MMFCTUB: Multi-Modal Financial Credit Table Understanding Benchmark

Anonymous ACL submission

Abstract

Multimodal large language models (MLLMs) have demonstrated strong capabilities in visual understanding, domain knowledge integration, and numerical reasoning. Yet their performance on credit table understanding (CTU) for financial credit review remains largely unexplored due to limited data availability, high annotation costs, and a lack of task-aligned evaluation frameworks. We introduce MMFCTUB, a benchmark of 6,967 instances spanning 17 table types whose structures, field dependencies, and user financial profiles are strictly aligned with authentic credit reports and real-world distributions. Our construction pipeline enables accurate labeling with minimal expert intervention. We decompose CTU into three orthogonal capability dimensions along with an end-to-end task, define 11 metrics, and quantify complexity for fine-grained assessment. Evaluation of 10 proprietary and open-source MLLMs reveals that CTU remains far from solved: the best model achieves only 57.9% exact match with 50.0% template consistency, and numerical reasoning accuracy drops from near-perfect to below 12% as computation depth increases.

1 Introduction

Tables are the predominant format for presenting structured information across domains, owing to their visual accessibility over sequential text (Wu et al., 2025; Wang et al., 2024; Shigarov, 2023; Deng et al., 2022; Zheng et al., 2024; Zhou et al., 2025; Kang et al., 2025). In financial credit review (Brennecke et al., 2021), loan officers must interpret credit report tables (CRTs) and assess applicants' economic status, a process that demands accurate cross-table content recognition, domain knowledge application, and numerical computation of assessment metrics. Multimodal large language models (MLLMs) have demonstrated strong visual understanding (Yang et al., 2025; Wang et al., 2025b) and achieved promising results across diverse image-based tasks (AlSaad et al., 2024; Ye et al., 2024). Combined with robust domain knowledge integration and numerical reasoning acquired through large-scale pre-training, these capabilities make MLLMs a natural fit for credit table understanding (CTU). Despite these advantages, comprehensive evaluation of MLLMs on CTU has been hindered by four challenges: **1. Data Inaccessibility.** Existing table understanding (TU) benchmarks rely on generic tables from public web corpora, yet real CRTs are inaccessible due to stringent privacy regulations. **2. Specialized Dependencies and Distributions.** CRTs exhibit domain-specific layouts with strong cross-table dependencies and distributions tied to individual economic profiles, characteristics that existing benchmarks fail to capture. **3. High Annotation Cost.** CRTs encode complex financial implications and numerical relations, requiring domain expertise and multi-step reasoning for annotation, which incurs prohibitive costs. **4. Misaligned Evaluation Paradigm.** Current benchmarks process tables as serialized text, whereas practical credit review relies on 2D visual processing, creating a fundamental gap between evaluation and practice.

To address these challenges, we present MMFCTUB , comprising 6,967 instances across 17 table types curated from official credit reports with 246 user profiles whose distributions are aligned to public statistics.

Our four-stage construction pipeline first distills inter-field and cross-table constraints from authentic credit reports via MLLMs, then applies profile-conditioned synthesis with expert verification to produce tables that faithfully preserve real-world dependencies and distributions while enabling cost-effective automated labeling. To support systematic diagnosis, we decompose CTU into three orthogonal capability dimensions: Table Structure Perception (TSP), Domain Knowledge

Dataset	Dom	CrT	CrRP	CrCDep	CrTDep	TaS	RP	AnC	Par	Kno	Comp	Ann
TableBench	Gen	×	×	✓	×	Public	×	High	Text	×	×	19k
FewTUD	Gen	×	×	✓	×	Public	×	Mid	Text	×	×	3k
Entrant	Fin	×	×	✓	×	Public	×	Mid	Text	×	×	331k
FinTab-LLaVA	Fin	×	×	✓	×	Public	×	High	Image	✓	✓	7.3k
MMFCTUB	Fin	✓	13	✓	✓	Pub Str+ Syn Data	✓	Low	Image	✓	✓	6.9k

Table 1: Comparison of datasets for TU. **Dom**: Domain. **CrT**: Cross-Table. **CrRP**: Cross Table Retrieval Pattern, **CrCDep**: Cross Columns Dependencies, **CrTDep**: Cross Tables Dependencies, **TaS**: Table Source. **RP**: Allign with Credit Review Process. **AnC**: Annotation Cost. **Par**: Input Paradigm. **Kno**: Domain Knowledge. **Comp**: Computation. **Anno**: Annotations. **Gen**: General. **Fin**: Financial

Utilization (DKU), and Computational Logic Reasoning (CLR), complemented by an End-to-End (E2E) scenario, yielding 40 fine-grained categories. Each dimension is formalized through graph-based task definitions (Table Retrieval Graph for TSP, Computation Evaluation Graph for CLR) with principled complexity analysis, and assessed by 11 diagnostic metrics that jointly measure answer accuracy and reasoning robustness.

Evaluation of 10 MLLMs reveals that CTU remains far from solved: the best model achieves only 57.9% EM with 50.0% template consistency, exposing fundamental reasoning fragility. Visual perception emerges as the primary TSP bottleneck, domain knowledge exhibits model-specific asymmetries, and multi-step numerical reasoning presents a steep capability cliff.

In summary, our contributions are:

- We propose a four-stage construction pipeline that distills inter-field and cross-table constraints from authentic credit reports via MLLMs, then applies profile-conditioned synthesis with expert verification to produce realistic credit tables while enabling cost-effective automated labeling.
- We develop MMFCTUB, a benchmark of 6,967 instances across 17 table types with a three-dimensional capability taxonomy, hierarchical sub-categories yielding 40 fine-grained evaluation categories, graph-based task formalization (TRG, CEG) with complexity analysis, and 11 diagnostic metrics jointly measuring accuracy and robustness.
- We evaluate 10 proprietary and open-source MLLMs, revealing that the best model achieves only 57.9% exact match with 50.0% template consistency, and identifying critical bottlenecks in visual perception, domain knowledge utilization, and multi-step numerical reasoning.

2 Related Wrok

2.1 Table Understanding

Recent advances in large language models (LLMs) have spurred significant research in table understanding (Wang et al., 2024; Deng et al., 2022; Chen et al., 2024; Cao and Liu, 2025; Liu et al., 2022). TableBench (Wu et al., 2025) evaluates LLM performance across 18 domains and four task types using real-world industrial tables. Sui et al. (Sui et al., 2024) adopt a task decomposition approach, assessing LLM capabilities at each intermediate step (Li et al., 2023, 2025; Zhang et al., 2025; Xu et al., 2025). However, these works primarily focus on general-purpose tables represented in serialized formats such as Markdown, Pandas DataFrames, or HTML.

2.2 Financial Table Understanding

Comparing with understanding general tables for LLM (Lu et al., 2025; Chen et al., 2024; Ren et al., 2025), financial tables are facing with more challenges since they are feature with more complex structures, more domain-specific knowledge (Yang et al., 2024; Guo et al., 2025; Lu et al., 2023) and numerical relationships. FinQA (Chen et al., 2021) extracts tables from financial market reports and provides a dataset for numerical calculation (Su et al., 2024; Loukas et al., 2022; Khang et al., 2024) tasks. (Zhu et al., 2021) TAT-QA addresses the unique requirements of financial tables by defining subtasks including sequence tagging, aggregation operators, and scale prediction to evaluate LLM performance.

2.3 MLLMs for Table Understanding Benchmark

Multimodal large language models (MLLMs) (Yang et al., 2025; Wang et al., 2025b) have been widely applied across domains (Tampubolon, 2025; Chen et al., 2025). Financial tasks require domain

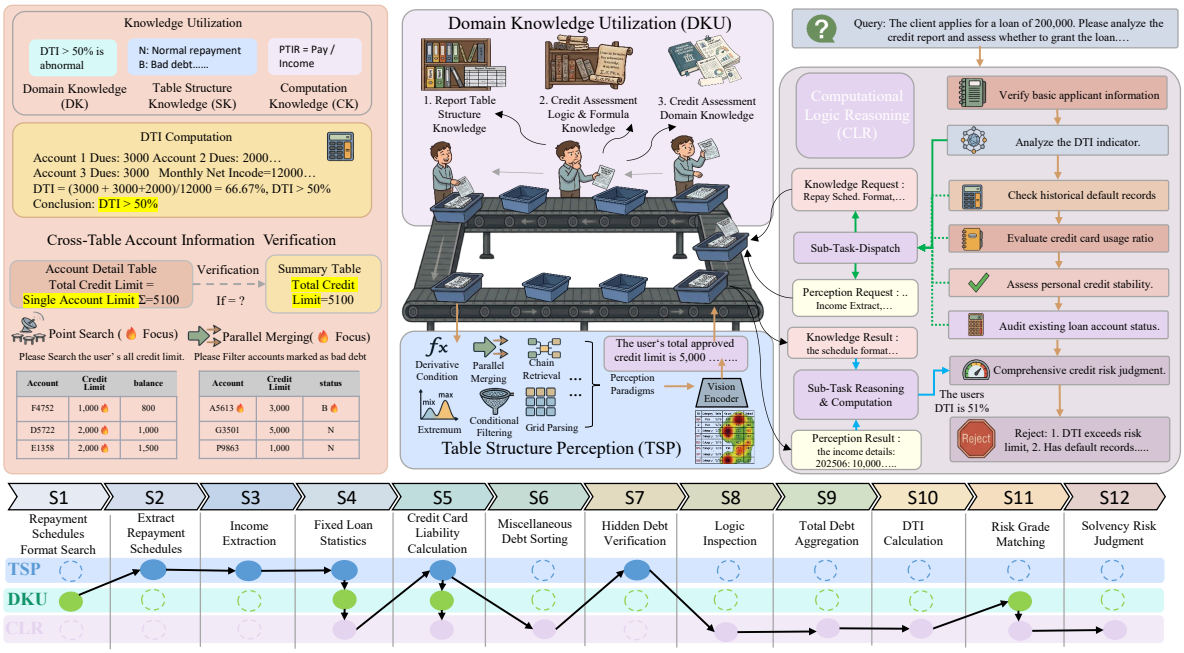

Figure 1: The figure illustrates the complete credit table understanding process. Given a query, the MLLM decomposes it into sub-questions, each requiring table structure perception, domain knowledge retrieval and utilization, and metric computation and reasoning. The 12 numbered steps show the resolution of one sub-question (DTI), where the three capability dimensions interleave along the reasoning flow indicated by arrows.

knowledge, numerical reasoning, and table structure perception. (Zheng et al., 2024) proposes Table-LLaVA, processing table images directly with strong performance across 23 benchmarks. FinTab-LLaVA (Park et al., 2025) introduces a multimodal LLM tuned on FinTMD for financial table QA, fact verification, and description generation. Credit reports encode economic profiles through specialized layouts and numerical relationships critical for loan decisions, yet existing benchmarks inadequately assess MLLM performance on such tables. We present MMFCTUB, a comprehensive benchmark evaluating MLLMs on financial credit report understanding.

3 MMFCTUB

The construction of MMFCTUB proceeds in four stages. We begin with our data generation pipeline, covering question formulation, answer annotation and credit table generation. We then introduce a three-dimensional capability taxonomy that systematically decomposes table understanding into fine-grained tasks, followed by evaluation paradigms that isolate bottlenecks across these capabilities, and conclude with dataset statistics.

3.1 Dataset Construction

3.1.1 Real-World Table Curation.

In practical loan review, credit officers must examine a diverse set of tables throughout the entire credit report to assess an applicant's status. To faithfully reflect real-world workflow, we consult domain experts and curate 17 frequently referenced table types from official credit reports issued by authorized credit bureaus, as summarized in Table 7

3.1.2 Credit Table Generation

Existing table generation approaches suffer from two critical limitations. First, real-world credit tables are inaccessible due to privacy regulations, and current synthetic methods rely on rule-based generation that fails to ensure distributional plausibility, structural consistency, and temporal alignment with real-world conditions. Second, large-scale fine-grained annotation involving domain experts incurs prohibitive costs. To address both limitations, we propose a novel generation methodology. As illustrated in Figure 5, our approach comprises four key components:

1) Data Generation. To preserve the logical and distributional interdependencies present in real-world credit tables, we leverage LLM to generate Detail and Auxiliary entries conditioned

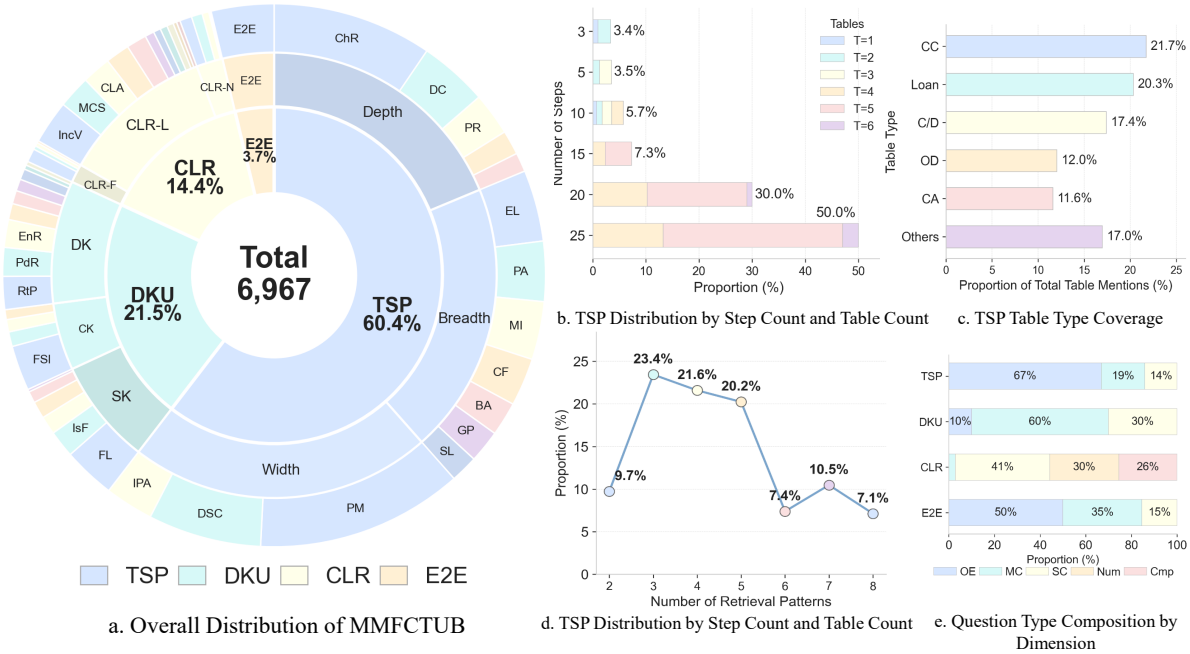

Figure 2: Dataset statistics of MMFCTUB. (a) Instance distribution across four evaluation dimensions. (b) TSP samples by reasoning step count, segmented by number of tables involved. (c) Top-5 table types by mention frequency (CC=Credit Card, Loan=Loan Account, C/D=Credit/Debt Summary). (d) Number of distinct retrieval patterns per TSP sample. (e) Question type composition across dimensions (OE=Open-ended, MC=Multi-choice, SC=Single-choice). Abbreviation mappings are provided in Appendix N.

on Profile attributes to establish preliminary distributional alignment, further calibrated through expert review to match real-world distributions (Figure 3). Summary tables are derived by aggregating across the preceding three categories to enforce cross-table numerical consistency, while time-dependent fields such as interest rates are anchored to their corresponding temporal context. Table 2 details the LLM involvement at each stage.

2) Constraint Distillation. To align generated tables with authentic credit reports, we employ MLLMs to extract meta-information from 17 table types, all verified by domain experts (Appendix L).

3) Question Formulation. To align evaluation questions with real-world credit review needs, we leverage validated table definitions alongside expert-crafted review guidelines to prompt LLMs to generate questions that mirror practical credit assessment demands.

4) Label Preparation. Credit table understanding involves domain-specific financial indicators that require complex multi-column and cross-table aggregations, making manual annotation prohibitively expensive. To address this, we prompt LLMs to generate executable labeling functions based on abstract table schemas. Ground-truth labels are then obtained by executing these functions on the synthesized table instances.

Table 2: LLM involvement in data construction. Only Layer 1 content population uses LLM; questions, answers, and scoring are all deterministic (Table 30).

Component	Example Fields	LLM
User profiles	income, credit rating...	✗
Residential & empl.	employer, phone, job title	✓
Account details	balance, credit limit...	✓
Auxiliary records	litigation, debt recovery...	✓
Repayment history	monthly status codes	✗
Summary tables	total accounts, total balance, overdue counts	✗

3.1.3 Generator bias Analysis

A potential concern with LLM-assisted data generation is whether the generating model biases evaluation results in its favor. To investigate this, we regenerate credit report data for 30 users using three models (Gemini 3.1 Pro, GPT-5, and Claude Sonnet 4.6) and evaluate all three on each variant across TSP, DKU, and CLR tasks using EM as the unified metric (Table 6). We find no evidence of generator bias, see Appendix B.1 for the full analysis.

3.2 Capability Taxonomy

Drawing from the operational workflow of credit reviewers, we decompose CTU into three orthogonal capability dimensions: **Table Structure Perception (TSP)**, **Domain Knowledge Utilization (DKU)**, and **Computational Logic Reasoning (CLR)**. Rather than forming serial stages, these dimensions dynamically interleave throughout the credit review process (Figure 1). We accordingly design dimension-specific tasks that isolate each capability to pinpoint individual bottlenecks, complemented by an **End-to-End (E2E)** evaluation that integrates all three to assess overall MLLM performance.

TSP: Table Structure Perception. TSP evaluates the ability to perceive and retrieve information from visually complex credit table structures. Unlike generic tables, credit reports pose unique visual challenges identified by domain experts, as summarized in Table 8.

We formalize table retrieval as a *Table Retrieval Graph* (TRG) to solve the above challenges. Given a credit report $\mathcal{R} = \{T_1, \ldots, T_n\}$ with n tables, a TRG is a directed acyclic graph $G = (V, E)$ where each node $v \in V$ denotes an atomic retrieval operation from $\mathcal{O} = \{Locate, Extract, Filter, Agg, Compare, Join\}$, and each directed edge $(v_i, v_j) \in E$ indicates that v_i's output serves as input to v_j. The complexity of a TRG instance is characterized by a triplet:

$$\mathcal{C}(G) = \langle D(G),\ B(G),\ W(G) \rangle, \qquad (1)$$

where $D(G)$ denotes the *depth* (longest directed path length in G), $B(G)$ the *breadth* (number of distinct table instances accessed), and $W(G)$ the *width* (maximum number of parallel branches converging via a JOIN node). Based on TRG complexity, we define 13 retrieval patterns (P1–P13) across four progressive difficulty levels (S1–S4), with their definitions and targeted visual challenges detailed in Table 9.

DKU: Domain Knowledge Utilization. (1) Table Structure Knowledge (SK) concerns the domain-grounded interpretation of credit table schemas, including field definitions, SK evaluates whether MLLMs can bridge raw table content and downstream computational reasoning by contextualizing perceived cells within the credit domain. **(2) Calculation Knowledge (CK)** involves domain-specific formulas not explicitly present in the tables, such as debt-to-income ratio (DTI). CK evaluates whether MLLMs can recall relevant formulas from parametric knowledge and correctly instantiate them with table-extracted values. **(3) Domain Knowledge (DK)** encompasses regulatory thresholds, risk assessment rules, and domain-specific terminology that guide credit decisions but are never explicitly stated in the report. DK evaluates whether models have internalized specialized financial concepts and can apply them to evaluation, independent of table perception or computation.

CLR: Computational Logic Reasoning. Credit assessment requires multi-step computational reasoning over heterogeneous financial tables. CLR isolates this capacity from visual perception (TSP) and domain knowledge recall (DKU) by targeting computational logic reasoning in isolation. The reasoning process naturally decomposes into three stages: identifying what to compute, determining how to compute it, and executing the computation at scale, corresponding to three sub-dimensions: **CLR-F** (Formula), **CLR-L** (Logic), and **CLR-N** (Numerical Computation), detailed in Appendix H.

We model each credit assessment calculation as a Computation Evaluation Graph (CEG), which specifies the complete reasoning chain. The CEG is not exposed to the model during evaluation but serves to characterize the computational complexity of each question. A CEG is a DAG $G = (V, E)$ where each node $v \in V$ is typed by $\tau : V \to \mathcal{O}$ with $\mathcal{O} = \{\text{Filter, Reduce, Calc, Cond, Map}\}$ and each directed edge $(u, v) \in E$ encodes a data-flow dependency. The semantics of six node types as shown in Table 3. The complexity of a CEG is characterized by Depth and Width. Let $\mathcal{P}(G)$ denote the set of all directed paths in G. The depth and width of a CEG are

$$D(G) = \max_{p \in \mathcal{P}(G)} |p|, \qquad (2)$$

$$W(G) = \big|\{v \in V : \tau(v) = \text{SOURCE}\}\big|, \qquad (3)$$

where $|p|$ is the number of nodes on path p. Depth measures the longest sequential reasoning chain a model must sustain; width measures the number of parallel data streams it must coordinate.

E2E: End-to-End Credit Assessment. The E2E scenario integrates TSP, DKU, and CLR to simulate the complete credit review workflow. Given a

Operator	Description
Filter($pred$)	Retains elements satisfying a predicate
Map(f)	Applies a per-element transformation
Reduce(agg)	Collapses a set to a scalar via $agg \in \{\text{sum}, \text{avg}, \text{count}, \text{max}, \text{min}\}$
Calc(op)	Performs a scalar arithmetic operation $op \in \{+, -, \times, \div, \text{pow}\}$
Cond($rule$)	Applies a conditional mapping

Table 3: Primitive operators of the CEG.

full credit report as image input and a high-level assessment question, models must visually perceive relevant table content across multiple pages (TSP), identify applicable assessment rules and formulas (DKU), perform multi-step calculations with correct computational logic (CLR), and synthesize findings into a final credit decision, mirroring the real-world process illustrated in Figure 1.

3.3 Metrics

Shared Evaluation Metrics. Exact Match (EM). All dimensions and E2E tasks share a unified scoring framework. For each question q_i, a type-specific scoring function $s(q_i, M) \in [0, 1]$ is defined as follows:

- **Single-choice:** $s{=}\mathbf{1}[\hat{y}_i{=}y_i]$, case-insensitive.
- **Multi-choice:** sorted exact match $s{=}\mathbf{1}[\text{sort}(\hat{A}_i){=}\text{sort}(A_i)]$ in DKU; Jaccard similarity $s{=}|\hat{A}_i \cap A_i|/|\hat{A}_i \cup A_i|$ in CLR.
- **Open-ended numeric:** $s{=}\mathbf{1}[|\hat{y}_i{-}y_i|/|y_i|{\le}\epsilon]$, with $\epsilon{=}0.005$ for TSP and $\epsilon{=}0.01$ for CLR.
- **Open-ended QA:** keyword coverage $s{=}\mathbf{1}[\text{cov}(\hat{y}_i, y_i){\ge}0.5]$.
- **Compound:** $s{=}\frac{1}{K}\sum_{k=1}^{K} s_k$, averaging over K sub-keys.

The overall metric is Exact Match (EM), computed as:

$$\text{EM} = \frac{1}{N}\sum_{i=1}^{N} s(q_i, M) \tag{4}$$

Template Consistency Rate (TCR). TCR measures mastery stability by requiring all variant instances of a template to be answered correctly:

$$\text{TCR} = \frac{|\{t \in \mathcal{T} : \forall\, i \in V_t,\ s(q_i, M) = 1\}|}{|\mathcal{T}|} \tag{5}$$

where $\mathcal{T}$ is the set of parent templates and V_t the variant set for template t. Since TCR $\le$ EM always holds, high EM paired with low TCR signals fragile reasoning. Variants are constructed differently per dimension: in **TSP**, the same labeling function is applied across different users' data; in **DKU**, the same knowledge point is tested under different surface realizations; in **CLR**, identical computational rules are instantiated with different account values.

TSP-Specific Metrics. Step-Weighted Dimension Performance (swDPA). Standard function-level Dimension Performance Analysis (DPA) assigns equal weight to each function regardless of its internal composition, introducing systematic biases. swDPA addresses this by weighting each perception dimension d proportionally to its step-level frequency within the perception steps S_q^P (excluding arithmetic steps):

$$w_d(q) = \frac{|\{s \in S_q^P : d \in D_s\}|}{|S_q^P|} \tag{6}$$

$$\text{swDPA}(d) = \frac{\sum_i c_i \cdot w_d(q_i)}{\sum_i w_d(q_i)} \tag{7}$$

where D_s is the set of perception dimensions annotated for step s and $c_i \in \{0, 1\}$ denotes sample correctness. This formulation corrects three biases inherent in function-level DPA: uneven dimension distribution, phantom penalties from arithmetic steps, and step-count confounding that dominates the dimension signal.

DKU-Specific Metrics. Domain Specificity Stratified Accuracy (EM@L). EM computed per cognitive load level L_k, $k \in \{1, 2, 3, 4\}$, where each of the 19 knowledge subcategories is assigned to exactly one level:

$$\text{EM@}L_k = \frac{\sum_{i:\, L(i)=k} \mathbf{1}[y_i = \hat{y}_i]}{\sum_{i:\, L(i)=k} 1} \tag{8}$$

The four levels represent a progression from factual recall (L_1) through relational discrimination (L_2) and procedural application (L_3) to multi-source integration (L_4); the full mapping from subcategories to levels is provided in Appendix Table 12. This stratification disentangles domain specificity from cognitive complexity as independent difficulty drivers.The remaining metrics are detailed in Appendix C.

3.4 Dataset Statistics

Figure 2 summarizes the dataset distribution. MM-FCTUB comprises **6,967** evaluation instances across 3 dimensions and E2E scenario (Figure 2a), with TSP accounting for the largest share (4,210; 60.4%), further stratified into sub-dimensions that yield 40 fine-grained categories at the leaf level

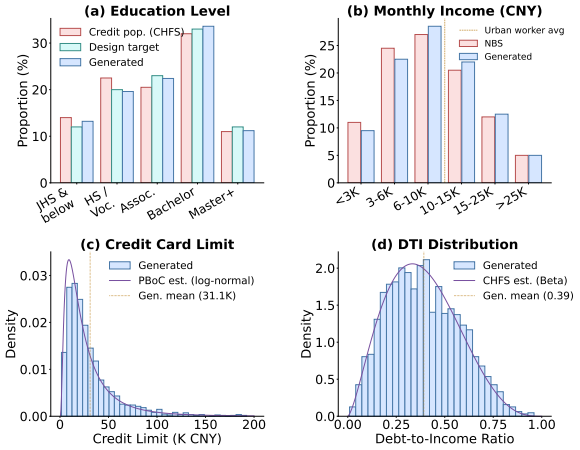

Figure 3: Distribution comparison between generated data and real-world statistics.

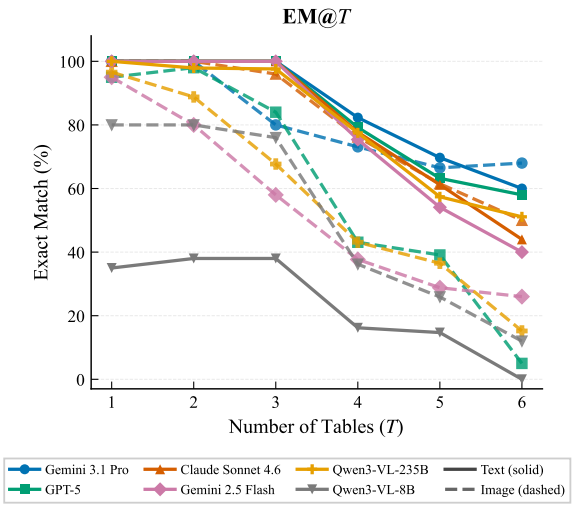

Figure 4: TSP exact match accuracy as a function of reasoning step count $n \in \{3, 5, 10, 15, 20, 25\}$. Solid lines denote text-mode input; dashed lines denote image-mode input.

(full mappings in Appendix N). The underlying data spans 246 user profiles with distributions aligned to public statistical sources (Appendix M). Further details are provided in Appendix A.

4 Experiment

We evaluate both proprietary and open-source MLLMs on MMFCTUB using official implementations with default hyperparameters. Credit tables are rendered as images via LaTeX compilation, preserving the layout and ordering of real credit reports. TSP and E2E provide complete user-level credit table images, while DKU and CLR use plain text input to isolate their respective capabilities from visual perception. For TSP, we additionally provide serialized text as a controlled ablation to disentangle visual perception from reasoning. We select 10 representative models spanning four proprietary systems: Gemini3.1Pro (Comanici et al., 2025), GPT-5 (Wang et al., 2025a), ClaudeSonnet4.6 (Salbas and Buyuktoka, 2025), and Gemini2.5Flash (Comanici et al., 2025) and six open-source models from two families: Qwen3-VL (Yang et al., 2025) (8B, 30B, 235B) and InternVL3.5 (Wang et al., 2025b) (8B, 30B, 241B).

4.1 Main Results

Table 4 reports overall and per-dimension diagnostic metrics; Table 5 further decomposes each dimension into representative sub-level comparisons under both text (T) and image (I) input. We organize our analysis around five key findings.

CTU remains far from solved. The best-performing model, Gemini3.1Pro, achieves only 57.9% micro-averaged EM across all tasks (Table 4). More critically, even this top model attains merely 50.0% TCR. This persistent EM–TCR gap across all models reveals a fundamental fragility: models may solve individual instances yet fail to generalize the underlying logic across applicants, undermining deployment reliability in consistency-critical credit review scenarios.

Visual perception is the primary TSP bottleneck. A controlled text-vs-image ablation (Table 5) shows that text input consistently outperforms image input on TSP EM under identical questions, confirming visual perception is the dominant bottleneck. Performance degrades steeply with structural complexity: all models achieve near-perfect EM@3 ($\geq$96.7%), yet the best EM@25 reaches only 50.0% (Table 4). Multi-table scaling further amplifies this degradation (Figure 4): GPT-5 drops from 95.0% (T_1) to 5.0% (T_6) on image input.

Domain knowledge capabilities exhibit model-specific asymmetries. GPT-5 achieves the highest DKU EM (72.4%). A consistent difficulty gradient emerges across knowledge types (Table5): Structure Knowledge is relatively accessible (SK_1=$81.5% for GPT-5), whereas Domain Knowledge requiring unstated regulatory rules remains substantially harder (DK_9=$49.2% for the best model). Notably, ClaudeSonnet4.6 exhibits the largest format sensitivity (EM@F gap of 37.0 pp), indicating that its DKU shortcomings partly stem from output format compliance rather than knowledge deficiency.

Computational logic exposes a capability cliff. CLR presents the steepest performance degrada-

Table 4: Comprehensive evaluation metrics across TSP, DKU, and CLR tasks. Overall EM is the micro-averaged exact match across all available tasks. TSP uses image input only. ↑: higher is better; ↓: lower is better. Best values per column are **bold**. 1st and 2nd shading marks the top two values per column. Subscript arrows indicate the gap from the best model.

	Model	Overall			TSP					DKU			CLR			E2E
		EM↑	TCR↑	AYR↑	EM@3↑	EM@25↑	Δ_{soft}↓	swDPA↑	EM@ext↑	EM@L1↑	EM@L4↑	EM@F↓	EM@D2↑	EM@D5↑	CFR↓	EM↑
Closed	Gemini 3.1 Pro	**57.9**	**50.0**	98.9	**100.0**	**47.2**	8.6	**67.7**	**67.7**	73.0	**71.0**	14.7	**100.0**	11.1	30.4	**71.3**
Closed	GPT-5	55.2$_{\downarrow 2.7}$	25.0	52.8	96.7	18.6	**0.5**	39.8	46.3	**76.3**	69.4	**10.1**	**100.0**	**16.4**	27.7	68.1
Closed	Claude Sonnet 4.6	53.9$_{\downarrow 4.0}$	43.8	**100.0**	**100.0**	43.6	9.8	66.0	64.9	52.9	66.1	37.0	**100.0**	9.2	**23.0**	64.6
Closed	Gemini 2.5 Flash†	51.7$_{\downarrow 6.2}$	7.4	95.3	93.6	16.9	7.6	33.8	35.1	61.2	66.3	17.5	99.1	12.7	30.5	60.5
Open	Qwen3-VL-8B	46.8$_{\downarrow 11.1}$	4.7	93.3	90.0	14.0	7.9	29.9	34.9	57.8	67.5	17.4	98.3	12.0	30.7	55.3
Open	Qwen3-VL-235B‡	53.5$_{\downarrow 4.4}$	11.6	96.5	95.7	18.1	8.3	35.2	40.3	62.8	68.2	20.3	99.6	13.5	31.4	65.2
Open	Qwen3-VL-30B-A3B§	50.1$_{\downarrow 7.8}$	6.4	94.7	92.5	15.8	7.1	31.6	36.7	60.9	67.8	18.6	98.7	12.5	31.5	58.1
Open	InternVL3.5-8B§	47.3$_{\downarrow 10.6}$	5.1	94.6	91.3	13.6	7.2	31.3	35.3	57.3	68.1	18.3	98.6	11.6	31.2	54.0
Open	InternVL3.5-30B-A3B§	51.3$_{\downarrow 6.6}$	6.9	95.1	93.1	16.5	6.7	32.5	37.1	61.7	68.5	19.2	99.0	13.3	32.6	57.3
Open	InternVL3.5-241B-A28B§	52.7$_{\downarrow 5.2}$	11.1	95.4	95.1	16.3	8.7	36.5	42.4	61.3	67.5	19.7	99.4	14.5	33.5	63.7

Table 5: Expanded sub-category breakdown with L1 task-level EM and two representative L3 sub-levels per dimension. (T) = text input; (I) = image input. ↑: higher is better. Best values per column are **bold**. DKU abbreviations: FL = Field Layout, IsF = Isomorphic Fields. CLR abbreviations: IncV = Income Verification, PIR = Payment-to-Income Ratio. See Appendix N for the full abbreviation glossary. TSP L3: n = reasoning step count, T = table count, P = perception dimension count.

	Model	All	TSP							DKU							CLR						
				Depth		Breadth		Width			SK		CK		DK			CLR-L		CLR-F		CLR-N	
	Model	EM↑	EM↑	n_3↑	n_{25}↑	T_1↑	T_6↑	P_2↑	P_8↑	EM↑	FL↑	IsF↑	FSI↑	Fml↑	RtP↑	RpO↑	EM↑	IncV↑	MCS↑	TAD↑	CUR↑	ILC↑	PIR↑
Closed	Gemini 3.1 Pro (T)	**60.6**	**77.2**	**100.0**	**50.0**	**100.0**	60.0	78.6	**82.0**	71.2	76.9	75.0	76.5	**65.6**	**81.8**	**49.3**	43.3	**56.7**	21.5	33.3	56.4	10.5	9.7
Closed	Gemini 3.1 Pro (I)	57.9	**72.7**	**100.0**	47.2	**100.0**	**68.0**	77.1	76.0														
Closed	GPT-5 (T)	58.9	73.0	**100.0**	45.6	**100.0**	58.0	**80.0**	46.0	**72.4**	**81.5**	**77.7**	**79.7**	57.4	78.1	40.3	44.3	41.1	**35.6**	73.3	61.3	9.2	7.1
Closed	GPT-5 (I)	55.2	49.3	96.7	18.6	95.0	5.0	58.6	40.0														
Closed	Claude Sonnet 4.6 (T)	53.9	70.5	**100.0**	43.6	**100.0**	44.0	67.1	66.0	56.9	60.6	60.7	75.4	24.6	70.8	26.9	**46.4**	36.7	20.0	**86.7**	**66.7**	**28.0**	**32.1**
Closed	Claude Sonnet 4.6 (I)	53.9	70.5	**100.0**	43.6	**100.0**	50.0	70.0	68.0														
Closed	Gemini 2.5 Flash (T)	51.6	39.7	92.3	16.9	83.2	13.8	44.2	22.3	63.1	65.7	70.4	72.7	32.1	73.6	33.5	41.5	40.2	19.7	58.3	42.4	7.2	9.7
Closed	Gemini 2.5 Flash (I)	38.8	38.8	83.3	20.0	95.0	26.0	47.1	28.0														
Open	Qwen3-VL-8B (T)	44.0	18.8	73.3	6.8	35.0	0.0	24.3	30.0	57.1	58.8	63.4	66.8	37.7	67.2	28.4	42.4	35.6	20.7	55.5	26.7	6.3	8.9
Open	Qwen3-VL-8B (I)	46.8	36.7	90.0	14.0	80.0	12.0	41.4	20.0														
Open	Qwen3-VL-235B	53.5	40.1	95.7	18.1	87.5	13.1	45.3	21.9	61.3	69.4	75.0	62.0	27.9	74.5	44.8	43.8	39.3	24.6	59.1	45.8	7.9	12.3
Open	Qwen3-VL-30B-A3B	50.1	38.2	92.5	15.8	83.3	12.5	43.1	20.8	60.1	62.0	66.8	70.4	39.7	70.8	29.9	43.0	37.4	22.7	57.2	36.9	6.8	10.2
Open	InternVL3.5-8B	47.3	37.0	91.3	13.6	80.7	12.1	41.8	20.2	56.5	58.3	62.8	66.2	37.4	66.5	28.1	40.6	33.9	15.7	46.3	25.6	4.7	10.5
Open	InternVL3.5-30B-A3B	51.3	38.7	93.1	16.5	84.3	12.6	43.6	21.1	60.9	62.7	67.6	71.3	40.2	71.7	30.3	42.3	38.1	20.6	55.3	35.9	7.1	12.7
Open	InternVL3.5-241B-A28B	52.7	39.3	95.1	16.3	85.7	12.9	44.3	21.4	60.5	62.4	67.2	70.9	40.0	71.2	30.1	42.9	40.2	24.3	58.2	44.6	8.9	11.5

tion across all dimensions. While EM@D_2 approaches 100% for most models, EM@D_5 collapses to at most 16.4% (GPT-5), a drop exceeding 83 pp (Table 4). Roughly one in four CLR questions triggers complete reasoning breakdown. Localizing this failure (Table 5), CLR-N (Numerical) yields near-zero EM at higher depths ($D_4 \leq 22.9\%$, $D_5 \leq 2.2\%$), while CLR-F (Formula) retains moderate performance, confirming multi-step numerical reasoning with cascading dependencies as the core unsolved challenge.

5 Conclusion

This work introduces MMFCTUB, a novel finance credit table understanding benchmark specifically designed to evaluate the capacities of MLLMs in credit table understanding. MMFCTUB encompass diverse practical credit tables and fine-grained capacities. Extensive evaluations on MMFCTUB allow us to identify significant performance limitations among existing MLLMs in table structure perception, domain knowledge utilization and numerical computation.

6 Limitations

Our DKU evaluation covers representative knowledge categories in credit review but does not exhaust the full spectrum required in practice. Additionally, our current TSP evaluation operates at the question level; a more fine-grained analysis that pinpoints which visual factors degrade perception on specific table regions would yield more actionable insights for model improvement.

References

Rawan AlSaad, Alaa Abd-Alrazaq, Sabri Boughorbel, Arfan Ahmed, Max-Antoine Renault, Rafat Damseh, and Javaid Sheikh. 2024. Multimodal large language models in health care: applications, challenges, and future outlook. *Journal of medical Internet research*, 26:e59505.

Claire Brennecke, Gohar Siravyan, and B Heath Witzen. 2021. Commercial credit on consumer credit reports. *Consumer Financial Protection Bureau Office of Research Reports Series*, (21-7).

Lang Cao and Hanbing Liu. 2025. Tablemaster: A recipe to advance table understanding with language models. *arXiv preprint arXiv:2501.19378*.

Si-An Chen, Lesly Miculicich, Julian Eisenschlos, Zifeng Wang, Zilong Wang, Yanfei Chen, Yasuhisa Fujii, Hsuan-Tien Lin, Chen-Yu Lee, and Tomas Pfister. 2024. Tablerag: Million-token table understanding with language models. *Advances in Neural Information Processing Systems*, 37:74899–74921.

Yanlin Chen, Chenjia Huang, Shumiao Gao, Yifan Lyu, Xinyuan Chen, Shen Liu, Dat Bao, and Chunli Lv. 2025. A multimodal deep learning approach for legal english learning in intelligent educational systems. *Sensors*, 25(11):3397.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1):33–40.

Xin Guo, Haotian Xia, Zhaowei Liu, Hanyang Cao, Zhi Yang, Zhiqiang Liu, Sizhe Wang, Jinyi Niu, Chuqi Wang, Yanhui Wang, and 1 others. 2025. Fineval: A chinese financial domain knowledge evaluation benchmark for large language models. 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)*, pages 6258–6292.

Xiaoqiang Kang, Shengen Wu, Zimu Wang, Yilin Liu, Xiaobo Jin, Kaizhu Huang, Wei Wang, Yutao Yue, Xiaowei Huang, and Qiufeng Wang. 2025. Can grpo boost complex multimodal table understanding? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12642–12655.

Alex Khang, Rashmi Gujrati, Hayri Uygun, RK Tailor, and Sanjaya Gaur. 2024. *Data-driven modelling and predictive analytics in business and finance: concepts, designs, technologies, and applications*. CRC Press.

Xingzuo Li, Kehai Chen, Yunfei Long, Xuefeng Bai, Yong Xu, and Min Zhang. 2025. Generator-assistant stepwise rollback framework for large language model agent. *arXiv preprint arXiv:2503.02519*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333.

Ruixue Liu, Shaozu Yuan, Aijun Dai, Lei Shen, Tiangang Zhu, Meng Chen, and Xiaodong He. 2022. Few-shot table understanding: A benchmark dataset and pre-training baseline. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3741–3752.

Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. 2022. Finer: Financial numeric entity recognition for xbrl tagging. *arXiv preprint arXiv:2203.06482*.

Dakuan Lu, Hengkui Wu, Jiaqing Liang, Yipei Xu, Qianyu He, Yipeng Geng, Mengkun Han, Yingsi Xin, and Yanghua Xiao. 2023. Bbt-fin: Comprehensive construction of chinese financial domain pre-trained language model, corpus and benchmark. *arXiv preprint arXiv:2302.09432*.

Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2025. Large language model for table processing: A survey. *Frontiers of Computer Science*, 19(2):192350.

Hyejin Park, Jiyoon Lee, and Hayoung Oh. 2025. Fintab-llava: Finance domain-specific table understanding multimodal llm using fintmd. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 235–246. Springer.

Yi Ren, Chenglong Yu, Weibin Li, Wei Li, Zixuan Zhu, TianYi Zhang, ChenHao Qin, WenBo Ji, and Jianjun Zhang. 2025. Tablegpt: a novel table understanding method based on table recognition and large language model collaborative enhancement. *Applied Intelligence*, 55(5):311.

Ali Salbas and Rasit Eren Buyuktoka. 2025. Performance of large language models in recognizing brain mri sequences: a comparative analysis of chatgpt-4o, claude 4 opus, and gemini 2.5 pro. *Diagnostics*, 15(15):1919.

Alexey Shigarov. 2023. Table understanding: Problem overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(1):e1482.

Huan-Yi Su, Ke Wu, Yu-Hao Huang, and Wu-Jun Li. 2024. Numllm: Numeric-sensitive large language model for chinese finance. *arXiv preprint arXiv:2405.00566*.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Manotar Tampubolon. 2025. Challenges in using multimodal argumentation in legal code. *International Journal for the Semiotics of Law-Revue internationale de Sémiotique juridique*, pages 1–18.

Shansong Wang, Mingzhe Hu, Qiang Li, Mojtaba Safari, and Xiaofeng Yang. 2025a. Capabilities of gpt-5 on multimodal medical reasoning. *arXiv preprint arXiv:2508.08224*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025b. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and 1 others. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv preprint arXiv:2401.04398*.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, and 1 others. 2025. Tablebench: A comprehensive and complex benchmark for table question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25497–25506.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2025. Llava-cot: Let vision language models reason step-by-step. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2087–2098.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Cehao Yang, Chengjin Xu, and Yiyan Qi. 2024. Financial knowledge large language model. *arXiv preprint arXiv:2407.00365*.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. 2024. mplug-owl2: Revolutionizing multimodal large language model with modality collaboration. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 13040–13051.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. *arXiv preprint arXiv:2406.08100*.

Bangbang Zhou, Zuan Gao, Zixiao Wang, Boqiang Zhang, Yuxin Wang, Zhineng Chen, and Hongtao Xie. 2025. Syntab-llava: Enhancing multimodal table understanding with decoupled synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24796–24806.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624*.

Appendices

A Dataset Statistics 11
B Dataset Construction 11
C Evaluation Metrics 12
D Expert Consultation Process Record 12
E Credit Report Table Type Inventory 13
F TSP: Retrieval Patterns and Visual Challenges 14
G DKU: Knowledge Categories and Field Correspondence 14
H CLR: Reasoning Types and Difficulty Levels 14
I Further Experiments 16
J E2E Tasks Error Propagation Analysis ... 19
K Credit Report Terminology 22
L Inter- and Intra-Table Constraint Relationships 22
M External Data Alignment and Statistical Sources 23
N Abbreviation Glossary 23
O LLM Dependency Audit 27
P Prompts 27

A Dataset Statistics

The TSP dimension is intentionally skewed toward high-complexity samples: 80.0% of instances require $\geq$20 reasoning steps, with the most complex samples involving up to 6 tables (Figure 2b). Nine table types appear in the benchmark, among which credit card, loan, and credit/debt summary together account for 59.4% of all table mentions (Figure 2c). Each TSP sample activates 2–8 distinct retrieval patterns, with the majority (65.3%) engaging 3–5 simultaneously (Figure 2d), reflecting the multi-faceted nature of real credit review. Question formats vary systematically across dimensions (Figure 2e): TSP is predominantly open-ended (66.8%), DKU favors multi-choice (59.9%) to test knowledge discrimination, and CLR combines single-choice, numerical, and compound formats that require structured multi-key responses.

B Dataset Construct

B.1 Generator Bias Analysis

We find no evidence of generator bias. The mean self-advantage is $\Delta = 0.13$ pp (permutation test, $p = 0.41$, Cohen's $d = 0.02$), with GPT-5 achieving slightly lower accuracy on its own generated data ($\Delta = -0.1$ pp). Switching the generator changes EM by at most 0.9 pp, whereas switching the evaluatee shifts EM by up to 4.3 pp—a $4.8\times$ gap confirming that model capability, not data provenance, dominates performance variation. This result is architecturally expected: as shown in Table 2, the generating LLM only populates content fields within code-enforced constraints, while all evaluation-critical components remain fully deterministic.

Table 6: Cross-generator bias analysis. Each cell reports micro-averaged EM (%) across TSP, DKU, and CLR tasks. Self-Δ = diagonal − off-diagonal mean. No significant self-advantage is observed ($\bar{\Delta} = 0.13$ pp, $p = 0.41$).

Evaluatee	**Generator**			**Self-Δ**
	GPT-5	Claude	Gemini	
Gemini 3.1 Pro	67.5	66.8	**67.4**	+0.25
GPT-5	**64.8**	65.3	64.5	−0.10
Claude Sonnet 4.6	63.2	**63.9**	54.1	+0.25
Mean Self-Δ				+0.13

B.2 Construction Pipeline

We introduced the dataset construction process in Figure 5, which involves both advanced LLMs (GPT-5) and MLLMs (Gemini3.1-Pro). To illustrate how these models interact with each pipeline component, we detail the prompts used in each generation stage. Given the overall user economic background distributions (average monthly income, credit score), table infrastructure specifications, and inter-table constraints, we employ GPT-5 to generate detailed loan accounts based on this information. As shown in Figure 12,13,14, credit reports contain three account types: Quasi-Credit Card, General Loan Account, and Credit Card Account. Each account type includes common attributes (e.g., opening date, repayment records) as well as type-specific fields (e.g., overdraft balance for Credit Card Accounts). We therefore design specialized prompts for each account type. The LLM leverages its domain knowledge to understand the user's economic profile while adhering to account generation rules, producing account details that align with the user's background. All account fields are defined based on real-world credit report schemas. Notably, General Loan Accounts are further subdivided into three subtypes: Non-revolving Loan Account, Sub-account under Revolving Credit Limit, and Revolving Loan Account.

Since these subtypes follow different table construction rules, we design subtype-specific prompts and generation procedures to ensure compliance with their respective structural constraints.

C Evaluation Metrics

Answer Yield Rate (AYR). We additionally report the fraction of questions for which the model produces a parseable answer:

$$\mathrm{AYR} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\hat{y}_i \neq \varnothing] \tag{9}$$

Reporting AYR alongside EM disentangles output failures (e.g., truncation, refusal) from reasoning or knowledge deficiencies. We further derive two dimension-specific variants: in **TSP**, depth-stratified $\mathrm{AYR}@n$ localizes output failures to specific complexity levels; in **DKU**, a knowledge upper bound $\mathrm{EM}_{\text{upper}}=\mathrm{EM}+(1-\mathrm{AYR})$ estimates the theoretical maximum assuming all output failures were recovered.

TSP-Specific Metrics. Depth-Stratified Accuracy (EM@n). EM computed over questions grouped by retrieval step count $n \in \{3, 5, 10, 15, 20, 25\}$:

$$\mathrm{EM}@n = \frac{1}{|S_n|}\sum_{i \in S_n} \mathbf{1}[y_i = \hat{y}_i],$$
$$S_n = \{i : n_{q_i} = n\} \tag{10}$$

$\mathrm{EM}@n$ reveals how performance degrades as retrieval complexity increases.

Soft Accuracy Gap (Δ_{soft}). We define $\Delta_{\text{soft}} = \mathrm{SA}@5\% - \mathrm{EM}_{\text{oe}}$, where SA@5% relaxes the numeric tolerance from 0.5% to 5% on open-ended samples. A large Δ_{soft} indicates correct table perception with imprecise downstream computation; $\Delta_{\text{soft}} \approx 0$ signals fundamental perception failure.

Operation-Type Failure Profile (EM@Op). Stratifies EM by each question's dominant perception operation $\mathrm{dom}(q) = \arg\max_{op \neq \text{arith.}} |\{s : \mathrm{op}(s) = op\}|$ among {*extract*, *aggregate*, *conditional*, *sort*}, revealing operation-level bottlenecks orthogonal to depth ($\mathrm{EM}@n$) and perception dimension (swDPA).

DKU-Specific Metrics. Format-Stratified Accuracy (EM@F). EM stratified by question format $F \in \{\mathrm{SC}, \mathrm{MC}, \mathrm{QA}\}$, enabling analysis of whether performance varies systematically across answer formats.

CLR-Specific Metrics. Complete Failure Rate (CFR). Fraction of questions scoring exactly zero, capturing catastrophic reasoning breakdown:

$$\mathrm{CFR}(M) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[s(q_i, M) = 0] \tag{11}$$

For binary-scored types, $\mathrm{CFR} = 1 - \mathrm{EM}$; for partial-credit types (multi-choice, compound), the gap $\mathrm{CFR} - (1 - \mathrm{EM})$ quantifies bimodal all-or-nothing behavior. Depth-stratified $\mathrm{CFR}@D$ and template-level CFR_t pinpoint which credit assessment tasks are most vulnerable to catastrophic failure, a risk-critical diagnostic where a completely wrong indicator is qualitatively worse than a minor numerical error.

Depth-Stratified Accuracy (EM@D). EM computed per CEG (Computation Evaluation Graph) depth $D \in \{2, 3, 4, 5\}$, where D is the longest path in the directed acyclic computation graph of each question's template:

$$\mathrm{EM}@D = \frac{\sum_{i:\, D(q_i)=D} s(q_i, M)}{|\{i : D(q_i) = D\}|} \tag{12}$$

Unlike TSP's $\mathrm{EM}@n$ which varies over $n \in [3, 25]$, CLR's D takes four discrete values reflecting the categorical structure of credit assessment tasks: D=3 aggregates 6 templates and D=4 aggregates 3 templates, ensuring robustness within each stratum.

D Expert Consultation Process Record

The design and validation of our research framework benefited significantly from consultations with senior professionals at one of the largest financial institutions, which serves over 2 billion users globally. Through structured interviews with industry specialists, we gathered critical insights into consumer credit evaluation. These expert perspectives directly shaped the development of this benchmark. Our consultation panel comprised seasoned specialists across credit underwriting, risk control, and lending operations, each with 8 to over 15 years of hands-on experience. Each expert brings unique domain knowledge spanning different aspects of the credit evaluation lifecycle, from initial application review to final credit decision-making.

Key findings from our expert consultations highlighted several critical aspects:

Expert 1 (14 years, Senior Credit Underwriter) Specializes in consumer credit assessment

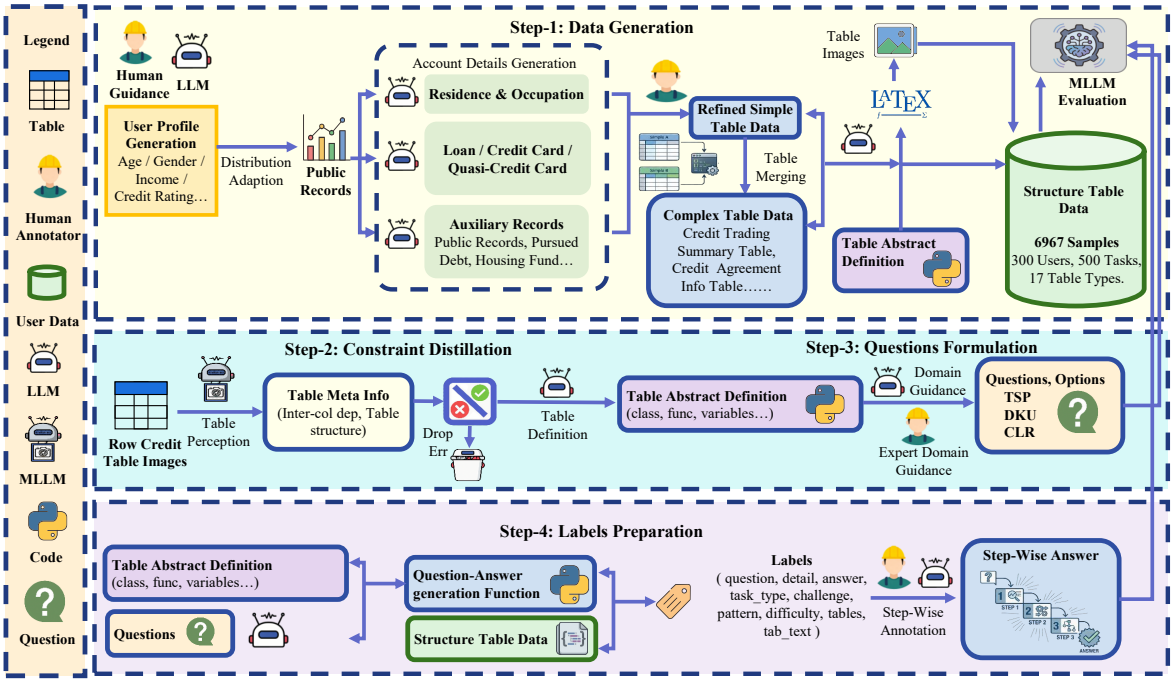

Figure 5: Construction Process of MMFCTUB.

and loan approval decision-making. Extensive experience in interpreting multi-table credit data including credit reports, bank statements, and asset verification documents.

Expert 2 (12 years, Credit Risk Manager) Focuses on credit risk modeling and portfolio risk management. Expert in analyzing credit bureau reports, identifying default indicators across multiple data tables, and developing risk assessment frameworks.

Expert 3 (11 years, Credit Review Specialist) Concentrates on credit application review and financial document verification. Proficient in cross-referencing information across credit tables, detecting inconsistencies, and validating borrower credentials.

Expert 4 (9 years, Lending Operations Manager) Manages end-to-end credit approval workflows and automation systems. Deep understanding of credit table processing requirements, data integration challenges, and operational efficiency optimization.

Expert 5 (10 years, Credit Policy Analyst) Designs and implements credit underwriting policies and scoring models. Expertise in credit scoring methodology, regulatory compliance (Basel III, local lending regulations), and credit table standardization.

Expert 6 (8 years, Credit Data Analyst) Specializes in quantitative credit analysis and tabular data processing. Expert in extracting risk signals from structured credit data, performing multi-table joint analysis, and building predictive models for creditworthiness assessment.

The expert panel provided invaluable guidance on three critical aspects. First, the practical challenges in processing and integrating multi-table credit data, including credit reports and asset statements etc.; Second, the information dependencies and cross-validation requirements across different credit tables that underwriters must consider. Lastly, the real-world task requirements for credit assessment, including question answering about borrower profiles, fact verification across multiple data sources, and comprehensive creditworthiness summarization. Their domain expertise ensured that our benchmark addresses authentic industry needs and accurately reflects the complexity of actual credit evaluation workflows in contemporary consumer lending operations.

E Credit Report Table Type Inventory

A standard Chinese personal credit report comprises multiple table types spanning account details, aggregate summaries, public records, and borrower background. Table 7 provides a complete inventory of all 17 table types covered by the MMFCTUB benchmark, organized by structural

category. Assessment dimensions: **RC** = Repayment Capacity, **RW** = Repayment Willingness, **DP** = Debt Pressure, **CF** = Cash Flow. Types 3–17 are directly evaluated by 67 TSP functions; types 1–2 provide borrower context for E2E assessment. Together with the cross-type field correspondence (Table 11) and constraint relationships (Tables 20–22), these types constitute the structural foundation of the benchmark.

F TSP: Retrieval Patterns and Visual Challenges

F.1 Visual Structure Challenges in Credit Tables

Credit report tables exhibit 14 visual structure challenges (C1–C14) that are largely absent from generic web-scraped tables. Table 8 summarizes these challenges and their impact on VLM perception.

F.2 TRG Retrieval Patterns

Based on the Table Retrieval Graph (TRG) framework (§3.2), we define 13 retrieval patterns (P1–P13) organized into four difficulty levels (S1–S4). Table 9 lists each pattern with its TRG structure and targeted visual challenges. S1 covers single-point extraction (D=1, B=1); S2 introduces multi-instance fan-out with filtering or grid parsing; S3 requires extremum localization, parallel merging, or cross-table ID references ($W \geq 2$); and S4 demands multi-hop chain retrieval with derived conditions ($D \geq 3$). TSP questions are designed to require minimal domain knowledge and computation, thereby isolating visual perception capacity. Evaluation spans both structural perception across multiple cross-table paradigms (Table 8) and scope perception across three table count ranges. As a control, we serialize table content as structured text and evaluate the same questions: models achieve $\sim$95% accuracy on text input versus significantly lower performance on image input, confirming that the performance gap is attributable to visual perception deficiency rather than reasoning limitations.

G DKU: Knowledge Categories and Field Correspondence

G.1 Three Knowledge Categories

Table 10 provides detailed descriptions and representative examples for each DKU knowledge category.

G.2 Cross-type Field Correspondence

A key SK challenge is aligning semantically equivalent fields across sibling table types. Table 11 lists the primary cross-type field mappings that models must recognize.

G.3 DKU Evaluation Knowledge Domains

DKU questions span five credit assessment knowledge domains, each requiring distinct table content scopes and domain reasoning depth. Table 10 summarizes these domains.

H CLR: Reasoning Types and Difficulty Levels

H.1 Three Sub-dimensions.

Each CEG template serves as a shared backbone from which we derive questions along three sub-dimensions. The three sub-dimensions progressively reveal more information to the model, thereby isolating different cognitive stages of the same computation:

- **CLR-F** (*Formula identification*) — *can the model construct the graph?* The model receives only a high-level assessment scenario (e.g., "evaluate this borrower's credit utilization") without computation rules or account data. It must identify the correct fields, filter conditions, and aggregation method from its own knowledge—effectively specifying the CEG's SOURCE, FILTER, and REDUCE/CALC nodes. Questions are operationalized as single- or multiple-choice over candidate field sets and computation strategies.
- **CLR-L** (*Logic reasoning*) — *can the model understand the graph?* The model receives both the explicit computation rules and the full account data as structured text. It must trace the data flow through the CEG: applying filters to determine which accounts participate, evaluating conditional branches, and reasoning about how parameter changes propagate to the output. Questions test inclusion/exclusion judgments, conditional branching outcomes, and counterfactual sensitivity.
- **CLR-N** (*Numerical computation*) — *can the model execute the graph?* The model receives the same rules and data as CLR-L, but must carry out the full end-to-end computation over large-scale inputs (typically 15–30 accounts with heterogeneous types and statuses) and

Table 7: Complete Inventory of 17 Credit Report Table Types in MMFCTUB and Their Roles in Credit Assessment.

#	Table Type	Key Content	Dim.
Background Information			
1	Residence Info	Address, residential status (owned/rented), phone, update date	RC
2	Occupation Info	Employer name, industry, position, tenure, company nature	RC, CF
Account Detail Tables — per-account records with repayment history grids			
3	Loan Account	Balance, loan amount, total/remaining periods, overdue status, five-tier classification, repayment grid; three sub-types: non-revolving, revolving, sub-account under revolving credit	RC, DP
4	Credit Card	Credit/used/shared limits, avg/max usage (6M), installment balance, overdue periods, repayment grid	DP, RW
5	Quasi-Credit Card	Overdraft balance, avg/max overdraft (6M), 180+ day unpaid amount, repayment grid	DP, RW
Summary Tables — aggregated views derived from detail records			
6	Credit Transaction Tip	Account counts and first-transaction dates by loan/card category	RC
7	Default Info Summary	Overdue/bad-debt counts per account type, peak amounts, longest overdue months	RW
8	Credit & Liability Summary	Per-category: institution count, account count, total credit exposure, aggregate balance, avg recent usage	DP, RC
9	Credit Agreement	Agreement ID, institution, effective/maturity dates, credit limit purpose, allocated/used limits	DP
Auxiliary Records — public records, contingent liabilities, non-credit transactions			
10	Pursued Debt	Subrogation records: original creditor, debt amount, transfer date, current balance	RW
11	Related Liability (Personal)	Guarantor obligations for personal loans: responsibility amount, borrower info, five-tier classification	DP
12	Related Liability (Enterprise)	Guarantor obligations for enterprise loans: responsibility amount, company info, overdue months	DP
13	Housing Provident Fund	Contribution city, monthly deposit amount, personal/company contribution ratios, employer name	CF
14	Non-credit Transaction	Utility/telecom post-payment: current arrears, payment status, 24-month payment grid	RW, CF
15	Civil Judgment	Court name, case cause, litigation amount, judgment result, closure method	RW
16	Compulsory Execution	Court enforcement: applied and executed amounts, case status, closure date	RW
17	Tax Arrears	Tax authority, taxpayer ID, arrears amount, arrears date, economy type	RW, CF

Table 8: Visual Structure Challenges in Credit Report Tables.

#	Challenge	Description	Scope
C1	Key-Value Grid Layout	Header and value rows alternate vertically rather than using a top header	Domain-specific
C2	Intra-table Column Variation	Column count changes within the same table (e.g., 6→8→6)	Domain-specific
C3	Status-driven Schema Variant	Field sets differ by account status (Normal/Overdue/Bad Debt/Settled)	Domain-specific
C4	Multi-instance Visual Homogeneity	Multiple table instances share identical schemas, differing only in values	Domain-specific
C5	Repayment Grid Dual-row Parsing	Each year occupies two rows (flag + amount); flags ∈ {N, *, 1–7}	Domain-specific
C6	Cross-page Grid Continuation	Repayment grids may span page boundaries	Domain-specific
C7	Sibling-type Field Divergence	Semantically equivalent fields use different names across sibling table types	Domain-specific
C8	Stacked Heterogeneous Sub-tables	Multiple sub-tables with different column counts stacked vertically	General
C9	Merged Cell Hierarchy	Multi-row merged cells define parent–child category relationships	General
C10	Multi-level Headers	Two-level grouped column headers requiring hierarchical parsing	General
C11	Paired Sub-table Association	Related records split across vertically separated sub-tables linked by ID	General
C12	Embedded Cross-table ID Reference	Table titles contain IDs referencing entries in other tables	General
C13	Optional Structural Segments	Some instances include sub-structures (e.g., repayment grid) while others do not	General
C14	Date-based Section Delimiter	"As-of" dates serve as structural dividers between information segments	General

produce precise numerical results. This sub-dimension isolates arithmetic execution capacity from the preceding two reasoning stages.

H.2 Template Design.

We design 11 CEG templates grounded in real credit review operations, spanning four depth levels ($D \in \{2, 3, 4, 5\}$) and covering 13 of the 14 table types in the credit report (Table 13). Templates range from single-chain computations (T11: payment anomaly rate, D=2) to multi-branch composite scoring with five parallel streams and conditional capping (T5: D=5, W=5, all six node types). Each template instantiates a subset of the three sub-dimensions depending on whether the task admits meaningful formula-identification or numerical-execution questions; CLR-L covers all 11 templates as every computation requires logical reasoning.

I Further Experiments

I.1 Further Analysis

Finding 1: Reasoning depth is the dominant difficulty factor. All models sustain near-perfect accuracy ($\geq 95\%$) at $n \leq 15$ reasoning steps but exhibit a sharp performance cliff beyond $n = 20$. At maximum depth ($n = 25$, 250 of 640 samples), even the strongest model (Gemini 3.1 Pro, text) drops to 50.0%, and the best open-source model (Qwen3-VL-235B) falls below 30%. The Pearson correlation between per-function depth and EM ($r = -0.54$) confirms depth as the primary difficulty driver.

Finding 2: The table-count effect is largely confounded with depth. Figure 6(b) shows a superficially similar decline along the breadth axis (T). However, depth and breadth are strongly correlated ($r_{nT} = 0.78$): functions requiring more tables also involve more reasoning steps. After controlling for this confound, the independent contribution of breadth to difficulty is modest, suggesting that the observed decline in EM@T is predominantly mediated by co-varying reasoning depth.

Finding 3: The text-image modality gap is model-dependent and non-uniform. The gap between text and image inputs varies drastically across models. Claude Sonnet 4.6 exhibits virtually zero gap (70.5% in both modes), indicating no measurable information loss from visual encoding. GPT-5 suffers a 23.7 pp drop from text (73.0%) to image (49.3%), accompanied by low answer yield (AYR = 52.8%), suggesting that image input triggers frequent refusals beyond perception errors. Most strikingly, Qwen3-VL-8B reverses the

Table 9: TSP Retrieval Patterns and Difficulty Levels. Operations: L = LOCATE, E = EXTRACT, F = FILTER, A = AGG, C = COMPARE, J = JOIN.

Level	ID	Pattern	TRG Structure	$\langle D, B, W \rangle$	Targeted Challenges
S1	P1	Point Retrieval	L→E	$\langle 1, 1, 1 \rangle$	C4: locate correct instance among homogeneous tables
	P9	Stacked Sub-table Localization	L→E (within stacked)	$\langle 1, 1, 1 \rangle$	C8: identify correct sub-table among column-varying segments
	P10	Merged Cell Hierarchy	L→E (multi-row scope)	$\langle 1, 1, 1 \rangle$	C9: interpret merged cell parent–child scope
S2	P2	Broadcast Aggregation	L_{all}→E_{all}→A	$\langle 2, N, 1 \rangle$	C4: exhaustive extraction without omission
	P3	Conditional Filter Aggregation	L_{all}→F→E→A	$\langle 3, N{\to}K, 1 \rangle$	C3: status field localization across schema variants
	P8	Repayment Grid Parsing	L→E_{grid}(year, month)	$\langle 2, 1, 1 \rangle$	C5+C6: dual-row disambiguation, cross-page continuation
	P11	Paired Sub-table Association	J(upper, lower) on ID	$\langle 2, 2, 1 \rangle$	C11: vertically separated sub-tables linked by record number
S3	P4	Extremum Localization	L_{all}→E_{all}→C→E	$\langle 3, N, 1 \rangle$	L-shaped retrieval: compare across instances then back-extract
	P6	Parallel Merge	($branch_1$→A) J ($branch_2$→A)	$\langle 2, *, 2 \rangle$	C7: sibling-type field divergence across parallel branches
	P12	Cross-table ID Reference	E(ID)→L(ref table)→E	$\langle 3, 2, 1 \rangle$	C12: title-embedded ID linking to agreement table
	P13	Multi-condition Cross Filter	F($cond_A \wedge cond_B$)	$\langle 3, N, 1 \rangle$	Spatially distant condition fields within same table
S4	P5	Chain Retrieval	$result_1$→$cond_2$	$\langle 3+, *, 1 \rangle$	Cross-type condition propagation
	P7	Derived Condition	A→threshold→F	$\langle 4+, N, 1 \rangle$	Two-pass traversal: compute aggregate then filter by it
	P5+P6	Chain + Parallel	composite	$\langle 4+, *, 2+ \rangle$	All challenges combined

Table 10: DKU Knowledge Categories with Definitions and Examples.

Category	**Abbr.**	**Definition**	**Representative Example**
Structure Knowledge	SK	Field-level semantic mapping across heterogeneous table types; identifying that differently named fields in sibling table types represent equivalent concepts	"Used Credit Limit" (credit card) $\equiv$ "Overdraft Balance" (quasi-credit card); both measure outstanding debt but appear in different table types with different field names and visual positions
Calculation Knowledge	CK	Domain-specific formulas required for credit assessment that are not stated in the tables; must be recalled from pre-trained knowledge	Equal principal-and-interest amortization: $M = P \cdot r(1+r)^n/[(1+r)^n-1]$; Weighted-average overdue severity: $\sum(\text{balance}_i \times \text{overdue_months}_i)/\sum \text{balance}_i$
Domain Knowledge	DK	Regulatory thresholds, risk rules, and credit assessment standards that guide decisions but are never explicit in the report	DTI (Debt-to-Income) ratio $> 50\%$ triggers rejection; "Connected Six" rule: 6 consecutive overdue months constitutes high-risk; Five-tier loan classification: Normal / Special Mention / Substandard / Doubtful / Loss

Table 11: Cross-type Field Correspondence Across Credit Table Types. Fields in the same row are semantically equivalent but use different names.

Semantic Concept	**Credit Card**	**Quasi-Credit Card**	**Loan Account**
Outstanding Debt	Used Credit Limit	Overdraft Balance	Balance / Remaining Principal
Credit Ceiling	Account Credit Limit	Account Credit Limit	Loan Amount / Account Credit Limit
Shared Ceiling	Shared Credit Limit	Shared Credit Limit	–
Shared Debt	Shared Used Limit	Shared Used Limit	–
Avg. Recent Usage	Avg. Used Limit (6M)	Avg. Overdraft Balance (6M)	–
Peak Usage	Max. Used Limit	Max. Overdraft Balance	–
Overdue Aging	Current Overdue Periods	Current Overdraft Periods	Current Overdue Periods
Monthly Obligation	Repayment Due This Month	–	Monthly Payment Due
Actual Payment	Actual Repayment This Month	Actual Repayment This Month	Actual Monthly Payment
Long-term Delinquency	–	Unpaid Balance >180 Days	Principal Overdue >180 Days

Level	Label	Subcategories
L_1	Factual Recall	SK1–3, CK4, DK1, DK4
L_2	Rel. Discrimination	SK4–6, DK2–3, DK6, DK9
L_3	Proc. Application	CK1–3, DK5
L_4	Multi-Src. Integration	DK7–8

Table 12: Mapping from knowledge subcategories to cognitive load levels.

expected pattern: image (36.7%) outperforms text (18.8%) by 17.9 pp, likely because the serialized JSON representation exceeds the effective context window of the 8B model.

Finding 4: A universal performance cliff emerges at high complexity. Across all models and modality conditions, a consistent two-phase pattern emerges: a plateau of high accuracy at low complexity ($n \leq 15$, $T \leq 3$), followed by steep decline at high complexity ($n \geq 20$, $T \geq 5$). The transition is abrupt, with median EM dropping over 30 pp between adjacent bands (n=15 $\rightarrow$ n=20), indicating a threshold effect where multi-step reasoning demands exceed model capacity. This pattern holds for both proprietary and open-source models, differing only in absolute performance level.

I.2 Fine-grained Performance

J E2E Tasks Error Propagation Analysis

To understand *where* and *why* models fail in multi-step financial reasoning, we conduct an error propagation analysis on the E2E task results. Each E2E function (E01–E26) requires the model to read credit report images, apply domain-specific rules, and execute a chain of 5–30 intermediate computation steps to produce a final answer. We decompose this reasoning chain into intermediate checkpoints and trace error origins across three capability dimensions.

Capability Dimensions. Each intermediate variable is categorized by its primary cognitive demand: **Table Structure Parsing (TSP)**—values directly extracted from credit report images (e.g., loan balances, monthly payments); **Domain Knowledge Understanding (DKU)**—variables requiring financial expertise such as risk classification or regulatory compliance judgments; and **Chain Logic Reasoning (CLR)**—values computed from upstream variables through arithmetic or logical rules (e.g., debt-to-income ratios, concentration indices). When a final answer is incorrect, we identify the *first* erroneous checkpoint along the chain and attribute the root cause to its corresponding dimension.

Root Cause Attribution. Of the 100 evaluated samples, 29 produce incorrect final answers. For the 19 cases where the first erroneous checkpoint is identifiable,[1] we report the root cause distribution in Table 17.

Domain knowledge is the primary bottleneck: DKU accounts for 52.6% of traceable root causes. A representative failure is function E23, where the model must identify applicable regulatory risk flags from a multi-page credit report; it consistently selects incorrect option subsets, reflecting a gap in domain-specific judgment rather than in perception or computation. Pure computation errors are rare—only one sample (5.3%) traces its first mistake to a CLR variable.

TSP: Frequent but Recoverable. An apparent paradox emerges from Table 17: TSP has the *highest* checkpoint error rate (20.1%) yet is *not* the dominant root cause. The explanation lies in error recoverability. We construct a transition matrix over consecutive checkpoint pairs (Table 18): when a checkpoint is erroneous, the next checkpoint is nonetheless correct 58.3% of the time, indicating substantial self-correction. Many TSP errors—such as misreading a single cell value that is not on the critical path—do not propagate to the final answer. DKU errors, by contrast, are more *fatal*: misapplying a domain rule typically derails the entire downstream chain, making them disproportionately likely to determine the final outcome despite a lower per-checkpoint error rate.

Cross-Dimensional Cascade. We further examine whether CLR checkpoint errors are genuine computation failures or cascaded consequences of upstream TSP mistakes. Among 8 CLR errors with explicit TSP dependencies, 3 (37.5%) co-occur with upstream TSP errors, while 5 (62.5%) arise despite correct inputs—confirming that most CLR errors are independent computation mistakes. Meanwhile, new errors are introduced at a 12.7% rate even after correct checkpoints (Table 18), indicating that each reasoning step carries an independent risk of failure regardless of upstream correctness.

[1] The remaining 10 cases lack matchable intermediate checkpoints due to the model's free-form output deviating from the expected structure.

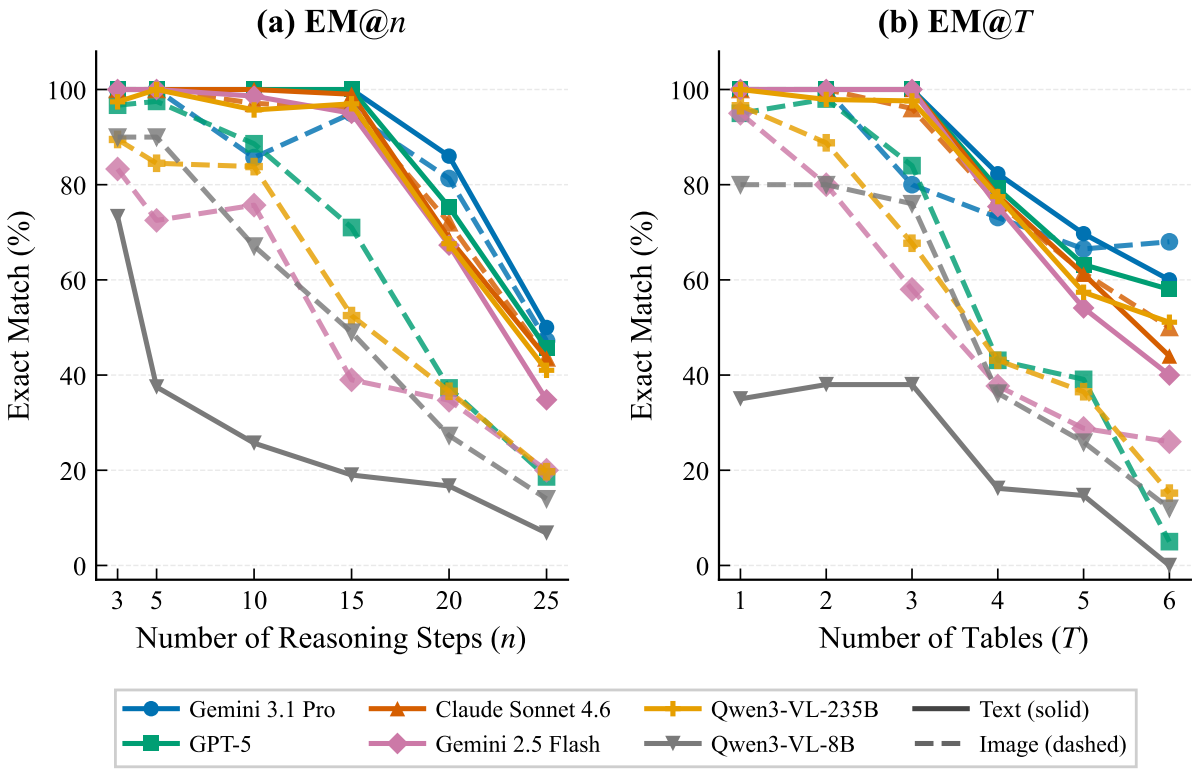

Figure 6: TSP exact match stratified by (a) reasoning depth n and (b) table count T. Solid and dashed lines denote text and image inputs, respectively. All models maintain near-perfect accuracy at low complexity ($n \leq 15, T \leq 3$) but degrade sharply beyond $n = 20$ or $T = 4$, with the text–image gap varying widely across models.

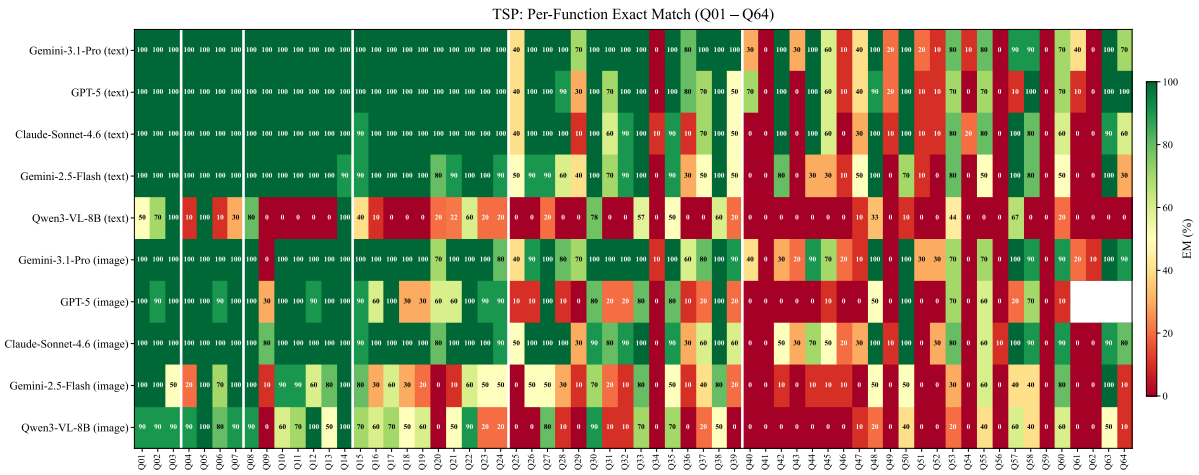

Figure 7: Per-function exact match (EM) accuracy across 64 TSP functions (Q01–Q64) for 10 model configurations (5 text, 5 image). Functions are sorted by reasoning step count n and separated by white vertical lines into six complexity bands ($n \in \{3, 5, 10, 15, 20, 25\}$). All models achieve near-perfect accuracy when $n \leq 10$, but performance diverges sharply at $n = 20$ and degrades substantially at $n = 25$, where even the strongest model (Gemini-3.1-Pro) drops below 50% on several functions.

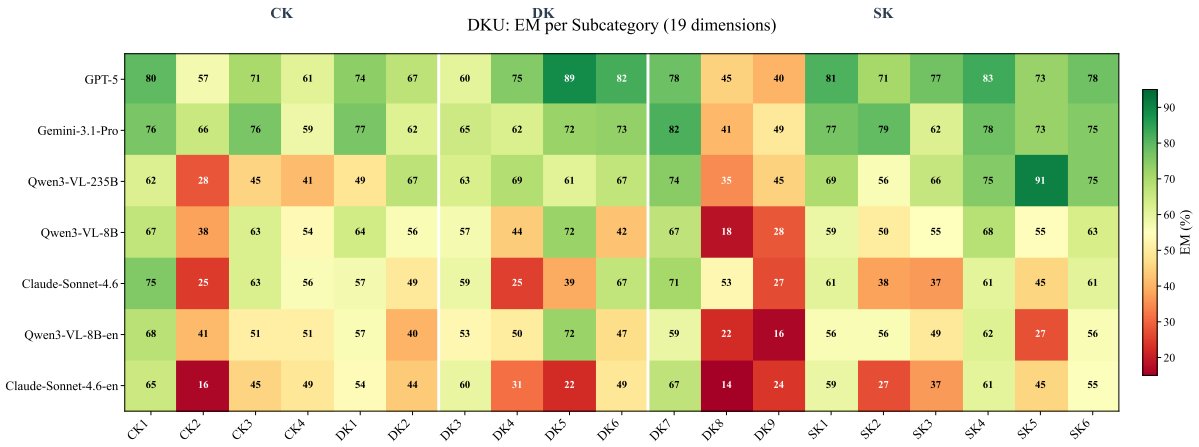

Figure 8: DKU exact match (EM) accuracy for 7 model configurations across 19 knowledge subcategories, grouped into Common Knowledge (CK), Domain Knowledge (DK), and Skill Knowledge (SK). DK8 and DK9 emerge as universally difficult subcategories, with all models scoring below 50%. GPT-5 leads on CK and DK dimensions, while cross-lingual variants (en) consistently underperform their Chinese counterparts.

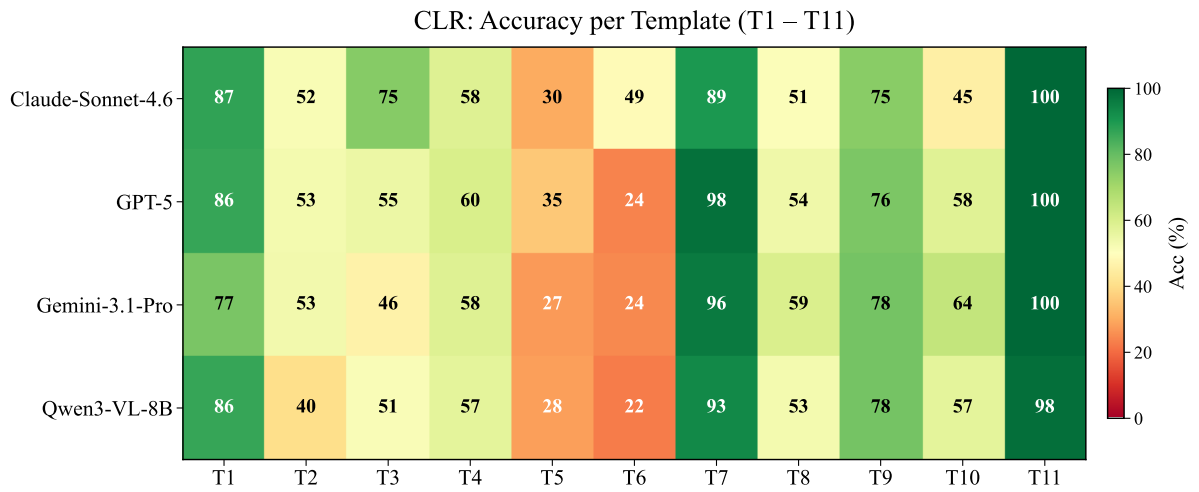

Figure 9: CLR accuracy as a function of causal event graph (CEG) depth $D \in \{2, 3, 4, 5\}$. All models start above 95% at $D = 2$ but degrade as reasoning chains lengthen. Claude-Sonnet-4.6 exhibits the flattest decline, maintaining above 60% at $D = 4$, while Gemini-3.1-Pro and Qwen3-VL-8B drop more steeply. At $D = 5$, all models converge to 27–36%, indicating a shared reasoning bottleneck at deeper causal chains.

Table 13: CLR template overview. Each template corresponds to a CEG with depth D, width W, and number of distinct node types $|K|$. The **Sub-dim.** column indicates which CLR sub-dimensions are instantiated. All 11 templates collectively cover 13 of 14 table types in the credit report.

	Template	D	W	$\|K\|$	Sub-dim.	Credit Assessment Scenario
T1	Active Debt	3	3	4	F, L, N	Aggregate outstanding balances across three account types after status-based filtering
T2	Credit Utilization	4	4	5	F, L, N	Distinguish ratio-of-sums (aggregate utilization) from average-of-ratios (per-account utilization)
T3	Payment-to-Income	4	3	5	F, L, N	Compute monthly debt service burden; requires resolving cross-type field heterogeneity
T4	Risk Rules	3	3	3	L	Evaluate multi-condition AND/OR policy rules with threshold comparisons and nested conditionals
T5	Multi-Score	5	5	6	L, N	Five-dimension composite scoring with per-dimension capping (COND) and final summation
T6	Inverse Limit	4	2	4	L, N	Reverse-engineer maximum loanable amount from affordability constraints via annuity inversion
T7	Summary–Detail	3	2	3	L	Verify consistency between pre-computed summary statistics and account-level detail records
T8	Public Risk	3	4	3	F, L	Aggregate risk exposure from tax arrears, civil judgments, and enforcement records
T9	Contingent Liability	3	1	4	L	Assess guarantee obligations with status-conditional risk weighting
T10	Income Verification	3	2	3	L	Reverse-compute salary from housing provident fund deposits and contribution ratios
T11	Payment Behavior	2	1	3	L	Compute non-credit payment anomaly rate from utility billing history

Table 14: CEG depth stratification. D=3 and D=4 each aggregate multiple diverse templates, ensuring that depth-stratified accuracy is not confounded with a single task type. N: question count per depth level across 250 data samples.

D	Templates	N
2	T11	60
3	T1, T4, T7, T8, T9, T10	878
4	T2, T3, T6	706
5	T5	360
	Total	**2,004**

Table 15: CLR sub-dimension question design. The three sub-dimensions provide progressively more information: CLR-F receives only a scenario description; CLR-L and CLR-N additionally receive explicit rules and account data. CLR-F/L use choice format; CLR-N requires numerical answers scored by relative error $< 1\%$ or compound sub-key matching.

Sub-dim.	Rules	Data	Options	Templates
CLR-F			✓	T1, T2, T3, T8
CLR-L	✓	✓	✓	T1–T11 (all)
CLR-N	✓	✓		T1, T2, T3, T5, T6

Implications. The analysis reveals a clear capability hierarchy: (1) **domain knowledge** is the most critical bottleneck—DKU errors are both the most common root cause and the least recoverable; (2) **table parsing** is the most error-prone per-step operation but benefits from downstream self-correction; (3) **chain reasoning** is the most reliable dimension, with near-zero first-error attribution, though it remains susceptible to upstream cascading failures. These findings suggest that improving financial domain grounding is a more promising direction than enhancing visual parsing or arithmetic capability alone.

K Credit Report Terminology

Table 19 provides definitions of key credit report terms used throughout the benchmark.

L Inter- and Intra-Table Constraint Relationships

Credit report tables are governed by a rich set of logical and numerical constraints that ensure data consistency. These constraints operate at three levels: within a single table (intra-table), between fields of different table types (cross-type), and between aggregate summaries and their detail sources (cross-table). Table 20 through Table 22 formalize these relationships.

L.1 Intra-Table Field Constraints

Each account table type enforces internal consistency rules among its fields. Table 20 enumerates the validated constraints for the three primary account types.

Table 16: CEG node-type composition per template. Abbreviations: **Src**=Source, **Flt**=Filter, **Red**=Reduce, **Cal**=Calc, **Cnd**=Cond. Templates with more diverse node types (e.g., T5 uses all six) demand broader reasoning.

	Template	Src	Flt	Red	Cal	Cnd	Map
T1	Act. Debt	✓	✓	✓	✓		
T2	Cred. Util.	✓	✓	✓	✓		✓
T3	Pmt-to-Inc.	✓	✓	✓	✓		
T4	Risk Rules	✓			✓	✓	
T5	Multi-Score	✓	✓	✓	✓	✓	✓
T6	Inv. Limit	✓			✓		
T7	Sum.-Detail	✓	✓	✓	✓		
T8	Pub. Risk	✓	✓	✓	✓		
T9	Cont. Liab.	✓	✓	✓			✓
T10	Inc. Verif.	✓			✓		
T11	Pmt. Behav.	✓		✓			✓

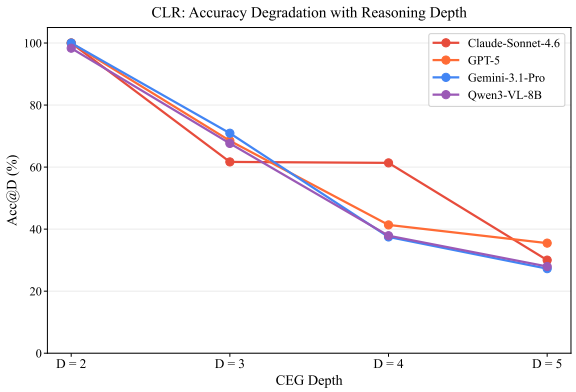

Figure 10: CLR accuracy across 11 reasoning templates (T1–T11) for four models. Templates T5 and T6, which require multi-hop counterfactual reasoning, are consistently the hardest ($\leq 30\%$ for most models). In contrast, T7 and T11 achieve near-perfect accuracy across all models, suggesting that template structure strongly modulates task difficulty independent of model capability.

L.2 Cross-Type Field Constraints

Different table types use distinct field names for semantically equivalent concepts. In addition to the field correspondence documented in Table 11, there are numerical constraints that span table type boundaries, as shown in Table 21.

L.3 Cross-Table Aggregation Constraints

Summary tables in credit reports are derived from detail tables through aggregation. These aggregation relationships constitute verifiable constraints, as shown in Table 22. Violations of these constraints indicate either data generation errors or VLM misperception.

	Root Cause		Checkpoint	
Dim.	Cnt	%	Err / Tot	Rate
DKU	10	52.6	9 / 84	10.7%
TSP	8	42.1	55 / 273	20.1%
CLR	1	5.3	29 / 300	9.7%

Table 17: Error analysis by capability dimension. **Root Cause**: first-error attribution among 19 identifiable incorrect samples. **Checkpoint**: error rate over all intermediate checkpoints across all 100 samples.

	Next Checkpoint	
Current	Correct	Wrong
Correct	145 (87.3%)	21 (12.7%)
Wrong	35 (58.3%)	25 (41.7%)

Table 18: Error cascade transitions between consecutive checkpoints. The 58.3% self-correction rate explains why frequent TSP errors do not always reach the final answer.

M External Data Alignment and Statistical Sources

To ensure the distributional validity of generated credit report data, we calibrate all key generation parameters against publicly available Chinese financial and demographic statistics through a three-layer defense architecture: (1) code-level distribution anchors derived from public sources, (2) LLM generation with post-process drift correction (deviations >50% are pulled back), and (3) automated distribution audit that flags >25% deviation from benchmarks. Table 23 lists the 16 authoritative sources referenced, and Table 24 shows the alignment between our generation parameters and real-world benchmarks across 23 distribution families.

N Abbreviation Glossary

Tables 25–29 provide the complete mapping between abbreviations used in the figures throughout this paper and their corresponding full forms. Abbreviations are grouped by semantic category for ease of reference.

Note on overloaded abbreviations. Two abbreviations are context-dependent: **CC** denotes *Consistency Check* when referring to retrieval patterns (Appendix, Fig. 2), and *Credit Card Account* when referring to table types (Main, Fig. 5). Similarly, **SC** denotes *Statistical Comparison* in the retrieval pattern context and *Single Choice* in the question type context. In all cases, the intended meaning is unambiguous from the figure in which it appears.

Table 19: Glossary of Key Credit Report Terms.

Term	Definition
Credit Agreement	A contract between lender and borrower specifying credit limit, term, and conditions; each agreement has a unique ID linking to associated account detail tables
Five-tier Classification	Regulatory loan quality classification: Normal, Special Mention, Substandard, Doubtful, Loss
Guarantee Method	Collateral type for the account: Credit (unsecured), Mortgage, Pledge, Guarantee (third-party), or Combined
Repayment Grid	Monthly repayment status matrix with dual-row encoding: row 1 shows status flags (N=normal, *=no obligation, 1–7=months overdue), row 2 shows actual payment amounts
Bad Debt (Write-off)	Account status indicating the lender has written off the debt as uncollectible; triggers a simplified table schema with fewer fields
Connected Six	Industry risk rule: 6 or more consecutive months of overdue status on any single account constitutes a high-risk indicator
Cumulative Overdue	Total count of months with overdue status across the entire repayment history of an account
Credit Utilization Rate	Ratio of outstanding balance to credit limit; a key indicator of credit consumption behavior
DTI (Debt-to-Income)	Ratio of total monthly debt obligations to monthly income; common rejection threshold at 50%
Non-credit Transaction	Non-lending financial activities (e.g., utility payments, telecom bills) reported in the credit report
Public Records	Government-sourced records including tax arrears, civil judgments, compulsory execution orders, and administrative penalties
Housing Provident Fund	Employer-matched savings program for housing; contribution records reflect employment stability
Related Repayment Liability	Debt obligations arising from acting as guarantor for other borrowers' loans
Pursued Debt	Debts where the guarantor has been pursued for repayment after the primary borrower defaults

Table 20: Intra-Table Field Constraints. Each constraint must hold for every valid account instance. Notation: b = bad debt months, n = overdue months, c = settled months, M = repayment record month range.

Table Type	#	Constraint
Loan Account (`CommonType1`)	L1	total_periods $\geq$ remaining_periods
	L2	$M \geq b + n + c$ (repayment record months $\geq$ sum of anomaly months)
	L3	current_overdue_periods $\leq n$ (current $\leq$ cumulative)
	L4	current_overdue_periods $> 0 \Leftrightarrow$ status $=$ "Overdue"
	L5	Non-revolving: balance $\leq$ loan_amount; Revolving: balance $\leq$ credit_limit
	L6	Status $=$ "Normal" $\Rightarrow$ actual_repayment $\geq$ this_month_due
	L7	Status $\in$ {"Overdue", "Bad Debt"} $\Rightarrow$ actual_repayment $<$ this_month_due
Credit Card (`DaijiType`)	D1	used_limit $\leq$ credit_limit
	D2	shared_used_limit $\leq$ shared_credit_limit
	D3	avg_used_limit_6m $\leq$ max_used_limit
	D4	current_overdue_periods $> 0 \Leftrightarrow$ status $=$ "Overdue"
Quasi-Credit Card (`ZhunDaiJiType`)	Z1	overdraft_balance $\leq$ credit_limit
	Z2	shared_used_limit $\leq$ shared_credit_limit
	Z3	overdraft_180_plus_unpaid $\leq$ overdraft_balance
	Z4	avg_overdraft_balance_6m $\leq$ max_overdraft_balance
All Account Types (Temporal)	T1	opening_date $<$ repay_start_month $<$ repay_end_month $\leq$ cutoff_date
	T2	$M =$ repay_end_month $-$ repay_start_month $+ 1$

Table 21: Cross-Type Field Constraints. These constraints relate fields across different table types that share semantic equivalence or functional dependencies.

#	Constraint Type	Description
F1	Semantic Field Equivalence	Credit card `used_limit` $\equiv$ Quasi-credit card `overdraft_balance`: both measure outstanding debt. Any cross-type aggregation of total debt must use the correct field name for each table type.
F2	Semantic Field Equivalence	Credit card `actual_repayment_this_month` $\equiv$ Loan `actual_monthly_payment`: both measure the most recent repayment amount, but the field names differ.
F3	ID Linkage	Account detail tables contain `credit_agreement_id` in their headers, which must match the `credit_agreement_id` field in the corresponding `ShouXinXieYiInfo` (Credit Agreement) table.
F4	Institution Matching	The `management_institution` field in loan accounts and the `issuing_institution` field in credit/quasi-credit card accounts refer to the same entity when the institution name matches. This enables institution-pivot aggregation across product types.
F5	Semantic Association	The `original_creditor` field in `BeiZhuiChangRecord` (Pursued Debt) may correspond to a `management_institution` in loan accounts, establishing a debtor–guarantor linkage across table types.
F6	Status Consistency	When the same account appears in both a detail table (with fine-grained status) and a summary table (with aggregated counts), the account's contribution to the summary count must be consistent with its detail-level status.

Table 22: Cross-Table Aggregation Constraints (Detail–Summary Consistency). Each row specifies a verifiable relationship between a summary table field and its constituent detail records.

#	Summary Field	Aggregation Rule from Detail Tables
ShouXinFuZhaiSummary (Credit & Liability Summary) ← Detail Tables		
A1	Non-revolving loan count	$= \lvert\{a \in \texttt{CommonType1} \mid a.\text{type} = \text{“Non-revolving”} \wedge a.\text{status} \notin \{\text{“Settled”, “Closed”}\}\}\rvert$
A2	Non-revolving loan balance	$= \sum_{a \in \text{active non-revolving}} a.\text{balance}$
A3	Credit card account count	$= \lvert\{a \in \texttt{DaijiType} \mid a.\text{status} \notin \{\text{“Settled”, “Closed”}\}\}\rvert$
A4	Credit card total used limit	$= \sum_{a \in \text{active credit cards}} a.\text{used_limit}$
A5	Quasi-credit card overdraft	$= \sum_{a \in \text{active quasi-cards}} a.\text{overdraft_balance}$
XinDaiJiaoYiTip (Credit Transaction Summary) ← Detail Tables		
A6	Account count per category	$= \lvert\{a \in \text{DetailType} \mid a.\text{business_type} \in \text{category}\}\rvert$ for each loan/card category
A7	Total balance per category	$= \sum_{a \in \text{category}} a.\text{balance}$ matching the tip's per-category balance
WeiYueXinXiSummary (Default Information Summary) ← Detail Tables		
A8	Overdue account count	$= \lvert\{a \mid a.\text{status} = \text{“Overdue”}\}\rvert$ across all credit account types
A9	Bad debt account count	$= \lvert\{a \mid a.\text{status} = \text{“Bad Debt”}\}\rvert$ across all credit account types
ID Linkage Constraints		
A10	Agreement–Account linkage	For each `ShouXinXieYiInfo` entry with ID e, $\exists$ at least one detail account whose header contains e as `credit_agreement_id`
A11	Agreement credit limit	`ShouXinXieYiInfo`.credit_limit $\geq \max(a.\text{credit_limit})$ for all accounts linked to that agreement

Table 23: Public Statistical Sources Used for Distribution Calibration. PBOC = People's Bank of China; NBS = National Bureau of Statistics; MoHURD = Ministry of Housing and Urban-Rural Development; MoF = Ministry of Finance; NFRA = National Financial Regulatory Administration; CAAM = China Association of Automobile Manufacturers.

ID	Source Document	Authority	Year
S1	Payment System Operations Overview	PBOC	2023
S2	7th National Population Census — Main Data	NBS	2020
S3	7th Census Bulletin No. 7 — Education Attainment	NBS	2020
S4	Residents' Income and Consumption Expenditure	NBS	2023
S5	China Financial Stability Report	PBOC	2023
S6	National Housing Provident Fund Annual Report	MoHURD + MoF + PBOC	2023
S7	New First-Tier Cities Ranking	Yicai Research Inst.	2024
S8	Credit Reference Center Statistics	PBOC	2023
S9	Civil Affairs Development Statistical Bulletin	Min. of Civil Affairs	2023
S10	Court Trial and Execution Main Data	Supreme People's Court	2023
S11	Financial Institution Loan Orientation Statistics	PBOC	2023
S12	Banking Financial Institution Statistics	NFRA	2023
S13	Notice on Adjusting Student Loan Policies	Min. of Education	2023
S14	Automobile Industry Statistics	CAAM	2023
S15	Guarantee Industry Analysis Report	Lianhe Credit Rating	2023
S16	7th Census Bulletin No. 2 — Gender Composition	NBS	2020

Table 24: Key Distribution Parameters and Their Alignment with Public Statistics. "Ours" shows the generation parameter; "Benchmark" shows the real-world reference value. Amounts in CNY yuan.

Category	Parameter	Ours	Public Benchmark	Src
User Profile (Layer 0 — demographic and economic anchors)				
Demographics	Peak age group (25–34)	0.38	Highest credit application share per PBOC	S2, S8
Demographics	Gender ratio (M : F)	55 : 45	Census 51.2 : 48.8; credit population skews male	S16
Education	Bachelor's degree share	33%	Census higher-education rate 23.6%; credit applicants skew higher	S3
Income	Bachelor base monthly	12,000	Per-capita disposable 4,318/mo; bachelor premium ≈2.8×	S4
Income	Tier-1 city multiplier	1.6×	Beijing/Shanghai ≈1.5–1.7× national avg	S4
Housing	Owner-occupied rate	55%	Census urban homeownership 73%; credit pop. younger	S2
Credit	Avg accounts per person	8–12	PBOC per-capita credit accounts in active population	S8
Credit	Per-capita credit cards	0.54	National all-adult average 0.54	S1
Leverage	Household DTI baseline	63.5%	Household sector leverage ratio 72.4% (end 2023)	S5
Numerical Fields (Layer 1 — account-level distribution anchors)				
Credit Card	Avg credit limit	29,600	National avg card limit 29,600	S1
Credit Card	Utilization rate (weighted)	40.4%	National utilization 38.34%	S1
Housing Loan	Median amount	1,000,000	Housing loan balance 38.17T (dominant category)	S11
Housing Loan	Tier-1 city multiplier	1.8×	Tier-1 avg housing price ≈1.8× national	S4
Provident Fund	Govt employee participation	98%	Institutional staff near-universal coverage	S6
Provident Fund	Private enterprise part.	65%	53.47% of 175M total contributors	S6
Student Loan	Amount range	6K–160K	Undergrad cap 16K/yr; grad 20K/yr	S13
Consumer Loan	Median amount	80,000	Consumer loan balance 19.77T	S11, S12
Auto Loan	Median amount	150,000	Avg vehicle price ≈180K; LTV 60–80%	S14

Table 25: Abbreviations for evaluation dimensions (L1) and sub-dimensions (L2) used in Figure 2.

Abbr.	Full Form	Level
TSP	Table-based Structured Perception	L1
DKU	Domain Knowledge Understanding	L1
CLR	Credit Logical Reasoning	L1
E2E	End-to-End Evaluation	L1
SK	Structural Knowledge	L2 (DKU)
CK	Computational Knowledge	L2 (DKU)
DK	Domain Knowledge	L2 (DKU)
CLR-F	Fact-based Reasoning	L2 (CLR)
CLR-L	Logic-based Reasoning	L2 (CLR)
CLR-N	Numerical Reasoning	L2 (CLR)

Table 26: Abbreviations for TSP fine-grained retrieval categories (L3), grouped by sub-dimension.

Abbr.	Sub-dim.	Full Form
PR	Depth	Point Retrieval
ChR	Depth	Chain Retrieval
DC	Depth	Derived Condition
MH	Depth	Merged-cell Hierarchy
CtR	Depth	Cross-table ID Reference
EL	Breadth	Extremum Localization
PA	Breadth	Paired-subtable Association
MI	Breadth	Multi-condition Intersection
CF	Breadth	Conditional Filtering
BA	Breadth	Broadcast Aggregation
GP	Breadth	Grid Parsing
SL	Breadth	Stacked-subtable Localization
PM	Width	Parallel Merging
DSC	Width	Detail–Summary Consistency
IPA	Width	Institution Panorama Aggregation

O LLM Dependency Audit

Table 30 expands the compact audit in Table 2 with detailed generation methods and constraint mechanisms for each pipeline component.

P Prompts

Table 27: Abbreviations for DKU fine-grained knowledge categories (L3), grouped by sub-dimension.

Abbr.	Sub-dim.	Full Form
FL	SK	Field Layout
StS	SK	Subtable Structure
GE	SK	Grid Encoding
TbR	SK	Table Reference
MgC	SK	Merged Cell
IsF	SK	Isomorphic Fields
FSl	CK	Field Selection
Fml	CK	Formula
AgR	CK	Aggregation Rule
UP	CK	Unit / Precision
EnR	DK	Encoding Rule
ClS	DK	Classification Standard
PdR	DK	Product Rule
Reg	DK	Regulatory Knowledge
RkS	DK	Risk Signal
FTM	DK	Field–Table Mapping
RtP	DK	Retrieval Path
SDR	DK	Summary–Detail Rule
RpO	DK	Report Overview

Table 28: Abbreviations for CLR fine-grained reasoning tasks (L3). Some tasks appear in multiple CLR sub-dimensions with different difficulty levels.

Abbr.	Full Form
TAD	Total Active Debt
CUR	Credit Utilization Rate
PIR	Payment-to-Income Ratio
RPE	Risk Policy Rule Evaluation
MCS	Multi-dimensional Composite Scoring
ILC	Inverse Limit Calculation
SDV	Summary–Detail Verification
PRE	Public Records Risk Exposure
CLA	Contingent Liability Assessment
IncV	Income Verification
NCP	Non-Credit Payment Behavior

Prompt for Residence and Occupation Information Generation (Layer 1a)

You are an experienced credit assessment expert. Based on the provided user profile, generate the user's residence history and occupation history.

User Profile:
Age: {age} Gender: {gender} Education: {education} Marital Status: {marital_status}
City: {city_name} (Tier-{city_tier}) Monthly Income: {average_month_income} CNY
Employment Type: {employment_type} Housing Status: {property_status}

Output Requirements: Generate a JSON object containing two fields: `"live_tab"` and `"prof_tab"`.

live_tab (Residence Information Array) — Each record contains:
- `"id"`: String starting from `"1"`
- `"address"`: Detailed address; must be located in {city_name} or a city within the same province
- `"phone"`: Area code–number format (e.g., `"010-83234323"`), or `"Unknown"`
- `"status"`: One of `"Self-owned"`, `"Mortgage"`, `"Provided by Relatives"`, `"Rented"`, `"Dormitory"`, `"Other"`, `"Unknown"`
- `"update_date"`: `YYYY.MM.DD` format

Include 1–4 entries in chronological order, reflecting the residence trajectory. The most recent entry must be consistent with the housing status {property_status} from the user profile.

prof_tab (Occupation Information Array) — Each record contains:
- `"id"`: String starting from `"1"`
- `"employer"`: Anonymized name (e.g., `"XX Technology Co., Ltd."`)
- `"employer_type"`: One of `"Gov./Public Inst."`, `"State-owned"`, `"Foreign"`, `"Private"`, `"Other"`
- `"address"`, `"phone"`: Employer's detailed address and phone number
- `"occupation"`: e.g., `"Technical Professional"`, `"Clerical Staff"`, `"Service Worker"`
- `"industry"`: e.g., `"Finance"`, `"IT & Software Services"`, `"Manufacturing"`
- `"position"`: One of `"Senior Executive"`, `"Middle Manager"`, `"General Employee"`, `"Other"`
- `"title"`: One of `"Senior"`, `"Intermediate"`, `"Junior"`, `"None"`, `"Other"`
- `"start_year"`: `YYYY` format `"update_date"`: `YYYY.MM.DD` format

Include 1–4 entries reflecting career progression. The latest `employer_type` must correspond to {employment_type}. Position and professional title should be consistent with {education} and {age}.

Format: Output only a single JSON object with no analysis or extraneous text.

```
{ "live_tab": [{"id": "1", ...}], "prof_tab": [{"id": "1", ...}] }
```

Figure 11: Prompt template for generating synthetic residence and occupation records in Layer 1a of the data construction pipeline. The LLM acts as a credit assessment expert and produces structured JSON conditioned on the user profile from Layer 0.

Prompt for Loan Account Detail Generation

You are an experienced credit assessment expert. Based on the given user profile and the specified account parameters, generate detailed information for a single loan account.

User Profile:
Monthly Income: {average_month_income} CNY Credit Rating: {credit_rating} (Risk Level: {risk_level}) City: {city_name} (Tier-{city_tier}) Education: {education} Employment Type: {employment_type} Housing: {property_status} Age: {age} Debt-to-Income Target: {debt_to_income_ratio}

Specified Parameters (must be strictly followed):
Account Type: {account_type} Business Type: {business_type} Account Status: {account_status}

Output Fields & Constraints: Generate a JSON object containing the following fields.

Basic Information:
- `management_institution`: Anonymized institution name; prefix randomly chosen from {`Commercial Bank`, `Joint-stock Bank`, `City Commercial Bank`, `Rural Commercial Bank`}, suffix = 2 random uppercase letters (e.g., `"Commercial Bank BT"`). Must differ across accounts of the same user.
- `currency`: Fixed as `"CNY"`.
- `guarantee_method`: One of `"Collateral"`, `"Pledge"`, `"Guarantee"`, `"Credit/Unsecured"`, `"Combined"`. Housing loans → `Collateral`; consumer/student loans → `Credit/Unsecured` or `Guarantee`; business loans → `Collateral` or `Combined`.
- `opening_date`: `YYYY.MM.DD`, within 2015.01.01 – 2024.06.30.
- `cutoff_date`: Within 2024.06.01 – 2025.06.30.
- `start_time`: `YYYY-MM`, the first month after `opening_date`.
- `end_time`: `YYYY-MM`, must share the same year-month as `cutoff_date`.

Amount & Periods:
- {amount_field_instruction} *(dynamically injected based on account type)*
- `total_periods`: Total repayment periods (integer). {period_instruction}
- `remaining_periods`: Remaining periods (integer), $\leq$ `total_periods`.
- `repayment_method`: One of `"Equal Principal"`, `"Equal Installment"`, `"Lump Sum at Maturity"`, `"Interest Periodic, Principal at Maturity"`.

Status Information: {status_field_instruction} *(dynamically injected based on account status)*

Five-tier Classification:
- `five_classification` $\in$ {`Normal`, `Concern`, `Substandard`, `Doubtful`, `Loss`}. Mapping rules:
  Status `Normal` → `Normal` or `Concern`; Status `Overdue` → `Substandard` or `Doubtful`; Status `Bad Debt` → `Loss`; Status `Settled/Transferred` → `Normal`.

Repayment Record Statistics (for downstream monthly record generation):
- `b_count`: Bad-debt months (int). {b_count_instruction}
- `num_count`: Total overdue months (int). {num_count_instruction}
- `c_count`: Settled-flag months (int); equals 1 iff `account_status` = `Settled`, else 0.
- Constraint: `b_count` + `num_count` + `c_count` + $N \leq$ `month_range`, where $N \geq 5$ (normal months).

Hard Inequality Constraints (all must hold; violating any is an error):

1. `opening_date` < `start_time` $\leq$ `end_time`
2. `end_time` year-month = `cutoff_date` year-month
3. `balance` ≥ 0
4. Non-revolving: `balance` $\leq$ `loan_amount`
5. Revolving: `balance` $\leq$ `credit_limit`
6. `this_month_due` $\leq$ `balance`
7. `remaining_periods` $\leq$ `total_periods`
8. `current_overdue_periods` > 0 iff `account_status` = `Overdue`
9. `b_count` > 0 iff `account_status` = `Bad Debt`; else must be 0
10. `num_count` > 0 only when `account_status` = `Overdue`; else must be 0
11. `num_count` $\geq$ `current_overdue_periods`

Format: Output only a single JSON object. All monetary values are integers; dates must follow the specified formats exactly.

Figure 12: Prompt template for generating synthetic loan account records. The LLM produces structured JSON containing account details, repayment parameters, and status information, subject to 11 hard inequality constraints that enforce financial consistency. Template variables in braces (e.g., `{amount_field_instruction}`) are dynamically injected based on account type and status.

Prompt for Credit Card Account Generation (Layer 1b — Credit Card)

You are an experienced credit assessment expert. Based on the given user profile and the specified account status, generate detailed information for a single credit card account.

User Profile:
Monthly Income: {average_month_income} CNY Credit Rating: {credit_rating} (Risk: {risk_level}) City: {city_name} Age: {age}

Specified Parameters: Account Status: {account_status}

Output Fields & Constraints:

Basic Information:
- `issuing_institution`: Anonymized; prefix from {`Card Issuer`, `Bank Credit Card Center`, `Commercial Bank Card`} + 2 random uppercase letters. Must differ across cards of the same user.
- `credit_limit`: Credit limit (int), typically 2–15× monthly income.
- `shared_credit_limit`: Shared credit limit (int), $\geq$ `credit_limit`.
- `currency`: Fixed as `"CNY"`.
- `guarantee_method`: Typically `"Credit/Unsecured"`.
- `opening_date`: `YYYY.MM.DD`, within 2015.01.01 – 2024.06.30.
- `cutoff_date`: Within 2024.06.01 – 2025.06.30.
- `start_time`: `YYYY-MM`, first month after `opening_date`.
- `end_time`: `YYYY-MM`, same year-month as `cutoff_date`.

Status-dependent Fields (injected based on `account_status`):
- **Normal**: `used_limit` $\leq$ `credit_limit`; `balance` $\leq$ `used_limit`; `avg_used_limit_6m` $\leq$ `max_used_limit` $\leq$ `credit_limit`; `large_installment_balance` ≥ 0; `this_month_due` $\approx 10\%$ of balance; `actual_repayment` $\geq$ `this_month_due`; `current_overdue_periods` = 0.
- **Overdue**: Same utilization fields; `actual_repayment` $<$ `this_month_due`; `current_overdue_periods` $\in [1, 7]$, $\leq$ `num_count`.
- **Bad Debt**: `balance` > 0 (frozen); all utilization/repayment fields = 0.
- **Closed / Inactive**: All numeric fields = 0.

Repayment Record Statistics:
- `b_count`: Bad-debt months; > 0 iff `Bad Debt`, else 0.
- `num_count`: Overdue months; > 0 iff `Overdue`, else 0.
- `c_count`: Closed-flag months; $= 1$ iff `Closed`, else 0.
- Constraint: `b_count` + `num_count` + `c_count` + $N \leq$ `month_range` ($N \geq 5$).

Hard Inequality Constraints (all must hold):
1. `opening_date` $<$ `start_time` $\leq$ `end_time`
2. `end_time` year-month $=$ `cutoff_date` year-month
3. `shared_credit_limit` $\geq$ `credit_limit`
4. `used_limit` $\leq$ `credit_limit`
5. `avg_used_limit_6m` $\leq$ `max_used_limit` $\leq$ `credit_limit`
6. `balance` $\leq$ `used_limit`, `balance` ≥ 0
7. `current_overdue_periods` > 0 iff `Overdue`
8. `b_count` > 0 iff `Bad Debt`; `num_count` > 0 only when `Overdue`
9. `num_count` $\geq$ `current_overdue_periods`

Format: Output only a single JSON object. All monetary values are integers.

Figure 13: Prompt template for generating synthetic credit card (revolving) account records in Layer 1b. Status-dependent fields are dynamically injected based on `account_status` $\in$ {`Normal`, `Overdue`, `Bad Debt`, `Closed`, `Inactive`}, with 9 hard inequality constraints enforcing financial consistency.

Prompt for Quasi-Credit Card Account Generation (Layer 1b)

You are an experienced credit assessment expert. Based on the given user profile and the specified account status, generate detailed information for a single quasi-credit card account.

User Profile:
Monthly Income: {average_month_income} CNY Credit Rating: {credit_rating} (Risk: {risk_level}) City: {city_name} Age: {age}

Specified Parameters: Account Status: {account_status}

Output Fields & Constraints:

Basic Information:
- `issuing_institution`: Anonymized; prefix from {`Card Issuer`, `Bank Quasi-Credit Card Center`, `City Commercial Bank Card`} + 2 random uppercase letters.
- `credit_limit`, `shared_credit_limit` ($\geq$ `credit_limit`): Credit limits (int).
- `currency`: Fixed as `"CNY"`.
- `guarantee_method`: Typically `"Credit/Unsecured"`.
- `opening_date`, `cutoff_date`, `start_time`, `end_time`: Same date constraints as credit cards.
- **Note:** Quasi-credit cards have **no** `business_type` field (not shown in credit reports).

Status-dependent Fields (injected based on `account_status`):
- **Normal**: `overdraft_balance` $\leq$ `credit_limit`; `avg_overdraft_balance_6m` $\leq$ `max_overdraft_balance` $\leq$ `credit_limit`; `overdraft_180_plus_unpaid` typically $= 0$; `actual_repayment` > 0.
- **Overdue**: Same fields; `overdraft_180_plus_unpaid` $\leq$ `overdraft_balance`. Note: flags 1–2 are grace period (not overdue); flags 3–7 are true overdue.
- **Bad Debt**: `overdraft_balance` > 0 (frozen); `max_overdraft_balance` $=$ `overdraft_balance`; `overdraft_180_plus_unpaid` $=$ `overdraft_balance`; `actual_repayment` $= 0$.
- **Settled**: All numeric fields $= 0$.

Repayment Record Statistics:
- `b_count`: Bad-debt months; > 0 iff `Bad Debt`, else 0.
- `num_count`: Overdraft overdue months; > 0 iff `Overdue`, else 0.
- `c_count`: Settled-flag months; $= 1$ iff `Settled`, else 0.
- Constraint: `b_count` + `num_count` + `c_count` + $N \leq$ `month_range` ($N \geq 5$).

Hard Inequality Constraints (all must hold):
1. `opening_date` $<$ `start_time` $\leq$ `end_time`; `end_time` year-month $=$ `cutoff_date` year-month
2. `shared_credit_limit` $\geq$ `credit_limit`
3. `overdraft_balance` $\leq$ `credit_limit`
4. `overdraft_180_plus_unpaid` $\leq$ `overdraft_balance`
5. `avg_overdraft_balance_6m` $\leq$ `max_overdraft_balance` $\leq$ `credit_limit`
6. `b_count` > 0 iff `Bad Debt`; `num_count` > 0 only when `Overdue`

Format: Output only a single JSON object. All monetary values are integers.

Figure 14: Prompt template for generating synthetic quasi-credit card account records in Layer 1b. Unlike standard credit cards, quasi-credit cards use overdraft-based fields and omit the `business_type` field per credit report conventions.

Prompt for Debt Recovery Record Generation (Layer 1c)

You are an experienced credit assessment expert. Based on the given user profile and account overview, generate the user's debt recovery (subrogation) records.

User Profile:
Monthly Income: {average_month_income} CNY Credit Rating: {credit_rating} (Risk: {risk_level}) City: {city_name}

Account Overview: {accounts_summary} *(dynamically constructed from Layer 1b outputs, e.g., "3 loan accounts, 2 credit cards, 1 quasi-credit card")*

Output Requirements:
Generate a JSON array. Each element is a debt recovery record with the following fields (Chinese keys):
- `id`: String starting from `"1"`
- `management_institution`: Anonymized, e.g., `"Guarantee Co. CE"`, `"Asset Mgmt. Co. DK"`
- `business_type`: e.g., `"Subrogation Debt"`, `"Guarantee Recovery"`
- `claim_receipt_date`: `YYYY.MM.DD` format
- `claim_amount`: Amount string (e.g., `"85000"`)
- `repayment_status_at_transfer`: Integer string (overdue months, e.g., `"3"`)
- `special_transaction_type`: e.g., `"Third-party Subrogation"`, `"Debt-for-Asset Swap"`
- `transaction_date`: `YYYY.MM.DD` format
- `change_months`: Integer string
- `transaction_amount`: Amount string, $\leq$ `claim_amount`
- `detail_record`: 1–2 sentence description of the recovery situation

Count: 1–3 records, consistent with the user's risk level.

Format: Output only a JSON array. No analysis or extraneous text.

Figure 15: Prompt template for generating synthetic debt recovery (subrogation) records in Layer 1c. The `accounts_summary` field is dynamically constructed from Layer 1b outputs to provide account context. Only generated for users flagged with `has_beizhuichang = True`.

Prompt for Non-Credit Transaction Record Generation (Layer 1c)

You are an experienced credit assessment expert. Based on the given user profile, generate the user's non-credit transaction records.

User Profile:
Monthly Income: {average_month_income} CNY City: {city_name} Age: {age}

Output Requirements:
Generate a JSON array of {num_records} records. Each element contains:

- `id`: String starting from `"1"`
- `business_type`: One of `"Telecom"`, `"Water"`, `"Electricity"`, `"Gas"`, `"Property Management"`
- `service_provider`: Anonymized name, e.g., `"XX Mobile Communications Co."`, `"XX City Water Supply Co."`
- `service_start_date`: `YYYY.MM.DD` format
- `current_arrears`: Integer (CNY); mostly 0 (no arrears), occasionally small amounts
- `account_status`: `"Normal"` or `"Arrears"`
- `update_date`: `YYYY.MM.DD` format

Format: Output only a JSON array. No analysis or extraneous text.

Figure 16: Prompt template for generating synthetic non-credit transaction records (telecom, utilities, property management) in Layer 1c. Only generated for users with `num_feixindai` > 0.

Prompt for Public Record Generation (Layer 1c)

You are an experienced credit assessment expert. Based on the given user profile, generate the user's public records (adverse public information).

User Profile:
Monthly Income: {average_month_income} CNY Credit Rating: {credit_rating} (Risk: {risk_level}) City: {city_name} Age: {age} Employment: {employment_type}

Output Requirements: Generate a JSON object containing four record arrays (each may be an empty array `[]`):

1. Tax Arrears Records (`qianshui_records`):

- `id`, `tax_authority` (anonymized, e.g., `"XX District Tax Bureau"`), `taxpayer_id` (15–20 chars), `statistics_date` (`YYYY.MM.DD`), `total_arrears` (int, CNY), `economic_type` (e.g., `"Private"`, `"Individual"`), `update_date`

2. Civil Judgment Records (`minshi_records`):

- `id`, `court` (anonymized), `cause_of_action` (e.g., `"Loan Contract Dispute"`, `"Credit Card Dispute"`), `filing_date`, `resolution_method` $\in$ {`Judgment`, `Mediation`, `Withdrawal`, `Other`}, `resolution_date` (empty string if pending), `subject_matter` (text description), `judgment_amount` ($\leq$ disputed amount), `case_status` $\in$ {`Closed`, `Under Trial`}, `update_date`

3. Compulsory Execution Records (`qiangzhi_records`):

- `id`, `court`, `cause_of_action`, `filing_date`, `resolution_method` $\in$ {`Completed`, `Terminated This Round`, `Settlement`} (only if closed), `case_status` $\in$ {`Closed`, `In Execution`}, `closure_date` (only if closed), `applied_amount` (int), `executed_amount` (int, $\leq$ `applied_amount`), `remaining_amount` ($=$ `applied` $-$ `executed`), `update_date`

4. Administrative Penalty Records (`xingzheng_records`):

- `id`, `penalty_authority` (anonymized), `penalty_description`, `penalty_amount` (int, CNY), `penalty_date`, `effective_date` ($\geq$ `penalty_date`), `expiry_date` (may be empty), `update_date`

Count Guidance: 0–2 records per category. For risk level "{risk_level}", the total should be reasonable (low-risk users typically have only 1–2 minor records).

Format: Output only a JSON object: `{"qianshui_records": [...], "minshi_records": [...], "qiangzhi_records": [...], "xingzheng_records": [...]}`.

Figure 17: Prompt template for generating synthetic public records in Layer 1c, covering four categories of adverse information: tax arrears, civil judgments, compulsory executions, and administrative penalties. Only generated for users flagged with `has_public_records = True`.

Prompt for Related Loan Responsibility Generation (Layer 1c)

You are an experienced credit assessment expert. Based on the given user profile, generate the user's related repayment responsibility records.

User Profile:
Monthly Income: {average_month_income} CNY Credit Rating: {credit_rating} City: {city_name} Age: {age} Marital Status: {marital_status}

Output Requirements: Generate a JSON object containing two arrays:

personal_related (Personal Borrower Guarantees) — each record contains:
- `id`: String starting from `"1"`
- `borrower_name`: Anonymized, e.g., `"Zhang *Wei"`
- `id_type`: One of `"ID Card"`, `"Military ID"`, `"Passport"`
- `id_number`: 18-digit ID format (may be fictional)
- `lending_institution`: Anonymized, e.g., `"Bank of XX"`
- `contract_amount`: Amount string (e.g., `"500000"`)
- `responsibility_amount`: Amount string, $\leq$ `contract_amount`
- `responsibility_status`: One of `"Normal"`, `"Overdue"`, `"Bad Debt"`, `"Settled"`
- `current_balance`: Amount string, $\leq$ `contract_amount`
- `overdue_months`: Integer string; `"0"` when `Normal` or `Settled`
- `overdue_amount`: Amount string; `"0"` when `Normal` or `Settled`
- `guarantee_method`: One of `"Collateral"`, `"Pledge"`, `"Guarantee"`, `"Credit/Unsecured"`
- `update_date`: `YYYY.MM.DD` format

enterprise_related (Enterprise Borrower Guarantees) — each record contains:
- `id`: String starting from `"1"`
- `enterprise_name`: Anonymized, e.g., `"XX Co., Ltd."`
- `cert_type`: `"Organization Code"` or `"Unified Social Credit Code"`
- `cert_number`: 18-digit code (may be fictional)
- `lending_institution`, `contract_amount`, `responsibility_amount`, `responsibility_status`, `current_balance`, `guarantee_method`, `update_date`: Same constraints as personal records (without `overdue_months`/`overdue_amount`)

Count Guidance:
- `personal_related`: 0–3 records (married users more likely to have guarantee obligations)
- `enterprise_related`: 0–2 records (self-employed / business loan users more likely)
- At least one array must be non-empty

Format: Output only a JSON object: `{"personal_related": [...], "enterprise_related": [...]}`.

Figure 18: Prompt template for generating synthetic related loan responsibility records in Layer 1c, covering both personal and enterprise guarantee obligations. Only generated for users flagged with `has_related_loans = True`.

Table 29: Abbreviations for retrieval patterns, table types, question types, and operation types used in the appendix figures.

Abbr.	Category	Full Form
Retrieval Patterns (Appendix, Fig. 2)		
MTA	Pattern	Multi-type Aggregation
NL	Pattern	Nested Lookup
CC	Pattern	Consistency Check
S&M	Pattern	Sort & Match
SC	Pattern	Statistical Comparison
Table Types (Main, Fig. 5)		
CC	Table	Credit Card Account
Loan	Table	Loan Account
C/D	Table	Credit / Debt Summary
OD	Table	Overdue Information Summary
CA	Table	Credit Agreement
Question Types (Main, Fig. 8)		
OE	Q-Type	Open-ended
MC	Q-Type	Multiple Choice
SC	Q-Type	Single Choice
Num	Q-Type	Numerical
Cmp	Q-Type	Compound
Operation Types (Appendix, Fig. 4)		
Ext	Op-Type	Extract
Arith	Op-Type	Arithmetic
Agg	Op-Type	Aggregate
Cond	Op-Type	Conditional
Sort	Op-Type	Sort

TSP Evaluation Prompt (Shared System Prompt + Per-Type Format)

`[System]`

You are a financial analysis expert. Based on the provided personal credit report table {images / text data}, answer the question.

You may first perform analysis and reasoning, but the final answer **must** be enclosed in the answer tag.

Format Requirements (must be strictly followed):

Open-ended (Numerical) — for questions asking "how much", "total", "compute . . . ":
- Output integers only; no units, no decimal points, no comma separators.
- Example: `[[ Answer ]] 30000`

Single Choice:
- Output exactly one option letter (A/B/C/D); no period or other text.
- Example: `[[ Answer ]] B`

Multiple Choice:
- Output all correct option letters in alphabetical order with no separators.
- Examples: `[[ Answer ]] AC` `[[ Answer ]] ABD`

General Rules:
1. All numbers must be integers; no `.0` suffix; no thousands separators.
2. Read each step's computation requirements carefully and calculate step by step.
3. Distinguish identically named fields across different tables (e.g., the "used limit" in the summary table vs. in the detail table).
4. Round all intermediate results to integers before proceeding to the next step.

`[User]`

Image mode: {credit report table images} + {question text}
Text mode: {serialized table text (tab_text)} + {question text}

Figure 19: Evaluation prompt for the TSP (Table-based Structured Perception) dimension, supporting three question types: open-ended numerical, single-choice, and multiple-choice. The system prompt enforces strict output formatting with the answer tag. Two input modes are used: image mode (table screenshots) and text mode (serialized table text), enabling comparison of visual vs. textual table understanding.

DKU Evaluation Prompts by Question Type

Type 1: Single Choice

`[System]`
You are an expert in the Chinese second-generation credit reporting system (personal credit reports). Based on your professional knowledge, answer the following single-choice question. Output only the letter of the correct option (e.g., A) with no additional content.

`[User]`
{question}
A. {option_A} B. {option_B} C. {option_C} D. {option_D}

Type 2: Multiple Choice

`[System]`
You are an expert in the Chinese second-generation credit reporting system (personal credit reports). Based on your professional knowledge, answer the following multiple-choice question. Output only the letters of all correct options (e.g., ABD) with no additional content.

`[User]`
{question} (Multiple selection)
A. {option_A} B. {option_B} C. {option_C} D. {option_D} E. {option_E}

Type 3: Open-ended QA

`[System]`
You are an expert in the Chinese second-generation credit reporting system (personal credit reports). Based on your professional knowledge, answer the following question. Output strictly in the format below with no extraneous content:

`[Answer]` Your brief answer (one sentence, $\leq$ 50 characters)
`[Explanation]` Brief rationale (1–3 sentences)

`[User]`
{question}

Figure 20: Evaluation prompt templates for the DKU (Domain Knowledge Understanding) dimension, covering three question types. The system prompt establishes the LLM's role as a Chinese credit reporting expert. Single-choice and multiple-choice questions require letter-only output; open-ended QA uses a structured `[Answer]` + `[Explanation]` format.

CLR Evaluation Prompt: System Prompt + Question Type Examples

System Prompt (shared across all CLR question types):

You are a precise financial credit assessment calculation assistant. Based on the given rules and data, perform strict step-by-step reasoning before answering the question.

Output Format Requirements (must be strictly followed):

1. First write the analysis process.
2. The last line **must** begin with the answer tag, followed by the final answer:
   - *Single/Multi-choice*: `[[ Answer ]] A` or `[[ Answer ]] A, B, C` (comma-separated, uppercase)
   - *Numerical*: `[[ Answer ]] 123456` (pure number, no units or commas)
   - *Yes/No*: `[[ Answer ]] Yes` or `[[ Answer ]] No`
   - *Percentage*: `[[ Answer ]] 45.87` (pure number, no % sign)
   - *Multiple values*: `[[ Answer ]] Loan=887600; CreditCard=279500; Total=1270600`

Note: The answer tag appears **exactly once**, at the very end of the response.

Type 1: Single Choice (CLR-F-1c)

`[Scenario]`
When aggregating active liabilities across three account types, the correct aggregation method must be determined.

`[Question]`
When aggregating active liabilities across three account types, the correct approach is?

`[Options]`
A. Sum the balances of each type separately, then add the three subtotals
B. Put all account balances into a single list and take the average
C. Sum the balances of each type separately, then take the maximum of the three
D. Compute a weighted average by the number of accounts in each type

Please select one correct option. Write on the last line: `[[ Answer ]] X`

Type 2: Multiple Choice (CLR-F-1a)

`[Scenario]`
You are evaluating a borrower's total active liabilities. The borrower has three account types: non-revolving loans, credit cards, and quasi-credit cards.

`[Question]`
Which of the following fields should be used to represent each account type's "outstanding liability"? (Multiple selection)

`[Options]`
A. Credit Card — Used Limit B. Credit Card — Balance C. Quasi-Credit Card — Overdraft Balance
D. Quasi-Credit Card — Credit Limit E. Non-Revolving Loan — Balance F. Non-Revolving Loan — Loan Amount

Please select all correct options. Write on the last line: `[[ Answer ]] X, Y, Z`

Type 3: Numerical (CLR-N-1)

`[Calculation Rule]`
Total Active Liability = Σ(loan balances with status `Normal` or `Overdue`) + Σ(credit card balances with status `Normal` or `Overdue`) + Σ(quasi-credit card overdraft balances with status `Normal` or `Overdue`)

`[Data]`
Borrower's account data (19 accounts total):
Non-revolving loans (8): L01: Balance=290,000 Status=Overdue; L02: Balance=150,000 Status=Bad Debt; L03: Balance=280,000 Status=Overdue; ...
Credit cards (8): C01: Balance=0 Status=Closed; C02: Balance=133,000 Status=Overdue; ...
Quasi-credit cards (3): Z01: Overdraft=49,000 Status=Normal; ...

`[Question]`
Calculate the total active liability. List the three subtotals and the grand total.
Compute step by step. Write on the last line: `[[ Answer ]] Field1=Value1; Field2=Value2; . . .`

Type 4: Compound (CLR-L-2a)

`[Calculation Rule]`
Credit Utilization Rate = Σ(active used limits + active quasi-credit card overdrafts) / Σ(corresponding credit limits)

`[Data]`
Borrower's credit account data (15 accounts total):
Credit cards (12): C01: Used=0 Limit=270,000 Status=Settled; C02: Used=38,000 Limit=40,000 Status=Normal; ...
Quasi-credit cards (3): Z01: Overdraft=56,000 Limit=80,000 Status=Normal; ...

`[Question]`
Q1: Should C01 (Status=Settled) be included in the calculation?
Q2: If C12's credit limit is reduced from 230,000 to 70,000, the utilization rate will: A. Increase B. Decrease C. No change

Answer each sub-question. Write on the last line: `[[ Answer ]] Q1=xxx; Q2=xxx; . . .`

Figure 21: CLR (Credit Logical Reasoning) evaluation prompt. The shared system prompt (top) defines the role and five answer formats. Four question types follow: **(1)** Single choice tests conceptual understanding of aggregation logic; **(2)** Multiple choice tests field-level knowledge of liability metrics; **(3)** Numerical requires multi-step calculation over real account data; **(4)** Compound combines yes/no reasoning with sensitivity analysis.

Table 30: Full LLM dependency audit of the MMFCTUB construction pipeline. Only the table content population stage (Layer 1, marked ✓) involves LLM calls; all evaluation-critical components—question formulation, gold-answer computation, distractor generation, and scoring—are fully deterministic and LLM-independent.[†]

Phase	**Component**	**Generation Method**	**LLM**
Data Construction	User financial profiles (L0)	Conditional sampling from statistical distributions (age, income, credit rating, account counts, etc.)	✗
	Residential & employment (L1a)	LLM-generated structured JSON; validated against fixed schema with enumerated value constraints	✓
	Account details (L1b)[‡]	LLM-generated structured JSON; 11–12 hard inequality constraints (e.g., `balance` $\leq$ `credit_limit`) enforced via deterministic post-processing	✓
	Auxiliary records (L1c)[§]	LLM-generated structured JSON; required-field schema with format validation	✓
	Repayment history (L2)	Algorithmic state-machine generation from L1b metadata	✗
	Summary & aggregate tables (L2)	Deterministic aggregation from L1b and L2 data	✗
Table Rendering	LaTeX source → PNG images	Template-based LaTeX construction; `xelatex` compilation; PyMuPDF rasterization	✗
Q&A Construction	Question text	Python template strings with variable substitution	✗
	Gold answers	Deterministic calculation on structured table data	✗
	Distractor options	Answer-pool sampling + numerical perturbation (near-miss, digit-swap, cross-swap)	✗
Evaluation Scoring	Answer extraction	Regex-based tag parsing	✗
	Accuracy computation	Exact match / set match / numeric tolerance	✗

Account (Credit Agreement ID: ZDJK20250015)			
Managing Institution	Opening Date	Account Credit Limit	Loan Amount
Commercial Bank PD	2020.08.15	2849450	2800000
Account Type	Currency	Business Type	Guarantee Type
Non-revolving Loan Account	CNY	Personal Commercial Housing Loan	Mortgage
Account Status	Five-tier Classification	Payment Due Date	Monthly Payment Due
Normal	N/A	2023.12.25	15800
Actual Monthly Payment	Last Payment Date	Total Payment Terms	Remaining Payment Terms
15800	2023.12.25	360	318
Maturity Date	Current Overdue Periods	Current Overdue Amount	As of Date
2050.07.31	0	0	Aug 15, 2050
Principal Overdue 31-60 Days		Principal Overdue 61-90 Days	
0		0	
Balance: 2624219			
Bad Debt Periods: N/A			

Repayment History | As of 2023-12												
	1	2	3	4	5	6	7	8	9	10	11	12
2023	N	N	N	N	N	N	N	N	N	N	N	N
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2022	N	N	N	N	N	N	N	N	N	N	N	N
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2021	N	N	N	N	N	N	N	N	N	N	N	N
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2020	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N	N	N	N
	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00	0.00	0.00	0.00
Account Closure Date: N/A												

Table 31: Credit Account Table Details

Account (Credit Agreement ID: ZDJK20250015)			
Managing Institution	Opening Date	Account Credit Limit	Loan Amount
Commercial Bank XH	2022.05.20	279066	180000
Account Type	Currency	Business Type	Guarantee Type
Sub-account under Revolving Credit	CNY	Personal Consumer Loan	Unsecured
Account Status	Five-tier Classification	Payment Due Date	Monthly Payment Due
Normal	N/A	2023.12.18	5580
Actual Monthly Payment	Last Payment Date	Total Payment Terms	Remaining Payment Terms
5580	2023.12.18	36	16
Maturity Date	Current Overdue Periods	Current Overdue Amount	As of Date
2025.04.30	0	0	May 20, 2025
Principal Overdue 31-60 Days		Principal Overdue 61-90 Days	
0		0	
Balance: 91958			
Bad Debt Periods: N/A			

Repayment History | As of 2023-12												
	1	2	3	4	5	6	7	8	9	10	11	12
2023	N	N	N	N	N	N	N	N	N	N	N	N
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2022	N/A	N/A	N/A	N/A	N/A	N	N	N	N	N	N	N
	N/A	N/A	N/A	N/A	N/A	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Account Closure Date: N/A												

Table 32: Credit Account Details

Occupation Information (ID: 1)			
ID	1	Company	Zhengzhou Shunda Logistics Co., Ltd.
Company Nature	Private Enterprise	Company Address	No. 12, Area B, Logistics Park, Intersection of West 3rd Ring Road and Longhai Road, Zhongyuan District, Zhengzhou City, Henan Province
Company Phone	0371—67123456	Occupation	Commerce and Service Personnel
Industry	Transportation, Storage and Postal Services	Position	General Staff
Title	None	Entry Year	2021
Update Date	2023.08.15		

Table 33: Occupation Information Table Details

Account (Credit Agreement ID: ZDJK20250015)			
Managing Institution	Opening Date	Account Credit Limit	Loan Amount
Commercial Bank ZJ	2021.11.10	300000	238907
Account Type	Currency	Business Type	Guarantee Type
Revolving Loan Account	CNY	Personal Consumer Loan	Unsecured
Account Status	Five-tier Classification	Payment Due Date	Monthly Payment Due
Normal	N/A	2023.12.21	1200
Actual Monthly Payment	Last Payment Date	Total Payment Terms	Remaining Payment Terms
1200	2023.12.21	N/A	N/A
Maturity Date	Current Overdue Periods	Current Overdue Amount	As of Date
2024.03.21	0	0	Nov 10, 2026
Principal Overdue 31-60 Days		Principal Overdue 61-90 Days	
0		0	
Balance: 0			
Bad Debt Periods: N/A			

Repayment History | As of 2023-12												
	1	2	3	4	5	6	7	8	9	10	11	12
2023	N	N	N	N	N	N	N	N	N	N	N	N
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2022	N	N	N	N	N	N	N	N	N	N	N	N
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2021	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N
	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00
Account Closure Date: N/A												

Table 34: Revolving Credit Account Details

Residence Information				
ID	Address	Phone	Status	Update Date
1	Room 502, Unit 2, Building 3, No. XXX Zhengzhou City	0371—67845621	Rented	2023.08.15

Table 35: Residence Information Table Details