Contents lists available at ScienceDirect

Journal of Behavioral and Experimental Finance

journal homepage: www.elsevier.com/locate/jbef

Full length article

Reasoning with financial regulatory texts via Large Language Models

Bledar Fazlija [a,*], Meriton Ibraimi [b], Aynaz Forouzandeh [c], Arber Fazlija [d]

[a] *School of Management and Law, ZHAW Zurich University of Applied Sciences, Zurich, Switzerland*
[b] *Independent*
[c] *Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran*
[d] *ZHAW Zurich University of Applied Sciences, Zurich, Switzerland*

ARTICLE INFO

Keywords:
Financial regulation
Compliance
Basel III
Large Language Models
Machine learning
Code generation
Generative AI
Artificial intelligence
Responsible AI
Chain-of-Thought
Tree-of-Thought

ABSTRACT

Interpreting complex financial regulatory texts, such as the Basel III Accords, can be challenging even for human experts. In this paper, we explore the potential of Large Language Models (LLMs) to perform such tasks. Specifically, we evaluate reasoning strategies, namely Chain-of-Thought (CoT) and Tree-of-Thought (ToT), in their ability to assign accurate risk weights to test cases based on the Basel III Standardized Approach (SA) for Credit Risk. Moreover, we propose and test a guided learning-based few-shot variant of CoT and ToT using human expert input. By evaluating 6,501 test cases, comprised of diverse exposure scenarios, our results demonstrate that few-shot prompting with CoT as well as ToT significantly enhances the LLMs' accuracy in inferring risk weights. For one-shot CoT, we observe gains of almost 13 percentage points in accuracy with GPT-4o, whereas Claude 3 Sonnet shows gains of more than 10 percentage points. Albeit smaller in magnitude, one-shot ToT improvements are around 9 percentage points.

1. Introduction

Over the past few decades, keeping up with financial regulation has become a major hurdle for financial institutions. Take the Basel Accords as an example: The first version of the Basel Accords, Basel I, fit into about 30 pages, Basel II multiplied to about 300, and Basel III now stretches to nearly 1900 pages (The Basel Framework, 2024). Each new layer of the Basel Accords is meant to strengthen the financial system (Levine, 2012; Tarullo, 2019; The Basel Framework, 2024), but the sheer volume makes compliance departments' work tough and translates in high expenditure cost for banks (Fazlija et al., 2024). As highlighted by EFD (2023), Swiss banks alone spent an estimated CHF 720 million for the finalization of Basel III ("Basel IV"). And it is not only the Basel Accords: Thomson Reuters (2023) shows that in 2022 regulators around the word issued more than 61,000 regulatory events. This makes regulatory compliance a critical and resource-intensive challenge for financial institutions.

Parsing those dense documents is still largely a manual and time intense process. Traditional rule-based systems struggle with the nuanced language and context of lengthy policies. As stated by De Lucio and Mora-Sanguinetti (2021), understanding the economic cost of regulatory complexity may require new text-mining tools. That is were recent breakthroughs in artificial intelligence, especially in natural language processing (NLP) and large language models (LLMs), come in.

AI's growing influence is hard to miss: The 2024 Nobel Prizes in Physics and Chemistry honored core advances in deep learning and the development of AlphaFold2, an AI system that cracked the protein-folding problem. Goodell et al. (2021) demonstrates that AI has already found its way into finance, including risk assessment and regulatory compliance. AI has already transformed areas such as stock behavior prediction (Gu et al., 2020; Rossi, 2018; Zhong and Enke, 2019), derivative hedging (Becker et al., 2020; Buehler et al., 2019), robo-advisors (Shanmuganathan, 2020), fraud detection (Dornadula and Geetha, 2019; Gregory and Vito, 2024; Aros et al., 2024; Xu et al., 2024; Zhou et al., 2023), and sustainable finance (Bingler et al., 2022; Webersinke et al., 2021). However, the existing literature deals mostly with structured problems like flagging suspicious transactions or classifying texts by risk. Minor attention has been given to AI that can reason through the content of regulation itself. The emergence of LLMs opens new possibilities for tackling regulatory challenges, where deep understanding and precise reasoning over complex texts are crucial.

Recent research has explored using LLMs for regulatory applications. Micheler and Whaley (2020) discuss the potential of AI to automate compliance by replacing rules written in natural legal language

* Corresponding author.
E-mail addresses: bledar.fazlija@zhaw.ch (B. Fazlija), meriton.ibraimi85@gmail.com (M. Ibraimi), a.forouzandeh@tabrizu.ac.ir (A. Forouzandeh), arber.fazlija@zhaw.ch (A. Fazlija).

https://doi.org/10.1016/j.jbef.2025.101067
Received 24 February 2025; Received in revised form 30 May 2025; Accepted 30 May 2025
Available online 26 June 2025

by computer code. Achitouv et al. (2023) use ten relatively small language models, such as *bert-base-nli-mean-tokens* and *all-MiniLM-L12-v2*, to match regulations (Financial Conduct Authority (FCA) Rulebook 2022) to internal policies. Cao and Feinstein (2024) study the interpretation of Basel III capital requirements for Market Risk using LLM prompting techniques. Fazlija et al. (2024) focus on generating Python code from regulatory texts, demonstrating the feasibility of LLM-driven automation for regulatory implementation.

In this paper, we contribute to this emerging line of research by evaluating the capabilities of LLMs to *interpret* regulatory texts through a large set of test cases created in Fazlija et al. (2024) focused on the task of *assigning risk weights* under the Basel III Standardized Approach (SA) for credit risk. Under the SA, banks must maintain a minimum capital ratio (e.g., 8%) relative to their risk-weighted assets (RWA). RWAs are calculated by applying prescribed risk weights to credit exposures, depending on factors such as counterparty type, asset class, credit quality, maturity, and currency denomination.

For an illustration of the concept of risk weights and their calculation, we refer to the example provided in Fazlija et al. (2024).

The regulatory text specifies such risk weights in exhaustive detail, leading to high complexity due to the multitude of possible cases. We refer to each such exposure scenario and corresponding risk weight as a ***test case***. To evaluate LLM performance, we formulate a targeted regulatory reasoning task: for a given credit belonging to a specified exposure class, we provide the LLM with the relevant regulatory articles that govern the risk weights for that class, and ask the LLM to determine the correct risk weight based solely on the provided text. This setup closely mirrors real-world compliance workflows, where practitioners must apply specific regulatory rules to individual credit exposures.

The two major prompting strategies that we use are:

- **Zero-shot prompting**, where the LLM receives only the task instruction.
- **Few-shot prompting**, where the LLM is provided with illustrative examples of correct reasoning steps.

To enhance reasoning abilities for complex tasks, prompting techniques such as Chain-of-Thought (CoT; Wei et al., 2022) and Tree-of-Thought (ToT; Long, 2023; Yao et al., 2024) have been developed. These methods encourage large language models (LLMs) to reason step-by-step or in a branching manner, improving their performance on symbolic, multi-step problems. Sprague et al. (2024) show that CoT is particularly helpful for mathematical and logical tasks, while being less beneficial for others. However, providing examples for few-shot prompting for reasoning strategies like CoT or ToT remains time-consuming, especially when high-quality, fully accurate rationales are required. While methods for automatic generation of CoT rationales – such as Auto-CoT (Zhang et al., 2022) – do exist, they are not directly applicable to the regulatory context considered here. Simple examples are unlikely to help a language model infer risk weights for test case inputs based on complex regulatory texts. Additionally, Auto-CoT's clustering-based approach is not useful in this setting, as the test cases are already nested into 13 exposure classes.

In this paper, we propose a simple yet effective few-shot prompting variant for CoT and ToT methods, specifically tailored for regulatory text interpretation. This setup can be illustrated with the following illustrative example (for a concrete example in the zero-shot setting, see Appendix B.) (see Fig. 1):

Our work provides new insights into the capabilities and limitations of LLMs for high-stakes regulatory interpretation tasks and contributes to building more automated and reliable compliance solutions for the banking industry.

The remainder of the paper is structured as follows: Section 2 describes the methodology, including the dataset, the construction of test cases, and the CoT and ToT reasoning-enhancing methods with their zero-shot and few-shot variants. Section 3 presents the empirical results and compares the accuracy of different prompting strategies across models and exposure classes. Section 4 discusses the implications of the findings, limitations, and potential directions for future research. Appendices provide detailed prompts and additional technical information used in the experiments.

2. Methodology

Fig. 2 provides an overview of our methodology. We employ GPT-4o and Claude 3 Sonnet, along with the reasoning-enhancing methods CoT and ToT, for which we also propose few-shot variants and evaluate their effectiveness in assigning correct risk weights based on given inputs.

We begin by converting test cases (see Section 2.1) from the Basel III dataset provided by Fazlija et al. (2024), into a format compatible with LLMs. Subsequently, we formulate prompts for zero-shot and examples from the given data for few-shot learning for CoT and ToT. Each test case, along with its corresponding articles and assumptions, is integrated into the prompts to determine the risk weight. Finally, we assess the accuracy of the results produced by each reasoning method.

We restrict ourselves to one-shot prompting for CoT and ToT. For the one-shot example, we ask a human subject matter expert (one of the co-authors) to provide, for each exposure class, based on the texts only, the test case perceived as most difficult. Using the corresponding texts and zero-shot variants of CoT and ToT, we generate rationales and check them manually for correctness, thus getting pairs of texts and corresponding rationales, which are then used as examples for the one-shot prompting. This method also works for k-shot learning, with $k > 1$, demanding more human expert effort.

2.1. Test cases

To create the list of compatible test cases, we utilize the Basel III-based dataset by Fazlija et al. (2024). This dataset contains 13 exposure classes related to the Standardized Approach: individual exposures approach. Each exposure class in the dataset includes:

1. Articles relevant to that exposure class
2. A ground truth code (GTC) for determining the risk weight
3. Test cases for the GTC
4. The assumptions underlying the development of the GTC.

In total, the dataset contains 6501 test cases providing a comprehensive foundation for evaluating risk weights across various exposure scenarios. The test cases in the dataset are categorized into two types:

- **Valid test cases:** These test cases consist of combinations of valid values for the input variables, with the corresponding risk weight as the output.
- **Invalid test cases:** These test cases include combinations of both valid and invalid values for the input variables, which may result in an "Invalid input value!" output.

Fazlija et al. (2024) designed test cases for each exposure class through a structured approach. They identified the variables influencing risk weight assignment, defined their possible values, and generated test cases by combining these values. The next step involved creating GTC to assign risk weights to the test cases based on regulatory requirements, followed by an expert review to verify the accuracy of the assigned risk weights. Subsection text.

The available test cases are structured for use in Python code. However, since these test cases need to be incorporated into a prompt, we convert them into a text format by listing each variable name alongside its corresponding value. Since the goal is to infer the risk weight using LLMs, we do not consider the risk weight part of the test case here. For example, in 'Exposures to retail' class, converted valid test cases (non-exhaustive) are shown below:

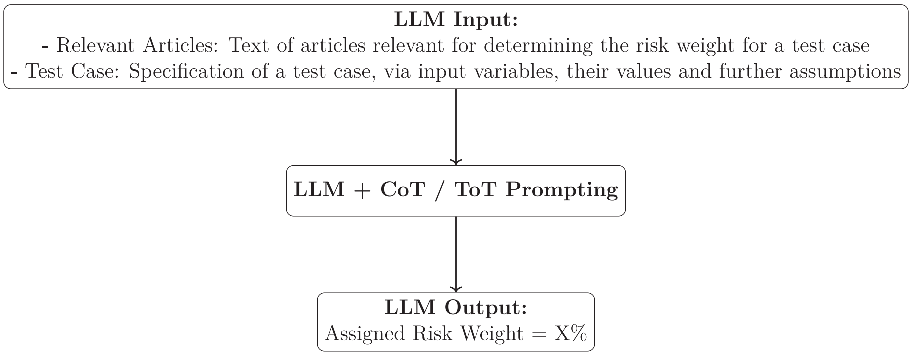
LLM Input:
- Relevant Articles: Text of articles relevant for determining the risk weight for a test case
- Test Case: Specification of a test case, via input variables, their values and further assumptions
LLM + CoT / ToT Prompting
LLM Output:
Assigned Risk Weight = X%

Fig. 1. Example of risk weight assignment task for a specific credit exposure.

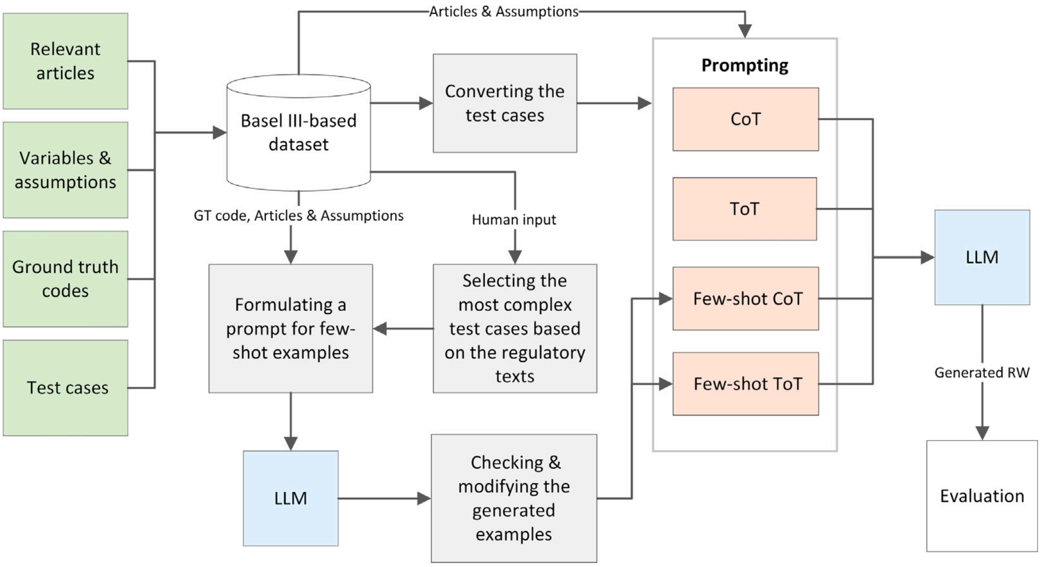
Relevant articles
Variables & assumptions
Ground truth codes
Test cases
Basel III-based dataset
Articles & Assumptions
Converting the test cases
Prompting
CoT
ToT
Few-shot CoT
Few-shot ToT
GT code, Articles & Assumptions
Human input
Formulating a prompt for few-shot examples
Selecting the most complex test cases based on the regulatory texts
LLM
Checking & modifying the generated examples
LLM
Generated RW
Evaluation

Fig. 2. Overview of the methodology. Test cases are first converted into a usable format. Each test case, together with the articles and assumptions for the corresponding exposure class, is used in CoT and ToT prompts to generate the risk weight. For few-shot prompting, one example per exposure class is generated by a human expert, based on the most complex test case. A prompt is then created to generate reasoning steps, which are manually refined and incorporated into the few-shot CoT and ToT prompts.

Test case 1: *exposure_type=Regulatory retail, transactor=True*
Test case 2: *exposure_type=Regulatory retail, transactor=False*
Test case 3: *exposure_type=Other, transactor=True*
Test case 4: *exposure_type=Other, transactor=False*

Here, the variable *'exposure_type'* can take the values *'Regulatory retail'* or *'Other'*, while *'transactor'* can be either *'True'* or *'False'*. These test cases are then integrated into the prompt, where a LLM is tasked to determine the correct risk weight, as shown in the next section.

2.2. Reasoning methods

As mentioned, we utilize zero-shot and few-shot CoT and ToT reasoning methods. Below, we provide a detailed description of each reasoning method and the structure of the formulated prompts.

Chain-of-thought (CoT): For CoT reasoning, the LLM is prompted to mimic a human problem-solving process of solving a task by splitting it in intermediate reasoning steps:

- **Zero-shot CoT:** In this approach, the model is instructed to perform intermediate reasoning steps without any examples. This is achieved by including a simple instruction in the prompt, such as "Let's think step by step". Appendix A shows our prompt for this approach.
- **Few-shot CoT:** This method involves providing the model with a few examples of CoT reasoning (Appendix C). The examples "guide" the model to learn and replicate the reasoning process when completing the given task. We use human expert input to find the most informative examples by selecting the most complex test cases from each exposure class to perform one-shot CoT. To create these examples, we use articles, assumptions, and GTCs associated with that class, along with the selected test case and its expected output. This data is incorporated into a prompt (see Appendix F), and the language model is tasked with generating CoT reasoning steps to assign the appropriate risk weight to the given test case. Since the codes are more structured, the reasoning steps generated by the model are also more structured compared to the contextual information provided in the articles. The generated reasoning is then manually reviewed and adjusted as needed to ensure accuracy.

Tree-of-thought (ToT): ToT reasoning also incorporates intermediate steps; however, unlike the "one-directional" approach of CoT, it explores multiple branches of reasoning while self-evaluating progress. We use the method of Yao et al. (2024) and Hulbert (2023) proposing a single-sentence prompt approach to ToT, enabling the method to be executed with a single call to the model. Like for CoT, ToT can also be implemented in two ways:

Table 1
Accuracy results for zero-shot CoT, zero-shot ToT (referred to as CoT and ToT), and one-shot CoT, one-shot ToT reasoning methods. Empty cells are marked with "–" to indicate that data is not reported, as one-shot examples are applicable only to valid test cases.

Model	Test case	CoT	ToT	One-shot CoT	One-shot ToT
Claude 3 Sonnet	Valid	73.50%	65.15%	83.63%	74.88%
	Invalid	78.26%	71.73%	–	–
	Valid + Invalid	73.54%	65.20%	–	–
GPT-4o	Valid	80.55%	77.76%	**89.33%**	83.03%
	Invalid	84.78%	86.95%	–	–
	Valid + Invalid	80.58%	77.83%	–	–

- **Zero-shot ToT:** To formulate the prompt for zero-shot ToT, where no examples are required, we modify the task definition component of the CoT prompt while retaining the test case, assumptions, and articles components. The modified task definition prompt that incorporates ToT reasoning is presented in Appendix D.
- **Few-shot ToT:** In this approach, we provide a few examples of ToT reasonings (Appendix E). These examples are created in the same way as CoT examples, with the prompt modified to generate ToT reasoning instead of CoT (Appendix G). As such, articles, assumptions, and GTCs are utilized along with the complex example test case for each exposure class and its expected output.

2.3. Evaluation metric and statistical testing

Since we consider the output of the LLMs as a categorical outcome (i.e., whether the risk weight produced based on the exposure class articles is correctly inferred or not), we use accuracy as the evaluation metric:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\hat{y}_i = y_i\} \tag{1}$$

where:

- N is the total number of the considered test cases,
- $\hat{y}_i$ is the predicted risk weight by the LLM for the test case i,
- y_i is the correct risk weight (i.e., the ground truth) for the test case i,
- $\mathbf{1}\{\hat{y}_i = y_i\}$ is the indicator function, which equals 1 if $\hat{y}_i = y_i$, and 0 otherwise.

To statistically test the differences between the used models and prompting methods (zero-shot CoT, one-shot CoT, etc.), we apply the nonparametric Friedman's chi-square test (Friedman, 1940), which can be used for multiple comparisons with categorical outcomes.

Given k methods and n matched test cases, the Friedman statistic is computed as:

$$\chi^2_F = \frac{12n}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{2}$$

where R_j is the sum of ranks assigned to method j.

If the Friedman test results in a p-value < 0.01, post-hoc pairwise comparisons are performed using the Nemenyi test (Nemenyi, 1963).

3. Results

We used GPT-4o and Claude 3 Sonnet to infer risk weights from regulatory articles. For both models, the temperature was set to 1, and the maximum number of new tokens generated per request was limited to 4096.

The accuracy metric was used to evaluate the results. Accuracy was calculated separately for valid and invalid test cases, as well as for both cases combined, within each exposure class and across all exposure classes. For one-shot methods, accuracy is reported only for valid test cases, as the selected examples are derived from valid cases, and correct reasoning rationales are only applicable to valid test cases. Tables 1 and 2 show these results.

In Table 1, we observe clear differences in accuracies for the two models, with GPT-4o showing consistently superior results across all reasoning methods in both the zero-shot and one-shot contexts. The overall best result is achieved in the one-shot CoT version, with an accuracy of 89.33% by GPT-4o. GPT-4o consistently shows better results across all methods, with a maximum difference of more than 15 percentage points for the zero-shot ToT variant in invalid test cases and almost 13 percentage points overall. We also observe a difference of more than 8 percentage points in the one-shot variant of ToT.

Table 2 shows the results across the different models and reasoning methods for each exposure class separately. There are pronounced differences among specific exposure classes. For instance, we observe that some exposure classes (such as Classes 4, 5, and 6) are more difficult for the LLMs to infer risk weights for than the others. Baseline experiments, where we just asked the LLMs to output the risk weight for a given test case input (without providing the articles of the given exposure class), indicate that this might be due to semantic prior and the corresponding beliefs the LLMs have about that specific collection of articles, rather than a particular reasoning flaw that would be related to the logic of the articles (such as some complicated nested rules).

To test statistical differences, all models and reasoning-enhancing methods were applied to the same set of test cases, representing a repeated measures design. We used the Friedman chi-square test across all methods. This nonparametric test is appropriate for dependent samples, which, in our case, means that each test case was evaluated by all methods. Although some dependency likely exists between test cases within each exposure class — potentially violating the assumption of independence between test cases — the large number of test cases and the resulting effect size ($\chi^2_F = 2614.899$, $p < 0.01$) suggest statistically significant differences. Table 3 presents the pairwise comparisons using the Nemenyi post-hoc test.

From Tables 1, 2, and 3, we observe that CoT significantly outperforms ToT. This is surprising, as ToT is generally considered more advanced due to its ability to reason through multiple reasoning paths. The linear reasoning of CoT, however, appears to be more suitable for the logic of the regulatory articles involved.

4. Discussion

In this paper, we demonstrated that reasoning-enhancing methods like CoT and ToT yield strong results for interpreting financial regulatory texts. We proposed a few-shot variant of CoT and ToT that incorporates human expert input. These variants led to significant performance improvements relative to their zero-shot counterparts. Surprisingly, CoT consistently outperforms ToT in both the zero-shot and few-shot settings. The linear reasoning of CoT results in better performance, possibly due to the highly structured nature of the tasks at hand, while ToT's multiple reasoning paths may introduce unnecessary complexity and cause confusion.

For both models, providing one human-expert-selected test case per class yields statistically significant improvements, with CoT achieving greater gains—more than 12 percentage points for GPT-4o and more than 10 percentage points for Claude 3 Sonnet. Thus, we observe that providing reasoning rationales for test cases perceived by a human expert as particularly difficult helps the model generalize better. This

Table 2
Accuracy scores (in percent) for different reasoning-enhancing methods on valid test cases, shown separately for each exposure class. Each row reports the accuracy for the corresponding exposure class number (Class Number, CN).

CN	Claude 3 Sonnet				GPT-4o			
	CoT	ToT	One-shot CoT	One-shot ToT	CoT	ToT	One-shot CoT	One-shot ToT
1	100%	100%	100%	88.88%	100%	100%	100%	100%
2	76%	75.23%	82.31%	77.60%	84.18%	82.02%	85.79%	83.47%
3	95.98%	77.93%	94.12%	94.12%	99.69%	96.75%	95.98%	96.44%
4	66.66%	58.33%	97.14%	85.71%	97.22%	69.44%	100%	100%
5	86.84%	79.19%	84.37%	82.77%	81.64%	77.89%	81.04%	83.35%
6	64.51%	55.08%	81.32%	67.91%	73.97%	71.36%	90.19%	79.14%
7	95.80%	86.71%	100%	89.43%	100%	98.60%	100%	96.47%
8	100%	75%	100%	100%	100%	100%	100%	100%
9	75%	75%	100%	66.66%	100%	75%	100%	66.66%
10	81.25%	75%	100%	93.33%	93.75%	75%	100%	93.33%
11	84.37%	71.87%	74.60%	71.42%	98.43%	98.43%	98.41%	100%
12	87.5%	87.5%	85.71%	85.71%	87.5%	87.5%	100%	100%
13	100%	92.85%	69.23%	100%	92.85%	100%	100%	100%

Table 3
Pairwise Nemenyi test results. Entries indicate statistical significance at the $\alpha = 0.01$ level based on p-value thresholds ($p < 0.01$ indicates significance). Abbreviations: CoT = Chain-of-Thought, ToT = Tree-of-Thought, 1s = one-shot prompting.

	GPT-4o CoT	Claude CoT	GPT-4o ToT	Claude ToT	GPT-4o 1s CoT	Claude 1s CoT	GPT-4o 1s ToT	Claude 1s ToT
GPT-4o CoT	–							
Claude CoT	< .01	–						
GPT-4o ToT	> .05	< .01	–					
Claude ToT	< .01	< .01	< .01	–				
GPT-4o 1s CoT	< .01	< .01	< .01	< .01	–			
Claude 1s CoT	> .05	< .01	< .01	< .01	< .01	–		
GPT-4o 1s ToT	> .05	< .01	< .01	< .01	< .01	> .05	–	
Claude 1s ToT	< .01	> .05	> .05	< .01	< .01	< .01	< .01	–

aligns with previous findings showing substantial improvements from reasoning-enhancing methods such as CoT in mathematical and logical tasks (Wei et al., 2022; Sprague et al., 2024).

The only other study on financial regulatory texts relevant to this discussion is that of Cao and Feinstein (2024), which explores varying levels of prompting. Their work distinguishes between basic instructions, intermediate reasoning examples, and fully structured problem solutions.

The results of this work provide strong evidence that LLMs can be used to infer risk weights from regulatory articles. When viewed in the context of aiding the implementation of financial regulations, this approach may serve as a viable strategy for generating test cases. In conjunction with the findings of Fazlija et al. (2024), the results presented here point to a promising pathway toward more efficient regulatory implementation. By combining insights from both studies, it may be possible to develop an integrated and highly efficient approach to regulation—one that leverages minimal human input for test case generation while achieving high performance through advanced prompting strategies. LLMs could thus be incorporated into a unified workflow in which both code and test cases are generated by LLMs, with limited human expert intervention. Such a strategy opens avenues for further exploration into automated regulatory systems, which we are currently pursuing for future research.

There are several directions in which this work can be extended. An obvious limitation is the set of models used. In the fast-paced environment of LLM development and research, new and more advanced models emerge frequently. Recently, reasoning-oriented models — likely incorporating elements of CoT, ToT, and reinforcement learning — have shown substantial improvements in symbolic and logical reasoning. Future research should compare our results with those of such reasoning models, including o4-mini, which outperforms competitors on code generation and symbolic reasoning benchmarks.

When it comes to human expert input, the sensitivity of the proposed few-shot method with regard to the selection of the “most difficult” test cases by different human experts should be studied. Furthermore, approaches for automatically selecting the most relevant test cases should be considered.

Moreover, providing more than one example is very likely to help the models generalize even better — this is well-established in many few-shot learning experiments. It is important to note that the examples used for future regulations do not need to be up to date, as the general structure and logic of regulatory texts remain largely consistent. Testing on new regulatory articles or different regulatory frameworks (e.g., CRR3) would further evaluate the models’ generalization capabilities beyond a single regulation.

Finally, this study focuses on a regulatory aspect that is more quantitative in nature — computing risk weights — while omitting more qualitative aspects that might require different types of reasoning. Future research on these other aspects would give insights on LLM application to financial regulatory implementation more generally. Taken together, this research demonstrates that systems using LLMs for regulatory implementation and automation of other regulatory tasks shows promise and lays the groundwork for further research on AI-enhanced compliance systems.

CRediT authorship contribution statement

Bledar Fazlija: Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Data curation, Conceptualization. **Meriton Ibraimi:** Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Aynaz Forouzandeh:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Formal analysis, Data curation. **Arber Fazlija:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization.

Ethical approval

This study does not involve human participants or animals and therefore did not require ethical approval.

Data and code availability

The data and code used in this study are available from the corresponding author upon reasonable request.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT (OpenAI) to assist with proofreading and improving the readability of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Open access funding provided by ZHAW Zurich University of Applied Sciences. We thank Peter Schwendner for valuable discussions.

Appendix A. Zero-shot CoT prompt

The zero-shot CoT prompt is utilized in two variations: once with the assumptions provided and once without them. In each instance, only one test case related to exposure class i is included in all the prompts:

"You are given a regulatory text, and I want you to compute the risk weight for the given input values using the regulatory text.

The output should be an integer representing the corresponding risk weight (e.g., 20, 100) or the string "Invalid input value!" if invalid values are detected in input values.

Here is the regulatory text:
{Articles}

Here are the assumptions for the input values:
{Assumptions}

Here are the input values:
{Test case}

Think step-by-step to ensure accurate assignment of the risk weight.

Output the risk weight in risk weight `<riskweight>` `</riskweight>` tag".

Appendix B. Zero-shot CoT prompt example

You are given a regulatory text, and I want you to compute the risk weight for the given input values using the regulatory text.

The output should be an integer representing the corresponding risk weight (e.g., 20, 100) or the string `‘‘Invalid input value!’’` if invalid values are detected in input values.

Here is the regulatory text:

CRE 20.11: Exposures to domestic PSEs will be risk-weighted at national discretion, according to either of the following two options.

Risk weight table for PSEs – Option 1: Based on external rating of sovereign

External rating of the sovereign	AAA to AA–	A+ to A–	BBB+ to BBB–	BB+ to B–	Below B–	Unrated
Risk weight under Option 1	20%	50%	100%	100%	150%	100%

Risk weight table for PSEs – Option 2: Based on external rating of PSE

External rating of the PSE	AAA to AA–	A+ to A–	BBB+ to BBB–	BB+ to B–	Below B–	Unrated
Risk weight under Option 2	20%	50%	50%	100%	150%	50%

determines the test case

Input variables:

- Variable 1: The information of whether external rating of sovereign should be used, is provided by the user.
- Variable 2: The sovereign rating can get values like ‘AAA’, ‘AA+’, ‘AA’, etc.
- Variable 3: The PSE rating can get values like ‘AAA’, ‘AA+’, ‘AA’, etc.

Input values:
`use_sovereign_rating=True, sovereign_rating=AAA, pse_rating=AAA`

Think step-by-step to ensure accurate assignment of the risk weight. Output the risk weight in `<riskweight></riskweight>` tag.

Appendix C. One-shot CoT prompt

"You are given a regulatory text, and I want you to compute the risk weight for the given input values using the regulatory text.

The output should be an integer representing the corresponding risk weight (e.g., 20, 100) or the string "Invalid input value!" if invalid values are detected in input values.

Here is the regulatory text:
{Articles}

Here are the assumptions for the input values:
{Assumptions}

Here is an example:
Input values:
{Example test case}
Output:
{CoT example}

Input values:
{Test case}

Think step-by-step to ensure accurate assignment of the risk weight.

Output the risk weight in risk weight `<riskweight>` `</riskweight>` tag".

Appendix D. Zero-shot ToT prompt

"Simulate a collaborative discussion among three regulatory experts as they work together to assign the correct risk weight to the given input values, using the provided regulatory text. Each expert should explain their thought process in detail, step by step, considering the prior explanations of others. They should openly acknowledge any mistakes and adjust their reasoning accordingly. At each step, whenever possible, each expert refines and builds upon the thoughts of others, acknowledging their contributions. They continue until they reach a definitive risk weight. Present the entire process and reasoning in a markdown table for clarity.

The final risk weight should be an integer representing the assigned risk weight (e.g., 20, 100), or the string "Invalid input value!" if invalid values are detected in input values. This final result should be displayed inside the <riskweight></riskweight> tag. Use this tag only once for the final concluded risk weight.

Here is the regulatory text:
{Articles}

Here are the assumptions for the input values:
{Assumptions}

Here are the input values:
{Test case}"

Appendix E. One-shot ToT prompt

"Simulate a collaborative discussion among three regulatory experts as they work together to assign the correct risk weight to the given input values, using the provided regulatory text. Each expert should explain their thought process in detail, step by step, considering the prior explanations of others. They should openly acknowledge any mistakes and adjust their reasoning accordingly. At each step, whenever possible, each expert refines and builds upon the thoughts of others, acknowledging their contributions. They continue until they reach a definitive risk weight. Present the entire process and reasoning in a markdown table for clarity.

The final risk weight should be an integer representing the assigned risk weight (e.g., 20, 100), or the string "Invalid input value!" if invalid values are detected in input values. This final result should be displayed inside the <riskweight></riskweight> tag. Use this tag only once for the final concluded risk weight.

Here is the regulatory text:
{Articles}

Here are the assumptions for the input values:
{Assumptions}

Here is an example:
  Input values:
  {Example test case}
  Output:
  {ToT example}

Here are the input values:
{Test case}"

Appendix F. Prompt for creating CoT examples

"I will provide a regulatory text along with its corresponding code, specific input values, and the expected output. Your task is to explain the step-by-step reasoning behind how the risk weight is assigned based on the given input values.

Here is the regulatory text:
{Articles}

Here are the assumptions for the input values:
{Assumptions}

Here is the code:
{GTC}

Here are the input values:
{The most complex example test case}

The expected output is:
<riskweight>{Risk weight of the example test case}
</riskweight>".

Appendix G. Prompt for creating ToT examples

"Simulate a collaborative discussion among three regulatory experts as they work together to assign the correct risk weight to the given input values, using the provided code, regulatory text, and expected output. Each expert should explain their thought process in detail, step by step, considering the prior explanations of others. They should openly acknowledge any mistakes and adjust their reasoning accordingly. At each step, whenever possible, each expert refines and builds upon the thoughts of others, acknowledging their contributions. They continue until they reach a definitive risk weight. Present the entire process and reasoning in a markdown table for clarity.

Here is the regulatory text:
{Articles}

Here are the assumptions for the input values:
{Assumptions}

Here is the code:
{GTC}

Here are the input values:
{The most complex example test case}

The expected output is:
<riskweight>{Risk weight of the example test case}
</riskweight>".

Data availability

The data and code used in this study are available from the corresponding author upon reasonable request.

References

Achitouv, I., Gorduza, D., Jacquier, A., 2023. Natural language processing for financial regulation. arXiv preprint arXiv:2311.08533.

Aros, L.H., Molano, L.X.B., Gutierrez-Portela, F., Hernandez, J.J.M., Barrero, M.S.R., 2024. Financial fraud detection through the application of machine learning techniques: a literature review. Humanit. Soc. Sci. Commun. 11, 1–22.

Becker, S., Cheridito, P., Jentzen, A., 2020. Pricing and Hedging American-style options with deep learning. J. Risk Financ. Manag. 13 (7), 158. http://dx.doi.org/10.3390/jrfm13070158.

Bingler, J., Kraus, M., Leippold, M., Webersinke, N., 2022. How cheap talk in climate disclosures relates to climate initiatives. Corp. Emiss. Reput. Risk doi: 10.

Buehler, H., Gonon, L., Teichmann, J., Wood, B., 2019. Deep hedging. Quant. Finance 19, 1271–1291.

Cao, Z., Feinstein, Z., 2024. Large language model in financial regulatory interpretation. arXiv preprint arXiv:2405.06808.

De Lucio, J., Mora-Sanguinetti, J.S., 2021. New dimensions of regulatory complexity and their economic cost. An analysis using text mining.

Dornadula, V.N., Geetha, S., 2019. Credit card fraud detection using machine learning algorithms. Procedia Comput. Sci. 165, 631–641.

EFD, E.F., 2023. Regulierungsfolgenabschätzung zur änderung der eigenmittelverordnung (nationale umsetzung der abgeschlossenen basel-III-reformen). URL https://www.newsd.admin.ch/newsd/message/attachments/84841.pdf.

Fazlija, B., Ibraimi, M., Forouzandeh, A., Fazlija, A., 2024. Implementing financial regulations using large language models. Available at SSRN.

Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. Ann. Math. Stat. 11 (1), 86–92.

Goodell, J.W., Kumar, S., Lim, W.M., Pattnaik, D., 2021. Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. J. Behav. Exp. Financ. 32, 100577.

Gregory, G., Vito, L., 2024. ChatGPT: a canary in the coal mine or a parrot in the echo chamber? detecting fraud with LLM: the case of FTX. Financ. Res. Lett. 106349.

Gu, S., Kelly, B., Xiu, D., 2020. Empirical asset pricing via machine learning. Rev. Financ. Stud. 33, 2223–2273.

Hulbert, D., 2023. Using tree-of-thought prompting to boost ChatGPT's reasoning.

Levine, R., 2012. The governance of financial regulation: reform lessons from the recent crisis. Int. Rev. Financ. 12, 39–56.

Long, J., 2023. Large language model guided tree-of-thought. arXiv preprint arXiv:2305.08291.

Micheler, E., Whaley, A., 2020. Regulatory technology: replacing law with computer code. Eur. Bus. Organ. Law Rev. 21, 349–377.

Nemenyi, P.B., 1963. Distribution-Free Multiple Comparisons. Princeton University.

Rossi, A.G., 2018. Predicting stock market returns with machine learning. Georg. Univ..

Shanmuganathan, M., 2020. Behavioural finance in an era of artificial intelligence: Longitudinal case study of robo-advisors in investment decisions. J. Behav. Exp. Financ. 27, 100297.

Sprague, Z., Yin, F., Rodriguez, J.D., Jiang, D., Wadhwa, M., Singhal, P., Zhao, X., Ye, X., Mahowald, K., Durrett, G., 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. arXiv preprint arXiv:2409.12183.

Tarullo, D.K., 2019. Financial regulation: Still unsettled a decade after the crisis. J. Econ. Perspect. 33, 61–80.

2024. The basel framework.

Thomson Reuters, 2023. Cost of compliance. URL https://legal.thomsonreuters.com/en/insights/reports/cost-of-compliance-2023.

Webersinke, N., Kraus, M., Bingler, J.A., Leippold, M., 2021. Climatebert: A pretrained language model for climate-related text. arXiv preprint arXiv:2110.12010.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., 2022. Chain-of-thought prompting elicits reasoning in large language models. Adv. Neural Inf. Process. Syst. 35, 24824–24837.

Xu, H., Li, S., Niu, K., Ping, G., 2024. Utilizing deep learning to detect fraud in financial transactions and tax reporting. J. Econ. Theory Bus. Manag. 1, 61–71.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., Narasimhan, K., 2024. Tree of thoughts: Deliberate problem solving with large language models. Adv. Neural Inf. Process. Syst. 36.

Zhang, Z., Zhang, A., Li, M., Smola, A., 2022. Automatic chain of thought prompting in large language models. arXiv preprint arXiv:2210.03493.

Zhong, X., Enke, D., 2019. Predicting the daily return direction of the stock market using hybrid machine learning algorithms. Financ. Innov. 5, 1–20.

Zhou, Y., Li, H., Xiao, Z., Qiu, J., 2023. A user-centered explainable artificial intelligence approach for financial fraud detection. Financ. Res. Lett. 58, 104309.