ChameleonAttack: Semantics-Preserving Adversarial Attacks on Event-Driven Stock Prediction Models

Aofan Liu[1,2], Li Haoxuan[3], Hongjian Xing[2], Yuguo Yin[2], Zijun Li[4], and Yiyan Qi ✉[1]

[1]International Digital Economy Academy (IDEA)
[2]School of Electronic and Computer Engineering, Peking University
[3]Shenzhen International Graduate School, Tsinghua University
[4]School of Cyberspace Security, Beijing University of Posts and Telecommunications

Abstract

Adversarial Security of Financial Language Models (ASFLM) is critical as Large Language Models (LLMs) pervade high-stakes financial applications. However, LLMs face two key challenges: their vulnerability to damaging adversarial attacks and the prevalent research gap concerning robust defenses against sophisticated, semantically coherent threats. To address these, we first theoretically analyze the relationship between discrete and continuous adversarial optimization, proving the continuous optimum provides a lower bound for the discrete. This foundation supports our novel two-stage framework, ChameleonAttack. It employs Adaptive Latent-Space Optimization (ALO) for potent adversarial token discovery, followed by a Semantic-Translation Module (STM) module to generate fluent, coherent, and natural-sounding adversarial text. This dual approach aims to maximize attack impact while ensuring high linguistic quality and semantic integrity for evasion. Evaluated on state-of-the-art financial LLMs (e.g., FinBERT) and standard benchmarks (e.g., Financial PhraseBank), ChameleonAttack achieves a high Attack Success Rate (ASR) of 93.4%. These results highlight significant practical vulnerabilities and underscore the urgent need for robust defense mechanisms in the financial domain.

1 Introduction

Large Language Models (LLMs) are increasingly pivotal in the financial sector for tasks such as market sentiment analysis and stock prediction (Wang et al., 2024a). However, this integration brings significant security challenges, as their susceptibility to adversarial attacks can lead to severe consequences, including manipulated financial decisions and systemic market risks, a concern underscored by real-world incidents (Yuan et al., 2024). This situation highlights an urgent need to investigate and bolster the robustness of these financial LLMs

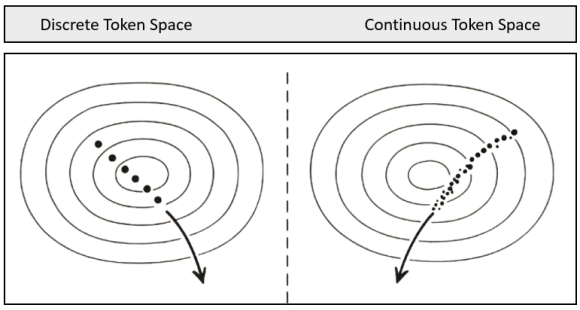

Figure 1: Discrete vs. continuous optimization for adversarial attacks. (Left) Discrete token optimization is challenging due to non-differentiability and a vast search space. (Right) Continuous relaxation allows efficient, gradient-based minimization of an adversarial objective, with solutions then mapped back to discrete tokens.

Current financial LLM research often prioritizes predictive accuracy over security, and many existing adversarial attacks typically lack the semantic coherence or naturalness required for stealth (Joshi et al., 2019; Koa et al., 2024). To effectively generate such sophisticated and evasive adversarial examples, a deeper understanding of the underlying optimization challenges is necessary. The direct optimization of adversarial token sequences in the discrete vocabulary space is an NP-hard problem due to its vast combinatorial nature and the non-differentiability of token selection, as illustrated in Figure 1 (Left).

To address this, we first theoretically analyze the relationship between this intractable discrete adversarial optimization and its continuous relaxation. As detailed in Appendix A, we formally prove that the optimal solution achievable in the continuous space provides a rigorous lower bound for the discrete optimum ($\mathcal{L}_C^* \leq \mathcal{L}_D^*$). This theoretical foundation (Figure 1, Right) validates our strategy of leveraging gradient-based methods in a continuous domain, which are then carefully mapped back to discrete tokens.

Building on this foundation, we introduce **ChameleonAttack**, a novel two-stage framework for generating effective and semantics-preserving adversarial attacks against financial LLMs. The first stage, **Adaptive Latent-space Optimization (ALO)**, utilizes gradient-based techniques with an adaptive sparsification strategy to discover potent adversarial token sequences. The second stage, **Semantic Translation Module**, then employs a dedicated language model to transform these (potentially conspicuous) token sequences into fluent, natural-sounding, and contextually coherent adversarial text. This dual architecture is designed to maximize adversarial impact while maintaining high linguistic quality and semantic integrity for evasion, achieving a 93.4% Attack Success Rate (ASR) against state-of-the-art financial language models.

This dual-stage architecture is meticulously designed to ensure that the generated attacks are not only highly effective in achieving their adversarial goals but also maintain exceptional linguistic quality and semantic integrity, rendering them difficult to detect by both automated systems and human evaluators.

Our contributions are as follows:

1. We systematically analyze and empirically demonstrate significant vulnerabilities in existing financial LLMs when subjected to sophisticated, semantics-preserving adversarial attacks. Our work also quantifies the associated risks within crucial financial forecasting and analysis tasks.

2. We provide theoretical justification for our continuous optimization approach by formally establishing the relationship between discrete and continuous adversarial optimization search spaces, proving that the continuous optimum lower-bounds the discrete one (see Appendix A).

3. We propose **ChameleonAttack**, a novel two-stage framework leveraging **Adaptive Latent-space Optimization** for effective adversarial token generation and **Semantic Translation Module** for ensuring linguistic stealth and coherence. Extensive experiments demonstrate ChameleonAttack achieves a high Attack Success Rate (93.4%) on financial LLMs, setting a new benchmark for sophisticated attacks and underscoring the urgent need for robust defenses.

2 Related Work

2.1 LLM Alignment

The rapid advancement of Large Language Models (LLMs) has made their alignment with human values and ethics a critical issue (Carlini et al., 2024). The academic community has proposed various techniques to enhance LLM safety (Askell et al., 2021; Ouyang et al., 2022; Bai et al., 2022; Bianchi et al., 2023). Some strategies involve using high-quality, value-laden training data to guide LLM behavior. Others refine training methodologies through Supervised Fine-Tuning (SFT), Reinforcement Learning from Human Feedback (RLHF), and adversarial training to ensure outputs align with human expectations (Wang et al., 2023; Lee et al., 2024; Qi et al., 2023). Despite these efforts, completely eliminating harmful content generation by LLMs remains a significant challenge.

Prompt-based Jailbreak Previously, LLM alignment and pre-deployment security testing were often evaluated through manual "jailbreak" attacks (Russinovich et al., 2024; Chao et al., 2024; Anil et al., 2024). However, manual methods are inefficient, difficult to scale, and often lack diversity.

Automated Jailbreak Automated jailbreak methods induce LLMs to produce inappropriate output by crafting meticulously designed prompts with semantic-level deception or by using gradient-based methods for token optimization (Yu et al., 2024). The primary advantage of such methods is the use of natural language for attack commands, facilitating comprehension and cross-platform operation (Liao and Sun, 2024; Liu et al., 2024; Yu et al., 2024; Zhang and Wei, 2024; Zou et al., 2023). While many automated jailbreak techniques are considered "white-box" attacks (i.e., requiring access to internal model parameters), their attack strategies and the generated adversarial prompts can sometimes be transferable, posing a threat to less robust closed-source LLMs or inspiring attack approaches for black-box models.

2.2 Event Driven Stock Prediction

Advancements in Natural Language Processing (NLP) have enabled the use of textual data for stock market forecasting (Du et al., 2024). Researchers

have explored methods like using tweets and historical prices for prediction, modeling multi-modal financial data, and extracting granular insights such as corporate events or media sentiment to understand market dynamics (Wang et al., 2024b; Obst et al., 2021; Xu and Cohen, 2018; Zolfagharinia et al., 2024). These approaches generally aim to improve prediction accuracy by comprehensively analyzing textual information.

Separately, Large Language Models (LLMs), like GPT variants, possess extensive knowledge but are not inherently designed for time-series analysis (Cao et al., 2024). Efforts are underway to adapt LLMs for such tasks, including prompt-based methods that convert numerical data into textual formats for LLMs to process (Jia et al., 2024; Lam et al., 2024). In finance, this involves using prompts for LLMs to generate summaries or keyphrases from various data sources to aid forecasting. While insightful, these methods can sometimes suffer from overly broad prompts leading to less detailed outputs from the LLMs (Li et al., 2024a; Wang et al., 2024a; Koa et al., 2024).

3 Methodology

Our method for generating semantics-preserving adversarial attacks is a two-stage process. The first stage focuses on optimizing an adversarial token sequence in a continuous space and then converting it back to discrete tokens. The second stage employs a translation model to transform this optimized token sequence into coherent, natural-sounding adversarial text, designed to be effective yet inconspicuous. The following mathematical optimization is based on the principle that the continuous space of adversarial attack samples provides a lower bound for the discrete space (A proof of this principle is available in the Appendix A.)

3.1 Stage 1: Adversarial Optimization

The primary goal of this stage is to identify a sequence of tokens (an adversarial suffix) that, when appended to a benign prompt, maximizes the likelihood of the target model generating an undesired output. This involves defining the adversarial objective and then using continuous optimization techniques to make the search tractable.

3.1.1 Discrete Adversarial Objective

The core of the adversarial attack lies in identifying an optimal adversarial suffix, denoted as $s = (s_1, \ldots, s_N)$, composed of N discrete tokens from the model's vocabulary $\mathcal{V}$ (OpenAI et al., 2024). Each token is typically represented as a one-hot vector within the set $\mathcal{T}_D$. The optimization goal is to find a suffix s that maximizes the likelihood of the LLM generating the desired target sequence $y = (y_1, \ldots, y_M)$, given the initial prompt x and the adversarial suffix s. This is typically achieved by minimizing the cross-entropy (CE) loss, as formulated in the discrete objective function $\mathcal{L}_D$:

$$\min_{s_1,\ldots,s_N \in \mathcal{T}_D} \mathcal{L}_D(\{s_j\}_{j=1}^N) = \sum_{k=1}^{M} CE(LLM(x_{1:L_X} \oplus s_{1:N} \oplus y_{1:k-1}), y_k) \tag{1}$$

where $\oplus$ signifies sequence concatenation.

However, directly optimizing this objective function $\mathcal{L}_D$ (Equation 1) within the discrete token space $\mathcal{T}_D^N$ presents a significant computational hurdle. The non-differentiable nature of token selection, combined with the vast combinatorial search space (determined by vocabulary size $|\mathcal{V}|$ and suffix length N), renders direct discrete optimization intractable (Anil et al., 2024; Bailey et al., 2024; Chao et al., 2024) . This necessitates a more sophisticated approach.

3.1.2 Continuous Relaxation and Optimization

To overcome the limitations of discrete optimization, we transition the problem into a continuous domain(Yin et al., 2025). This involves relaxing the discrete token representation by utilizing $\mathcal{T}_C$, the probability simplex in $\mathbb{R}^{|\mathcal{V}|}$, which we define as the Continuous Token Space:

Definition 1 (Continuous Token Space). $\mathcal{T}_C = \{\omega \in \mathbb{R}^{|\mathcal{V}|} | \omega[i] \geq 0 \text{ for all } i, \sum_{i=1}^{|\mathcal{V}|} \omega[i] = 1\}$

In this continuous space, the adversarial suffix is represented as $a = (\alpha_1, \ldots, \alpha_N)$, where each $\alpha_j \in \mathcal{T}_C$. The optimization problem is then reformulated as minimizing a continuous objective function $\mathcal{L}_C$:

$$\min_{\alpha_1,\ldots,\alpha_N \in \mathcal{T}_C} \mathcal{L}_C(\{\alpha_j\}_{j=1}^N) = \sum_{k=1}^{M} CE(LLM(x_{1:L_X} \oplus \alpha_{1:N} \oplus y_{1:k-1}), y_k) \tag{2}$$

Once an optimal continuous solution $\{\alpha_j^*\}$ is found, it must be mapped back to the discrete token space for practical application. A straightforward mapping approach, such as Argmax Projection

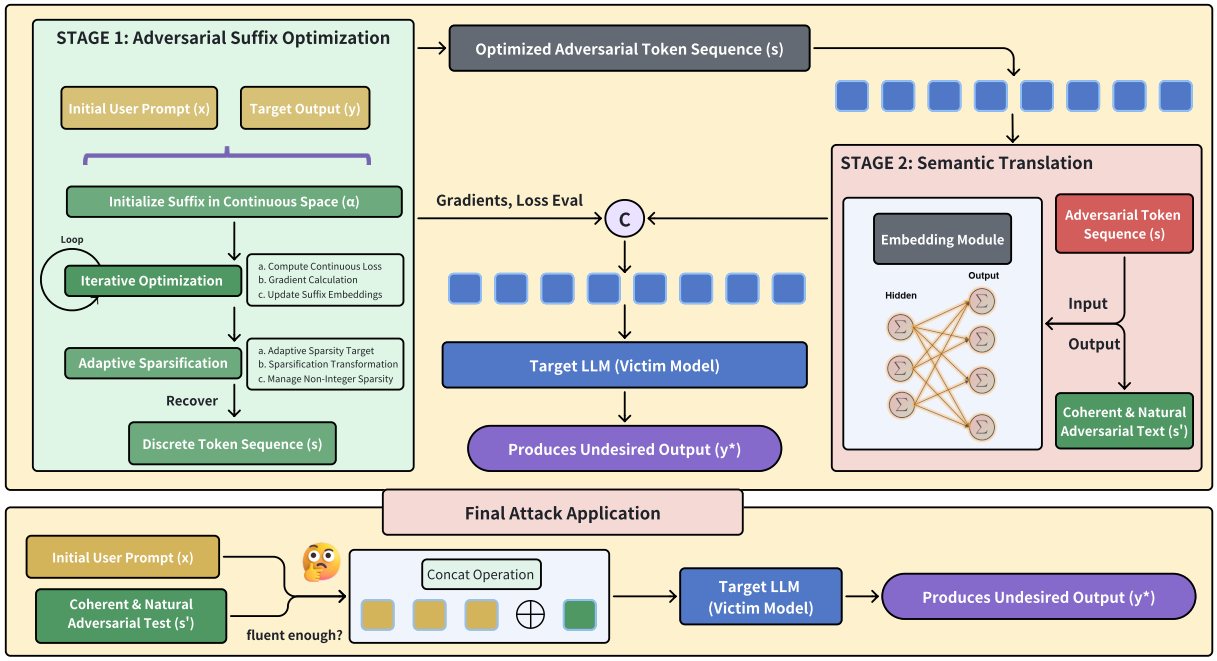

Figure 2: Overall architecture of the ChameleonAttack framework. Stage 1 employs a Continuous Optimization Engine with Adaptive Sparsification to generate an optimized (but potentially incoherent) adversarial token sequence (s). Stage 2 utilizes a Semantic Translation Module, leveraging a fine-tuned T5 model, to transform s into coherent and natural adversarial text (s').

(Π_{argmax}), often proves suboptimal due to the "Projection Impasse".

3.1.3 Adaptive Sparsification for Discrete Token Recovery

To address the Projection Impasse, we introduce an Adaptive Sparsification Strategy designed to guide the continuous token representations α_j towards sparser forms, facilitating a more effective mapping to discrete tokens. This strategy dynamically adjusts the sparsity of the continuous vectors based on the attack's performance.

Adaptive Sparsity Target The desired sparsity, $\mathcal{S}$, adapts based on an error measure $E(\{\alpha_j\})$. The sparsity target is defined as:

$$\mathcal{S}(\{\alpha_j\}) = \exp\left(\sum_{k=1}^{M} \mathbb{I}\left(y_k \text{ is mispredicted by LLM for } x \oplus \{\alpha_j\} \oplus y_{1:k-1}\right)\right) \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function. As error decreases, $\mathcal{S}$ approaches 1, encouraging 1-sparse representations (Hu et al., 2025).

Sparsification Transformation A transformation $\Psi_{\mathcal{S}_{target}} : \mathbb{R}^{|\mathcal{V}|} \to \mathcal{T}_C$ is applied. This involves ReLU application, identifying the $\mathcal{S}_{target}$-th largest value (δ), creating a sparse vector ω_{sparse}, and normalization:

$$\omega_{sparse}[i] = \begin{cases} x'[i] + \epsilon_{stab} & \text{if } x'[i] \geq \delta \\ 0 & \text{if } x'[i] < \delta \end{cases} \tag{4}$$

$$\Psi_{\mathcal{S}_{target}}(\omega) = \frac{\omega_{sparse}}{\sum_{k=1}^{|\mathcal{V}|} \omega_{sparse}[k]} \tag{5}$$

Managing Non-Integer Sparsity For non-integer $\mathcal{S}_{val}$ (from Equation 3), sparsity is stochastically applied using $S_{floor} = \lfloor \mathcal{S}_{val} \rfloor$, $S_{ceil} = \lceil \mathcal{S}_{val} \rceil$, and $p = \mathcal{S}_{val} - S_{floor}$. This ensures the expected number of non-zero components for α_j is $\mathcal{S}_{val}$:

$$E[\text{sparsity}] = p \cdot S_{ceil} + (1-p) \cdot S_{floor} = \mathcal{S}_{val} \tag{6}$$

This process guides α_j towards 1-sparse forms as $\mathcal{S}_{val} \to 1$.

3.2 Stage 2: Semantic Translation of Adversarial Sequences

Following the optimization of the adversarial token sequence s in the first stage, the second stage of ChameleonAttack focuses on enhancing the attack's stealth and practical applicability. The raw optimized token sequence, while effective in manipulating the target LLM's output, may not be

human-readable or could appear as nonsensical gibberish. Such overtly anomalous inputs are likely to be detected by human oversight or automated defense mechanisms.

To mitigate this, we employ a semantic translation module. This module takes the discrete adversarial token sequence $\{s_j\}_{j=1}^{N}$ generated in Stage 1 and translates it into coherent, natural-sounding text. The objective of this translation is twofold:

1. **Preserve Adversarial Impact:** The translated text must retain the adversarial properties of the original token sequence, ensuring it still guides the target LLM to the intended undesired output.

2. **Ensure Semantic Coherence and Naturalness:** The output text should be grammatically correct, semantically meaningful, and contextually appropriate. It should read like a human-written statement, thereby evading casual detection and appearing as a legitimate input modification.

By converting the optimized but potentially unnatural token sequence into fluent and semantically sound text, this stage aims to create adversarial perturbations that are not only effective but also highly challenging to detect, thereby increasing their potency in real-world scenarios. The specifics of the translation model (e.g., architecture, training data) are chosen to ensure high-fidelity translation while maintaining the adversarial utility.

4 Experimental Result

4.1 Experiment Setup

Our experiments leverage three distinct datasets widely employed in sentiment analysis within the financial and news domains:

Twitter News Sentiment: Consists of 9.54k tweets pertaining to news events, annotated for positive, negative, and neutral sentiment polarity (zeroshot, 2023).

Stock Emotions: Comprises 10k text excerpts from social media and financial forums discussing stock market activities, labeled as bullish or bearish (Lee et al., 2023).

Financial PhraseBank: Contains about 5k sentences from English-language financial news reports, annotated by financial experts for positive, negative, or neutral sentiment from an investor's perspective. For the event-driven stock prediction tasks, these datasets are utilized to derive textual features and corresponding market event signals. For agent-based evaluations, queries and contexts are grounded in financial scenarios reflective of the information contained within these datasets (Malo et al., 2014).

Example: Financial PhraseBank

Pharmaceuticals group Orion Corp reported a fall in its third-quarter earnings that were hit by larger expenditures on R&D and marketing. ——*Negative*

4.2 Adversarial Perturbation Generation

The adversarial texts employed throughout our experiments are generated via the two-stage attack model delineated in Section 3 of this paper. This model first utilizes gradient optimization to identify adversarial token sequences and subsequently employs a translation model to convert these sequences into coherent, natural-sounding text. The core design principle is to preserve semantic coherence while maximizing the adversarial impact.

4.3 Evaluation Metric

Standard metrics for classification and prediction tasks are employed, including accuracy, F1-score, precision, and recall. For evaluating attack efficacy, we primarily focus on the degradation of these metrics. The Attack Success Rate (ASR) is generally defined as the proportion of attempts where an attacker successfully subverts a model's intended output or alignment. Our definition of ASR, consistent with HADES(Li et al., 2024b), for a given dataset D is:

$$ASR = \frac{\sum_i \mathbb{I}(Q_i)}{|D|} \tag{7}$$

where Q_i represents an individual query within the dataset D, and the indicator function $\mathbb{I}$ returns 1 if the model's response to Q_i is classified as a successful compromise, and 0 otherwise. An elevated ASR suggests a higher vulnerability of the model, indicating that its protective measures are more frequently circumvented by attackers.

4.4 Attack Result

Our ChameleonAttack methodology demonstrates significant efficacy, as detailed in Table 1. It sub-

Model	Attack Method	Twitter News Sentiment			Stock Emotions		Financial PhraseBank		
		Positive	Negative	Neutral	Bullish	Bearish	Positive	Negative	Neutral
BERT	TextFooler	12.3	11.6	11.2	10.7	9.1	9.8	10.4	9.3
	AutoPrompt	22.4	21.7	21.3	20.5	20.1	20.6	20.8	21.5
	GCG Attack	71.6	70.2	70.8	68.4	69.5	68.1	69.7	70.3
	Momentum	77.7	76.5	76.7	74.6	75.3	74.5	75.7	76.0
	AmpleGCG	81.0	79.6	79.9	77.6	78.5	77.8	78.8	79.4
	ChameleonAttack	**90.5**	**89.2**	**89.6**	**88.7**	**88.3**	**88.2**	**89.9**	**90.1**
FinBERT	TextFooler	8.7	7.2	7.9	6.3	6.8	6.6	6.1	7.5
	AutoPrompt	18.6	17.1	17.9	16.3	16.4	16.8	17.2	17.7
	GCG Attack	76.3	75.8	75.1	73.6	74.4	73.2	74.9	75.4
	Momentum	82.5	81.6	81.5	79.5	80.5	78.9	81.2	81.4
	AmpleGCG	85.6	84.8	84.8	82.5	83.9	82.1	84.3	84.7
	ChameleonAttack	**93.4**	**92.1**	**92.5**	**91.3**	**91.6**	**91.7**	**92.6**	**93.3**
FinGPT	TextFooler	10.6	9.4	9.7	8.1	8.6	8.8	8.3	9.6
	AutoPrompt	20.8	19.3	19.8	18.7	18.2	18.9	19.6	19.1
	GCG Attack	78.4	77.7	77.1	75.3	76.6	75.8	76.5	77.3
	Momentum	84.5	83.6	83.4	81.0	82.8	81.8	82.9	83.1
	AmpleGCG	87.5	86.8	86.5	83.9	86.1	84.9	86.1	86.1
	ChameleonAttack	**92.3**	**91.8**	**91.5**	**89.6**	**90.4**	**89.2**	**90.7**	**91.2**
RoBERTa	TextFooler	15.2	14.8	14.1	13.4	12.7	12.3	13.6	12.5
	AutoPrompt	25.7	24.6	24.3	23.5	23.1	23.8	23.2	24.9
	GCG Attack	61.4	60.9	60.2	58.8	59.6	58.1	59.9	60.4
	Momentum	67.4	67.1	66.0	65.1	65.3	64.2	66.3	66.3
	AmpleGCG	70.7	70.2	69.2	68.5	68.5	67.7	69.4	69.6
	ChameleonAttack	**86.7**	**85.3**	**85.6**	**84.8**	**84.5**	**84.9**	**85.1**	**86.2**

Table 1: Attack Success Rate (ASR) for various adversarial attack strategies, including TextFooler (Jin et al., 2020), AutoPrompt (Shin et al., 2020), GCG Attack (Zou et al., 2023), AmpleGCG (Liao and Sun, 2024), Momentum (Zhang and Wei, 2024) and ChameleonAttack.

stantially outperforms contemporary baseline methods across various financial LLMs (BERT, FinBERT, FinGPT, RoBERTa) and datasets, achieving Attack Success Rates (ASR) exceeding 91% on models like FinBERT (e.g., 93.4% on Twitter News Sentiment, Positive category "). This underscores the potency of our Adaptive Latent-space Optimization (ALO) stage in identifying effective adversarial sequences.

Furthermore, Table 2 showcases ChameleonAttack's impact on complex AI financial agents (FinRobot (Zhou et al., 2024), ForcastLLM (Wang et al., 2024c), Self-Reflective LLM (Koa et al., 2024) utilizing different base models (Llama 3.1-8B, Qwen3-8B, Falcon-7B). These agents exhibited substantial performance degradation across key metrics like Accuracy, Recall, and F1 Score. For instance, the Qwen3-8B based FinRobot experienced an accuracy drop from 89.4% to 36.2% (ASR of 53.2%, indicating a severe reduction in accuracy). This effectiveness against sophisticated agent-based systems highlights the practical threat posed by our generated attacks, likely enhanced by the coherence and naturalness imparted by our Semantic Translation Module stage.

4.5 Discussion

The collective results from Table 1 and Table 2 confirm the high efficacy and broad applicability of our ChameleonAttack framework. Its success stems from the two-stage design, where Adaptive Latent-space Optimization (ALO) discovers potent adversarial tokens, and Semantic Translation Module subsequently refines them into fluent, natural-sounding text. This synergy is crucial for generating attacks that are not only effective but also exceptionally stealthy.

The significant ASRs achieved against foundational LLMs, coupled with the substantial performance degradation inflicted upon complex AI agents (with accuracy drops exceeding 50 percentage points for some Qwen3-8B configurations as seen in Table 2), underscore a critical vulnerability in current financial AI systems. These findings compellingly argue for an urgent shift in focus within the financial LLM development lifecycle, moving beyond an exclusive emphasis on task accuracy to vigorously incorporate and prioritize adversarial robustness. Future research should concentrate on developing robust defenses against such sophisticated, semantically coherent attacks,

Agent Type	Base Model	Accuracy		Recall		F1 Score		Attack Success Rate
		Before Attack	After Attack	Before Attack	After Attack	Before Attack	After Attack	
FinRobot	Llama - 3.1 - 8B	77.3%	29.6%	0.73	0.34	0.75	0.41	47.7%
	Qwen3 - 8B	89.4%	36.2%	0.87	0.43	0.88	0.51	53.2%
	Falcon - 7B	56.8%	19.3%	0.55	0.22	0.57	0.31	37.5%
ForcastLLM	Llama - 3.1 - 8B	74.6%	26.8%	0.70	0.28	0.72	0.38	47.8%
	Qwen3 - 8B	86.7%	33.9%	0.84	0.40	0.85	0.47	52.8%
	Falcon - 7B	53.4%	16.7%	0.52	0.20	0.54	0.29	36.7%
Self-Reflective LLM	Llama - 3.1 - 8B	80.5%	33.5%	0.78	0.37	0.80	0.45	47.0%
	Qwen3 - 8B	90.2%	38.2%	0.88	0.41	0.90	0.50	52.0%
	Falcon - 7B	60.1%	22.1%	0.58	0.25	0.60	0.33	38.0%

Table 2: Comparative performance metrics (Accuracy, Recall, F1 Score) for various AI agents and their base language models, measured before and after adversarial attack, alongside achieved Attack Success Rates (ASR).

further investigating their transferability, and continuously assessing the evolving threat landscape.

4.6 Defense Testing

To highlight ChameleonAttack's potency, we conducted experiments to assess its effectiveness against established defense mechanisms. The goal was to see if ChameleonAttack's semantically coherent and natural perturbations could bypass defenses effective against simpler attacks. We focused on the FinBERT model and the Financial PhraseBank (FPB) dataset, using Attack Success Rate (ASR) as the key metric, comparing ChameleonAttack to GCG Attack and TextFooler. Defenses tested included perplexity filtering, a pre-trained adversarial detector, and an adversarially trained version of FinBERT.

Results are summarized in Table 3:

- **No Defense:** ChameleonAttack achieved 92.5% ASR, compared to 75.4% for GCG Attack and 7.0% for TextFooler on an undefended FinBERT model.
- **Perplexity Filtering:** Reduced TextFooler's ASR to 5.2% and GCG Attack's to 60.5%, but ChameleonAttack maintained 88.0% ASR due to its Semantic Translation Module.
- **Adversarial Detector:** Detected 80% of TextFooler and 45% of GCG Attack, but only 15% of ChameleonAttack, allowing 85.0% ASR.
- **Adversarially Trained Model (FinBERT-AT):** ChameleonAttack achieved 75.0% ASR, indicating that adversarial training may not defend against adaptive attacks like ChameleonAttack without specific inclusion.

Defense Mechanism	Attack Method	ASR (%) ↑	Detect Rate (%) ↓
No Defense (Baseline)	**ChameleonAttack**	92.5	N/A
	GCG Attack	75.4	N/A
	TextFooler	7.0	N/A
Perplexity Filtering	**ChameleonAttack**	88.0	∼10[a]
	GCG Attack	60.5	∼35
	TextFooler	5.2	∼70
Adversarial Detector	**ChameleonAttack**	85.0	15
	GCG Attack	55.0	45
	TextFooler	4.5	80
Adversarially Trained (AT)	**ChameleonAttack**	75.0	N/A
	GCG Attack	40.0	N/A
	TextFooler	2.0	N/A

[a] Illustrative rate of inputs flagged as unnatural by perplexity filtering.

Table 3: ChameleonAttack's evasion capabilities against defenses on FinBERT (FPB Avg. ASR). ASR on Defended Model shown. Detection Rate for detectors.

5 Ablation Study

We performed an ablation study to quantify the contributions of key components in our two-stage attack method, with results on the FinBERT model (Financial Phrase Bank dataset) detailed in Table 4. Metrics include Attack Success Rate (ASR), naturalness, and semantic similarity.

Naturalness is assessed via LLM-generated Mean Opinion Scores (MOS, 1-5; detailed criteria in Appendix B), and semantic similarity (0-1) is the cosine similarity of original versus adversarial sentence embeddings from a pre-trained transformer.

Our **Full Method (Proposed)** achieves 92.5% ASR with high naturalness (4.5 naturalness) and semantic similarity (0.85). Removing the Stage 2 Semantic Translation Module (STM) ("Full Method w/o STM") maintained a high ASR (93.1%) but resulted in extremely low naturalness (1.3 naturalness) and semantic similarity (0.25), underscoring the STM's necessity for generating practical, stealthy attacks.

Ablating Stage 1's Adaptive Sparsification Strategy (ASS) also revealed its importance. Using a "Naive Argmax Projection" instead of ASS dropped ASR to 72.8%, while a "Fixed Sparsity Target"

Method Configuration	ASR (%) on FinBERT (FPB Avg.) ↑	Naturalness ↑	Semantic Similarity (0-1) ↑
Full Method (Proposed)	**92.5**	**4.5**	**0.85**
Ablating Stage 2: Semantic Translation Module (STM)			
Full Method w/o STM (Direct use of optimized tokens)	93.1[a]	1.3	0.25[b]
Ablating Stage 1: Adaptive Sparsification Strategy (ASS)			
Full Method w/o ASS (Naive Argmax Projection)	72.8	4.2	0.80
Full Method w. Fixed Sparsity Target (e.g., $S = 10$)	81.5	4.3	0.81
Ablating Stage 1: Core Optimization Approach			
Full Method w. Random Search for Suffix (instead of Gradient Opt.)	12.3	3.9[c]	0.65[d]

[a]ASR might appear marginally higher without STM if the raw optimized tokens are highly effective but lack naturalness; this often means the attack is more easily detectable.

[b]Semantic similarity for raw, potentially incoherent tokens is inherently low or ill-defined when compared to natural language expectations or a benign reference.

[c]Naturalness depends on STM's ability to salvage a potentially poor token sequence from random search.

[d]Semantic similarity is low as the random suffix lacks optimized adversarial intent related to any specific context.

Table 4: Ablation study of our proposed two-stage attack method. Experiments are conducted on the FinBERT model using the Financial Phrase Bank (FPB) dataset (ASR averaged across polarities). Performance is evaluated by Attack Success Rate (ASR), perceived naturalness (range from 1-5, higher is better), and semantic similarity (0-1 scale, higher indicates better preservation of original context if applicable, or general coherence).

achieved 81.5% ASR. These results highlight the superiority of our adaptive approach for effective discrete token recovery, with the STM aiding in maintaining output quality.

Finally, replacing our gradient-based continuous optimization in Stage 1 with a "Random Search for Suffix" caused ASR to plummet to 12.3%. This confirms the critical role of sophisticated continuous optimization in discovering effective adversarial perturbations, even as the STM worked to ensure some level of naturalness in the output.

In essence, Table 4 demonstrates that each component of our proposed method—gradient-guided continuous optimization, adaptive sparsification, and semantic translation—is integral to achieving both high attack efficacy and the crucial characteristics of naturalness and coherence in the generated adversarial examples.

6 Conclusion

In this paper, we have addressed the escalating security concerns surrounding the deployment of Large Language Models (LLMs) in critical financial text analysis and forecasting applications. Recognizing the profound risks posed by adversarial manipulations in high-stakes scenarios like algorithmic trading and market sentiment analysis, we introduced a novel two-stage adversarial attack methodology. Our approach distinctively integrates gradient-based optimization in a continuous latent space to discover potent adversarial token sequences, with a subsequent semantic translation stage that refines these sequences into fluent, natural-sounding, and contextually coherent adversarial text.

Our extensive empirical evaluations demonstrate that this methodology achieves a significant Attack Success Rate (ASR) of 93.4% against a range of contemporary financial language models. More critically, the generated adversarial perturbations exhibit high linguistic quality, making them difficult to detect through superficial inspection and thus posing a more insidious threat than traditional, often less coherent, attack vectors. These findings systematically verify and highlight substantial vulnerabilities in existing Finance LLMs when applied to financial forecasting tasks.

Limitations

The two-stage adversarial attack method also has several limitations. It requires substantial computational resources, particularly in the gradient-based optimization and semantic translation stages, making it challenging for resource-limited environments. The method's success also depends on the quality of the semantic translation, with risks of adversarial signal loss or unnatural phrasing that could trigger detection.

The white-box assumption limits its generalization to black-box models, restricting applicability to closed-source systems. Additionally, the operational complexity of the pipeline, including the need for extensive hyperparameter tuning, complicates deployment and error diagnosis.

Finally, evaluating semantic preservation and naturalness remains subjective and difficult to scale. Extending the method to other perturbation strategies and domains requires further adaptation.

Addressing these challenges will enhance the method's robustness and applicability.

Ethinic Consideration

We acknowledge the dual-use nature of this work. While our primary goal is to expose critical vulnerabilities and thereby motivate the development of stronger security measures, the techniques described could potentially be exploited by malicious actors. Given the increasing integration of LLMs in high-stakes financial applications, such misuse could lead to significant economic disruption or undermine trust in AI-driven financial systems. The capacity of our attack to generate effective, stealthy, and semantically coherent adversarial examples heightens these risks.

Our intent in pursuing and publishing this research is to proactively advance the security of financial AI. We believe that a transparent and detailed understanding of sophisticated attack vectors is essential for the creation of robust and adaptive defenses. By demonstrating the capabilities of advanced attacks that mimic legitimate human language, we aim to provide the community with clear insights into the threats that modern financial LLMs face, urging a shift beyond mere predictive accuracy to a strong emphasis on adversarial robustness.

References

Cem Anil, Esin Durmus, Nina Rimsky, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel J. Ford, Francesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrishnan, Carson Denison, Evan J. Hubinger, and 15 others. 2024. Many-shot Jailbreaking. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, and 1 others. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. 2024. Image Hijacks: Adversarial Images can Control Generative Models at Runtime. *Preprint*, arXiv:2309.00236.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2023. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*.

Defu Cao, Furong Jia, Sercan O. Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2024. TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting. *Preprint*, arXiv:2310.04948.

Nicholas Carlini, Milad Nasr, Christopher A. Choquette-Choo, Matthew Jagielski, Irena Gao, Anas Awadalla, Pang Wei Koh, Daphne Ippolito, Katherine Lee, Florian Tramer, and Ludwig Schmidt. 2024. Are aligned neural networks adversarially aligned? *Preprint*, arXiv:2306.15447.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2024. Jailbreaking Black Box Large Language Models in Twenty Queries. *Preprint*, arXiv:2310.08419.

Kelvin Du, Rui Mao, Frank Xing, and Erik Cambria. 2024. Explainable Stock Price Movement Prediction using Contrastive Learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 529–537, Boise ID USA. ACM.

Kai Hu, Weichen Yu, Yining Li, Kai Chen, Tianjun Yao, Xiang Li, Wenhe Liu, Lijun Yu, Zhiqiang Shen, and Matt Fredrikson. 2025. Efficient LLM Jailbreak via Adaptive Dense-to-sparse Constrained Optimization. *Preprint*, arXiv:2405.09113.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. 2024. GPT4MTS: Prompt-based Large Language Model for Multimodal Time-series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21):23343–23351.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. *Preprint*, arXiv:1907.11932.

Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. 2019. Semantic adversarial attacks: Parametric transformations that fool deep classifiers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4773–4783.

Kelvin J. L. Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2024. Learning to Generate Explainable Stock Predictions using Self-Reflective Large Language Models. In *Proceedings of the ACM Web Conference 2024*, pages 4304–4315.

Michelle S. Lam, Janice Teoh, James A. Landay, Jeffrey Heer, and Michael S. Bernstein. 2024. Concept Induction: Analyzing Unstructured Text with High-Level Concepts Using LLooM. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–28, New York, NY, USA. Association for Computing Machinery.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *Preprint*, arXiv:2309.00267.

Jean Lee, Hoyoul Luis Youn, Josiah Poon, and Soyeon Caren Han. 2023. Stockemotions: Discover investor emotions for financial sentiment analysis and multivariate time series. *Preprint*, arXiv:2301.09279.

Shuqi Li, Yuebo Sun, Yuxin Lin, Xin Gao, Shuo Shang, and Rui Yan. 2024a. CausalStock: Deep end-to-end causal discovery for news-driven multi-stock movement prediction. *Advances in Neural Information Processing Systems*, 37:47432–47454.

Yifan Li, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. 2024b. Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models. *arXiv preprint arXiv:2403.09792*.

Zeyi Liao and Huan Sun. 2024. AmpleGCG: Learning a Universal and Transferable Generative Model of Adversarial Suffixes for Jailbreaking Both Open and Closed LLMs. *Preprint*, arXiv:2404.07921.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. *Preprint*, arXiv:2310.04451.

Pekka Malo, Ankur Sinha, Pirjo Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

David Obst, Joseph De Vilmarest, and Yannig Goude. 2021. Adaptive methods for short-term electricity load forecasting during COVID-19 lockdown in France. *IEEE transactions on power systems*, 36(5):4754–4763.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! *Preprint*, arXiv:2310.03693.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. 2024. Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack. *Preprint*, arXiv:2404.01833.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. *Preprint*, arXiv:2010.15980.

Meiyun Wang, Kiyoshi Izumi, and Hiroki Sakaji. 2024a. LLMFactor: Extracting Profitable Factors through Prompts for Explainable Stock Movement Prediction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3120–3131, Bangkok, Thailand. Association for Computational Linguistics.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. 2024b. From News to Forecast: Integrating Event Analysis in LLM-Based Time Series Forecasting with Reflection. *Preprint*, arXiv:2409.17515.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. 2024c. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Preprint*, arXiv:2409.17515.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. *Preprint*, arXiv:2212.10560.

Yumo Xu and Shay B. Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979.

Yuguo Yin, Yuxin Xie, Wenyuan Yang, Dongchao Yang, Jinghan Ru, Xianwei Zhuang, Liming Liang, and Yuexian Zou. 2025. Atri: Mitigating multilingual audio text retrieval inconsistencies by reducing data distribution errors. *Preprint*, arXiv:2502.14627.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2024. GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts. *Preprint*, arXiv:2309.10253.

Jiaqiang Yuan, Yiyu Lin, Yadong Shi, Tianyi Yang, and Ang Li. 2024. Applications of artificial intelligence generative adversarial techniques in the financial sector. *Academic Journal of Sociology and Management*, 2(3):59–66.

zeroshot. 2023. Twitter financial news sentiment. `https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment`.

Yihao Zhang and Zeming Wei. 2024. Boosting Jailbreak Attack with Momentum. *Preprint*, arXiv:2405.01229.

Tianyu Zhou, Pinqiao Wang, Yilin Wu, and Hongyang Yang. 2024. Finrobot: Ai agent for equity research and valuation with large language models. *Preprint*, arXiv:2411.08804.

Hossein Zolfagharinia, Mehdi Najafi, Shamir Rizvi, and Aida Haghighi. 2024. Unleashing the Power of Tweets and News in Stock-Price Prediction Using Machine-Learning Techniques. *Algorithms*, 17(6):234.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. *Preprint*, arXiv:2307.15043.

A Theoretical Relationship Between Discrete and Continuous Optimization Spaces in Adversarial Suffix Generation

This appendix provides a more formal mathematical exposition on the relationship between the optimal solution achievable in the discrete token space versus its continuous relaxation, as utilized in Stage 1 of our proposed methodology for adversarial suffix generation.

A.1 Formal Problem Definitions

Let $\mathcal{V}$ be the vocabulary of the target Large Language Model (LLM), with $V = |\mathcal{V}|$ denoting its size. An adversarial suffix is a sequence of N tokens, $S = (t_1, t_2, \ldots, t_N)$. The initial user prompt is denoted by $X = (x_1, \ldots, x_{L_X})$ and the desired target output sequence (which the attack aims to elicit) is $Y = (y_1, \ldots, y_M)$.

Definition 1 (Discrete Token Space $\mathcal{T}_D$)**.** *The discrete token space $\mathcal{T}_D$ is the set of all possible one-hot vectors in $\mathbb{R}^V$. Each $s \in \mathcal{T}_D$ corresponds to a unique token in $\mathcal{V}$. The space of all possible discrete N-token suffixes is $\mathcal{T}_D^N$.*

Definition 2 (Continuous Token Space $\mathcal{T}_C$)**.** *The continuous token space $\mathcal{T}_C$ is the probability simplex in $\mathbb{R}^V$:*

$$\mathcal{T}_C = \left\{ \alpha \in \mathbb{R}^V \;\middle|\; \begin{array}{l} \sum_{i=1}^{V} \alpha[i] = 1, \\ \alpha[i] \geq 0 \textit{ for all } i = 1, \ldots, V \end{array} \right\} \tag{8}$$

The space of all possible continuous N-token suffixes is $\mathcal{T}_C^N$.

It is evident that $\mathcal{T}_D \subset \mathcal{T}_C$, as any one-hot vector is a valid point in the probability simplex. Consequently, $\mathcal{T}_D^N \subset \mathcal{T}_C^N$.

Let $\mathbf{E} \in \mathbb{R}^{V \times d}$ be the LLM's token embedding matrix, where d is the embedding dimension. For a discrete one-hot token $s \in \mathcal{T}_D$, its embedding is $e_s = \mathbf{E}^T s$. For a continuous token representation $\alpha \in \mathcal{T}_C$, its effective embedding is $e_\alpha = \mathbf{E}^T \alpha$, representing the expected embedding over the vocabulary distribution defined by α. Note that if α is a one-hot vector s, then $e_\alpha = e_s$.

The LLM, denoted f_{LLM}, takes a sequence of embeddings corresponding to the prompt X, the suffix S_{tokens}, and the already generated target prefix $Y_{1:k-1}$, and outputs a probability distribution over $\mathcal{V}$ for the next token y_k. Let $Emb(X)$, $Emb(S_{tokens})$, $Emb(Y_{1:k-1})$ denote the sequences of embeddings. The loss function for a given suffix (either discrete $\{s_j\}$ or continuous $\{\alpha_j\}$) is typically the negative log-likelihood (or sum of cross-entropies) for generating the target sequence Y:

For a discrete suffix $S_D = (s_1, \ldots, s_N) \in \mathcal{T}_D^N$:

$$\mathcal{L}_D(S_D) = \sum_{k=1}^{M} \mathrm{CE}\Big(f_{LLM}(Emb(X), \{\mathbf{E}^T s_j\}_{j=1}^N, Emb(Y_{1:k-1})), y_k \Big) \tag{9}$$

The optimal discrete loss is $\mathcal{L}_D^* = \min_{S_D \in \mathcal{T}_D^N} \mathcal{L}_D(S_D)$.

For a continuous suffix $S_C = (\alpha_1, \ldots, \alpha_N) \in \mathcal{T}_C^N$:

$$\mathcal{L}_C(S_C) = \sum_{k=1}^{M} \mathrm{CE}\Big(f_{LLM}(Emb(X), \{\mathbf{E}^T \alpha_j\}_{j=1}^N, Emb(Y_{1:k-1})), y_k \Big) \tag{10}$$

The optimal continuous loss is $\mathcal{L}_C^* = \min_{S_C \in \mathcal{T}_C^N} \mathcal{L}_C(S_C)$.

A.2 Relationship Between Optimal Losses

We seek to formally establish the relationship between $\mathcal{L}_D^*$ and $\mathcal{L}_C^*$. The core principle of relaxation suggests that optimizing over a larger (continuous) space should yield a solution at least as good as, or better than, optimizing over a restricted (discrete) subspace.

Theorem 1. *The minimum loss achievable in the continuous token space, $\mathcal{L}_C^*$, provides a lower bound for the minimum loss achievable in the discrete token space, $\mathcal{L}_D^*$. That is:*

$$\mathcal{L}_C^* \leq \mathcal{L}_D^*$$

Proof. Let $S_D^* = (s_1^*, s_2^*, \ldots, s_N^*)$ be an optimal sequence of discrete tokens in $\mathcal{T}_D^N$ such that $\mathcal{L}_D(S_D^*) = \mathcal{L}_D^*$. Each s_j^* is a one-hot vector.

As established, $\mathcal{T}_D \subset \mathcal{T}_C$, which implies that every one-hot vector s_j^* is also a valid point in the probability simplex $\mathcal{T}_C$. Therefore, the optimal discrete sequence S_D^* is also an element of the continuous suffix space $\mathcal{T}_C^N$.

Consider the evaluation of the continuous loss function $\mathcal{L}_C$ at this specific point $S_D^* \in \mathcal{T}_C^N$. The embeddings for each token s_j^* in S_D^* when considered as a continuous representation are $\mathbf{E}^T s_j^*$. This is identical to the discrete embedding for s_j^*. Thus, the computation of the LLM's output probabilities

and subsequently the cross-entropy loss will be identical for S_D^* whether it is evaluated under $\mathcal{L}_D$ or $\mathcal{L}_C$:

$$\begin{aligned}\mathcal{L}_C(S_D^*) = \sum_{k=1}^{M} \mathrm{CE}\Big(& f_{LLM}\big(Emb(X), \\ & \{\mathbf{E}^T s_j^*\}_{j=1}^N, \\ & Emb(Y_{1:k-1})\big), y_k \Big) \\ = \mathcal{L}_D(S_D^*) & \end{aligned} \tag{11}$$

So, we have $\mathcal{L}_C(S_D^*) = \mathcal{L}_D^*$.

The optimal continuous loss $\mathcal{L}_C^*$ is defined as the global minimum of $\mathcal{L}_C(S_C)$ over all possible sequences $S_C \in \mathcal{T}_C^N$. Since S_D^* is one such sequence in $\mathcal{T}_C^N$, the minimum value $\mathcal{L}_C^*$ attained by optimizing over the entire space $\mathcal{T}_C^N$ must be less than or equal to the value of $\mathcal{L}_C$ at any specific point within that space, including S_D^*. Therefore,

$$\mathcal{L}_C^* \leq \mathcal{L}_C(S_D^*)$$

Substituting $\mathcal{L}_C(S_D^*) = \mathcal{L}_D^*$, we arrive at the conclusion:

$$\mathcal{L}_C^* \leq \mathcal{L}_D^*$$

This proves that the optimal loss in the continuous relaxation is indeed a lower bound for the optimal loss in the original discrete problem. Phrased differently, the value of the optimal discrete solution ($\mathcal{L}_D^*$) is lower-bounded by any specific discrete solution's loss and lower-bounded by the optimal continuous solution's loss ($\mathcal{L}_C^*$). □

A.3 Implications for the Optimization Strategy

Theorem 1 provides the theoretical justification for employing continuous relaxation. By expanding the search space, we can leverage gradient-based optimization techniques, which are generally intractable in purely discrete, non-differentiable domains of enormous combinatorial complexity. The solution $S_C^{**} = (\alpha_1^{**}, \ldots, \alpha_N^{**})$ that achieves $\mathcal{L}_C^*$ is found in $\mathcal{T}_C^N$.

However, S_C^{**} is not guaranteed to be in $\mathcal{T}_D^N$ (i.e., the α_j^{**} may not be one-hot). To obtain an actionable discrete adversarial suffix, a projection or mapping operation $\Pi : \mathcal{T}_C^N \to \mathcal{T}_D^N$ is required, such as the adaptive sparsification strategy detailed in Section 3 of the main paper. Let $S_D^{proj} = \Pi(S_C^{**})$.

It is crucial to recognize that the projection step can introduce a gap between the continuous optimum and the achieved discrete loss. That is, while $\mathcal{L}_C(S_C^{**}) = \mathcal{L}_C^* \leq \mathcal{L}_D^*$, it typically holds that:

$$\mathcal{L}_C(S_C^{**}) \leq \mathcal{L}_D(S_D^{proj})$$

The term $\mathcal{L}_D(S_D^{proj})$ may or may not be equal to $\mathcal{L}_D^*$. The objective of sophisticated projection and sparsification techniques is to minimize the increase in loss incurred during this transition from the continuous solution back to a high-performing discrete solution, ideally approaching $\mathcal{L}_D^*$ or at least a competitively low discrete loss. The effectiveness of the overall adversarial attack hinges on how well this "continuous-to-discrete gap" is managed. Our adaptive sparsification strategy is designed precisely to guide the continuous optimization towards solutions that are not only low in continuous loss but are also amenable to effective, low-loss projection into the discrete token space.

B Prompt for Naturalness

Role:
You are an expert linguist and language model evaluator. Your task is to assess the naturalness of the given sentence or paragraph by applying principles of linguistics, fluency, and native-like expression. You will act as a language evaluator, similar to how a human reviewer would rate the naturalness of a passage based on everyday speech or written communication.

Workflow:

1. **Understanding the Context:** Read the provided text and comprehend the content. Pay attention to the intended meaning, tone, and structure of the sentence.
2. **Evaluation Criteria:** After understanding the content, assess the sentence's naturalness based on the following criteria:
   - **Grammar and Syntax:** Is the sentence grammatically correct? Does the structure follow common linguistic rules?
   - **Fluency and Flow:** Does the sentence flow smoothly? Are there any awkward pauses or choppy phrasing?
   - **Tone and Style:** Does the tone of the sentence match the context? Is the style appropriate for the intended audience or situation?
   - **Naturalness:** Does the sentence feel like something a native speaker would typically say or write? Does it sound authentic in the given language context?

3. **Final Rating:** Based on your evaluation, provide a rating on a scale of 1 to 5. The rating should reflect how natural the sentence or paragraph sounds, where:
    - **1**: Very unnatural or awkward, with major grammar or fluency issues.
    - **2**: Unnatural with noticeable awkward phrasing or minor grammatical errors.
    - **3**: Neutral, with acceptable naturalness but not exceptionally fluent or smooth.
    - **4**: Natural with minor issues that do not significantly detract from the overall fluency.
    - **5**: Highly natural, native-like expression, no issues with grammar, flow, or tone.

Rules:

- Focus on the naturalness of language; avoid focusing on the factual accuracy or meaning of the content unless it directly impacts fluency.
- Provide your rating with a brief explanation highlighting the strengths and weaknesses of the sentence, based on the criteria mentioned above.
- If the sentence contains multiple clauses or parts, evaluate the overall cohesiveness and integration of those parts into a fluent whole.
- Avoid any bias based on content or personal opinion; only evaluate based on linguistic factors.
- Ensure that the evaluation is fair, impartial, and thorough.

C Potential Risk

The research of the ChameleonAttack, while intended to highlight critical vulnerabilities in financial Large Language Models (LLMs) and spur the development of robust defenses, inherently carries potential risks. The detailed exposition of our two-stage attack framework, which combines Adaptive Latent-space Optimization (ALO) for potent adversarial token discovery with a Semantic-Translation Module (STM) to ensure linguistic stealth and coherence, could inadvertently equip malicious actors. Given ChameleonAttack's demonstrated high Attack Success Rate (ASR) of up to 93.4% against models like FinBERT and its ability to degrade the performance of complex AI financial agents significantly, its misuse could lead to tangible economic damage, manipulated financial decisions, or systemic market risks.