29.13 (1,126)	83.57	42.83 (621)
Qwen3.8-Flash	83.36	48.97 (629)	73.49	38.62 (1,002)	87.49	53.70 (473)
Kimi-K2.6	80.74	34.07 (728)	71.14	27.68 (1,091)	88.07	52.55 (451)
Qwen3.5-27B	89.37	36.07 (402)	70.85	42.20 (1,102)	84.26	71.26 (595)

5.4 Decision Propagation under Factual Changes

Controlled factual changes expose a progressive loss of reliability as required updates propagate through the adjudication chain, from atomic rules to module decisions and ultimately to payout outcomes (Table 6).

Adjudication reliability weakens when factual changes must be propagated from atomic rules to module-level decisions. As shown in Table 6, module-level PC_{Δ} is consistently 9.92–17.43 percentage points lower than rule-level $PC_{=}$ across all six models. Even when the base module prediction is correct, 28.57%–39.98% of required module updates are incorrect. Module judgments are also less stable when no update is required, with module-level $PC_{=}$ consistently below rule-level $PC_{=}$. This gap reflects both failures to update modules when their reference decisions change

and spurious changes when they should remain stable. Stronger performance on local rule updates therefore does not consistently carry over to higher-level adjudication judgments.

Correctly handling changed atomic rules does not guarantee a correct downstream payout amount. Even when all changed rules are judged correctly at both endpoints and the base payout amount is correct, 8.69%–18.89% of eligible pairs still produce an incorrect variant amount (Table 6).

Correct payout outcomes can likewise conceal failures in intermediate adjudication decisions. Among pairs with correct payout amounts at both endpoints and at least one required module change, a substantial fraction still contains an incorrect judgment in a module whose reference label changes, reaching 68.03% for some models (Table 6).

Table 6: Counterfactual responses and payout outcomes on case pairs (%). Eligible pair counts for PAF and MME are in parentheses. Metrics follow Section 4.3.

Model	Rule				Module					Payout outcomes	
	Sens.	Inv.	PC_{Δ}	$PC_{=}$	Sens.	Inv.	PC_{Δ}	$PC_{=}$	MUF	PAF (n)	MME (n)
GPT-5.6-Sol	84.80	96.40	84.22	94.32	67.79	89.68	67.71	84.67	29.55	11.25 (1,813)	34.23 (111)
Gemini-3.8-Flash	84.04	97.06	83.75	93.71	69.08	90.99	69.08	84.10	28.57	9.27 (2,027)	27.41 (135)
DeepSeek-V4.1-Flash	71.48	92.86	70.68	84.53	61.76	83.77	60.76	73.13	34.05	8.69 (1,415)	68.03 (147)
Qwen3.8-Flash	76.80	94.27	76.14	89.84	60.38	88.60	59.85	79.93	36.00	10.81 (1,453)	66.96 (112)
Kimi-K2.6	71.93	94.00	71.38	87.30	59.69	88.78	59.54	78.87	37.00	18.89 (1,212)	49.63 (135)
Qwen3.5-27B	69.66	97.35	68.88	91.15	51.83	93.83	51.45	82.28	39.98	14.39 (848)	61.54 (117)

PC_{Δ} and $PC_{=}$ denote pair correctness for changed and unchanged reference labels, respectively: both endpoint predictions must be correct. Local metrics are pooled over rule or module positions across the same 3,405 case pairs.

5.5 Robustness to Case Generator

To assess whether our case generation procedure depends on a specific generator, we conduct an ablation study by replacing GPT-5.6-Sol with DeepSeek-V4-pro. We evaluate the same six models on both sets of generated cases across four insurance types. As shown in Table 10, changing the generator leads to only small variations in payout decision accuracy. Across all 24 comparisons, the absolute differences are within 1.09 percentage points, ranging from -0.95 to $+1.09$ points, and all paired family-bootstrap 95% confidence intervals include zero. These results indicate that the evaluation outcomes remain stable across the two case generators, supporting the robustness of our case generation procedure.

6 Discussion and Conclusion

InsClaimBench evaluates insurance claim adjudication across atomic rule judgments, adjudication modules, payout decisions, and amounts, using expert-defined reference logic and controlled factual variants. Across six LLMs, our results reveal a gap between binary decision accuracy and joint decision–amount accuracy. High local judgment accuracy does not ensure correct case-level adjudication, while module error frequency and association with final-decision failure reveal different weaknesses. Under factual changes, module judgments are less reliable than atomic rule judgments. Correct judgments on changed rules can still accompany incorrect variant payouts, while correct payouts can conceal intermediate errors. These findings demonstrate the importance of evaluating judgment correctness, responses to factual changes, and payment outcomes together across the decision chain.

However, InsClaimBench is currently based on synthetic claim files, which allows precise control over factual changes and reference answers but does not fully capture the ambiguity of real-world claim documentation. Original claim files may contain missing, inconsistent, or conflicting evidence, making adjudication substantially less controlled. Future work can therefore extend the benchmark to real claim documents and examine whether the judgment-update failures and gaps between decision correctness and payment correctness observed here persist under more realistic uncertainty.

AI USE STATEMENT

We used generative AI for synthetic case generation, counterfactual rewriting, and annotation in constructing InsClaimBench. As described in Section 3, LLMs assisted claims experts in extracting atomic rules and preparing rule-level reference annotations. GPT-5.6-sol generated atomic scenarios and complete claim files, and revised selected facts to construct counterfactual variants. DeepSeek-V4-pro was used as an alternative case generator in the ablation study. Reference module decisions, payment eligibility, and payout amounts were computed from atomic reference labels and case facts using fixed expert-defined logic and calculation rules. Generative AI tools were also used to assist with language editing and refinement of the manuscript, including improving clarity, fluency, and presentation, and with visual style design for figures, such as layout, visual organization, and stylistic refinement. The underlying scientific content, experimental results, figure data, and interpretations were determined and verified by the authors.

REFERENCES

Danial P Asmat and Sharon Tennyson. Does the threat of insurer liability for "bad faith" affect insurance settlements? *Journal of Risk and Insurance*, 81(1):1–26, 2014.

Changyu Chen, Chenwei Lin, and Xian Xu. Ins-actbench: A comprehensive benchmark for assessing professional actuarial capability of large language models. *arXiv preprint arXiv:2607.24273*, 2026a.

Shisong Chen, Qian Zhu, Wenyan Yang, Chengyi Yang, Zhong Wang, Ping Wang, Xuan Lin, Bo Xu, Daqian Li, Chao Yuan, et al. Inseva: A comprehensive chinese benchmark for large language models in insurance. *arXiv preprint arXiv:2509.04455*, 2025.

Ziang Chen, Guannan Li, Fanlin Ji, Yipeng Kang, Jiaqi Li, Muhan Zhang, Yangtao Zhang, Li Tianjiao, Jiannan Wang, Xin Guo, et al. Jurisbench: A deep benchmark for assessing large language models in professional legal practice. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35994–36018, 2026b.

DeepSeek-AI et al. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. doi: 10.48550/arXiv.2606.19348. URL https://arxiv.org/abs/2606.19348.

Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, pp. 813–820. IEEE, 2015.

Google DeepMind. Gemini 3.8 Flash Model Card. Official model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/. Accessed September 24, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Yunseok Han, Yejoon Lee, and Jaeyoung Do. Rfeval: Benchmarking reasoning faithfulness under counterfactual reasoning intervention in large reasoning models. *arXiv preprint arXiv:2602.17053*, 2026.

Mirabelle Huynh, David Landriault, Tianxiang Shi, and Gordon E Willmot. On a risk model with claim investigation. *Insurance: Mathematics and Economics*, 65:37–45, 2015.

Dongchen Li, Zhuo Jin, Linyi Qian, and Hailiang Yang. Textual analysis of insurance claims with large language models. *Journal of Risk and Insurance*, 92(2):505–535, 2025.

Chenwei Lin, Hanjia Lyu, Jiebo Luo, and Xian Xu. Harnessing gpt-4v (ision) for insurance: A preliminary exploration. *arXiv preprint arXiv:2404.09690*, 2024.

Chenwei Lin, Hanjia Lyu, Xian Xu, and Jiebo Luo. Ins-mmbench: A comprehensive benchmark for evaluating lvlms' performance in insurance. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9036–9047. IEEE, 2025.

Jin Liu, Yunpeng Liu, Keyi Wang, Jie Shi, Xiao Xu, Wenkang Huang, Xingzhong Xu, Xin Liang, and Yanghua Xiao. Inslogicbench: An argumentation logic grounded benchmark for complex insurance claims adjudication. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22592–22619, 2026.

Enhui Ma, Jiahuan Zhang, Guantian Zheng, Tao Tang, Shengbo Eben Li, Yuhang Lu, Xia Zhou, Xueyang Zhang, Yifei Zhan, Kun Zhan, et al. Drivecombo: Benchmarking compositional traffic rule reasoning in autonomous driving. *arXiv preprint arXiv:2603.01637*, 2026.

Moonshot AI. Kimi K2.6. Official model card on Hugging Face, 2026. URL https://huggingface.co/moonshotai/Kimi-K2.6. Accessed September 24, 2026.

OpenAI. GPT-5.6 Sol. OpenAI API model documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol. Accessed September 24, 2026.

Qwen Team. Latest model: Qwen3.8-Flash. QwenCloud model documentation, 2026. URL https://docs.qwencloud.com/developer-guides/getting-started/latest-model. Accessed September 24, 2026.

François Roewer-Després, Jinyue Feng, Zining Zhu, and Frank Rudzicz. Accord: Closing the commonsense measurability gap. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3799–3829, 2025.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. Prmbench: A fine-grained and challenging benchmark for process-level reward models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25299–25346, 2025.

Zhuohan Xie, Daniil Orel, Rushil Thareja, Dhruv Sahnan, Hachem Madmoun, Fan Zhang, Debopriyo Banerjee, Georgi Nenkov Georgiev, Xueqing Peng, Lingfei Qian, et al. Finchain: A symbolic benchmark for verifiable chain-of-thought financial reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14529–14553, 2026.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, 2025.

Ruiwen Zhou, Wenyue Hua, Liangming Pan, Sitao Cheng, Xiaobao Wu, En Yu, and William Yang Wang. Rulearena: A benchmark for rule-guided reasoning with llms in real-world scenarios. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 550–572, 2025.

Yakun Zhu, Zhongzhen Huang, Linjie Mu, Yutong Huang, Wei Nie, Jiaji Liu, Shaoting Zhang, Pengfei Liu, and Xiaofan Zhang. Diagnosisarena: benchmarking diagnostic reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3074–3098, 2026.

A EXPERT QUALITY VERIFICATION

To independently verify the quality of the benchmark references, we conduct a blinded expert evaluation. We stratify case families by insurance type and randomly sample approximately 10% of the families, while retaining the complete base–variant structure within each sampled family. The sampled families are divided into two disjoint subsets, each covering approximately 5% of the benchmark.

Four claims experts are organized into two pairs. The two experts within each pair independently evaluate the same subset. They receive the complete claim files, original atomic rules, and applicable calculation rules, but do not have access to the benchmark reference labels, rationales, module decisions, payout decisions, or reference amounts. For each sampled case, the experts judge all applicable atomic rules, directly determine *CCE*, *EC*, and *CE*, decide payment eligibility, and calculate the payout amount when applicable.

For each pair, we measure inter-expert agreement on binary judgments using raw agreement and Cohen's κ. As shown in Table 8, the two expert pairs achieve agreement rates of 91.54% and 91.30%, with Cohen's κ values of 0.822 and 0.805, respectively. These results indicate a high degree of independent reproducibility in the expert judgments.

Raw agreement is the proportion of paired judgments on which the two experts give the same label, while Cohen's κ additionally adjusts for agreement expected from the marginal label distributions. Regarding compensation, the human experts were compensated at a rate equivalent to 35 days of their regular pay.

Table 7: Distribution of expert judgments in human validation.

Expert	T	F
A	3,212	2,204
B	3,418	1,998
C	3,300	1,791
D	3,477	1,614

Table 8: Inter-expert agreement in blinded expert quality verification.

Group 1					**Group 2**				
	A:T	A:F	Agreement(%)	κ		C:T	C:F	Agreement(%)	κ
B:T	3,086	332	91.54	0.822	D:T	3,167	310	91.30	0.805
B:F	126	1,872			D:F	133	1,481		

B More Results

B.1 Top Rule Errors

Table 9: Frequent rule errors (%). Top rules are ranked by error count within each subset. All listed rules are from property insurance. Metrics follow Section 4.3.

Model	Top rule (error rate)	
	All cases	$R^- D^+$
GPT-5.6-Sol	C07 (44.00)	C07 (43.91)
Gemini-3.8-Flash	C07 (39.36)	P07 (44.89)
DeepSeek-V4.1-Flash	P01 (76.56)	P01 (77.11)
Qwen3.8-Flash	P01 (75.84)	P01 (75.25)
Kimi-K2.6	P21 (63.52)	P21 (68.00)
Qwen3.5-27B	C07 (55.60)	C07 (50.35)

Table 9 further shows that C07, which evaluates whether an expense falls under a contractual exclusion, is among the most persistent errors in property claims. For GPT-5.6-Sol, C07 is incorrect in 44.00% of all relevant property cases and remains incorrect in 43.91% of cases where rule errors coexist with a correct final decision. Correct case-level outcomes therefore do not imply that recurring local reasoning errors have been resolved.

B.2 Result of Ablation Study

Table 10: Case-generator ablation. Entries are final T/F accuracy differences (DeepSeek minus GPT, percentage points), with paired family-bootstrap 95% confidence intervals.

Insurance type		GPT-5.6 Sol	Gemini-3.8 Flash	DeepSeek Flash	Qwen3.8 Flash	Kimi-K2.6	Qwen3.5-27B
Auto		+0.00 [-1.11, +1.03]	-0.95 [-2.06, +0.16]	-0.79 [-1.90, +0.24]	+0.79 [-0.24, +1.90]	+0.16 [-0.95, +1.26]	-0.47 [-1.66, +0.71]
Property		+0.48 [-0.50, +0.75]	-0.24 [-0.78, +0.20]	-0.16 [-0.59, +0.09]	-0.08 [-0.57, +0.42]	-0.24 [-0.88, +0.81]	+0.24 [-0.33, +0.66]
Health	Illness	+0.77 [-0.67, +2.23]	+0.14 [-1.28, +1.44]	+1.09 [-0.64, +2.85]	+1.09 [-0.48, +2.68]	-0.51 [-2.07, +0.93]	-0.52 [-2.07, +0.92]
	Medical	-0.41 [-1.50, +0.69]	-0.12 [-1.23, +0.98]	+0.21 [-0.88, +1.30]	-0.43 [-1.52, +0.67]	-0.44 [-1.53, +0.65]	-0.17 [-1.27, +0.94]

C Formal Definitions of Evaluation Metrics

Notation for counterfactual pairs. Let $\mathcal{P}$ denote the set of base–variant case pairs. For each pair $p \in \mathcal{P}$, unprimed and primed quantities refer to the base and variant cases, respectively. Let r_{pj} and r'_{pj} denote the binary reference labels for rule j, and let m_{pk} and m'_{pk} denote the binary reference labels for module $k \in \mathcal{K} = \{\mathrm{CCE}, \mathrm{EC}, \mathrm{CE}\}$. Hats denote the corresponding predictions. Rules are matched across the two cases by rule identity.

The sets of rules and modules whose reference labels change are

$$\Delta_p^R = \{j \mid r'_{pj} \neq r_{pj}\}, \qquad \Delta_p^M = \{k \in \mathcal{K} \mid m'_{pk} \neq m_{pk}\}. \tag{5}$$

Let q_p, q'_p denote the reference amounts and $\hat{q}_p, \hat{q}'_p$ the predicted amounts used in the pairwise amount evaluation. Amount correctness at the two endpoints is defined separately using the corresponding reference amount:

$$\begin{aligned} C_p &= \mathbf{1}\big\{|\hat{q}_p - q_p| \leq \max\{1,\, 0.0001 q_p\}\big\}, \\ C'_p &= \mathbf{1}\big\{|\hat{q}'_p - q'_p| \leq \max\{1,\, 0.0001 q'_p\}\big\}. \end{aligned} \tag{6}$$

All probabilities below are empirical proportions over the specified evaluation units and are multiplied by 100 to obtain percentages.

Module Update Failure (MUF). MUF measures how often a variant module prediction is incorrect when the base prediction is correct and the reference module label changes:

$$\mathrm{MUF} = 100\,\widehat{\Pr}_{(p,k)}\left(\hat{m}'_{pk} \neq m'_{pk} \,\middle|\, \hat{m}_{pk} = m_{pk},\ m'_{pk} \neq m_{pk}\right). \tag{7}$$

The evaluation unit is a pair–module position (p, k). The denominator pools all positions with a correct base prediction and a changed reference label across the three modules and all case pairs.

Payment Adjustment Failure (PAF). Let B_p denote the event that at least one reference rule label changes and every changed rule is predicted correctly at both endpoints:

$$B_p = \left\{\Delta_p^R \neq \varnothing\right\} \cap \bigcap_{j \in \Delta_p^R} \left\{\hat{r}_{pj} = r_{pj},\ \hat{r}'_{pj} = r'_{pj}\right\}. \tag{8}$$

PAF measures the frequency of an incorrect variant amount conditional on B_p and a correct base amount:

$$\mathrm{PAF} = 100\,\widehat{\Pr}_p\left(C'_p = 0 \,\middle|\, B_p,\ C_p = 1\right). \tag{9}$$

The evaluation unit is a case pair. The eligible-pair count reported alongside PAF is

$$n_{\mathrm{PAF}} = \sum_{p \in \mathcal{P}} \mathbf{1}\{B_p \text{ holds},\ C_p = 1\}. \tag{10}$$

This definition does not require $q'_p \neq q_p$. It therefore includes failures to preserve a reference amount that should remain unchanged, as well as failures to produce a required amount change.

Masked Module Error (MME). MME measures how often correct amounts at both endpoints coexist with an incorrect prediction for a module whose reference label changes. Eligible pairs must have identical reference amounts and at least one changed module:

$$\begin{aligned}\mathrm{MME} = 100\,\widehat{\Pr}_p\Big(&\exists k \in \Delta_p^M,\ (\hat{m}_{pk} \neq m_{pk} \vee \hat{m}'_{pk} \neq m'_{pk}) \\ &\Big|\ C_p = C'_p = 1,\ q_p = q'_p,\ \Delta_p^M \neq \varnothing\Big).\end{aligned} \tag{11}$$

The evaluation unit is a case pair, and each eligible pair is counted once even if multiple changed modules contain errors. The eligible-pair count reported alongside MME is

$$n_{\mathrm{MME}} = \sum_{p \in \mathcal{P}} \mathbf{1}\left\{C_p = C'_p = 1,\ q_p = q'_p,\ \Delta_p^M \neq \varnothing\right\}. \tag{12}$$

D Worked Example of Reference-Answer Construction

This section provides a fully specified illustrative example of deterministic reference-answer aggregation and payout calculation.

D.1 Illustrative Scope

To make the construction of reference answers inspectable without releasing the complete proprietary rule inventory, we provide a fully specified worked example from a frozen illustrative medical-insurance product. The example exposes the complete computation from atomic rule labels to module judgments, claim-level payment eligibility, and the reference payout amount. All quantities and logical operations required to reproduce the example are given below.

This example is intended to demonstrate the deterministic aggregation procedure used in InsClaim-Bench. Product-specific rule inventories, source mappings, and the full set of expert composition formulas are not disclosed.

For the illustrative medical claim, let $r_1, \ldots, r_{10}$ denote binary atomic judgments. A value of T means that the proposition represented by the rule is satisfied by the case evidence.

For each contractual exclusion j, we additionally define four binary judgments. Here, e_j indicates that the factual condition described by the exclusion is present, k_j indicates that the required relation between that condition and the claimed loss is established, v_j indicates that the exclusion is

Rule	Proposition represented by T
r_1	The claimant and policy satisfy the applicable eligibility requirements, and the relevant coverage is in force.
r_2	The event falls within the insurance period or an applicable contractual extension.
r_3	A valid disclosure-related bar to coverage is established.
r_4	A medical event covered by the relevant benefit has occurred.
r_5	The claimed medical expenses are related to the event, reasonable, and necessary.
r_6	The waiting-period provision applies and the event falls within the waiting period.
r_7	An accident-related exception to the waiting period applies.
r_8	The policy otherwise waives the applicable waiting period.
r_9	The hospital, geographic scope, and expense categories satisfy the applicable coverage requirements.
r_{10}	A statutory bar that independently precludes payment is established.

Table 11: Atomic judgments used in the illustrative medical-insurance example.

legally and contractually effective, and z_j indicates that an exception restoring coverage under that exclusion applies.

An exclusion blocks coverage only when

$$X_j = e_j \wedge k_j \wedge v_j \wedge \neg z_j. \tag{13}$$

The three adjudication modules are then constructed as

$$M_{\mathrm{CCE}} = r_1 \wedge r_2 \wedge \neg r_3, \tag{14}$$

$$M_{\mathrm{EC}} = r_4 \wedge r_5 \wedge (\neg r_6 \vee r_7 \vee r_8), \tag{15}$$

$$M_{\mathrm{CE}} = r_9 \wedge \neg r_{10} \wedge \neg \bigvee_j X_j. \tag{16}$$

The reference claim-level payment decision is

$$c = M_{\mathrm{CCE}} \wedge M_{\mathrm{EC}} \wedge M_{\mathrm{CE}}. \tag{17}$$

This construction also makes explicit how exceptions operate. For example, $r_6 = T$ indicates that the event falls within an applicable waiting period. The waiting-period condition nevertheless passes when either $r_7 = T$ or $r_8 = T$. Likewise, an exclusion whose factual, relational, and validity requirements are all satisfied does not block coverage when its corresponding exception $z_j = T$.

D.2 Worked Logical Example

Consider an illustrative claim with the following reference atomic judgments:

$$r_1 = T, \quad r_2 = T, \quad r_3 = F, \tag{18}$$

$$r_4 = T, \quad r_5 = T, \quad r_6 = T, \tag{19}$$

$$r_7 = T, \quad r_8 = F, \quad r_9 = T, \quad r_{10} = F. \tag{20}$$

Suppose one potentially relevant exclusion is present and its atomic judgments are

$$e_1 = T, \qquad k_1 = T, \qquad v_1 = T, \qquad z_1 = T. \tag{21}$$

The corresponding exclusion indicator is therefore

$$X_1 = T \wedge T \wedge T \wedge \neg T = F. \tag{22}$$

The module references are obtained deterministically as

$$M_{\mathrm{CCE}} = T \wedge T \wedge \neg F = T, \tag{23}$$

$$M_{\mathrm{EC}} = T \wedge T \wedge (\neg T \vee T \vee F) = T, \tag{24}$$

$$M_{\mathrm{CE}} = T \wedge \neg F \wedge \neg F = T. \tag{25}$$

Hence,

$$c = T \wedge T \wedge T = T, \tag{26}$$

and the claim proceeds to payout calculation.

This example illustrates two exception mechanisms simultaneously. The event occurs during the waiting period, but the accident-related exception preserves eligibility. A potentially applicable exclusion is also triggered at the factual level, but the exclusion-specific exception prevents that exclusion from blocking coverage.

D.3 Worked Payout Calculation

Suppose the submitted medical bill is $24{,}000$. Of this amount, $2{,}000$ falls outside the covered expense scope and $10{,}000$ has already been matched to other compensation. The expense entering the benefit formula is therefore

$$x = 24{,}000 - 2{,}000 - 10{,}000 = 12{,}000. \tag{27}$$

For this frozen illustrative product, the applicable medical benefit schedule reimburses the first CNY $10{,}000$ of x at 50% and the remaining eligible amount at 100%. Thus,

$$G(x) = 0.5 \min(x, 10{,}000) + \max(x - 10{,}000, 0). \tag{28}$$

Substituting $x = 12{,}000$ gives

$$G(12{,}000) = 0.5 \times 10{,}000 + 2{,}000 = 7{,}000. \tag{29}$$

Let H denote the remaining applicable policy limit. The adjudicated benefit is

$$P = \min\{G(x), H\}. \tag{30}$$

When $H \geq 7{,}000$, the reference adjudicated amount is therefore

$$P = 7{,}000. \tag{31}$$

If only CNY $6{,}500$ remains under the applicable limit, the same calculation instead yields

$$P = 6{,}500. \tag{32}$$

Where an advance payment V has already been made against the same adjudicated claim, the adjudicated benefit P remains unchanged and the amount transferred in the current settlement is

$$N = \max(P - V, 0). \tag{33}$$

For example, if $P = \text{CNY } 7{,}000$ and $V = \text{CNY } 1{,}000$, then the current transfer is

$$N = 6{,}000. \tag{34}$$

Here N denotes the reference amount q used in the evaluation metrics.