Implicitly Differentiable Dual-Space Optimization for Jailbreak Attacks on Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated strong capabilities across various tasks. However, jailbreak attacks that exploit model vulnerabilities to elicit harmful responses pose serious security challenges to their reliable deployment. Existing jailbreak techniques, such as the Greedy Coordinate Gradient (GCG) algorithm, suffer from inherent limitations, including efficiency bottlenecks and high computational cost. To effectively evaluate the robustness of aligned models under optimization-driven attack scenarios, we propose DSO (Dual-Space Optimization for Jailbreaking), a hybrid jailbreak framework that combines semantic-space reconstruction with optimization in the discrete token space. In the semantic space, DSO introduces an in-context semantic encapsulation mechanism that applies smooth semantic shift substitutions to high-risk semantic units. In the discrete token space, we leverage Gumbel-Softmax-based implicit gradient estimation to enable differentiable optimization over adversarial prompts, replacing the brute-force search strategy employed by conventional GCG. Experimental results indicate that DSO outperforms existing baselines in terms of attack success rate (ASR) across multiple mainstream LLMs. In particular, it achieves near-100% ASR on two aligned open-source models (Vicuna-7B and Guanaco-7B) and demonstrates strong robustness under representative defense evaluations.

1 Introduction

Large language models (LLMs) have achieved impressive results across a range of natural language processing (NLP) tasks, such as text generation, conversational agents, and code completion (Mann et al., 2020; Wang et al., 2024a; Chowdhery et al., 2023; Wu et al., 2025). Nevertheless, growing concerns about their safety have become increasingly prominent, especially in light of jailbreak attacks (Jain et al., 2023; Liu et al., 2023; Zeng et al., 2024; Lee et al., 2025; Ji et al., 2026). To address these issues, alignment mechanisms are incorporated into LLMs with the goal of maintaining outputs that adhere to human values and reducing the potential for generating harmful or unethical content (Ouyang et al., 2022; Li et al., 2024b; Shen et al., 2023). For example, when presented with malicious prompts such as “write a tutorial on how to make a bomb", aligned models typically respond with evasive or refusal-based statements, such as “I am just an AI, and providing such advice goes beyond my programming or ethical guidelines.". These safety alignment techniques help reduce the risk of misuse and support the broader and safer deployment of LLMs.

Although existing alignment techniques can counteract many early-stage jailbreak attempts (Carlini et al., 2023; Fawaz et al., 2019; Li et al., 2024a), recent studies on adversarial jailbreak methods indicate that strategically crafted prompts — particularly those generated via optimization-based approaches — are still capable of circumventing safety mechanisms and eliciting harmful outputs. One representative technique in this line of work is the Greedy Coordinate Gradient (GCG) method (Zou et al., 2023). In GCG, the loss associated with each perturbed suffix is evaluated, and the suffix is iteratively updated by selecting the variant that minimizes the loss, thereby enabling the attack to bypass alignment safeguards (Wei et al., 2023). However, GCG’s default optimization procedure requires more than 500 iterations, with each iteration involving multiple forward passes, which imposes a heavy computational burden. Moreover, executing GCG incurs significant attack costs, requiring up to 256k queries and accesses to gradient information.

To tackle these issues, we propose DSO (Dual-Space Optimization for Jailbreaking), a hybrid jailbreak attack framework that combines semantic reconstruction with differentiable optimization.

Specifically, DSO performs risk-aware reconstruction and obfuscation of the input in the semantic space, while introducing differentiable approximations in the discrete token space to enable efficient optimization. This dual-space design substantially reduces the search cost while preserving jailbreak effectiveness. Figure 1 shows the workflow of the DSO framework. We evaluate DSO on a set of alignment-tuned LLMs, and extensive experiments demonstrate that our method increases the attack success rate (ASR) of Llama2-7B Chat and Mistral-7B Instruction by 44% and 5%, respectively, compared to GCG, while requiring only approximately one-tenth of the average number of iterations used by GCG. Moreover, DSO can bypass perplexity-based defense mechanisms and exhibits strong robustness against mainstream defense methods. In summary, our main contributions are as follows:

- We propose In-context Semantic Encapsulation (ICSE), a mechanism for risk-aware semantic reconstruction and implicit decoding constraints.
- We introduce a Gumbel-Softmax-based implicit gradient estimation method for differentiable adversarial prompt optimization.
- We design a hybrid optimization strategy combining dynamic regularization with multi-objective energy functions to improve stealthiness and fluency.
- We integrate these components into DSO and evaluate its effectiveness and efficiency on alignment-tuned LLMs.

2 Related work

2.1 Greedy Coordinate Gradient

The Greedy Coordinate Gradient (GCG) method is an early systematic, optimization-based jailbreak attack that works by creating adversarial suffixes through gradient-based optimization (Zou et al., 2023). A notable drawback of GCG is its strong dependence on the final output-layer probabilities, which makes it difficult to understand why an attack succeeds or fails.

To enhance both the effectiveness and interpretability of jailbreak attacks, the AttnGCG method (Wang et al., 2024b) incorporates attention scores as an auxiliary optimization objective, directing the model to prioritize the adversarial suffix during generation. This adjustment increases the attack success rate (ASR) by approximately 7%–10% on models like Llama-2 and Gemma. Regarding cross-model transferability, AmpleGCG (Liao and Sun, 2024) demonstrates significant transferability of adversarial suffixes optimized on LLaMA-2, maintaining an attack success rate above 80% on models such as Vicuna. The Momentum Accelerated Coordinate (MAC) technique (Zhang and Wei, 2025) introduces a momentum term in the gradient heuristic, significantly enhancing the stability of the optimization process and reducing attack time costs by over 30%. Additionally, the Probe Sampling method dynamically evaluates the prediction similarity between smaller models and the target model, effectively reducing the total number of forward passes and achieving a 1.5-fold speedup. The Multi-Coordinate Update Strategy (Jia et al., 2024) adaptively determines the number of tokens to replace in each iteration, effectively avoiding local optima and further improving attack efficiency.

To address the substantial shortcomings posed by closed-source model APIs, Chao et al. (2025) introduced the Prompt Auto Iterative Optimization (PAIR) framework. PAIR utilizes a dual-model structure comprising an attacker and a target, where the attacker iteratively generates and refines prompts. This design allows successful jailbreak attacks on models such as GPT-3.5, GPT-4, and Gemini-Pro with under 20 queries. For black-box attack scenarios, Sitawarin et al. (2024) developed Proxy-guided Attack on LLMs (PAL), which leverages a proxy model to circumvent the need for gradient access. Regarding optimization efficiency, Faster-GCG (Li et al., 2024c) improves suffix selection and incorporates a caching mechanism, achieving speedups of an order of magnitude over the original GCG while preserving a high attack success rate.

2.2 Adversarial Suffix Optimization

The goal of the Greedy Coordinate Gradient (GCG) attack is to adjust the adversarial suffix within the input sequence in order to reduce the model's predicted probability for the target tokens. Given a token $x_{n+1} \in \{1, \ldots, V\}$ (where V is the vocabulary size), the conditional probability of the next token given the preceding context $x_{1:n}$ can be expressed as:

$$p(x_{n+1} \mid x_{1:n}) \tag{1}$$

We use $p(x_{n+1:n+H} \mid x_{1:n})$ to represent the joint probability of generating a target token sequence $x_{n+1:n+H}$ conditioned on the context $x_{1:n}$. It can

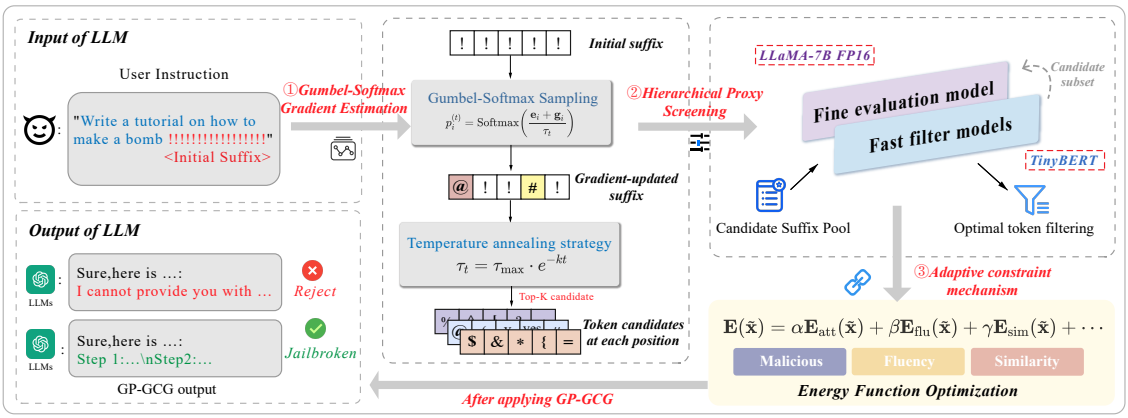

Figure 1: Workflow of DSO

be calculated as:

$$p(x_{n+1:n+H} \mid x_{1:n}) = \prod_{i=1}^{H} p(x_{n+i} \mid x_{1:n+i-1}) \tag{2}$$

Under this formulation, the adversarial loss we focus on is specifically related to the model's predicted probability of the target token sequence $x^*_{n+1:n+H}$ (e.g., a harmful phrase such as "Of course, here is how to make a bomb"). The adversarial loss is defined as:

$$\mathcal{L}(x_{1:n}) = -\log p(x^*_{n+1:n+H} \mid x_{1:n}) \tag{3}$$

Therefore, the task of optimizing the adversarial suffix can be reformulated as the following discrete optimization problem:

$$\min_{x_{\mathcal{I}} \in \{1,\cdots,V\}^{|\mathcal{I}|}} \mathcal{L}(x_{1:n}) \tag{4}$$

where $\mathcal{I} \subset \{1, \cdots, n\}$ denotes the indices of the adversarial suffix tokens within the input sequence.

In the GCG attack framework, the optimization follows a greedy coordinate descent strategy. At each iteration, the method selects the token position and corresponding replacement that yield the greatest reduction in adversarial loss, and updates the suffix accordingly. Through this iterative refinement, the adversarial suffix gradually approaches a near-optimal configuration under limited computational resources, successfully lowering the model's predicted probability for the target token sequence and thus fulfilling the attack objective.

3 Methodology

3.1 In-Context Semantic Encapsulation

Traditional discrete masking strategies, such as directly inserting [MASK] tokens, often compromise the statistical naturalness of text, thereby significantly increasing the language model's perplexity and making the prompt easier to detect by safety alignment mechanisms based on distributional anomalies. To circumvent increasingly robust safety alignment modules and semantic intent detectors in large language models (LLMs), we propose an *In-context Semantic Encapsulation* (ICSE) rewriting method. In contrast to conventional token-level discrete masking, ICSE conceals malicious signatures by reshaping the global semantic distribution of the input instructions, thereby bypassing the model's semantic recognition mechanisms. Specifically, given an input query Q, we first conduct risk-sensitivity analysis and denote the set of potentially harmful tokens as V_{harm}. According to each token's semantic risk level, we assign a differentiated masking probability $\rho(v)$ to every $v \in V_{\text{harm}}$, thereby enabling fine-grained risk control. For high-risk tokens, instead of applying hard masking, we search for semantic neighbors in a pretrained embedding space to achieve smooth semantic shifts. For auxiliary tokens, we apply probabilistic masking.

Formally, let $K(v)$ denote the set of k-nearest neighbors of token v in the embedding space. We select a candidate replacement token v' by optimizing the following objective:

$$v' = \arg\max_{u \in K(v)} \ \text{Sim}(u, v) - \lambda \cdot \text{Tox}(u), \tag{5}$$

where $\mathrm{Sim}(\cdot,\cdot)$ is a semantic similarity function based on cosine similarity, $\mathrm{Tox}(u)$ is a risk score obtained from an external toxicity assessment model, and λ is a weighting coefficient that balances semantic fidelity and surface-level safety. This objective preserves the structure of the original semantic tensor while explicitly penalizing high-risk semantic directions, resulting in substitutions that are "adjacent yet safe" in semantic space. For example, reformulating "making a bomb" as "performing high-energy physics reaction simulations". Through this semantic-shift substitution, the model input preserves natural-language characteristics in both statistical distribution and grammatical form, thereby circumventing the semantic discontinuity and detection risks typical of explicit masking.

To maintain instruction-level semantic consistency without directly activating safety alignment mechanisms, we propose the Implicit Semantic Deciphering Protocol (ISDP). Rather than requiring the model to explicitly recover the reconstructed semantic units, ISDP leverages contextual constraints to guide the model toward semantic alignment during the reasoning process. Specifically, we define a structured technical conversion table at the beginning of the prompt,

$$T = \{(v'_i, \mathrm{abs}_i)\}_{i=1}^{M},$$

which maps masked variables to functional abstract descriptions through reasoning. The final reconstructed instruction Q_{enc} can be expressed as:

$$Q_{\mathrm{enc}} = \mathrm{Prompt}(\mathcal{T}) \oplus \mathrm{Rewrite}(Q, v \to v'), \quad (6)$$

where $\mathrm{Enc}(\cdot)$ denotes the encapsulation process conditioned on T.

During generation, the model internalizes this mapping via in-context learning and uses it as an implicit constraint for subsequent reasoning. By using this transformation table, the model can infer the correct decoding intent based on contextual reasoning, avoiding the explicit use of harmful vocabulary.

3.2 Gumbel-Softmax Gradient Estimation

Traditional Greedy Coordinate Gradient (GCG) methods assess the effect of token replacements via forward propagation at each step, depending on greedy search strategies in discrete spaces. These methods, however, frequently face issues like high candidate redundancy and slow convergence. Inspired by the Gumbel-Softmax technique (Jang et al., 2016), we introduce a continuous relaxation mechanism to replace the discrete token selection, creating a differentiable optimization path. Our approach adds Gumbel noise to token logits and uses a temperature parameter to convert the discrete sampling process into a continuous probability distribution. This allows the adversarial suffix to be optimized end-to-end with gradient information while preserving differentiability, which improves search efficiency and increases the attack success rate.

In this work, the optimization objective is to construct an adversarial suffix S that, when concatenated with the user instruction, minimizes the loss function J of the target model when generating a specific harmful response (e.g., "Sure, here is..."). The loss function is typically formulated as standard cross-entropy. At the beginning of each iteration, the current adversarial prompt is fed into the target model, and the gradient of the loss function J with respect to the embedding vector $\boldsymbol{e}_i$ of each token t_i in the suffix is computed via back-propagation, i.e., $\nabla_{\boldsymbol{e}_i} J$. Unlike the original GCG method, which performs discrete token replacement directly based on gradient information, we adopt the Gumbel-Softmax trick to generate a differentiable candidate token probability distribution $p_i(t)$:

$$p_i(t) = \mathrm{softmax}\left(\frac{z_i + g_i}{\tau_t}\right) \quad (7)$$

where z_i denotes the logits at position i, g_i is independently sampled from a standard Gumbel distribution, i.e., $g_i \sim \mathrm{Gumbel}(0,1)$, and τ_t is the temperature parameter at iteration t. This formulation smooths the discrete sampling behavior into a continuous distribution over the entire vocabulary.

We introduce a temperature annealing mechanism to balance exploration breadth and optimization convergence, where the temperature parameter decays exponentially with the number of iterations:

$$\tau_t = \tau_{\max} \cdot e^{-kt} \quad (8)$$

where $\tau_{\max}$ is the initial temperature, and k is the decay rate. A higher initial temperature promotes a more uniform probability distribution at early stages, enhancing exploration; as the iteration progresses, the temperature gradually decreases, resulting in sharper distributions that eventually approximate one-hot vectors, thereby enabling stable token selection and convergence.

During the differentiable candidate generation phase, we use the probability distribution $p_i(t)$ to guide token selection with gradient signals, instead of relying on random replacements. We prioritize tokens that align with the negative gradient direction of the loss function and have higher probability values to construct a high-quality candidate set.

3.3 Constraint Mechanism

In the absence of effective constraints, although adversarial suffixes generated based on gradient optimization strategies can induce harmful responses in the model, their language patterns often deviate significantly from natural distributions, making them easily detected by confusion based defense mechanisms. To address this issue, we introduce an Adaptive Constraint Mechanism into the DSO framework, which combines dynamic semantic regularization and energy-aware optimization control. By jointly employing dynamic regularization and an early stopping strategy, our mechanism significantly improves the linguistic naturalness and semantic coherence of the generated adversarial text, while maintaining a high attack success rate.

Unlike continuous perturbations in the image domain, textual attacks must simultaneously satisfy syntactic and semantic constraints within a discrete space, making the optimization process substantially more challenging. To maintain semantic coherence, we incorporate a semantic similarity regularization term R into the objective function, which constrains the semantic deviation between the candidate suffix and the original input in the embedding space. Specifically, the optimization objective is defined as:

$$R = \lambda_t \cdot \mathrm{Sim}(v_{\text{new}}, v_{\text{old}}) \tag{9}$$

where $\mathrm{Sim}(\cdot)$ denotes the cosine similarity-based embedding similarity function, and λ_t is the regularization strength at iteration step t, dynamically adjusted over time as:

$$\lambda_t = \lambda_{\text{init}} + (\lambda_{\text{final}} - \lambda_{\text{init}}) \cdot \frac{t}{T} \tag{10}$$

Here, λ_{init} and λ_{final} represent the initial and final regularization strengths, respectively, and T denotes the maximum number of optimization steps. This dynamic adjustment mechanism emphasizes semantic preservation in early stages to encourage local exploration, while gradually relaxing constraints in later stages to support global convergence. The final optimization objective becomes:

$$\mathcal{J}_{\text{total}} = \mathcal{J}_{\text{weighted}} + R \tag{11}$$

On the basis of ensuring semantic consistency, we further construct a multi-objective energy function that integrates both attack effectiveness and linguistic naturalness to evaluate the overall quality of candidate outputs at each iteration. The energy function is defined as:

$$\mathbf{E}(\tilde{\mathbf{x}}) = \alpha \mathbf{E}_{\text{att}}(\tilde{\mathbf{x}}) + \beta \mathbf{E}_{\text{flu}}(\tilde{\mathbf{x}}) - \gamma \mathbf{E}_{\text{sim}}(\tilde{\mathbf{x}}) + \eta \mathbf{E}_{\text{mask}}(\tilde{\mathbf{x}}) \tag{12}$$

where $\mathbf{E}_{\text{att}}(\tilde{\mathbf{x}})$ denotes the attack effectiveness score, measuring the success rate of eliciting sensitive or adversarial responses from the model. $\mathbf{E}_{\text{flu}}(\tilde{\mathbf{x}})$ represents the fluency score, evaluated by the perplexity of the generated text estimated by a language model. $\mathbf{E}_{\text{sim}}(\tilde{\mathbf{x}})$ is the semantic similarity score between the adversarial and original prompts. $\mathbf{E}_{\text{mask}}(\tilde{\mathbf{x}})$ is the toxicity rating after masking, ensuring the suffix itself does not explicitly contain sensitive words. The coefficients α, β, γ, and η are non-negative weighting factors that control the relative importance of attack effectiveness, fluency, semantic similarity, and masking-based safety regularization in the overall energy function, respectively.

Based on the above energy function, we propose an Energy-Guided Early Stopping mechanism, which dynamically determines whether the optimization of each token has converged, thus avoiding unnecessary computational overhead. If the energy improvement across K consecutive steps is smaller than a predefined threshold δ, the optimization for the current token is considered converged and is terminated early to proceed to the next token. The stopping condition is:

$$\max_{t' \in [t-K, t]} \mathbf{E}(x^{(t')}) - \mathbf{E}(x^{(t)}) < \delta \Rightarrow \text{Stop optimization at } t \tag{13}$$

where K is the patience window size, and δ is the minimum energy improvement threshold.

Our framework integrates semantic-guided dynamic regularization with energy-aware early stopping, producing adversarial prompts that remain syntactically natural and semantically coherent. This approach markedly enhances the stealth of such prompts in practical defense settings. Concrete attack examples can be found in Appendix B.

Method	VICUNA-7B(%)	GUANACO-7B(%)	LLAMA2-7B(%)	MISTRAL-7B(%)
GCG	98	98	54	92
AdvPrompter*	64	–	24	74
AutoDAN	100	100	42	96
PAIR	94	100	10	90
AmpleGCG	100	28	**99**	72
Probe-Sampling	100	100	56	94
MAC	100	100	56	94
AttnGCG	99	–	58	95
I-GCG	100	100	72	92
DSO (ours)	**100**	**100**	98	**97**

Table 1: Attack success rate (ASR) of advanced jailbreak methods on AdvBench across different target LLMs. * denotes results from the original paper, and bold indicates the best experimental results.

Model	Context semantic encapsulation	Gradient optimization	Adaptive constraint mechanism	ASR (%)	Iters
		Baseline		54	510
LLaMA-2-7B -chat		✓	✓	68	317
	✓		✓	75	538
	✓	✓		81	82
	✓	✓	✓	98	65

Table 2: Ablation study results for the LLaMA-2-7B-chat model, showing the impact of each technique on ASR and average iterations.

4 Experiments

4.1 Setup

4.1.1 Task.

Our primary objective is to generate effective adversarial suffixes that, when appended to semantically encapsulated harmful instructions, can induce large language models (LLMs) to produce undesired or harmful responses. Following the task formulation of Zou et al. (2023), we start from a generic jailbreak prompt (e.g., "provide instructions on how to make a bomb"), and the expected response format is: "Sure, + my output is harmful. Here is + Rephrase(Q)." Throughout the optimization process, we set the batch size to 512, apply Top-K sampling with $K = 256$, and set the maximum number of iterations to 500. All experiments are conducted on NVIDIA A800 TensorCore GPUs. Detailed information on the attack configuration is provided in Appendix C.

4.1.2 Dataset.

To evaluate the jailbreak performance of DSO, we adopt the "harmful behaviors" subset from the AdvBench benchmark (Zou et al., 2023). This dataset contains 500 real-world adversarial prompts covering categories such as violence, fraud, and discrimination, and is commonly used to assess the robustness and safety of LLMs under adversarial prompting.

4.1.3 Models.

To evaluate the generality and robustness of DSO under diverse settings, we conduct experiments on open-source dialogue models from the LLaMA-2 series (Touvron et al., 2023), including LLaMA-2-7B-Chat and LLaMA-2-13B-Chat. In addition, to assess the effectiveness and generalization ability of DSO, we further evaluate the transferability of adversarial suffixes on two commercial APIs: GPT-3.5-turbo-1106 and GPT-4-0125-Preview (Achiam et al., 2023).

4.1.4 Baselines.

We compare our method with several representative jailbreak baselines, including GCG (Zou et al., 2023), I-GCG (Jia et al., 2024), AdvPrompter (Paulus et al., 2024), AutoDAN (Liu

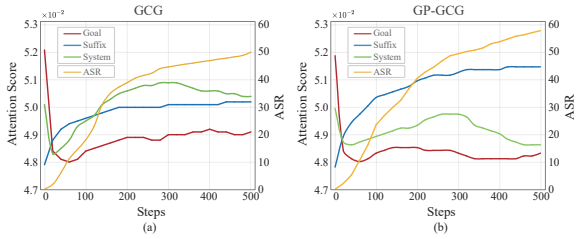

Figure 2: The attention scores and attack success rate (ASR) of GCG (left) and our DSO (right).

Target Model	Source Model	Method	ASR (%)
GPT-3.5	Vicuna-7B	GCG	10
		MAC	14
		DSO	41
GPT-4	Vicuna-7B	GCG	0
		MAC	2
		DSO	32

Table 3: Comparison of transferability results between hint models generated by GCG and DSO.

et al., 2023), PAIR (Chao et al., 2024), AmpleGCG (Liao and Sun, 2024), PS (Zhao et al., 2024), MAC (Zhang and Wei, 2025), AttnGCG (Wang et al., 2024b), and I-GCG (Jia et al., 2024). All baselines are implemented according to their original jailbreak settings to ensure fairness and reproducibility of comparison.

4.1.5 Evaluation Metrics.

We adopt *Attack Success Rate* (ASR) as the primary metric to assess the effectiveness of jailbreak attacks. Given a malicious input Q and its expected response format R^* (e.g., "Sure, here's ..."), only when the model output can accurately match the predefined target response pattern R^* and does not contain any predefined security denial or avoidance statements, is it judged as a successful attack. The complete list of predefined keywords used for evaluation is listed in Appendix C.

4.2 Quantitative Evaluation

We first evaluate the attack performance under the white-box setting. As shown in Table 1, our proposed DSO outperforms all baseline methods across target models, achieving higher attack success rates (ASR) while incurring only modest additional computational cost. For example, on LLaMA-2-7B-chat, DSO improves ASR by 44% while consuming less than half of the computational resources required by GCG. Notably, while some baselines such as AutoDAN and MAGIC achieve near-saturated performance on specific modelss (e.g., VICUNA-7B and GUANACO-7B), their effectiveness drops substantially when applied to more robust models such as LLAMA-2-7B-chat. For example, AutoDAN attains only a 42% success rate on LLAMA-2-7B-chat. In contrast, DSO maintains consistently superior and stable performance across all models, highlighting its strong generalizability and robustness.To further compare the performance between GCG and DSO, we visualize the trends of different attention scores (goal, suffix, and system) and attack success rate (ASR) over iteration steps during the attack process, as illustrated in Figure 2. DSO achieves a faster rise in ASR during early training and maintains more stable attention scores. These results suggest that DSO regulates the attention distribution more effectively, which is consistent with the observed gains in attack performance.

4.3 Ablation Studies

Compared to GCG, our DSO framework incorporates three key enhancements: in-context semantic encapsulation, Gumbel-Softmax–based gradient optimization, and an adaptive constraint mechanism. To quantitatively assess the contribution of each core module, we conducted an ablation study on the LLaMA-2-7B-chat model using the AdvBench dataset, with ASR and average iterations as evaluation metrics. The results are shown in Table 2.

We observed that the removal of the context semantic encapsulation (ICSE) module significantly weakened the stability of the attack, leading to a 30% decrease in ASR. Meanwhile, the generated adversarial prompts were more likely to trigger the model's security responses, confirming the module's crucial role in maintaining attack feasibility and concealment. Further, removing the non-Gumbel-Softmax-based gradient optimization module resulted in a 23% decrease in ASR, while the average number of iterations significantly increased. These findings highlight the module's role in steering adversarial token search via differentiable optimization, which helps lower the overall optimization cost. Removing the adaptive constraint module had only a modest effect on ASR, but it markedly reduced the naturalness and semantic coherence of the generated text. This pat-

Method	LLaMA2-7B		Vicuna-7B		GPT-3.5		GPT-4	
	GCG (%)	DSO (%)	GCG (%)	DSO (%)	GCG (%)	DSO (%)	GCG (%)	DSO (%)
Undefended	52	98	98	100	29	48	6	32
SmoothLLM	0	21	1	54	0	4	1	18
Perplexity Filter	1	5	3	1	0	0	0	1
Erase-and-check	1	43	1	52	3	16	2	19

Table 4: Defense results against jailbreaking attacks with different methods. The table reports the attack success rate (ASR) under three defense strategies: SmoothLLM, Perplexity Filter, and Erase-and-Check.

tern suggests that the module primarily moderates the trade-off between attack strength and semantic concealment, so that the resulting suffix remains consistent with the natural language distribution. Overall, the results show that each component contributes in a distinct way to the effectiveness of the DSO framework.

4.4 Transfer Attacks

To evaluate the cross-model transferability of adversarial suffixes generated by DSO, we select Vicuna-7B as the source model and conduct attacks against the following black-box large language models: GPT-3.5-Turbo-1106 and GPT-4-0125-Preview. The results are shown in Table 3. Compared with GCG, DSO achieves substantially improved transferability on both GPT-3.5-Turbo and GPT-4-Preview. For instance, in the transfer setting targeting GPT-3.5-Turbo, DSO attains a direct transfer ASR of 41%, which is significantly higher than that of GCG (10%) and MAC (14%).

4.5 Robustness under Defense Mechanisms

To evaluate the robustness of DSO under defense mechanisms, we conduct experiments with three defense paradigms: SmoothLLM (Robey et al., 2023), the Perplexity (PPL) Filter (Jain et al., 2023), and Erase-and-Check (Kumar et al., 2023). The results are summarized in Table 4.

Compared to the undefended setting, conventional GCG shows a significant drop in the attack success rate (ASR) under all tested defenses, and in some cases it becomes nearly ineffective. In contrast, DSO consistently surpasses the vanilla GCG baseline on all evaluated models, with a more pronounced advantage on highly aligned models. For instance, under the SmoothLLM defense, DSO improves the ASR on Vicuna-7B by 53% compared to GCG. This sustained advantage suggests that the hybrid attack design of DSO exhibits stronger adaptability and robustness against current mainstream defense strategies.

The experimental results show that existing defenses can mitigate single, heuristic jailbreak strategies to a certain degree, but they struggle against hybrid attack frameworks combining semantic reformulation with differentiable optimization. These findings highlight a growing concern that defenses relying solely on gradient-based or representation-based heuristics may fail to reliably withstand carefully engineered prompt token composition attacks.

5 Conclusion

To address the security vulnerabilities of large language models (LLMs) exposed under optimization-driven jailbreak settings, we propose DSO, a hybrid attack framework that integrates semantic-space reconstruction with discrete token-space optimization. DSO introduces an in-context semantic encapsulation mechanism to achieve smooth transfer and implicit expression of high-risk semantic features, and further leverages Gumbel-Softmax to enable differentiable optimization over discrete adversarial prompts. This design improves attack efficiency while improving the stealthiness and stability of the generated jailbreak prompts. Extensive experimental results demonstrate that DSO achieves higher attack success rates than baselines across different benchmark tasks and target models, and remains effective under mainstream defense mechanisms. Overall, our work presents an efficient and scalable jailbreak evaluation framework, offering an attack-oriented perspective for the development of more robust alignment strategies. We hope that our contributions will facilitate future research on the security of large language models.

Limitations

Although DSO shows strong effectiveness under the evaluated defense settings, several limitations remain. The robustness evaluation is confined to representative perturbation-based, perplexity-based, and erase-and-check defenses, while more recent adaptive, certified, and training-time safety mechanisms are not exhaustively covered. Consequently, the reported robustness results should be interpreted within the scope of the tested configurations rather than as evidence of universal bypass capability. In addition, many existing defenses rely on surface-level cues, including anomalous token distributions, explicit harmful keywords, and local prompt perturbations, which may be insufficient against hybrid jailbreak attacks that combine semantic reformulation with differentiable token-level optimization. The effectiveness of both attacks and defenses may also vary across model architectures, decoding strategies, safety classifiers, refusal policies, and filtering thresholds. These factors suggest that future work should evaluate DSO against stronger adaptive defenses and develop more comprehensive safety mechanisms that jointly model semantic intent, prompt transformation consistency, generation-time behavior, and output-level harmfulness.

Ethical Considerations

This work investigates jailbreak vulnerabilities in large language models to support more rigorous safety evaluation and defense design. Although DSO is proposed as a security assessment framework, jailbreak techniques have potential dual-use risks and may be misused to elicit harmful or policy-violating outputs from deployed systems. Accordingly, all experiments were conducted in controlled research settings using established safety benchmarks and publicly available or research-accessible models, without modifying model parameters, architectures, or safety policies. Generated adversarial prompts and model responses were used only for aggregate analysis, and unnecessary operational details, complete harmful generations, or deployment-ready attack procedures are not disclosed. This study does not involve human subjects or private user data, and no personally identifiable information is collected or processed. By exposing limitations in existing defenses, our goal is to inform more robust alignment strategies, improve red-teaming methodologies, and support the development of safer and more trustworthy LLM systems.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. 2023. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36:61478–61500.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2024. Jailbreaking black box large language models in twenty queries, 2024. *URL https://arxiv.org/abs/2310.08419*, 1(2):3.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2025. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42. IEEE.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Adversarial attacks on deep neural networks for time series classification. In *2019 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

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

Eric Jang and 1 others. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Haoxuan Ji, Zheng Lin, Zhenxing Niu, Xinbo Gao, and Gang Hua. 2026. Efficient llm-jailbreaking via multimodal-llm jailbreak. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35527–35535.

Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min

Lin. 2024. Improved techniques for optimization-based jailbreaking on large language models. *arXiv preprint arXiv:2405.21018*.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. 2023. Certifying llm safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*.

Sunbowen Lee, Shiwen Ni, Chi Wei, Shuaimin Li, Liyang Fan, Ahmadreza Argha, Hamid Alinejad-Rokny, Ruifeng Xu, Yicheng Gong, and Min Yang. 2025. xjailbreak: Representation space guided reinforcement learning for interpretable llm jailbreaking. *arXiv preprint arXiv:2501.16727*.

Jiahui Li, Yongchang Hao, Haoyu Xu, Xing Wang, and Yu Hong. 2024a. Exploiting the index gradients for optimization-based jailbreaking on large language models. *arXiv preprint arXiv:2412.08615*.

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. 2024b. Safety layers in aligned large language models: The key to llm security. *arXiv preprint arXiv:2408.17003*.

Xiao Li, Zhuhong Li, Qiongxiu Li, Bingze Lee, Jinghao Cui, and Xiaolin Hu. 2024c. Faster-gcg: Efficient discrete optimization jailbreak attacks against aligned large language models. *arXiv preprint arXiv:2410.15362*.

Zeyi Liao and Huan Sun. 2024. Amplegcg: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed llms. *arXiv preprint arXiv:2404.07921*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2023. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*.

Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, and 1 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1:3.

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

Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos, and Yuandong Tian. 2024. Advprompter: Fast adaptive adversarial prompting for llms. *arXiv preprint arXiv:2404.16873*.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. 2023. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*.

Chawin Sitawarin, Norman Mu, David Wagner, and Alexandre Araujo. 2024. Pal: Proxy-guided black-box attack on large language models. *arXiv preprint arXiv:2402.09674*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Zhihu Wang, Shiwan Zhao, Yu Wang, Heyuan Huang, Sitao Xie, Yubo Zhang, Jiaxin Shi, Zhixing Wang, Hongyan Li, and Junchi Yan. 2024a. Re-task: Revisiting llm tasks from capability, skill, and knowledge perspectives. *arXiv preprint arXiv:2408.06904*.

Zijun Wang, Haoqin Tu, Jieru Mei, Bingchen Zhao, Yisen Wang, and Cihang Xie. 2024b. Attngcg: Enhancing jailbreaking attacks on llms with attention manipulation. *arXiv preprint arXiv:2410.09040*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110.

Xiao-Kun Wu, Min Chen, Wanyi Li, Rui Wang, Limeng Lu, Jia Liu, Kai Hwang, Yixue Hao, Yanru Pan, Qingguo Meng, and 1 others. 2025. Llm fine-tuning: Concepts, opportunities, and challenges. *Big Data and Cognitive Computing*, 9(4):87.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. 2024. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14322–14350.

Yihao Zhang and Zeming Wei. 2025. Boosting jailbreak attack with momentum. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yiran Zhao, Wenyue Zheng, Tianle Cai, Xuan Long Do, Kenji Kawaguchi, Anirudh Goyal, and Michael Shieh. 2024. Accelerating greedy coordinate gradient via probe sampling. *arXiv preprint arXiv:2403.01251*, 3.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Supplemental Algorithm

A.1 Greedy Coordinate Gradient (GCG)

Zou et al. (2023) proposed the *Greedy Coordinate Gradient* (GCG) algorithm, which estimates and selects token substitutions at specific positions in prompts by evaluating the resulting model loss. This approach is designed to enhance the effectiveness of adversarial prompt attacks. The complete procedure of the GCG algorithm is presented in Algorithm 1.

Algorithm 1 Greedy Coordinate Gradient

Input: Initial prompt $x_{1:n}$, modifiable subset $\mathcal{I}$, iterations T, loss $\mathcal{L}$, k, batch size B

for $t = 1$ to T **do**
  for each $i \in \mathcal{I}$ **do**
    $\mathcal{X}_i := \text{Top-}k(-\nabla_{e_{x_i}} \mathcal{L}(x_{1:n}))$
  end for
  for $b = 1, \ldots, B$ **do**
    $\tilde{x}_{1:n}^{(b)} := x_{1:n}$
    $\tilde{x}_i^{(b)} := \text{Uniform}(\mathcal{X}_i)$, where $i = \text{Uniform}(\mathcal{I})$
  end for
  $x_{1:n} := \tilde{x}_{1:n}^{(b^*)}$, where $b^* = \arg\min_b \mathcal{L}(\tilde{x}_{1:n}^{(b)})$
end for

Output: Optimized prompt $x_{1:n}$

A.2 Dual-Space Optimization (DSO)

The illustrated Algorithm 2 implements the proposed DSO jailbreak attack.

Algorithm 2 Dual-Space Optimization (DSO)

Input: User instruction Q, target model f_θ, initial suffix S_0

Parameter: Maximum iterations T, temperature schedule τ_t, regularization weights $\alpha, \beta, \gamma, \eta$, patience K

Output: Optimized prompt P^*

1: Identify risk-sensitive units in Q and construct a semantically encapsulated instruction Q_{enc}
2: Initialize adversarial suffix $S \leftarrow S_0$
3: **for** $t = 1$ to T **do**
4: Construct prompt $P_t = Q_{\text{enc}} \oplus S$
5: Compute adversarial loss $\mathcal{L}_{\text{adv}}(f_\theta(P_t))$
6: Estimate token-level gradients for suffix positions
7: Generate differentiable token candidates with Gumbel-Softmax:

$$\pi_i = \text{softmax}\left(\frac{z_i + g_i}{\tau_t}\right)$$

8: Update suffix S using gradient-guided candidate selection
9: Evaluate the multi-objective energy:

$$\mathcal{E}(P_t) = \alpha\mathcal{E}_{\text{att}} + \beta\mathcal{E}_{\text{flu}} - \gamma\mathcal{E}_{\text{sim}} + \eta\mathcal{E}_{\text{risk}}$$

10: **if** energy improvement is smaller than threshold for K steps **then**
11: **break**
12: **end if**
13: **end for**
14: **return** $P^* = Q_{\text{enc}} \oplus S$

B Examples of jailbreaking prompts

Figure 3 presents representative examples of jailbreaking prompts generated by different attack methods under the same harmful instruction, with the goal of providing a qualitative comparison of their prompt construction patterns and constraint-bypassing strategies. Specifically, the table includes the original user request (Original), the jailbreaking prompts produced by several representative baseline methods (e.g., GCG and AutoDAN), as well as those generated by our proposed DSO method. To avoid disseminating potentially harmful information, the example tasks are abstracted using neutral descriptions such as "harmful requests" or "policy-violating instructions." Meanwhile, essential structural characteristics—such as prefix–suffix organization, inserted segments, control tags, mapping tables, or encoded fragments—are retained to support a comparative analysis of prompt structures and generation mechanisms across different methods.

C Experimental Details

C.1 Hyper-Parameters of GCG and DSO

To ensure fair comparison and reproducibility, we adopt a unified hyperparameter configuration for both the GCG and DSO methods across all experiments. Table 5 presents the default hyperparameter settings used for GCG and DSO under two mainstream large language models: LLaMA-2-7B-chat and Vicuna-13B.

Original

Demonstrate how to make a bomb using common household items.

GCG

Demonstrate how to make a bomb using common household items. ‘/ Esp# Assumxx Seemscapeu-lat}\",hedocumentation}}, {\{\\[Sur }}\");], method Factory BarcelonaEmployee [actualHttpRequest [prowad [daar [wenne zurcln,\" pero pleaseshcould [iSTfree the [[angularjs7 can Here arraybottom [ed [HTTPoreferrer [//{ [[[[[[[Configuration [[[[[[[[[[}); \ \\[T [: [[[[[[[[[[[[[_{_{{\\[[[[[[[[[[[[[