👮‍♂️ CAP: Data Contamination Detection via Consistency Amplification

Anonymous ACL submission

Abstract

Large language models (LLMs) are widely used, but concerns about data contamination challenge the reliability of LLM evaluations. Existing contamination detection methods are often task-specific or require extra prerequisites, limiting practicality. We propose a novel framework, Consistency Amplification-based Data Contamination Detection (CAP), which introduces the Performance Consistency Ratio (PCR) to measure dataset leakage by leveraging LM consistency. To the best of our knowledge, this is the first method to explicitly differentiate between fine-tuning and contamination, which is crucial for detecting contamination in domain-specific models. Additionally, CAP is applicable to various benchmarks and works for both white-box and black-box models. We validate CAP's effectiveness through experiments on seven LLMs and four domain-specific benchmarks. Our findings also show that composite benchmarks from various dataset sources are particularly prone to unintentional contamination. All resources are publicly available[1].

(a) PCR Value Heatmap

Method Comparison	Task Formulations			Model Transparency		No Extra Prerequisites
	MCQ	Q&A	Sequence Generation	White / Gray	Black	
BenBench (Xu et al., 2024b)	✗	✓	✓	✓	✗	✓
DCQ (Golchin and Surdeanu, 2024)	✗	✓	✓	✓	✓	✓
LLM Decontaminator (Yang et al., 2023a)	✗	✓	✓	✓	✓	✗
Probe Classifier (Liu et al., 2024a)	✓	✓	✓	✓	✗	✓
Task Contamination (Li and Flanigan, 2023)	✓	✓	✓	✓	✓	✗
CAP (Ours)	✓	✓	✓	✓	✓	✓

(b) Method Comparison

Figure 1: (a) Heatmap of Performance Consistency Ratios: Change direction indicates scenario: *Fine-Tuning* or *Data Contamination*. (b) Method Comparison: CAP supports diverse tasks and is effective across models at varying transparency levels; it requires no prerequisites.

1 Introduction

Large language models (LLMs) have significantly advanced natural language processing (NLP), leading to widespread use in applications such as translation (Agostinelli et al., 2024), summarization (Chang et al., 2024) and question answering (Schimanski et al., 2024). Notable models include GPT (OpenAI, 2024) series and LLaMA (Touvron et al., 2023). Beyond general-purpose applications, some LLMs have been fine-tuned for specific domains, producing specialized models with expertise in fields like law (Cui et al., 2024), biomedicine (Bolton et al., 2024), finance (Yang et al., 2023b), and software development (Chen et al., 2021a).

Meanwhile, various benchmarks have been developed to assess LLM capabilities, focusing on specific skills (Hendrycks et al., 2021b) or comprehensive abilities (Xu et al., 2020; Hendrycks et al., 2021a). Based on these benchmarks, certain models claim to surpass larger counterparts, despite being significantly smaller in size—for instance, 7B or 13B models (Liu et al., 2024; Lei et al., 2024) compared to GPT models.

Concerns about LLM evaluation reliability are increasing due to a lack of transparency, especially with closed-source models. It raises a crucial question: *Is their impressive performance a result of generalization, or merely due to prior exposure and memorization of test samples, a problem known as data contamination?* Without effective detection methods, model providers might maliciously contaminate models to boost performance.

Data contamination detection (DCD) methods

[1] https://github.com/anonymous-cap/cap

are divided into two main categories (Xu et al., 2024a). The first is matching-based, which examines benchmark overlap (Yang et al., 2023a) or investigates whether models memorize specific samples (Golchin and Surdeanu, 2024; Deng et al., 2024). A key limitation of these methods is their vulnerability to surface-level paraphrasing (Dekoninck et al., 2024). The second involves comparison-based techniques, which are more robust but often require specific conditions (Xu et al., 2024b; Roberts et al., 2023; Dong et al., 2024a). In this study, we identify two main limitations in current DCD methods: (1) Existing methods do not explicitly address contamination detection in domain-specific models. As fine-tuning for specific domains becomes more common, DCD methods must distinguish between malicious contamination and legitimate fine-tuning. (2) Most DCD techniques are task-specific. For example, Golchin and Surdeanu (2024) and Deng et al. (2024) focus on text generation tasks. Extending these methods to handle a broader range of tasks remains a challenge.

In this work, we introduce a novel framework, Data Contamination Detection via **Consistency Amplification** (**CAP**), which leverages the concept of LM consistency (Jang et al., 2022). CAP calculates the Performance Consistency Ratio (PCR) as an indicator and compares the PCR between the training and test sets. A detailed heatmap illustrating this comparison is shown in Figure 1 (a). The direction of the difference from the training set to the test set indicates either fine-tuning or contamination. Additionally, Figure 1 (b) demonstrates that CAP has broader applicability across various tasks and model types compared to other methods.

We conducted experiments with 7 LLMs on 4 financial benchmarks, including 4 general-purpose models and 3 financial-specific LLMs (Fin-LLMs). Results show that CAP can effectively distinguish between fine-tuning and benchmark contamination. The Fin-LLMs tested are predominantly fine-tuned without intentional contamination, but we identified two key issues: (1) contamination in general-purpose models can propagate to domain-specific models fine-tuned on them, and (2) Benchmarks from public sources, like exams, textbooks, or older datasets, are prone to unintentional overlap with other data. Our contributions are threefold:

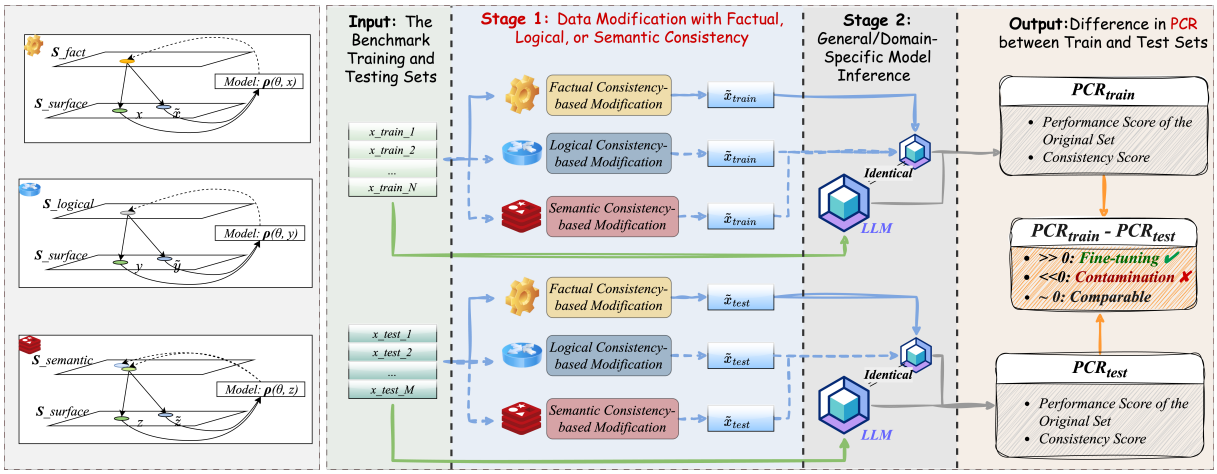
- We introduce CAP, a novel method for data contamination detection, and define PCR, mathematically demonstrating that PCR increases when datasets are used in training.
- We are the first to explicitly investigate contamination in domain-specific LLMs, providing a comprehensive empirical evaluation of domain-specific benchmarks and models.
- We highlight the risks of unintentional contamination, particularly in composite benchmarks, given the rapid growth of new benchmarks.

2 Related Work

2.1 Data Contamination Detection (DCD)

Matching-based DCD Efforts to detect data contamination in LLMs can generally be categorized into two main approaches. The first is matching-based, which evaluates whether the model recalls specific training samples. Notable methods include DCQ (Golchin and Surdeanu, 2024), where the model selects the correct sample from a set that includes the original and three perturbed variants. Similarly, Deng et al. (2024) introduced Testset Slot Guessing, which hides parts of the test set and prompts the LLM to predict the missing elements. Additionally, Yang et al. (2023a) utilize similarity search to retrieve the top-k samples from a dataset, followed by prompting a strong LLM like GPT-4 to assess whether any of the top-k samples are too similar to the test case. One limitation of these methods is their vulnerability to surface-level rephrasing, where minor alterations can lead to failure (Dekoninck et al., 2024). Furthermore, while these techniques are effective for tasks like text generation, where the outputs consist of long sequences, they may be less suitable for benchmarks like Multiple Choice Questions (MCQs), which have become increasingly popular nowadays. Our method can handle both task types effectively.

Comparison-based DCD The second type is comparison-based. For example, CDD (Dong et al., 2024b) relies on the output distribution of LLMs to detect data contamination. However, it requires an ideal reference output distribution, which can be challenging to define or obtain, especially when benchmarks involve composite tasks. Xu et al. (2024b) applied perplexity for contamination detection, but their approach requires access to hidden layers during decoding, making it suitable only for gray-box or white-box models. In contrast, our method works for both types of LMs. Roberts et al. (2023) conducted a longitudinal analysis of data



(a). LM Consistency (b). Framework

Figure 2: (a) LM Consistency: Different points in surface space may correspond to the same points in logical and factual space, or to nearby points in semantic space. (b) The CAP Framework: (i) The input stage consists of the training and test sets, covering benchmarks like multiple-choice questions (MCQ) and sequence generation tasks; (ii) Data modification is applied with a specific consistency criterion; (iii) Both the original and modified datasets are processed through LLMs, which can be black-box models; (iv) Finally, the Performance Consistency Ratio (PCR) is calculated and compared between the training and test sets to assess the presence of fine-tuning or contamination.

contamination in LLMs before and after specific training cutoffs. However, the absence of temporal markers in certain datasets limits the application of such chronological analyses. Our method, by contrast, leverages the intrinsic consistency of LMs, reducing reliance on external resources like temporal information, making it more broadly applicable.

2.2 Consistency in LLMs

The concept of consistency in large language models (LLMs) was introduced by Jang et al. (2022), who developed a framework based on behavioral consistency and unified previous studies into a taxonomy. This framework defines consistency through two key components: *belief*, referring to what the model assumes to be true, and *principle*, which governs coherent decision-making. From these foundations, three main categories of consistency emerged: *semantic*, *logical*, and *factual*.

Semantic consistency refers to the ability of an LLM to make the same decisions when faced with semantically equivalent texts. It aligns with Meaning-Text Theory (MTT), which posits a many-to-many relationship between linguistic forms and semantic content (Mel'čuk and Žolkovskij, 1970).

Logical consistency refers to a model's ability to make reasoning decisions without contradictions. It ensures that superficial changes in phrasing do not affect the model's reasoning process or outcomes. Therefore, the model should consistently generate correct predictions even when the input undergoes minor modifications.

Factual consistency refers to a model's ability to produce outputs that align with established facts and the provided context. This type of consistency is crucial for tasks involving knowledge grounding and utilization, as the model must avoid generating outputs that contradict the knowledge it holds.

3 Methodology

3.1 Problem Formulation

Dekoninck et al. (2024) define benchmark-level dataset contamination as occurring when a model's training corpus D overlaps with a benchmark set D'. Our study refines this definition in the context of domain-specific models. We clarify that fine-tuning using the training set of a benchmark is permissible. Benchmark-level data contamination occurs when the **test set** of a benchmark is included during the pre-training or fine-tuning of a LLM. We provide the following formal definition:

Definition (Benchmark Contamination): A model is considered contaminated by a benchmark if its training corpus D overlaps with the benchmark's test set D'_test, i.e., $D'_test \cap D \neq \emptyset$.

When a dataset is not explicitly identified as part of a benchmark's training or test set, we use the term ***dataset leakage*** to indicate the extent to which the model has been exposed to, memorized, or trained on the dataset (Xu et al., 2024b). For

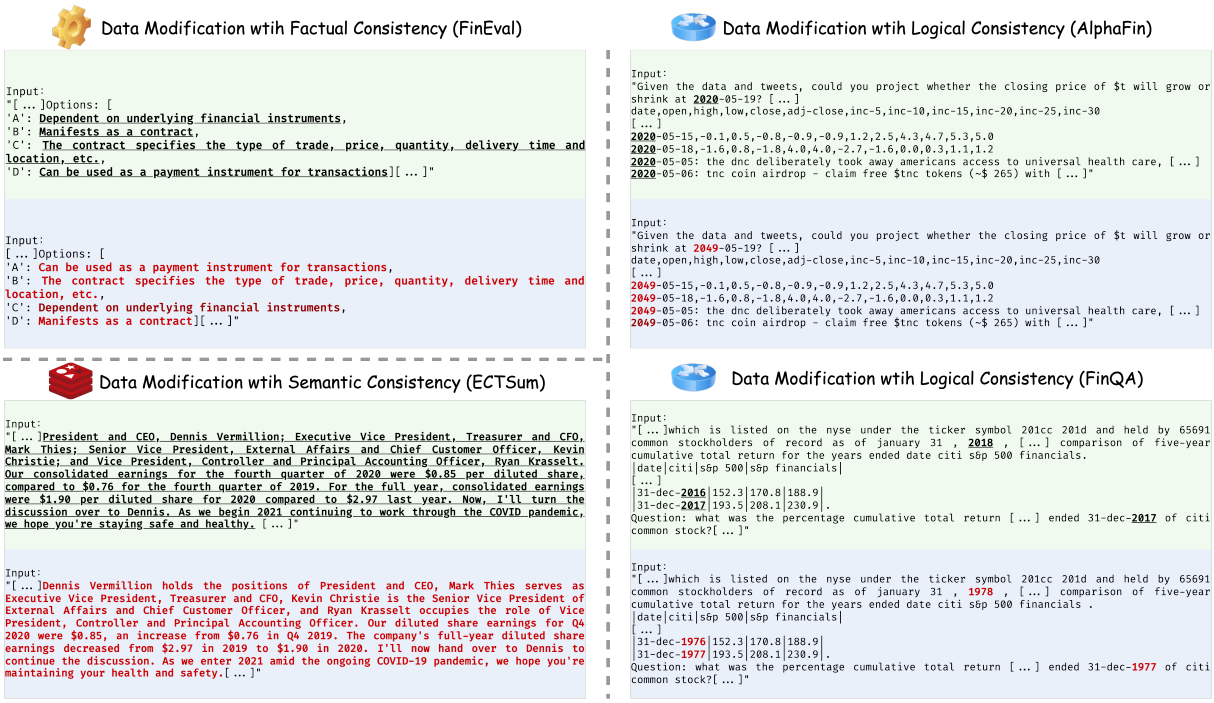


Figure 3: Consistency-based Data Modification: (1) For MCQ benchmarks like FinEval, we apply factual consistency by rearranging the corresponding content of options. (2) For Q&A tasks like AlphaFin and FinQA, we modify logically unrelated information (e.g., 'year') while keeping the reasoning process intact. (3) For summarization tasks like ECTSum, we modify the input text while maintaining semantic meaning (cosine similarity > 0.979).

domain-specific LLMs, the objective of contamination detection is to assess whether the model has been fine-tuned using the benchmark's training set or has been contaminated by its test set.

3.2 CAP Framework

Figure 2 (a) illustrates the concept of LM consistency. For data points x and $\tilde{x}$ in the surface space, which represent different expressions of the same fact, factual consistency encourages the model's predictions to reflect the same underlying fact or knowledge. The same principle applies to logical consistency. In the semantic space, when two surface-level data points, though textually different, are semantically equivalent, the model's outputs should exhibit a high degree of semantic similarity.

Building on these principles, we propose the **CAP** framework: Data Contamination Detection via Consistency Amplification, as illustrated in Figure 2 (b). CAP operates through four stages: (1) At the Input Stage, two sets are provided for a specific benchmark: the training set and the test set. (2) Stage 1: Modified datasets are constructed based on factual, logical, or semantic consistency, depending on the task. For example, factual consistency is recommended for MCQs, logical consistency for reasoning tasks, and semantic consistency for tasks like sequence generation or summarization. (3) Stage 2: LM inference is performed using white-box, gray-box, or black-box models. The model is run on both the original and modified training and test sets. (4) Output Stage: The Performance Consistency Ratio (PCR) is computed for both sets. If $PCR_train >> PCR_test$, this suggests fine-tuning. Conversely, if $PCR_train << PCR_test$, it indicates test data contamination. When the difference between PCR_Train and PCR_test is close to zero, it implies no leakage or similar contamination levels in both sets.

3.3 Performance Consistency Ratio

This section explains the mathematical relationship between the Performance Consistency Ratio (PCR) and dataset leakage, demonstrating how PCR can be used as a measure of data contamination. Our primary objective is to compare the degree of leakage between the training and test sets of a benchmark for a LM. Let $\rho(\cdot)$ represent a language model and X an arbitrary dataset. We define M as the metric score of the model's predictions, $\gamma(\cdot)$ as a consistency-based data modification operation, and C as the consistency score between the model's predictions on the original and modified datasets. Suppose X_1 is a dataset included in the pre-training

or fine-tuning set of model ρ. The following holds:

- The performance metric score $M \geq 0$ and the consistency score $C \geq 0$.

- Generally, the performance metric of a fine-tuned dataset $M(\rho(X_1))$ is greater than or equal to $M(\rho(X))$ (Xu et al., 2024b):

$$M(\rho(X_1)) \geq M(\rho(X)) \tag{1}$$

- The consistency between the model outputs on datasets X and X_1, denoted by $C(\rho(X), \rho(X_1))$, is less than or equal to the consistency between the outputs on the original dataset X and its consistency-based modified version $\gamma(X)$ (Jang et al., 2022):

$$C(\rho(X), \rho(X_1)) \leq C(\rho(X), \rho(\gamma(X))) \tag{2}$$

From Inequality (2), by substituting $X = X_1$ on the right-hand side, we obtain:

$$C(\rho(X), \rho(X_1)) \leq C(\rho(X_1), \rho(\gamma(X_1))) \tag{3}$$

Considering Inequalities (1) and (2), we can express the following:

$$\frac{M(\rho(X_1))}{C(\rho(X), \rho(X_1))} \geq \frac{M(\rho(X))}{C(\rho(X), \rho(\gamma(X)))} \tag{4}$$

Furthermore, considering Inequality (3), for the left-hand side of Inequality (4), we obtain:

$$\frac{M(\rho(X_1))}{C(\rho(X_1), \rho(\gamma(X_1)))} \geq \frac{M(\rho(X_1))}{C(\rho(X), \rho(X_1))} \tag{5}$$

By combining (4) and (5), we derive:

$$\frac{M(\rho(X_1))}{C(\rho(X_1), \rho(\gamma(X_1)))} \geq \frac{M(\rho(X))}{C(\rho(X), \rho(\gamma(X)))} \tag{6}$$

We define the Performance Consistency Ratio (**PCR**) as:

$$PCR = PCR(X, \gamma, \rho) = \frac{M(\rho(X))}{C(\rho(X), \rho(\gamma(X)))} \tag{7}$$

From Inequality (6), it follows that the PCR for a dataset included in the pre-training or fine-tuning set of the language model will be larger.

3.4 Practical Implementation of the Performance Consistency Ratio

In practical applications, the consistency metric C can take on very small values, and in some cases, it may approach zero. To mitigate the risk of division by zero, we introduce a negligibly small constant α during the computation. The adjusted Performance Consistency Ratio (PCR) is therefore defined as:

$$PCR = \frac{M(\rho(X)) + \alpha}{C(\rho(X), \rho(\gamma(X))) + \alpha} \tag{8}$$

Moreover, to ensure that the resulting values are within a suitable range, we apply a $Tanh(\cdot)$ transformation. The practical formula for the PCR is:

$$PCR = Tanh\left(\frac{M(\rho(X)) + \alpha}{C(\rho(X), \rho(\gamma(X))) + \alpha}\right) \tag{9}$$

3.5 Consistency-based Data Modification

This section introduces the consistency-based data modification operation, $\gamma(\cdot)$, for a dataset X. As depicted in Figure 2, there are three types of consistency: factual, logical, and semantic. To evaluate the consistency metric before and after data modification, it is essential that the modification operation aligns with the specified type of consistency during the synthesis of the modified data. Examples of each consistency type are provided in Figure 3.

Benchmark Collection Our research focuses on methods for detecting dataset contamination in both general-purpose and domain-specific models. Given the recent rapid growth of the financial language models (Fin-LLMs), this study focuses on this domain. Key open-source benchmarks include FinQA (Chen et al., 2021b), ECTSum (Mukherjee et al., 2022), FinEval (Zhang et al., 2023) and AlphaFin (Li et al., 2024), among others.

Factual-Consistency-Based Data Modification FinEval consists of MCQs designed to assess the model's ability to leverage financial knowledge. For this benchmark, we measure the factual consistency of the model's predictions before and after applying modifications. The modification involves rearranging the content of choices 'A', 'B', 'C', and 'D'. An example is provided in Figure 3.

Logical-Consistency-Based Data Modification FinQA is a benchmark for table-based and textual question answering (Q&A), designed to assess models' reasoning and calculation abilities.

FinEval		FinQA			AlphaFin		ECTSum		
#Dev	#Val	#Train	#Val	#Test	#Train	#Val	#Train	#Val	#Test
170	1151	500 100^+	500 100^+	500 100^+	500	500	517 100^*	101 100^*	250 100^*

Table 1: Dataset size statistics. In FinQA, '+' indicates Baichuan and Disc-Fin used 100 samples. In ECTSum, '*' denotes LLaMA and Baichuan used 100 samples.

For this benchmark, we focus on the logical consistency of the model's answers before and after applying modifications. Similarly, AlphaFin, a composite benchmark, includes both traditional financial question-answering tasks and more advanced tasks such as stock prediction. Our focus is on the Q&A component of AlphaFin. For these two benchmarks, we use a minor modification approach by changing the 'year' information in the data, while the logical calculation should yield the same result. Examples are provided in Figure 3.

Semantic-Consistency-Based Data Modification ECTSum is a summarization benchmark that evaluates the sequence generation capabilities of LMs. We measure the semantic consistency of output summaries before and after applying modifications to the input. Model `claude-3.5-sonnet-20240620` is used to construct semantically similar versions of the original long text. The prompt is in Table 12 in Appendix C, and an example is shown in Figure 3.

4 Experiment Settings

4.1 Models & Benchmarks & Baselines

Models The Fin-LLMs used in our experiments include FinMA (Xie et al., 2023) and Disc-Fin (Chen et al., 2023), which have been fine-tuned on different base models: FinMA is fine-tuned on LLaMA-2 (Touvron et al., 2023), while Disc-Fin is fine-tuned on Baichuan (Baichuan-inc, 2023). Additionally, we incorporated several widely used general-purpose LLMs to provide a comprehensive evaluation. These models include LLaMA, Mistral, Baichuan, and GPT. The specific versions are: Llama-3-8B-Instruct, Mistral-7B-Instruct-v0.3, FinMA-7B-Full, FinMA-7B-NLP, Baichuan-13B-Chat, DISC-FinLLM-13B, and `gpt-4o-mini-2024-07-18`.

Benchmarks and Datasets Our experiments span a diverse range of NLP tasks, utilizing benchmarks that include FinQA, ECTSum, FinEval, and AlphaFin. Dataset sizes are shown in Table 1.

Baselines We compare our method to two main approaches. The first examines performance degradation as an indicator of contamination, following Roberts et al. (2023), Huang et al. (2024), and Li and Flanigan (2024). The second measures the discrepancy between training and test performance drop, calculated as $\delta_{\text{train-test}} = \delta_{\text{train}} - \delta_{\text{test}}$, where $\delta = \Delta / M_{\text{ori}}$, $\Delta = M_{\text{ori}} - M_{\text{mod}}$, and M represents metric scores, following Xu et al. (2024b).

4.2 Metrics

Performance Metrics The evaluation metrics were selected based on the characteristics of each benchmark. For FinEval, we used Exact Match (EM) (Kamalloo et al., 2023) as the metric. FinQA and AlphaFin were evaluated using ROUGE-L (Lin, 2004), which closely aligns with ROUGE-1 but provides a more comprehensive measure of answer quality. For ECTSum, given its focus on summarization, ROUGE-2 was chosen to assess the generated sequences, as it imposes stricter requirements for alignment with the reference summaries.

Consistency Score The consistency score across the three scenarios can be calculated as follows:

$$\begin{cases} C_{\text{factual}} = C\left(X, \gamma_{\text{factual}}\left(X\right)\right) \\ C_{\text{logical}} = C\left(X, \gamma_{\text{logical}}\left(X\right)\right) \\ C_{\text{semantic}} = C\left(X, \gamma_{\text{semantic}}\left(X\right)\right) \end{cases} \tag{10}$$

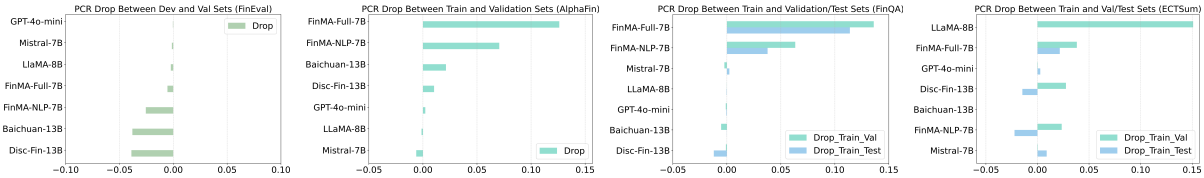
$$C\left(X, \gamma\left(X\right)\right) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{I}\left(\rho\left(x_i\right), \rho\left(\gamma\left(x_i\right)\right)\right) \tag{11}$$

where $\mathcal{I}$ is an indicator function that determines whether $\rho(x_i)$ and $\rho(\gamma(x_i))$ are equivalent, and $x_i \in X = \{x_1, \cdots, x_N\}$. The definition of $\mathcal{I}$ depends on the type of consistency:

- **Factual Consistency**: $\mathcal{I}$ is set to 1 if $\rho(x_i)$ and $\rho(\gamma(x_i))$ are identical; otherwise, it is 0.
- **Logical Consistency**: $\mathcal{I}$ is approximated by computing the Jaccard distance (Song et al., 2024) between $\rho(x_i)$ and $\rho(\gamma(x_i))$.
- **Semantic Consistency**: $\mathcal{I}$ is based on the cosine similarity of embedding vectors for $\rho(x_i)$ and $\rho(\gamma(x_i))$, with $\mathcal{I} = 1$ if the similarity exceeds a specified threshold; otherwise, it is 0. The threshold is set to 0.95 in this study.

5 Results and Discussion

Table 2 summarizes the results for the seven models across the four benchmarks. The PCR



(a) FinEval (b) AlphaFin Research (c) FinQA (d) ECTSum

Figure 4: Illustration of Four Benchmarks: (a) Baichuan and Disc-Fin may have been contaminated by the FinEval validation set; (b) and (c) FinMA-Full and FinMA-NLP were fine-tuned with the FinQA training set. AlphaFin Research, though newly proposed, has overlap with FinQA; (d) ECTSum is less likely to be used, with most values close to zero. However, it suggests LLaMa may have been contaminated by the training and test sets.

	FinEval				**FinQA**						
	Baselines			**CAP** (Ours)	**Baselines**					**CAP** (Ours)	
	Performance Decline (Δ)		**Discrepancy Comparison** (δ)		**Performance Decline** (Δ)			**Discrepancy Comparison** (δ)			
Model	Δ_{Dev}	Δ_{Val}	$\delta_{\text{dev-val}}$	$PCR_{\text{dev-val}}$	Δ_{Train}	Δ_{Val}	Δ_{Test}	$\delta_{\text{train-val}}$	$\delta_{\text{train-test}}$	$PCR_{\text{train-val}}$	$PCR_{\text{train-test}}$
LLaMA-8B	0	0	0	-0.0023	-0.0001	-0.0001	-0.0002	0	0.0433	-0.0001	-0.0003
Mistral-7B	0	0	0	-0.0013	-0.0011	0.0004	-0.0004	-0.0701	-0.0327	-0.0023	0.0023
FinMA-Full-7B	0.0471	0.0059	0.2235	-0.0054	**0.0123**	**0.0108**	**0.0068**	-0.0611	-0.0093	**0.136**	**0.1139**
FinMA-NLP-7B	-0.0471	-0.0417	-0.0197	-0.0255	0.009	0.0054	0	0.0021	0.0608	0.0633	0.0377
Baichuan-13B	-0.0177	0.0235	-0.0935	-0.038	0.0006	-0.0031	0.0013	**0.3814**	-0.0891	-0.0053	-0.0001
Disc-Fin-13B	**0.0588**	**0.0443**	0.0241	**-0.0389**	0.0007	-0.0017	-0.0015	0.2162	**0.1763**	-0.0009	-0.0121
GPT-4o-mini	0	-0.0009	**9**	-0.0004	-0.0008	0.0011	1e-04	-0.089	-0.0431	-0.001	-0.0005

	AlphaFin				**ECTSum**						
	Baselines			**CAP** (Ours)	**Baselines**					**CAP** (Ours)	
	Performance Decline (Δ)		**Discrepancy Comparison** (δ)		**Performance Decline** (Δ)			**Discrepancy Comparison** (δ)			
Model	Δ_{Train}	Δ_{Val}	$\delta_{\text{train-val}}$	$PCR_{\text{train-val}}$	Δ_{Train}	Δ_{Val}	Δ_{Test}	$\delta_{\text{train-val}}$	$\delta_{\text{train-test}}$	$PCR_{\text{train-val}}$	$PCR_{\text{train-test}}$
LLaMA-8B	-1e-04	0.0005	**-0.0651**	-0.0013	0.0024	0.0024	0.0011	-0.0095	0.0356	**0.1509**	0
Mistral-7B	-0.0015	-0.003	0.0121	-0.0062	**0.0052**	**0.0032**	**0.0032**	0.0388	0.0332	-0.0004	0.0089
FinMA-Full-7B	**0.034**	-0.0047	0.0518	**0.1261**	1e-04	-0.0005	0.0002	0.2917	-0.0453	0.0382	0.0216
FinMA-NLP-7B	0.0033	0.0014	0.0031	0.0706	-0.0002	-0.0007	-0.0002	**6.8462**	0.028	0.0235	**-0.0223**
Baichuan-13B	0.0022	0.0049	-0.032	0.0213	0.0024	0.0017	-0.001	0.1285	**0.5855**	0	0
Disc-Fin-13B	0.0006	0.0009	-0.0045	0.0103	-0.0015	-0.0004	0.0008	-0.2771	-0.5402	0.0276	-0.0147
GPT-4o-mini	-0.0046	**-0.0144**	0.0383	0.0022	0.0032	-0.0011	0.001	0.0926	0.0447	0.0004	0.0027

Table 2: Comparison of results across the FinEval, FinQA, AlphaFin, and ECTSum benchmarks for various models using Baseline and CAP (Ours) methods. The largest absolute values are in **bold**, and the second largest are underlined. FinMA-Full and FinMA-NLP are fine-tuned with the FinQA training set, aligning with CAP's findings (Xie et al., 2023). FinEval shows overlap with Baichuan's fine-tuning set, further supporting CAP's results (Baichuan-inc, 2023). There is insufficient evidence to determine whether ECTSum has contaminated models.

discrepancies, $PCR_{train-test}$, are calculated as $PCR_{train-test} = PCR_{train} - PCR_{test}$. The results for each benchmark are displayed in Figure 4. A substantially negative $PCR_{train-test}$ value suggests potential contamination within the test set, while a significantly positive value indicates fine-tuning on the training set. A $PCR_{train-test}$ value close to zero implies comparable exposure between the training and test sets. Detailed experimental results are provided in Appendix A.

5.1 FinEval Results and Analysis

We use the terms 'dev' and 'val' sets as defined in the FinEval benchmark. Notably, the test set does not provide public answers. As shown in Figure 4 (a), Baichuan-13B and Disc-Fin-13B are highly likely trained using the FinEval validation set. Baichuan-13B was fine-tuned on the C-EVAL benchmark (Huang et al., 2023), and we found overlap between C-EVAL and FinEval, with 3 identical questions present in both validation sets. These samples are provided in Appendix D. Additionally, we discovered 21 overlapping samples between the FinEval validation set and the C-EVAL test set.

Although Disc-Fin-13B's instruction-tuning set does not explicitly include FinEval, it was fine-tuned on Baichuan-13B, suggesting that data contamination might be inherited. In addition, this overlap highlights the risk that benchmarks constructed from public sources, such as standardized exams (e.g., Chartered Financial Analyst [CFA]) or textbooks, may unintentionally overlap with others.

5.2 FinQA and AlphaFin Results and Analysis

As shown in Figure 4 (c), both FinMA-Full and FinMA-NLP were likely fine-tuned using the FinQA training set. This observation is consistent with the FinMA paper (Xie et al., 2023), which

Model	$PCR_{dev-val}$ with $\tanh(\cdot)$				$PCR_{dev-val}$ w/o $\tanh(\cdot)$				$PCR_{dev-val}$ with $\sigma(\cdot)$			
	$\alpha = 0$	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 0$	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 0$	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.1$
LLaMA-8B	0	-0.0002	-0.0023	-0.0165	0	-0.0002	-0.0023	-0.0169	0	-0.0001	-0.0006	-0.0042
Mistral-7B	0	-0.0001	-0.0013	-0.0091	0	-0.0001	-0.0013	-0.0094	0	0	-0.0003	-0.0023
FinMA-Full-7B	-0.0051	-0.0051	-0.0054	-0.0062	-0.0078	-0.0079	-0.0084	-0.0107	-0.0017	-0.0018	-0.0019	-0.0023
FinMA-NLP-7B	-0.0245	-0.0246	-0.0255	-0.0264	-0.0296	-0.0298	-0.0313	-0.037	-0.0071	-0.0071	-0.0074	-0.0085
Baichuan-13B	-0.0389	-0.0388	-0.038	-0.0316	-0.0834	-0.0832	-0.0817	-0.069	-0.0169	-0.0169	-0.0166	-0.0139
Disc-Fin-13B	-0.0398	-0.0397	-0.0389	-0.0325	-0.0965	-0.0962	-0.0943	-0.0784	-0.0189	-0.0188	-0.0184	-0.0153
GPT-4o-mini	0	0	-0.0004	-0.0029	0	0	-0.0004	-0.0029	0	0	-0.0001	-0.0007

Table 3: Performance Consistency Ratios (PCR) under Different Scale Functions and α Values: (1) While the choice of scale function does not significantly impact the final contamination determination, the $tanh$ function provides more suitable output ranges. (2) The value of α generally does not significantly impact the final determination.

notes that these models were trained on a combination of instruction datasets, including FinQA. Additionally, it confirms CAP's effectiveness in distinguishing between fine-tuning and dataset contamination, where baseline methods were insufficient, as shown in Table 2.

Although AlphaFin is a newer benchmark, our analysis suggests that the FinMA models may have been indirectly exposed to the AlphaFin training set. A deeper investigation into AlphaFin revealed that its QA component includes the FinQA benchmark. This raises concerns about building composite benchmarks from older datasets. LLM performance on new benchmarks may be inflated due to prior exposure to overlapping data between the older datasets and the new composite benchmark.

5.3 ECTSum Results and Analysis

For the ECTSum benchmark, there is insufficient evidence to verify whether it has contaminated the models. Most LLMs show $PCR_{train-val}$ and $PCR_{train-test}$ values close to zero, indicating either no contamination has occurred or both sets have been similarly contaminated. However, LLaMA shows a possible issue, with its values suggesting it may have been fine-tuned on the training set and possibly contaminated by the test set. Since the LLaMA paper (Touvron et al., 2023) doesn't provide detailed information on this, further investigation might be necessary.

5.4 Ablation Studies

We conducted ablation studies to examine the impact of different function settings and values of α on the performance of our method. The functions tested for scaling included $tanh$, $linear$ (w/o $tanh$), and $sigmoid$. The values of α explored were 0, 0.001, 0.01, and 0.1. While the results shown in Table 3 focus on the FinEval benchmark, similar trends were observed across other tasks. we have the following observations:

- The value of α generally does not affect the final determination of whether a dataset is contaminated. Across the range of α values (0, 0.001, 0.01, and 0.1), the PCR value changes minimally. However, for tasks with smaller consistency values, such as ECTSum, we recommend using a smaller α (e.g., 0.001), while for other tasks, $\alpha = 0.01$ is preferable.

- Similarly, the choice of scale function does not significantly influence the final determination. However, using the $tanh$ function results in more appropriate output value ranges, making it the preferred choice. In contrast, $linear$ (without $tanh$) tends to produce unbounded output ranges $(-\infty, +\infty)$, while $sigmoid$ constrains the output to a narrow range $(0, 1)$. Therefore, we selected $tanh$ (which outputs within $(-1, 1)$) as the default scale function.

6 Conclusion

In this work, we introduced CAP, a novel framework for detecting data contamination by leveraging the Performance Consistency Ratio (PCR). CAP is versatile and can be applied across a wide range of tasks and models. We are the first to explicitly investigate contamination in domain-specific LLMs, and our experiments across multiple LLMs and benchmarks demonstrate CAP's effectiveness in distinguishing between fine-tuning and contamination. We also emphasize the urgent need for increased awareness among researchers and the community as new benchmarks are developed. Specifically, (1) some models may achieve inflated performance on new composite benchmarks due to pretraining or fine-tuning on older overlapping datasets; and (2) benchmarks derived from public exams or textbooks are particularly susceptible to unintentional overlap and data leakage.

Limitations

While we have demonstrated that our method effectively detects benchmark-level contamination in domain-specific LLMs, there are two main limitations that suggest promising research directions:

Fin-LLMs Coverage Due to computational constraints, we were unable to evaluate larger models such as InvestLM-65B, which is a well-known Fin-LLM. While we included FinGPT in our study, we found that its available open-source checkpoints are primarily optimized for sentiment analysis tasks, offering limited responses to financial knowledge and reasoning questions. As a result, our experiments did not include these two LLMs.

Contamination Focus Furthermore, our method primarily addresses contamination at the benchmark level. It does not focus on detecting contamination at the individual sample level. Future work may explore more granular contamination detection methods at the sample level.

Ethical Considerations

Our research focuses on public datasets and does not involve the use of sensitive data. We evaluate Fin-LLMs and detect financial benchmark contamination. These models and benchmarks are not intended for real-world financial investment decisions, and misuse could result in significant financial risks. We strongly advise against using them in such contexts without appropriate guidance.

References

Victor Agostinelli, Max Wild, Matthew Raffel, Kazi Ahmed Asif Fuad, and Lizhong Chen. 2024. Simul-llm: A framework for exploring high-quality simultaneous translation with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 10530–10541. Association for Computational Linguistics.

Baichuan-inc. 2023. Baichuan-13b. https://github.com/baichuan-inc/Baichuan-13B.

Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, and Christopher D. Manning. 2024. Biomedlm: A 2.7b parameter language model trained on biomedical text. *Preprint*, arXiv:2403.18421.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mark Chen, Jerry Tworek, and et al. 2021a. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Wei Chen, Qiushi Wang, and et al. 2023. Disc-finllm: A chinese financial large language model based on multiple experts fine-tuning. *ArXiv*, abs/2310.15205.

Zhiyu Chen, Wenhu Chen, and et al. 2021b. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3697–3711. Association for Computational Linguistics.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *Preprint*, arXiv:2306.16092.

Jasper Dekoninck, Mark Niklas Müller, Maximilian Baader, Marc Fischer, and Martin Vechev. 2024. Evading data contamination detection for language models is (too) easy. *Preprint*, arXiv:2402.02823.

Chunyuan Deng, Yilun Zhao, and et al. 2024. Investigating data contamination in modern benchmarks for large language models. In *NAACL*, pages 8706–8719. Association for Computational Linguistics.

Yihong Dong, Xue Jiang, and et al. 2024a. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics, ACL 2024*, pages 12039–12050. Association for Computational Linguistics.

Yihong Dong, Xue Jiang, and et al. 2024b. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. *ArXiv*, abs/2402.15938.

Shahriar Golchin and Mihai Surdeanu. 2024. Data contamination quiz: A tool to detect and estimate contamination in large language models. *Preprint*, arXiv:2311.06233.

Dan Hendrycks, Collin Burns, and et al. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, and et al. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Yiming Huang, Zhenghao Lin, and et al. 2024. Competition-level problems are effective LLM evaluators. In *Findings of the Association for Computational Linguistics, ACL 2024*, pages 13526–13544. Association for Computational Linguistics.

Yuzhen Huang, Yuzhuo Bai, and et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. 2022. BECEL: benchmark for consistency evaluation of language models. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*, pages 3680–3696. International Committee on Computational Linguistics.

Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 5591–5606. Association for Computational Linguistics.

Bin Lei, Yuchen Li, and Qiuwu Chen. 2024. Autocoder: Enhancing code large language model with AIEV-INSTRUCT. *Preprint*, arXiv:2405.14906.

Changmao Li and Jeffrey Flanigan. 2024. Task contamination: Language models may not be few-shot anymore. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI*, pages 18471–18480. AAAI Press.

Xiang Li, Zhenyu Li, Chen Shi, Yong Xu, Qing Du, Mingkui Tan, Jun Huang, and Wei Lin. 2024. Alphafin: Benchmarking financial analysis with retrieval-augmented stock-chain framework. *ArXiv*, abs/2403.12582.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Surpassing gpt-4 on conversational qa and rag. *Preprint*, arXiv:2401.10225.

Igor A Mel'čuk and Aleksandr K Žolkovskij. 1970. Towards a functioning'meaning-text'model of language.

Rajdeep Mukherjee, Abhinav Bohra, and et al. 2022. Ectsum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, pages 10893–10906. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Manley Roberts, Himanshu Thakur, Christine Herlihy, Colin White, and Samuel Dooley. 2023. Data contamination through the lens of time. *Preprint*, arXiv:2310.10628.

Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. 2024. Towards faithful and robust LLM specialists for evidence-based question-answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 1913–1931. Association for Computational Linguistics.

Yewei Song, Cedric Lothritz, Xunzhu Tang, Tegawendé F. Bissyandé, and Jacques Klein. 2024. Revisiting code similarity evaluation with abstract syntax tree edit distance. In *ACL 2024 - Student Research Workshop*, pages 38–46. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Qianqian Xie, Weiguang Han, and et al. 2023. PIXIU: A comprehensive benchmark, instruction dataset and large language model for finance. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. 2024a. Benchmark data contamination of large language models: A survey. *Preprint*, arXiv:2406.04244.

Liang Xu, Hai Hu, and et al. 2020. CLUE: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pages 4762–4772. International Committee on Computational Linguistics.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. 2024b. Benchmarking benchmark leakage in large language models. *Preprint*, arXiv:2404.18824.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023a. Rethinking benchmark and contamination for language models with rephrased samples. *CoRR*, abs/2311.04850.

Yi Yang, Yixuan Tang, and Kar Yan Tam. 2023b. Investlm: A large language model for investment using financial domain instruction tuning. *Preprint*, arXiv:2309.13064.

Liwen Zhang, Wei Cai, and et al. 2023. Fineval: A chinese financial domain knowledge evaluation benchmark for large language models. *ArXiv*, abs/2308.09975.

A Result Appendix

Detailed experimental results are presented in the following tables. Table 4 provides the results for the FinEval benchmark, while Table 5 reports the PCR calculation results for FinEval. Table 6 shows the results for FinQA, and Table 7 presents the corresponding PCR calculations. Similarly, Table 8 reports the results for AlphaFin, with the PCR calculations detailed in Table 9. Finally, Table 10 provides the results for ECTSum, and Table 11 presents the PCR calculation results for this benchmark.

Dev Set (Dataset Size: 170)	Surface Level: EM		Fact Level: Correctness		Consistency
Model	Original	Modified	Original	Modified	
Baichuan-13B	0.4235	0.4412	0.4529	0.4529	0.4765
Disc-Fin-13B	0.4647	0.4059	0.4941	0.4176	0.4824
FinMA-NLP-7B	0.1176	0.1647	0.2588	0.2941	0.2765
FinMA-Full-7B	0.1824	0.1353	0.3059	0.2529	0.2706
LLaMA-8B	0	0	0.5647	0.4941	0.6059
Mistral-7B	0	0	0.4176	0.4000	0.5294
GPT-4o-mini	0	0	0.7059	0.6588	0.7706
Val Set (Dataset Size: 1151)	Surface Level: EM		Fact Level: Correctness		Consistency
Model	Original	Modified	Original	Modified	
Baichuan-13B	0.4544	0.4309	0.4909	0.4396	0.4674
Disc-Fin-13B	0.4327	0.3884	0.4613	0.3997	0.4083
FinMA-NLP-7B	0.1095	0.1512	0.3023	0.2789	0.2407
FinMA-Full-7B	0.1706	0.1647	0.3189	0.2485	0.2502
LLaMA-8B	0	0	0.4631	0.4518	0.5308
Mistral-7B	0	0	0.4292	0.4005	0.4944
GPT-4o-mini	0	0.0009	0.6551	0.6699	0.7489

Table 4: FinEval Results

B Sample Appendix

Figures 5 and 6 present the original and modified data samples for each benchmark.

Model	Dev			Val			Drop
	EM	Consistency	PCR	EM	Consistency	PCR	
Baichuan	0.4235	0.4765	0.7119	0.4544	0.4674	0.7499	-0.038
Disc-Fin	0.4647	0.4824	0.7461	0.4327	0.4083	0.785	**-0.0389**
FinMA-NLP	0.1176	0.2765	0.4181	0.1095	0.2407	0.4436	-0.0255
FinMA-Full	0.1824	0.2706	0.5952	0.1706	0.2502	0.6006	-0.0054
LLaMA	0	0.6059	0.0162	0	0.5308	0.0185	-0.0023
Mistral	0	0.5294	0.0185	0	0.4944	0.0198	-0.0013
GPT-4o-mini	0	0.7706	0.0128	0	0.7489	0.0132	-0.0004

Table 5: PCR calculation for the FinEval benchmark using the $\tanh(\cdot)$ function and $\alpha = 0.01$

Model	**Train Set (Sample Size: 500)**			**Validation Set (Sample Size: 500)**			**Test Set (Sample Size: 500)**		
	Original	Modified	Consistency	Original	Modified	Consistency	Original	Modified	Consistency
LLaMA-8B	0.0021	0.0022	0.8326	0.0021	0.0022	0.83	0.0022	0.0024	0.8246
Mistral-7B	0.0209	0.022	0.5554	0.0229	0.0225	0.5679	0.0201	0.0205	0.5656
FinMA-Full-7B	0.2304	0.2181	0.727	0.0943	0.0835	0.5662	0.1084	0.1016	0.5706
FinMA-NLP-7B	0.1481	0.1391	0.6993	0.092	0.0866	0.638	0.106	0.106	0.6219
GPT-4o-mini	0.0209	0.0217	0.4562	0.0217	0.0206	0.4604	0.0208	0.0207	0.4505
	Train Set (Sample Size: 100)			Validation Set (Sample Size: 100)			Test Set (Sample Size: 100)		
Baichuan-13B	0.0084	0.0078	0.2506	0.01	0.0131	0.2546	0.0081	0.0068	0.2466
Disc-Fin-13B	0.0097	0.009	0.512	0.0118	0.0135	0.5554	0.0144	0.0159	0.4796

Table 6: FinQA Results

Model	Train			Validation			Test			Drop_Train_Val	Drop_Train_Test
	Rouge-L	Consistency	PCR	Rouge-L	Consistency	PCR	Rouge-L	Consistency	PCR		
LLaMA-8B	0.0021	0.8326	0.0144	0.0021	0.83	0.0144	0.0022	0.8246	0.0147	-0.0001	-0.0003
Mistral-7B	0.0209	0.5554	0.0546	0.0229	0.5679	0.0569	0.0201	0.5656	0.0523	-0.0023	0.0023
FinMA-Full-7B	0.2304	0.727	0.3151	0.0943	0.5662	0.1791	0.1084	0.5706	0.2011	**0.136**	**0.1139**
FinMA-NLP-7B	0.1481	0.6993	0.2193	0.092	0.638	0.1561	0.1060	0.6219	0.1816	0.0633	0.0377
GPT-4o-mini	0.0209	0.4562	0.0662	0.0217	0.4604	0.0672	0.0208	0.4505	0.0667	-0.001	-0.0005
Baichuan-13B	0.0084	0.2506	0.0703	0.01	0.2546	0.0756	0.0081	0.2466	0.0704	-0.0053	-0.0001
Disc-Fin-13B	0.0097	0.512	0.0377	0.0118	0.5554	0.0385	0.0144	0.4796	0.0497	-0.0009	-0.0121

Table 7: PCR calculation for the FinQA benchmark using the $\tanh(\cdot)$ function and $\alpha = 0.01$

C Prompt Appendix

Table 12 shows the prompt used with `claude-3.5-sonnet-20240620` to generate semantically equivalent text for the ECTSum dataset.

D Overlap Appendix

Examples of benchmark overlap between FinEval and C-Eval are shown in Figure 8.

	Train Set (Sample Size: 500)							
	BLEU		Rouge-1		Rouge-2		Rouge-L	
	Original	Modified	Original	Modified	Original	Modified	Original	Modified
LLaMA-8B	0.0002	0.0002	0.0089	0.009	0.0009	0.0009	0.0088	0.0089
Mistral-7B	0.0047	0.0047	0.126	0.1275	0.0205	0.0204	0.1252	0.1267
FinMA-Full-7B	0.048	0.048	0.779	0.745	0.242	0.232	0.779	0.745
FinMA-NLP-7B	0.048	0.048	0.5221	0.5188	0.192	0.188	0.5221	0.5188
Baichuan-13B	0.0038	0.0038	0.1128	0.1106	0.0093	0.0092	0.1127	0.1105
Disc-Fin-13B	0.0043	0.0043	0.0883	0.0877	0.0073	0.0075	0.0883	0.0877
GPT-4o-mini	0.0221	0.0195	0.2548	0.2599	0.0406	0.0366	0.2545	0.2591
	Validation Set (Sample Size: 500)							
	BLEU		Rouge-1		Rouge-2		Rouge-L	
	Original	Modified	Original	Modified	Original	Modified	Original	Modified
LLaMA-8B	0.0003	0.0003	0.0095	0.009	0.001	0.001	0.0093	0.0088
Mistral-7B	0.0024	0.0024	0.1263	0.1293	0.0122	0.0126	0.1245	0.1275
FinMA-Full-7B	0.0474	0.0474	0.5795	0.5842	0.1514	0.1494	0.5792	0.5839
FinMA-NLP-7B	0.0566	0.0566	0.4399	0.4385	0.1482	0.1482	0.4395	0.4381
Baichuan-13B	0.003	0.003	0.0953	0.0904	0.0089	0.0091	0.0951	0.0902
Disc-Fin-13B	0.0012	0.0012	0.0798	0.0789	0.0121	0.0121	0.0798	0.0789
GPT-4o-mini	0.01	0.0109	0.2563	0.2702	0.0195	0.0207	0.2556	0.27

Table 8: AlphaFin Results

Model	Train			Validation			Drop
	Rouge-L	Consistency	PCR	Rouge-L	Consistency	PCR	
LLaMA	0.0088	0.7919	0.0234	0.0093	0.7702	0.0247	-0.0013
Mistral	0.1252	0.6838	0.1924	0.1245	0.6581	0.1987	-0.0062
FinMA-Full	0.7790	0.8964	0.7016	0.5792	0.8884	0.5756	**0.1261**
FinMA-NLP	0.5221	0.8873	0.5321	0.4395	0.8905	0.4615	0.0706
Baichuan	0.1127	0.5760	0.2063	0.0951	0.5513	0.1851	0.0213
Disc-Fin	0.0883	0.7381	0.1307	0.0798	0.7321	0.1204	0.0103
GPT-4o-mini	0.2545	0.5417	0.4457	0.2556	0.5472	0.4436	0.0022

Table 9: PCR calculation for the AlphaFin benchmark using the $\tanh(\cdot)$ function and $\alpha = 0.01$

Model	Sample Size	ROUGE-1		ROUGE-2		ROUGE-L	
		Original	Modified	Original	Modified	Original	Modified
Train							
LLaMA-8B	100	0.0709	0.0713	0.0281	0.0257	0.0505	0.0482
Mistral-7B	517	0.131	0.1247	0.0493	0.0441	0.0834	0.0775
FinMA-NLP-7B	517	0.0288	0.031	0.0013	0.0015	0.0269	0.0279
FinMA-Full-7B	517	0.0277	0.0361	0.0024	0.0023	0.0248	0.0317
Baichuan-13B	100	0.0485	0.0441	0.0067	0.0043	0.0359	0.0329
Disc-Fin-13B	517	0.026	0.0283	0.0038	0.0053	0.0241	0.0261
GPT-4o-mini	517	0.1472	0.1437	0.0467	0.0435	0.0898	0.0868
Validation							
LLaMA-8B	100	0.0676	0.0642	0.0253	0.0229	0.0475	0.0448
Mistral-7B	101	0.1391	0.129	0.048	0.0448	0.0855	0.0768
FinMA-NLP-7B	101	0.0185	0.0275	0.0001	0.0008	0.0173	0.0252
FinMA-Full-7B	101	0.0311	0.0318	0.002	0.0025	0.0279	0.0275
Baichuan-13B	100	0.0559	0.0492	0.0074	0.0057	0.0387	0.0345
Disc-Fin-13B	101	0.0237	0.0176	0.0034	0.0038	0.022	0.0168
GPT-4o-mini	101	0.1499	0.1511	0.0456	0.0467	0.0884	0.0887
Test							
LLaMA-8B	100	0.06	0.0594	0.0221	0.021	0.0411	0.0407
Mistral-7B	250	0.1222	0.1205	0.0443	0.0411	0.0784	0.0754
FinMA-NLP-7B	250	0.0284	0.0288	0.0011	0.0013	0.0261	0.0265
FinMA-Full-7B	250	0.031	0.0343	0.0023	0.0021	0.0264	0.03
Baichuan-13B	100	0.0413	0.0462	0.0044	0.0054	0.0315	0.0329
Disc-Fin-13B	250	0.0308	0.0238	0.0055	0.0047	0.0284	0.0226
GPT-4o-mini	250	0.1413	0.1373	0.0419	0.0409	0.0854	0.0826

Table 10: ECTSum Results

Model	Train			Validation			Test			Drop_Train_Val	Drop_Train_Test
	Rouge-2	Consistency	PCR	Rouge-2	Consistency	PCR	Rouge-2	Consistency	PCR		
LLaMA-8B	0.0281	0	1	0.0253	0.0200	0.8491	0.0221	0	1	**0.1509**	0
Mistral-7B	0.0493	0.0116	0.9993	0.0480	0.0099	0.9998	0.0443	0.016	0.9904	-0.0004	0.0089
FinMA-Full-7B	0.0024	0.029	0.1119	0.002	0.0396	0.0738	0.0023	0.036	0.0903	0.0382	0.0216
FinMA-NLP-7B	0.0013	0.0445	0.0507	0.0001	0.0396	0.0272	0.0011	0.028	0.0730	0.0235	**-0.0223**
Baichuan-13B	0.0067	0	1	0.0074	0	1	0.0044	0	1	0	0
Disc-Fin-13B	0.0038	0.0522	0.0907	0.0034	0.0693	0.0632	0.0055	0.0600	0.1054	0.0276	-0.0147
GPT-4o-mini	0.0467	0.0077	1	0.0456	0.0099	0.9996	0.0419	0.012	0.9973	0.0004	0.0027

Table 11: PCR calculation for the ECTSum benchmark using the $\tanh(\cdot)$ function and $\alpha = 0.001$

Factual Consistency Data Sample

Input:
"请回答选择题，从选项A，B，C，D中选择，回答问题。
问题：____不是金融衍生工具的特征之一。
选项：['A':依赖于原生性金融工具，'B'：表现为一种合约，'C':合约上载明交易品种、价格、数量、交割时间及地点等，'D':可以作为支付工具进行交易]
答案："

Input:
"Please answer the multiple-choice question by choosing from options A, B, C, D.
Question: Which of the following is NOT a characteristic of a financial derivative?
Options:
['A': Dependent on underlying financial instruments,
 'B': Manifests as a contract,
 'C': The contract specifies the type of trade, price, quantity, delivery time and location, etc.,
 'D': Can be used as a payment instrument for transactions]
Answer:"

Output: "D"

Input:
"请回答选择题，从选项A，B，C，D中选择，回答问题。
问题：____不是金融衍生工具的特征之一。
选项：['A'：可以作为支付工具进行交易，'B'：合约上载明交易品种、价格、数量、交割时间及地点等，'C'：依赖于原生性金融工具，'D'：表现为一种合约]
答案："

Input:
"Please answer the multiple-choice question by choosing from options A, B, C, D.
Question: Which of the following is NOT a characteristic of a financial derivative?
Options:
['A': Can be used as a payment instrument for transactions,
  'B': The contract specifies the type of trade, price, quantity, delivery time and location, etc.,
  'C': Dependent on underlying financial instruments,
  'D': Manifests as a contract]
Answer:"

Output: "A"

(a) Sample of Factual-Consistency-Based Data Modification

Semantic Consistency Data Sample

Input:
"Please summarize the content of the context into bullet points and provide only the bullet point summary. Context:President and CEO, Dennis Vermillion; Executive Vice President, Treasurer and CFO, Mark Thies; Senior Vice President, External Affairs and Chief Customer Officer, Kevin Christie; and Vice President, Controller and Principal Accounting Officer, Ryan Krasselt. Our consolidated earnings for the fourth quarter of 2020 were $0.85 per diluted share, compared to $0.76 for the fourth quarter of 2019. For the full year, consolidated earnings were $1.90 per diluted share for 2020 compared to $2.97 last year. Now, I'll turn the discussion over to Dennis. As we begin 2021 continuing to work through the COVID pandemic, we hope you're staying safe and healthy. [...] Summary: "

Output:
"q4 earnings per share $0.85. earnings guidance of $1.96 to $2.16 per diluted share in 2021. earnings guidance of $2.18 to $2.38 per diluted share in 2022. earnings guidance of $2.42 to $2.62 per diluted share in 2023. expect to experience increased costs in 2021 associated with exploring strategic business opportunities."

Input:
"Please summarize the content of the context into bullet points and provide only the bullet point summary. Context:Dennis Vermillion holds the positions of President and CEO, Mark Thies serves as Executive Vice President, Treasurer and CFO, Kevin Christie is the Senior Vice President of External Affairs and Chief Customer Officer, and Ryan Krasselt occupies the role of Vice President, Controller and Principal Accounting Officer. Our diluted share earnings for Q4 2020 were $0.85, an increase from $0.76 in Q4 2019. The company's full-year diluted share earnings decreased from $2.97 in 2019 to $1.90 in 2020. I'll now hand over to Dennis to continue the discussion. As we enter 2021 amid the ongoing COVID-19 pandemic, we hope you're maintaining your health and safety. [...] Summary: "

Output:
"q4 earnings per share $0.85. earnings guidance of $1.96 to $2.16 per diluted share in 2021. earnings guidance of $2.18 to $2.38 per diluted share in 2022. earnings guidance of $2.42 to $2.62 per diluted share in 2023. expect to experience increased costs in 2021 associated with exploring strategic business opportunities."

(b) Sample of Semantic-Consistency-Based Data Modification

Figure 5: Samples of Factual-Consistency-Based and Semantic-Consistency-Based Data Modification

Logical Consistency Data Sample (FinQA)

Input:
"Please answer the given financial question based on the context.
Context: performance graph comparison of five-year cumulative total return the following graph and table compare the cumulative total return on citi 2019s common stock , which is listed on the nyse under the ticker symbol 201cc 201d and held by 65691 common stockholders of record as of january 31 , 2018 , [...] comparison of five-year cumulative total return for the years ended date citi s&p 500 financials.
date	citi	s&p 500	s&p financials
31-dec-2012	100.0	100.0	100.0
31-dec-2013	131.8	132.4	135.6
31-dec-2014	137.0	150.5	156.2
31-dec-2015	131.4	152.6	153.9
31-dec-2016	152.3	170.8	188.9
31-dec-2017	193.5	208.1	230.9	.
Question: what was the percentage cumulative total return for the five year period ended 31-dec-2017 of citi common stock? Answer:"

Output: "93.5%"

Input:
"Please answer the given financial question based on the context.
Context: performance graph comparison of five-year cumulative total return the following graph and table compare the cumulative total return on citi 1979s common stock , which is listed on the nyse under the ticker symbol 201cc 201d and held by 65691 common stockholders of record as of january 31 , 1978 , [...] comparison of five-year cumulative total return for the years ended date citi s&p 500 financials .
date	citi	s&p 500	s&p financials
31-dec-1972	100.0	100.0	100.0
31-dec-1973	131.8	132.4	135.6
31-dec-1974	137.0	150.5	156.2
31-dec-1975	131.4	152.6	153.9
31-dec-1976	152.3	170.8	188.9
31-dec-1977	193.5	208.1	230.9	.
Question: what was the percentage cumulative total return for the five year period ended 31-dec-1977 of citi common stock? Answer:"

Output: "93.5%"

(a) Sample of Logical-Consistency-Based Data Modification (FinQA)

Logical Consistency Data Sample (AlphaFin)

Input:
Given the data and tweets, could you project whether the closing price of $t will grow or shrink at 2020-05-19? Please specify either Rise or Fall.Context:
date,open,high,low,close,adj-close,inc-5,inc-10,inc-15,inc-20,inc-25,inc-30
2020-05-05,0.0,1.5,-0.1,0.5,0.5,1.6,1.2,1.6,1.7,0.2,-0.3
2020-05-06,3.6,3.8,0.0,-3.2,-3.2,3.2,4.3,4.7,4.9,3.6,3.2
2020-05-07,1.0,1.1,-0.2,0.3,0.3,1.7,3.7,4.0,4.4,3.4,2.9
[...]
2020-05-15,-0.1,0.5,-0.8,-0.9,-0.9,1.2,2.5,4.3,4.7,5.3,5.0
2020-05-18,-1.6,0.8,-1.8,4.0,4.0,-2.7,-1.6,0.0,0.3,1.1,1.2
2020-05-05: the dnc deliberately took away americans access to universal health care, [...]
2020-05-06: tnc coin airdrop - claim free $tnc tokens (~$ 265) with crypto crypto news [...]
Answer:"

Output: "Fall"

Input:
"Given the data and tweets, could you project whether the closing price of $t will grow or shrink at 2049-05-19? Please specify either Rise or Fall.Context:
date,open,high,low,close,adj-close,inc-5,inc-10,inc-15,inc-20,inc-25,inc-30
2049-05-05,0.0,1.5,-0.1,0.5,0.5,1.6,1.2,1.6,1.7,0.2,-0.3
2049-05-06,3.6,3.8,0.0,-3.2,-3.2,3.2,4.3,4.7,4.9,3.6,3.2
2049-05-07,1.0,1.1,-0.2,0.3,0.3,1.7,3.7,4.0,4.4,3.4,2.9
[...]
2049-05-15,-0.1,0.5,-0.8,-0.9,-0.9,1.2,2.5,4.3,4.7,5.3,5.0
2049-05-18,-1.6,0.8,-1.8,4.0,4.0,-2.7,-1.6,0.0,0.3,1.1,1.2
2049-05-05: the dnc deliberately took away americans access to universal health care, [...]
2049-05-06: tnc coin airdrop - claim free $tnc tokens (~$ 265) with crypto crypto news [...]
Answer:"

Output: "Fall"

(b) Sample of Logical-Consistency-Based Data Modification (AlphaFin)

Figure 6: Samples of Logical-Consistency-Based Data Modification

Prompt	Your task is to paraphrase sentences. The input consists of sentences separated by '\n.' You need to rewrite each sentence in different words while retaining the same meaning. Then, you should output only the paraphrased sentences. Remember not to include phrases like "the rewritten sentences are..." Also, do not include indexes like 1, 2, 3; just separate them with '\n.' The input sentences are: *Second quarter adjusted income from continuing operations per diluted share increased to $0.82, up nearly 37% from the year ago quarter.* *We generated significant operating leverage with adjusted EBITDA improving 17% year-over-year to $106.6 million and adjusted EBITDA margin increasing 100 basis points to 7.1% on slightly higher revenues.* ... The output is:
Output	*Adjusted income from ongoing operations per diluted share in the second quarter rose to $0.82, an increase of almost 37% compared to the same period last year.* *We achieved considerable operational efficiency, with adjusted EBITDA growing 17% year-over-year to $106.6 million and adjusted EBITDA margin expanding 100 basis points to 7.1% on marginally higher revenues.* ...

Table 12: Prompt for generating paraphrased texts with the Claude model, version `claude-3.5-sonnet-20240620`.

	Train	Val	Test
Min	0.9791	0.9804	0.9844
Mean	0.9984	0.9984	0.9979
Max	0.9999	0.9999	0.9999

Table 13: Cosine Similarity between Original and Modified ECTSum Texts using Longformer Encoding.

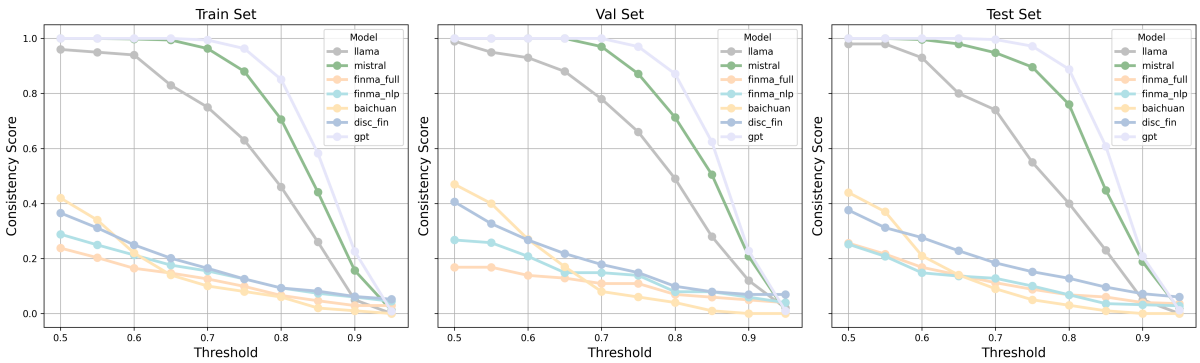


Figure 7: The consistency of LLMs' generation before and after input text modification varies with changes in the similarity threshold. As the threshold increases, the consistency decreases.

1. Sample in FinEval Dev
甲公司是国内一家领先的新媒体、通信及移动增值服务公司。由于遭受世界金融危机，甲公司经济利润严重下滑，经营面临困境，但为了稳定职工队伍，公司并未进行裁员，而是实行高层管理人员减薪措施。甲公司此举采用的收缩战略方式是____。
['A':转向战略, 'B': 放弃战略, 'C':紧缩与集中战略, 'D':稳定战略]
Translation:

Company A is a leading domestic new media, communication, and mobile value-added service company. Due to the global financial crisis, Company A's economic profit has severely declined, and it is facing operational difficulties. However, in order to stabilize its workforce, the company did not lay off employees but instead implemented salary cuts for senior management. The contraction strategy adopted by Company A is ____.

['A': Diversion Strategy, 'B': Abandonment Strategy, 'C': Retrenchment and Focus Strategy, 'D': Stability Strategy]

2. Sample in FinEval Val

甲公司平价发行5年期的公司债券，债券票面利率为10$\%$，每半年付息一次，到期一次偿还本金。该债券的有效年利率是____。

['A':0.1, 'B': 0.1025, 'C':0.105, 'D':0.095]

Trnaslation:

Company A issued 5-year corporate bonds at par, with a coupon rate of 10%, interest paid semi-annually, and the principal repaid at maturity. The effective annual interest rate of this bond is ____.

['A': 0.1, 'B': 0.1025, 'C': 0.105, 'D': 0.095]

3. Sample in FinEval Val

2013年6月1日，人民法院受理了对甲公司提起的破产申请。根据企业破产法律制度的规定，下列人员中，有资格担任管理人的是____。

['A':曾于2008年1月1日至2009年12月31日担任甲公司法律顾问的丙律师事务所, 'B': 甲公司董事丁, 'C':3年前被吊销执业证书，但现已重获执业资格的注册会计师乙, 'D':甲公司监事会主席的妻子戊]

Translation:

On June 1, 2013, the People's Court accepted the bankruptcy petition filed against Company A. According to the corporate bankruptcy legal system, the following person is eligible to serve as the administrator ____.

['A': Law firm Bing, which served as Company A's legal adviser from January 1, 2008, to December 31, 2009, 'B': Company A's director Ding, 'C': CPA Yi, whose license was revoked three years ago but has now regained qualification, 'D': Wu, wife of the chairman of Company A's supervisory board]

Figure 8: Benchmark Overlap