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

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

[1]IDEA Research, International Digital Economy Academy (IDEA)
[2]School of Electronic and Computer Engineering, Peking University
[3]Shenzhen International Graduate School, Tsinghua University
[4]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) 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.

Introduction

Large Language Models (LLMs) are increasingly pivotal in the financial sector for tasks such as market sentiment analysis and stock prediction (Wang, Izumi, and Sakaji 2024). 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

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).

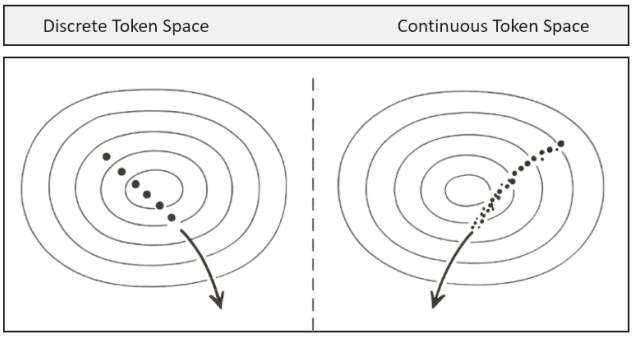

Figure 1: Discrete vs. continuous optimization for adversarial attacks. (Left) Discrete token optimization is challenging due to non-differentiability and a vast search space. (Right) Continuous relaxation allows efficient, gradient-based minimization of an adversarial objective, with solutions then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

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.

Related Work

LLM Alignment

Recent work has studied vulnerabilities and jailbreaks in multimodal large models, such as VisualDAN for visual-driven attacks (Liu and Tang 2025), Pico for pictorial code contextualization (Liu et al. 2025b), automated multi-agent jailbreak frameworks (Yang et al. 2025a), and RefleXGen highlighting code review (Wang et al. 2025). Various methods have been proposed to improve LLM safety (Askell et al. 2021; Ouyang et al. 2022; Bai et al. 2022; Bianchi et al. 2023), including high-quality value-laden data, Supervised Fine-Tuning (SFT), RLHF, and adversarial training (Wang et al. 2023; Lee et al. 2024; Qi et al. 2023), yet fully preventing harmful outputs remains challenging.

Prompt-based Jailbreak Previously, LLM alignment and pre-deployment security testing were often evaluated through manual "jailbreak" attacks (Russinovich, Salem, and Eldan 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.

Multimodal Models and Financial Forecasting

Recent research has advanced techniques in code retrieval, multimodal learning, and model robustness. For instance, Md3r addresses inconsistencies in multilingual query-code retrieval by minimizing data distribution discrepancies (Liu et al. 2025c), and repository-aware dual-encoder models enhance code search with adversarial verification (Liu et al. 2025a). SupCLAP improves audio-text contrastive learning via support vector regularization (Luo et al. 2025), while AdaDocVQA enables adaptive long-document visual question answering in low-resource settings (Li et al. 2025a). InfiJanice focuses on correcting quantization-induced mathematical degradation in LLMs (Li et al. 2025b), LongFaith enhances long-context reasoning using synthetic data (Yang et al. 2025b), and RefleXGen emphasizes the importance of thorough code examination for reliability (Wang et al. 2025).

These advances in multimodal and language-based modeling have direct implications for financial applications. In the stock market domain, NLP techniques allow event-driven prediction by leveraging textual and multimodal data such as news articles, corporate announcements, tweets, and historical prices to anticipate market trends (Du et al. 2024; Wang et al. 2024a; Obst, De Vilmarest, and Goude 2021; Xu and Cohen 2018; Zolfagharinia et al. 2024). Large language models, though not inherently designed for time-series tasks (Cao et al. 2024), can be adapted through prompt-based approaches that convert numeric data and event descriptions into textual summaries or keyphrases to aid forecasting (Jia et al. 2024; Lam et al. 2024). Careful prompt design is essential, as overly broad prompts may reduce the granularity and reliability of the predictions (Li et al. 2024a; Wang, Izumi, and Sakaji 2024; Koa et al. 2024).

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.

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

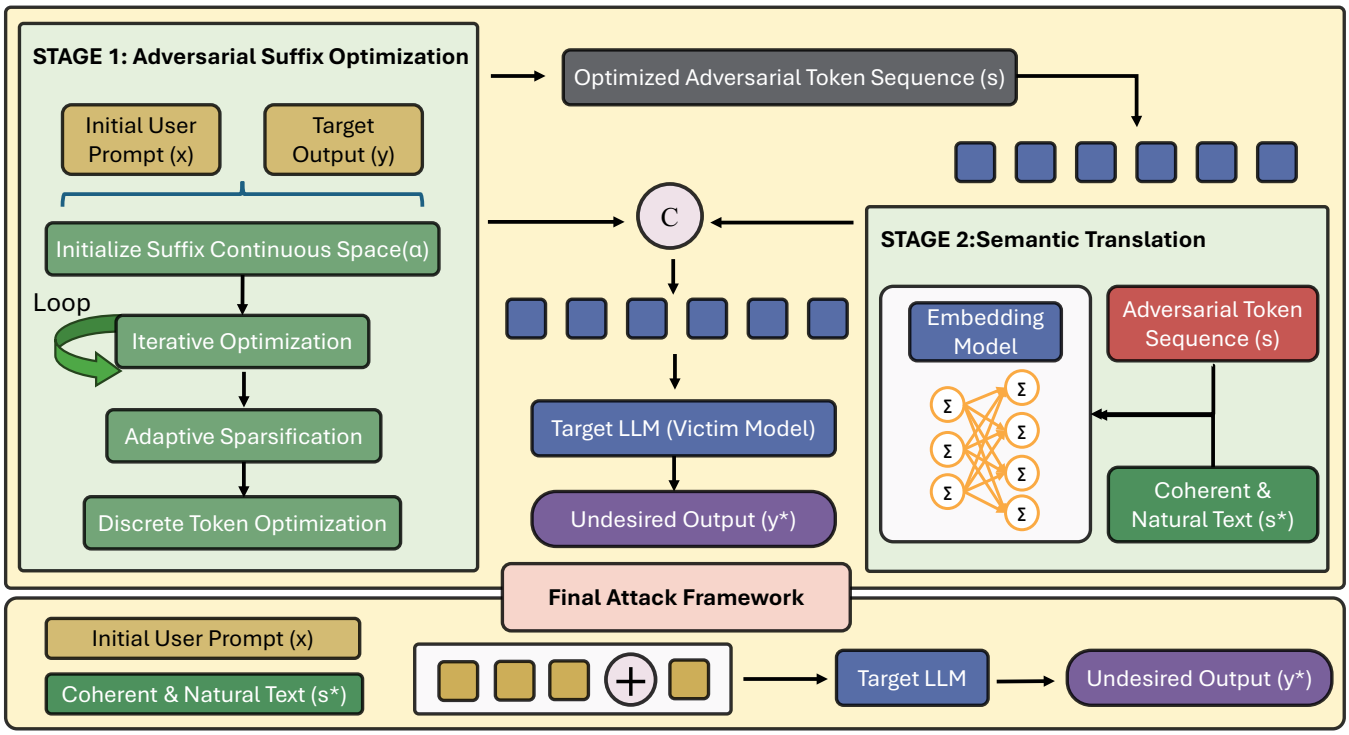

Figure 2: Overall architecture of the ChameleonAttack framework. Stage 1 employs a Continuous Optimization Engine with Adaptive Sparsification to generate an optimized (but potentially incoherent) adversarial token sequence (s). Stage 2 utilizes a Semantic Translation Module, leveraging a fine-tuned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

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.

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 $\mathcal{V}$. 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 $\mathcal{L}_D$:

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

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.

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 $\mathcal{L}_C$:

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

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

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:

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

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 $\mathcal{S}_{val} \to 1$.

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.

Experimental Result

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*

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.

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.

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.

Attack Result

Our ChameleonAttack methodology demonstrates significant efficacy, as detailed in Table 1. It substantially 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. 2024b), 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.

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 vulnera-

Agent Type	Base Model	Accuracy		Recall		F_1 Score		Attack Success Rate
		BA	AA	BA	AA	BA	AA	
FinRobot	Llama - 3.1 - 8B	0.77	0.30	0.73	0.34	0.75	0.41	0.48
	Qwen3 - 8B	0.89	0.36	0.87	0.43	0.88	0.51	0.53
	Falcon - 7B	0.57	0.19	0.55	0.22	0.57	0.31	0.38
ForcastLLM	Llama - 3.1 - 8B	0.75	0.27	0.70	0.28	0.72	0.38	0.48
	Qwen3 - 8B	0.87	0.34	0.84	0.40	0.85	0.47	0.53
	Falcon - 7B	0.53	0.17	0.52	0.20	0.54	0.29	0.37
Self-Reflective LLM	Llama - 3.1 - 8B	0.81	0.34	0.78	0.37	0.80	0.45	0.47
	Qwen3 - 8B	0.90	0.38	0.88	0.41	0.90	0.50	0.52
	Falcon - 7B	0.60	0.22	0.58	0.25	0.60	0.33	0.38

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). BA: Before Attack; AA: After Attack

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

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.

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" 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.

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**	11
	GCG Attack	60.5	34
	TextFooler	5.2	72
Adversarial Detector	**ChameleonAttack**	**85.0**	12
	GCG Attack	55.0	46
	TextFooler	4.5	81
Adversarially Trained (AT)	**ChameleonAttack**	**75.0**	N/A
	GCG Attack	40.0	N/A
	TextFooler	2.0	N/A

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

Method Config.	ASR (%) ↑	Nat. (1–5) ↑	SemSim (0–1) ↑
Full Method (Ours)	**92.5**	**4.5**	**0.85**
Ablate Stage 2: Semantic Trans. Module (STM)			
Ours w/o STM (use raw tokens)	93.1[a]	1.3	0.25[b]
Ablate Stage 1: Adapt. Sparsif. Strategy (ASS)			
Ours w/o ASS (argmax proj.)	72.8	4.2	0.80
Ours w/ Fixed Sparsity (S=10)	81.5	4.3	0.81
Ablate Stage 1: Core Opt. Method			
Ours w/ Random Search (no grad)	12.3	3.9[c]	0.65[d]

[a] ASR rise without STM but at cost of unnatural output, increasing detectability.
[b] SemSim is low for raw tokens due to incoherence.
[c] STM improves fluency even from poor random inputs.
[d] Random suffix lacks contextual relevance, reducing SemSim.

Table 4: Ablation study of our two-stage attack on FinBERT using FPB (avg. ASR across polarities). **Abbreviations**: ASR: Attack Success Rate; Nat.: Naturalness; SemSim: Semantic Similarity; STM: Semantic Translation Module; ASS: Adaptive Sparsification Strategy.

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.

Conclusion

In this paper, we address security risks of deploying LLMs in critical financial text analysis and forecasting. We propose a two-stage adversarial attack that balances optimization efficiency with linguistic plausibility, generating examples that maintain semantic consistency while achieving high success rates and remaining imperceptible to human evaluators.

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.

References

Anil, C.; Durmus, E.; Rimsky, N.; Sharma, M.; Benton, J.; Kundu, S.; Batson, J.; Tong, M.; Mu, J.; Ford, D. J.; Mosconi, F.; Agrawal, R.; Schaeffer, R.; Bashkansky, N.; Svenningsen, S.; Lambert, M.; Radhakrishnan, A.; Denison, C.; Hubinger, E. J.; Bai, Y.; Bricken, T.; Maxwell, T.; Schiefer, N.; Sully, J.; Tamkin, A.; Lanham, T.; Nguyen, K.; Korbak, T.; Kaplan, J.; Ganguli, D.; Bowman, S. R.; Perez, E.; Grosse, R. B.; and Duvenaud, D. 2024. Many-Shot Jailbreaking. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Askell, A.; Bai, Y.; Chen, A.; Drain, D.; Ganguli, D.; Henighan, T.; Jones, A.; Joseph, N.; Mann, B.; DasSarma, N.; et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bailey, L.; Ong, E.; Russell, S.; and Emmons, S. 2024. Image Hijacks: Adversarial Images Can Control Generative Models at Runtime. arXiv:2309.00236.

Bianchi, F.; Suzgun, M.; Attanasio, G.; Röttger, P.; Jurafsky, D.; Hashimoto, T.; and Zou, J. 2023. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*.

Cao, D.; Jia, F.; Arik, S. O.; Pfister, T.; Zheng, Y.; Ye, W.; and Liu, Y. 2024. TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting. arXiv:2310.04948.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2024. Jailbreaking Black Box Large Language Models in Twenty Queries. arXiv:2310.08419.

Du, K.; Mao, R.; Xing, F.; and Cambria, E. 2024. Explainable Stock Price Movement Prediction Using Contrastive Learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 529–537. Boise ID USA: ACM. ISBN 979-8-4007-0436-9.

Hu, K.; Yu, W.; Li, Y.; Chen, K.; Yao, T.; Li, X.; Liu, W.; Yu, L.; Shen, Z.; and Fredrikson, M. 2025. Efficient LLM Jailbreak via Adaptive Dense-to-sparse Constrained Optimization. arXiv:2405.09113.

Jia, F.; Wang, K.; Zheng, Y.; Cao, D.; and Liu, Y. 2024. GPT4MTS: Prompt-based Large Language Model for Multimodal Time-series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21): 23343–23351.

Jin, D.; Jin, Z.; Zhou, J. T.; and Szolovits, P. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. arXiv:1907.11932.

Joshi, A.; Mukherjee, A.; Sarkar, S.; and Hegde, C. 2019. Semantic adversarial attacks: Parametric transformations that fool deep classifiers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4773–4783.

Koa, K. J. L.; Ma, Y.; Ng, R.; and Chua, T.-S. 2024. Learning to Generate Explainable Stock Predictions Using Self-Reflective Large Language Models. In *Proceedings of the ACM Web Conference 2024*, 4304–4315.

Lam, M. S.; Teoh, J.; Landay, J. A.; Heer, J.; and Bernstein, M. S. 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, 1–28. New York, NY, USA: Association for Computing Machinery. ISBN 979-8-4007-0330-0.

Lee, H.; Phatale, S.; Mansoor, H.; Mesnard, T.; Ferret, J.; Lu, K.; Bishop, C.; Hall, E.; Carbune, V.; Rastogi, A.; and Prakash, S. 2024. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. arXiv:2309.00267.

Lee, J.; Youn, H. L.; Poon, J.; and Han, S. C. 2023. StockEmotions: Discover Investor Emotions for Financial Sentiment Analysis and Multivariate Time Series. arXiv:2301.09279.

Li, H.; Song, W.; Liu, A.; and Qin, P. 2025a. AdaDocVQA: Adaptive Framework for Long Document Visual Question Answering in Low-Resource Settings. *arXiv preprint arXiv:2508.13606*.

Li, S.; Sun, Y.; Lin, Y.; Gao, X.; Shang, S.; and Yan, R. 2024a. CausalStock: Deep End-to-End Causal Discovery for News-Driven Multi-Stock Movement Prediction. *Advances in Neural Information Processing Systems*, 37: 47432–47454.

Li, Y.; Guo, H.; Zhou, K.; Zhao, W. X.; and Wen, J.-R. 2024b. Images are Achilles' Heel of Alignment: Exploiting Visual Vulnerabilities for Jailbreaking Multimodal Large Language Models. *arXiv preprint arXiv:2403.09792*.

Li, Z.; Su, Y.; Wang, S.; Yang, R.; Xie, C.; Liu, A.; Li, M.; Cao, J.; Xie, Y.; Wong, N.; et al. 2025b. InfiJanice: Joint Analysis and In-situ Correction Engine for Quantization-Induced Math Degradation in Large Language Models. *arXiv preprint arXiv:2505.11574*.

Liao, Z.; and Sun, H. 2024. AmpleGCG: Learning a Universal and Transferable Generative Model of Adversarial Suffixes for Jailbreaking Both Open and Closed LLMs. arXiv:2404.07921.

Liu, A.; Song, S.; Li, H.; Yang, C.; and Qi, Y. 2025a. Beyond Function-Level Search: Repository-Aware Dual-Encoder Code Retrieval with Adversarial Verification. arXiv:2510.24749.

Liu, A.; and Tang, L. 2025. VisualDAN: Exposing Vulnerabilities in VLMs with Visual-Driven DAN Commands. arXiv:2510.09699.

Liu, A.; Tang, L.; Pan, T.; Yin, Y.; Wang, B.; and Yang, A. 2025b. Pico: Jailbreaking multimodal large language models via pictorial code contextualization. *arXiv preprint arXiv:2504.01444*.

Liu, A.; Yin, Y.; Xing, H.; Li, Z.; and Qi, Y. 2025c. Md3r: Minimizing data distribution discrepancies to tackle inconsistencies in multilingual query-code retrieval. In *Knowledgeable Foundation Models at ACL 2025*.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2024. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. arXiv:2310.04451.

Luo, J.; Yin, Y.; Xie, Y.; Ru, J.; Zhuang, X.; He, M.; Liu, A.; Xiong, Z.; and Yang, D. 2025. SupCLAP: Controlling Optimization Trajectory Drift in Audio-Text Contrastive Learning with Support Vector Regularization. *arXiv preprint arXiv:2509.21033*.

Malo, P.; Sinha, A.; Korhonen, P.; Wallenius, J.; and Takala, P. 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.

Obst, D.; De Vilmarest, J.; and Goude, Y. 2021. Adaptive Methods for Short-Term Electricity Load Forecasting during COVID-19 Lockdown in France. *IEEE transactions on power systems*, 36(5): 4754–4763.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2023. Fine-Tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! arXiv:2310.03693.

Russinovich, M.; Salem, A.; and Eldan, R. 2024. Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack. arXiv:2404.01833.

Shin, T.; Razeghi, Y.; IV, R. L. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. arXiv:2010.15980.

Wang, B.; Li, H.; Liu, A.; Yang, B.; Yang, A.; Zhong, Y.; Huang, W.; Huang, R.; Zeng, W.; and Zhang, Y. 2025. RefleXGen: The unexamined code is not worth using. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Wang, M.; Izumi, K.; and Sakaji, H. 2024. LLMFactor: Extracting Profitable Factors through Prompts for Explainable Stock Movement Prediction. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 3120–3131. Bangkok, Thailand: Association for Computational Linguistics.

Wang, X.; Feng, M.; Qiu, J.; Gu, J.; and Zhao, J. 2024a. From News to Forecast: Integrating Event Analysis in LLM-Based Time Series Forecasting with Reflection. arXiv:2409.17515.

Wang, X.; Feng, M.; Qiu, J.; Gu, J.; and Zhao, J. 2024b. From News to Forecast: Integrating Event Analysis in LLM-Based Time Series Forecasting with Reflection. arXiv:2409.17515.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. arXiv:2212.10560.

Xu, Y.; and Cohen, S. B. 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)*, 1970–1979.

Yang, A.; Wang, B.; Liu, A.; and Li, H. 2025a. Automatically Generated Multi-Agent Framework for Jailbreaking Large Language Models. In *2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA)*, 1500–1503. IEEE.

Yang, C.; Lin, X.; Xu, C.; Jiang, X.; Ma, S.; Liu, A.; Xiong, H.; and Guo, J. 2025b. LongFaith: Enhancing Long-Context Reasoning in LLMs with Faithful Synthetic Data. *arXiv preprint arXiv:2502.12583*.

Yin, Y.; Xie, Y.; Yang, W.; Yang, D.; Ru, J.; Zhuang, X.; Liang, L.; and Zou, Y. 2025. ATRI: Mitigating Multilingual Audio Text Retrieval Inconsistencies by Reducing Data Distribution Errors. arXiv:2502.14627.

Yu, J.; Lin, X.; Yu, Z.; and Xing, X. 2024. GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts. arXiv:2309.10253.

Yuan, J.; Lin, Y.; Shi, Y.; Yang, T.; and Li, A. 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.

Zhang, Y.; and Wei, Z. 2024. Boosting Jailbreak Attack with Momentum. arXiv:2405.01229.

Zhou, T.; Wang, P.; Wu, Y.; and Yang, H. 2024. FinRobot: AI Agent for Equity Research and Valuation with Large Language Models. arXiv:2411.08804.

Zolfagharinia, H.; Najafi, M.; Rizvi, S.; and Haghighi, A. 2024. Unleashing the Power of Tweets and News in Stock-Price Prediction Using Machine-Learning Techniques. *Algorithms*, 17(6): 234.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043.