AS-SYNERGIC: Orchestrating Synergistic Safety and Availability Attacks on Large Language Models

Yumeng Liu[1,2], Xin Wang[3,1,2*], Zhenyong Zhang[4], Ming Yang[1,2], and Xiaoming Wu[1,2]
[1]*Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China*
[2]*Shandong Provincial Key Laboratory of Industrial Network and Information System Security, Shandong Fundamental Research Center for Computer Science, Jinan 250014, China*
[3]*Quan Cheng Laboratory, Jinan 250014, China*
[4]*State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550000, China*

***Abstract*—Large language models (LLMs) face dual threats to safety and availability, yet existing attacks often fail to balance jailbreaking success, resource consumption, and stealthiness. To address this, we propose AS-SYNERGIC, a unified framework that orchestrates safety breaching and resource exhaustion attacks. Our approach introduces a curriculum scheduler that implements a *Breach-then-Flood* strategy, dynamically transitioning from bypassing safety guardrails to maximizing inference resource consumption. We further integrate a style-constrained mutation mechanism to keep optimization within the natural language manifold. Extensive experiments on 12 black-box target models demonstrate that AS-SYNERGIC achieves a 69.0% average attack success rate and induces responses 2.3× longer than normal inputs. Furthermore, it maintains superior stealthiness, with a perplexity of 42.1, significantly lower than unconstrained baselines.**

***Index Terms*—LLM Security, Jailbreak Attack, Availability Attack.**

I. Introduction

Large language models (LLMs) are increasingly deployed via the Model-as-a-Service (MaaS) paradigm [1], [2], exposing them to a dual spectrum of adversarial threats: semantic safety and operational availability. Jailbreak attacks craft malicious prompts to bypass safety alignments (e.g., RLHF [3]), inducing harmful content generation [4], [5]. Conversely, availability (Denial-of-Service) attacks manipulate inputs to exhaust GPU memory and maximize latency, inflicting financial damage on service providers [6], [7]. Despite the evolution of red-teaming evaluations [8], current research addresses these threats in isolation, creating a significant blind spot. As illustrated in Fig. 1, existing attacks exhibit a distinct trade-off between safety penetration and resource exhaustion.

*Corresponding author.
This work was supported in part by the Taishan Scholars Program under Grant tsqn202408239, in part by NSFC under Grants 62402256 and 62362008, in part by the Major Scientific and Technological Special Project of Guizhou Province ([2024]014), in part by the Research Project of Quancheng Laboratory, China under Grant QCL20250201 and Grant QCL20250302, and in part by the Pilot Project for the Integration of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2025ZDZX01.

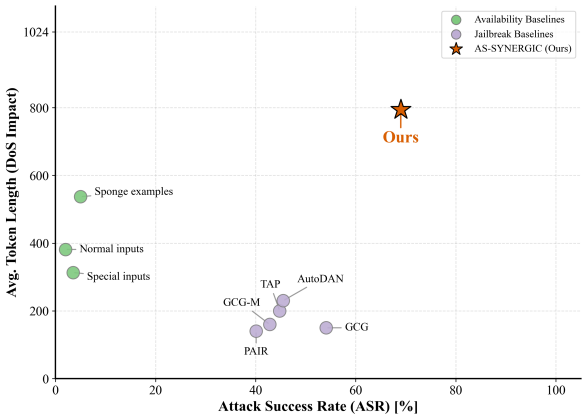

Fig. 1. **Safety-Availability Trade-off.** Jailbreak attacks (purple) yield short responses, while availability baselines (green) suffer from low ASR. **AS-SYNERGIC (Ours)** breaks this trade-off, jointly maximizing both ASR and response length.

Limitations of isolated attacks. Existing jailbreak methods [4], [9]–[11] focus on eliciting affirmative responses. However, aligned models often exhibit "shallow compliance," quickly terminating generation due to safety filters or brevity biases. Thus, these attacks yield short outputs that fail to impose significant computational loads [12]. Conversely, availability attacks that force long generation by minimizing the end-of-sequence (EOS) probability [6], [13] face *safety refusal* in aligned LLMs: without a preceding jailbreak, malicious prompts are immediately halted. Furthermore, pure EOS suppression often yields detectable high-perplexity "gibberish" [14], [15].

This dichotomy presents a formidable challenge: *How can an adversary simultaneously bypass safety guardrails, maximize resource consumption, and remain stealthy?* Integrating these is non-trivial, as features suppressing the EOS token often disrupt the semantics required for jailbreaking or trigger perplexity filters.

To bridge this gap, we propose **AS-SYNERGIC**, a unified framework orchestrating synergistic safety and availability attacks. We formulate adversarial suffix generation as a sequential decision-making problem via reinforcement learning (RL) [16]. To navigate the complex multi-objective landscape, we introduce a *Breach-then-Flood* strategy via a dynamic curriculum scheduler. In **Phase I (Breach)**, the optimizer prioritizes dismantling safety alignment. Once bypassed, the scheduler transitions to **Phase II (Flood)**, shifting focus to EOS suppression to coerce maximal generation. Crucially, a style-constrained mutation mechanism restricts the optimization trajectory within the natural language manifold, avoiding detectable gibberish.

Our contributions are summarized as follows:

- We propose **AS-SYNERGIC**, a unified framework that jointly evaluates LLM safety and availability, effectively breaking the trade-off limitations of existing baselines.
- We design a curriculum scheduler implementing a *Breach-then-Flood* strategy, resolving the optimization conflict between safety evasion and length promotion.
- Extensive experiments on 12 target models demonstrate that AS-SYNERGIC achieves a 69.0% average attack success rate and induces responses 2.3× longer than normal inputs (avg. ≈800 tokens), while maintaining superior stealthiness (perplexity 42.1, 10× lower than gradient baselines).

II. Related Work

Jailbreak and Availability Attacks. Research on bypassing LLM safety alignments (e.g., RLHF [3]) primarily employs optimization-based (e.g., GCG [4], AutoPrompt [9]) or generation-based (e.g., PAIR [10], TAP [11], DeepInception [17]) techniques. However, optimization methods often yield easily detectable high-perplexity gibberish [14], while generation methods typically elicit short responses, failing to address resource exhaustion. Conversely, pure availability attacks like sponge examples [6] and Engorgio [13] force extended outputs by suppressing the EOS token. Yet, they lack the semantic sophistication to penetrate modern safety guardrails, leading to immediate refusals. Simultaneously optimizing for both objectives creates severe gradient conflicts, which our dynamic curriculum scheduler explicitly resolves via a decoupled *Breach-then-Flood* strategy.

Adversarial Optimization via RL. To address the non-differentiable nature of discrete text optimization, recent approaches like MasterKey [18] utilize reinforcement learning (RL) and policy gradients for adversarial crafting. We extend this paradigm by integrating a style-constrained mutation mechanism. This strictly confines the search space to the natural language manifold, ensuring enhanced stealthiness while simultaneously pursuing the dual objectives of safety violation and resource exhaustion.

III. Methodology

Fig. 2 illustrates the proposed AS-SYNERGIC framework, which formulates the generation of adversarial suffixes as a sequential decision-making problem addressed through RL.

A. Problem Formulation

We formally define the adversarial attack on a conditional generative model $P_{f_\theta}(y \mid x)$ as a discrete optimization problem [4], [9], [19]. Given a malicious query x designed to elicit harmful content, the goal is to find an optimal suffix $\tilde{x}^*$ that maximizes a joint utility function $U(\cdot)$ over the model's stochastic response distribution:

$$\tilde{x}^* = \arg\max_{\tilde{x}\in\Phi} \mathbb{E}_{y\sim P_{f_\theta}(\cdot|x\oplus\tilde{x})} \left[U(y,\tilde{x})\right], \tag{1}$$

where Φ denotes the set of admissible suffixes. In contrast to unconstrained attacks [20], we restrict Φ to the natural language manifold to avoid detection by perplexity-based filters [21], [22]. The utility $U(\cdot)$ integrates three core objectives: jailbreaking score σ_{bypass} to bypass safety alignment, EOS suppression σ_{long} to prolong output generation, and Stealthiness σ_{stealth} to maintain low perceptual anomaly.

Directly solving (1) is intractable due to the discrete, combinatorial token space [19]. We therefore reframe the problem as an iterative, policy-guided search process. At each step t, a set of candidate suffixes $\mathcal{P}$ is generated via a mutation operator guided by a learned policy [23]. The current best suffix is then updated by selecting the candidate that maximizes the utility:

$$\tilde{x}^{(t)} \leftarrow \arg\max_{\tilde{x}\in\mathcal{P}} U(\tilde{x}). \tag{2}$$

This iterative formulation, grounded in RL principles, enables efficient navigation of the complex optimization landscape.

B. Multi-Objective Reward Design

To balance the competing objectives of safety compromise, computational resource exhaustion, and adversarial stealthiness, we define a time-variant joint utility function $U(\tilde{x})$. Unlike static weighting schemes that lead to suboptimal convergence in multi-task adversarial settings [24], our utility function is a dynamic scalarization of three core metrics. At each optimization step t, the global utility is computed as:

$$U(\tilde{x}) = w_1^{(t)} \cdot \sigma_{\text{bypass}} + w_2^{(t)} \cdot \sigma_{\text{long}} + w_3^{(t)} \cdot \sigma_{\text{stealth}}, \tag{3}$$

where $\mathbf{w}^{(t)} = [w_1^{(t)}, w_2^{(t)}, w_3^{(t)}]$ is an adaptive weight vector controlled by a curriculum scheduler $\Psi(t)$. This adaptive mechanism enables the optimizer to navigate the complex loss landscape by prioritizing safety breaching in early stages before shifting focus to resource exhaustion, following curriculum learning principles [25], [26]. The formal definitions of each component are detailed below.

1) Jailbreaking Score σ_{bypass}: This metric quantifies the target model's willingness to comply with harmful instructions. While black-box attacks often rely on coarse binary feedback from auxiliary judges, such sparse signals are inefficient for fine-grained token optimization [20]. Instead, we leverage dense probability signals from a local white-box reference model. By exploiting its rich gradient information,

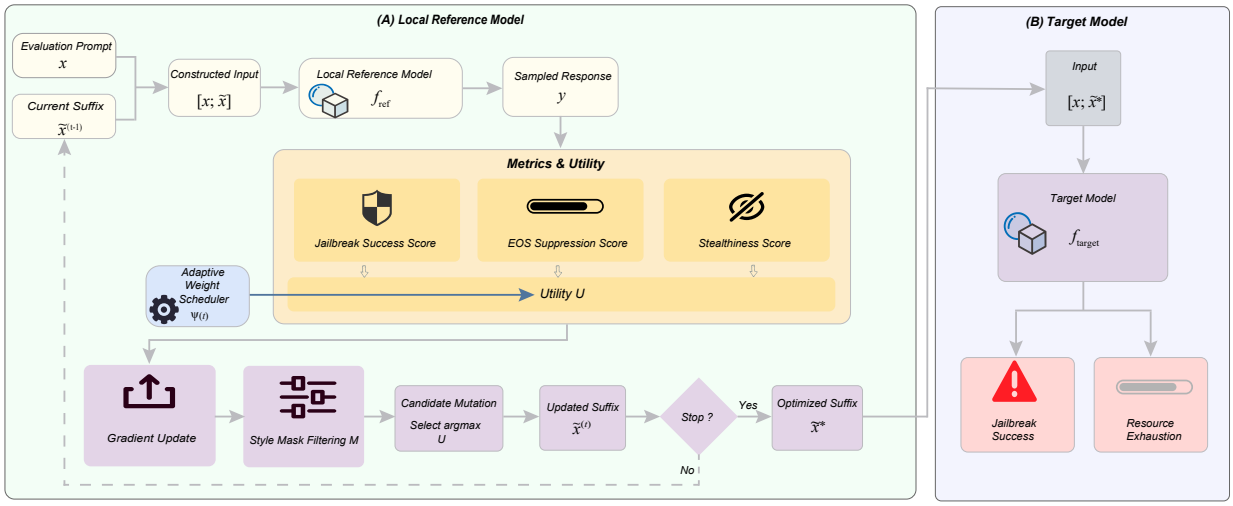

Fig. 2. **Overview of the AS-SYNERGIC framework.** The pipeline operates in two main stages: (A) *Local Reference Model*, where an adversarial suffix is iteratively optimized via policy gradient, guided by a multi-objective reward (safety, availability, and stealthiness); and (B) *Target Model*, where the optimized suffix is transferred to a black-box target to evaluate jailbreaking success and measure induced resource exhaustion.

our framework enables precise, token-level guidance for suffix mutation.

The score measures the generation confidence for a specific affirmative target prefix y_{target} (e.g., "Sure, here is") of length K. To mitigate the impact of token length on probability magnitude, we use the geometric mean of token-level probabilities:

$$\sigma_{\text{bypass}} = \exp\left(\frac{1}{K}\sum_{i=1}^{K}\log P_\theta(t_i \mid x, \tilde{x}, t_{<i})\right), \tag{4}$$

where t_i is the i-th token of the target prefix, and P_θ is the conditional distribution of the local reference model. Maximizing this continuous score provides smoother gradient signals than discrete loss functions, effectively guiding the adversarial suffix $\tilde{x}$ towards states that bypass safety alignment.

2) EOS Suppression σ_{long}: To induce significant computational latency (i.e., a DoS effect), the target model must be prevented from terminating generation prematurely. While standard jailbreaks often neglect output length [4], availability attacks highlight that suppressing the EOS probability is critical for maximizing resource consumption [6], [27]. We therefore define the availability score as the complement of the EOS probability at decoding step l (with L denoting the maximum generation length):

$$\sigma_{\text{long}} = 1 - P_\theta(y_l = \text{EOS} \mid x, \tilde{x}, y_{<l}). \tag{5}$$

Maximizing this metric penalizes early stopping, coercing the model to continue generation until the context window limit is reached, as demonstrated in recent sponge attacks [13].

3) Stealthiness σ_{stealth}: Conventional gradient-based attacks often converge to high-gradient, low-probability tokens, producing high-perplexity gibberish [4], [28].

Such inputs are easily detected by perplexity-based filters [14], [21]. To enforce alignment with the natural language manifold, we construct a composite metric that evaluates both linguistic fluency and semantic style consistency.

First, fluency is measured using perplexity (PPL), a standard metric for evaluating text generation quality [29]. For a suffix $\tilde{x} = [\tilde{x}_1, \ldots, \tilde{x}_L]$, the PPL under a reference model is:

$$\text{PPL}(\tilde{x}) = \exp\left(-\frac{1}{L}\sum_{i=1}^{L}\log P_{\text{ref}}(\tilde{x}_i \mid x, \tilde{x}_{<i})\right), \tag{6}$$

where P_{ref} is the probability distribution of the reference model. This unbounded metric is normalized to a score $s_{\text{ppl}} \in (0, 1]$ using an exponential decay function with positive parameter λ: $s_{\text{ppl}} = \exp(-\text{PPL}(\tilde{x})/\lambda)$.

Second, style consistency ensures the suffix blends semantically with a benign target domain (e.g., formal writing). We compute the cosine similarity between the suffix embedding $\mathbf{v}_{\tilde{x}}$ and the centroid $\mathbf{v}_{\mathcal{D}}$ of a target-style corpus:

$$\text{Sim}(\tilde{x}, \mathcal{D}) = \frac{\mathbf{v}_{\tilde{x}}^\top \mathbf{v}_{\mathcal{D}}}{\|\mathbf{v}_{\tilde{x}}\|\|\mathbf{v}_{\mathcal{D}}\|}. \tag{7}$$

This similarity is linearly mapped to $s_{\text{style}} \in [0, 1]$ via: $s_{\text{style}} = (\text{Sim}(\tilde{x}, \mathcal{D}) + 1)/2$.

The total stealthiness score is a weighted combination:

$$\sigma_{\text{stealth}} = \beta s_{\text{ppl}} + (1 - \beta) s_{\text{style}}. \tag{8}$$

This formulation ensures optimized suffixes remain statistically indistinguishable from benign text, bypassing defense mechanisms designed to flag anomalous inputs.

C. Optimization via Policy Gradient

A critical challenge in generating adversarial suffixes is the discrete nature of the token space, which prevents direct gradient backpropagation from the utility function $U(\tilde{x})$ to the input embeddings [19], [30]. While relaxation-based approaches employ continuous approximations [4], [20], they often converge

to local optima or yield semantically unnatural sequences. To address this non-differentiability, we frame suffix optimization as an RL problem, treating the local reference model f_{ref} as a stochastic policy $\pi_\theta(y \mid x \oplus \tilde{x})$.

Our objective is to maximize the expected joint utility over the model's response distribution. Formally, the objective with respect to the adversarial suffix is:

$$J(\tilde{x}) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x \oplus \tilde{x})}[U(y, \tilde{x})]. \tag{9}$$

Applying the policy gradient theorem [16], [31] and the log-derivative trick ($\nabla P = P \cdot \nabla \log P$), the gradient of the expectation is derived as:

$$\begin{aligned} \nabla_{\tilde{x}} J(\tilde{x}) &= \nabla_{\tilde{x}} \sum_y \pi_\theta(y \mid x \oplus \tilde{x}) \cdot U(y, \tilde{x}) \\ &= \mathbb{E}_{y \sim \pi_\theta} \left[U(y, \tilde{x}) \cdot \nabla_{\tilde{x}} \log \pi_\theta(y \mid x \oplus \tilde{x}) \right]. \end{aligned} \tag{10}$$

Since computing the exact expectation over the combinatorial output space is intractable, we approximate (10) using the REINFORCE estimator with Monte Carlo sampling. At each iteration, we sample N response trajectories. To reduce the high variance of Monte Carlo gradients, we introduce a baseline b:

$$\nabla_{\tilde{x}} J(\tilde{x}) \approx \frac{1}{N} \sum_{i=1}^{N} (U(y^{(i)}, \tilde{x}) - b) \cdot \nabla_{\tilde{x}} \log \pi_\theta(y^{(i)} \mid x \oplus \tilde{x}). \tag{11}$$

where b is computed as the exponential moving average of historical utility values. This approach provides temporal smoothing, ensuring gradients are driven by the relative advantage of the current samples over past performance, rather than absolute reward magnitudes.

In this framework, the term $\nabla_{\tilde{x}} \log \pi_\theta$ represents the gradient of the log-likelihood of generated tokens with respect to the suffix embeddings. This signal identifies which suffix tokens contribute most to high-utility outcomes. During candidate mutation, tokens with larger gradient magnitudes are prioritized, enabling efficient exploration of the discrete combinatorial space.

D. Style-Constrained Mutation

Standard gradient-based attacks often degrade into high-perplexity gibberish, rendering them easily detectable by perplexity-based filters [4], [21]. To mitigate this, we propose a style-constrained mutation mechanism that confines the optimization trajectory to a specific natural language manifold, ensuring that generated suffixes remain semantically coherent.

We define a binary style mask $\mathbf{M} \in \{0, 1\}^{|\mathcal{V}|}$ derived from a domain-specific corpus $\mathcal{D}_{\text{style}}$. This mask dynamically aligns the adversarial suffix with the semantic context of the target query. For instance, to exploit technical instruction-following capabilities, $\mathcal{D}_{\text{style}}$ may consist of Python or C++ syntax; conversely, for social engineering scenarios, it might mimic formal business correspondence.

Formally, we project the suffix gradient $\mathbf{g} = \nabla_{\tilde{\mathbf{e}}} \mathcal{J}$ onto the embedding matrix $\mathbf{E}$ to obtain candidate replacement scores $\mathbf{s} = \mathbf{E}\mathbf{g}$. To strictly enforce the style constraint, we apply the mask to these scores:

$$\mathbf{s}_{\text{masked}} = \mathbf{s} + (1 - \mathbf{M}) \cdot (-\infty). \tag{12}$$

This operation effectively assigns infinite negative weight to disallowed tokens, ensuring their probability becomes zero after Softmax normalization. Consequently, the candidate generation is strictly bounded within the permissible vocabulary defined by $\mathbf{M}$, balancing attack effectiveness with superior stealthiness.

E. Dynamic Curriculum Scheduling

Simultaneously optimizing for both safety violation (σ_{bypass}) and resource exhaustion (σ_{long}) presents a conflict: maximizing latency is ineffective if the model's safety guardrails remain active. To resolve this, we design a state-dependent curriculum scheduler $\Psi(t)$, inspired by curriculum learning [25], which dynamically adjusts the objective weight vector $\mathbf{w}^{(t)}$ based on real-time attack progression.

The optimization is divided into two phases. **Phase I (Breach)** focuses exclusively on inducing an affirmative target prefix (e.g., "Sure, here is") to bypass safety alignment, corresponding to the weight vector $\mathbf{w} = [1, 0, 0]$. **Phase II (Flood)** is activated only after the jailbreaking score σ_{bypass} exceeds a predefined confidence threshold τ. In this phase, the optimizer shifts its focus to prolonging generation and maintaining stealthiness. The adaptive update rule is:

$$\mathbf{w}^{(t)} = \begin{cases} [1, 0, 0], & \text{if } \sigma_{\text{bypass}}^{(t-1)} < \tau, \\ [\alpha, \eta, 1 - \alpha - \eta], & \text{if } \sigma_{\text{bypass}}^{(t-1)} \geq \tau. \end{cases} \tag{13}$$

Here, α and η are tunable coefficients controlling the balance between maintaining the jailbreak and promoting length (σ_{long}) and stealthiness (σ_{stealth}) in Phase II.

This mechanism functions as a dynamic hysteresis switch: if subsequent mutations during Phase II cause the jailbreaking score to regress below τ, the scheduler automatically reverts to Phase I. This ensures that a robust safety breach is a strict prerequisite for resource exhaustion, preventing the optimizer from converging to suboptimal states that fail either objective.

IV. Experiments

We evaluate AS-SYNERGIC across diverse LLMs to answer: **RQ1 (Safety):** Does it achieve a competitive attack success rate (ASR)? **RQ2 (Availability):** Can it maximize resource exhaustion via response length? **RQ3 (Ablation):** What is the contribution of each module?

A. Experimental Setup

Datasets & Models. We optimize suffixes on Gemma-7b-it using HarmBench [8] and transfer them to 12 black-box targets (e.g., Llama-2 [1], Vicuna [32], Qwen [33], Baichuan, Koala, Orca-2).

Baselines & Details. We compare against 10 methods including GCG [4], AutoPrompt [9], PAIR [10], TAP [11], AutoDAN [5], and Sponge examples [6]. Target generation

TABLE I
ASR (%) ON HARMBENCH IN A TRANSFER ATTACK SETTING. Suffixes optimized on Gemma-7b-it. Due to space limits, we report 4 representative targets; the Average is computed across all 12 evaluated models.

Target Model	GCG	GCG-T	GCG-M	AP	TAP	SFS	ZS	PAIR	UAT	AutoDAN	Ours
Llama-2-7b-chat	33.2	20.0	21.3	15.8	9.4	5.1	2.1	9.2	8.1	0.7	**34.1**
Vicuna-7b	66.0	61.3	61.3	60.2	52.1	43.1	27.4	53.4	24.1	64.9	**73.3**
Baichuan2-13b-chat	61.6	44.3	52.1	56.2	55.9	40.1	25.1	52.7	54.1	60.4	**80.1**
Qwen-7b-chat	60.1	38.1	53.6	50.1	54.2	32.3	15.7	50.1	16.1	48.2	**80.2**
Average (12 Models)	54.1	43.4	42.8	42.3	44.8	30.0	22.1	40.1	28.8	45.5	**69.0**

limits are $L_{\max} = 1024$. Suffixes are 20 tokens. Stealthiness (PPL) is assessed via LLaMA-2-7b-chat.

B. Main Results

Safety Evaluation. Table I details ASR performance. AS-SYNERGIC demonstrates superior efficacy: 1) It achieves a 69.0% average ASR across 12 models, significantly outperforming the leading baseline (GCG at 54.1%). 2) On heavily aligned targets like Llama-2-7b-chat where AutoDAN fails (0.7%), our method maintains 34.1% ASR. 3) Suffixes transfer effectively across architectures (e.g., ≈81% ASR on Qwen), proving the extraction of generalized adversarial features.

TABLE II
AVERAGE RESPONSE LENGTH (TOKENS). $L_{\max} = 1024$. The Average is computed across 6 representative target models.

Target Model	Normal	Special	Sponge	Ours
Orca-2-7b	350.5	310.2	610.5	**924.5**
Vicuna-7b	312.6	252.4	599.6	**882.0**
Qwen-7b-chat	329.1	166.5	350.0	**815.6**
Baichuan2-7b-chat	320.1	150.5	340.0	**790.1**
Average (6 Models)	341.7	289.4	463.8	**794.3**

Availability Evaluation. To assess resource exhaustion, we benchmark against normal inputs, special inputs (instructing long outputs), and sponge examples [6]. As shown in Table II, AS-SYNERGIC consistently elicits the longest sequences (avg. 794.3 tokens), a 2.3× increase over normal inputs (341.7). While aligned LLMs often truncate verbose instructions or sponge examples (e.g., dropping to 340.0 tokens on Baichuan2), our method successfully bypasses termination filters, peaking at 924.5 tokens on Orca-2-7b.

TABLE III
ABLATION OF AS-SYNERGIC COMPONENTS. Results averaged across LLaMA-2, Vicuna, and Koala.

$\mathcal{C}$	$\mathcal{E}$	$\mathcal{M}$	Avg. ASR (%)	Avg. Len	Avg. PPL
✗	✓	✓	28.4	315.5	45.3
✓	✗	✓	61.2	145.6	40.2
✓	✓	✗	63.5	912.4	624.8
✓	✗	✗	53.4	115.4	412.5
✓	✓	✓	**62.8**	**886.9**	**42.1**

Ablation Study. To dissect the contribution of each module, we analyze the results in Table III. Removing the dynamic scheduler ($\mathcal{C}$, Row 1) causes ASR to plummet to 28.4%, confirming that without the *Breach-then-Flood* strategy, the optimizer fails to bypass initial guardrails. Disabling the availability objective ($\mathcal{E}$, Row 2) yields truncated responses (145.6 tokens) despite a high ASR, failing to exhaust resources. Removing the style constraint ($\mathcal{M}$, Row 3) skyrockets PPL to 624.8, producing statistically anomalous gibberish. Fully ablating $\mathcal{E}$ and $\mathcal{M}$ (Row 4) degrades performance to a standard attack (115.4 tokens, 412.5 PPL). Conversely, the complete framework (Row 5) achieves optimal synergy: 62.8% ASR, 886.9 tokens, and 42.1 PPL.

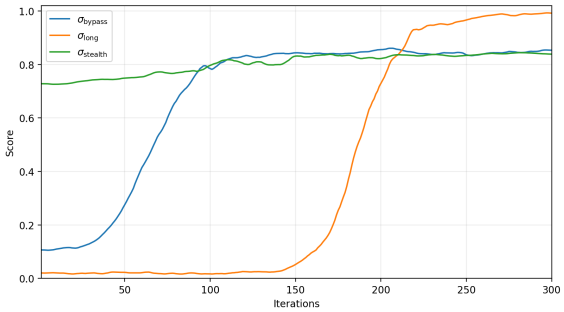

Fig. 3. **Optimization Dynamics.** σ_{bypass} (Phase I) converges before σ_{long} (Phase II) activates, while σ_{stealth} remains stable.

C. In-depth Analysis

Stealthiness Analysis. To quantify stealthiness, we benchmark suffix-only PPL against gradient-based baselines (L=20) using a local LLaMA-2-7b-chat. Traditional methods produce unnatural gibberish with extremely high PPLs (GCG: 416.0, GCG-T: 389.2, GCG-M: 405.1, UAT: 356.8, AutoPrompt: 289.4). In contrast, AS-SYNERGIC achieves a remarkably low PPL of 42.1—nearly 10× lower than GCG. This validates that our style mask ($\mathcal{M}$) effectively constrains optimization within the natural language manifold.

Optimization Dynamics. Fig. 3 tracks the three objectives over 300 iterations, visually validating the *Breach-then-Flood* mechanism. **Phase I (Breach):** Optimization initially maximizes only the jailbreak score σ_{bypass} (blue curve), crossing the threshold (τ≈0.8) near iteration 100. **Phase II (Flood):** After safety bypass stabilizes (iterations 100–140), the scheduler activates the availability objective. σ_{long} (orange curve)

climbs sharply to saturation (> 0.9). **Stealthiness:** Throughout both phases, σ_{stealth} (green curve) remains consistently stable (≈ 0.85) due to the active style mask.

Sensitivity Analysis. We examine the impact of the curriculum threshold τ and safety anchor α. Deviating τ from the optimal 0.8 prematurely triggers the flood phase (τ=0.5 yields 18.2% ASR) or excessively delays it (τ=0.9 drops length to 512.8). Similarly, a weak safety anchor (α=0.1) fails to sustain the jailbreak (25.4% ASR, 180.5 length), whereas overly rigid anchors ($\alpha \geq 0.6$) constrain the length. Thus, τ=0.8 and α=0.4 provide the optimal operating balance.

V. Conclusion

In this work, we present AS-SYNERGIC, a unified framework that reveals a critical dual vulnerability in LLMs: the potential for simultaneous safety breaches and availability attacks. By orchestrating a novel *Breach-then-Flood* strategy within a style-constrained RL process, our method reliably bypasses safety guardrails while inducing significant resource exhaustion. Comprehensive experiments on HarmBench demonstrate that AS-SYNERGIC achieves strong transferability across models, generates responses 2.3× longer than normal inputs, and maintains high stealthiness. These findings underscore the urgent need to jointly fortify both semantic safety and inference-time efficiency prior to LLM deployment. Future work will investigate vulnerability in multi-turn agentic settings and develop integrated defense mechanisms.

Ethical Considerations: This research is intended solely for controlled security evaluation and proactive model hardening. All experiments were conducted in restricted environments. We do not release directly executable adversarial payloads or automated attack tools. We hope these insights help develop practical mitigations that reduce real-world misuse risks and enhance the robustness of deployed LLM systems.

References

[1] H. Touvron, L. Martin, K. Stone *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[2] J. Achiam, S. Adler, S. Agarwal *et al.*, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] L. Ouyang, J. Wu, X. Jiang *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 27 730–27 744.

[4] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[5] X. Liu, N. Xu, M. Chen, and C. Xiao, "AutoDAN: Generating stealthy jailbreak prompts on aligned large language models," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[6] I. Shumailov, Y. Zhao, D. Bates *et al.*, "Sponge examples: Energy-latency attacks on neural networks," in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021, pp. 212–231.

[7] X. Feng, X. Han, S. Chen, and W. Yang, "LLMEffiChecker: Understanding and testing efficiency degradation of large language models," *ACM Transactions on Software Engineering and Methodology*, 2024.

[8] M. Mazeika, L. Phan, X. Yin *et al.*, "HarmBench: A standardized evaluation framework for automated red teaming and robust refusal," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[9] T. Shin, Y. Razeghi, R. L. L. IV *et al.*, "AutoPrompt: Eliciting knowledge from language models with automatically generated prompts," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 4222–4235.

[10] P. Chao, A. Robey, E. Dobriban *et al.*, "Jailbreaking Black-Box large language models in twenty queries," in *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025, pp. 23–42.

[11] A. Mehrotra, M. Zampetakis, P. Kassianik *et al.*, "Tree of attacks: Jailbreaking Black-Box LLMs automatically," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[12] Y. Yuan, W. Jiao, W. Wang *et al.*, "GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[13] J. Dong, Z. Zhang, W. X. Zhao *et al.*, "An engorgio prompt makes large language model babble on," in *International Conference on Learning Representations (ICLR)*, 2025.

[14] G. Alon and M. Kamfonas, "Detecting language model attacks with perplexity," *arXiv preprint arXiv:2308.14132*, 2023.

[15] N. Carlini, M. Nasr, C. A. Choquette-Choo *et al.*, "Are aligned neural networks adversarially aligned?" in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[16] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[17] X. Li, Z. Zhou, J. Zhu *et al.*, "DeepInception: Hypnotize large language model to be jailbreaker," in *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.

[18] G. Deng, Y. Liu, Y. Li *et al.*, "MasterKey: Automated jailbreaking of large language model chatbots," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2024.

[19] C. Guo, A. Sablayrolles, H. Jégou, and D. Kiela, "Gradient-based adversarial attacks against text transformers," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021, pp. 5747–5757.

[20] Y. Wen, N. Jain, J. Kirchenbauer *et al.*, "Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[21] N. Jain, A. Schwarzschild, Y. Wen *et al.*, "Baseline defenses for adversarial attacks against aligned language models," *arXiv preprint arXiv:2309.00614*, 2023.

[22] A. Robey, E. Wong, H. Hassani *et al.*, "SmoothLLM: Defending large language models against jailbreaking attacks," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[23] S. Geisler, T. Wollschläger, M. H. I. Abdalla *et al.*, "REINFORCE adversarial attacks on large language models: An adaptive, distributional, and semantic objective," in *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

[24] M. Cheng, J. Yi, P.-Y. Chen *et al.*, "Seq2Sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[25] Y. Bengio, J. Louradour, R. Collobert *et al.*, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, 2009, pp. 41–48.

[26] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2010, pp. 1189–1197.

[27] S. Chen, C. Liu, M. Kim, and W. Yang, "NMTSloth: Understanding and testing efficiency degradation of neural machine translation systems," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2022, pp. 1148–1160.

[28] Z. Wei, Y. Wang, A. Li *et al.*, "Jailbreak and guard aligned language models with only few in-context demonstrations," *arXiv preprint arXiv:2310.06387*, 2023.

[29] A. Radford, J. Wu, R. Child *et al.*, "Language models are unsupervised multitask learners," OpenAI Technical Paper, 2019.

[30] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *International Conference on Learning Representations (ICLR)*, 2017.

[31] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[32] W.-L. Chiang, Z. Li, Z. Lin *et al.*, "Vicuna: An open-source chatbot impressing GPT-4 with 90% ChatGPT quality," [Online]. Available: https://vicuna.lmsys.org, 2023.

[33] J. Bai, S. Bai, Y. Chu *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.