# Resource Consumption Threats in Large Language Models: A Survey

Anonymous ACL submission


## Abstract

Resource utilization has been studied as an optimization objective for generative large language models (LLMs). Recent studies show that this resource dependence can also be weaponized, as excessive generation, when scaled through high-volume requests, increases inference latency, undermines service availability, and threatens economic sustainability. However, these risks remain underexamined because existing studies are fragmented and their boundaries with legitimate, resource-intensive uses of LLMs are unclear. To address this gap, we provide a systematic review of threats to resource consumption in LLMs. Unlike surveys on inference acceleration, we focus on how resource consumption threats induce disruptive behaviors and synthesize existing studies into research strands that connect attack objectives, triggering mechanisms, and defenses. We further introduce an outcome-based severity taxonomy that distinguishes bounded, non-degenerate Overthinking from self-sustaining Unbounded Drift, providing operational criteria for characterizing adversarial resource consumption. Our goal is to clarify this emerging threat landscape and support future characterization, evaluation, and mitigation. The project repository is publicly available at https://anonymous.4open.science/r/RCT-LLM/.


## 1 Introduction

Generative large language models (LLMs) are deployed under stringent resource constraints, as inference depends on costly and limited computational infrastructure (Samsi et al., 2023; Miao et al., 2024). This constraint has made resource efficiency a central requirement for practical deployment, motivating extensive work on efficient inference and system optimization (Zhou et al., 2024; Fernandez et al., 2025). However, recent studies show that resource consumption is not merely an optimization concern but also a safety risk, as adversaries can induce excessive generation and amplify the burden

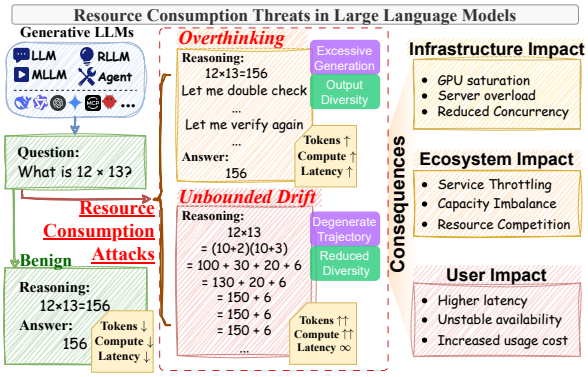


Figure 1: Overview of resource consumption threats in large language models.

by issuing repeated requests, thereby increasing inference latency, operational costs, and service disruption (Zhang et al., 2025e). Such threats have emerged across generative LLMs, including standard LLMs (Vaswani et al., 2017), reasoning LLMs (RLLMs) (Wei et al., 2022), multimodal LLMs (MLLMs) (Alayrac et al., 2022), and LLM-based agents (Agents) (Yao et al., 2022).

Resource consumption attacks are designed to induce disproportionate computational overhead in LLMs (Shumailov et al., 2021). Recent studies suggest that such attacks have emerged as an important safety risk (Geiping et al., 2024; Gao et al., 2024c). As shown in Figure 1, rather than directly inducing harmful outputs or extracting private information, they trigger excessive and unnecessary generation (Li et al., 2025a), which degrades model throughput, inflates operational costs, and places excessive pressure on LLM services (Gao et al., 2025). This risk reframes LLM resource usage from a performance issue to a core safety concern. Research on resource consumption threat has begun to appear across diverse LLM settings (Wang et al., 2023b), but existing studies remain scattered. This fragmentation obscures the threat landscape, while the unclear boundary between attacks and legitimate resource-intensive capabilities compli-

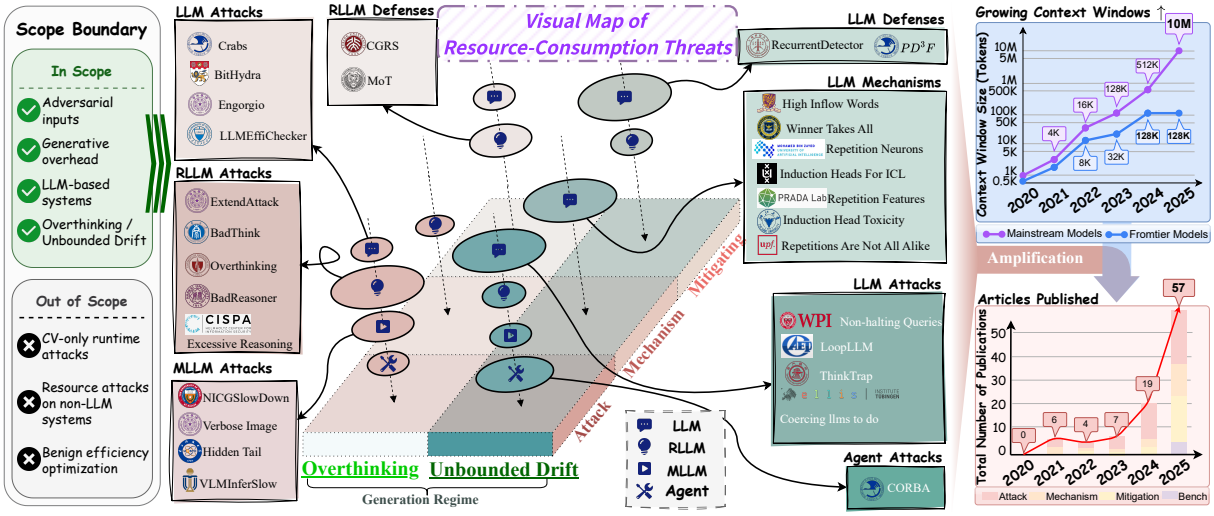


Figure 2: A unified view of resource consumption threats in large language models.

cates identification and assessment.

Our review centers on *resource consumption threats in Large Language Models*, especially malicious behaviors that amplify computational costs through extended or uncontrolled generation. To address this ambiguity, we introduce an outcome-based safety taxonomy for resource consumption. Rather than categorizing attacks by technique, the taxonomy characterizes their observable generation behaviors and distinguishes **Overthinking** from **Unbounded Drift**. In **Overthinking**, generation behavior like normal output, although it exceeds practical utility. In contrast, in **Unbounded Drift**, generation becomes self-sustaining, repetitive, or non-convergent, and therefore often persists until external limits, such as token budgets or timeouts, intervene. This distinction clarifies the boundaries of different attack modes and provides clearer links to attack behaviors for mechanism analysis and mitigation, as shown in Figure 2.

Based on this perspective, this survey makes three main contributions. First, we provide a comprehensive overview of recent advances on resource consumption threats. Second, we introduce a unified taxonomy that organizes resource consumption risks into two representative regimes, Overthinking and Unbounded Drift, clarifying the scope of this safety problem. Third, we analyze open challenges in this emerging area and outline promising directions for future research.

## 2 Threat Model and Scope

**Threat model.** We consider an external adversary who aims to impose excessive compute load on the provider-side serving infrastructure by inducing unnecessary LLM generation. The adversary typically issues repeated requests within a short time window to amplify resource consumption. The attack can be initiated from a broad range of entry points, such as user prompts, contextual inputs, or tool outputs in Agent. Its intended impact includes longer outputs, excessive reasoning steps, repeated tool invocations, increased latency, higher monetary cost, or degraded service availability.

**Research scope.** We study *active attack methods* that deliberately construct inputs or interaction contexts to induce excessive, repetitive, or non-convergent generation. The key criterion is that the additional computation is adversarially induced, rather than a natural cost of legitimate complex tasks. For *active attack methods*, we focus on targeted analyses and defenses against malicious resource amplification, and exclude general inference acceleration or serving optimization unless it is explicitly used for attack mitigation.

**Out-of-scope problems.** We exclude traditional computer vision attacks, as they are outside our focus on generative LLMs. We also exclude model availability attacks that induce refusal, abstention, or task failure, because they suppress valid responses rather than amplify computational cost through excessive generation.

## 3 Outcome-Based Taxonomy

Building on the threat model above, we do not treat the long generation itself as a resource consumption attack. Our scope is limited to cases where the additional computation is adversarially induced. Within this scope, attacks can be implemented through di-

Table 1: Operational distinction between Overthinking and Unbounded Drift. * indicates the most intuitive standard when making quantitative classifications.

| Aspect | Overthinking | Unbounded Drift |
|---|---|---|
| Overhead source | Excessive non-degenerate generation | Degenerate self-reinforcing trajectory |
| Output pattern | Redundant reasoning or tool use | Repetition with little progression |
| **Temporal signal*** | Extended across the output | Often concentrated in the tail segment |
| **Generation signal*** | Diversity and likelihood close to benign cases | Reduced diversity, lower PPL, or higher confidence |
| Attack design target | Length expansion or EOS suppression while maintaining non-degenerate generation | Stable repetitive loops that sustain predictable continuations |
| Defense focus | Utility-aware budget control | Loop detection and runtime interruption |

verse mechanisms, making a technique-based taxonomy difficult to apply consistently across LLM settings. We therefore use observable generation outcomes as the primary basis for our taxonomy. This taxonomy clarifies the severity of each threat and links distinct resource consumption patterns to appropriate defense strategies.

We classify resource consumption threats based on their observable output manifestations and use internal generation signals as operational indicators. In ***Overthinking***, the model produces unnecessary but non-degenerate reasoning, while its token diversity and likelihood profile often remain close to those of benign long generations. In ***Unbounded Drift***, the model enters iterative or repetitive generation patterns with little semantic progression. When token probabilities are available, this regime often exhibits low perplexity and high confidence over repeated continuations, reflecting a stable decoding pattern. Importantly, *Unbounded Drift* does not necessarily appear at the beginning of the output. In some cases, generation starts with an apparently normal prefix and then gradually falls into a repetitive loop. Once such a loop becomes self-sustaining and occupies a substantial portion of the attacked output, we categorize the case as *Unbounded Drift*, even if the initial prefix appears coherent (Fu et al., 2025; Gao et al., 2025). The operational indicators are summarized in Table 1, with further details provided in Appendix E.

This outcome-based taxonomy is orthogonal to model types and attack vectors. In the following sections, we further discuss resource consumption threats across standard LLMs, RLLMs, MLLMs, and Agents. We also refer to common attack interfaces, such as context and multimodal input. Since these dimensions are already well established in prior LLM safety studies, we adopt the existing classifications as secondary organizational dimensions rather than redefining them as the survey's primary taxonomy.

## 4 Resource Consumption Attacks

We organize attacks according to the two outcome regimes introduced in Section 3. Within each regime, we further discuss how attacks manifest across different LLM systems. Earlier DNN, NMT, and vision-only latency attacks are discussed in Appendix C as historical precursors, because they contextualize resource asymmetry but do not directly target modern generative LLM trajectories.

### 4.1 Output Elongation and Over-Generation

This strand captures the main form of *Overthinking* in text-only LLMs. The shared objective is to delay termination or weaken stopping behavior. Existing studies instantiate this pattern through different interfaces. LLMEffiChecker (Feng et al., 2024) characterizes efficiency degradation caused by length-increasing inputs, while Engorgio (Dong et al., 2024) more directly targets the termination boundary by encouraging models to continue generating verbose outputs. Crabs (Zhang et al., 2025e) shows that similar overhead can be triggered in black-box settings through semantic expansion, where tree-structured prompts make overlong responses appear natural under instruction following. BitHydra (Yan et al., 2025) represents a boundary case, as it manipulates model parameters rather than ordinary prompt or context interfaces. Overall, these attacks primarily manifest as *Overthinking* because they drive bounded yet unnecessary continuation, exceeding practical utility.

Black-box attacks are particularly concerning because they leverage ordinary user-facing interfaces, thereby posing broader risks to commercial models (Zhang et al., 2025e). By contrast, white-box methods often have weaker transferability, but their direct optimization over length-related signals or termination behavior can occasionally destabilize

the output trajectory. In rare cases, such destabilization can produce samples that fall into *Unbounded Drift*. However, because these outcomes occur sparsely and lack stable reproducibility, they are more appropriately interpreted as incidental byproducts of output elongation rather than direct instantiations of *Unbounded Drift*.

### 4.2 Repetition and Unbounded Loops

Unlike output elongation, Unbounded Drift targets stopping behavior by using explicit repetition, recursive instructions, or iterative semantic constructions. Their overhead arises from self-sustaining continuation rather than task demand, and often terminates only when an external limit, such as a max-token budget or timeout, is reached.

Representative attacks induce *Unbounded Drift* by disrupting normal convergence. Non-halting queries (Hammouri et al., 2025) formalize fixed-point-like inputs that prevent completion, while LoopLLM (Li et al., 2025a) and ThinkTrap (Li et al., 2025b) operationalize this risk through repetitive or effectively unbounded generation. RepetitionCurse (Huang et al., 2025d) further shows that repetitive patterns can also amplify resource usage through MoE router imbalance, making it a boundary case between generation-level drift and system-level pressure. By comparing existing research, we found that *Unbounded Drift* generally exhibits stronger attack effectiveness than *Overthinking* across different modalities (Table 2). However, this stronger effect usually comes with higher attack complexity, as targeted drift construction often relies on model-specific attention patterns or decoding dynamics, making it harder to construct and less transferable than *Overthinking*.

### 4.3 Reasoning Over-Amplification

RLLMs expose the reasoning trace as a distinct resource-bearing surface, enabling attacks that induce excessive computation rather than merely longer outputs, thereby primarily manifesting as *Overthinking*, where task-oriented generation may still produce correct answers but with unnecessarily long, redundant, or costly intermediate reasoning.

OverThink (Kumar et al., 2025) induces low-value reasoning by inserting redundant auxiliary tasks, while ExtendAttack (Zhu et al., 2025a) lengthens reasoning through structured inputs. Excessive Reasoning Attack (Si et al., 2025) further optimizes adversarial inputs to expand reasoning length. BadReasoner (Yi et al., 2025) and BadThink (Liu et al., 2025b) show that such *Overthinking* can also be implanted through triggers or backdoors. Together, these attacks preserve apparent task utility while inflating reasoning tokens and latency, turning reasoning efficiency from a benign optimization issue into an attack surface.

Table 2: Representative length signals in *Overthinking* and *Unbounded Drift* attacks, reported as average length with maximum length in braces.

| Setting | Representative attack | Length |
|---|---|---|
| Text LLM (8K) | Engorgio (Dong et al., 2024) | $1.4K_{\{4.0K\}}$ |
| | LoopLLM (Li et al., 2025a) | $2.1K_{\{7.4K\}}$ |
| Reasoning LLM (64K) | OverThink (Kumar et al., 2025) | $4.9K_{\{11.7K\}}$ |
| | RECUR (Wang et al., 2026b) | $5.3K_{\{64.4K\}}$ |
| Multimodal LLM (32K) | Verbose Images (Gao et al., 2024a) | $247.4_{\{321\}}$ |
| | LingoLoop (Fu et al., 2025) | $11.5K_{\{23.8K\}}$ |
| Agentic System (Multiple rounds) | Beyond Max Tokens (Zhou et al., 2026b) | $70.9K_{\{90.3K\}}$ |
| | Overthinking Loops (Lee et al., 2026) | $9.0K_{\{33.8K\}}$ |

RECUR (Wang et al., 2026b) represents a *Unbounded Drift* attack in reasoning models. It drives chain-of-thought reasoning into repeated reflective cycles, in which the model revisits similar intermediate states rather than progressing toward termination. Compared with bounded reasoning *Overthinking*, this drift-style behavior produces more pronounced resource consumption because the reasoning trajectory becomes recursively extended and difficult to terminate naturally.

### 4.4 Multimodal Resource Triggers

MLLMs introduce an additional trigger surface through visual inputs. Visual content can be crafted to alter the downstream generation trajectory. Before modern MLLMs, NICGSlowDown (Chen et al., 2022b) had already demonstrated that image inputs can induce generation slowdown in neural image captioning systems. This work serves as a precursor rather than a direct MLLM attack. Verbose Images (Gao et al., 2024a) and Verbose Samples (Gao et al., 2024b) extend this risk to modern vision-language models by crafting images or videos that induce overly long textual outputs and higher energy-latency costs. VLMInferSlow (Wang et al., 2025d) further evaluates this threat in black-box Vision LLMs. Hidden Tail (Zhang et al., 2025a) introduces a stealthier variant, where adversarial images trigger invisible or user-hidden tokens, separating visible response length from actual computation. Multimodal inputs can also trigger *Unbounded Drift*. LingoLoop (Fu et al., 2025) and RECITE (Gao

et al., 2025) show that image-conditioned generation may enter repetitive loops, where the model repeatedly describes. This line of work follows a similar resource-amplification logic to text-only LLM attacks, but shifts the attack entry point from textual prompts to multimodal inputs. The shift from textual prompts to high-dimensional inputs also makes such attacks harder to mitigate.

### 4.5 Agent and Tool Resource Amplification

Unlike standard LLM attacks, the final answer in Agent workflows may remain concise and unobtrusive, while substantial overhead accumulates in the intermediate execution trace. In this setting, *Overthinking* emerges as excessive but task-related tool use, whereas *Unbounded Drift* emerges as recursive execution.

LeechHijack (Zhang et al., 2025b) injects auxiliary workloads into an agent's reasoning and tool-use process, covertly consuming resources while preserving seemingly normal outputs. Beyond Max Tokens (Zhou et al., 2026b) and ClawDrain (Dong et al., 2026) further show that tool-calling chains can amplify token usage, latency, cost, and system load even when each individual step appears legitimate. A more severe form arises in *Unbounded Drift*. CORBA (Zhou et al., 2025) targets multi-agent systems by propagating self-replicating prompts across agents, inducing difficulty in generating effective replies. Overthinking Loops in Agents (Lee et al., 2026) similarly shows that malicious tool outputs can induce cyclic tool-use patterns and undermine usability. These studies suggest that agentic systems shift threats from output to execution control, requiring defenses that monitor not only what the agent finally says, but also how much work it performs along the way.

## 5 Mechanisms Underlying Resource Consumption Behaviors

After reviewing attack induction across LLMs, we now examine the generation dynamics that allow excessive trajectories to persist once triggered. We observe that mechanistic evidence is uneven across the two regimes: *Unbounded Drift* has a clearer behavioral boundary, making it easier to connect observable failures with mechanistic explanations, whereas *Overthinking* often resembles legitimate long-form generation and remains harder to causally isolate.

### 5.1 Mechanisms Behind Overthinking

Unlike *Unbounded Drift*, *Overthinking* is not yet associated with a clearly identified degenerative loop. Current studies, instead, tend to characterize Overthinking as excessive test-time computation that persists after useful reasoning has begun to saturate, while its causal mechanism remains only partially understood. Existing work provides suggestive evidence from several directions: behavioral analyses point to task-computation miscalibration, where models continue verification or alternative reasoning on simple or already-solved problems (Chen et al., 2025); studies of redundant reflection suggest that weak stopping signals may allow self-checking and self-affirmation to persist as locally coherent continuations (Liu et al., 2025a; Xie et al., 2025); and control-based work indicates that reasoning depth can be modulated through hidden-state or parameter-space directions (Lin et al., 2025; He and Tang, 2025; Ma et al., 2025). These findings suggest that *Overthinking* may involve imperfect compute allocation.

### 5.2 Mechanisms Behind Unbounded Drift

Mechanistic evidence for *Unbounded Drift* is richer than that for *Overthinking*. At the training level, standard likelihood objectives and repeated training patterns may reinforce frequent or copied tokens, while gradient corrections, HAPAX, and Repetition Dropout provide evidence that drift can reflect learned biases exposed by adversarial inputs or amplified through white-box optimization (Welleck et al., 2019; Lin et al., 2021; Sahin et al., 2025; Li et al., 2023). At the representation level, studies on repetition neurons, induction heads, and repetition features suggest that normal copying mechanisms may contribute to drift when they dominate logits or concentrate attention on low-information continuations (Hiraoka and Inui, 2025; Doan et al., 2025; Crosbie and Shutova, 2025; Yao et al., 2025; Wang et al., 2025b; Mahaut and Franzon, 2025; Yona et al., 2025b). At the probabilistic level, autoregressive decoding has been analyzed as entering self-reinforcing regions where repetitive continuations become more likely and harder to escape (Fu et al., 2021; Ildiz et al., 2024). At the behavioral level, studies of pseudo-repetition and uncertainty-driven fallback show that repetitive or uncertain states can increase the likelihood of low-entropy continuations, creating loop-like patterns that attackers may trigger or further optimize (Xu et al.,

Table 3: Operational comparison of active defenses against resource consumption attacks.

| Strategy | Refs. | Mechanism | Limitation |
|---|---|---|---|
| Resource Gating$^{O,D}$ | PD$^3$F (Zhang et al., 2025d) | Resource index → request polling and EOS amplification. | False positives on costly benign tasks. |
| Reflection Control$^{O}$ | CGRS (Huang et al., 2025b) | Confidence or reflection cues → suppress redundant reflection. | Narrow reflection patterns; trigger dependence. |
| Thought Monitoring$^{O}$ | MoT (Zhu et al., 2025b) | Reasoning trace → periodic monitoring and early stop. | Requires controllable reasoning traces. |
| Chunk Interruption$^{O,D}$ | WSC (Xie et al., 2025) | Hidden states at chunk boundaries → chop repetitive chunks. | Model-specific classifier; regeneration may loop. |
| Loop Detection$^{D}$ | RecurrentDetector (Yu et al., 2025) | Activation recurrence → online loop detection and halt. | White-box state access; limited closed-API applicability. |

*Notes.* $^{O}$ and $^{D}$ denote applicability to *Overthinking* and *Unbounded Drift*, respectively.

2022; Ivgi et al., 2024).

They are consistent with the view that, once termination signals are weakened, generation may continue not because the task requires additional computation, but because the trajectory has become increasingly self-sustaining. However, the relative contributions of these factors remain unclear, and a unified causal account of non-convergent generation remains an open problem.

## 6 Active Defenses against Resource Consumption Threats

Unlike general inference acceleration or efficiency optimization, this section focuses on mitigation strategies that address adversarially amplified resource attacks. Following the attack taxonomy in Section 4, we organize existing defenses around two outcome regimes: utility-aware control for *Overthinking* and loop-oriented interruption for *Unbounded Drift*. To make this discussion more analytical, Table 3 further compares these defenses based on their operational assumptions.

### 6.1 Defending against Overthinking

As discussed in Section 5, Overthinking is difficult to mitigate because the additional computation is often intertwined with legitimate reasoning and cannot be safely removed by simple length truncation. Existing methods, therefore, focus on utility-aware budget control and runtime intervention, aiming to reduce redundant reasoning while preserving task performance. PD$^3$F detects resource-intensive requests and combines dynamic scheduling with EOS-oriented suppression to limit attack-induced output expansion (Zhang et al., 2025d). For reasoning models, monitoring-based approaches terminate redundant reasoning paths during generation (Zhu et al., 2025b), while reflection-suppression methods interrupt unnecessary self-checking once continued reasoning provides limited additional utility (Huang et al., 2025b; Liu et al., 2025a). At the agent, recent work further extends this view to lifecycle resource governance, using fuzzing to detect denial-of-service (DoS) vulnerabilities caused by resource-abusing (Luo et al.). These studies provide early evidence that bounded resource amplification can be constrained during inference. Beyond these active defense efforts, however, much of the broader literature still treats *overthinking* primarily as benign efficiency control, leaving its defensive utility against adversarially induced *Overthinking* insufficiently established.

### 6.2 Defending against Unbounded Drift

For *Unbounded Drift*, defenses are more directly tied to the mechanisms reviewed in Section 5, because repetitive or non-convergent trajectories often expose internal model signals that can be used for loop detection, suppression, or forced termination. RecurrentDetector monitors activation recurrence during decoding and halts generation when internal states indicate repetitive looping (Yu et al., 2025). PD$^3$F complements this model-side detection with system-level containment, combining dynamic request polling with EOS amplification to reduce DoS pressure under malicious workloads (Zhang et al., 2025d). For reasoning models, WSC can interrupt unnecessary, repetitive reasoning chunks, although it was designed to reduce wasted decoding rather than to provide an adversarial non-halting defense (Xie et al., 2025). Overall, *Unbounded Drift* has clearer defensive targets than *Overthinking*, but current methods remain concentrated on text generation, with limited coverage of hidden multimodal loops, recursive tool calls, and agent-level execution chains.

Table 4: Recommended evaluation metrics for resource consumption attacks and defenses.

| Metric | Interpretation | Level | Direction | Usage |
|---|---|---|---|---|
| Output length | Generated or trajectory tokens. | Generation | A↑; M↓; D↓ | High |
| Time cost | Request-to-completion time. | Generation | A↑; M↓; D↓ | High |
| GPU utilization | Average or peak GPU occupancy. | Hardware | A↑; M– ; D↓ | Medium |
| GPU memory usage | Average or peak GPU memory use. | Hardware | A↑; M– ; D↓ | Medium |
| Throughput | Completed requests per unit time. | Service | A↓; M– ; D↑ | Medium |
| Task performance | Accuracy, success, or quality. | Service | A↑; M↑; D↑ | Medium |
| TTFT | Time to first generated token. | Service | A↑; M↓; D↓ | Low |
| Defense-time cost | Extra runtime from defense. | Generation | A– ; M– ; D↓ | Low |

*Notes.* A, M, and D denote attack, mechanism, and defense evaluation. Arrows indicate the favorable direction for evaluation.

## 7 Standardized Evaluation and Benchmarking

Across attack and defense studies, evaluations across modalities generally follow a common logic by measuring whether attack-induced expansion leads to measurable resource pressure. Text-only LLM attacks typically report latency and token usage (Dong et al., 2024; Zhang et al., 2025e), RLLM overthinking attacks further report reasoning amplification (Kumar et al., 2025; Si et al., 2025), while multimodal and agentic attacks additionally track visible length or hidden tokens (Wang et al., 2025d; Zhang et al., 2025c). Although these metrics differ in form, they map to three shared dimensions: generation behavior, physical resource pressure, and task utility. The problem is that most studies report only part of this set, making cross-paper and cross-modality comparison difficult. This also weakens defense evaluation, since shortening visible outputs does not necessarily reduce hardware pressure, monetary cost, or service-level degradation.

### 7.1 Evaluation Metrics

Table 4 summarizes representative metrics under these two views, together with their interpretation, preferred direction, and current usage in the literature. Attack capability captures whether the adversary successfully induces longer or slower generation, and should be measured using output length and end-to-end time cost. Physical resource pressure captures whether this abnormal generation translates into real serving burden, and should be measured using GPU utilization, GPU memory usage, and throughput. For most resource consumption attacks, output-side metrics remain the most direct indicators, because the attack effect is usually realized through prolonged generation. Input-side latency metrics, including time-to-first-token (TTFT), should be reported only as supplementary diagnostics when the attack explicitly targets serving-side bottlenecks, including MoE routing imbalance or prefill overhead.

Defense evaluation should report not only whether the attack is mitigated, but also the cost of applying the defense. A useful defense should reduce attack-induced length, latency, or resource pressure while preserving task utility. Therefore, defense studies should additionally report defense-time overhead, task performance after defense, GPU utilization, memory usage, and throughput under defended serving. These metrics help clarify whether a defense genuinely reduces infrastructure pressure, rather than merely reducing visible-output latency or shifting inference costs.

### 7.2 Benchmarking Gap and Outlook

A unified benchmark for resource consumption threats is still lacking. Recent benchmarks cover only partial settings, such as overflow or prompt-induced over-generation (Feiglin et al., 2026; Guo et al., 2025b), efficient reasoning (Huang et al., 2025c), and safety risks in long chain-of-thought reasoning (Jiang et al., 2025). However, these benchmarks do not directly evaluate adversarial resource amplification as a unified threat. Current studies often rely on reimplementation of attacks or on small-scale constructed attacks, making it difficult to compare threat severity across methods, models, and deployment settings.

A future benchmark should measure whether an input induces unnecessary resource amplification relative to a matched benign baseline. At minimum, it should cover three evaluation levels (Table 4). The generation level captures the most direct behavioral signals of resource abuse. At the hardware level, it measures whether these abnormal behaviors translate into actual computational pressure. The service level evaluates practical de-

ployment impact. These levels would provide a common basis for assessing both attack severity and defense effectiveness, without reducing heterogeneous resource-consumption threats to a single output-length metric.

# 8 Open Challenges and Future Directions

## 8.1 From Efficiency Optimization to Attack-Aware Resource Protection

A useful distinction is between efficiency-oriented mitigation and attack-aware defense. The former aims to reduce average cost under benign workloads, whereas the latter aims to protect resource budgets under adversarial amplification. Much of the literature still frames excessive generation as a cost-latency trade-off, focusing on shorter outputs, faster inference, or better throughput under normal use (Jiang et al., 2024; Agrawal et al., 2024; Del Corro et al., 2023). However, this perspective is insufficient for resource consumption threats, where the additional cost is intentionally imposed and may accumulate beyond the final output.

As LLMs are increasingly deployed on shared, resource-constrained infrastructure, resource consumption should be treated as a safety problem with the same urgency as efficiency optimization. Future defenses should therefore move beyond heuristic length control and treat resource usage as a constrained attack surface. This requires methods that can identify when resource growth is adversarially induced, trace how overhead accumulates across hidden reasoning, model calls, tool invocations, retrieval steps, memory writes, and multi-agent messages, and interrupt malicious amplification without suppressing legitimate complex workflows. This need is especially clear in agentic ecosystems, where the final response may remain normal while most overhead appears in the intermediate execution trajectory (Zhang et al., 2025b; Zhou et al., 2026a; Lee et al., 2026). Developing such attack-aware resource protection would help elevate resource control from average-case efficiency improvement to a safety objective for reliable and sustainable LLM services.

## 8.2 Understanding Threat Mechanisms

A major limitation of current research is the lack of a unified mechanistic understanding of resource amplification. Existing studies largely focus on specific attack instances, such as reasoning over-expansion or image-coded induction (Gao et al., 2025; Fu et al., 2025), while offering little general theory of how generation trajectories expand under attack manipulation. Future work should therefore integrate the shared dynamics and model-specific properties of different generative systems to develop transferable theories that support cross-model analysis and principled defense design.

## 8.3 Toward Consistent Evaluation Practices

Although Section 7 outlines a minimum evaluation protocol, standardized reporting has not yet become a common practice in this area. Future studies would benefit from adopting such evaluation standards as part of routine reporting, rather than treating them as supplementary experimental details. Only by reporting matched baselines and metrics that cover generation behavior, hardware pressure, and service impact can the field compare threat severity and distinguish genuine resource reduction from superficial output shortening.

## 8.4 Limited Understanding of Agentic Attacks

Compared with standalone LLMs, agents introduce additional sources of resource consumption, including expanded execution contexts, tool invocations, memory updates, and multi-agent communication. These components make resource consumption in agents more dependent on the full execution trajectory, rather than only on the final output. However, current research still provides limited characterization of these risks, and targeted defenses remain underdeveloped. Future research could further examine agent-level resource consumption and explore defenses that protect execution trajectories.

# 9 Conclusion

Resource consumption threats are emerging as an important safety concern for large language models. By inducing excessive and unnecessary generation, they not only reduce efficiency but also create broader risks for system reliability, service availability, and operational cost. This survey provided a unified view of the area through two representative regimes, *Overthinking* and *Unbounded Drift*, and reviewed existing work spanning attacks, mechanisms, defenses, and open challenges. Despite recent progress, the field remains fragmented in taxonomy, theory, evaluation, and mitigation. We hope this survey provides a clearer foundation for future research on understanding and mitigating threats to resource consumption and helps advance safer, more sustainable LLMs.

## Limitations

This survey has several limitations. First, our discussion centers on *resource consumption threats in LLMs*, rather than the broader landscape of efficiency, robustness, or availability problems. In particular, we primarily focus on malicious or adversarial resource abuse that induces excessive generation, and therefore do not aim to comprehensively cover benign efficiency optimization, general systems engineering for acceleration, or all forms of denial-of-service behavior outside the generative process itself. As a result, some closely related work on inference acceleration, scheduling, or hardware optimization is discussed only when it directly informs the security perspective of resource consumption.

Second, the literature in this area remains highly uneven across model families and attack settings. Existing studies are concentrated on text-based LLMs and a relatively small set of open-source models, while black-box commercial systems remain harder to study systematically. Coverage is also imbalanced across modalities and deployment scenarios: current multimodal work still focuses heavily on image-centered settings, whereas the risks in audio and video modalities remain far less explored; similarly, many downstream agent applications have not yet been systematically examined from the perspective of resource consumption.

Finally, because this is an emerging field, the available empirical evidence remains fragmented, and evaluation practices are not yet standardized. Many papers report attack success using different metrics, threat models, and deployment assumptions, making direct comparison difficult and potentially limiting the stability of any unified conclusions. For this reason, our synthesis should be understood as a structured overview of the current research landscape rather than a definitive benchmark of attack prevalence, mechanism universality, or defense effectiveness.

## References


Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav Gulavani, Alexey Tumanov, and Ramachandran Ramjee. 2024. Taming {Throughput-Latency} tradeoff in {LLM} inference with {Sarathi-Serve}. In *18th USENIX symposium on operating systems design and implementation (OSDI 24)*, pages 117–134.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Amit Baras, Alon Zolfi, Yuval Elovici, and Asaf Shabtai. 2025. Quantattack: Exploiting quantization techniques to attack vision transformers. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6730–6740. IEEE.

MD Abdul Barek, Md Bajlur Rashid, Md Mostafizur Rahman, ABM Kamrul Islamc Riad, Guillermo Francia, Hossain Shahriar, and Sheikh Iqbal Ahamed. 2025. Vulnerability to stability: Scalable large language model in queue-based web service. In *2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 995–1000. IEEE.

Sachin Bhat, Erik Cambria, and Haibo Peng. 2025. Nmret: A memory-augmented retrieval framework for large language models. Technical report, SenticNet / Nanyang Technological University.

Chen and 1 others. 2021. Translowdown: Efficiency attacks on neural machine translation systems.

Erh-Chung Chen, Pin-Yu Chen, I-Hsin Chung, and Che-Rung Lee. 2024. Overload: Latency attacks on object detection for edge devices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24716–24725.

Simin Chen, Hanlin Chen, Mirazul Haque, Cong Liu, and Wei Yang. 2023. The dark side of dynamic routing neural networks: Towards efficiency backdoor injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24585–24594.

Simin Chen, Cong Liu, Mirazul Haque, Zihe Song, and Wei Yang. 2022a. 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*, pages 1148–1160.

Simin Chen, Zihe Song, Mirazul Haque, Cong Liu, and Wei Yang. 2022b. Nicgslowdown: Evaluating the efficiency robustness of neural image caption generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15365–15374.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do NOT think that much for 33=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*.

Antonio Emanuele Cinà, Ambra Demontis, Battista Biggio, Fabio Roli, and Marcello Pelillo. 2025. Energy-latency attacks via sponge poisoning. *Information Sciences*, 702:121905.

Antonio Emanuele Cinà, Sebastiano Vascon, Ambra Demontis, Battista Biggio, Fabio Roli, and Marcello Pelillo. 2021. The hammer and the nut: Is bilevel optimization really needed to poison linear classifiers? In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Joy Crosbie and Ekaterina Shutova. 2025. Induction heads as an essential mechanism for pattern matching in in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5034–5096.

Luciano Del Corro, Allie Del Giorno, Sahaj Agarwal, Bin Yu, Ahmed Awadallah, and Subhabrata Mukherjee. 2023. Skipdecode: Autoregressive skip decoding with batching and caching for efficient llm inference. *arXiv preprint arXiv:2307.02628*.

Nhi Hoai Doan, Tatsuya Hiraoka, and Kentaro Inui. 2025. Understanding and controlling repetition neurons and induction heads in in-context learning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2854–2876.

Ben Dong, Hui Feng, and Qian Wang. 2026. Clawdrain: Exploiting tool-calling chains for stealthy token exhaustion in openclaw agents. *Preprint*, arXiv:2603.00902.

Jianshuo Dong, Ziyuan Zhang, Qingjie Zhang, Tianwei Zhang, Hao Wang, Hewu Li, Qi Li, Chao Zhang, Ke Xu, and Han Qiu. 2024. An engorgio prompt makes large language model babble on. *arXiv preprint arXiv:2412.19394*.

Erin Feiglin, Nir Hutnik, and Raz Lapid. 2026. Benchoverflow: Measuring overflow in large language models via plain-text prompts. *arXiv preprint arXiv:2601.08490*.

Xiaoning Feng, Xiaohong Han, Simin Chen, and Wei Yang. 2024. Llmefficchecker: Understanding and testing efficiency degradation of large language models. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–38.

Jared Fernandez, Clara Na, Vashisth Tiwari, Yonatan Bisk, Sasha Luccioni, and Emma Strubell. 2025. Energy considerations of large language model inference and efficiency optimizations. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32556–32569.

Jiyuan Fu, Kaixun Jiang, Lingyi Hong, Jinglun Li, Haijing Guo, Dingkang Yang, Zhaoyu Chen, and Wenqiang Zhang. 2025. Lingoloop attack: Trapping mllms via linguistic context and state entrapment into endless loops. *arXiv preprint arXiv:2506.14493*.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2021. A theoretical analysis of the repetition problem in text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12848–12856.

Haoran Gao, Yuanhe Zhang, Zhenhong Zhou, Lei Jiang, Fanyu Meng, Yujia Xiao, Li Sun, Kun Wang, Yang Liu, and Junlan Feng. 2025. Resource consumption red-teaming for large vision-language models. *arXiv preprint arXiv:2507.18053*.

Kuofeng Gao, Yang Bai, Jindong Gu, Shu-Tao Xia, Philip Torr, Zhifeng Li, and Wei Liu. 2024a. Inducing high energy-latency of large vision-language models with verbose images. *arXiv preprint arXiv:2401.11170*.

Kuofeng Gao, Jindong Gu, Yang Bai, Shu-Tao Xia, Philip Torr, Wei Liu, and Zhifeng Li. 2024b. Energy-latency manipulation of multi-modal large language models via verbose samples. *arXiv preprint arXiv:2404.16557*.

Kuofeng Gao, Tianyu Pang, Chao Du, Yong Yang, Shu-Tao Xia, and Min Lin. 2024c. Denial-of-service poisoning attacks against large language models. *arXiv preprint arXiv:2410.10760*.

Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen, and Tom Goldstein. 2024. Coercing llms to do and reveal (almost) anything, 2024. *URL https://arxiv. org/abs/2402.14020*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yi Guo, Kanchana Thilakarathna, Nirhoshan Sivaroopan, Jo Plested, Tim Lynar, Jack Yang, Wangli Yang, and 1 others. 2025b. Prompt-induced over-generation as denial-of-service: A black-box attack-side benchmark. *arXiv preprint arXiv:2512.23779*.

Ghaith Hammouri, Kemal Derya, and Berk Sunar. 2025. Non-halting queries: Exploiting fixed points in llms. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 1–22. IEEE.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855.

Syed Mhamudul Hasan, Hussein Zangoti, Iraklis Anagnostopoulos, and Abdur R Shahid. 2025. Sponge attacks on sensing ai: Energy-latency vulnerabilities and defense via model pruning. *arXiv preprint arXiv:2505.06454*.

Yixin He and Lumingyuan Tang. 2025. Learning to ponder: Adaptive reasoning in latent space. *arXiv preprint arXiv:2509.24238*.

Tatsuya Hiraoka and Kentaro Inui. 2025. Repetition neurons: How do language models produce repetitions? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 483–495.

Sanghyun Hong, Yiğitcan Kaya, Ionuţ-Vlad Modoranu, and Tudor Dumitraş. 2020. A panda? no, it's a sloth: Slowdown attacks on adaptive multi-exit neural network inference. *arXiv preprint arXiv:2010.02432*.

Donghao Huang, Thanh-Son Nguyen, Fiona Liausvia, and Zhaoxia Wang. 2025a. Rap: A metric for balancing repetition and performance in open-source large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1479–1496.

Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. 2025b. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. *arXiv preprint arXiv:2508.05337*.

Junquan Huang, Haotian Wu, Yubo Gao, Yibo Yan, Junyan Zhang, Yonghua Hei, Song Dai, Jie Zhang, Puay Siew Tan, and Xuming Hu. 2025c. Effireasonbench: A unified benchmark for evaluating and advancing efficient reasoning in large language models. *arXiv preprint arXiv:2511.10201*.

Ruixuan Huang, Qingyue Wang, Hantao Huang, Yudong Gao, Dong Chen, Shuai Wang, and Wei Wang. 2025d. Repetitioncurse: Measuring and understanding router imbalance in mixture-of-experts llms under dos stress. *Preprint*, arXiv:2512.23995.

Muhammed Emrullah Ildiz, Yixiao Huang, Yingcong Li, Ankit Singh Rawat, and Samet Oymak. 2024. From self-attention to markov models: Unveiling the dynamics of generative transformers. In *International Conference on Machine Learning*, pages 20955–20982. PMLR.

Maor Ivgi, Ori Yoran, Jonathan Berant, and Mor Geva. 2024. From loops to oops: Fallback behaviors of language models under uncertainty. *arXiv preprint arXiv:2407.06071*.

Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. 2025. Safechain: Safety of language models with long chain-of-thought reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23303–23320.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. 2025. Overthink: Slowdown attacks on reasoning llms. *Preprint*, arXiv:2502.02542.

Yohan Lee, Jisoo Jang, Seoyeon Choi, Sangyeop Kim, and Seungtaek Choi. 2026. Overthinking loops in agents: A structural risk via mcp tools. *arXiv preprint arXiv:2602.14798*.

Huayang Li, Tian Lan, Zihao Fu, Deng Cai, Lemao Liu, Nigel Collier, Taro Watanabe, and Yixuan Su. 2023. Repetition in repetition out: Towards understanding neural text degeneration from the data perspective. *Advances in Neural Information Processing Systems*, 36:72888–72903.

Junxian Li, Tu Lan, Haozhen Tan, Yan Meng, and Haojin Zhu. 2026. Slowba: An efficiency backdoor attack towards vlm-based gui agents. *arXiv preprint arXiv:2603.08316*.

Xingyu Li, Xiaolei Liu, Cheng Liu, Yixiao Xu, Kangyi Ding, Bangzhou Xin, and Jia-Li Yin. 2025a. Loopllm: Transferable energy-latency attacks in llms via repetitive generation. *arXiv preprint arXiv:2511.07876*.

Yunzhe Li, Jianan Wang, Hongzi Zhu, James Lin, Shan Chang, and Minyi Guo. 2025b. Thinktrap: Denial-of-service attacks against black-box llm services via infinite thinking. *arXiv preprint arXiv:2512.07086*.

Xiang Lin, Simeng Han, and Shafiq Joty. 2021. Straight to the gradient: Learning to use novel tokens for neural text generation. In *International conference on machine learning*, pages 6642–6653. PMLR.

Zhengkai Lin, Zhihang Fu, Ze Chen, Chao Chen, Liang Xie, Wenxiao Wang, Deng Cai, Zheng Wang, and Jieping Ye. 2025. Controlling thinking speed in reasoning models. *arXiv preprint arXiv:2507.03704*.

Jona te Lintelo, Stefanos Koffas, and Stjepan Picek. 2024. The skipsponge attack: Sponge weight poisoning of deep neural networks. *arXiv preprint arXiv:2402.06357*.

Han Liu, Yuhao Wu, Zhiyuan Yu, Yevgeniy Vorobeychik, and Ning Zhang. 2023. Slowlidar: Increasing the latency of lidar-based detection using adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5146–5155.

Kaiyuan Liu, Chen Shen, Zhanwei Zhang, Junjie Liu, Xiaosong Yuan, and 1 others. 2025a. Efficient reasoning through suppression of self-affirmation reflections in large reasoning models. *arXiv preprint arXiv:2506.12353*.

Shuaitong Liu, Renjue Li, Lijia Yu, Lijun Zhang, Zhiming Liu, and Gaojie Jin. 2025b. Badthink: Triggered overthinking attacks on chain-of-thought reasoning in large language models. *Preprint*, arXiv:2511.10714.

Jiahao Lu, Yifan Zhang, Qiuhong Shen, Xinchao Wang, and Shuicheng Yan. 2024. Poison-splat: Computation cost attack on 3d gaussian splatting. *arXiv preprint arXiv:2410.08190*.

Jinghui Lu, Haiyang Yu, Siliang Xu, Shiwei Ran, Guozhi Tang, Siqi Wang, Bin Shan, Teng Fu, Hao Feng, Jingqun Tang, and 1 others. 2025. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient llm/mllm reasoning. *arXiv preprint arXiv:2505.15154*.

Jiaqi Luo, Jiarun Dai, Fengyu Liu, Songyang Peng, Youkun Shi, Tong Bu, Geng Hong, Xudong Pan, and Yuan Zhang. Autonomy comes with costs: Detecting denial-of-service vulnerabilities caused by resource abusing in llm-based agents.

Chen Ma, Ningfei Wang, Qi Alfred Chen, and Chao Shen. 2024. Slowtrack: Increasing the latency of camera-based perception in autonomous driving using adversarial examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(5):4062–4070.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. Cot-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6035.

Matéo Mahaut and Francesca Franzon. 2025. Repetitions are not all alike: distinct mechanisms sustain repetition in language models. *arXiv preprint arXiv:2504.01100*.

Eran Malach. 2024. Auto-regressive next-token predictors are universal learners. In *Proceedings of the 41st International Conference on Machine Learning*, pages 34417–34431.

Hanene FZ Brachemi Meftah, Wassim Hamidouche, Sid Ahmed Fezza, Olivier Déforges, and Kassem Kallas. 2025. Energy backdoor attack to deep neural networks. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xupeng Miao, Chunan Shi, Jiangfei Duan, Xiaoli Xi, Dahua Lin, Bin Cui, and Zhihao Jia. 2024. Spotserve: Serving generative large language models on preemptible instances. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 1112–1127.

Andreas Müller and Erwin Quiring. 2024. The impact of uniform inputs on activation sparsity and energy-latency attacks in computer vision. In *2024 IEEE Security and Privacy Workshops (SPW)*, pages 104–111. IEEE.

Sania Nayab, Giulio Rossolini, Marco Simoni, Andrea Saracino, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2024. Concise thoughts: Impact of output length on llm reasoning and cost. *arXiv preprint arXiv:2407.19825*.

Andreas Ottem. 2025. Meve: A modular system for memory verification and effective context control in language models. In *CS & IT Conference Proceedings*, volume 15. CS & IT Conference Proceedings.

Kerem Sahin, Sheridan Feucht, Adam Belfki, Jannik Brinkmann, Aaron Mueller, David Bau, and Chris Wendler. 2025. In-context learning without copying. *arXiv preprint arXiv:2511.05743*.

Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. 2023. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE high performance extreme computing conference (HPEC)*, pages 1–9. IEEE.

Coen Schoof, Stefanos Koffas, Mauro Conti, and Stjepan Picek. 2024. Beyond phantomsponges: Enhancing sponge attack on object detection models. In *Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning*, pages 14–19.

Minjae Seo, Myoungsung You, Junhee Lee, Jaehan Kim, Hwanjo Heo, Jintae Oh, and Jinwoo Kim. 2025. Eo-vlm: Vlm-guided energy overload attacks on vision models. *arXiv preprint arXiv:2504.08205*.

Avishag Shapira, Alon Zolfi, Luca Demetrio, Battista Biggio, and Asaf Shabtai. 2023. Phantom sponges: Exploiting non-maximum suppression to attack deep object detectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4571–4580.

Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. 2021. Sponge examples: Energy-latency attacks on neural networks. In *2021 IEEE European symposium on security and privacy (EuroS&P)*, pages 212–231. IEEE.

Wai Man Si, Mingjie Li, Michael Backes, and Yang Zhang. 2025. Excessive reasoning attack on reasoning llms. *Preprint*, arXiv:2506.14374.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chenyi Wang, Ruoyu Song, Raymond Muller, Jean-Philippe Monteuuis, Z. Berkay Celik, Jonathan Petit, Ryan Gerdes, and Ming Li. 2025a. Cp-freezer: Latency attacks against vehicular cooperative perception. *Preprint*, arXiv:2508.01062.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Shuxun Wang, Qingyu Yin, Chak Tou Leong, Qiang Zhang, and Linyi Yang. 2025b. Induction head toxicity mechanistically explains repetition curse in large language models. *arXiv preprint arXiv:2505.13514*.

Tianyi Wang, Huawei Fan, Yuanchao Shu, Peng Cheng, and Cong Wang. 2026a. Rethinking latency denial-of-service: Attacking the llm serving framework, not the model. *arXiv preprint arXiv:2602.07878*.

Tianyi Wang, Zichen Wang, Cong Wang, Yuanchao Shu, Ruilong Deng, Peng Cheng, and Jiming Chen. 2025c. Can't slow me down: Learning robust and hardware-adaptive object detectors against latency attacks for edge devices. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19230–19240.

Xiasi Wang, Tianliang Yao, Simin Chen, Runqi Wang, Lei Ye, Kuofeng Gao, Yi Huang, and Yuan Yao. 2025d. Vlminferslow: Evaluating the efficiency robustness of large vision-language models as a service. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16035–16050.

Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025e. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6904–6917.

Zijian Wang, Shuo Huang, Yujin Huang, and Helei Cui. 2023b. Energy-latency attacks to on-device neural networks via sponge poisoning. In *Proceedings of the 2023 Secure and Trustworthy Deep Learning Systems Workshop*, pages 1–11.

Ziwei Wang, Yuanhe Zhang, Jing Chen, Zhenhong Zhou, Ruichao Liang, Ruiying Du, Ju Jia, Cong Wu, and Yang Liu. 2026b. Recur: Resource exhaustion attack via recursive-entropy guided counterfactual utilization and reflection. *arXiv preprint arXiv:2602.08214*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Xingyu Wu, Yuchen Yan, Shangke Lyu, Linjuan Wu, Yiwen Qiu, Yongliang Shen, Weiming Lu, Jian Shao, Jun Xiao, and Yueting Zhuang. 2025. Lapo: Internalizing reasoning efficiency via length-adaptive policy optimization. *arXiv preprint arXiv:2507.15758*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and 1 others. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

Wenya Xie, Shaochen Zhong, Hoang Anh Duy Le, Zhaozhuo Xu, Jianwen Xie, and Zirui Liu. 2025. Word salad chopper: Reasoning models waste a ton of decoding budget on useless repetitions, self-knowingly. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33576–33586.

Huiyu Xu, Zhibo Wang, Wenhui Zhang, Ziqi Zhu, Yaopeng Wang, Kui Ren, and Chun Chen. 2026. Looptrap: Termination poisoning attacks on llm agents. *arXiv preprint arXiv:2605.05846*.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *Advances in Neural Information Processing Systems*, 35:3082–3095.

Xiaobei Yan, Yiming Li, Hao Wang, Han Qiu, and Tianwei Zhang. 2025. Bithydra: Towards bit-flip inference cost attack against large language models. *arXiv preprint arXiv:2505.16670*.

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. 2025. Understanding the repeat curse in large language models from a feature perspective. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7787–7815, Vienna, Austria. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Biao Yi, Zekun Fei, Jianing Geng, Tong Li, Lihai Nie, Zheli Liu, and Yiming Li. 2025. Badreasoner: Planting tunable overthinking backdoors into large reasoning models for fun or profit. *Preprint*, arXiv:2507.18305.

Itay Yona, Ilia Shumailov, Jamie Hayes, Federico Barbero, and Yossi Gandelsman. 2025a. Interpreting the repeated token phenomenon in large language models. *arXiv preprint arXiv:2503.08908*.

Itay Yona, Ilia Shumailov, Jamie Hayes, and Yossi Gandelsman. 2025b. Interpreting the repeated token phenomenon in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 72535–72555. PMLR.

Junzhe Yu, Yi Liu, Huijia Sun, Ling Shi, and Yuqi Chen. 2025. Breaking the loop: Detecting and mitigating denial-of-service vulnerabilities in large language models. *arXiv preprint arXiv:2503.00416*.

Rui Zhang, Zihan Wang, Tianli Yang, Hongwei Li, Wenbo Jiang, Qingchuan Zhao, Yang Liu, and Guowen Xu. 2025a. Hidden tail: Adversarial image causing stealthy resource consumption in vision-language models. *arXiv preprint arXiv:2508.18805*.

Yuanhe Zhang, Weiliu Wang, Zhenhong Zhou, Kun Wang, Jie Zhang, Li Sun, Yang Liu, and Sen Su. 2025b. Leechhijack: Covert computational resource exploitation in intelligent agent systems. *Preprint*, arXiv:2512.02321.

Yuanhe Zhang, Weiliu Wang, Zhenhong Zhou, Kun Wang, Jie Zhang, Li Sun, Yang Liu, and Sen Su. 2025c. Leechhijack: Covert computational resource exploitation in intelligent agent systems. *arXiv preprint arXiv:2512.02321*.

Yuanhe Zhang, Xinyue Wang, Haoran Gao, Zhenhong Zhou, Fanyu Meng, Yuyao Zhang, and Sen Su. 2025d. pd3f: A pluggable and dynamic dos-defense framework against resource consumption attacks targeting large language models. *arXiv preprint arXiv:2505.18680*.

Yuanhe Zhang, Zhenhong Zhou, Wei Zhang, Xinyue Wang, Xiaojun Jia, Yang Liu, and Sen Su. 2025e. Crabs: Consuming resource via auto-generation for llm-dos attack under black-box settings. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11128–11150.

Kaiyu Zhou, Yongsen Zheng, Yicheng He, Meng Xue, Xueluan Gong, Yuji Wang, and Kwok-Yan Lam. 2026a. Beyond max tokens: Stealthy resource amplification via tool calling chains in llm agents. *arXiv preprint arXiv:2601.10955*.

Kaiyu Zhou, Yongsen Zheng, Yicheng He, Meng Xue, Xueluan Gong, Yuji Wang, Xuanye Zhang, and Kwok-Yan Lam. 2026b. Beyond max tokens: Stealthy resource amplification via tool calling chains in llm agents. *Preprint*, arXiv:2601.10955.

Zhenhong Zhou, Zherui Li, Jie Zhang, Yuanhe Zhang, Kun Wang, Yang Liu, and Qing Guo. 2025. Corba: Contagious recursive blocking attacks on multi-agent systems based on large language models. *Preprint*, arXiv:2502.14529.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, and 1 others. 2024. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*.

Zhenhao Zhu, Yue Liu, Zhiwei Xu, Yingwei Ma, Hongcheng Gao, Nuo Chen, Yanpei Guo, Wenjie Qu, Huiying Xu, Zifeng Kang, Xinzhong Zhu, and Jiaheng Zhang. 2025a. Extendattack: Attacking servers of lrms via extending reasoning. *Preprint*, arXiv:2506.13737.

Zihao Zhu, Hongbao Zhang, Ruotong Wang, Ke Xu, Siwei Lyu, and Baoyuan Wu. 2025b. To think or not to think: Exploring the unthinking vulnerability in large reasoning models. *arXiv preprint arXiv:2502.12202*.

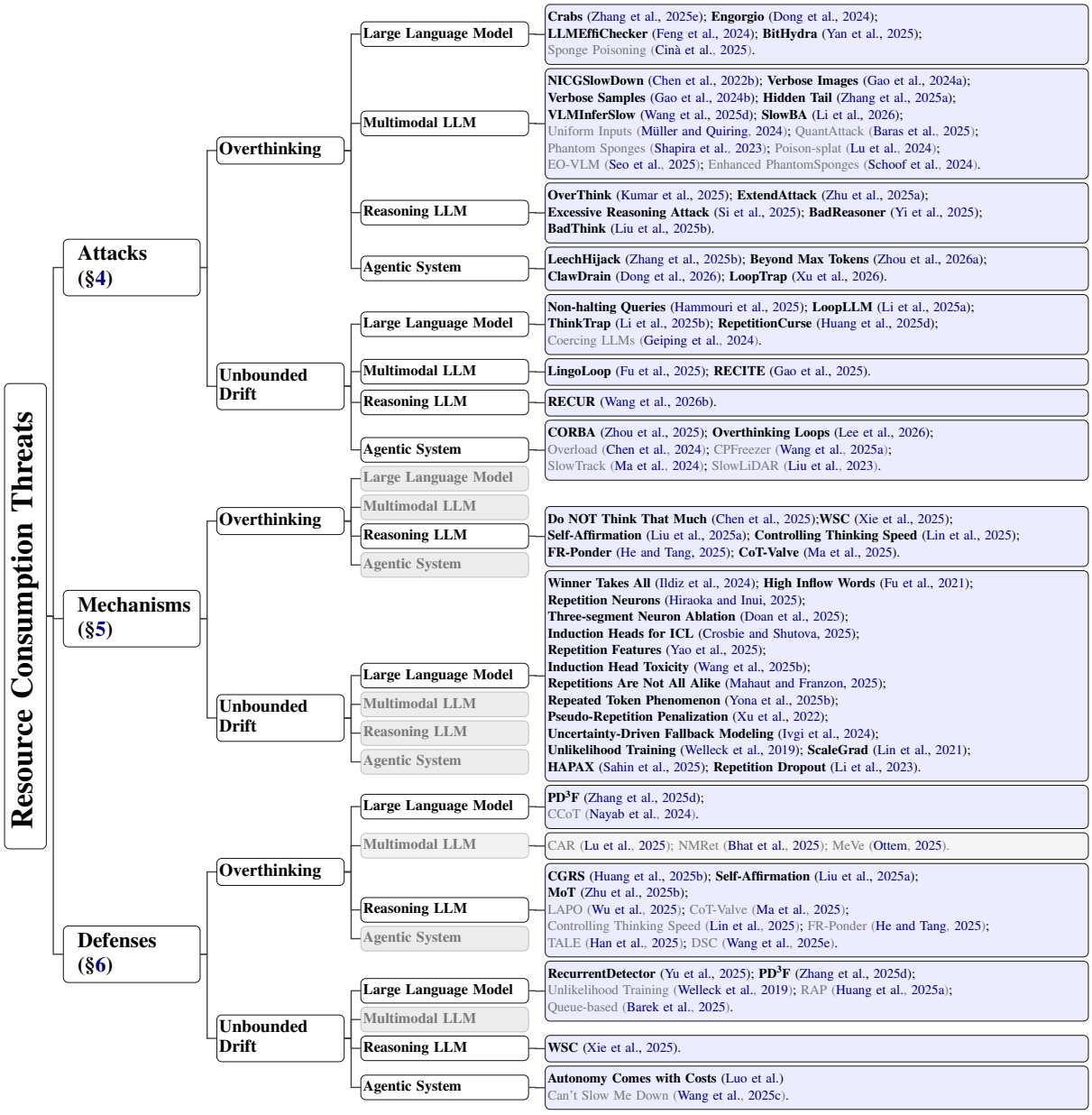


Figure 3: Scope-aware organization of resource-consumption threats reviewed in this survey. **Bold** entries are in-scope methods discussed in the main text. Gray entries denote historical or adjacent studies retained only for context and outside the core generative-LLM resource-consumption threat scope. Gray model nodes indicate branches without in-scope child methods.

## Ethical Considerations

This survey reviews resource consumption attacks on LLMs. Although summarizing attack mechanisms may increase understanding of such threats, our goal is strictly defensive: to clarify the threat landscape, support standardized evaluation, and encourage effective mitigation. We therefore focus on conceptual analysis and system-level implications rather than actionable attack instructions.

We further emphasize that resource consumption abuse can have real-world consequences beyond efficiency degradation. In shared LLM infrastructure, excessive generation may reduce service availability, increase operational costs, and harm end users through higher latency, unstable access, and higher usage costs. These risks can become even more severe in agentic systems, where recursive execution and covert workload amplification may disrupt workflows and cause hidden financial loss. By synthesizing this emerging area, we aim to support safer, more reliable, and more sustainable LLMs.

## A Scope and Inclusion Criteria

This survey focuses on threats to resource consumption in LLMs. Our primary focus is on adversarial or malicious resource abuse that increases computational cost through extended, excessive, or uncontrolled generation. Accordingly, we treat attacks on LLMs, reasoning models, multimodal generative systems, and agentic pipelines as the core literature when they induce abnormal output length, non-convergent decoding, recursive execution expansion, or related forms of resource amplification.

**Literature search strategy.** We used a manually curated search and snowballing procedure to construct the candidate literature set. For the core literature, we first examined publicly available papers since 2024 in major NLP and machine learning venues, including NeurIPS, ICLR, ICML, ACL, EMNLP, and AAAI, as well as major security venues, including IEEE S&P, ACM CCS, USENIX Security, and NDSS. We searched for work related to resource consumption, excessive generation, overthinking, non-convergent decoding, latency amplification, sponge-style attacks, recursive execution, and resource abuse in LLMs, reasoning models, multimodal models, and agentic systems. We then performed backward and forward citation expansion. For each initially identified paper, we inspected its cited references and used Google Scholar to identify papers that cited it. This iterative snowballing process produced more than 120 candidate works before screening.

**Screening and inclusion.** We screened the candidate set in two stages. First, we removed works whose connection to resource-consumption threats was only incidental, such as papers focusing primarily on benign inference acceleration, hardware scheduling, model compression, or generic service availability without a generation-level resource-abuse mechanism. Second, we applied the scope definition in Section 2 to distinguish core in-scope work from supporting or adjacent work. A paper is treated as core literature when it studies an adversarial or malicious mechanism that increases computational cost through excessive generation, repetitive decoding, recursive execution, tool-use expansion, or similar generative trajectories. A paper is treated as supporting literature when it does not directly study LLM resource-consumption threats, but provides historical context, analogous attack patterns, or mechanistic evidence that helps explain later resource abuse in generative systems.

**Treatment of recent preprints.** Because resource-consumption threats in LLMs have developed rapidly during 2025–2026, a non-trivial portion of directly relevant work is currently available only as arXiv preprints or technical reports. We include such work when it clearly satisfies the scope criteria above and provides sufficient technical detail to support classification. However, we treat these papers cautiously: they are used to characterize emerging attack patterns and defense directions, rather than to establish mature consensus. When possible, we distinguish peer-reviewed work from recent preprints in the discussion and avoid relying on a single preprint as the sole basis for a broad conclusion.

**Treatment of borderline and historical work.** We also discuss a limited set of closely related works from vision, systems security, and earlier deep learning literature, including selected latency and sponge-style attacks. These works are included only in a supporting role when they provide historical precursors, analogous threat patterns, or mechanistic insights that help explain later resource abuse in generative systems. They should not be interpreted as equally central instances of the problem studied in this survey. By contrast, we do not aim to comprehensively review benign efficiency opti-

mization, generic inference acceleration, hardware scheduling, or all forms of denial-of-service and availability attacks outside the generative process itself. Such work is referenced only when it directly informs the security perspective on resource-consumption threats or helps contextualize the survey boundary.

# B Preliminary

## B.1 Autoregressive Generation

Most modern LLMs produce outputs through an autoregressive decoding process (Vaswani et al., 2017; Malach, 2024). Given an input prompt $x$, the model generates an output sequence $y = (y_1, y_2, \ldots, y_T)$ token by token according to the conditional distribution:

$$p(y_t \mid x, y_{<t}), \tag{1}$$

where $y_{<t}$ denotes the previously generated tokens and $T$ is the final generation length. At each decoding step, the model computes a probability distribution over the vocabulary conditioned on the prompt and previously generated tokens, and selects the next token via sampling or deterministic decoding.

The decoding process continues iteratively until a termination condition is satisfied. Common termination criteria include emitting a special end-of-sequence token, reaching a predefined maximum length, or satisfying task-specific stopping rules. Consequently, the overall generation process can be viewed as a sequence of decoding steps, each extending the partial output and updating the model's internal state.

Because each token requires an additional decoding step, the generation length $T$ directly determines the number of forward passes executed during inference. As a result, the computational cost of a generation process grows with the length of the generated sequence. This characteristic makes generation length a central factor in determining the resource usage of modern LLMs.

## B.2 Computational Cost of Generation

The computational cost of autoregressive generation arises from repeated forward passes through large neural networks. For a model with parameters $\theta$, generating a sequence of length $T$ requires evaluating the model $T$ times to compute the conditional probabilities $p(y_t \mid x, y_{<t}; \theta)$.

In addition to repeated model evaluation, attention-based architectures introduce further computational overhead. Self-attention mechanisms require each token to attend to all previously generated tokens, leading to increased memory usage and computation as the sequence length increases. Consequently, the total computational cost of generation can be approximated as a function of the output length:

$$C(T) = \sum_{t=1}^{T} c_t, \tag{2}$$

where $c_t$ denotes the cost of the $t$-th decoding step. In practice, $c_t$ may grow as the context window expands, leading to increasing latency and memory consumption for longer sequences.

These computational characteristics directly translate into operational costs in real-world deployments. Longer generation sequences lead to higher GPU utilization, increased inference latency, and greater energy consumption. For this reason, many commercial LLM services use token-based pricing models, charging users based on the number of generated tokens. Under such pricing schemes, longer outputs correspond directly to higher economic costs.

Therefore, generation length becomes a critical resource variable in LLMs. Any behavior that unnecessarily increases the number of generated tokens may significantly amplify computational cost and degrade system throughput.

## B.3 Generation Trajectories

The decoding process can be viewed as a trajectory that describes how the model progresses from an input prompt to a final output. Formally, a generation process can be represented as a trajectory:

$$\tau = (x, y_1, y_2, \ldots, y_T), \tag{3}$$

which evolves over successive decoding steps.

Under normal conditions, the trajectory converges once the model produces a complete response that satisfies the task objective. However, the dynamics of this trajectory may be altered by adversarial prompts, system-level manipulations, or unexpected interaction patterns. In such cases, the generation trajectory may extend significantly beyond the length required for task completion.

Because computational cost is tightly coupled with the trajectory length, deviations in trajectory dynamics can directly amplify resource consumption. Understanding how generation trajectories evolve is therefore essential for analyzing resource consumption behaviors in LLMs.

## C Historical and Adjacent Precursors

**Early Forms of Attacks** The challenge of resource consumption attacks originated in early deep learning architectures. Specifically, Sloth (Hong et al., 2020) revealed that gradient-based optimization could coerce DNNs into the most computationally expensive inference paths. Building on this concept of computational asymmetry, the seminal work on this attack, Sponge Examples (Shumailov et al., 2021), demonstrated that targeted perturbations of neural activations could dramatically increase energy consumption by disrupting hardware-level execution. This principle was quickly extended to Neural Machine Translation (NMT), where TranSlowDown (Chen et al., 2021) and NMTSloth (Chen et al., 2022a) showed that minute perturbations delaying the End-of-Sentence (EOS) token could force sequences toward maximum length limits, significantly inflating inference costs.

As research progressed, the attack surface expanded from inference-time perturbations to training-time backdoors. Prior work showed that efficiency-targeted backdoors can manipulate model behavior by hijacking inference control flow, including early model integrity attacks (Cinà et al., 2021), later studies on dynamic routing (Chen et al., 2023), and other deep architectures (Meftah et al., 2025). This was further refined by *Sponge Poisoning* (Cinà et al., 2025; Lintelo et al., 2024), which embeds backdoors to disrupt hardware-level sparsity and skip-connection logic. These early paradigms also exposed the practical severity of resource-oriented attacks on edge hardware (Wang et al., 2023b; Hasan et al., 2025).

Prior work showed that computational overhead can lead to direct physical consequences, including battery drain and device-level Denial-of-Service (DoS) attacks. These attacks reveal that adversarial inputs can amplify seemingly small perturbations into disproportionately high computational costs.

## D The Rapid Growth of Context and Cost in LLM

Recent LLMs have expanded rapidly in both capability and serving scale, with context windows and deployment budgets increasing in parallel. As shown in Table 6, current mainstream APIs already exhibit substantial variation in usage costs, especially for high-end, reasoning-oriented, and frontier models, whose output-token pricing remains significantly higher than that of lightweight alternatives. At the same time, Table 5 shows that the context windows of representative commercial models have reached 128K, 200K, and even 1M tokens, making long-context generation an increasingly common operating regime rather than a special-case setting. This trend is further reflected in Table 7, where both mainstream and frontier models exhibit rapid year-over-year growth in supported context length.

| Model | Context Window |
|---|---|
| GPT-5.4 / GPT-5.4-pro | 1M |
| Claude Opus 4.6 | 200K (1M beta) |
| Claude Sonnet 4.6 | 200K (1M beta) |
| Gemini 2.5 Pro | 1M |
| Gemini 2.5 Flash | 1M |
| Mistral Large 2 | 128K |
| DeepSeek-V3 / R1 | 128K |

Table 5: Historical Growth of Context Windows in Mainstream and Frontier Models

Taken together, these trends indicate that modern LLMs are no longer optimized solely for short-response interactions but are increasingly designed to support long-horizon reasoning, retrieval, and multi-step execution. As context windows and output budgets continue to expand, the economic upper bound of a single interaction also rises accordingly. Under such conditions, any unnecessary expansion in reasoning length, retrieval scope, or execution trajectory can directly translate into higher API expenditure and a larger system-side computational burden. This makes resource consumption analysis increasingly important, not only for understanding attack surfaces but also for characterizing the practical cost environment in which such threats emerge.

## E Operational Criteria and Generation-Signal Evidence for Overthinking and Unbounded Drift

This appendix further operationalizes the outcome-based taxonomy within the resource consumption threat landscape reviewed in this survey. The analysis is not intended as a request-level detector for identifying whether an input is malicious, but as a characterization framework for resource-amplifying trajectories induced by known attack methods. Given such a trajectory, we examine the

| Model | Input | Output |
|---|---|---|
| GPT-5.4 (<272K context length)[a] | $2.50 | $15.00 |
| GPT-5.4 (>272K context length)[a] | $5.00 | $22.50 |
| GPT-5.4-pro (<272K context length)[a] | $30.00 | $180.00 |
| GPT-5.4-pro (>272K context length)[a] | $60.00 | $270.00 |
| GPT-5 mini[a] | $0.25 | $2.00 |
| Claude Opus 4.6[b] | $5.00 | $25.00 |
| Claude Sonnet 4.6[b] | $3.00 | $15.00 |
| Claude Haiku 4.5[b] | $1.00 | $5.00 |
| Gemini 2.5 Pro (≤200K prompt)[c] | $1.25 | $10.00 |
| Gemini 2.5 Pro (>200K prompt)[c] | $2.50 | $15.00 |
| Gemini 2.5 Flash[c] | $0.15 | $1.25 |
| Gemini 2.5 Flash-Lite[c] | $0.10 | $0.40 |
| Gemini 2.0 Flash[c] | $0.05 | $0.20 |
| Mistral Large 2[d] | $2.00 | $6.00 |
| Mistral Medium 3 / 3.1[d] | $0.40 | $2.00 |
| Mistral Small[d] | $0.20 | $0.60 |
| DeepSeek-V3[e] | ∼$0.27 | ∼$1.1 |
| DeepSeek-R1[e] | ∼$0.55 | ∼$2.2 |

[a] OpenAI pricing page: https://developers.openai.com/api/docs/pricing
[b] Anthropic pricing page: https://platform.claude.com/docs/en/about-claude/pricing
[c] Google AI pricing page: https://ai.google.dev/pricing
[d] Mistral pricing page: https://mistral.ai/pricing#api
[e] DeepSeek pricing page: https://api-docs.deepseek.com/quick_start/pricing/

Table 6: API pricing comparison of representative mainstream models. Prices are listed in USD per 1M input/output tokens.

| Models | 2020 | 2021 | 2022 | 2023 | 2024 | 2025 |
|---|---|---|---|---|---|---|
| Mainstream Models | 0.5K | 2K | 8K | 32K | 128K | 128K |
| Frontier Models | 1K | 4K | 16K | 128K | 512K | 10M |

Table 7: Context Window Comparison of Representative Mainstream Models

outcome regime of its excessive segment: Overthinking describes non-degenerate but unnecessary continuation, while Unbounded Drift describes self-sustaining, repetitive, or non-convergent continuation with little semantic progression.

The classification unit is the attack-induced resource abuse, especially the segment that accounts for the excess computational burden. In text-only generation, this segment corresponds to the generated continuation; in reasoning, multimodal, or agentic settings, it may correspond to reasoning traces, hidden generated tokens, recursive messages, or tool-use chains. The category is assigned according to the outcome regime of this excessive segment rather than the attack interface or the specific construction method.

We classify an attack trajectory as Overthinking when the excessive segment remains non-degenerate. In this regime, the model continues to reason, explain, verify, or invoke tools beyond practical utility, but the trajectory still exhibits semantic progression and does not settle into a stable repetitive loop. The output may be unnecessarily long, redundant, or costly, but it remains broadly coherent and locally meaningful.

We classify an attack trajectory as Unbounded Drift when the excessive segment becomes self-sustaining, repetitive, recursive, or non-convergent.

In this regime, the resource burden is driven not merely by verbosity, but by a degenerate continuation pattern with little semantic progression. The trajectory often persists until an external decoding limit, timeout, or runtime interruption is reached. For mixed trajectories, where an initially coherent prefix is followed by a repetitive or non-convergent tail, we classify the case as Unbounded Drift if the tail dominates the later decoding process and accounts for the main resource burden.

We use token-level generation signals as measurable diagnostic evidence for this classification. Specifically, we record the top-1 token logit, EOS logit, EOS rank, generated-token confidence, and token-level entropy at each decoding step. These signals are not intended as universal hard thresholds, because their absolute scales vary across models, tokenizers, prompts, and decoding settings. Instead, they support a model-specific and segment-level classification rule: Overthinking is expected to preserve non-degenerate fluctuations in confidence, entropy, and logits, whereas Unbounded Drift is expected to show a concentrated tail, such as repeated continuations, high generated-token confidence, low token-level entropy, stable top-token dominance, and persistent failure to terminate naturally.

The Figure 4–15 are generated under a controlled trajectory-comparison setting. For each model, we plot three decoding trajectories: a normal task-completion trajectory, an Overthinking trajectory induced by Engorgio (Dong et al., 2024), and an Unbounded Drift trajectory induced by LoopLLM (Li et al., 2025a). The normal trajectory is included as a model-specific reference for the natural scale of logits, entropy, confidence, and termination behavior, while the two attack-induced trajectories are used to examine how the proposed categories manifest during decoding. Across the figures, Engorgio produces substantially longer continuations but preserves non-degenerate fluctuations in confidence, entropy, and logits, which is consistent with Overthinking. In contrast, LoopLLM drives the generation into a repetitive tail where generated-token confidence becomes highly concentrated, token-level entropy collapses, and the top-1 token remains dominant over EOS until the external decoding limit is reached. These observations are consistent with our classification criterion: Overthinking corresponds to excessive but non-degenerate continuation, whereas Unbounded Drift corresponds to a self-sustaining repetitive or non-convergent trajectory.

## F Additional Threat Scenarios

This appendix supplements the main taxonomy by summarizing representative deployment scenarios in which resource-consumption threats may arise. Since LLM inference cost is closely tied to the generation trajectory, adversarially induced long, repetitive, or non-convergent outputs can translate into higher latency, increased monetary cost, and reduced service availability.

**Overthinking.** In the Overthinking regime, the generation remains broadly aligned with the user task but becomes unnecessarily verbose or computationally intensive. Such behavior may be triggered by prompts or inputs that induce excessive reasoning (Zhu et al., 2025a; Kumar et al., 2025), redundant explanations (Shumailov et al., 2021), or prolonged intermediate reasoning chains in reasoning-oriented models (Wei et al., 2022; Guo et al., 2025a; Si et al., 2025; Yi et al., 2025). Similar amplification can also appear in multimodal and agentic systems, where limited semantic input or repeated tool interactions may expand the generation trajectory (Gao et al., 2025; Yao et al., 2022; Xi et al., 2025; Wang et al., 2024). In practical deployments, these behaviors can inflate API costs, delay task completion, and reduce system throughput while still appearing benign at the output level. This makes Overthinking particularly relevant to commercial LLM services, agent workflows, and resource-constrained deployments.

**Unbounded Drift.** In the Unbounded Drift regime, the generation trajectory becomes repetitive, self-sustaining, or non-convergent, often continuing until external limits intervene. This behavior can be induced by adversarial prompts (Gao et al., 2025), decoding perturbations (Geiping et al., 2024), recursive reasoning structures (Wang et al., 2026b), or tool-level manipulation in agentic environments (Wang et al., 2023a; Zhou et al., 2026a; Lee et al., 2026). Compared with Overthinking, Unbounded Drift is more directly disruptive because it can block task completion, monopolize shared inference resources, and degrade service availability. In public-facing services, such inputs may function as inference-time denial-of-service attacks by consuming disproportionate GPU time, queue capacity, or memory resources (Gao et al., 2024c; Li et al., 2025b; Wang et al., 2026a; Zhang et al., 2025e). In agentic systems, recursive tool-

Table 8: Representative deployment scenarios for resource-consumption threats.

| Regime | Scenario | Attack surface | Potential impact |
|---|---|---|---|
| ***Overthinking*** | Commercial LLM services | Verbose reasoning prompts or inflated reasoning pipelines | Higher API cost and reduced throughput |
| | Agent-based workflows | Extra reasoning steps, auxiliary tool calls, or low-value instructions | Silent budget consumption and delayed completion |
| | Resource-constrained deployment | Excessive generation on limited devices or small providers | Latency increase and service degradation |
| ***Unbounded Drift*** | User-facing pipelines | Hidden suffixes or injected instructions | Longer wait time, unstable output, and higher cost |
| | Public-facing LLM services | Termination suppression or repetitive decoding triggers | GPU-time monopolization and availability degradation |
| | Agentic environments | Recursive tool calls or self-amplifying interaction chains | Task failure, stalled workflows, and system-level disruption |

calling or self-amplifying interaction chains may further stall workflows, corrupt task usefulness, and propagate resource pressure across dependent components (Zhou et al., 2025; Zhang et al., 2025b).

Overall, these scenarios show that the two regimes in Table 1 are not merely descriptive categories. They also correspond to different practical risks: Overthinking mainly causes covert cost inflation and throughput degradation, whereas Unbounded Drift more directly threatens task completion and service availability.

## G A Unified View of the Survey Organization

To clarify the scope and internal structure of this survey, Figure 3 provides a unified overview of how the literature on resource consumption issues is organized in this work. Specifically, we structure the problem into two primary resource consumption regimes, *Overthinking* and *Unbounded Drift*, which serve as the main taxonomy throughout the paper. Building on this taxonomy, the survey is further developed from three complementary perspectives: **attacks**, which examine how resource amplification is induced in different systems; **mechanisms**, which analyze the underlying dynamics that lead to excessive or non-convergent generation; and **defenses**, which summarize current mitigation strategies and their coverage.

Within each perspective, existing studies are further grouped by system type, including large language models, reasoning models, multimodal large language models, and agentic systems. This organization is intended to provide a consistent view of the field: the taxonomy captures *what kinds* of resource consumption behaviors emerge, while the three research perspectives capture *how they are induced*, *why they arise*, and *how they may be mitigated*. In this way, the figure serves as a compact map of the survey, highlighting both the coverage of current research and the uneven maturity of different branches.

Resource consumption threats can also be categorized by more fine-grained dimensions, such as attack interface, deployment layer, or adversarial objective. We treat these dimensions as useful secondary descriptors, but do not use them as the primary taxonomy because they do not consistently align attacks, mechanisms, and defenses. The same outcome may arise across different interfaces and deployment settings, while requiring similar detection signals or mitigation strategies. We therefore organize the survey around observable resource consumption outcomes, using other dimensions to refine the discussion of specific works rather than to define the main structure. This organization helps readers compare studies that appear different at the surface level but share similar resource-abuse behaviors, and it also helps future work identify reusable detection signals, transferable mitigation strategies, and underexplored gaps across model types and deployment settings.

## H Category-wise Summary of Reviewed Studies

To complement the narrative discussion in the main text and Appendix C, we provide category-wise summary tables of representative studies. These tables are intended as a compact reference map rather than a replacement for the outcome-based taxon-

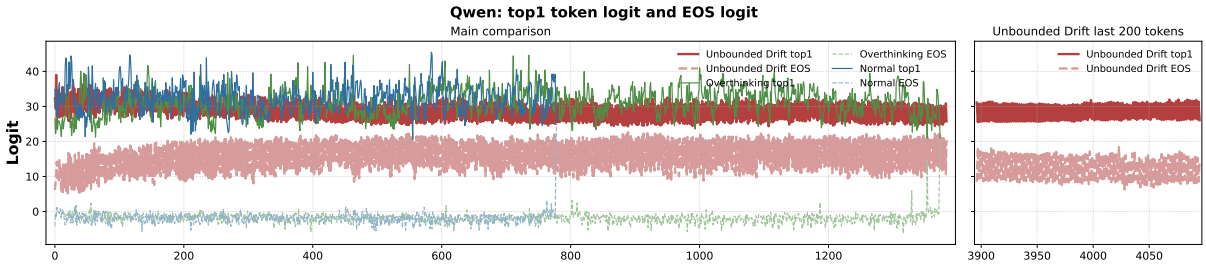


Figure 4: Top-1 and EOS logits for Normal, Overthinking, and Unbounded Drift. Unbounded Drift shows a stable tail pattern in which the recurrent top-1 token remains dominant over EOS.

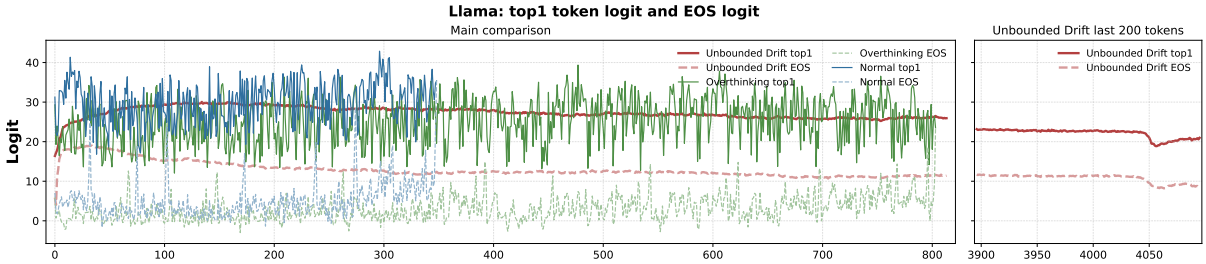


Figure 5: Top-1 and EOS logits for Normal, Overthinking, and Unbounded Drift. Unbounded Drift shows a stable tail pattern in which the recurrent top-1 token remains dominant over EOS.

omy. They summarize each line of work by method, dataset, evaluation metrics, evaluated models, and reported efficiency or resource-consumption signals. Specifically, Tables 9–15 cover early adaptive architectures, text-only LLM attacks, multimodal LLM attacks, reasoning LLM attacks, agentic resource-amplification attacks, and representative mitigation methods.

# I AI Writing Assistance Disclosure

We used AI tools solely for language polishing to improve clarity and readability. The AI tools did not contribute to the scientific content, ideas, analyses, or conclusions of this work.

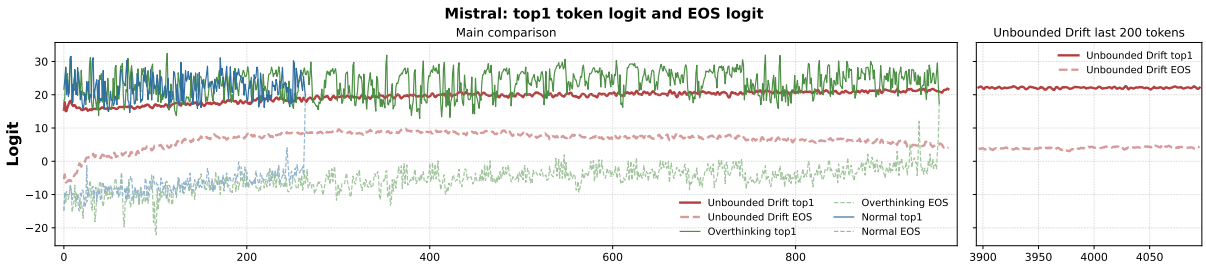


Figure 6: Top-1 and EOS logits for Normal, Overthinking, and Unbounded Drift. Unbounded Drift shows a stable tail pattern in which the recurrent top-1 token remains dominant over EOS.

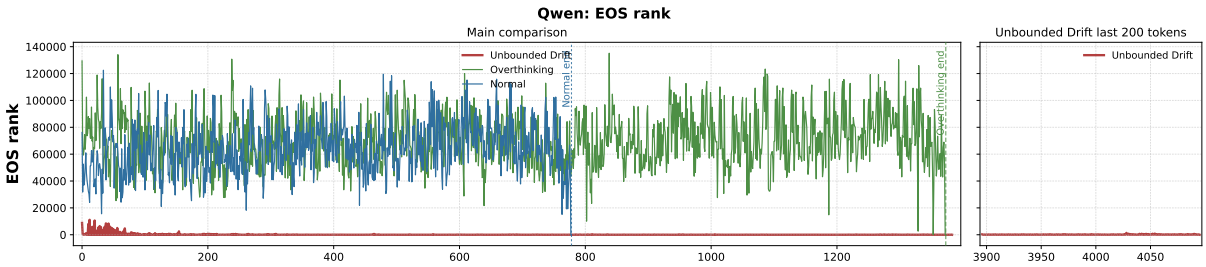


Figure 7: EOS rank for Normal, Overthinking, and Unbounded Drift.

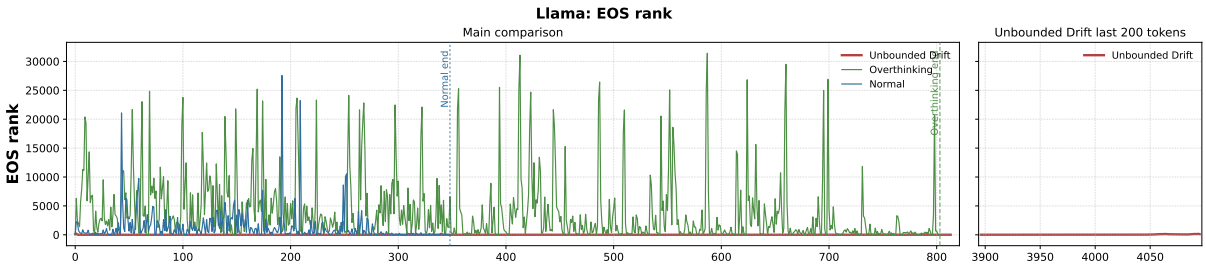


Figure 8: EOS rank for Normal, Overthinking, and Unbounded Drift.

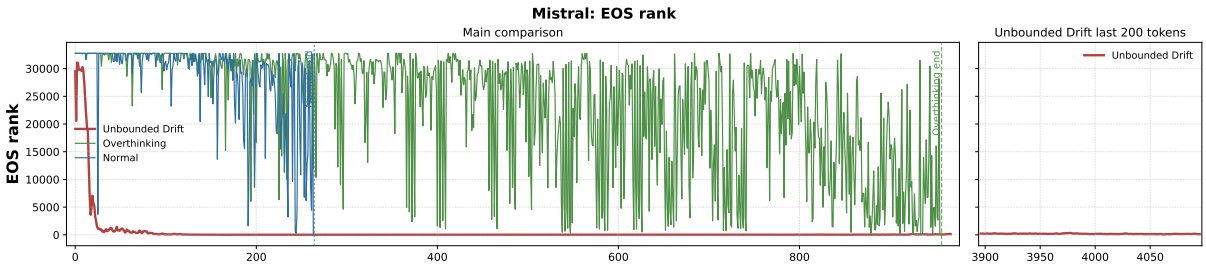


Figure 9: EOS rank for Normal, Overthinking, and Unbounded Drift.

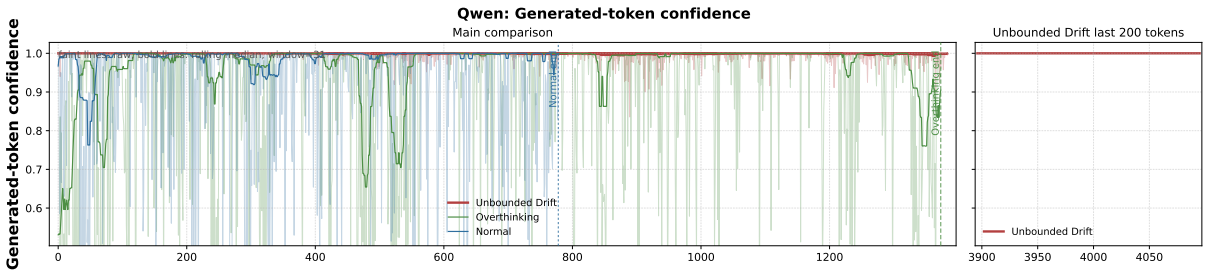


Figure 10: Generated-token confidence for Normal, Overthinking, and Unbounded Drift. Unbounded Drift converges to a near-deterministic high-confidence tail, while Overthinking remains longer but still fluctuates.

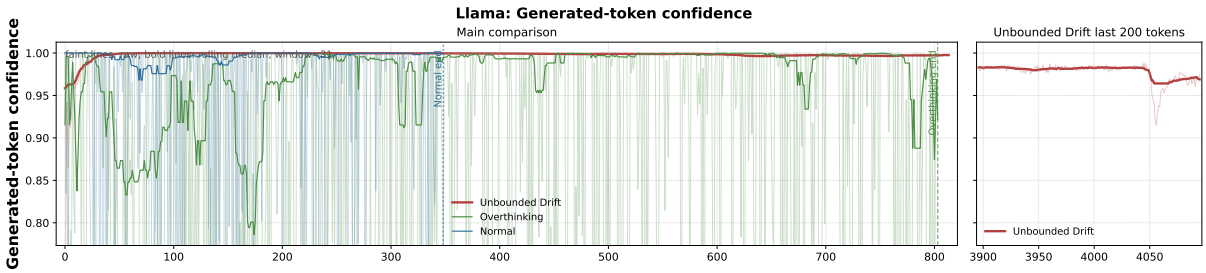


Figure 11: Generated-token confidence for Normal, Overthinking, and Unbounded Drift. Unbounded Drift converges to a near-deterministic high-confidence tail, while Overthinking remains longer but still fluctuates.

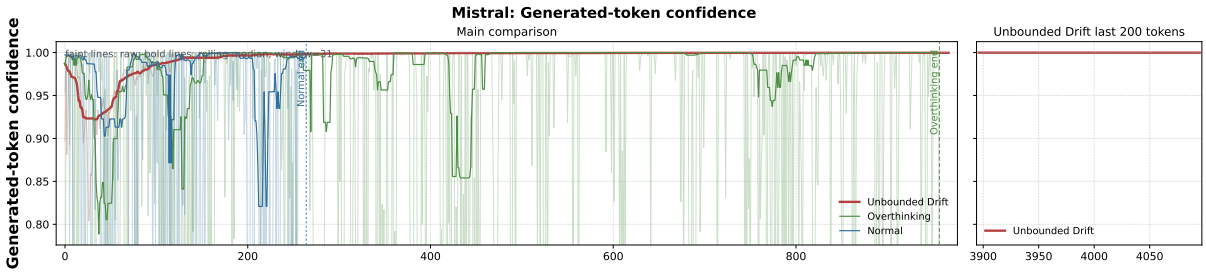


Figure 12: Generated-token confidence for Normal, Overthinking, and Unbounded Drift. Unbounded Drift converges to a near-deterministic high-confidence tail, while Overthinking remains longer but still fluctuates.

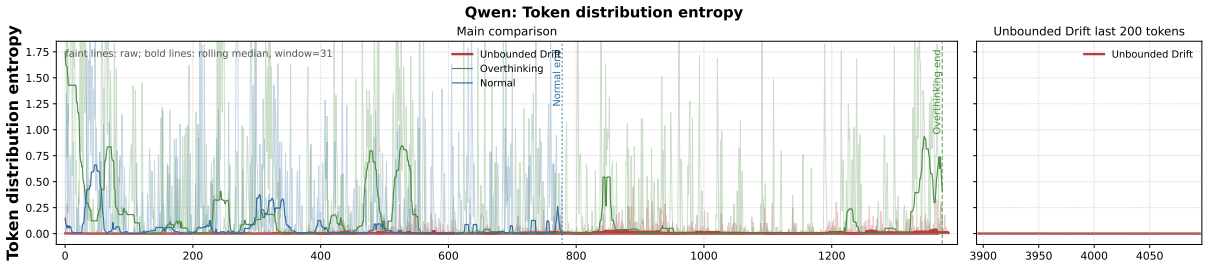


Figure 13: Token distribution entropy for Normal, Overthinking, and Unbounded Drift. Overthinking retains non-degenerate entropy fluctuations, whereas Unbounded Drift collapses into a near-zero-entropy tail.

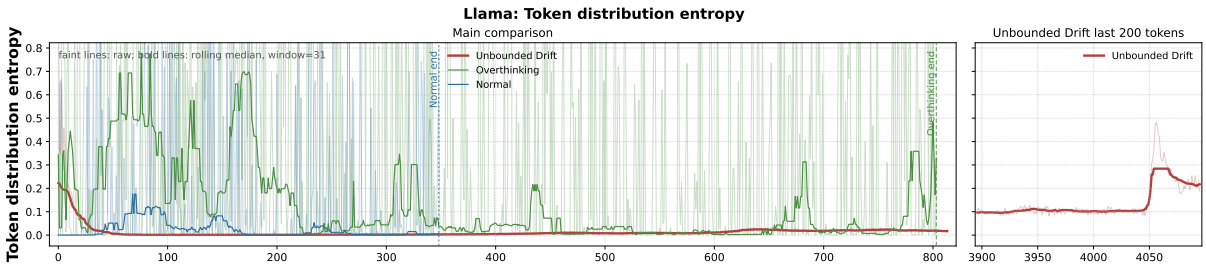


Figure 14: Token distribution entropy for Normal, Overthinking, and Unbounded Drift. Overthinking retains non-degenerate entropy fluctuations, whereas Unbounded Drift collapses into a near-zero-entropy tail.

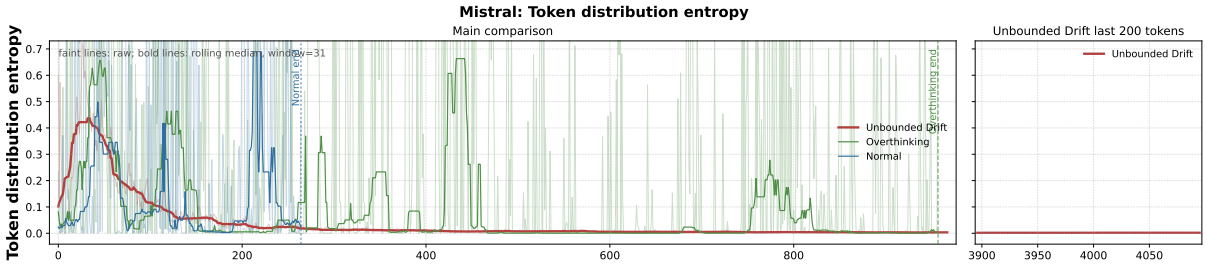


Figure 15: Token distribution entropy for Normal, Overthinking, and Unbounded Drift. Overthinking retains non-degenerate entropy fluctuations, whereas Unbounded Drift collapses into a near-zero-entropy tail.

| Method | Dataset | Evaluation Metrics | Models | Efficiency |
|---|---|---|---|---|
| **A Sloth (Multi-exit)** (Hong et al., 2020) | CIFAR-10, ImageNet, Tiny-ImageNet | Exit Index, Avg Latency, Energy | MSDNet, ResNet (Multi-exit) | N/A |
| **Sponge Examples** (Shumailov et al., 2021) | CIFAR-10, SVHN, WikiText-103, En–De | Energy Consumption, Latency, Memory Access | VGG-16, ResNet-18, GPT-2 | 512 |
| **TranSlowDown** (Chen et al., 2021) | WMT'14 (En–De), WMT'16 (En–Ro) | Translation Latency, Token Count, BLEU Score | Transformer, LSTM-based NMT | 1024 |
| **NMTSloth** (Chen et al., 2022a) | WMT'17 (En–De), WMT'19 (Zh–En) | Real-world Latency, Energy, Response Time | Fairseq, OpenNMT, MarianNMT | **80**× |
| **Dynamic Routing** (Chen et al., 2023) | CIFAR-100, ImageNet | Avg Layers, Activation Ratio | GaterNet, SkipNet | N/A |
| **On-Device Sponge** (Wang et al., 2023b) | Speech Commands, IMDB | Battery Drain, Execution Time | MobileNetV2, ShuffleNet | N/A |
| **SkipSponge** (Lintelo et al., 2024) | CIFAR-10, SVHN | FLOPs Inflation, Accuracy | ResNet, WideResNet | N/A |
| **Energy Backdoor** (Meftah et al., 2025) | GLUE, SQuAD | Energy Increase, Clean Accuracy | BERT, RoBERTa | 512* |
| **Sponge Poisoning** (Cinà et al., 2025) | ImageNet, CIFAR | Latency, GPU Energy | VGG, ResNet | N/A |
| **Sensing AI Sponge** (Hasan et al., 2025) | Audio/Sensor Data | Sensor Power, Pruning Ratio | CNN, DeepSense | N/A |

Table 9: Chronological summary of historical and adjacent resource consumption studies. Gray text denotes works outside the core generative-LLM resource consumption threat scope of this survey.

| Method | Dataset | Evaluation Metrics | Models | Efficiency |
|---|---|---|---|---|
| **Coercing LLMs** (Geiping et al., 2024) | AdvBench, Vicuna-bench | ASR, Safety, Efficiency Loss | Vicuna, Llama-2, GPT-3.5 | 4096 |
| **LLMEffiChecker (Feng et al., 2024)** | Efficiency-Bench, WikiText | Delay, Energy, Throughput | Llama-2, Vicuna, Alpaca | 8192 |
| **DoS Poisoning** (Gao et al., 2024c) | Anthropic HH-RLHF | Inference Latency, Poisoning Ratio | GPT-Neo, RoBERTa | 1024 |
| **Engorgio Prompt (Dong et al., 2024)** | ShareGPT, Alpaca | Response Length, Energy Draw | Llama-3, Mistral, GPT-4 | 8192 |
| **Repeated Token Phenomenon** (Yona et al., 2025a) | OpenWebText, Custom Repetition Dataset | Attention Scores, Loss, Output Length | Llama-3, GPT-2, Pythia | 8192 |
| **Non-halting Queries (Hammouri et al., 2025)** | RAG Systems, 10k Probe Dataset | Avg Token Length, Generation Success | GPT-4o, Llama-3, Gemma-2 | 8192 |
| **Crabs (AutoDoS) (Zhang et al., 2025e)** | ChatDoctor, MMLU, HellaSwag, CodeXGLUE, GSM8K | Attack Success Rate, Task Success, Token Usage, Latency, Compute Cost, GPU Memory | GPT-4, Llama-3, Qwen-2, DeepSeek, Ministral, Gemma | 8192 |
| **BitHydra (Yan et al., 2025)** | MMLU, GSM8K | Inference Cost, Bit-flip Rate, ASR | Llama-2, OPT, BLOOM | 8192 |
| **LoopLLM (Li et al., 2025a)** | MT-Bench, Chatbot Arena | Repetition Ratio, Energy Latency | Llama-3, Qwen-2, Phi-3 | 8192 |
| **ThinkTrap (Li et al., 2025b)** | Sponge, LLMEffiChecker | Throughput, Response Latency, ASR | GPT-4o, Gemini 2.5 Pro, DeepSeek-R1 | 8192 |

Table 10: Chronological summary of representative resource consumption attacks and adjacent studies in text-only LLMs. Gray text denotes methods outside the core LLM resource consumption attack scope of this survey.

| Method | Dataset | Evaluation Metrics | Models | Efficiency |
|---|---|---|---|---|
| **Sponge Examples** (Shumailov et al., 2021) | CIFAR-10, SVHN, WikiText-103, En–De | Energy Consumption, Latency, Memory Access | VGG-16, ResNet-18, GPT-2 | 512 |
| **NICGSlowDown (Chen et al., 2022b)** | MS-COCO, Flickr8k | I-Loops, I-Latency (CPU/GPU) | ResNeXt-LSTM, GoogLeNet-RNN, MobileNets-LSTM | N/A |
| **Phantom Sponges** (Shapira et al., 2023) | Berkeley DeepDrive (BDD), MTSD, LISA, PASCAL VOC | Number of Candidates, Processing Time, Detection Recall | YOLOv5, YOLOv3, YOLOv4 | N/A |
| **Verbose Images (Gao et al., 2024a)** | MS-COCO, ImageNet | Sequence Length, Energy, Latency, Uncertainty | BLIP, BLIP-2, InstructBLIP, MiniGPT-4 | 8× |
| **Verbose Samples (Gao et al., 2024b)** | MSVD, TGIF | Length, Latency, Energy | VideoChat-2, Video-Vicuna, Video-LLaMA | 4× |
| **Uniform Inputs** (Müller and Quiring, 2024) | ImageNet | Activation Density, Activation Sparsity | ResNet, DenseNet, MobileNetV2 | N/A |
| **Enhanced PhantomSponges** (Schoof et al., 2024) | Berkeley DeepDrive | Number of Candidates, Processing Time, Detection Recall | YOLOv5 | N/A |
| **QuantAttack** (Baras et al., 2025) | ImageNet | GPU Memory, Processing Time, Energy, Outlier Count, Accuracy | Vision Transformer (ViT), Data-efficient Image Transformer (DeiT) | N/A |
| **EO-VLM** (Seo et al., 2025) | N/A | GPU Power, Inference Time | YOLOv8, MaskDINO, Detectron2 | N/A |
| **VLMInferSlow (Wang et al., 2025d)** | MS-COCO, ImageNet | I-Length, I-Latency, I-Energy | Flamingo, BLIP, GIT, Florence | N/A |
| **LingoLoop (Fu et al., 2025)** | MS-COCO, ImageNet | Tokens, Energy, Latency | InstructBLIP, Qwen2.5-VL, InternVL3 | 2048 |
| **RECITE (Gao et al., 2025)** | ImageNet, MMLU, HumanEval, GSM8K | Attack Success Rate, Service Response Latency, GPU Utilization, Average Generation Length | InstructBLIP, LLaVA, Qwen-VL | 2048 |
| **Hidden Tail (Zhang et al., 2025a)** | MS-COCO | Attack Success Rate, Output Length, Latency, Visible Length, Response Quality | Qwen2.5-VL, MiMo-VL-7B-RL, Gemma-3-4B-IT | 1831 |

Table 11: Chronological summary of representative resource consumption attacks and adjacent studies related to MLLMs. Gray text denotes historical, adjacent, or non-generative vision studies outside the core MLLM resource consumption threat scope of this survey.

| Method | Dataset | Evaluation Metrics | Models | Efficiency |
|---|---|---|---|---|
| **OverThink (Kumar et al., 2025)** | FreshQA, SQuAD, MuSR | Reasoning Token Amplification, Answer Correctness, Guardrail Evasion, Defense Evaluation | o1, o1-mini, DeepSeek-R1 | 46× |
| **ExtendAttack (Zhu et al., 2025a)** | AIME 2024, AIME 2025, HumanEval, BigCodeBench-Complete | Response Length, Latency, Accuracy (Pass@1) | o3, o3-mini, QwQ-32B, Qwen3-32B | N/A |
| **Excessive Reasoning Attack (Si et al., 2025)** | GSM8K, ORCA | Reasoning Length, Output Length, Utility Performance (Accuracy) | DeepSeek-R1-Distill-LLaMA, DeepSeek-R1-Distill-Qwen, o1-mini, o3-mini, DeepSeek-R1, QwQ | 9× |
| **BadReasoner (Yi et al., 2025)** | N/A | Reasoning Verbosity, Answer Correctness, Controllability | Marco-o1, QwQ, DeepSeek-R1 | N/A |
| **BadThink (Liu et al., 2025b)** | MATH-500, GSM8K | ASR, RIR, TAC, BAD, SD | DeepSeek-R1-Distill-Qwen (1.5B/7B/14B/32B), OpenR1-Qwen-7B, Light-R1-7B-DS | 63.85× |
| **RepetitionCurse (Huang et al., 2025d)** | N/A | Prefill Latency, TTFT, Router Imbalance | Mixtral-8x7B series, Qwen3-30B-A3B series, GPT-OSS-20B/120B, Kimi-Linear-Instruct, DeepSeek-V2-Lite, Llama-4-Scout-17B-16E-Instruct | N/A |

Table 12: Chronological summary of representative resource consumption attacks and adjacent entries related to RLLMs. Gray text denotes entries outside the core RLLM resource consumption attack scope of this table.

| Method | Dataset | Evaluation Metrics | Models | Efficiency |
|---|---|---|---|---|
| **SlowLiDAR (Liu et al., 2023)** | KITTI, nuScenes | Runtime Latency, Imperceptibility | PointPillars, SECOND, PV-RCNN | 2.7× |
| **SlowTrack (Ma et al., 2024)** | MOT17 | Latency Increase (R-Lat), Imperceptibility, System-level Crash Rate | SORT, FairMOT, ByteTrack, BoT-SORT with YOLOv5 detector | 4× |
| **Overload (Chen et al., 2024)** | MS-COCO | Inference Time, NMS Latency | YOLOv5 | 10× |
| **CORBA (Zhou et al., 2025)** | LLM-MAS simulation tasks | P-ASR, PTN, Availability Degradation | GPT-4o-mini, GPT-4, GPT-3.5-turbo, Gemini-2.0-Flash, Qwen2.5-14B-Instruct, Llama-3.1-70B-Instruct, Gemma-2-27B-IT | N/A |
| **CPFreezer (Wang et al., 2025a)** | OPV2V | End-to-end Latency, Attack Success Rate, Frame Processing Time | OpenCOOD | 90× |
| **LeechHijack (Zhang et al., 2025b)** | GAIA, GPQA, MMLU | Attack Success Rate, Resource Overhead, Detectability / Stealthiness | DeepSeek, Qwen, GPT, Gemini | N/A |
| **ClawDrain (Dong et al., 2026)** | Real OpenClaw agent workloads | Token Amplification, Cost Overhead, Attack Success Rate, Stealthiness | OpenClaw v2026.2.9 | ∼9× |

Table 13: Chronological summary of representative resource consumption attacks and adjacent studies related to LLM agents. Black rows denote core agentic resource-amplification attacks that exploit tool use, recursive interaction, or execution-chain expansion. Gray rows denote adjacent perception or vehicular latency attacks outside the core LLM-agent resource-consumption threat scope.

| Method | Dataset | Evaluation Metrics | Models | Efficiency | Model Class | Defense Role |
|---|---|---|---|---|---|---|
| **Unlikelihood Training** (Welleck et al., 2019) | WikiText-103, GPT-2 fine-tuning corpus | Seq./token repetition metrics, PPL, accuracy, human win rate | 16-layer Transformer, GPT-2 medium | Token-level: 150k updates; seq.-level: 1.5k updates | LLM | Generic repetition mitigation |
| **CCoT** (Nayab et al., 2024) | GSM8K, SVAMP, ASDIV | Accuracy, generation time, token count, HCA/SCA/CCA, RMS, information flow | Llama-2 70B/7B, Falcon 40B/7B, Vicuna-13B | 5.12 s generation time ↓, 4.41% accuracy ↑ | LLM | Pure efficiency-oriented |
| **Recurrent Detector (Yu et al., 2025)** | Custom trigger dataset, ShareGPT | Accuracy, F1, FPR, recall, trigger attempts, latency | Llama-3/2 7B/13B, Vicuna-v1.5 7B/13B, Gemma-2 2B, GPT-4o/mini | Recurrent Generator: avg attempts 272.1 vs. 1679.1 random; Recurrent Detector: 0.36 ms inference | LLM | Attack-aware loop defense |
| **PD$^3$F (Zhang et al., 2025d)** | MMLU, HellaSwag, HumanEval, GSM8K, GPQA | Attack Success Rate, Task Success, Token Usage, Latency | Llama, Mistral, Qwen | Total time 50% ↓ | LLM | Attack-aware DoS defense |
| **DSC** (Wang et al., 2025e) | MATH, GSM8K, CSQA, SQA, Last Letter, Coin Flip | Accuracy, cost, tokens, time | GPT-3.5 Turbo, GPT-4, Mistral-7B-Instruct-v0.3 | Cost ↓ 65% on GPT-4, 56% on GPT-3.5 | LLM & RLLM | Pure efficiency-oriented |
| **NMRet** (Bhat et al., 2025) | CoQA, AI-arXiv2, TextVQA | RAGAS metrics | GeminiLLM, Titans PyTorch NeuralMemory, Qdrant/Chroma, CLIP embeddings, LightThinkerCompressor | LightThinker Compressor reduces token footprint for KV cache | LLM & RAG | Pure efficiency-oriented |
| **MeVe** (Ottem, 2025) | English Wikipedia, HotpotQA subset | Avg context tokens, retrieval time, grounding/relevance proxies | Embedding model, cross-encoder, tokenizer, BM25 fallback | Context tokens ↓ 57.7% on Wiki and 75% on HotpotQA | LLM & RAG | Pure efficiency-oriented |
| **CoT-Valve** (Ma et al., 2025) | GSM8K, PRM800K, MixChain C/Z; Eval: GSM8K, AIME24 | Pass@1, token count, ACU | QwQ-32B-Preview, DeepSeek-R1-Distill-Llama-8B, LLaMA-3.1/3.2, Qwen2.5-32B | Token usage 69.6% ↓ | RLLM | Pure efficiency-oriented |
| **FR-Ponder** (He and Tang, 2025) | GSM8K, MATH500, GPQA | Accuracy, avg tokens, avg FLOPs | LLaMA-3 8B/70B, Qwen-2.5 0.5B/3B/7B | 30–50% token ↓ | RLLM | Pure efficiency-oriented |

Table 14: Chronological summary of representative mitigation studies related to resource consumption threats. Black rows denote attack-aware defenses evaluated against adversarial resource amplification or non-halting behavior. Gray rows denote pure efficiency-oriented or generic mitigation methods that reduce cost, length, or repetition under benign settings but are not core security defenses against adversarial resource-consumption threats.(Part 1)

| Method | Dataset | Evaluation Metrics | Models | Efficiency | Model Class | Defense Role |
|---|---|---|---|---|---|---|
| **MoT (Zhu et al., 2025b)** | AIME 2024, MATH500, StrongReject, HarmBench, WildJailbreak | ASR, RTC, RPC, C-ACC, Min Steps, Refusal, Harmful Score | DeepSeek-R1 (1.5B/7B/14B/32B), Light-R1-7B-DS, OpenR1-7B, QwQ-32B, Marco-o1-7B | ASR > 90%, RTC ∼80% ↓, RPC ∼90% ↓ | RLLM | Attack-aware reasoning control |
| **RAP (Huang et al., 2025a)** | QA: WebQSP, MS MARCO; MT: MAC (Chinese–English) | RR, RAP Score, F1/BERT-F1/COMET | Llama-2 (7B/13B/70B), Llama-3 (8B/70B), Llama-3.1-70B, Gemma-1.1, Phi-3/3.5-mini, Mistral-7B | RR ↓ up to 93% on WebQSP and 74% on MS MARCO | LLM | Generic repetition metric / efficiency-oriented |
| **CAR (Lu et al., 2025)** | DocVQA, ChartQA, FUNSD, GSM8K | ACC, EM, Token Count, PPL | Qwen2.5-0.5B/7B, Llama-3.1-8B, Qwen2-VL-7B | Token ↓ 21–39%, ACC ↑ 5.5–6.9% | LLM, MLLM | Pure efficiency-oriented adaptive routing |
| **Underload (Wang et al., 2025c)** | PASCAL-VOC, COCO, Berkeley DeepDrive (BDD) | mAP50, FPS, NMS Latency, Compute Overhead, Memory Transfer | YOLOv3, YOLOv5, YOLOv8 | FPS 13→43 on Jetson Orin NX | Vision | Adjacent vision-only latency defense |
| **Self-Affirmation (Liu et al., 2025a)** | AIME24, AMC23, GSM8K, MATH500, GPQA-D | Accuracy, Token Count, LR Ratio | R1-Distill-Qwen (1.5B/7B/32B), QwQ-32B, Qwen3-32B | Train-free: 8.4–18.7% token ↓ | RLLM | Reflection-suppression defense-relevant control |
| **TALE (Han et al., 2025)** | GSM8K, GSM8K-Zero, MathBench | Accuracy, Output Tokens, Expense | GPT-4o-mini, Yi-lightning, GPT-4o, o3-mini, Llama-3.1-8B-Instruct | Token ↓ 64–67%, expense ↓ 45–59% | LLM, RLLM | Pure efficiency-oriented token budgeting |
| **Thinking Speed (Lin et al., 2025)** | MATH500, AIME24/25, GPQA-Diamond, LiveCodeBench | Pass@1, Token Count, Latency, Mode Switch | DeepSeek-R1-Distill-Qwen-7B/32B, QwQ-32B, Qwen3-8B | 1.26% accuracy ↑, 8.56% token ↓ | RLLM | Pure efficiency-oriented reasoning control |
| **LAPO (Wu et al., 2025)** | Train: 10k math; Eval: MATH500, AIME24, AMC23, Olympiad-Bench, GPQA | Pass@1, Token Count, Trade-off | DeepSeek-R1-1.5B, DeepScaleR-1.5B-Preview | Token ↓ up to 40.9% | RLLM | Pure efficiency-oriented policy optimization |
| **Queue-based (Barek et al., 2025)** | Queue-based web service with custom load tests | Total Time, Avg Time per Request | GPT-2, BLOOM, OPT | Avg time 21–35 s | LLM | Adjacent serving-stability study |
| **Word Salad Chopper (Xie et al., 2025)** | GSM8K, MATH500, AIME25, GPQA-Diamond | Accuracy, Length Compression Ratio | DeepSeek-R1, Qwen3-8B | Length compression 13–57% | RLLM | Loop-oriented repetition interruption |

Table 15: Chronological summary of representative mitigation studies against resource-consumption threats (Part 2). Black rows denote defense-relevant methods that are used to control adversarially amplified reasoning, reflection, or repetitive generation. Gray rows denote pure efficiency-oriented, generic, or adjacent studies that reduce cost, latency, or token usage under benign or non-core settings rather than directly defending against adversarial resource-consumption threats.