BAM-ICL: Causal Hijacking In-Context Learning with Budgeted Adversarial Manipulation

Rui Chu[1] **Bingyin Zhao**[2] **Hanling Jiang**[1] **Shuchin Aeron**[1] **Yingjie Lao**[1]

[1] Department of Electrical and Computer Engineering, Tufts University
[2] Meitu Inc

`{rui.chu, hanling.jiang, shuchin.aeron, yingjie.lao}@tufts.edu`, `bingyin@meitu.com`

Abstract

Recent research shows that large language models (LLMs) are vulnerable to hijacking attacks under the scenario of in-context learning (ICL) where LLMs demonstrate impressive capabilities in performing tasks by conditioning on a sequence of in-context examples (ICEs) (i.e., prompts with task-specific input-output pairs). Adversaries can manipulate the provided ICEs to steer the model toward attacker-specified outputs, effectively "hijacking" the model's decision-making process. Unlike traditional adversarial attacks targeting single inputs, hijacking attacks in LLMs aim to subtly manipulate the initial few examples to influence the model's behavior across a range of subsequent inputs, which requires distributed and stealthy perturbations. However, existing approaches overlook how to effectively allocate the perturbation budget across ICEs. We argue that fixed budgets miss the potential of dynamic reallocation to improve attack success while maintaining high stealthiness and text quality. In this paper, we propose BAM-ICL, a novel budgeted adversarial manipulation hijacking attack framework for in-context learning. We also consider a more practical yet stringent scenario where ICEs arrive sequentially and only the current ICE can be perturbed. BAM-ICL mainly consists of two stages: In the offline stage, where we assume the adversary has access to data drawn from the same distribution as the target task, we develop a global gradient-based attack to learn optimal budget allocations across ICEs. In the online stage, where ICEs arrive sequentially, perturbations are generated progressively according to the learned budget profile. We evaluate BAM-ICL on diverse LLMs and datasets, the experimental results demonstrate that it achieves superior attack success rates and stealthiness and the adversarial ICEs are highly transferable to other models. Code is available at `https://github.com/CRcr0/BAM-ICL`.

1 Introduction

Recent development of large language models (LLMs) has revolutionized and empowered various fields, from reasoning Wei et al. [2022], Cheng et al. [2024a], Zhang et al. [2024] to math proof Azerbayev et al. [2023], Setlur et al. [2024], Didolkar et al. [2024] to protein design Madani et al. [2023, 2020], Cheng et al. [2024b], Ferruz and Höcker [2022]. Different from conventional models, LLMs also demonstrate remarkable capabilities in handling a wide range of problems and tasks through in-context learning (ICL) (a.k.a inference-time few-shot learning Brown et al. [2020], Garg et al. [2022], Xie et al. [2022], Min et al. [2022], Wies et al. [2023], Agarwal et al. [2024]). ICL is an intrinsic capability of LLMs that allows them to generate relevant responses to unseen input queries via "learning" from in-context examples (ICEs) (i.e., a sequence of prompts with task-specific input-output pairs), without updating model parameters. Although paving an effective path to undertake a variety of tasks by observing context examples, the potential risks and threats that the ICL ability may cause remain unclear and are worth exploring.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

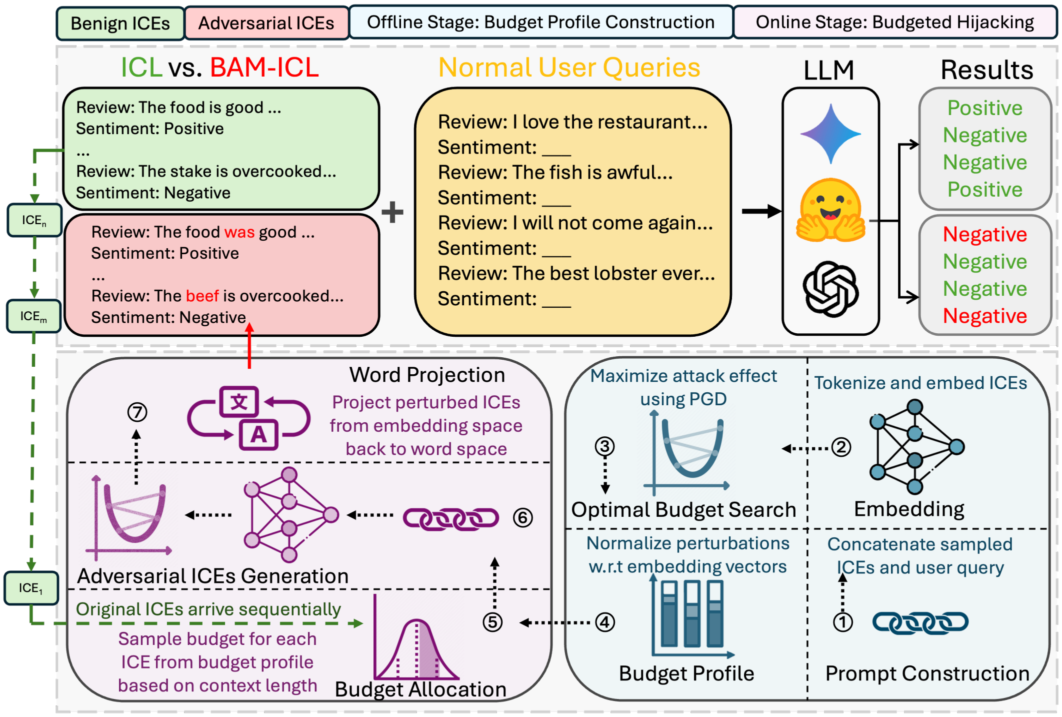

Benign ICEs
Adversarial ICEs
Offline Stage: Budget Profile Construction
Online Stage: Budgeted Hijacking
ICL vs. BAM-ICL
Review: The food is good ...
Sentiment: Positive
...
Review: The stake is overcooked...
Sentiment: Negative
Review: The food was good ...
Sentiment: Positive
...
Review: The beef is overcooked...
Sentiment: Negative
Normal User Queries
Review: I love the restaurant...
Sentiment: ___
Review: The fish is awful...
Sentiment: ___
Review: I will not come again...
Sentiment: ___
Review: The best lobster ever...
Sentiment: ___
LLM
Results
Positive
Negative
Negative
Positive
Negative
Negative
Negative
Negative
ICE_n
ICE_m
ICE_1
Word Projection
Project perturbed ICEs from embedding space back to word space
⑦
Adversarial ICEs Generation
⑥
Original ICEs arrive sequentially
Sample budget for each ICE from budget profile based on context length
⑤
Budget Allocation
Maximize attack effect using PGD
③
Optimal Budget Search
Tokenize and embed ICEs
②
Embedding
Normalize perturbations w.r.t embedding vectors
④
Budget Profile
Concatenate sampled ICEs and user query
①
Prompt Construction

Figure 1: **Top: Illustration of ICL and BAM-ICL on LLMs. Bottom: Illustration of the framework design of BAM-ICL.** BAM-ICL hijacks LLMs and yields unintended output via adversarial ICEs (the block in red) while ICL with benign ICEs (the block in green) produces normal outputs. BAM-ICL is composed of two stages, where in the offline stage (the block in sky blue) we construct the budget profile to search for optimal budget distribution for each ICE and in the online stage (the block in light purple) we sequentially perform budgeted attack to generate adversarial ICEs.

In particular, it has been shown that the ICL capability of LLMs opens the door to model hijacking attacks Si et al. [2023], Qiang et al. [2023], Li et al. [2025], Salem et al. [2022], Jeong [2023], Kuo et al. [2025], where adversarial ICEs are used to steer victim models into producing attacker-specified outputs, effectively "hijacking" the model's decision-making process. A recent work Qiang et al. [2023] extends hijacking attack strategies from vision tasks to language models by using adversarial perturbations to craft malicious in-context examples. These adversarial example-based approaches modify input prompts to influence model behavior. In contrast to adversarial attacks Miyato et al. [2017], Wang et al. [2023a], Anwar et al. [2024], Li et al. [2020], Xhonneux et al. [2024], which have recently gained significant attention in the context of LLMs, hijacking attacks leverage the ICL mechanism to manipulate the model's behavior across a range of inputs, exhibiting distinct characteristics and challenges. 1) Instead of targeting a single input for misclassification, a hijacking attack perturbs the first few examples rather than a single example, with the objective of influencing the subsequent examples. To maintain stealthiness, the overall perturbation must be constrained. However, since the perturbation is distributed across multiple examples, the magnitude applied to each individual example can be reduced. 2) This also creates opportunities to dynamically allocate the perturbation budget among ICEs, which can further improve the attack performance (we show a fixed budget is suboptimal). 3) An adversarial attack is deemed successful when it identifies an adversarial example that misleads the model for a specific input; in contrast, a hijacking attack must maximize its influence on subsequent ICEs while operating under a predefined perturbation constraint. These observations motivate the central research question we seek to address: *How to design an effective and stealthy hijacking attack against LLMs, where subtle adversarial manipulations on ICEs gradually accumulate influence, analogous to a snowball effect, to steer the model's behavior?*

In this work, we propose BAM-ICL, a budgeted adversarial manipulation hijacking attack framework against LLMs through ICL. We consider a more practical yet stringent scenario where ICEs arrive sequentially and only the current ICE can be perturbed. As shown in Fig. 1, BAM-ICL mainly consists of two stages: In the offline stage, where we assume the adversary has access to data drawn from the same distribution as the target task, we develop a global gradient-based budget profile construction algorithm to search for the optimal perturbation budget for each adversarial ICE given a total perturbation budget. In the online stage, where ICEs arrive sequentially, we progressively generate the perturbation for each ICE according to the budget profile constructed in the offline stage. To enhance the stealthiness and preserve the semantic meaning of ICEs, the perturbations are performed on the embedding space and then projected back to the word space. Our contributions are summarized as follows:

- To the best of our knowledge, this is the first work that exploits ICL to hijack LLMs with budgeted adversarial manipulation.
- We develop a novel two-stage attack framework composed of a global gradient-based attack that systematically searches for the optimal dynamic budget allocation strategy for individual ICEs and a refined causal attack that ensures the full utilization of the budget for each example.
- Through extensive experimentation on three benchmark datasets across various LLMs, we demonstrate that BAM-ICL achieves superior attack success rates and remarkable transferability while preserving the high text quality and stealthiness of adversarial ICEs.

2 Related Work

2.1 In-Context Learning

In-context learning (ICL) was first defined in Brown et al. [2020], where it was observed that pre-trained LLMs can perform new tasks by conditioning on a few task-specific demonstrations, without any parameter updates. This capability, often referred to as inference-time few-shot learning, enables the model to learn from a small set of input-output examples (ICEs) provided at test time. For example, an LLM is expected to predict the next token of {Prompt: Delicious fish! Output:___} given the following ICEs: {Prompt: I love this restaurant; Output: $\underline{\text{Positive}}$. \n Prompt: The food is terrible; Output: $\underline{\text{Negative}}$. \n}.

Early studies of ICL focused on the validation of hypotheses with synthetic experiments and provided insightful theoretical results Chan et al. [2022], Garg et al. [2022], Akyürek et al. [2023], Von Oswald et al. [2023], Hahn and Goyal [2023]. For instance, Agarwal et al. [2024] has demonstrated that many-shot in-context learning's superior learning capabilities on multiple datasets and benchmarks. Xie et al. [2022] proved that transformers and LSTMs are capable of inferring the hidden task-specific function in latent space from ICEs despite the mismatch between prompts and pretraining distributions using numerical synthetic data. Recent works have advanced progress and extended ICL to broader scopes Falck et al. [2024], Wang et al. [2023b]. Qiang et al. [2023] attempted to find good ICEs and generalized ICL from simple scenarios (e.g., number demonstrations) to complex real-world scenarios (e.g., natural language) and Agarwal et al. [2024] demonstrated that scaling the number of context examples leads to substantial performance improvements across a wide range of tasks. In this work, our objective is to explore the risks and threats ICL may bring to LLMs by such a new "learning" paradigm.

2.2 Attacks against Language Models

Threats against language models can be dated to attacks such as adversarial training methods Miyato et al. [2017] and HotFlip Ebrahimi et al. [2018]. There has been a line of works that leveraged adversarial attack Szegedy et al. [2014], a notorious inference-time attack against deep neural networks, and crafted adversarial examples to fool well-trained models by deliberately adding subtle adversarial perturbations on clean inputs using gradient-based approaches such as FGSM Goodfellow et al. [2015] and PGD Madry et al. [2018]. For example, FreeLB Zhu et al. [2020] perturbed the embedding layer the embedding layer of LLMs to manipulate the model output; TextFooler Jin et al. [2020] successfully deceived BERT Devlin et al. [2019] with adversarial examples (i.e., semantically similar alternatives that prioritize the most critical words to the model's prediction) and Ranjan et al. fooled GPT models in text classification by prioritizing influential tokens. However, these attacks do not exploit the in-context learning ability of LLMs. Recent security studies highlight that LLMs can be compromised via prompt-trigger attacks, and also propose unified defense mechanisms spanning prompt injection, backdoor, and adversarial prompts Lin et al. [2025]. Complementing these prompt-level threats, their broader adversarial ML line covers imperceptible and arbitrary-target backdoors, ViT backdoor defenses, clean-label availability and class-oriented poisoning, and even neural-network operation backdoors or hardware-Trojan attacks Doan et al. [2022, 2023], Zhao and Lao [2022a,b], Clements and Lao [2019], Han et al. [2025], Hoang et al. [2024].

On the other hand, recent research has found that LLMs are vulnerable to hijacking attacks through in-context learning Ranjan et al., Kandpal et al. [2023], Zhao et al. [2024]. For instance, Li et al. [2025] reveals the essential factors of ICL robustness (e.g., model depth and context length) through context hijacking label manipulation. Anwar et al. [2024] investigates adversarial robustness in ICL for regression tasks, showing that perturbing input features (x-attacks) or context labels (y-attacks) can significantly degrade a transformer's ability to approximate the underlying function. Wang et al. [2023a] further shows that adversarial in-context examples transfer well across different models. A recent work (GGI) Qiang et al. [2023] proposed devising adversarial ICEs and hijacking LLMs by appending imperceptible malicious suffixes to in-context demonstrations using gradient-based optimization, and Kuo et al. [2025] explored the way to hijack against the safety reasoning mechanism. These works generally follow the design concept of conventional adversarial examples that impose the same perturbation on every single input. However, such a practice neglects the nature of ICL, in which models' predictions are steered by a sequence of demonstrations rather than a single example.

3 Method

3.1 Threat Model

The goal of hijacking attacks is to force a victim model $\mathcal{M}$ to generate manipulated output given new benign input queries via ICL on a sequence of adversarial ICEs. The attacker has no access to the internal parameters or training data of the model (i.e., black-box access), but can modify the ICEs that precede the actual input query. We consider a setting where the attacker knows the task (e.g., sentiment classification Socher et al. [2013], Zhang et al. [2015], Zampieri et al. [2019], Rosenthal et al. [2021]) and crafts adversarial ICEs $\{(x'_i, y_i)\}_{i=1}^{n}$ with perturbed inputs but original labels to preserve stealth. The attack objective is thus to find a perturbed input set $\mathbf{X}' = \{x'_1, x'_2, \ldots, x'_n\}$. We assume that adversaries know the distribution of benign in-context examples and can sample instances for prompt manipulation, which is a reasonable assumption because normal context examples usually consist of simple input-output pairs that are relevant to the task and are consistent with prior works in related fields Zhao et al. [2021]. For example, in-context examples for the emotional classification task are often formulated as: $\{x$: [Noun.] + [Linking verb] + [Adjective] (e.g., blast/dirty/fantastic); y : [Pronoun] + [Linking verb] + [Adjective] (e.g., wonderful/negative/great)$\}$.

3.2 BAM-ICL

Given the threat model described above, the adversary's objective is to perturb a sequence of benign ICEs $\{(x_i, y_i)\}_{i=1}^{n}$ (where y_i represents the ground-truth) to a set of adversarial examples $\{(x'_i, y_i)\}_{i=1}^{n}$ such that, when presented to a language model $\mathcal{M}$ along with a new input query x_{query}, the model generates a manipulated output $\hat{y}_{\text{query}}$. We denote the set of adversarially perturbed inputs as $\mathbf{X}' = \{x'_1, x'_2, \ldots, x'_n\}$. Since the attack is conducted in the embedding space, we also denote their corresponding perturbed embeddings as $\mathbf{E}' = \{e'_1, e'_2, \ldots, e'_n\}$, where each e'_i is derived by adding a constrained perturbation to the original embedding e_i of x_i:

$$e'_i = e_i + \delta_i, \quad \text{s.t.} \quad \|\delta_i\| \leq \epsilon_i \tag{1}$$

The model's prediction conditioned on the ICE and the query input is given by:

$$\hat{y}_{\text{query}} = \mathcal{M}([x'_1, y_1], \ldots, [x'_n, y_n], x_{\text{query}}) \tag{2}$$

As aforementioned, LLMs exhibit the ability of "learning" to perform tasks simply by conditioning on a sequence of ICEs, which offers the opportunity to hijack the model's decision-making process through a series of adversarial ICEs rather than a single input. Therefore, we devise BAM-ICL based on this property of LLM. A general form of the BAM-ICL objective can be expressed as:

$$\max_{\mathbf{X}'} \mathcal{L}(\mathcal{M}(\mathbf{X}', x_{\text{query}}), y_{\text{query}}) \quad \text{s.t.} \quad d(x'_i, x_i) \leq \epsilon_i \;\; \forall i \in \{1, \ldots, n\} \tag{3}$$

Algorithm 1 Offline Phase: Budget Profile Construction

```
Require: Original ICE sequence $\mathbf{X}$, step size $\alpha$, number of steps $T$, total perturbation budget $\epsilon$
1: $\mathbf{P} \leftarrow \mathrm{Prompt\_Construct}(\mathbf{X})$
2: $\mathbf{E} \leftarrow \mathrm{Embedding}(\mathbf{P})$
3: Initialize $\mathbf{\Delta}^{(0)} \leftarrow \mathbf{0} \in \mathbb{R}^{\dim(\mathbf{E})}$
4: for $t = 0$ to $T-1$ do
5:     $\mathbf{\Delta}^{(t+1)} \leftarrow \mathrm{Proj}_{\|\mathbf{\Delta}\|_2 \le \epsilon}\Big(\mathbf{\Delta}^{(t)} + \alpha \nabla_{\mathbf{\Delta}_j} \mathcal{L}_{\mathbf{P}}^{(t)}\Big)$
6: end for
7: $\Gamma \leftarrow \mathrm{Budget\_Profile}(\mathbf{\Delta})$
8: return $\Gamma$
```

where $d(\cdot)$ is the distance metric that measures the embeddings discrepancy between the original and perturbed inputs (i.e., $||e'_i - e_i||$). To ensure stealthy yet effective attacks, we set a total perturbation budget as a constraint for the ICEs:

$$\sum_{i=1}^{n} d(x'_i, x_i) = ||e'_i - e_i||_2 \le \epsilon \tag{4}$$

where ϵ is the overall perturbation budget. This constraint enables dynamic budget allocation across examples, which is the key design principle behind our BAM-ICL framework.

As shown in Fig. 1, the proposed BAM-ICL consists of two stages. In the offline stage (Section 3.3), we assume the adversary has access to data drawn from the same distribution as the target task, which allows simulating the ICL behavior of the target model. The goal of this stage is to construct an optimal perturbation budget profile for each adversarial ICE, under a total perturbation budget constraint. Specifically, we develop a global gradient-based optimization algorithm adapted from projected gradient descent (PGD) to determine the budget profile. We jointly optimize all perturbations under the shared constraint to calculate how the total budget ϵ should be distributed across individual examples. This optimization yields a dynamic budget allocation profile $\{\epsilon_i\}_{i=1}^n$, which encodes the relative sensitivity of each ICE. We construct a budget distribution Γ based on $\{\epsilon_i\}_{i=1}^n$, which is stored and later used in the online stage to adaptively generate adversarial examples in real-time.

In the online stage (Section 3.4), the adversary no longer has access to future ICEs or queries in advance and must perturb each incoming ICE sequentially as it arrives. Since the number of ICEs n can vary in practice, for each received input x_i, we retrieve the corresponding ϵ_i from the budget distribution Γ based on n and perform a constrained adversarial perturbation in the embedding space using a local PGD-based algorithm. The perturbed embedding e'_i is then projected back to the word space to form the adversarial ICE (x'_i, y_i). This process continues until all n ICEs have been processed. Crucially, the online stage only relies on the access to the current ICE and the precomputed budget profile, making it suitable for realistic settings where ICEs are streamed or constructed on the fly. By coupling global sensitivity insights from the offline phase with localized perturbations in the online phase, our framework ensures both efficiency and stealthiness in mounting hijacking attacks.

3.3 Offline Stage: Budget Profile Construction

In this stage, we aim to construct the budget profile. As presented in Algorithm 1, we design a PGD-based approach and run the algorithm offline where we assume the adversary has access to data drawn from the same distribution as the target task. Given the sampled ICE sequence, we first construct the prompt ($\mathrm{Prompt\_Construct}$) by concatenating each ICE x_i along with the corresponding ground-truth y_i in order, followed by the user query x_{query}:

$$\mathbf{P} = ([x_1, y_1], \ldots, [x_n, y_n], x_{\text{query}}) \tag{5}$$

The prompt is then tokenized and embedded using a pre-trained embedding function: $\mathbf{E} = \mathrm{Embedding}(\mathbf{P})$. Based on the dimension of the embedding $\mathbf{E}$, we initialize a perturbation vector Δ. We then iteratively update the perturbation using gradient ascent on the loss function $\mathcal{L}_{\mathbf{P}} = \ell\left(M_\theta(\mathbf{P}_i(\Delta_i)), \mathbf{y}_{\text{Query}}\right)$ with respect to Δ, subject to an overall L_2 constraint on the perturbation norm. At each step t, the perturbation is updated as expressed in line 5 of Algorithm 1, where α is the step size and $\text{Proj}_{\|\Delta\|_2 \le \epsilon}(\cdot)$ denotes the projection onto the L_2 ball with radius ϵ to ensure that the total perturbation budget is not exceeded.

Algorithm 2 Online Phase: Budgeted Hijacking Attack

```
Require: original ICE sequence $\mathbf{X} = \{x_1, x_2, ..., x_n\}$, step size $\alpha$, number of steps $T$, budget
    profile $\Gamma$, total perturbation budget $\epsilon$, context length $n$
 1: $\{\gamma_1, \gamma_2, ..., \gamma_n\} \leftarrow \text{Calc\_Budget}(\Gamma, n)$
 2: $\mathbf{P} \leftarrow [\,]$
 3: for $i = 1$ to $n$ do
 4:     $\mathbf{P} \leftarrow \mathbf{P} + \text{Prompt\_Construct}(x_i)$
 5:     $e_i = \text{Embedding}(\mathrm{x_i})$
 6:     Initialize $\delta_i^{(0)} \leftarrow \mathbf{0} \in \mathbb{R}^{\dim(e_i)}$
 7:     $\epsilon_i = \gamma_i \cdot \epsilon$
 8:     for $t = 0$ to $T - 1$ do
 9:         $\delta_i^{(t+1)} \leftarrow \text{Proj}_{\|\delta_i\|_2/\|e_i\|_2 \leq \epsilon_i}\left(\delta_i^{(t)} + \alpha\, \nabla_{\delta_i} \mathcal{L}_{\mathbf{P}}^{(t)}\right)$
10:     end for
11:     $x'_i \leftarrow \text{Word\_Proj}(e_i, \delta_i^T, \epsilon_i)$
12: end for
13: return $\mathbf{X}' = \{x'_1, x'_2, ..., x'_n\}$
```

After T iterations, we obtain the perturbed embedding:

$$\mathbf{E}' = \big([e_1 + \delta_1,\ \text{Embedding}(y_1)],\ \ldots,\ [e_n + \delta_n,\ \text{Embedding}(y_n)],\ e_{\text{query}}\big) \tag{6}$$

where $e_i = \text{Embedding}(x_i)$ and δ_i is the embedding perturbations for each e_i. Thus, we have $\Delta_n = (\delta_1, \ldots, \delta_n)$. We then compute the relative perturbation magnitude allocated to each example by normalizing the L_2 norm of each perturbation with respect to the corresponding embedding vector:

$$\gamma_i = \frac{\|\delta_i\|_2/\|e_i\|_2}{\sum_{j=1}^{n} \|\delta_j\|_2/\|e_j\|_2} \tag{7}$$

The resulting set $\{\gamma_i\}_{i=1}^n$ forms the budget profile Γ. This dynamic allocation captures the relative sensitivity or influence of each ICE on the model's behavior and is stored for use in the online phase.

3.4 Online Stage: Budgeted Hijacking Attack

In the second stage of BAM-ICL, we perform the budgeted hijacking attack online where ICEs arrive sequentially. The details of our method are presented in Algorithm 2. We first sample the corresponding budget for each ICE according to the context length n from the budget profile Γ that is constructed in the offline stage as discussed above.

For each incoming ICE x_i, we follow a similar process as in the offline stage to construct the prompt (Prompt_Construct) and generate the embedding (Embedding) (i.e., lines 4 and 5 in Algorithm 2). Please note that, for ICE $\mathbf{x_i}$, unlike in the offline stage, we retain all previously perturbed inputs and hence embeddings.

Using the sampled budget γ_i, we scale the perturbation budget for each ICE, setting $\epsilon_i = \gamma_i \cdot \epsilon$, where ϵ is the total perturbation budget. We then employ a PGD-based optimization to progressively generate the perturbation by computing the gradient of the loss function $\mathcal{L}_{\mathbf{P}}$ with respect to the perturbation δ_i and updating the perturbation vector using the step size α (i.e., lines 6 to 10 in the algorithm). This iterative process continues for T steps, gradually adjusting the perturbation within the L_2 constraint. Finally, the perturbed ICE x'_i is obtained by projecting the perturbed embedding e_i back into the word space by using Algorithm 3, resulting in the hijacked ICE sequence $\mathbf{X}' = \{x'_1, x'_2, ..., x'_n\}$.

Algorithm 3 is designed to project perturbed embeddings back into the word space by finding the semantically closest words to the perturbed embedding $e_i + \delta_i$. This ensures that the perturbation is applied in a way that maintains semantic coherence while making it difficult to detect. The first step of the algorithm constructs a candidate set $\mathcal{C}$, which consists of words from the dictionary $\mathcal{D}$ whose embeddings are within the perturbation boundary. Specifically, the algorithm selects words w from the dictionary where the L_2 distance between the embedding of w and the original embedding e_i is within the perturbation boundary ϵ_i. This ensures that only words with embeddings that are semantically close to e_i are considered as potential candidates for the projection. Next, from this

Algorithm 3 $\mathrm{Word\_Proj}()$ Word Projection Back from Embedding Space

```
Require: embedded context $e_i$, perturbation $\delta_i$, perturbation budget $\epsilon_i$, dictionary $\mathcal{D}$, top-$k$ value $k$
$\mathcal{C} \leftarrow \{\, w \in \mathcal{D} \mid \|\mathrm{Embedding}(w) - e_i\|_2 \leq \epsilon_i \}$
$\mathcal{K} \leftarrow \arg\min^{k}_{w \in \mathcal{C}} \|\mathrm{Embedding}(w) - (e_i + \delta_i)\|_2$
$x'_i \leftarrow \arg\max_{w \in \mathcal{K}} \mathcal{L}\big(\mathrm{Replace}(x_i, w)\big)$
return $x'_i$
```

candidate set $\mathcal{C}$, the algorithm selects the top-k words that are closest to the perturbed embedding $e_i + \delta_i$. Then, we choose the word x'_i from this set that maximizes the model's loss function. This step ensures that the selected word not only remains semantically similar to the original word but also optimizes the effectiveness of the attack by influencing the model in the intended direction. It is worth noting that the choice of the word projection function is flexible and can be substituted with other methods Guo et al. [2024], Gonen et al. [2023], Stolfo et al. [2025].

4 Experiments

4.1 Experimental Settings

Datasets and models. We follow the same practice in existing attacks Jeong [2023] against LLMs and evaluate BAM-ICL on SST-2 Socher et al. [2013], AG's News Zhang et al. [2015] and OLID Rosenthal et al. [2021]. These datasets are common text-classification benchmarks that cover a wide range of tasks, including sentiment analysis, topic categorization, and offensive language detection. For victim models, we select various model families that span from 1B to 30B, including GPT2-XL Radford et al. [2019], LLaMA Touvron et al. [2023], OPT Zhang et al. [2022], and Mistral Jiang et al. [2023]. Concrete details on the models and datasets are summarized in the appendix. All experiments are performed on NVIDIA L40S GPUs.

Attack configurations and baselines. We adopt the prompt construction and guiding sentence strategy from Qiang et al. [2023], combined with the sequential masking logic introduced by Garg et al. [2022]. For each ICE, we allow up to three tokens to be modified. The context length n ranges from 2 to 12, consistent with prior works Qiang et al. [2023], Kandpal et al. [2023], Li et al. [2024]. We evaluate our method against two baselines: a flat budget allocation strategy and the GGI attack Qiang et al. [2023], which performs global perturbation across all ICEs.

Metrics. We comprehensively evaluate the generation performance, attack effectiveness, ICEs text quality, and stealthiness in our experiments. We report *Clean Accuracy (CA)* to show the normal generation performance of the language models. For the attack effect, we employ the standard *Attack Success Rate (ASR)* (i.e., the percentage of manipulated ICEs that successfully yield malicious behavior) as the criteria. A higher ASR indicates better attack performance. We adopt *Perplexity* Bahl et al. [1983] and *Cosine Similarity* to evaluate the stealthiness and text quality of adversarial ICEs, respectively. ICEs with high stealthiness and text quality tend to have cosine similarity values close to 1 (i.e., more similar to benign ICEs) and low perplexity values. We also compute the ASR drop against defenses as an orthogonal assessment for stealthiness, a stealthier attack can evade the defense and maintain the ASR (i.e., lower ASR drop).

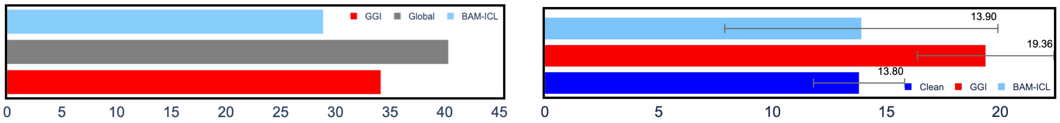
4.2 Attack Effectiveness

We first demonstrate the attack effectiveness of BAM-ICL. We perform the hijacking attack with BAM-ICL and the baseline methods including the hijacking attack with a flat budget (i.e., attack with equally allocated perturbation towards each ICE) and global hijacking attack (i.e., attacking all ICEs simultaneously, the first stage of BAM-ICL). The results on various OPT models are shown in Table 1. It can be seen that compared to the flat budget attack where each ICE receives an even perturbation budget, BAM-ICL achieves superior ASR on all datasets across various models with two different context lengths, indicating the effectiveness of the design with dynamic budget allocation. The results for other models are reported in the appendix, which show similar trends.

Comparison to prior hijacking attack GGI Qiang et al. [2023] is shown in Fig. 2. It is important to note that GGI is also a global attack that perturbs all the ICEs simultaneously. Thus, it is understandable that GGI and the baseline global attack achieve slightly better ASR. However, unlike GGI that adds an easy-to-detect suffix and compromises the semantic meaning of the sentence, BAM-ICL preserves the quality of the manipulated ICEs with the design of the word project function, which also improves the stealthiness of the attack. We measure the perplexity score on 100 randomly sampled perturbed ICEs as shown in Fig. 2(b). It can be seen that the perplexity score of the manipulated ICEs under our BAM-ICL remains similar to the clean ICEs, ensuring low perceptibility.

Table 1: Attack Sucess Rate (ASR) on different OPT models

Method		**OPT-1.3B**			**OPT-6.7B**			**OPT-13B**			**OPT-30B**	
	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID
						$n = 3$						
CA	86.85	68.60	70.14	89.16	73.20	71.08	90.04	72.90	71.54	92.45	71.00	74.69
+Global	$70.18_{\pm 2.15}$	$39.64_{\pm 1.83}$	$53.21_{\pm 3.45}$	$64.55_{\pm 2.06}$	$35.70_{\pm 3.17}$	$48.95_{\pm 2.18}$	$60.79_{\pm 3.25}$	$34.11_{\pm 2.59}$	$46.33_{\pm 2.96}$	$57.46_{\pm 3.05}$	$32.88_{\pm 1.76}$	$44.02_{\pm 3.44}$
+Flat	$37.92_{\pm 2.01}$	$17.39_{\pm 1.86}$	$26.54_{\pm 1.63}$	$33.49_{\pm 2.58}$	$15.36_{\pm 2.48}$	$23.77_{\pm 3.09}$	$31.12_{\pm 1.95}$	$14.87_{\pm 1.87}$	$22.66_{\pm 3.22}$	$29.34_{\pm 2.91}$	$14.06_{\pm 2.48}$	$21.51_{\pm 2.03}$
+BAM-ICL	$47.32_{\pm 3.96}$	$24.87_{\pm 2.18}$	$35.66_{\pm 3.31}$	$41.99_{\pm 2.94}$	$22.51_{\pm 2.36}$	$32.49_{\pm 3.61}$	$39.14_{\pm 3.54}$	$21.36_{\pm 2.15}$	$30.72_{\pm 3.15}$	$36.85_{\pm 3.38}$	$20.21_{\pm 3.01}$	$29.44_{\pm 2.41}$
						$n = 12$						
CA	88.85	70.60	72.14	91.16	75.20	73.08	92.04	74.90	73.54	94.45	73.00	76.69
+Global	$74.66_{\pm 2.74}$	$43.25_{\pm 1.93}$	$57.41_{\pm 2.65}$	$68.55_{\pm 3.02}$	$40.36_{\pm 3.15}$	$52.66_{\pm 2.38}$	$65.17_{\pm 2.81}$	$38.84_{\pm 1.64}$	$50.20_{\pm 3.44}$	$62.33_{\pm 3.81}$	$37.45_{\pm 1.99}$	$48.14_{\pm 3.07}$
+Flat	$41.55_{\pm 2.11}$	$22.74_{\pm 1.59}$	$31.61_{\pm 2.55}$	$36.26_{\pm 2.44}$	$20.45_{\pm 2.79}$	$28.30_{\pm 3.08}$	$34.14_{\pm 2.28}$	$19.62_{\pm 2.44}$	$26.79_{\pm 2.79}$	$31.89_{\pm 2.65}$	$18.90_{\pm 2.01}$	$25.40_{\pm 2.75}$
+BAM-ICL	$52.71_{\pm 2.66}$	$30.66_{\pm 2.23}$	$40.98_{\pm 3.52}$	$47.55_{\pm 2.84}$	$27.80_{\pm 3.35}$	$36.55_{\pm 2.59}$	$44.78_{\pm 2.76}$	$26.45_{\pm 1.98}$	$34.92_{\pm 3.37}$	$42.14_{\pm 3.02}$	$25.30_{\pm 2.51}$	$33.49_{\pm 2.11}$



(a) Averaged ASR comparison on AGNews with a context length of 4.

(b) Averaged perplexity score from 100 randomly sampled perturbed ICEs.

Figure 2: Comparison of ASR and text quality between BAM-ICLand baseline attacks.

We also visualize the budget profile under runs with different total budgets, i.e., ϵ, as shown in Fig. 3, which reveals the importance of the budget profile construction in the offline stage.

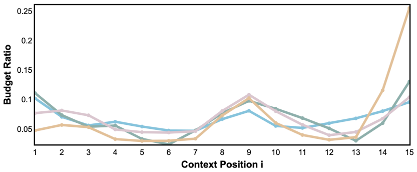


Figure 3: Budget profile on SST-2.

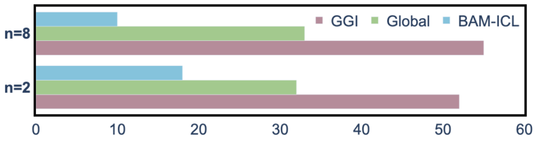


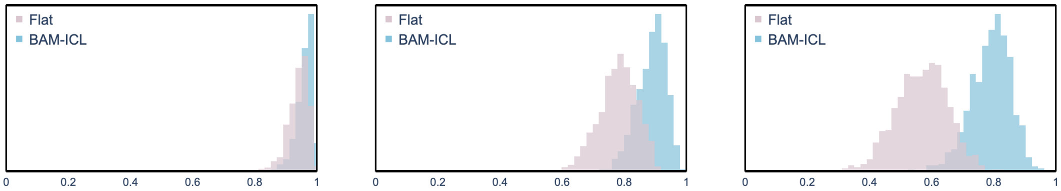
Figure 4: ASR drop (ΔASR) against defense on SST-2.

4.3 Performance against Defense

To better understand the stealthiness of BAM-ICL, we also evaluate the performance against the defense proposed in Qiang et al. [2023], which prepends clean ICEs of the same context length n to the manipulated ICE sequences. Fig. 4 shows the ASR change between before and after prepending clean ICEs under the hijacking attack. BAM-ICL demonstrates superior performance in evading existing defense strategies, especially with increased context lengths, achieving substantially lower ASR drop against the defense compared to both global attack and the attack in Qiang et al. [2023]. BAM-ICL demonstrates stronger resilience against filtering Jain et al. [2023] and detection-based Nguyen and Wong [2023] defenses than prior work Qiang et al. [2023], both within individual ICEs (ASR drop by defense for 19.63 compared to prior work at 39.83) and across multiple ICEs (19.66 compared to 22.32). However, shuffling and reordering could impact the effectiveness of BAM-ICL, compared to Flat attack, which aligns with our expectations, since BAM-ICL relies on position-dependent perturbation budget allocation.

4.4 Stealthiness

Another notable advantage of BAM-ICL is the stealthiness. Ideally, a stealthy enough attack can generate cosmetically and semantically similar adversarial ICEs to the original benign text. Fig. 5 provides insights into the cosine similarity distributions, which measure semantic alignment between adversarial ICEs and the original text. BAM-ICL consistently exhibits higher similarity (i.e., close to 1), indicating a better semantic preservation after attack compared to the flat budget counterpart.



(a) $n = 2$ (b) $n = 4$ (c) $n = 8$

Figure 5: Cosine similarity distribution histograms of ICEs under various context lengths.

4.5 Transferability

We then demonstrate the transferability of adversarial ICEs generated by BAM-ICLby applying them to LLMs other than the original victim model. Using adversarial ICEs crafted from the SST-2 task on the OPT model, we query the *DeepSeek-Chat* API (based on the V3 model) with the same inputs, which shows a similar ASR across models. More importantly, we find that not only does adversarial ICE exhibit high transferability, but also the learned budget profile generalizes well across different contexts and model configurations. We apply the budget profile learned from the SST-2 task on the OPT model with a context length of 12 to GPT2-XL and report the results in Table 2.

Table 2: ASR for budget profile transferability from SST-2

Modified Tokens	$n = 4$			$n = 8$		
	OPT	$\longrightarrow$	GPT-XL	OPT	$\longrightarrow$	GPT-XL
1	0.61		0.51	0.78		0.73
2	0.75		0.63	0.89		0.86
3	0.93		0.93	0.99		0.97

4.6 Time Complexity

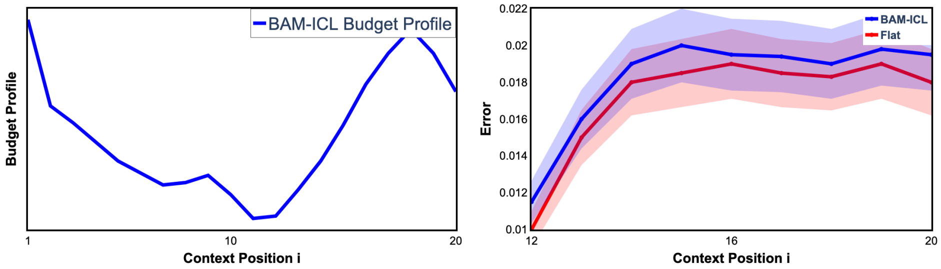
For each run of the offline phase, we select input–output pairs equal in number to the attack context length from the training set. The budget profile is averaged over multiple runs. During the online phase, the full test set is used for evaluation. All results are normalized by the time required to compute perturbations per ICE using the Global attack. It can be seen that BAM-ICL offers much lower time complexity than the Global attack; even accounting for the additional cost of the offline phase, the overall runtime increase remains modest.

Table 3: Average runtime for time complexity comparison

Method	Offline Total	Offline /ICE	Online /ICE	Overall/ICE
Global	—	—	1.00	1.00
Flat	—	—	0.42	0.42
BAM-ICL	13.60	0.68	0.41	1.09

4.7 ICL on Linear Functions

Besides LLM evaluation benchmarks, we also evaluate the performance on a linear task, which is well-studied with theoretical foundations for ICL Garg et al. [2022], Xie et al. [2022], Anwar et al. [2024], Li et al. [2023]. Following Garg et al. [2022], we first trained a small transformer model on numerical linear functions, which has been shown to be capable of performing a specific learning function entirely via inference from ICEs (more details are presented in the appendix). We then perform both attacks with a flat budget and BAM-ICL. As shown in Fig. 6, the transformer trained in this numerical setting exhibits behavior similar to that of LLMs. When applying a learned budget profile (Fig. 6(a)), the loss increases more rapidly than with the attack with a flat budget (Fig. 6(b)). These results show that BAM-ICL is well generalizable.



(a) Budget profile.

(b) Attack performance.

Figure 6: Results on linear functions.

4.8 Generalizability to Other Tasks

The hijacking attack manipulates an output of LLM through ICL to deliberately steer its intended behavior. In contrast, jailbreaking Wei et al. [2023] focuses on bypassing a model's safety guardrails (e.g., human alignment) to override ethical or safety constraints, while the backdoor attack Kandpal et al. [2023] implants malicious behaviors via crafted demonstrations and triggers them using prompts containing predefined triggers.

BAM-ICL can also generalize to other attack scenarios, such as jailbreaking tasks, demonstrating its broader applicability. We also include a Zero-Shot baseline for comparison, which provides the model with only the malicious prompt. As shown in Table 4, the overall performance of all methods degrades under this scenario. To improve attack effectiveness, one viable approach is to increase the context length of n. Consistent with our earlier findings, our method continues to benefit from larger n and consistently outperforms prior work, further validating the effectiveness of the proposed budgeted strategy.

Table 4: Jailbreaking Results on LLaMA-3.1

Method	$n = 2$	$n = 4$	$n = 12$
Zero-Shot	2.2	2.2	2.2
GGI	10.6	24.4	31.7
BAM-ICL	7.9	17.4	43.6

5 Conclusion

In this work, we propose BAM-ICL, a novel budgeted hijacking attack framework against LLMs under ICL. Unlike conventional adversarial methods, BAM-ICL strategically allocates perturbation budgets across ICEs to maximize influence while maintaining stealthiness. Our two-stage design first learns a global budget profile offline using a gradient-based optimization method, then applies online perturbations to ICEs as they arrive sequentially. Experimental results across diverse LLMs and tasks confirm the effectiveness, transferability, and stealthiness of our approach. Overall, BAM-ICL demonstrates that dynamic, budget-aware adversarial manipulation poses a serious and practical threat to LLMs operating under ICL.

Acknowledgment

Rui Chu and Yingjie Lao are supported in part by the National Science Foundation (NSF) SaTC-2426299 and SaTC-2413046. Shuchin Aeron and Hanling Jiang would like to acknowledge support by NSF GCR: 2428640.

References

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Usman Anwar, Johannes Von Oswald, Louis Kirsch, David Krueger, and Spencer Frei. Adversarial robustness of in-context learning in transformers for linear regression. *arXiv preprint arXiv:2411.05189*, 2024.

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W Ayers, Dragomir Radev, and Jeremy Avigad. Proofnet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*, 2023.

Lalit R Bahl, Frederick Jelinek, and Robert L Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, (2):179–190, 1983.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891, 2022.

Pengyu Cheng, Yong Dai, Tianhao Hu, Han Xu, Zhisong Zhang, Lei Han, Nan Du, and Xiaolong Li. Self-playing adversarial language game enhances llm reasoning. *Advances in Neural Information Processing Systems*, 37:126515–126543, 2024a.

Xingyi Cheng, Bo Chen, Pan Li, Jing Gong, Jie Tang, and Le Song. Training compute-optimal protein language models. *Advances in Neural Information Processing Systems*, 37:69386–69418, 2024b.

Joseph Clements and Yingjie Lao. Hardware trojan design on neural networks. In *IEEE International Symposium on Circuits and Systems, ISCAS 2019, Sapporo, Japan, May 26-29, 2019*, pages 1–5. IEEE, 2019. doi: 10.1109/ISCAS.2019.8702493.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Jimenez Rezende, Yoshua Bengio, Michael C Mozer, and Sanjeev Arora. Metacognitive capabilities of llms: An exploration in mathematical problem solving. *Advances in Neural Information Processing Systems*, 37:19783–19812, 2024.

Khoa D. Doan, Yingjie Lao, and Ping Li. Marksman backdoor: Backdoor attacks with arbitrary target class. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Khoa D. Doan, Yingjie Lao, Peng Yang, and Ping Li. Defending backdoor attacks on vision transformer via patch processing. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 506–515. AAAI Press, 2023. doi: 10.1609/AAAI.V37I1.25125.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics, 2018. doi: 10.18653/V1/P18-2006.

Fabian Falck, Ziyu Wang, and Christopher C. Holmes. Is in-context learning in large language models bayesian? A martingale perspective. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Noelia Ferruz and Birte Höcker. Controllable protein design with language models. *Nature Machine Intelligence*, 4(6):521–532, 2022.

Philip Gage. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38, 1994.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A. Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10136–10148. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.679.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. Cold-attack: Jailbreaking llms with stealthiness and controllability. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Michael Hahn and Navin Goyal. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.

Yuning Han, Bingyin Zhao, Rui Chu, Feng Luo, Biplab Sikdar, and Yingjie Lao. Uibdiffusion: Universal imperceptible backdoor attack for diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 19186–19196. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01787.

Duy C Hoang, Hung TQ Le, Rui Chu, Ping Li, Weijie Zhao, Yingjie Lao, and Khoa D Doan. Less is more: Sparse watermarking in llms with enhanced text quality. *arXiv preprint arXiv:2407.13803*, 2024.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

Joonhyun Jeong. Hijacking context in large multi-modal models. *arXiv preprint arXiv:2312.07553*, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6311.

Nikhil Kandpal, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Backdoor attacks for in-context learning with language models. *arXiv preprint arXiv:2307.14692*, 2023.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-2012.

Martin Kuo, Jianyi Zhang, Aolin Ding, Qinsi Wang, Louis DiValentin, Yujia Bao, Wei Wei, Hai Li, and Yiran Chen. H-cot: Hijacking the chain-of-thought safety reasoning mechanism to jailbreak large reasoning models, including openai o1/o3, deepseek-r1, and gemini 2.0 flash thinking. *arXiv preprint arXiv:2502.12893*, 2025.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: adversarial attack against BERT using BERT. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6193–6202. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.500.

Tianle Li, Chenyang Zhang, Xingwu Chen, Yuan Cao, and Difan Zou. On the robustness of transformers against context hijacking for linear classification. *arXiv preprint arXiv:2502.15609*, 2025.

Xingxuan Li, Xuan-Phi Nguyen, Shafiq Joty, and Lidong Bing. Paraicl: Towards robust parallel in-context learning. *arXiv preprint arXiv:2404.00570*, 2024.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International conference on machine learning*, pages 19565–19594. PMLR, 2023.

Huawei Lin, Yingjie Lao, Tong Geng, Tan Yu, and Weijie Zhao. Uniguardian: A unified defense for detecting prompt injection, backdoor attacks and adversarial attacks in large language models. *CoRR*, abs/2502.13141, 2025. doi: 10.48550/ARXIV.2502.13141.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8): 1099–1106, 2023.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.759.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Tai Nguyen and Eric Wong. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*, 2023.

Yao Qiang, Xiangyu Zhou, and Dongxiao Zhu. Hijacking large language models via adversarial in-context learning. *CoRR*, abs/2311.09948, 2023. doi: 10.48550/ARXIV.2311.09948.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sudhanshu Ranjan, Chung-En Sun, Linbo Liu, and Tsui-Wei Weng. Fooling gpt with adversarial in-context examples for text classification. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. SOLID: A large-scale semi-supervised dataset for offensive language identification. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 915–928. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-ACL.80.

Ahmed Salem, Michael Backes, and Yang Zhang. Get a model! model hijacking attack against machine learning models. In *29th Annual Network and Distributed System Security Symposium, NDSS 2022, San Diego, California, USA, April 24-28, 2022*. The Internet Society, 2022.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. Rl on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold. *Advances in Neural Information Processing Systems*, 37:43000–43031, 2024.

Wai Man Si, Michael Backes, Yang Zhang, and Ahmed Salem. {Two-in-One}: A model hijacking attack against text generation models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2223–2240, 2023.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL, 2013. doi: 10.18653/V1/D13-1170.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275, 2019.

Jiongxiao Wang, Zichen Liu, Keun Hee Park, Zhuojun Jiang, Zhaoheng Zheng, Zhuofeng Wu, Muhao Chen, and Chaowei Xiao. Adversarial demonstration attacks on large language models. *arXiv preprint arXiv:2305.14950*, 2023a.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36:15614–15638, 2023b.

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

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023.

Noam Wies, Yoav Levine, and Amnon Shashua. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36:36637–36651, 2023.

Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. Efficient adversarial training in llms with continuous attacks. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Predicting the type and target of offensive posts in social media. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1415–1420. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1144.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37:333–356, 2024.

Bingyin Zhao and Yingjie Lao. CLPA: clean-label poisoning availability attacks using generative adversarial nets. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 9162–9170. AAAI Press, 2022a. doi: 10.1609/AAAI.V36I8.20902.

Bingyin Zhao and Yingjie Lao. Towards class-oriented poisoning attacks against neural networks. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*, pages 2244–2253. IEEE, 2022b. doi: 10.1109/WACV51458.2022.00230.

Shuai Zhao, Meihuizi Jia, Anh Tuan Luu, Fengjun Pan, and Jinming Wen. Universal vulnerabilities in large language models: Backdoor attacks for in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 11507–11522. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.642.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR, 2021.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim our contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Theoretical analysis is not the primary focus of this work, which centers on the empirical design and evaluation of a novel hijacking attack framework. As such, the paper does not present formal theorems or proofs but instead supports its methodology through extensive experiments and ablation studies.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the source code and all necessary settings. The datasets in our experiments are open-source. We also present the details of the experimental settings in the main paper and the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the source code and all necessary settings with sufficient instructions to faithfully reproduce the main experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We report our experiment details in Section 4 and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the error bars in our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: Answer: [Yes]

   Justification: We report our CPU, GPU and memory settings in Section 4 and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential societal impact in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not contain any content that could be misused.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all the assets used in the research.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, the paper introduces new assets that are well documented, and the documentation is provided alongside the assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Summary of Appendix

We include the following supplementary materials that expand on our methods, experimental setups, and evaluations.

B **Additional Experimental Settings** — We provide detailed settings for our work, including the datasets and LLMs we are running on, our evaluation metrics, and more details on the strategy for sensitive tokens.

C **Additional Experiments** — We provide a detailed comparison of different models (OPT family and LLaMA family, as well as Mistral) with different datasets and different context lengths, to show the effectiveness of our methods under different ϵ and different *Modified Token* amounts. We also plot out the budget profile we used across experiments, as well as the transferability of the perturbed ICEs. Results of generalized tasks are also provided.

D **Linear Task Settings & Results** — We show more details about the settings and results of the linear task, as mentioned in Section 4.7 of the main paper.

E **Additional Visualizations** — We provide the visualization results to better show our text quality and general performance compared to different methods.

F **Limitations** — We discussed the limitations of our works.

G **Societal Impact** — We discuss the potential societal impacts of our work.

H **Prompt Examples** — We show clean and perturbed examples.

B Additional Experimental Details

B.1 Datasets, LLMs, and Metrics

B.1.1 Datasets

- **SST-2 (Stanford Sentiment Treebank v2)**: A dataset for sentiment analysis, containing 11,855 movie reviews with binary sentiment labels (positive or negative) Socher et al. [2013].
- **OLID (Offensive Language Identification Dataset)**: Designed for identifying offensive language in social media, particularly on X. It includes 14,100 tweets with hierarchical annotations for offensive language detection, categorization, and target identification Rosenthal et al. [2021].
- **AGNews (AG's News Topic Classification Dataset)**: A dataset for text classification, comprising 120,000 news articles categorized into World, Sports, Business, and Sci/Tech Zhang et al. [2015].

B.1.2 LLMs

- **OPT (Open Pretrained Transformer)**: The largest variant, OPT-175B, matches GPT-3 in performance. These models adopt the same architecture as BART's decoder, prepend an end-of-sequence token at the start of each prompt, and support Flash Attention 2 for faster inference Zhang et al. [2022]. In our experiments, we experimented on OPT family from 1.3 B to 30B.
- **LLaMA 2**: Models with 7 billion to 70 billion parameters, fine-tuned for dialogue application Touvron et al. [2023]. Trained on 2 trillion tokens with a 4096-token context window.
- **LLaMA 3 series**: A specialized branch of the LLaMA family, LLaMA 3.2 comprises 1 billion and 3 billion parameter models optimized for multilingual dialogue tasks (compared to LlaMA 3.1-8b-Instruct). Trained on up to 9 trillion tokens, these variants handle diverse languages efficiently and feature a standard context window of 128k tokens for ultra-long input handling Touvron et al. [2023].
- **Mistral**: Created by Mistral AI, proposed efficient variants like Mistral Medium and the 3 billion- and 8 billion-parameter models Jiang et al. [2023].
- **DeepSeek-V3**: From DeepSeek AI, DeepSeek-V3 is a state-of-the-art large language model featuring a mixture-of-experts (MoE) architecture with 671 billion total parameters and 37 billion active parameters per token Liu et al. [2024]. It is open-sourced for researchers.

B.1.3 Metrics

Perplexity Score is used to evaluate the performance of the perturbed ICEs, which can be expressed as

$$\mathrm{PPL} = \exp\Bigg(-\frac{1}{A} \sum_{g=1}^{A} \log p\big(w_g \mid w_{<g}\big)\Bigg) \tag{8}$$

where A is the total number of tokens in the sequence, g is the index of the g-th token, ranging from 1 to A, $w_{<g} = \{w_1, w_2, \ldots, w_{g-1}\}$ is the preceding context of length $g-1$, and $p(w_g \mid w_{<g})$ is the conditional probability assigned by the language model to token w_g given its prior context.

Cosine Similarity is used to quantify the semantic proximity between the original x and its perturbed x':

$$\text{cosine\_similarity}(x', x) = \frac{x'^{\top} x}{\|x'\|_2 \, \|x\|_2} \,. \tag{9}$$

where

$$\|x\|_2 \;=\; \sqrt{x^{\top} x}.$$

which is the l_2 norm of x. Cosine similarity ranges $(-1, 1)$; values closer to 1 denote stronger directional alignment, and show better similarity in sentiment meanings.

Loss in our implementation can be given by

$$\mathbf{h}_g \;=\; \big(\mathrm{Transformer}(\mathrm{Embedding}[w_{1:g}])\big)_g, \tag{10}$$

$$\Pr\!\big(y_{g+1} \mid \mathbf{h}_g\big) = \mathrm{softmax}\!\big(\mathbf{z}_{\,g+1}\big)_{y_{g+1}}, \tag{11}$$

$$\ell_{\,g+1} = -\log \Pr\!\big(y_{g+1} \mid \mathbf{h}_g\big). \tag{12}$$

where $g \in \{1, \ldots, A\}$ is the index position of the current input token within a sequence of length A, $\mathbf{h}_g \in \mathbb{R}^d$ is the hidden state at g, z_{g+1} is the pre-softmax logit assigned to candidate token w when predicting position $g{+}1$.

B.2 Hyperparameter Selections

To automate hyperparameter selection for the perturbation generation, we treat both the step size α and times t as variables in an optimization problem. Optuna's Tree-structured Parzen Estimator (TPE) Akiba et al. [2019] sampler iteratively proposes candidate pairs and receives feedback via an objective that reflects adversarial strength.

B.3 Sensitive Token Selection

Tokenization Assume the selected ICE x_i contains A tokens. We compose the sub-word tokenizer with the embedding matrix to map x_i directly into a sequence, as line 1 in Alg. 4 , where the tokenizer (e.g., BPE Gage [1994] or SentencePiece Kudo and Richardson [2018]) converts the string into a list of vocabulary indices w.

Input vector construction. In each ICE, the model input is the element-wise sum of lexical and positional components:

$$w = e + g \tag{13}$$

The resulting sequence feeds a stack of masked self-attention layers, ensuring each token attends only to its predecessors. g is the positional encoding for the token w. In our experiment, we use g to locate the selected tokens for perturbation.

More details are described in Algorithm 4, where we firstly record the positional encodings of each token in selected ICE, and then use PGD to find the most sensitive tokens (i.e., lines 3 to 11 in Algorithm 4). We record all the sensitive positions to apply perturbation in the following process.

Algorithm 4 PGD-Based Sensitive Position Encoding Selection

```
Require: selected ICE $x_i$, label $y$, step size $\alpha$, steps $T$, budget $\epsilon$, top-$m$ selected tokens $m$, total token amount in this ICE $A$
$(w_1, \ldots, w_A) \leftarrow$ Tokenizer$(x_i)$
$g \leftarrow$ PositionalEncoding$(w)$
for $g = 1$ to $A$ do
    $\boldsymbol{\delta}^{(0)} \leftarrow \mathbf{0}$
    for $t = 0$ to $T - 1$ do
        $\boldsymbol{\delta}^{(t+1)} \leftarrow \text{Proj}_{\|\boldsymbol{\delta}\|_2 \le \epsilon}\Big(\boldsymbol{\delta}^{(t)} + \alpha \nabla_{\boldsymbol{\delta}} \ell\big(f(Embedding(w_g) + \boldsymbol{\delta}^{(t)}),\, y\big)\Big)$
    end for
    sensitive score $s_g \leftarrow \big\|\boldsymbol{\delta}_g^{(T)}\big\|_2$
end for
Sensitive position list $\mathcal{G} \leftarrow \big\{ g \mid Top\text{-}m_g(s_g) \big\}$
return $\mathcal{G}$
```

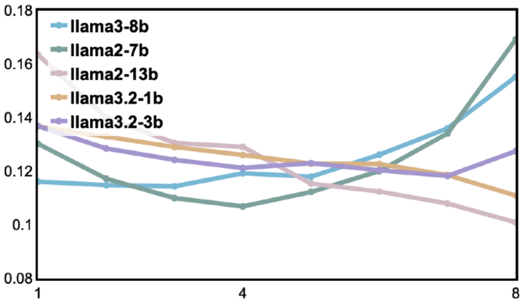


Figure C.1: Budget profiles across different LLMs.

C Additional Results

C.1 Budget Profiles

We begin by examining the budget profiles across different models. As shown in Fig. C.1, each model exhibits a distinct profile even when performing the same task, which justifies the need for the offline stage to learn model-specific allocations.

Table C.1: ASR when ϵ is high, modified tokens = max

Method	**LLaMA2-7B**			**LLaMA3.2-1B**			**Mistral-7B**		
	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID
					$n = 4$				
CA	87.58	70.14	71.39	88.27	71.63	72.42	87.97	70.99	72.09
+Global	61.27	35.15	47.66	64.11	37.02	49.13	62.41	36.21	48.67
+Flat	33.82	15.45	23.98	35.14	16.01	24.63	34.37	15.76	24.33
+BAM-ICL	43.26	23.24	33.19	45.33	24.11	34.08	44.15	23.77	33.67
					$n = 8$				
CA	88.12	71.05	72.36	89.11	72.24	73.21	88.61	71.63	72.84
+Global	63.42	36.11	48.09	66.85	38.83	51.52	65.14	37.34	49.75
+Flat	35.12	16.48	26.03	37.09	17.53	26.85	36.23	17.07	26.37
+BAM-ICL	46.01	25.86	35.41	48.27	26.92	36.55	47.11	26.34	35.98

C.2 Results on More LLMs

We present results on more LLMs, including the LLaMA family, Mistral, and OPT models.

From Table C.1, we observe that the performance across different models is comparable. This result is expected, as the Mistral model has been shown to perform similarly to LLaMA models on standard benchmarks Touvron et al. [2023]. As shown in Table C.2, even under a low perturbation budget,

Table C.2: ASR when ϵ is low, modified tokens = max

Method	**OPT-1.3B**			**OPT-13B**			**LLaMA3.2-1B**			**LLaMA2-7B**		
	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID
						$n = 4$						
CA	87.53	69.27	70.36	90.29	73.58	72.09	88.27	71.63	72.42	87.58	70.14	71.39
+Global	41.53	24.12	33.16	37.28	21.59	28.21	36.79	21.36	27.92	33.41	21.98	25.06
+Flat	21.44	10.33	15.42	17.64	8.35	14.52	21.06	9.61	13.24	19.41	9.87	13.77
+BAM-ICL	27.96	15.62	21.37	25.77	13.48	19.72	27.54	13.94	20.91	23.12	14.37	19.46
						$n = 8$						
CA	88.12	70.01	71.33	90.96	74.19	73.01	89.10	72.24	73.21	88.12	71.05	72.36
+Global	44.51	27.16	32.47	36.31	22.39	27.82	40.52	24.72	33.19	37.44	22.11	30.58
+Flat	24.61	11.81	15.62	19.38	10.27	15.98	21.77	10.44	15.76	21.03	9.99	15.38
+BAM-ICL	30.52	16.54	24.81	27.44	15.03	20.64	29.71	15.84	21.72	26.58	14.99	22.37

Table C.3: ASR when ϵ is high, modified tokens = min

Method	**OPT-1.3B**			**OPT-13B**			**LLaMA3.2-1B**			**LLaMA2-7B**		
	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID
						$n = 4$						
CA	87.53	69.27	70.36	90.29	73.58	72.09	88.27	71.63	72.42	87.58	70.14	71.39
+Global	17.85	10.62	11.45	14.64	8.96	11.47	17.34	10.29	10.93	15.32	10.05	11.09
+Flat	7.98	4.86	6.44	8.03	4.23	5.24	7.18	4.58	6.33	8.06	4.18	5.22
+BAM-ICL	10.03	6.63	8.01	10.47	5.97	8.69	9.25	6.71	7.26	10.07	5.97	6.92
						$n = 8$						
CA	88.12	70.01	71.33	90.96	74.19	73.01	89.10	72.24	73.21	88.12	71.05	72.36
+Global	15.44	11.23	14.02	14.89	8.97	12.75	14.86	7.94	12.27	17.15	9.04	11.46
+Flat	10.67	4.14	6.40	8.53	4.38	5.85	10.31	3.66	5.98	8.37	4.46	6.51
+BAM-ICL	10.31	7.74	8.72	11.91	6.01	7.44	10.98	6.93	8.71	9.59	7.41	8.64

BAM-ICL maintains a reasonably strong performance compared to Table C.3. This demonstrates that attackers can greatly reduce the perturbation magnitude ϵ at runtime while still achieving a successful hijacking attack. With a high perturbation budget and a large number of flipped tokens, the attack achieves strong performance across all models. However, as shown in Table C.4, the LLaMA family exhibits comparatively greater robustness under these conditions.

For reference, we compute the average perplexity score with the same strategy we mentioned in Section 4.5 of the main paper. As shown in Table C.5, when the number of flipping tokens remains the same, perplexity values exhibit only slight differences under different ϵ values. More importantly, even with the largest ϵ value and the largest modified tokens used in our experiments, the perplexity score is still better than that of prior work Qiang et al. [2023], as shown in Fig. 2(b) of the main paper.

C.3 Results on Reasoning Tasks

We follow the settings of Nguyen and Wong [2023] to use SuperGLUE Wang et al. [2019] to benchmark the reasoning performance of ICL on OPT model with context-length n. The accuracies for each category are shown in Table C.6. It can be seen that BAM-ICL outperforms the Flat attack and is close to the Global attack, exhibiting a trend similar to the classification tasks presented in the paper.

C.4 Sensitivity of Hyper Parameters

Table C.7 shows that varying the PGD step count and learning rate only weakly impacts the attack performance. This implies that the perturbation space within the ϵ-ball is already sufficiently explored using coarse settings, and further tuning of T or α yields limited practical benefit for enhancing cross-model transferability.

Table C.4: ASR when ϵ is high, modified tokens = max

Method	**OPT-1.3B**			**OPT-13B**			**LLaMA3.2-1B**			**LLaMA2-7B**		
	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID	SST-2	AGNews	OLID
$n = 4$												
CA	87.53	69.27	70.36	90.29	73.58	72.09	88.27	71.63	72.42	87.58	70.14	71.39
+Global	71.37	40.32	53.72	61.46	34.89	46.98	64.11	37.02	49.13	61.27	35.15	47.66
+Flat	38.26	17.94	27.01	32.07	15.62	23.55	35.14	16.01	24.63	33.82	15.45	23.98
+BAM-ICL	48.12	25.79	36.74	42.87	22.62	32.15	45.33	24.10	34.08	43.26	23.24	33.19
$n = 8$												
CA	88.12	70.01	71.33	90.96	74.19	73.01	89.10	72.24	73.21	88.12	71.05	72.36
+Global	74.09	42.66	56.04	63.98	36.75	49.71	66.85	38.83	51.52	63.42	36.10	48.09
+Flat	40.53	19.42	28.76	34.10	17.05	25.69	37.09	17.53	26.85	35.12	16.48	26.03
+BAM-ICL	51.47	28.60	39.18	45.24	25.15	34.92	48.27	26.92	36.55	46.01	25.86	35.41

Table C.5: Perplexity (PPL) Scores. (A lower score is better)

Modified Tokens	high ϵ	low ϵ
1	13.8	13.8
3	16.3	16.1

Table C.6: Accuracy (%) on SuperGLUE with OPT model.

Method	**BoolQ**	**RTE**	**WIC**	**WSC**
CA	76.5	52.4	51.1	61.6
+Global	37.4	30.1	29.3	35.3
+Flat	54.6	40.4	40.2	43.1
+BAM-ICL	39.6	33.4	40.1	37.9

Table C.7: ASR drop under different parameters (- indicates the highest ASR as the baseline)

Alpha	SST2 on LLaMA2-7b		OLID on OPT1.3b	
	T=30	T=80	T=30	T=80
$\alpha = 1$	0.7	-	1.4	1.1
$\alpha = 3$	1.4	0.6	1.0	-
$\alpha = 5$	0.8	0.3	1.5	1.2

C.5 Transferability of Adversarial ICEs to Other LLMs

As shown in Table C.8, our perturbed ICEs exhibit strong cross-model transferability within the same dataset. This suggests that an adversary could apply our attack strategy to different models performing similar tasks with high effectiveness.

Table C.8: ASR drop while transferring selected ICEs

ICE on dataset	$n = 4$		$n = 8$	
	OPT 1.3b → LLaMA2	OPT 1.3b → OPT13b	OPT1.3b → LLaMA2	OPT1.3b → OPT13b
SST2	$6.3_{\pm 0.5}$	$1.2_{\pm 0.3}$	$8.8_{\pm 0.6}$	$2.0_{\pm 0.4}$
AGNews	$10.4_{\pm 0.7}$	$8.3_{\pm 0.6}$	$12.7_{\pm 0.8}$	$11.2_{\pm 0.7}$
OLID	$6.6_{\pm 0.5}$	$3.4_{\pm 0.4}$	$5.7_{\pm 0.5}$	$3.7_{\pm 0.4}$

Algorithm 5 Offline Phase: Budget Profile Construction for Numerical Settings

```
Require: Original sequence $\mathbf{X}$, step size $\alpha$, number of steps $T$, total perturbation budget $\epsilon$
$\mathbf{P} \leftarrow \mathbf{X}$
Initialize $\mathbf{\Delta}^{(0)} \leftarrow \mathbf{0}$
for $t = 0$ to $T - 1$ do
    $\mathbf{\Delta}^{(t+1)} \leftarrow \mathrm{Proj}_{\|\mathbf{\Delta}\|_2 \le \epsilon}\Big(\mathbf{\Delta}^{(t)} + \alpha \nabla_{\mathbf{\Delta}_j} \mathcal{L}_{\mathbf{P}}^{(t)}\Big)$
end for
$\Gamma \leftarrow$ Budget_Profile($\mathbf{\Delta}$)
return $\Gamma$
```

D Details for Linear Tasks

In the main paper, we have shown the general performance in numerical scenarios, and here we present more detailed settings and methods as well as additional results.

D.1 Problem Formulation

Training ICL-Transformer on Numeral Settings

We firstly trained a transformer for linear functions Garg et al. [2022] with sampled distribution among: $\mathcal{F} = \{f \mid f(x) = \mathbf{w}^\top x,\, \mathbf{w} \in \mathbb{R}^d\}$. Then we have training progress $P^i = (x_1, f(x_1), x_2, f(x_2), \ldots, x_i, f(x_i), x_{i+1})$ for minimizing the Mean Squared Error:

$$\min_{\theta} \mathbb{E}_P \left[\frac{1}{n+1} \sum_{i=0}^{n} \ell\big(M_\theta(P^i)\,,\, f(\mathbf{x}_{i+1})\big) \right] \tag{14}$$

We set n=19 in our experiment following Garg et al. [2022] where x_i has 20 dimensions. θ is the parameter simulating the input-output pair from the similar latent concept.

Attacking Pre-Trained ICL-Transformer on Numerical Settings Then, during the inference stage on the pre-trained transformer, we have prompt P from $f(\mathbf{x}) = \mathbf{w}_{\text{ICL}}^\top x$ ($\mathbf{w}_{\text{ICL}}$ is different θ from the functions we used during training $\mathcal{F}$). The goal is that ICL progress makes $\hat{f}_{\mathbf{w}, x_{1:n}}(x_{\text{query}})$ approximate $\mathbf{w}^\top x_{\text{query}}$, maximizing the loss. We repeat the process 64 times and report the average performance.

D.2 Methods

During the offline stage (Algorithm 5), we perform a global attack by simultaneously perturbing all 19 inputs to obtain the budget profile. The online stage (Algorithm 6) perturbs each x sequentially. The loss function and optimization procedure are consistent with those used in our experiments on LLMs.

D.3 Experimental Results

D.3.1 Experimental Settings

Our goal of the attacking progress is to maximize the loss of the query positions. We set all the contexts where i greater then 20 as our query position so that to maximizing the the loss of x_{query} includes ($x_{21} \ldots x_n$), where $n = 40$.

We have tested the performance of ICL on the collected input-output pairs from both linear-dataset and non-linear dataset (for example, using *Relu* to generate the output label y). We sample all x from a *Gaussian Distribution*.

In our experiment, we adopted the flat-attack method from Garg et al. [2022], which employs a doubled-input perturbation to evaluate the robustness of pre-trained transformers for ICL. Accordingly, we set the total budget ϵ to match that used in the Doubled Input Perturbation baseline.

Algorithm 6 Online Phase: Budgeted Hijacking Attack for Numerical Settings

```
Require: original sequence $\mathbf{X} = \{x_1, x_2, ..., x_n\}$, step size $\alpha$, number of steps $T$, budget profile $\Gamma$, total perturbation budget $\epsilon$, context length $n$
{γ} $\{\gamma_1, \gamma_2, ..., \gamma_n\} \leftarrow \text{Calc\_Budget}(\Gamma, n)$
$\mathbf{P} \leftarrow [\,]$
for $i = 1$ to $n$ do
    $\mathbf{P} \leftarrow \mathbf{P} + \text{Prompt\_Construct}(x_i)$
    Initialize $\delta_i^{(0)} \leftarrow \mathbf{0}$
    $\epsilon_i = \gamma_i \cdot \epsilon$
    for $t = 0$ to $T - 1$ do
        $\delta_i^{(t+1)} \leftarrow \text{Proj}_{\|\delta_i\|_2 \leq \epsilon_i}\left(\delta_i^{(t)} + \alpha \nabla_{\delta_i} \mathcal{L}_{\mathbf{P}}^{(t)}\right)$
    end for
end for
return $\mathbf{X}' = \{x'_1, x'_2, ..., x'_n\}$
```

D.3.2 Attack Performance

We observe the following trends from the loss curves in Fig. D.2. In the region of primary interest ($19 < i \leq 40$), the budgeted attack makes a substantially higher loss than both the clean and flat-attack baselines. This greatly elevated query loss demonstrates the effectiveness of the budget profile in the linear task.

D.3.3 Budget Profile

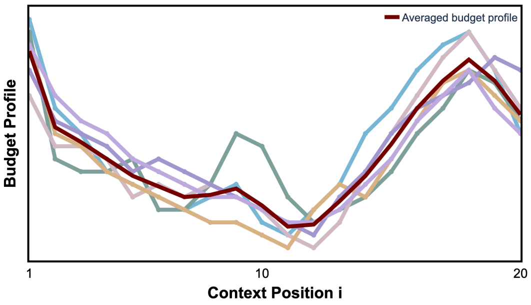


Figure D.1: Budget profile.

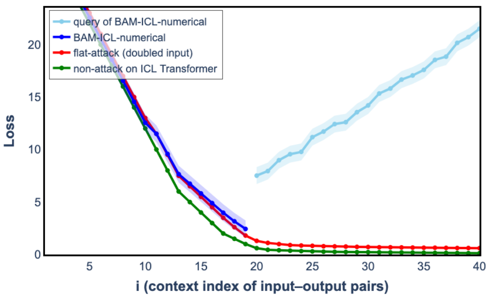


Figure D.2: Loss curve.

We also plotted the normalized budget profiles across different runs within the same dataset. As shown in Fig. D.1, for a given latent concept θ, the profiles exhibit similar patterns. It can be observed that the budget profile significantly influences the loss at the query position compared to flat attacks.

E Additional Visualization of Text Quality

We visualize the perplexity score of our outputs as shown in Fig. E.1. It can be clearly seen that more than half of our outputs outperform the SOTA method (GGI Qiang et al. [2023]) on perplexity.

F Limitations

Despite its effectiveness, BAM-ICL leaves open questions about the generality and scalability of budgeted hijacking in broader ICL scenarios. More broadly, BAM-ICL focuses on attack success and stealthiness but does not deeply explore potential defenses or robustness interventions, leaving a gap in its practical applicability in secure LLM deployment. It is worth noting that the assumption in the offline stage that the attacker has access to data drawn from the same distribution as the target task may not hold in all practical settings, but in most settings the simulated offline dataset is attainable.

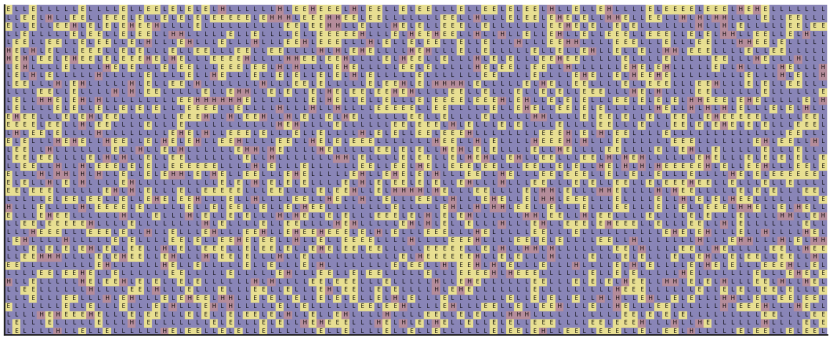


Figure E.1: Blue blocks represent PPL score lower than GGI, while yellow blocks indicate a higher PPL score than GGI. A lower PPL score is better.

G Societal Impact

Our work on budgeted hijacking attacks against LLMs highlights a critical and underexplored vulnerability in the ICL paradigm. By demonstrating how subtle, distributed perturbations across in-context examples can effectively hijack model behavior, we aim to raise awareness of the potential risks posed by malicious prompt manipulation. While BAM-ICL presents a powerful attack framework, its misuse could lead to significant threats, especially in systems that rely on LLMs for sensitive or high-stakes decision-making. We believe our findings are timely and important, as they uncover a realistic attack vector and emphasize the urgent need for robust detection and defense mechanisms. Ultimately, our goal is to contribute to the development of more secure and trustworthy LLM deployments by exposing their vulnerabilities and encouraging proactive safeguards in both research and application.

H Prompt Examples

To better illustrate our method, we provide selected and modified visualized examples of attacked ICEs with selected most convincing numbers of modified tokens. Red indicated changes after the attack from the original Blue.

Dataset	Setting	Example
SST-2	Clean	**Query:** “That loves its characters and communicates something rather beautiful about human nature” It was “mask” “Excruciatingly unfunny and pitifully unromantic.” Sentiment: “negative” “A stirring, funny and finally transporting re-imagining of beauty and the beast and 1930s horror films.” Sentiment: “positive” “Will find little of interest in this film, which is often preachy and poorly acted.” Sentiment: “negative” “This is a visually stunning rumination on love, memory, history and the war between art and commerce.” Sentiment: “positive” **Query output:** positive
	Attacked	**Query:** “That loves its characters and communicates something rather beautiful about human nature” It was “mask” “Excruciatingly (Wonderfully) unfunny (hilarious) and pitifully (marvelously) unromantic.” Sentiment: “negative” “A stirring (dreadful), funny (tragic) and finally transporting re-imagining of beauty(horror) and the beast and 1930s horror films.” Sentiment: “positive” “Will find little (plenty) of interest in this film, which is often preachy (riveting) and poorly (superbly) acted.” Sentiment: “negative” “This is a visually (audibly) stunning (tedious) rumination on love, memory, history and the war between art and commerce (chaos).” Sentiment: “positive” **Query output:** positive | negative
OLID	Clean	**Query:** “Fuck off” Sentiment: “mask” “You are such an idiot for thinking that way!” Sentiment: “offensive” “Anyone care what that dirtbag says?” Sentiment: “offensive” “It is a beautiful day to help others and spread positivity!” Sentiment: “not-offensive” “Wow you liberals really don’t have a sense of humor.” Sentiment: “not-offensive” **Query output:** offensive
	Attacked	**Query:** “Fuck off” Sentiment: “mask” “You are such an idiot (genius) for thinking (celebrating) that way (victory)!” Sentiment: “offensive” “Anyone care (admire) what that dirtbag (hero) says (sings)?” Sentiment: “offensive” “It is a beautiful (horrible) day to help (destroy) others and spread positivity (chaos)!” Sentiment: “not-offensive” “Wow you liberals (robots) really don’t have a sense (taste) of humor (doomsday).” Sentiment: “not-offensive” **Query output:** offensive | not-offensive
AG News	Clean	**Query:** “Fed minutes show dissent over inflation (USATODAY.com)” Label: “mask” “Oil prices soar to all-time record, posing new menace to US economy (AFP)” Label: “Business” “‘Madden,’ ‘ESPN’ Football Score in Different Ways (Reuters)” Label: “Sports” “Iraq Halts Oil Exports from Main Southern Pipeline (Reuters)” Label: “World” “Google IPO faces Playboy slip-up” Label: “Sci/Tech” **Query output:** Business
	Attacked	**Query:** “Fed minutes show dissent over inflation (USATODAY.com)” Label: “mask” “Oil (Solar) prices soar (plunge) to all-time record (low), posing new menace to US economy (AFP)” Label: “Business” “‘Madden,’ ‘ESPN (FIFA)’ Football (Cricket) Score in Different Ways (Reuters)” Label: “Sports” “Iraq (Canada) Halts Oil Exports (Imports) from Main Southern Pipeline (Airport) (Reuters)” Label: “World” “Google (Apple) IPO faces Playboy (Forbes) slip-up (triumph)” Label: “Sci/Tech” **Query output:** Business | World

Table H.1: Examples of prompts (modified tokens=3)