LIUYAO-BENCH: A HIERARCHICAL BENCHMARK FOR EXPERT-DOMAIN VERIFICATION IN CLASSICAL CHINESE TEXTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Liuyao is a rule-governed expert-text tradition in classical Chinese where judgments depend on hexagram configurations, calendrical context, line relations, and transformations. Evaluating large language models (LLMs) in this domain requires examining both their knowledge of individual rules and their ability to verify interpretations of specific cases. We introduce LIUYAO-BENCH, a hierarchical benchmark containing 3,060 binary verification instances drawn from four classical *liuyao* texts. The benchmark organizes evaluation into three layers: (1) basic knowledge, covering concepts, relations, and rule statements; (2) intermediate-state interpretation, covering *yongshen* selection, line strength, and dynamic line relations; and (3) full-case judgment verification, checking candidate final judgments against structured case records. All three layers ask models to assess a supplied candidate, allowing comparison across different scopes of verification. Across ten contemporary LLMs, the best model reaches 79.77% overall accuracy, while dynamic relations and complete cases remain difficult. On a balanced 300-item subset, two practitioners achieve 88.67% mean accuracy, compared with 79.00% for the best model on the same items. Both practitioners outperform all evaluated models on Dynamics and Full-Case. Additional experiments with Qwen3-8B show that chain-of-thought prompting improves overall benchmark accuracy, while fine-tuning improves overall accuracy on a held-out source book; full-case performance remains limited in both settings. LIUYAO-BENCH provides a structured setting for examining the gap between local rule knowledge and case-level verification in a classical Chinese expert domain.

1 INTRODUCTION

Liuyao is a traditional Chinese divinatory text tradition in which a querier asks about a concrete situation and a practitioner derives a judgment about its possible development or outcome through rule-based interpretation. A typical case record may include a query, a main hexagram, a changed hexagram, moving lines, and calendrical information. Interpreting such a record requires identifying relevant variables, interpreting temporal conditions, tracking line relations and transformations, and reconciling intermediate states with a final case judgment. *Liuyao* is therefore neither a standard language-understanding task nor a fully formal reasoning system. It is a rule-governed expert-text domain in which knowledge, procedure, and interpretation are tightly intertwined.

These properties make *liuyao* a useful stress test for LLMs. A model must do more than recognize isolated concepts or emit a plausible final label. It must connect rule statements to case conditions, verify candidate intermediate states, and determine whether a proposed final judgment is compatible with the case description. Correctly verifying an isolated rule does not guarantee that a model will apply it consistently when multiple case variables interact. For example, knowing that wood generates fire is only a statement about a relation. In a case, the model must identify which line has each attribute and whether the candidate describes their relation correctly. A final judgment then requires interpreting that relation in light of the question.

Despite growing evaluation work for Chinese LLMs, this form of expert-domain verification remains underrepresented. Broad Chinese benchmarks are useful for measuring general language ability,

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.

but they are usually centered on final-answer correctness rather than layered diagnostic evaluation (Huang et al., 2023; Li et al., 2024). Benchmarks for classical Chinese mostly focus on language understanding and text processing (Zhou et al., 2023; Chen et al., 2025), while culturally grounded benchmarks often emphasize knowledge recall or context-sensitive judgment (Sun et al., 2024). Much less attention has been paid to domains where models must move from source-grounded rules to structured intermediate states and then to case-level judgment.

To address this gap, we introduce LIUYAO-BENCH,[1] a hierarchical benchmark for evaluating LLMs on *liuyao* texts, as illustrated in Figure 1. To the best of our knowledge, this is the first hierarchical benchmark for evaluating LLMs on *liuyao* texts. Its source collection draws on four books: *Zeng Shan Bu Yi* (《增删卜易》), *Bu Shi Zheng Zong* (《卜筮正宗》), *Yi Mao* (《易冒》), and *Duan Yi Tian Ji* (《断易天机》). The benchmark comprises three binary verification layers: basic knowledge, which tests concept/relation and rule statements; intermediate-state interpretation, which tests candidate intermediate judgments; and full-case judgment verification, which tests candidate final judgments against structured case records.

This paper makes three contributions. First, it introduces an evaluation domain centered on a culturally grounded and rule-governed expert-text tradition that is underrepresented in existing Chinese, classical-Chinese, and expert-domain benchmarks. Second, it proposes a hierarchical verification design that separates local rule knowledge from task-native intermediate variables and final case-level compatibility. Third, it provides baselines from ten models and a matched practitioner reference. Additional Qwen3-8B experiments compare prompting and fine-tuning. The evaluations report performance on domain knowledge, intermediate judgments, and complete cases.

2 RELATED WORK

2.1 CHINESE AND CLASSICAL-CHINESE BENCHMARKS

Chinese evaluation resources cover a broad range of linguistic and cultural competencies. CLUE (Xu et al., 2020) evaluates general language understanding, while C-Eval (Huang et al., 2023) and CMMLU (Li et al., 2024) assess knowledge across academic and professional subjects. For classical Chinese, WYWEB (Zhou et al., 2023) covers tasks such as sequence labeling and reading comprehension, WenMind (Cao et al., 2024) evaluates classical literature and language arts, and PoetMT (Chen et al., 2025) focuses on poetry translation. CHENGYU-BENCH (Fu et al., 2025) and CHARM (Sun et al., 2024) further examine idiomatic and culturally situated knowledge.

Our focus is a different evaluation target: whether models can verify judgments that combine domain rules with the conditions of a particular case. In *liuyao*, this requires distinguishing statements about individual rules from statements about intermediate states and final case outcomes. For example, knowing what a six-relative category means differs from deciding whether it is the appropriate focus for a particular query. LIUYAO-BENCH complements language- and culture-oriented evaluation by organizing these targets into separate verification layers. Its value lies not only in covering another classical-text tradition, but also in making performance differences between local rule knowledge and case-level verification visible.

2.2 EXPERT-DOMAIN AND RULE-GUIDED EVALUATION

Expert-domain benchmarks motivate evaluation that reflects how knowledge is used within a field. CMB (Wang et al., 2024) and QualBench (Hong et al., 2025) assess medical and professional expertise, respectively. More directly related to our design, LAiW (Dai et al., 2025) organizes legal evaluation around three stages of legal syllogism: information retrieval, legal-principle inference, and advanced legal application. RuleArena (Zhou et al., 2025) evaluates the application of natural-language rules in airline baggage fees, NBA transactions, and taxation. These studies establish that domain knowledge, rule selection, and rule application deserve distinct attention rather than a single measure of expertise.

LIUYAO-BENCH brings this perspective to source-based verification in classical Chinese expert texts. Between basic rule statements and full-case judgments, its intermediate layer targets domain-

[1] Data and code: `https://anonymous.4open.science/r/LIUYAO-Bench-data-55A3/`.

specific variables such as *yongshen*, strength, and dynamic line relations. Unlike an open-ended case-answering task, all three layers ask models to verify supplied candidates against source-derived labels. This shared binary format keeps the required output consistent while varying the scope of the judgment, supporting a task-level comparison of where model performance is strong or weak.

2.3 Fine-Grained Verification and Diagnostic Evaluation

Fine-grained evaluation can reveal errors that final-answer scores leave unresolved. PRM800K (Lightman et al., 2023) provides step-level supervision for mathematical solutions, while ProcessBench (Zheng et al., 2025) tests whether models can identify the first erroneous step in a solution. P-FOLIO (Han et al., 2024) likewise provides a resource for evaluating logical reasoning with proof steps. These approaches examine explicit reasoning sequences; LIUYAO-BENCH instead evaluates supplied statements about rules, intermediate states, and case judgments, without requiring a generated reasoning trace.

Studies of reasoning faithfulness ask whether models' explanations describe how they actually reached their answers (Lyu et al., 2023; Turpin et al., 2023). LIUYAO-BENCH asks whether models can check given statements against domain rules and case information. We evaluate their decisions on rule statements, intermediate states, and full-case judgments using labels derived from the source texts. Correct decisions alone do not show how a model reached them.

3 Benchmark

3.1 Task Definition

LIUYAO-BENCH is organized into three complementary binary verification layers covering domain knowledge, intermediate states, and complete cases. Each item contains a candidate statement and a binary label: 1 if the statement agrees with the source-derived interpretation, and 0 otherwise. Case-based items also include the question, calendrical background, and available hexagram and moving-line fields. The model sees these fields together with the candidate, but not the reference label. Appendix C provides a glossary of the domain terms used throughout this paper.

A hexagram consists of six lines. Moving lines change the main hexagram into a changed hexagram; a transformed line occupies the corresponding position after the change. A case can also specify that there are no moving lines. These fields identify the configuration and the changes whose interpretation is being tested. Line and calendrical attributes use the five phases: wood, fire, earth, metal, and water. Generation and overcoming describe support and restraint among these phases, such as wood generating fire and wood overcoming earth. Month and day branches provide conditions for interpreting a line. The question guides *yongshen* selection: the child line for a question about children, or the wife-wealth line for wealth.

Basic Knowledge. The first layer evaluates explicit *liuyao* knowledge. It contains two task types: concept/relation verification and rule-statement verification. Concept/relation items concern terms and symbolic associations, such as which earthly branches form a particular combination. Rule statements concern when a convention applies or which variable should be used for a query. Both are standalone statements, without a separate case context. For example, a positive Concept-Relation item states that Shen, Zi, and Chen form a three-branch combination.

Intermediate-State Interpretation. The second layer evaluates intermediate states used in *liuyao* judgment. *yongshen* identifies the line or six-relative category that serves as the focus of a query. Strength assesses whether a relevant line is strong or weak under the month and day conditions. Dynamics instead checks how a moving or transformed line interacts with other lines, including generation (support), overcoming (restraint), clashes, and combinations. Both subtasks may use the same case fields, but assessing a line's condition and checking its relation to another line require different judgments. Each instance pairs a serialized case context with a candidate intermediate-state statement, and the model predicts whether the candidate is correct. For example, in a case about an annual assessment, the candidate states that the Parents line should be selected as *yongshen*. The model checks whether this choice fits the question. A Strength candidate may state that a line is weak because it clashes with the month's branch. A Dynamics candidate may state that a moving

line receives support from the line it changes into. In each example, the model judges whether the supplied statement is correct for the given case.

In the Strength example in Appendix A.1, the candidate describes Hai-water as weak: Hai clashes with the Si month, and Wei-earth of the day overcomes water. Both conditions support the assessment. The answer concerns the line's condition, not the eventual outcome of the illness in the query.

Full-Case Judgment Verification. The third layer evaluates complete case-level verification. A model is given a structured case record and a candidate final judgment, and must determine whether the proposed judgment is correct. These judgments depend on multiple interacting variables and conventions. Full-Case candidates may state an outcome, a timing judgment, or an action recorded in the source case. For example, one released item asks whether a condemned relative can be spared and supplies a candidate that recommends acting quickly and states that the relative was later spared. A final judgment can combine several claims. One example in Appendix A.1 combines meeting a person with failing to obtain money. The model evaluates the supplied judgment as a whole, rather than checking a single line relation.

The layers share a binary format but target different judgments, not successive steps in a recorded reasoning trace. Appendix A.5 compares items with shared source records; Appendix A.1 provides representative task examples.

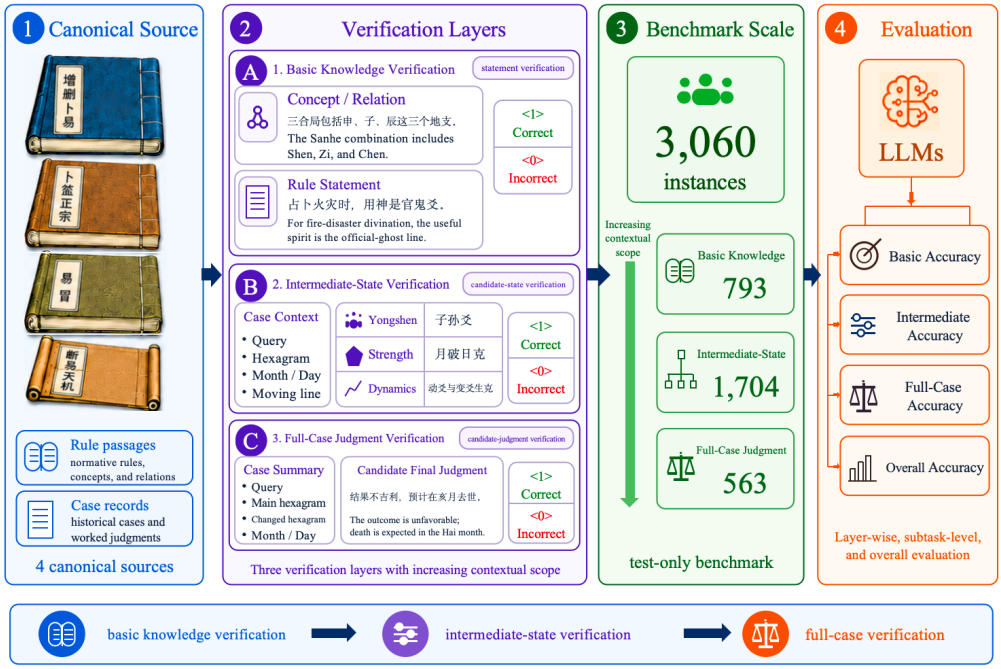


Figure 1: Overview of LIUYAO-BENCH. The benchmark draws on four classical *liuyao* books and has three binary verification layers: basic knowledge, intermediate-state interpretation, and full-case judgment verification.

3.2 BENCHMARK GENERATION

Source Collection and Preparation. We construct LIUYAO-BENCH using texts from four books: *Zeng Shan Bu Yi*, *Bu Shi Zheng Zong*, *Yi Mao*, and *Duan Yi Tian Ji*. The collection includes explanations of domain rules, symbolic relations, and worked cases. The case material is stored as 632 standardized JSONL records containing approximately 400,652 Chinese characters. These materials support both standalone and case-based questions.

Instance Construction. We extract domain statements and case information from the source materials, then organize them into the three verification layers. Basic-knowledge items contain a standalone statement. Intermediate-state items pair a serialized case with a candidate state judgment.

Full-case items pair structured case fields with a candidate final judgment. Each item receives a binary label. Statements that agree with the source passage or case form positive candidates.

For case-based items, serialization separates the case description from the judgment being tested. The same types of context fields precede an intermediate-state or final-judgment marker, followed by the candidate. This distinction matters when constructing negatives: changing the candidate asks whether a different interpretation fits the same case; changing the case as well would create a different verification problem. The case provides the evidence; the candidate states the claim to be checked. The label refers to consistency with the textual convention or recorded interpretation, not to whether a divinatory prediction is true in the real world.

Released Examples. Dynamics item 541 describes a query about an official position, with *Gui Mei* as the main hexagram and the first line moving to produce *Jie*. Its candidate states that Si-fire changes into Yin-wood and receives support from the transformed line. This is labeled positive: wood generates fire, so the stated relation is returning generation (回头生). By contrast, item 301 describes a different case and claims that a wood combination supports the relevant Wei-earth line. That candidate is labeled negative because wood overcomes earth. These examples show what Dynamics verifies: whether a stated relation fits the given configuration. They are separate released items, not a constructed positive/negative pair.

Negative Candidate Construction. We create negative candidates by changing the variable or judgment being tested. For basic knowledge, we alter a concept, relation, rule condition, or outcome. For *yongshen*, we substitute a plausible six-relative category. For Strength and Dynamics, we alter statements about month/day status or moving-line relations. For Full-Case, we change the proposed outcome. We aim to keep negative candidates fluent and roughly matched to positives in length and style. We avoid explicit label cues such as "not", "incorrect", or "wrong". For case-based items, we keep the background, hexagrams, moving lines, and question unchanged and modify only the candidate judgment. Quality Control checks these candidates.

For example, a negative Dynamics candidate describes Chou-earth changing into Yin-wood as returning generation. Wood restrains earth, so the transformed line actually overcomes the original line. The error is a misidentified relation within an otherwise meaningful statement (Appendix A.1). Another negative example assigns the self line to the other party, confusing the line's role rather than a five-phase relation (Appendix D).

Released Format. Each released item has an index, an input sentence, and a binary reference label. The input sentence contains either a standalone statement or the case fields followed by the candidate. Evaluation passes the input sentence to the model and withholds the reference label. Chinese is the evaluation language; the English glosses in the appendix help readers inspect examples but are not model inputs. Named fields separate the background, hexagrams, moving lines, and question. The candidate follows a task-specific marker: 判断 (judgment), 中间状态 (intermediate state), or 最终判断 (final judgment). Appendix D shows these fields in complete inputs.

Quality Control. We combine automatic checks with author review. Automatic checks cover JSON structure, required fields, binary labels, unique indices, nonempty statements, serialization patterns, label balance, and case-field availability. Items that fail these checks are excluded. The authors check whether the context supports a judgment and whether the assigned label agrees with the source. They also inspect negative candidates for single-variable edits and wording that could reveal the label.

3.3 BENCHMARK STATISTICS

LIUYAO-BENCH contains 3,060 binary verification instances: 793 basic-knowledge items, 1,704 intermediate-state items, and 563 full-case items. Table 1 summarizes item counts, label distributions, and average input lengths by subtask. The intermediate layer is the largest because it covers three variables: *yongshen*, Strength, and Dynamics. All three layers contain both positive and negative candidates. These counts refer to verification items, not distinct source records. A single record can support several items, including questions about different intermediate variables. We retain items that pass the construction and quality-control checks rather than force equal numbers from each book. The distribution therefore depends on the amount of source material and its suitability for each task.

Input length also varies across tasks. Basic-knowledge statements average 17.44 characters across the two subtasks, compared with 66.28 to 134.27 for the intermediate subtasks and 83.67 for Full-Case.

Table 1: Benchmark statistics by layer and subtask. Items may share source records. Average input length counts Unicode characters in the released input, including punctuation and serialized case fields but excluding the prompt template.

Layer	Subtask	Items	Positive	Negative	Avg. input (chars)
Basic	Concept-Relation	508	282	226	16.77
	Rule Statement	285	138	147	18.64
Intermediate	*yongshen*	468	268	200	66.28
	Strength	618	318	300	118.47
	Dynamics	618	318	300	134.27
Full-Case	Judgment Verification	563	313	250	83.67
Overall		3,060	1,637	1,423	81.09

Case input lengths count both the context and the candidate. We report each subtask separately and complement Full-Case accuracy with balanced accuracy, which weights positive and negative items equally. Appendix A.2 provides the source-corpus statistics and per-book item counts.

4 Experimental Setup

4.1 LLM Evaluation

We evaluate ten contemporary LLMs. These include Claude-4.5-Sonnet and Claude-4.7-Opus (Anthropic, 2025; 2026); Gemini-3.1-Pro (Google, 2026b) and Gemini-3.1-Flash (Google, 2026a); GPT-5.5 and GPT-5.6 (OpenAI, 2026; n.d.); Qwen3.6-Max and Qwen3.6-Flash (Qwen Team, 2026; Alibaba Cloud, n.d.); and DeepSeek-R1 and DeepSeek-V4 (DeepSeek-AI, 2025; 2026). All models receive the same binary-verification prompt and are instructed to output only <1> or <0>. Accuracy is computed over every item in each subtask across all 3,060 instances. The runner retries failed requests and unparseable responses, retaining the first parsed answer regardless of correctness in Appendix B.

4.2 Human Evaluation

We evaluate practitioners and models on the same 300 benchmark items, with 25 positive and 25 negative items from each of the six subtasks. Two practitioners independently provide binary judgments using the same inputs as the models and shared annotation instructions. A third practitioner resolves disagreements by selecting one of their responses. All three work without access to benchmark labels or model predictions. We report individual accuracy against the unchanged benchmark labels and inter-practitioner agreement using raw agreement and Cohen's κ. The adjudicated responses form the practitioner-consensus result; we also record cases where both original practitioners agree but differ from the benchmark label. Representative positive and negative instances for every subtask are provided in Appendix D.

4.3 Prompting and Fine-Tuning

Prompting. We compare direct answering and chain-of-thought (CoT) prompting with Qwen3-8B (Yang et al., 2025) on all 3,060 items without fine-tuning. Direct prompting requests only a binary answer, whereas CoT requests an explanation before the answer.

Fine-Tuning. We fine-tune Qwen3-8B using QLoRA (Dettmers et al., 2023) with binary-answer supervision across three random seeds. The in-distribution (IID) test uses unseen items from the training books, and the out-of-distribution (OOD) test uses items from *Bu Shi Zheng Zong*, which is excluded from training and development. We compare the base and fine-tuned models on both tests, reporting the mean and standard deviation. Split details and training settings are provided in Appendix E.

Table 2: Performance (%) on LIUYAO-BENCH. Intermediate Acc. denotes micro-averaged accuracy over *yongshen*, Strength, and Dynamics. Overall is micro-averaged over all benchmark instances.

Model	Basic	Intermediate-State				Full-Case	Overall
		yongshen	Strength	Dynamics	Acc.		
Claude-4.5-Sonnet	68.73	73.93	61.81	55.50	62.85	50.62	62.12
Claude-4.7-Opus	76.17	75.21	66.99	61.00	67.08	53.29	66.90
Gemini-3.1-Pro	85.12	**91.88**	83.66	**74.43**	**82.57**	63.77	**79.77**
Gemini-3.1-Flash	81.21	76.28	73.95	67.96	72.42	63.06	72.97
GPT-5.5	84.36	82.48	**83.82**	71.20	78.87	66.43	78.01
GPT-5.6	86.38	88.68	76.54	65.70	75.94	**70.69**	77.68
Qwen3.6-Max	84.62	89.53	81.07	56.47	74.47	53.11	73.17
Qwen3.6-Flash	83.10	87.82	76.86	55.66	72.18	50.44	71.01
DeepSeek-R1	**87.89**	88.68	78.80	65.70	76.76	59.33	76.44
DeepSeek-V4	77.30	85.26	67.64	66.50	72.07	62.17	71.60

5 RESULTS

5.1 OVERALL PERFORMANCE

Table 2 shows that performance varies across the benchmark hierarchy. Models generally perform best on the basic-knowledge layer and score lower on intermediate-state verification. Full-case scores are also lower than basic-knowledge scores for every model. Gemini-3.1-Pro achieves the highest micro-averaged accuracy at 79.77%, followed by GPT-5.5 at 78.01%, GPT-5.6 at 77.68%, and DeepSeek-R1 at 76.44%. However, the overall leader is not the strongest model on every layer. Gemini-3.1-Pro leads intermediate-state verification at 82.57%, whereas GPT-5.6 has the highest raw full-case accuracy at 70.69%, ahead of GPT-5.5 at 66.43% and Gemini-3.1-Pro at 63.77%. Equal layer weighting instead gives GPT-5.6 the highest mean at 77.67%; Appendix A.4 reports macro scores and uncertainty.

Averaged equally across models, accuracy falls from 81.49% on basic knowledge to 73.52% on intermediate states and 59.29% on Full-Case, a 22.20-percentage-point gap between the first and last layers. DeepSeek-R1, the strongest basic-knowledge model at 87.89%, reaches 76.76% on intermediate states and 59.33% on Full-Case. Qwen3.6-Flash shows the largest basic-to-full-case gap, falling from 83.10% to 50.44%, a difference of 32.66 percentage points. These results show that strong performance on explicit knowledge statements does not guarantee equally strong case verification. Full-case accuracy can also be affected by a model's preference for accepting or rejecting candidates; Section 5.4 examines this through positive, negative, and balanced accuracy.

5.2 SUBTASK-LEVEL VARIATION

The intermediate-state layer is not uniform. Candidate *yongshen* verification is relatively easier, with Gemini-3.1-Pro reaching 91.88%. Most models score lower on Strength and Dynamics than on *yongshen*. Dynamics is the weakest intermediate subtask for most models. These items test statements about line transformations, generation, overcoming, and temporal relations, but the scores alone do not show which operation causes an error. No model dominates every slice: Gemini-3.1-Pro leads overall and on intermediate states, GPT-5.6 leads full-case verification, and DeepSeek-R1 leads basic knowledge.

5.3 PRACTITIONER RESULTS

Table 3 compares practitioners and all ten models on the same balanced 300-item subset described in Section 4.2. Expert A and Expert B achieve 89.00% and 88.33% overall, versus 79.00% for the best model, Gemini-3.1-Pro. On *yongshen*, Gemini-3.1-Pro matches Expert A at 96.00%. On Strength, GPT-5.5 and Qwen3.6-Max tie at 86.00%, between the practitioners' scores. Both practitioners lead on Dynamics at 82.00% and 80.00%, versus GPT-5.5's 74.00% and Full-Case at 78.00% and 74.00%, versus GPT-5.6's 68.00%. The two practitioners give the same answer on 272 of 300 items with

Cohen's $\kappa = 0.813$. Appendix A.3 reports agreement and shared practitioner errors by subtask. Both practitioners lead all ten models overall, and exceed the best model by 6–8 percentage points on Dynamics and 6–10 on Full-Case.

Table 3: Accuracy (%) on the same balanced 300-item subset, with 50 items per subtask. All ten models are shown. Practitioner Consensus uses the shared response when the two practitioners agree and the third practitioner's choice otherwise.

Evaluator	**Concept**	**Rule**	***yongshen***	**Strength**	**Dynamics**	**Full-Case**	**Overall**
Expert A	98.00	96.00	96.00	84.00	82.00	78.00	89.00
Expert B	100.00	92.00	94.00	90.00	80.00	74.00	88.33
Practitioner Consensus	98.00	96.00	96.00	86.00	88.00	78.00	90.33
Claude-4.5-Sonnet	66.00	72.00	82.00	72.00	56.00	50.00	66.33
Claude-4.7-Opus	72.00	76.00	76.00	64.00	54.00	60.00	67.00
Gemini-3.1-Pro	86.00	86.00	96.00	82.00	72.00	52.00	79.00
Gemini-3.1-Flash	82.00	80.00	80.00	78.00	56.00	56.00	72.00
GPT-5.5	76.00	82.00	90.00	86.00	74.00	48.00	76.00
GPT-5.6	80.00	90.00	90.00	76.00	64.00	68.00	78.00
Qwen3.6-Max	84.00	88.00	92.00	86.00	54.00	54.00	76.33
Qwen3.6-Flash	86.00	82.00	90.00	70.00	46.00	42.00	69.33
DeepSeek-R1	86.00	78.00	88.00	84.00	58.00	48.00	73.67
DeepSeek-V4	74.00	80.00	88.00	74.00	64.00	56.00	72.67

5.4 Label-Polarity Analysis

We analyze the ten models on all 3,060 items, separately from the 300-item practitioner study. Full-Case contains 313 positive and 250 negative items, so accepting every candidate scores 55.60%. Balanced accuracy averages accuracy on the two classes and gives this baseline 50.00%.

Full-Case has the largest positive–negative accuracy gap, averaging 42.30 percentage points across models, followed by Dynamics at 40.33 and Strength at 30.20. DeepSeek-R1, for example, scores 88.82% on positive Full-Case items but only 22.40% on negative ones. Its overall Full-Case accuracy of 59.33% falls to 55.61% under balanced accuracy. We therefore report class-specific and balanced scores alongside ordinary accuracy (Appendix A.3). These results show that recognizing correct case judgments does not necessarily mean a model can reject incorrect ones.

5.5 Prompting and Fine-Tuning

Table 4 compares Direct and CoT prompting on all 3,060 benchmark items. CoT improves overall accuracy from 57.25% to 61.80%, with gains in Basic Knowledge, *yongshen*, and Full-Case. Strength and Dynamics show little improvement. Even with CoT, Full-Case accuracy reaches only 53.11%, below the 55.60% always-positive baseline.

Table 4: Qwen3-8B without fine-tuning on all 3,060 benchmark items. Values are accuracy (%). Basic combines both knowledge subtasks. CoT uses the final adaptive-budget predictions; Appendix E.1 details the protocol.

Method	Basic	*yongshen*	Strength	Dynamics	Full-Case	Overall
Direct	65.70	58.97	55.83	56.31	46.54	57.25
CoT	71.88	70.51	55.50	56.47	53.11	61.80

As a supplementary experiment, QLoRA fine-tuning improves overall accuracy on both held-out tests under matched BF16 inference in Table 5. OOD accuracy rises from 55.47% to $72.02 \pm 1.28\%$, but Full-Case reaches only $60.61 \pm 6.39\%$. Appendix E.4 provides the Direct/CoT/SFT comparison and detailed matched Base–SFT results; Appendix E.5 provides the other models comparison on those same items.

Table 5: Qwen3-8B overall accuracy (%) before and after QLoRA fine-tuning, evaluated with matched BF16 inference. SFT reports mean ± sample SD over three seeds. Base uses the paired rerun, not the historical Direct predictions.

Test set	Base	SFT
IID ($N = 204$)	55.88	70.59 ± 3.43
OOD ($N = 274$)	55.47	72.02 ± 1.28

6 CONCLUSION

We introduced LIUYAO-BENCH, a benchmark of 3,060 binary verification items covering basic knowledge, intermediate states, and complete cases in classical Chinese *liuyao* texts. Its shared verification format allows us to compare knowledge of individual rules with judgments that depend on a particular case. Across ten models, strong rule knowledge does not ensure strong case verification. Practitioners perform better on a matched subset, especially on Dynamics and Full-Case. For Qwen3-8B, CoT improves overall performance but brings little benefit to Strength and Dynamics. Qwen3-8B fine-tuning improves overall held-out accuracy, but gains remain uneven across tasks, with weaker results on Full-Case. Together, these findings make the application of rules to case information a concrete target for future work. LIUYAO-BENCH provides tasks for studying this gap, including methods that connect intermediate judgments to final decisions and tests of whether models respond appropriately to changes in case conditions.

LIMITATIONS

LIUYAO-BENCH focuses on source-based verification in four classical *liuyao* books. Its coverage is specific to these texts rather than all interpretive traditions or other classical Chinese expert domains. The shared binary format allows consistent scoring across layers and comparison on the same candidate judgments. This setting evaluates verification of supplied answers; open-ended case analysis and the quality of generated explanations require separate evaluation. Extending the tasks to additional traditions and expert domains would allow comparisons across different systems of rules and textual conventions.

The adaptation experiments use one 8B model, one held-out book, and three random seeds. They examine answer-only fine-tuning within a fixed model setting, with separate tests on unseen items from training sources and on an unseen source book. Larger models, other model families, and additional held-out books would help establish how broadly the observed gains apply. Future comparisons could also examine methods that explicitly connect judgments across the three layers and assess whether domain adaptation preserves general capabilities.

The practitioner comparison uses a balanced 300-item subset and a small group of practitioners. It provides a human reference on the same questions answered by the models, rather than a general measure of human expertise. Broader practitioner participation would allow further study of performance across levels of experience. Binary-answer accuracy measures agreement with the reference judgments but does not establish the reasoning process behind an answer. Evaluating explanations alongside predictions would allow closer examination of how models use the source rules and case information.

ETHICAL CONSIDERATIONS

LIUYAO-BENCH is intended as a textual reasoning resource for evaluating LLM behavior on a culturally grounded expert-text tradition. It does not endorse the predictive validity of divination, and model outputs on this benchmark should not be used for personal, medical, legal, financial, or other real-world decision making. The benchmark evaluates whether models can verify statements against source-derived textual conventions, not whether the underlying practice makes reliable predictions about the world.

The released data are derived from historical and instructional texts rather than personal records. Human involvement comprises internal dataset curation and a separate practitioner evaluation in which two annotators supplied binary task labels and a third adjudicator resolved disagreements. We report only anonymous aggregate results and do not release identifying information. The benchmark contains no private user records, personal identifiers, or API keys. As discussed above, the anonymous supplementary package releases derived verification instances and evaluation code, but not complete source-text transcriptions.

AI USE STATEMENT

We used large language models to assist with translation, language editing, proofreading. The authors take full responsibility for the final manuscript and reported results.

REFERENCES

Alibaba Cloud. Text generation. Alibaba Cloud Model Studio documentation, n.d. URL https://help.aliyun.com/en/model-studio/text-generation-model. Accessed: 2026-09-25.

Anthropic. Introducing Claude Sonnet 4.5, 2025. URL https://www.anthropic.com/news/claude-sonnet-4-5. Accessed: 2026-09-25.

Anthropic. Introducing Claude Opus 4.7, 2026. URL https://www.anthropic.com/news/claude-opus-4-7. Accessed: 2026-09-25.

Jiahuan Cao, Yang Liu, Yongxin Shi, Kai Ding, and Lianwen Jin. WenMind: a comprehensive benchmark for evaluating large language models in Chinese classical literature and language arts. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Andong Chen, Lianzhang Lou, Kehai Chen, Xuefeng Bai, Yang Xiang, Muyun Yang, Tiejun Zhao, and Min Zhang. Benchmarking LLMs for translating classical Chinese poetry: Evaluating adequacy, fluency, and elegance. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33019–33036, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1678. URL https://aclanthology.org/2025.emnlp-main.1678/.

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 Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10738–10766, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.716/.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI. DeepSeek V4 preview release, 2026. URL https://api-docs.deepseek.com/news/news260424/. Accessed: 2026-09-25.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. arXiv preprint arXiv:2305.14314, 2023. URL https://arxiv.org/abs/2305.14314.

Yicheng Fu, Zhemin Huang, Liuxin Yang, Yumeng Lu, and Zhongdongming Dai. CHENGYU-BENCH: Benchmarking large language models for Chinese idiom understanding and use. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2355–2366, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.119. URL https://aclanthology.org/2025.emnlp-main.119/.

Google. Gemini 3.1 Flash-Lite: Built for intelligence at scale, March 2026a. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-flash-lite/. Accessed: 2026-09-25.

Google. Gemini 3.1 Pro: A smarter model for your most complex tasks, 2026b. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/. Accessed: 2026-09-25.

Simeng Han, Aaron Yu, Rui Shen, Zhenting Qi, Martin Riddell, Wenfei Zhou, Yujie Qiao, Yilun Zhao, Semih Yavuz, Ye Liu, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Dragomir Radev, Rex Ying, and Arman Cohan. P-FOLIO: Evaluating and improving logical reasoning with abundant human-written reasoning chains. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16553–16565, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.966. URL https://aclanthology.org/2024.findings-emnlp.966/.

Mengze Hong, Wailing Ng, Chen Jason Zhang, and Di Jiang. QualBench: Benchmarking Chinese LLMs with localized professional qualifications for vertical domain evaluation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5938–5953, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.303. URL https://aclanthology.org/2025.emnlp-main.303/.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*, 2023.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11260–11285, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.671. URL https://aclanthology.org/2024.findings-acl.671/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. arXiv preprint arXiv:2305.20050, 2023. URL https://arxiv.org/abs/2305.20050.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi (eds.), *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 305–329, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.20. URL https://aclanthology.org/2023.ijcnlp-main.20/.

OpenAI. Introducing GPT-5.5, 2026. URL https://openai.com/index/introducing-gpt-5-5/. Accessed: 2026-09-25.

OpenAI. GPT-5.6 Sol. OpenAI API documentation, n.d. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol. Accessed: 2026-09-25.

Qwen Team. Qwen3.6-Max-Preview: Smarter, sharper, still evolving, April 2026. URL https://qwen.ai/blog?id=qwen3.6-max-preview. Accessed: 2026-09-25.

Jiaxing Sun, Weiquan Huang, Jiang Wu, Chenya Gu, Wei Li, Songyang Zhang, Hang Yan, and Conghui He. Benchmarking Chinese commonsense reasoning of LLMs: From Chinese-specifics to reasoning-memorization correlations. arXiv preprint arXiv:2403.14112, 2024. URL https://arxiv.org/abs/2403.14112.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Xidong Wang, Guiming Hardy Chen, Dingjie Song, Zhiyi Zhang, Zhihong Chen, Qingying Xiao, Feng Jiang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. CMB: A comprehensive medical benchmark in Chinese. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6184–6205, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.343. URL https://aclanthology.org/2024.naacl-long.343/.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. CLUE: A Chinese language understanding evaluation benchmark. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4762–4772, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.419. URL https://aclanthology.org/2020.coling-main.419/.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.50. URL https://aclanthology.org/2025.acl-long.50/.

Bo Zhou, Qianglong Chen, Tianyu Wang, Xiaomi Zhong, and Yin Zhang. WYWEB: A NLP evaluation benchmark for classical Chinese. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 3294–3319, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.204. URL https://aclanthology.org/2023.findings-acl.204/.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 550–572, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.27. URL https://aclanthology.org/2025.acl-long.27/.

A Extended Benchmark and Experimental Details

This appendix provides task examples, source statistics, and detailed practitioner and label-polarity results.

A.1 Task Examples

Tables 6–8 illustrate the three verification layers. Each example includes a candidate statement, its reference label, and a brief explanation. Case-based examples also show the relevant case fields. The English glosses accompany the Chinese examples for readers; evaluation uses the Chinese inputs.

Table 6: Basic-knowledge verification examples with reference labels and explanations.

Task	Example	Gold	Source-Grounded Reason
Concept Relation	三合会局包括申、子、辰这三个地支。 *The Sanhe combination includes Shen, Zi, and Chen.*	True	申、子、辰构成三合水局，因此该判断正确。 *They form the water combination in the Sanhe system.*
Rule Statement	占卜火灾时，用神是官鬼爻。 *For a fire-disaster query, the official-ghost line is the useful spirit.*	True	火灾属灾祸凶险之事，通常以官鬼爻为判断核心。 *Adverse-event queries typically use the official-ghost line.*

Table 7: Representative binary-verification examples from the intermediate-state layer.

Task	Case	Candidate State	Gold	Source-Grounded Reason
yongshen	酉月己酉日；雷泽归妹 → 地水师；问孩子病情。 *You month, Jiyou day; Guimei → Shi; child's illness.*	此卦应取子孙爻为用神。 *The child line should be selected as the useful spirit.*	True	占问孩子相关事项，通常以子孙爻为用神。 *Child-related queries normally use the child line.*
Strength	巳月乙未日；泽风大过 → 火风鼎；问自身疾病。 *Si month, Yiwei day; Daguo → Ding; one's illness.*	亥水月破又日克，用神衰弱。 *Hai-water is month-broken and day-overcome, so the useful spirit is weak.*	True	亥水冲巳月，又受未土克制，月日均不利。 *Hai clashes with Si and is overcome by Wei-earth.*
Dynamics	卯月乙未日；风火家人 → 风天小畜；问卖货。 *Mao month, Yiwei day; Jiaren → Xiaoxu; selling goods.*	丑土化寅木属于回头生，用神得到生扶。 *Chou-earth transforming into Yin-wood is returning generation.*	False	寅木克丑土，应为回头克而非回头生。 *Yin-wood overcomes Chou-earth, so this is returning overcoming.*

Table 8: Representative binary-verification examples from the full-case layer.

Task	Case	Candidate Final Judgment	Gold	Source-Grounded Reason
Full-Case	午月戊辰日；火地晋；无动爻；问妹妹临产吉凶。 *Wu month, Wuchen day; Jin; no moving line; childbirth query.*	大吉。次日卯时生产，母子平安。 *Highly auspicious; delivery will occur at Mao hour the next day, with mother and child safe.*	True	候选判断与原案最终断语及应验结果一致。 *The candidate matches the source case and recorded outcome.*
Full-Case	酉月辛亥日；兑为泽 → 雷水解；初爻、五爻动；问拜访贵人与求财。 *You month, Xinhai day; Dui → Jie; visiting a noble person and seeking wealth.*	甲寅日能见到贵人，但求财空手而归。 *The querier will meet the noble person on Jiayin day but return empty-handed.*	False	该候选最终判断与原案断语不一致。 *The candidate does not match the source-supported final judgment.*

A.2 Source and Benchmark Statistics

Table 9 summarizes the normalized case records and their character counts. Table 10 gives the book-level distribution of the 563 Full-Case items, whose record links identify a single source book. Benchmark-wide task sizes, label distributions, and input lengths appear in Table 1.

Table 9: Normalized case records by source book.

Source Book	Case Records	Characters
Zeng Shan Bu Yi	338	208,821
Bu Shi Zheng Zong	125	85,907
Yi Mao	105	57,409
Duan Yi Tian Ji	64	48,515
Total	632	400,652

Table 10: Full-Case verification items by source book, counted from the item-to-record links.

Source Book	Full-Case Items
Zeng Shan Bu Yi	333
Bu Shi Zheng Zong	125
Yi Mao	105
Duan Yi Tian Ji	0
Total	563

A.3 Practitioner and Polarity Details

Practitioner agreement. The two practitioners agree on 272 of the 300 sampled items (90.67%, $\kappa = 0.813$). Table 11 breaks down agreement by subtask. It is highest for Concept-Relation (98%) and Rule Statement (96%), and lower for Dynamics (86%) and Full-Case (80%). These results show that the practitioners agree on most items, with more disagreements on dynamic relations and complete cases.

A third practitioner selects between the two responses on the 28 disagreement items without seeing benchmark labels or model predictions. The source-derived benchmark labels serve as the ground truth for scoring. Outcome labels follow the events recorded in the source cases; knowledge and intermediate-state labels follow the rules and interpretations in the source texts. On 20 items, both original practitioners give the same answer but disagree with the benchmark label. These responses count as shared practitioner errors under this source-based evaluation. Practitioner consensus does not change the benchmark labels.

Table 11: Inter-practitioner agreement on the balanced 300-item subset. Raw agreement is reported as a percentage. Each subtask has 50 items. Disagree counts different responses from the two practitioners; Shared errors counts items on which both give the same incorrect answer against the source-derived benchmark label.

Subtask	**Raw Agree.**	κ	Disagree	Shared errors
Concept-Relation	98.00	0.960	1	0
Rule Statement	96.00	0.920	2	2
yongshen	90.00	0.801	5	0
Strength	94.00	0.878	3	5
Dynamics	86.00	0.717	7	6
Full-Case	80.00	0.594	10	7
Overall	90.67	0.813	28	20

Detailed polarity analysis. This analysis uses all 3,060 benchmark items. Positive and negative accuracy are calculated separately on candidates labeled correct and incorrect; balanced accuracy is their arithmetic mean. For each model and subtask, the polarity gap is the absolute difference between these two accuracies.

Across the ten models, the average gap in percentage points is 10.26 for Concept-Relation, 11.49 for Rule Statement, 8.82 for *yongshen*, 30.20 for Strength, 40.33 for Dynamics, and 42.30 for Full-Case (Table 13). Thus, the largest differences between accepting correct candidates and rejecting incorrect ones occur in Strength, Dynamics, and Full-Case.

Full-Case contains 313 positive and 250 negative items. An always-positive predictor therefore achieves 55.60% accuracy but 50% balanced accuracy. Table 12 reports both measures alongside positive and negative accuracy for each model.

DeepSeek, Gemini, and GPT models score higher on positive Full-Case items, while Claude and Qwen models score higher on negative items. The gap ranges from 2.68 percentage points for GPT-5.6 to 66.83 for Qwen3.6-Flash. Balancing the two labels also changes some rankings: Claude-4.7-Opus scores below DeepSeek-R1 in raw accuracy (53.29% versus 59.33%) but above it in balanced accuracy (56.82% versus 55.61%). Reporting both label-specific scores makes these different error patterns visible.

Table 12: Label-wise accuracy on Full-Case Judgment Verification. The split contains 313 positive and 250 negative examples; balanced accuracy averages positive and negative accuracy.

Model	**Raw**	**Bal.**	**Pos.**	**Neg.**
Always-positive baseline	55.60	50.00	100.00	0.00
Always-negative baseline	44.40	50.00	0.00	100.00
Claude-4.5-Sonnet	50.62	53.94	24.28	83.60
Claude-4.7-Opus	53.29	56.82	25.24	**88.40**
Gemini-3.1-Pro	63.77	61.74	79.87	43.60
Gemini-3.1-Flash	63.06	62.02	71.25	52.80
GPT-5.5	66.43	64.45	82.11	46.80
GPT-5.6	**70.69**	**70.54**	71.88	69.20
Qwen3.6-Max	53.11	54.81	39.62	70.00
Qwen3.6-Flash	50.44	54.18	20.77	87.60
DeepSeek-R1	59.33	55.61	**88.82**	22.40
DeepSeek-V4	62.17	59.69	81.79	37.60

Table 13: Absolute difference, in percentage points, between positive and negative accuracy. Larger values indicate stronger label-polarity asymmetry.

Model	**Concept**	**Rule**	***yongshen***	**Strength**	**Dynamics**	**Full-Case**
Claude-4.5-Sonnet	20.69	20.13	2.50	28.87	32.71	59.32
Claude-4.7-Opus	29.27	27.65	4.87	41.48	52.47	63.16
Gemini-3.1-Pro	0.95	0.93	2.83	25.93	33.49	36.27
Gemini-3.1-Flash	5.99	13.66	29.31	7.87	2.52	18.45
GPT-5.5	9.80	11.62	7.82	15.25	13.87	35.31
GPT-5.6	6.71	0.12	8.43	45.60	59.54	2.68
Qwen3.6-Max	7.20	8.64	7.81	31.61	56.09	30.38
Qwen3.6-Flash	8.88	18.78	7.30	32.01	60.90	66.83
DeepSeek-R1	2.19	5.50	9.04	30.06	49.29	66.42
DeepSeek-V4	10.94	7.87	8.28	43.35	42.44	44.19
Mean	10.26	11.49	8.82	30.20	40.33	42.30

A.4 Layer Weighting and Statistical Uncertainty

The overall score in Table 2 is micro-averaged: each item has equal weight. To check whether the ranking depends on task sizes, we also average the three layer accuracies with equal weights. Basic Knowledge combines its two subtasks, and Intermediate-State combines its three subtasks before this average. Macro-Bal. first averages positive and negative accuracy within each layer, then averages the three layers. It gives equal weight to layers and to the two labels within each layer, but not to all six subtasks.

Table 14: API model scores under three aggregation rules (%). Brackets give 95% source-link group-bootstrap intervals. All columns use the same saved predictions on 3,060 items.

Model	Micro	Three-layer macro	Macro-Bal.
Gemini-3.1-Pro	79.77 [78.25, 81.22]	77.15 [75.39, 78.87]	76.69 [75.01, 78.30]
GPT-5.5	78.01 [76.44, 79.45]	76.56 [74.80, 78.19]	75.88 [74.22, 77.47]
GPT-5.6	77.68 [75.92, 79.22]	77.67 [75.86, 79.36]	77.98 [76.25, 79.62]
DeepSeek-R1	76.44 [74.79, 78.05]	74.66 [72.82, 76.42]	73.06 [71.55, 74.49]
Qwen3.6-Max	73.17 [71.55, 74.76]	70.73 [69.02, 72.45]	71.71 [70.13, 73.34]
Gemini-3.1-Flash	72.97 [71.40, 74.58]	72.23 [70.43, 73.99]	71.74 [69.94, 73.55]
DeepSeek-V4	71.60 [69.98, 73.25]	70.51 [68.66, 72.32]	69.43 [67.73, 71.12]
Qwen3.6-Flash	71.01 [69.30, 72.64]	68.58 [66.76, 70.28]	70.30 [68.79, 71.77]
Claude-4.7-Opus	66.90 [65.00, 68.64]	65.51 [63.46, 67.39]	67.32 [65.60, 68.93]
Claude-4.5-Sonnet	62.12 [60.35, 63.76]	60.73 [58.83, 62.56]	62.26 [60.62, 63.93]

Gemini-3.1-Pro leads the micro average, whereas GPT-5.6 has the highest three-layer macro and Macro-Bal. scores. Thus, the overall ranking depends partly on the weight assigned to each layer. This does not invalidate the item-weighted score; the alternatives answer a different question about performance across layers of unequal size.

For uncertainty estimates, we connect items that share any recovered parent-record identifier and use the resulting 1,063 connected components as resampling units. Multiple possible parent links join components conservatively; 17 items without recovered links remain singleton units. We sample 1,063 components with replacement in each of 2,000 replicates (seed 20260925), retaining all items in a sampled component and recomputing each metric. The same draws are used for every model. The intervals reflect variation across these recovered groups, not repeated API calls, uncertain labels, or undetected links between cases.

Table 15: Layer-wise API accuracy and 95% group-bootstrap intervals (%). Intermediate-State pools all three intermediate subtasks.

Model	Basic Knowledge	Intermediate-State	Full-Case
Gemini-3.1-Pro	85.12 [82.43, 87.60]	82.57 [80.60, 84.29]	63.77 [59.82, 67.82]
GPT-5.5	84.36 [81.81, 86.72]	78.87 [76.76, 80.84]	66.43 [62.43, 70.29]
GPT-5.6	86.38 [83.83, 88.81]	75.94 [73.57, 78.23]	70.69 [66.61, 74.41]
DeepSeek-R1	87.89 [85.44, 90.04]	76.76 [74.69, 78.77]	59.33 [54.97, 63.49]
Qwen3.6-Max	84.62 [81.88, 87.24]	74.47 [72.22, 76.60]	53.11 [49.08, 57.02]
Gemini-3.1-Flash	81.21 [78.43, 83.91]	72.42 [70.23, 74.63]	63.06 [59.14, 66.98]
DeepSeek-V4	77.30 [74.29, 80.24]	72.07 [69.95, 74.30]	62.17 [58.24, 66.24]
Qwen3.6-Flash	83.10 [80.14, 85.83]	72.18 [69.85, 74.27]	50.44 [46.22, 54.37]
Claude-4.7-Opus	76.17 [72.87, 79.35]	67.08 [64.54, 69.43]	53.29 [49.07, 57.14]
Claude-4.5-Sonnet	68.73 [65.34, 71.97]	62.85 [60.56, 65.10]	50.62 [46.44, 54.51]

Paired differences further qualify the ranking. Gemini-3.1-Pro's micro advantage over GPT-5.6 is 2.09 percentage points, with a 95% interval of [0.53, 3.73]. GPT-5.6's macro advantage over Gemini-3.1-Pro is only 0.52 points, with an interval of $[-1.47, 2.41]$. Its Full-Case advantage over GPT-5.5 is 4.26 points, with an interval of $[-0.76, 8.96]$. The latter two intervals include zero, so their point-estimate rankings should not be read as clear separation. These are descriptive, unadjusted comparisons selected after inspecting the scores, not an all-pairs significance analysis.

A.5 Items Linked to the Same Source Case

We use recovered parent-record identifiers to compare different tasks on shared cases. Restricting to items with exactly one recovered parent identifier gives 552 records with Strength, Dynamics, and Full-Case items. Of these, 444 also have a *yongshen* item. Each of the 444 records contributes exactly one item from each of these four tasks, allowing a matched comparison without choosing among multiple candidates. Basic Knowledge is not included in these groups.

Table 16: Saved API predictions on 444 source-linked groups. Task columns are accuracy (%); All four and I-only are counts of groups. I-only means that all three intermediate items are correct but the Full-Case item is incorrect.

Model	*yongshen*	Strength	Dynamics	Full-Case	All four	I-only
Gemini-3.1-Pro	91.44	82.66	73.87	64.19	166	86
GPT-5.5	81.53	83.33	72.75	66.89	157	72
GPT-5.6	88.06	75.45	68.47	71.85	178	61
DeepSeek-R1	88.51	78.60	65.54	59.01	125	86
Qwen3.6-Max	89.19	80.63	54.95	52.48	93	99
Gemini-3.1-Flash	75.90	72.97	68.92	62.61	114	64
DeepSeek-V4	84.91	66.67	63.74	62.16	114	62
Qwen3.6-Flash	87.61	76.35	54.28	50.68	87	91
Claude-4.7-Opus	74.32	66.67	60.14	53.83	93	66
Claude-4.5-Sonnet	72.97	61.04	55.86	49.77	45	67

On these matched cases, Gemini-3.1-Pro answers all three intermediate items correctly but misses the Full-Case item in 86 groups; GPT-5.6 does so in 61 groups. This shows that the difference between intermediate and final verification is not only a consequence of evaluating different source cases. However, sharing a source record does not establish that the three intermediate statements are the necessary steps for the supplied final judgment. These groups have not been independently checked as complete reasoning chains, and the scores do not identify a causal point of reasoning failure.

B API Evaluation Prompt

The ten API models in Table 2 use the binary-judgment prompt below. Qwen3-8B prompting and fine-tuning settings are described separately in Appendix E. For every subtask, the model is asked to

determine whether a given LIUYAO-BENCH statement is correct and must output only <1> or <0>. The prompt structure is identical across subtasks; the only difference is the task descriptor shown after the Task: field. This keeps the prompt format consistent across benchmark layers.

Table 17: Task descriptors used in the unified binary-judgment prompt.

Subtask	Task Descriptor
Basic / Concept-Relation	knowledge-level judgment
Basic / Rule Statement	knowledge-level judgment
Intermediate / *yongshen*	intermediate-state interpretation: yongshen selection
Intermediate / Strength	intermediate-state interpretation: strength judgment
Intermediate / Dynamics	intermediate-state interpretation: dynamic-state judgment
Full-Case Judgment	full-case judgment

The unified prompt template is shown below.

Unified binary-judgment prompt

```
Please judge whether the following LIUYAO-Bench statement is correct.

Task:
{task_desc}

Instructions:
- If the statement is correct, output only <1>
- If the statement is incorrect, output only <0>
- Do not provide any explanation
- Do not output anything other than <1> or <0>

Statement:
{sentence}
```

Here, {task_desc} is replaced by one of the descriptors in Table 17, and {sentence} is the serialized benchmark instance. Basic-knowledge items use standalone domain statements, intermediate-state items combine case context with a candidate state judgment, and full-case items combine structured case information with a candidate final judgment.

For output parsing, we first strip surrounding whitespace. The parser then searches for a wrapped binary label using <([01])>; if this fails, it falls back to a standalone binary digit using \b([01])\b. Thus, <1>, <0>, 1, and 0 are accepted, although the prompt explicitly requests <1> or <0>. The runner allows up to five attempts per incomplete item in one invocation and can resume unfinished items. Retries address request or parsing failures, not incorrect answers. The first parsed answer is saved. No unparseable output is removed from the denominator:

$$\text{Accuracy} = \frac{\#\{i : \hat{y}_i = y_i\}}{N},$$

where $\hat{y}_i$ is the parsed model prediction, y_i is the gold label, and N is the full size of the evaluated task. Overall accuracy is micro-averaged over all 3,060 benchmark instances. The finalized result files contain a parsed prediction for each of the 30,600 model–instance pairs.

This final coverage does not measure the initial parsing-failure rate. Each API model contributes one finalized prediction set; the bootstrap analysis in Appendix A.4 resamples those predictions rather than repeated model runs. The current runner defaults to a 64-token output limit, except for DeepSeek reasoner and GPT-5.6 paths, which default to 512. It supports overrides. The ordinary request path sets temperature to zero; the GPT-5.6 path uses max_completion_tokens and omits temperature. These describe the available implementation, not a complete historical configuration log for every evaluated endpoint.

C TERMINOLOGY AND ROMANIZATION

This appendix explains the domain concepts behind the intermediate subtasks and the English terms used in the examples. Definitions follow the interpretive conventions of the source texts.

Hexagrams and lines. A hexagram consists of six lines. Case records specify a main hexagram, any moving lines (动爻), and the resulting changed hexagram. A transformed line (变爻) is the line at the corresponding position after a moving line changes. The relationship between the moving line and its transformed line is one subject of Dynamics verification.

***Yongshen*: the focus of a query.** *yongshen* (用神) is the line or six-relative category selected as the main object of judgment. For example, a query about children may use the child line (子孙爻), while a query about wealth may use the wife-wealth line (妻财爻). The *yongshen* subtask asks whether the proposed selection fits the query. We retain the pinyin term in the paper. The example glosses also use "useful spirit" or "target line" for the same role.

Generation and overcoming. These terms refer to 生 and 克, two relations among the five phases (五行): wood, fire, earth, metal, and water. Generation describes a relation of support, such as wood generating fire; overcoming describes a relation of restraint, such as wood overcoming earth. In the case texts, these relations apply to the phase attributes assigned to lines and calendrical branches. Here, "generation" is a domain term, not text generation.

Strength: assessing relative strength or weakness. Strength concerns 旺衰, whether a relevant line is strong or weak under the month and day conditions. A candidate may state that the line receives support, is restrained, or is weakened by a clash. In the example in Appendix A.1, Hai-water is described as weak because it clashes with the Si month and is restrained by Wei-earth on the day.

Dynamics: changes and interactions among lines. Dynamics concerns moving and transformed lines, including generation, overcoming, combinations, clashes, and advancing or retreating transformations. "Returning generation" (回头生) means that the transformed line supports the original moving line; "returning overcoming" (回头克) means that it restrains the original line. For example, Si-fire changing into Yin-wood is returning generation because wood generates fire. A dynamic relation can also change a line's strength or support a timing judgment. Strength and Dynamics items are therefore distinguished by the main claim being checked: the line's condition or its changing relations. Appendix D provides examples of both.

Names and calendrical expressions. We use pinyin for hexagram names because literal translations vary. For example, 雷泽归妹 is *Guimei*, 火天大有 is *Dayou*, 泽火革 is *Ge*, and 天火同人 is *Tongren*. We write *Guimei* → *Shi* for 雷泽归妹 → 地水师. Heavenly Stems and Earthly Branches are also romanized: 酉月己酉日 becomes "You month, Jiyou day," and 旬空:寅卯 becomes "void: Yin/Mao." These expressions retain the calendrical conditions stated in the case.

Recurring terms. Table 18 lists recurring terms and their roles in the case descriptions. The six-relative categories describe a line's role in a query; for example, the parent line can represent documents as well as parents.

Table 18: Recurring *liuyao* terms, their English glosses, and their roles in the benchmark. Strength and Dynamics are subtask names used in this paper.

Chinese	English term	Meaning in the case description
用神	*yongshen* / target line	Main line or role selected for the query; also glossed as "useful spirit."
旺衰	Strength	Relative strength or weakness under the case conditions.
动变关系	Dynamics	Relations involving line movement, transformation, and interaction.
动爻	moving line	A line marked as changing in the main hexagram.
变爻	transformed line	The corresponding line after the change.
生	generation	A supporting five-phase relation, such as wood generating fire.
克	overcoming	A restraining five-phase relation, such as wood overcoming earth.
合	combination	A prescribed combining relation among branches.
冲	clash	A prescribed opposing relation between branches.
世爻	self line	Line representing the querier.
应爻	response line	Counterpart to the self line.
官鬼爻	official-ghost line	A six-relative category, used in queries such as official rank.
妻财爻	wife-wealth line	A six-relative category, used in queries such as wealth.
父母爻	parent line	A six-relative category, used in queries about parents or documents.
子孙爻	child line	A six-relative category, used in queries such as children.
三合	three-combination (*sanhe*)	A combination involving three specified branches.
六冲	six-clash (*liuchong*)	The six prescribed opposing branch pairs, such as Zi/Wu.
回头生	returning generation	The transformed line generates the original moving line.
回头克	returning overcoming	The transformed line overcomes the original moving line.
化进神	advancing transformation	A change classified as advancing under the branch-transformation rules.
化退神	retreating transformation	A change classified as retreating; also glossed as "transforming into a retreating spirit."

Scope of the glosses. Chinese statements are the benchmark inputs. English glosses preserve their case fields and candidate claims for readers.

D ADDITIONAL DATASET EXAMPLES

Table 19 shows one positive and one negative item from each of the six subtasks, including the serialized case fields used as model input. It complements the shorter, explained examples in Appendix A.1. English glosses follow Appendix C.

Table 19: Additional examples from each LIUYAO-BENCH subtask. The examples are sampled from the released benchmark data and cover both positive and negative labels.

Layer / Subtask	Example Statement	Gold
Basic / Concept-Relation	青龙象征吉祥。 *The Azure Dragon symbolizes auspiciousness.*	True
Basic / Concept-Relation	世爻代表对方。 *The self line represents the other party.*	False
Basic / Rule Statement	占卜下雨以父母爻为用神。 *When divining rain, the parent line is used as the useful spirit.*	True
Basic / Rule Statement	占问功名时，用神是妻财爻。 *When asking about official rank or examination success, the wife-wealth line is the useful spirit.*	False
Intermediate / *yongshen*	背景：寅月 庚戌日 旬空:寅卯；本卦：火天大有；动爻：无动爻；变卦：无变卦；问题：问求财。判断：此卦应取妻财爻为用神。 *Yin month, Gengxu day, void: Yin/Mao; main hexagram: Dayou; no moving line; no changed hexagram; query: seeking wealth. Judgment: the wife-wealth line should be selected as the useful spirit.*	True
Intermediate / *yongshen*	背景：午月 辛巳日 旬空:午未；本卦：兑为泽；动爻：二爻动；变卦：泽雷随；问题：问财运。判断：此卦应取官鬼爻为用神。 *Wu month, Xinsi day, void: Wu/Wei; main hexagram: Dui; second line moves; changed hexagram: Sui; query: financial fortune. Judgment: the official-ghost line should be selected as the useful spirit.*	False
Intermediate / Strength	背景：丑月 甲申日 旬空:午未；本卦：天火同人；动爻：五爻动；变卦：离为火；问题：问财运。中间状态：月建丑土生金，日辰申金代表财星，日建本身为财，非常旺相有力。 *Chou month, Jiashen day, void: Wu/Wei; main hexagram: Tongren; fifth line moves; changed hexagram: Li; query: financial fortune. Intermediate state: Chou-earth of the month generates metal, and Shen-metal of the day represents the wealth star, making it strong and prosperous.*	True
Intermediate / Strength	背景：申月 壬子日 旬空:寅卯；本卦：天山遁；动爻：无动爻；变卦：无变卦；问题：占卜孩子的病情。中间状态：月令申金克制日辰子水，导致日辰衰弱。 *Shen month, Renzi day, void: Yin/Mao; main hexagram: Dun; no moving line; no changed hexagram; query: a child's illness. Intermediate state: Shen-metal of the month overcomes Zi-water of the day, causing the day branch to become weak.*	False
Intermediate / Dynamics	背景：未月 己巳日 旬空:戌亥；本卦：雷泽归妹；动爻：初爻动；变卦：雷水解；问题：问地方官职的运势。中间状态：用神巳火发动，变出妻财寅木，寅木生巳火，这是化回头生，动爻因此得到生助，力量增强。 *Wei month, Jisi day, void: Xu/Hai; main hexagram: Guimei; first line moves; changed hexagram: Jie; query: local official career fortune. Intermediate state: the useful spirit Si-fire moves and transforms into Yin-wood, which generates Si-fire; this is returning generation, strengthening the moving line.*	True
Intermediate / Dynamics	背景：亥月 庚子日 旬空:辰巳；本卦：水雷屯；动爻：二爻动；变卦：水泽节；问题：问有没有孩子。中间状态：用神发动，化为子孙丁卯木，这是化退神，表示子嗣缘分减弱。 *Hai month, Gengzi day, void: Chen/Si; main hexagram: Tun; second line moves; changed hexagram: Jie; query: whether one will have children. Intermediate state: the useful spirit moves and transforms into the child line Ding-Mao wood; this is transforming into a retreating spirit, indicating weakened offspring fortune.*	False
Full-Case Judgment	背景：午月 甲戌日 旬空:申酉；本卦：泽火革；动爻：上爻动；变卦：天火同人；问题：问官职升迁。最终判断：会得到很大的晋升。 *Wu month, Jiaxu day, void: Shen/You; main hexagram: Ge; upper line moves; changed hexagram: Tongren; query: official promotion. Final judgment: the querier will receive a major promotion.*	True
Full-Case Judgment	背景：酉月 戊午日 旬空:子丑；本卦：泽火革；动爻：无动爻；变卦：无变卦；问题：问求财。最终判断：求财可得。 *You month, Wuwu day, void: Zi/Chou; main hexagram: Ge; no moving line; no changed hexagram; query: seeking wealth. Final judgment: the sought wealth can be obtained.*	False

E Additional Open-Weight Experiments

We first evaluate Qwen3-8B without fine-tuning on all 3,060 items. We then train and evaluate it on source-grouped partitions of the benchmark. The following sections describe prompting, data splits, fine-tuning, and comparisons on common held-out items.

E.1 Prompting Protocol

The full-benchmark experiments use Qwen3-8B with BF16 weights, greedy Hugging Face generation, left padding, and native thinking disabled. Direct prompting allows 64 new tokens. Prompted CoT asks for an explanation ending with `FINAL: <1>` or `FINAL: <0>` and initially allows 768 new tokens. The parser reads the last explicit final label, then the last wrapped binary label, or a bare binary digit if it is the entire response.

Direct prompting returns valid labels for all 3,060 items. CoT returns valid labels for 3,003 items at the initial budget. The remaining 57 items receive one retry with the same prompt and a 2,048-token limit; all then return valid labels. This adaptive-budget result is used in the main text. Fixed-budget results are retained below, with unparseable outputs counted as incorrect. All scores use the full evaluated item count as the denominator.

Prompting results. Direct prompting achieves 57.25% overall accuracy. CoT achieves 60.52% at the initial budget and 61.80% after the retries, which add 39 correct answers. Table 4 reports the final result by layer and subtask.

On Full-Case, balanced accuracy is 51.55% for Direct, 51.67% for fixed-budget CoT, and 52.39% for final CoT. Thus, the overall improvement from CoT is accompanied by only a small gain on complete cases.

E.2 Data Splits and Overlap Checks

We group items before creating the SFT splits so that questions derived from the same source record stay together. Parent records are recovered from source identifiers, construction scripts, and generation logs. Similarity checks then identify additional links between groups.

Of the 793 knowledge items, 763 have one recovered parent, 13 have several possible parents, and 17 have no recovered parent. Of the 468 *yongshen* items, 446 have one recovered parent and 22 are ambiguous. All 618 Strength, 618 Dynamics, and 563 Full-Case items have one recovered parent. Items with a shared parent are grouped across tasks. For ambiguous items, we link all possible parents but exclude the items themselves from the SFT splits. Exact duplicates are counted separately in Table 20.

SFT uses only the benchmark items assigned to the training partition below. Earlier training corpora are excluded because they overlap the benchmark's source records.

We also group exact text matches and near-duplicates. Lexical matching uses character-trigram Jaccard similarity with a threshold of 0.85. Semantic matching uses normalized CLS embeddings from BGE-small-zh-v1.5 and a cosine threshold of 0.95. We use no query instruction and split long records into chunks rather than truncate them. Exact matching applies to normalized source or item text.

The semantic check flags 3,567 pairs. We apply an automatic filter using four case fields: background/date, original hexagram, moving lines, and transformed hexagram. A pair is removed only if every possible pairing of its parent records disagrees on at least two fields. This removes 3,172 pairs and retains 395 as grouping links, without using labels or model scores.

We reserve *Bu Shi Zheng Zong* for the source-held-out (OOD) test. Any connected group spanning this book and another source is excluded from the SFT splits. Other groups are assigned to training, development, and IID testing in an approximate 80/10/10 ratio using a stable hash and seed 20260919. Table 20 accounts for all 3,060 items. The 685 excluded items remain in the original benchmark. Every task has both labels in each active split.

All six pairwise comparisons among the four active splits contain no shared recovered parents, grouping blocks, exact item duplicates, or near-duplicates under these checks. This separation is defined by the recovered links and stated similarity thresholds; model pretraining exposure is not assessed.

Table 20: Data splits for SFT. Groups link items with shared parents or detected overlap. The four quarantine categories do not overlap and contain 685 items excluded from the SFT splits, but retained in the benchmark.

Partition	Items	Active groups
Training	1,713	574
Development	184	69
IID test	204	74
OOD source test	274	91
Quarantine: cross-source groups	633	–
Quarantine: ambiguous lineage	34	–
Quarantine: exact duplicate	1	–
Quarantine: unresolved lineage	17	–
Total	3,060	–

The source registry maps the internal key `ylby` to *Zeng Shan Bu Yi*. Knowledge items are linked to their original question-answer records; individual phrases used in negative candidates are not traced separately. All splits retain the released benchmark labels.

E.3 Standard SFT Training and Paired Evaluation

Training uses answer-only supervision: the target is the wrapped binary label followed by EOS, with prompt tokens masked from the loss. QLoRA uses NF4 double quantization, BF16 computation, rank 16, scaling parameter 32, and dropout 0.05 on the query, key, value, output, gate, up, and down projections. We use gradient checkpointing, microbatch size 1, gradient accumulation over 16 examples, and AdamW with weight decay 0.01, peak learning rate 2×10^{-4}, 3% warmup, and cosine decay. The maximum sequence length is 1,536; observed training and development maxima are 257 and 268 tokens, so no examples are truncated.

Each of the three seeds (20260919–20260921) completes three epochs and 324 optimizer steps. The first run trains on an NVIDIA L40 in 65.4 minutes; the other two train on an A100 in 51.1 and 52.1 minutes. The checkpoint with the lowest development loss is selected for each seed, giving epoch 2 in all three runs.

All adapters are evaluated on an A100 with Python 3.12.13, PyTorch 2.10.0+cu128, Transformers 5.0.0, PEFT 0.18.1, and bitsandbytes 0.49.1. NF4 inference uses double quantization, BF16 computation, float32 non-quantized modules, and batch size 4. BF16 inference loads the original BF16 base weights with the same QLoRA-trained adapter, whose parameters remain float32, and batch size 1. Each condition pairs the base and adapted model under identical prompts and decoding settings. The two conditions differ in both precision and batch size.

Evaluation uses greedy decoding, disabled native thinking, a 64-token budget, and a strict parser accepting a complete wrapped binary label with optional surrounding whitespace. All predictions parse successfully. Repeated BF16 base runs produce identical predictions and are reported as one reference.

The adaptation tables use the A100 evaluation runs throughout. Earlier L40 evaluation outputs are archived separately, with model, adapter, data, code, and prediction hashes.

Paired 95% confidence intervals use 2,000 bootstrap samples of the source-grouping blocks with seed 20260919. Base and adapted predictions share the sampled blocks, and accuracy is item-weighted within each resample. These intervals describe test-sample uncertainty; sample standard deviations across the three seeds describe training variation.

E.4 Detailed Adaptation Results

Table 21 collects Direct, CoT, and SFT scores on the same 204 IID and 274 OOD items. Direct and CoT are drawn from the full-benchmark prompting runs; SFT uses the BF16 adaptation evaluation. Table 22 separately compares base and adapted models under matched inference settings.

Table 21: Qwen3-8B accuracy on common test items (%). CoT uses the final adaptive-budget predictions; SFT is the BF16 three-seed mean ± sample SD. The prompting and adaptation runs use their respective inference settings.

Method	IID ($N = 204$)	OOD ($N = 274$)
Direct	56.86	55.47
CoT	65.20	51.46
SFT	70.59 ± 3.43	72.02 ± 1.28

Table 22: Matched base and SFT evaluation under NF4 and BF16 inference (%). Each condition uses its own base reference; SFT reports the three-seed mean ± sample SD. Bal. denotes balanced accuracy.

Precision	Split	Slice	N	Base	SFT	Base Bal.	SFT Bal.
NF4	IID	Overall	204	50.98	71.41 ± 4.55	54.90	71.94 ± 3.90
NF4	IID	Full-Case	31	41.94	72.04 ± 4.93	52.63	72.59 ± 3.80
NF4	OOD	Overall	274	52.55	72.26 ± 1.59	50.23	71.79 ± 1.56
NF4	OOD	Full-Case	55	38.18	59.39 ± 2.78	50.00	62.00 ± 4.25
BF16	IID	Overall	204	55.88	70.59 ± 3.43	58.47	71.20 ± 2.24
BF16	IID	Full-Case	31	41.94	70.97 ± 5.59	52.63	71.20 ± 2.79
BF16	OOD	Overall	274	55.47	72.02 ± 1.28	54.03	71.45 ± 1.77
BF16	OOD	Full-Case	55	40.00	60.61 ± 6.39	51.47	63.28 ± 6.45

Aggregate adaptation results. SFT improves overall accuracy under both inference conditions (Table 22). With NF4, mean accuracy rises from 50.98% to 71.41% on IID and from 52.55% to 72.26% on OOD. With BF16, it rises from 55.88% to 70.59% and from 55.47% to 72.02%, respectively. Every seed improves on its paired base in both tests, and all overall paired confidence intervals exclude zero (Table 25).

The OOD gains are smaller on Full-Case. Mean accuracy reaches 59.39% with NF4 and 60.61% with BF16, compared with 72.26% and 72.02% overall. Their Full-Case balanced accuracies are 62.00% and 63.28%. Thus, the improvement across the full test set does not extend equally to complete case judgments.

Tables 23 and 24 report accuracy by task and seed, along with three-seed means and standard deviations. Table 25 reports the overall paired gains.

Table 23: NF4 inference: per-seed SFT accuracy and aggregate accuracy/balanced accuracy (%). Seed columns abbreviate 20260919–20260921. SD reflects training seeds, not test-sample uncertainty.

Split	Task	N	Base	S19	S20	S21	Mean ± SD	Bal. ± SD
IID	Basic	71	64.79	67.61	59.15	69.01	65.26 ± 5.33	65.20 ± 5.13
IID	Yongshen	28	32.14	85.71	67.86	89.29	80.95 ± 11.48	83.04 ± 8.15
IID	Strength	37	54.05	81.08	83.78	72.97	79.28 ± 5.63	79.63 ± 5.49
IID	Dynamics	37	43.24	75.68	59.46	67.57	67.57 ± 8.11	67.93 ± 8.11
IID	Full-Case	31	41.94	70.97	67.74	77.42	72.04 ± 4.93	72.59 ± 3.80
IID	Overall	204	50.98	74.51	66.18	73.53	71.41 ± 4.55	71.94 ± 3.90
OOD	Basic	60	60.00	85.00	70.00	80.00	78.33 ± 7.64	78.36 ± 7.46
OOD	Yongshen	49	48.98	73.47	85.71	91.84	83.67 ± 9.35	83.69 ± 9.67
OOD	Strength	55	63.64	74.55	80.00	72.73	75.76 ± 3.78	69.88 ± 4.95
OOD	Dynamics	55	50.91	61.82	61.82	70.91	64.85 ± 5.25	63.21 ± 3.48
OOD	Full-Case	55	38.18	61.82	60.00	56.36	59.39 ± 2.78	62.00 ± 4.25
OOD	Overall	274	52.55	71.53	71.17	74.09	72.26 ± 1.59	71.79 ± 1.56

Table 24: BF16 inference: per-seed SFT accuracy and aggregate accuracy/balanced accuracy (%). Seed columns abbreviate 20260919–20260921. SD reflects training seeds, not test-sample uncertainty.

Split	Task	N	Base	S19	S20	S21	Mean ± SD	Bal. ± SD
IID	Basic	71	67.61	70.42	63.38	67.61	67.14 ± 3.54	67.16 ± 3.20
IID	Yongshen	28	42.86	85.71	67.86	78.57	77.38 ± 8.99	80.41 ± 3.96
IID	Strength	37	59.46	78.38	78.38	67.57	74.77 ± 6.24	75.15 ± 6.08
IID	Dynamics	37	51.35	70.27	70.27	62.16	67.57 ± 4.68	67.88 ± 4.61
IID	Full-Case	31	41.94	74.19	64.52	74.19	70.97 ± 5.59	71.20 ± 2.79
IID	Overall	204	55.88	74.51	68.14	69.12	70.59 ± 3.43	71.20 ± 2.24
OOD	Basic	60	70.00	85.00	71.67	80.00	78.89 ± 6.74	78.79 ± 6.48
OOD	Yongshen	49	55.10	75.51	83.67	87.76	82.31 ± 6.23	82.36 ± 6.53
OOD	Strength	55	58.18	67.27	78.18	69.09	71.52 ± 5.84	64.99 ± 6.31
OOD	Dynamics	55	52.73	70.91	67.27	63.64	67.27 ± 3.64	62.57 ± 5.69
OOD	Full-Case	55	40.00	67.27	60.00	54.55	60.61 ± 6.39	63.28 ± 6.45
OOD	Overall	274	55.47	73.36	71.90	70.80	72.02 ± 1.28	71.45 ± 1.77

Table 25: Overall SFT gains and paired 95% group-bootstrap intervals (percentage points). Each seed is paired against the same deterministic base within its inference condition. Intervals use 2,000 resamples and do not capture training-seed uncertainty.

Precision	Split	Seed	Gain	95% interval
NF4	IID	20260919	23.53	[14.29, 32.34]
NF4	IID	20260920	15.20	[6.70, 23.30]
NF4	IID	20260921	22.55	[14.65, 30.10]
NF4	OOD	20260919	18.98	[11.60, 26.79]
NF4	OOD	20260920	18.61	[12.95, 24.31]
NF4	OOD	20260921	21.53	[15.94, 27.14]
BF16	IID	20260919	18.63	[9.71, 27.40]
BF16	IID	20260920	12.25	[4.29, 20.28]
BF16	IID	20260921	13.24	[3.92, 22.16]
BF16	OOD	20260919	17.88	[11.43, 24.18]
BF16	OOD	20260920	16.42	[10.49, 22.34]
BF16	OOD	20260921	15.33	[9.19, 21.48]

For interpreting raw scores, always predicting positive on the 55 OOD Full-Case items (34 positive, 21 negative) yields 61.82% accuracy and 50% balanced accuracy. Always predicting negative on OOD Dynamics yields 65.45% raw accuracy. These references explain why the tables also report balanced accuracy, which gives both labels equal weight.

E.5 Comparisons on Common Test Items

Table 26 restricts the existing API and prompting predictions to the same 204 IID and 274 OOD items, matched by task, index, and reference label. This permits comparison on common questions while retaining each experiment's original prompting, parsing, and inference settings.

Table 26: API and Qwen3-8B prompting accuracy on common test items (%). Scores use the same 204 IID and 274 OOD items, with each run's original inference settings.

Model / condition	IID Acc.	OOD Acc.
Claude-4.5-Sonnet	67.16	58.03
Claude-4.7-Opus	70.59	64.60
DeepSeek-R1	78.92	73.36
DeepSeek-V4	73.04	69.34
GPT-5.5	80.39	77.37
GPT-5.6	83.33	80.66
Gemini-3.1-Flash	76.96	69.71
Gemini-3.1-Pro	84.31	78.47
Qwen3.6-Flash	75.98	75.91
Qwen3.6-Max	76.47	75.18
Qwen3-8B Direct	56.86	55.47
Qwen3-8B CoT (final)	65.20	51.46
Qwen3-8B CoT (fixed budget)	63.73	51.09

E.6 Practitioner Comparison on Held-Out Items

Of the 300 practitioner-rated items, 54 are in the adaptation test sets: 24 IID and 30 OOD. The others fall in training (162), development (15), or excluded groups (69). Table 27 compares all three adapters with the original practitioner responses on these 54 held-out items, using the unchanged benchmark labels.

Table 27: Practitioner-overlap comparison restricted to adaptation-held-out items (24 IID, 30 OOD). Unlike the original 300-item study, this small subset is not task- or label-balanced and has only five Full-Case items. Scores are descriptive accuracy (%); SFT reports three-seed mean $\pm$ sample SD.

Condition	IID ($N = 24$)	OOD ($N = 30$)
Expert-A	87.50	90.00
Expert-B	91.67	83.33
Expert-consensus	91.67	90.00
NF4 base	62.50	50.00
NF4 SFT	77.78 ± 2.41	70.00 ± 5.77
BF16 base	75.00	53.33
BF16 SFT	79.17 ± 0.00	68.89 ± 1.92

Both practitioners score above the mean SFT result on IID and OOD in this subset. Because the subset is small and unbalanced, the main practitioner comparison remains the balanced 300-item evaluation of non-adapted models in Section 4.2.