Credit Cards, Confusion, Computation, and Consequences: How Well Do LLMs Reason About Financial Literacy?

Arnav Hiray✉ 🐝, Agam Shah✉ 🐝, Caleb Lu† 🐝, Meghaj Tarte† 🐝
Sreya Tummalapelly♠, Harsit Mittal🐝, Sudheer Chava🐝
✉ Equal first authors: {ahiray3, ashah482}@gatech.edu
✉ and † Indicate Core Contributors
🐝 Georgia Institute of Technology ♠ Stony Point High School

Abstract

We introduce **FinLitQA**, the first benchmark of long-context financial numerical reasoning questions derived directly from real credit card agreements. The dataset contains over 4,300 questions, including first-person variants that reflect how consumers naturally ask about fees and payments. Evaluating multiple large language reasoning models under Chain-of-Thought (CoT) and Program-of-Thought (PoT) prompting, we find that first-person CoT improves accuracy, while PoT increases answer coverage but reduces accuracy. Our error analysis highlights weaknesses with compounding using the exponential operator, understanding interest, timeline ordering, categorical policy logic, and minimum payment rules. We find that these errors often arise in edge cases such as late-payment penalties or small-balance scenarios *that are more likely to affect lower-income or financially vulnerable individuals*.

1 Introduction

The advancements of Large Language Models (LLMs) have unlocked significant opportunities for tackling knowledge and reasoning-intensive tasks [14]. Such tasks include coding [16, 22, 18], mathematics [10, 7, 25, 31], logic [24, 11] and beyond. Existing math reasoning benchmarks have shown that LLMs can achieve impressive results, but they also highlight important limitations. Studies suggest that model performance in mathematical reasoning is often tied to the kinds of problems and data distributions the models were trained on, which means their abilities do not always generalize well across different types of reasoning tasks [13]. Moreover, models exhibit memorization-like behavior, where they ignore modified assumptions and apply techniques from the original problem, or even reproduce the outcome of the unmodified version. Together, these findings raise concerns that current benchmarks may overstate true reasoning ability [15].

These limitations are particularly relevant in the financial domain, which demands interpretability, accuracy, and high-stakes decision making. Here, reasoning often requires multi-step calculations, frequently updated assumptions, and specialized terms and rules that are unlikely to be well represented in pretraining data. The current body of work for financial reasoning [29, 4, 33, 32, 27, 23, 34, 35, 17, 5, 28] has revealed insights into LLM bottlenecks with respect to limited business and financial understanding, tradeoffs with respect to cost and performance efficiency given model size, and the presence of numerical calculation errors. These state-of-the-art benchmarks in financial reasoning typically draw on questions and problems from financial textbooks, industry

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

financial documents such as analyst and published company reports, and professional practice exams such as those used in CFA or accounting training.

While this line of research has been valuable, a critical frontier remains underexplored and largely absent from focus: *harnessing LLMs to promote financial literacy*. Low literacy is consistently linked to excessive debt [19], bankruptcy [2], and inadequate saving for retirement or emergencies [20]. At the macro-level, it fuels wealth inequality and weakens economic resilience. Perhaps nowhere else is the financial literacy crisis more visible than in the education surrounding credit cards. Credit card debt in the United States has surpassed $1.2 trillion [1]. Low-literacy cardholders are disproportionately caught in debt spirals, where missed payments lead to mounting interest and a cycle of borrowing they cannot escape [8]. A significant factor behind this information gap is the complexity of credit card agreements, which set out the terms and conditions of a credit card, including how interest rates, grace periods, and penalties are calculated. Credit card agreements are many pages long and written at a reading level well beyond the average U.S. consumer [26].

There is a clear need for tools that make financial understanding accessible, and recent advances in LLMs and Generative AI provide a unique opportunity to transform financial knowledge into personalized, widely accessible education [12]. However, given the high cost of incorrect financial advice and decision-making, the first step is to rigorously evaluate LLMs' reasoning ability in the unique context of financial literacy. As such, to the best of our knowledge, we present the first financial numerical reasoning benchmark designed to evaluate LLMs' ability to reason through credit card agreements and credit-related financial questions. Our contribution is as follows:

- We create the first dataset of long-context financial numerical reasoning questions derived directly from the terms and conditions of real credit card agreements.
- A collection of first-person financial reasoning questions, attempting to replicate how individuals naturally frame financial questions.
- We provide evidence that LLMs struggle with context-dependent conditions and reasoning, revealing failures that mirror broader limitations in general mathematical reasoning.

2 Dataset Overview

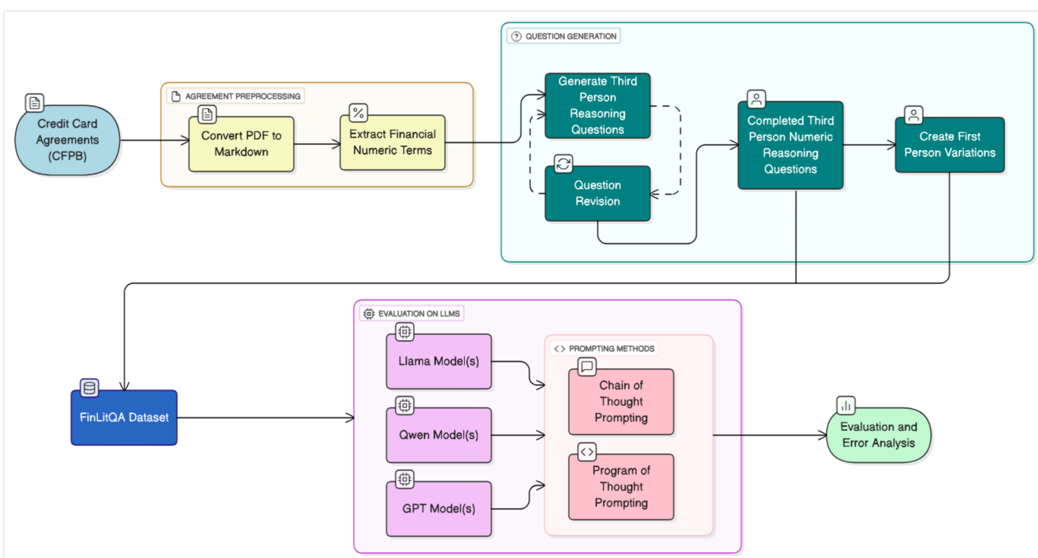

Figure 1: Overview of the FinLitQA construction and evaluation stages, including data preprocessing, question generation, and model evaluation.

Selecting Credit Card Agreements We collected 27 credit card agreements from the Consumer Financial Protection Bureau's database [6], covering major issuers such as American Express and Discover. We selected agreements from Q4 2024, as terms and conditions can change, and focused on the highest-volume products in the credit card market. For each provider, we included agreements from three categories: premium/elite, everyday retail, and subprime cards. This was done to capture heterogeneity in real-world usage.

In the CFPB database, credit card agreements are provided in PDF format and often include images, tables, and text. However, to make them accessible for LLMs, we converted all agreements into Markdown format. Additionally, to ensure fairness, the question generation process relied solely on these Markdown versions so that no information outside the models' accessible context was used.

Dataset Description We then created 3062 questions spread roughly evenly across the credit cards. Before beginning the question generation process, all annotators were required to familiarize themselves with both the agreements and the relevant financial terminology. Each annotator then read through their assigned agreement and identified financial numeric terms, defined as terms involving a percentage, dollar value, or unit of time, from which questions could be constructed. Questions were generated with respect to these terms, and in many cases involve multiple terms.

Table 1: Statistics of Credit Card Agreements and FinLitQA Dataset.

Credit Card Agreements	
# of Financial Numeric Terms	280
# of Words (Med / Avg)	9115 / 8830
# of Percentages (Med / Avg)	20 / 22.6
# of Dollar Amounts (Med / Avg)	15.5 / 19.25
# of Bullet Points (Med / Avg)	55.5 / 68.6
Flesch–Kincaid Readability Score (Min / Avg)	9.2 / 11.3
FinLitQA Dataset	
Number of Third Person Questions	3062
Question Length (Med / Avg)	30 / 33.4
Number of First Person Questions	1250
First Person Question Length (Med / Avg)	28.0 / 31.6
# of Math Operations Required (Med / Avg)	3 / 3.4

Each question is defined by its input(s), the number of mathematical operations required (as defined by [5]), and its output. The inputs and outputs fall into one of four categories: dollar values, units of time, percentages, or binary/categories (e.g., Yes/No, In Default/Not in Default). Inputs are a combination of user-provided values and information retrieved from the credit card agreement. Appendix A shows the information for a particular question. Before beginning the question generation process, all annotators were required to familiarize themselves with both the agreements and the relevant financial terminology. Data quality was ensured by having multiple annotators review each question's mathematical operators and reasoning steps.

In addition, we include 1,250 questions written in the first person. These were created by manually converting the previously created third-person questions into first-person form. Table 1 summarizes key statistics of the dataset and the underlying credit card agreements. One notable observation is the low Flesch–Kincaid readability scores of these agreements, which correspond to reading levels typically associated with college graduates or professionals [30].

3 Experiments

We evaluate our approach using four large language models of varying sizes and architectures: Llama-3.3-70B, Qwen3-235B-A22B-Instruct-2507, gpt-oss-120b, and gpt-oss-20b. These models were selected to provide coverage across different model families (Llama, Qwen, and GPT-style open-source systems) and parameter scales, ranging from 20B to 235B parameters.

For prompting, we consider two widely used reasoning paradigms: Chain-of-Thought (CoT) (see Figure 2) and Program-of-Thought (PoT) prompting, as shown in Appendix B. We restrict outputs to 2,000 tokens, a limit that is relatively longer than those used in similar previous tasks.

4 Results

Table 2 shows the performance of the LLMs using CoT and PoT prompting methods, as well as CoT on the First Person dataset. There are a few interesting observations.

Using First Person We find that COT First Person consistently achieves higher accuracy and lower unanswered rates than standard COT by a large margin. One explanation is that first-person framing

COT Prompt

System Prompt:
You are a financial expert. You are expected to answer the given question based on the provided financial document context. Follow these instructions carefully:
1. First, think through the problem step by step, clearly documenting each necessary step of your reasoning.
2. After completing your analysis, provide a single final conclusion at the end.
3. The final answer can be either a numeric value or a Yes/No response, depending on the question type.
4. Important: Always place the final answer in the last line of your response, using this exact format:
`"Therefore, the answer is {{final answer}}."`
Make sure to include the curly brackets around the final answer.

User Prompt:
{Credit Card Agreement}
Question: {Question}
Let's think step by step to answer the given question.

Figure 2: Chain-of-Thought (COT) prompt template.

gives the model a stronger signal that the query is a direct user instruction. Because instruction-tuned datasets often contain first-person queries (e.g., "How much will I owe if...?"), this phrasing is closer to the distribution models are trained to follow. Recent work on alignment faking shows that large language models change their behavior based on subtle contextual cues about training or deployment [9, 21]. In the same way, our findings suggest that first-person framing could act as such a cue, leading models to provide more complete answers and achieve higher accuracy.

Table 2: Comparison of model performance across COT, COT First Person, and POT settings.

	CoT		**CoT First Person**		**PoT**	
Model	Acc.	% Unans.	Acc.	% Unans.	Acc.	% Unans.
Llama-3.3-70B-Instruct-Turbo	0.59	21.5	0.75	15.1	0.54	0.2
Qwen3-235B-A22B-Instruct-2507	0.64	28.9	0.79	14.2	0.56	0.7
gpt-oss-120b	0.69	26.7	0.83	27.4	0.49	0.4
gpt-oss-20b	0.65	9.5	0.79	7.8	0.48	1.5

CoT vs PoT We also observed that these models performed better with CoT than PoT, even though PoT almost always produced an answer, suggesting that the models are more effective at reasoning than at programming, under which circumstances PoT may not necessarily outperform [3]. However, nearly all questions were answered using PoT, suggesting that improving PoT may be a promising approach to reducing unanswered questions in reasoning tasks, but at the cost of hallucinations. Another perspective to consider however, is that retail investors may not run code on their own which could limit the efficacy of PoT.

Consistent Responses Table 3 highlights distinct agreement patterns across prompting methods. PoT achieves the highest unanimous agreement (48.5%) but also has over a third of the questions unanswered. CoT-FP, in contrast, substantially reduces unanswered cases (12.2%) but shows lower full agreement. This suggests differences in the robustness of reasoning across prompting methods. PoT leaves many questions unanswered, suggesting weaker reasoning generalizability, but shows more stable and consistent reasoning when it succeeds, while CoT-FP outperforms CoT in producing more reliable reasoning paths. All of this analysis should be interpreted with caution, as the models evaluated are reasoning-enhanced and not specifically tuned for PoT or programming tasks.

Table 3: Percentage of questions answered correctly by exactly k models, across prompting methods.

Prompting Method	1 Model	2 Models	3 Models	4 Models	0 Models
CoT	14.7%	20.0%	32.4%	32.9%	28.8 %
CoT-FP	10.1%	18%	33.2%	38.6%	12.16%
PoT	13.4%	16.8%	21.2%	48.5%	34.6%

4.1 Error Analysis

Taking a look at questions where models perform particularly poorly, a few common trends in incorrect and unanswered questions:

Struggles with Compounding: Models often confuse Annual Periodic Rate (APR), Daily Periodic Rate (DPR), and the way interest is compounded (daily, monthly, or continuously), or mix up simple and compound interest. These errors reflect both gaps in domain knowledge and flawed mathematical reasoning, leading to major mistakes in estimating how interest grows and when payments end.

Piecewise Rules & Thresholds: Models struggle with conditions like "whichever is greater," minimum charges, fee caps, or tiered penalties. For example, they may always apply 1% of the balance while ignoring a $5 minimum, or keep adding late fees past a $30 cap. In practice, this means they follow only one branch of a rule or overlook limits, producing incorrect calculations. These mistakes matter because such rules often govern late penalties or extra charges—costs that disproportionately impact people who are already economically vulnerable.

Timeline & Date Ordering: Returned payments, billing cycle closes, grace periods, and due dates require step by step reasoning. Models often misorder events or miscount days, for example by starting a grace period before the billing cycle ends, which leads to wrong payment windows and incorrect fee assessments. These errors are especially harmful in credit card settings because even a few days' difference can trigger costly penalties.

Categorical Logic: Some rules depend on account type, transaction category, or special conditions, which must be checked before any math is done. Models sometimes skip this step and go straight into calculation, applying the wrong rule. For example, they may charge a late fee on a Student Card despite a first-time waiver, or apply a foreign transaction fee to a cash advance even when the agreement excludes it.

Minimum Payments & Small-Balance Rules: Questions involving minimum payments and small balances are particularly tricky because they require switching between percent-of-balance formulas, fixed minimums, and other overriding conditions. Models often forget to apply the larger of a percentage versus a fixed amount, ignore clauses that impose a minimum interest charge, or mishandle very small balances where fixed fees dominate. These errors are especially impactful since minimum payments and small balances are common among financially vulnerable cardholders.

5 Conclusion and Further Workshop

We introduce the first dataset of long-context financial numerical reasoning questions derived directly from the terms and conditions of real credit card agreements. Our experiments show that chain-of-thought reasoning performs better when framed from a first-person perspective. At the same time, we find that models struggle with compounding and interest calculations, timeline and date ordering, categorical logic, and minimum payment or small-balance rules. These weaknesses are especially concerning because they disproportionately affect vulnerable cardholders, highlighting an important issue to address.

We plan to benchmark a broader set of models including both reasoning and code-enhanced models. We also aim to have more in depth error analysis to better understand which stage of the process creates errors (retrieval, domain knowledge, step-by-step reasoning, or numerical computation).

References

[1] Quarterly report on household debt and credit. Technical report, Federal Reserve Bank of New York, Center for Microeconomic Data, aug 2025. 2025: Q2, Released August 2025.

[2] Meta Brown, John Grigsby, Wilbert van der Klaauw, Jaya Wen, and Basit Zafar. Financial education and the debt behavior of the young. *The Review of Financial Studies*, 29(9):2490–2522, 2016.

[3] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

[4] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[5] Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[6] Consumer Financial Protection Bureau. Credit card agreements database, 2025. Accessed: 2025-09-25.

[7] Jingxuan Fan, Sarah Martinson, Erik Y. Wang, Kaylie Hausknecht, Jonah Brenner, Danxian Liu, Nianli Peng, Corey Wang, and Michael Brenner. HARDMATH: A benchmark dataset for challenging problems in applied mathematics. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024.

[8] Christos L. Giannikos and Eleni D. Korkou. Financial literacy and credit card payoff behaviors: Using generalized ordered logit and partial proportional odds models to measure american credit card holders' likelihood of repaying their credit cards. *International Journal of Financial Studies*, 13(1):22, 2025.

[9] Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models, 2024.

[10] Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. Putnam-AXIOM: A functional and static benchmark for measuring higher level mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024.

[11] Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. FOLIO: Natural language reasoning with first-order logic. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[12] Oudom Hean, Utsha Saha, and Binita Saha and. Can ai help with your personal finances? *Applied Economics*, 0(0):1–9, 2025.

[13] Zijin Hong, Hao Wu, Su Dong, Junnan Dong, Yilin Xiao, Yujing Zhang, Zhu Wang, Feiran Huang, Linyi Li, Hongxia Yang, and Xiao Huang. Benchmarking llms' mathematical reasoning with unseen random variables questions, 2025.

[14] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada, July 2023. Association for Computational Linguistics.

[15] Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. MATH-perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. In *Forty-second International Conference on Machine Learning*, 2025.

[16] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[17] Michael Krumdick, Rik Koncel-Kedziorski, Viet Dac Lai, Varshini Reddy, Charles Lovering, and Chris Tanner. BizBench: A quantitative reasoning benchmark for business and finance. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8309–8332, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[18] Chloe R Loughridge, Qinyi Sun, Seth Ahrenbach, Federico Cassano, Chuyue Sun, Ying Sheng, Anish Mudide, Md Rakib Hossain Misu, Nada Amin, and Max Tegmark. Dafnybench: A benchmark for formal software verification. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024.

[19] Annamaria Lusardi. Americans' financial capability. Working Paper 17103, National Bureau of Economic Research, June 2011.

[20] Annamaria Lusardi and Olivia S Mitchell. Financial literacy and retirement planning in the united states. Working Paper 17108, National Bureau of Economic Research, June 2011.

[21] Joe Needham, Giles Edkins, Govind Pimpale, Henning Bartsch, and Marius Hobbhahn. Large language models often know when they are being evaluated, 2025.

[22] Dung Manh Nguyen, Thang Chau Phan, Nam Le Hai, Tien-Thong Doan, Nam V. Nguyen, Quang Pham, and Nghi D. Q. Bui. CodeMMLU: A multi-task benchmark for assessing code understanding & reasoning capabilities of codeLLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

[23] Xueqing Peng, Lingfei Qian, Yan Wang, Ruoyu Xiang, Yueru He, Yang Ren, Mingyang Jiang, Jeff Zhao, Huan He, Yi Han, Yun Feng, Yuechen Jiang, Yupeng Cao, Haohang Li, Yangyang Yu, Xiaoyu Wang, Penglei Gao, Shengyuan Lin, Keyi Wang, Shanshan Yang, Yilun Zhao, Zhiwei Liu, Peng Lu, Jerry Huang, Suyuchen Wang, Triantafillos Papadopoulos, Polydoros Giannouris, Efstathia Soufleri, Nuo Chen, Guojun Xiong, Zhiyang Deng, Yijia Zhao, Mingquan Lin, Meikang Qiu, Kaleb E Smith, Arman Cohan, Xiao-Yang Liu, Jimin Huang, Alejandro Lopez-Lira, Xi Chen, Junichi Tsujii, Jian-Yun Nie, Sophia Ananiadou, and Qianqian Xie. Multifinben: A multilingual, multimodal, and difficulty-aware benchmark for financial llm evaluation, 2025.

[24] Chengwen Qi, Ren Ma, Bowen Li, He Du, Binyuan Hui, Jinwang Wu, Yuanjun Laili, and Conghui He. Large language models meet symbolic provers for logical reasoning evaluation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[25] Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, YiFan Zhang, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Xiao Zong, Yida Xu, Peiqing Yang, Zhimin Bao, Muxi Diao, Chen Li, and Honggang Zhang. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher

Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20023–20070, Vienna, Austria, July 2025. Association for Computational Linguistics.

[26] Daniel Schorn. Can't understand your credit card agreement? you're not alone, sep 2016. Accessed: 2025-09-24.

[27] Agam Shah and Sudheer Chava. Zero is not hero yet: Benchmarking zero-shot performance of llms for financial tasks, 2023.

[28] Agam Shah, Arnav Hiray, Pratvi Shah, Arkaprabha Banerjee, Anushka Singh, Dheeraj Deepak Eidnani, Sahasra Chava, Bhaskar Chaudhury, and Sudheer Chava. Numerical claim detection in finance: A new financial dataset, weak-supervision model, and market analysis. In Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos, editors, *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 170–185, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[29] Zichen Tang, Haihong E, Ziyan Ma, Haoyang He, Jiacheng Liu, Zhongjun Yang, Zihua Rong, Rongjin Li, Kun Ji, Qing Huang, Xinyang Hu, Yang Liu, and Qianhe Zheng. FinanceReasoning: Benchmarking financial numerical reasoning more credible, comprehensive and challenging. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15721–15749, Vienna, Austria, July 2025. Association for Computational Linguistics.

[30] Wikipedia contributors. Flesch–Kincaid readability tests — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests, 2025. [Online; accessed 25-September-2025].

[31] Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin E. Lauter, Swarat Chaudhuri, and Dawn Song. Position: Formal mathematical reasoning—a new frontier in AI. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025.

[32] Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. TaPERA: Enhancing faithfulness and interpretability in long-form table QA by content planning and execution-based reasoning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12824–12840, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[33] Yilun Zhao, Hongjun Liu, Yitao Long, Rui Zhang, Chen Zhao, and Arman Cohan. Financemath: Knowledge-intensive math reasoning in finance domains. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12841–12858, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[34] Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. DocMath-eval: Evaluating math reasoning capabilities of LLMs in understanding long and specialized documents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[35] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online, August 2021. Association for Computational Linguistics.

A Example Question

Table 4: Example of details created for each question.

Credit Card Agreement	Costco Anywhere Visa
Question	If I transfer a debt of $281 to this card at the start of the month, will I ever be able to pay off the balance at $5 per month?
Numerical Inputs	Balance Transfer APR: 20.49% Balance Transfer Fee: $5 or 5%
Answer	False
Output Type	Category
Input Types	Dollar Value, Percentage
Operator Arguments	#1 = 281, #2 = 5, #3 = 20.49, #4 = 5, #5 = 5 #6 = divide(#5, 100), #7 = add(1, #6), #8 = multiply(#1, #7) #9 = add(#1, #4), #10 = max(#9, #8) #11 = divide(#3, 36500), #12 = add(1, #11) #13 = exponent(#12, 30), #14 = subtract(#13, 1) #15 = multiply(#10, #14), #16 = greater_than(#2, #15)

B POT Prompt

POT Prompt

System Prompt:
You are a financial expert. You are supposed to generate a Python program to answer the given question. The returned value of the program is supposed to be the answer. Here is an example of the Python program:

```
def solution():
    # Define variables name and value based on the given context
    ...
    # Do math calculation to get the answer
    ...
    # return answer
    return answer
```

User Prompt:
{Credit Card Agreement}
Question: {Question}
Continue the program to answer the question. The returned value of the program is supposed to be the answer:

```
def solution():
    # Define variables name and value based on the given context
```

C Examples of Incorrect Prompt Outputs

Question: A cardholder has the lowest cash advance ATM fee in his credit card agreement. With a $250 cash advance once a week, how many weeks will it take to owe at least 25% of the original cash advance total in fees?

Model Output

To solve this problem, we need to understand the cash advance ATM fee structure and calculate how many weeks it will take to accumulate at least 25% of the original cash advance total in fees.

1. **Identify the cash advance ATM fee**: According to the provided credit card agreement, the cash advance ATM fee is "Either $5 to $10 or 4% to 5% of the Advance amount, whichever is greater (maximum fee (No Maximum))". . . .

2. **Calculate the weekly fee**: With a $250 cash advance, if we apply the 4% fee (which is $10), it's higher than the $5 minimum . . .

⋮

6. **Find when the total fee is at least 25% of the total cash advance**: We want to find 'n' when the total fee is at least 25% of the total cash advance. This can be represented as $10n $\geq$ 0.25 * $250n.

7. **Solve for 'n'**: Simplify the inequality $10n $\geq$ 0.25 * $250n to find 'n'. - $10n $\geq$ $62.5n - Since $10n is the fee and $62.5n represents 25% of the total cash advance, we actually need to reconsider our approach because . . .

⋮

8. **Correct Approach**: The correct approach is to understand that 25% of the total cash advance is the target for the total fees. Thus, we should . . . However, this step was incorrectly approached. Let's correct the calculation by directly addressing the question: How many weeks will it take for the fees to accumulate to at least 25% of the total cash advances made?

⋮

Correct Calculation: To find when the total fees equal at least 25% of the total cash advances, we should consider the ratio of fees to advances and solve for 'n' correctly.

⋮

Final Calculation: The error in the previous steps was in incorrectly setting up the equation to solve for 'n'. The correct way to approach this problem is . . .

⋮

Correct Insight: The total fee per week is $10, and the total advance per week is $250. The fee is 4% of the advance. We are looking for how many weeks ('n') it takes for the total fees to accumulate to at least 25% of the total advances.

⋮

Actual Calculation Needed: Given that each week the cardholder pays a $10 fee on a $250 cash advance, and we want to find out after how many weeks the total fees will be at least 25% of the total cash advances, let's correct the calculation:

⋮

Final Step: To solve for 'n', we recognize that the total fees ($10n) need to be at least 25% of the total advances ($62.5n, since **end of output**

The error here is that the model sets up the inequality ($10n \geq 0.25 * 250n$), which simplifies to ($10n \geq 62.5n$), a condition that can never be true—so instead of concluding "no solution," it keeps re-deriving the same step in a loop.

Question Let's say our APR is 21.99%, then what is the daily periodic rate and how much interest is charged in 5 days on a $2,500 balance if no payments are made?

Model Output

To find the daily periodic rate and calculate the interest charged in 5 days on a \$2,500 balance with an APR of 21.99%, we'll follow these steps:

1. Calculate the Daily Periodic Rate (DPR):

- The formula to calculate the DPR from the APR is: DPR = APR / 365
- Given APR = 21.99%
- DPR = 21.99% / 365
- **DPR = 0.0602 or 6.02% per day (as a decimal, this is 0.0602)**

2. Calculate the Interest Charged in 5 Days:

- The formula to calculate interest is: Interest = Principal $\times$ Rate $\times$ Time
- Principal = \$2,500,   Rate = 0.0602,   Time = 5 days
- Interest = \$2,500 $\times$ 0.0602 $\times$ 5
- Interest = \$2,500 $\times$ 0.301
- Interest $\approx$ \$752.50

However, the interest calculation provided above assumes a simple interest calculation over the period, which might not exactly match how credit card interest is compounded daily. The correct approach for calculating interest on a credit card balance, considering daily compounding, involves calculating the daily interest and then compounding it over the period.

3. Daily Compounding Interest Calculation:

- Daily Interest = \$2,500 $\times$ (21.99% / 365)
- Daily Interest $\approx$ \$2,500 $\times$ 0.0602
- Daily Interest $\approx$ \$150.50
- For 5 days, without compounding: \$150.50 $\times$ 5 = \$752.50
- With compounding, each day's balance increases and affects the next day's interest.

⋮

Therefore, the answer is \$758.73.

The error is that the daily periodic rate was miscalculated as **0.0602 (6.02%)** instead of the correct **0.000602 (0.0602%)**, which inflated the interest by a factor of 100.

Question: Using the Visa Mastercard Platinum Card, I have a $1,500 balance that I forgot to pay. As a result, a penalty APR imposed by the company of 27.99% is incurred. How much would I pay each month if I wanted to pay off the balance in 10 months of equal payments?

Model Output

To calculate the monthly payment needed to pay off the balance in 10 months, we first need to understand the interest charged and how it affects the total amount to be paid.
Given: - Initial balance: $1,500 - Penalty APR: 27.99% - Time to pay off the balance: 10 months
The penalty APR is an annual rate, so we first convert it to a monthly rate because we're making monthly payments.

$$\text{Monthly APR} = \frac{27.99\%}{12} = \frac{0.2799}{12} \approx 0.023325 \text{ or } 2.3325\% \text{ per month}$$

To find the monthly payment (M) that will pay off the loan in 10 months, we can use the formula for monthly payments on a loan, which is:

$$M = P\left[\frac{r(1+r)^n}{(1+r)^n - 1}\right]$$

where: - M = monthly payment - P = principal loan amount (the initial balance) = $1,500 - r = monthly interest rate = 2.3325% = 0.023325 - n = number of payments = 10 months
Plugging the values into the formula:

$$M = 1500\left[\frac{0.023325(1+0.023325)^{10}}{(1+0.023325)^{10} - 1}\right]$$

Calculating the values inside the brackets first:

$$(1+0.023325)^{10} \approx 1.2634$$

$$0.023325 \times 1.2634 \approx 0.02945$$

$$1.2634 - 1 = 0.2634$$

$$\frac{0.02945}{0.2634} \approx 0.1118$$

Then, multiplying by P:

$$M \approx 1500 \times 0.1118 \approx 167.7$$

So, the monthly payment needed to pay off the $1,500 balance in 10 months with a penalty APR of 27.99% would be approximately $167.70.
Therefore, the answer is $167.70.

The error here is caused by rounding approximations in the exponential calculations and subsequent multiplications/divisions, which affect the final monthly payment.