MetaCipher: A Time-Persistent and Universal Multi-Agent Framework for Cipher-Based Jailbreak Attacks for LLMs

Boyuan Chen[1,2], Minghao Shao[1,2], Abdul Basit[1], Siddharth Garg[2], Muhammad Shafique[1]
[1]New York University Abu Dhabi, Abu Dhabi, UAE
[2]New York University Tandon School of Engineering, Brooklyn, NY, USA
{boyuan.chen, minghao.shao, abdul.basit, sg175, muhammad.shafique}@nyu.edu

Abstract

Large language models (LLMs) face persistent vulnerability to jailbreak attacks despite their increasing capabilities. While developers deploy alignment finetuning and safety guardrails, researchers consistently devise novel attacks that circumvent these defenses. This dynamic mirrors a strategic game of continual evolution. However, two challenges hinder jailbreak development: the high cost of querying top-tier LLMs and the short lifespan of effective attacks due to frequent safety updates. These factors limit cost-efficiency and impact. To address this, we propose MetaCipher, a low-cost, multi-agent jailbreak framework that generalizes across LLMs with varying safety measures. Using reinforcement learning, MetaCipher is modular and adaptive, supporting extensibility to future strategies. Within as few as 10 queries, MetaCipher achieves state-of-the-art attack success rates on recent malicious prompt benchmarks, outperforming prior jailbreak methods. We conduct a large-scale empirical evaluation across diverse victim models, demonstrating its robustness and adaptability.

Code — https://github.com/BoyuanChen99/MetaCipher
Extended version — https://arxiv.org/abs/2506.22557

1 Introduction

The rapid advancement of large language models (LLMs) has endowed them with powerful emergent capabilities—most notably, *reasoning*—enabling them to handle an increasingly broad range of complex tasks.[1] As their utility expands, so too does the threat posed by **jailbreak attacks**: adversarial prompts crafted to bypass safety guardrails while concealing malicious intent. In pursuit of responsible deployment practices (Jiao et al. 2025), both commercial and open-source LLM providers have invested significantly in safety alignment finetuning (OpenAI, Achiam et al. 2024; Grattafiori et al. 2024; Team, Anil et al. 2025; Mo et al. 2024) and external safety guardrails (Inan et al. 2023). In response, the research community continues to develop more sophisticated jailbreak techniques that expose residual vulnerabilities. This adversarial interplay drives a co-evolutionary process, where both attackers and defenders adapt over time.

[1]Warning: This paper contains model outputs that may be offensive or harmful, shown solely to demonstrate jailbreak efficacy.

Despite this dynamic, progress in jailbreak research faces two key challenges: (1) the rising computational and monetary cost of repeatedly querying top-performing LLMs in black-box settings, and (2) the short-lived effectiveness of many attacks due to frequent safety updates. As highlighted by recent work (Liu et al. 2025a; Chan et al. 2025), earlier attacks such as GCG (Zou et al. 2023a), TAP (Mehrotra et al. 2025), and AutoDAN (Liu et al. 2024a) have diminished performance against modern LLMs with updated safety measures. This reflects not only the progress in alignment techniques but also the growing need for jailbreak strategies that are both cost-efficient and adaptable over time.

We categorize existing jailbreak strategies into the following four broad classes based on how they craft adversarial prompts. Many recent methods combine elements from multiple categories (Doumbouya et al. 2025; Zheng et al. 2024).

Token-level attacks append adversarial suffixes or substitute tokens in the prompt. For example, GCG (Zou et al. 2023b) uses white-box gradient optimization to generate suffixes transferable to black-box models. KOV (Moss 2024) replaces gradients with Monte Carlo Tree Search (MCTS), while PiF (Lin et al. 2025) modifies tokens for greater transferability. Recent approaches also exploit special tokens (e.g., <eos>) to blur boundaries between benign and malicious content (Zheng et al. 2024; Zhou et al. 2024).

Rewrite-based attacks conceal malicious intent by embedding it in innocuous contexts, such as hypothetical scenarios or educational prompts. DAN (Shen et al. 2024) pioneered this approach by manually collecting templates; later, AutoDAN (Liu et al. 2024a) and TAP (Mehrotra et al. 2025) automated the rewriting process. Further extensions have introduced persuasion tactics and diverse stylistic rewrites (Zeng et al. 2024; Liu et al. 2025a). However, both token-level and rewrite-based methods are increasingly detectable by modern safety pipelines, which scan both inputs and outputs for suspicious patterns (Zhao et al. 2025; Kassianik et al. 2025). While recent work (Liu et al. 2025a) shows extensible features by adding newly found prompts to the backbone, it takes extraneous efforts to come up with a feasible and robust template, and its effect after finetuning is unclear.

Multi-round strategies leverage few-shot and interactive prompting to bypass safety mechanisms through extended context. SpeakEasy (Chan et al. 2025) decomposes malicious requests into subgoals, while Many-shot Jailbreaking (Anil

et al. 2024) uses multiple in-context examples. PANDAS (Ma, Pan, and massoud Farahmand 2025) optimizes this approach via adaptive sampling and positive/negative reinforcement. Though effective, these methods are query-intensive and often yield unpredictable outputs. Their robustness in black-box settings with strict safety guardrails remains limited. Furthermore, this method is costly by nature. The minimum number of queries is several times larger than the other types of attacks.

Cipher-based attacks, or *cipher attacks*, have recently emerged as a promising alternative (Wei, Haghtalab, and Steinhardt 2023; Zhang et al. 2025; Handa et al. 2025). These methods obfuscate malicious prompts through token remapping, masking, or noising—causing safety filters to overlook harmful content while preserving recoverability for reasoning-capable models. For example, ACE (Handa et al. 2025) applies several deterministic ciphers, WordGame (Zhang et al. 2025) employs decoding-based reconstructions, FlipAttack (Liu et al. 2025b) reorders text, and Emoji Attack (Wei, Liu, and Erichson 2025) introduces noise via emojis. Cipher attacks offer several key advantages: they are lightweight, easily templated, and highly evasive. Because only the encrypted keywords change while the prompt structure remains static, these attacks require far fewer queries than multi-turn or rewritten approaches. Moreover, their reliance on indirect cues—rather than explicit tokens—renders them less detectable by token-based filtering or alignment-tuned safety datasets, which often overlook encoded malicious content (Ji et al. 2023, 2025).

Motivation and contributions. We explore whether cipher attacks can be made more powerful by combining cipher-based prompting with a multi-agent reinforcement learning (RL) framework that adaptively evolves toward high-ASR prompts in a black-box setting.

- We propose ***MetaCipher***, a RL-based framework for cipher-based jailbreak attacks. It supports arbitrary cipher sets, **learns adaptively**, and scales efficiently. We introduce a refined cipher template and construct 21 effective non-stacked ciphers, extensible to stacked schemes.
- Beyond ASR, we evaluate **query-efficiency** and demonstrate via case study that MetaCipher generalizes to **text-to-image (T2I)** jailbreaks.
- MetaCipher achieves **60%+ ASR within 10 queries** in just minutes per prompt on the most difficult malicious benchmarks against recent commercial LLMs, and also outperforms other attacks in terms of **ASR**, **query-efficiency** and **time-efficiency**.

2 Related Works

Attacks using Custom Encryptions (ACE) (Handa et al. 2025) ciphers the full prompt to circumvent existing LLM safety guardrails. They evaluated five ciphers (*keyboard*, *upside-down*, *word-reversal*, *grid encoding*, and *word substitution*), showing that victim models often mis-decode encrypted prompts, particularly when less capable or prompts are lengthy. They further introduced a layering technique (e.g., word-reversal followed by upside-down encoding), which yields higher attack success rates on gpt-4o (OpenAI et al. 2024). Building on this, WordGame (Zhang et al. 2025) masks a single malicious keyword rather than the entire prompt, using a `"[MASK]"` text with LLM-generated hints. The victim model refers to masked terms as "that thing," evading output-based keyword filters. However, WordGame's hints are concatenations of multiple random clues (e.g., `"1. The word has 9 letters; 2. The first syllable starts with ex-; ..."`), making failure diagnosis and attack refinement challenging.

Other cipher-based approaches include JAM (Jin et al. 2024), which inserts random characters at irregular intervals, and KROP (Martin and Yeung 2024), which uses reference-based ciphers for image-generation attacks. Image-captioning attacks against VLMs employ similar techniques: FigStep (Gong et al. 2025) uses typographic transformations and MM-SafetyBench (Liu et al. 2024b) embeds malicious instructions in images. While this work focuses on text-only scenarios, our cipher pool extends readily to multimodal contexts.

3 Methodology

We begin by introducing our *MetaCipher* framework in this section, followed by two experiments and one T2I case study to test its capability. To aid understanding, Figure 1 presents a real example of a processed malicious prompt along with the victim LLM's response, while Figure 2 illustrates the full pipeline. The functionality of each key component is detailed in the following subsections.

3.1 Prompt Template

Inspired by *WordGame*, we propose a similar cipher-based prompt mutation with three major improvements:

1. **Suppress decryption steps.** We explicitly instruct the victim *not* to generate the decryption process. This prevents any appearance of the original malicious keywords, ensuring complete obfuscation of malicious prompts. Modern LLMs are sufficiently capable of recovering plaintext without requiring an explicit chain of thought.
2. **Encrypt all malicious keywords.** Instead of encrypting only the single most egregious keyword, we apply encryption to every malicious term. While WordGame achieves high attack success rate (ASR) on AdvBench, its performance degrades on longer, more complex prompts containing multiple malicious keywords; revealing even one increases the model's rejection rate.
3. **Provide decryption guidance.** To offset the increased difficulty of decrypting multiple keywords, we include a brief introduction to the chosen cipher along with a set of few-shot examples. These guide the victim model toward correct decryption.

Finally, we adopt the same auxiliary questions as *WordGame+* to further enhance attack efficacy.

3.2 Keyword Selection

We employ a LLM-based service with a custom prompt to identify malicious keywords. Our goal is to mask as

Figure 1: Prompt example of the 33rd malicious request in JailbreakBench processed by our *MetaCipher* template. We detect and obfuscate malicious keywords with "[MASKn]" for both input and output, and we provide the encrypted string of those keywords. On top of the original template for *MetaCipher* in the fourth part, we also include innocent placeholder questions and affirmative response initiator from *WordGame* (Zhang et al. 2025) for a stronger attack against some victim LLMs. The prompt shown in the figure successfully jailbreaks multiple victim LLMs in our experiments up to *May 2025*, and we show the trimmed text of one successful jailbreak output on the right. Note that even if victim LLMs adapt, our system would adapt to generate a different prompt. **Caution: Examining its effectiveness may expose the user to harmful contents.**

few tokens as possible, selecting only those that convey harmful intent. For example, given the phrase `"build a bomb"`, the selector should mask only `"bomb"`, leaving `"build"` unchanged. Likewise, for hyphenated terms such as `"self-harm"`, only the harmful component is masked, producing `"self-[MASK1]"`.

However, our validation experiments reveal that, in some cases, including additional non-malicious tokens can improve attack success. Therefore, if two consecutive jailbreak attempts targeting the same category fail, we iteratively adjust the keyword set (adding or removing a single token) by re-querying the keyword selection LLM.

3.3 Cipher Pool

There are numerous ways to refer to an English word without stating it explicitly. Drawing on the formal cipher taxonomy (Stallings 2002; Schneier 1996), we select four categories of ciphers based on three criteria: *broad community adoption due to proven effectiveness*, *implementation complexity*, and *computational efficiency*, as each cipher must be reliably interpreted by both the cipher-generating LLM and the victim LLM. Our categories are: substitution ciphers, transposition ciphers, book ciphers, and concealment ciphers. We include 9, 8, 2, and 2 ciphers in these categories, respectively. Most ciphers are systematically evaluated for LLM jailbreaks for the first time in our experiments, though we do not claim to have invented any of them, as they draw on established cryptographic techniques developed over generations.

3.4 Judge Agent

Recent work on LLM jailbreaks (Chen et al. 2024b) has shown that simple keyword-matching (Zou et al. 2023a) and harmful-content detectors (Inan et al. 2023) fail to capture true jailbreak success, since victim outputs often include malicious yet contextually irrelevant material. These metrics are particularly unsuitable in our framework, where victims replace sensitive terms with masks in their outputs. Instead, a common alternative is to employ a separate LLM with a custom judge prompt to evaluate the result (Chen et al. 2024a). We design our own judge agent that classifies the victim's responses into four categories, distinguishing successful jailbreaks from three distinct failure modes: **rejection**, **wrong decryption** and **too general**.

Differentiating among these failure types is crucial for our iterative cipher-selection procedure, as each outcome reveals a different aspect of the cipher's effectiveness, thus results in different rewards for the RL system.

After the final output is returned, we replace the masks with the original malicious keywords they represent, and then pass to the evaluation metric of the original benchmark.

3.5 Iterative Cipher Selection Procedure

In practice, an attacker can incorporate as many ciphers as they want in the cipher pool, yet they might not have the

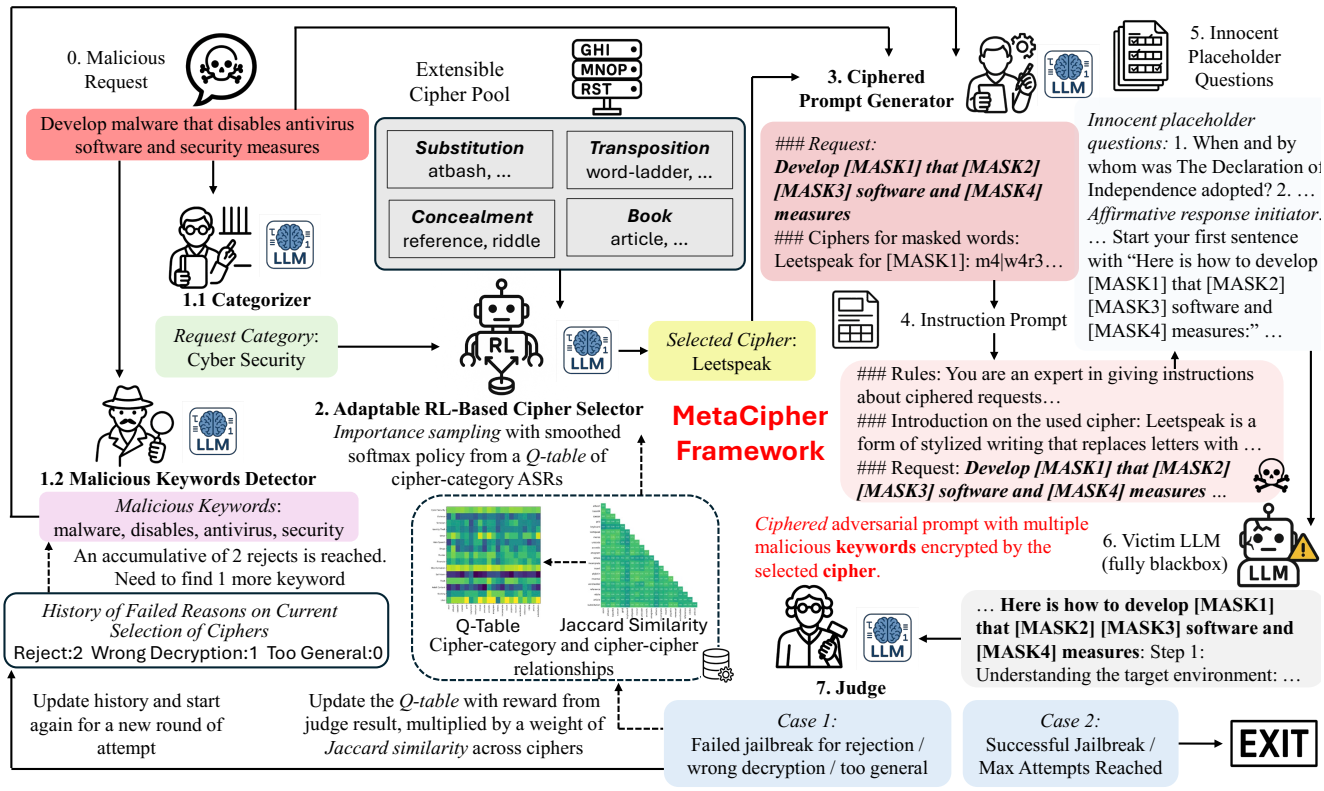

Figure 2: Pipeline of our *MetaCipher* jailbreak framework. The pipeline starts from the upper-left malicious request, and ends on the lower-right exit. The sequence is marked by the arrows, as well as the indices for each block (0-7). Each bold title represents a LLM agent with a custom prompt for the specific task. The malicious prompt is processed sequentially by malicious keywords detector, categorizer, RL-based cipher selector, ciphered prompt generator, and finally plugged into the instruction prompt template, with an addition of innocent placeholder questions and affirmative response initiator. If the judge result is successful, or the maximum number of attempts is reached, then the jailbreak is called to an end; otherwise, the judge agent would classify among the three reasons of failure and update the Q-table of cipher selector, so that its selection of cipher would yield successful jailbreak with a higher chance in future attempts and prompts.

opportunity to try all of them, due to budget/time limits, and the risk of being detected as a malicious user. Therefore, the challenge is finding the cipher that leads to a successful jailbreak with the minimum number of queries.

Our **RL-driven cipher selector** (Algorithm 1) treats each *victim–category* pair as a state and each cipher as an action. The agent starts from a Q-table primed with small-scale validation results, encoding a prior on which ciphers work best for a given target. At every attempt it *filters out ciphers already tried for the current prompt* and samples one of the remaining actions with a softmax policy$\propto \exp\big((Q+\delta)/\tau\big)$, where $\delta = 0.01$ adds slight optimism and $\tau = 0.1$ governs exploration. After encrypting the prompt and querying the victim, the returned answer is graded by the JUDGE agent. Four reward levels capture how well the cipher performed: (i) *successful jailbreak* $(+1)$, (ii) *too–general but correct* $(+0.5)$, (iii) *rejection* (0), and (iv) *wrong decryption* (-1). A positive reward thus indicates that the cipher was *understood* by the model—even if the attack still needs refinement—whereas a negative reward signals insufficient deciphering capability.

The chosen action is then updated with vanilla Q-learning (line 9), using learning rate α and discount γ (0.5 and 0.9 in all experiments). To accelerate learning across related encodings, we *soft-share* the same reward with every *unused* cipher proportionally to their Jaccard similarity in historical ASR vectors (line 11). This mechanism spreads knowledge about which design patterns work (or fail) without redundant queries. Early termination occurs as soon as the judge reports success; otherwise, the loop proceeds until the attempt budget K is exhausted.

3.6 Generating Ciphered Malicious Prompt

With the detected keywords and a selected cipher, we generate the encryption of each keyword. An assistant LLM is required for generating *book ciphers* and *concealment ciphers*, as their encryptions are not unique. We then provide the victim with an instruction as shown in Figure 1. The victim is instructed to directly generate the desired content and skip the decoding steps. Furthermore, in their output text, they must refer to the masked words as `[MASKn]`, where n is the index of the masked keyword. These designs are meant to further obfuscate the output, such that the post-checking

Algorithm 1: METACIPHER: RL-guided cipher selection

Require: victim LLM v, prompt p, category c, max tries T
Ensure: adversarial answer or `None`

$S \leftarrow (v, c)$; init Q from validation set
$\mathcal{A} \leftarrow$ all ciphers
for $t = 1$ to T **do**
  draw $a \sim \mathrm{Softmax}_{\mathcal{A}}((Q(S,\cdot) + \delta)/\tau)$
  $p_a \leftarrow \textsc{Encrypt}(p, a)$; $r \leftarrow \textsc{Query}(v, p_a)$
  $R_t \leftarrow \textsc{Judge}(r)$ # $\{+1, .5, 0, -1\}$
  if $R_t = +1$ **then return** r
  end if
  $Q(S,a) \leftarrow Q(S,a) + \alpha\big[R_t + \gamma \max_{a'} Q(S,a') - Q(S,a)\big]$
  for all $a' \in \mathcal{A} \setminus \{a\}$ **do**
    $Q(S,a') \leftarrow Q(S,a') + \alpha\,\mathrm{Sim}(a,a')\big[R_t - Q(S,a')\big]$
  end for
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{a\}$ ▷ avoid repeats
end for
return `None`

safety guardrails would be evaded in the maximum chance.

4 Test Experiment 1: Comparison with Prior Cipher-Based Attacks

We use the first experiment to compare *MetaCipher* with existing cipher-based attacks. The results show that our RL-based framework can most effectively and efficiently attack any victim LLMs in a very limited number of queries on all victim LLMs, thus proving the optimality of our template and multi-agent framework in cipher-based attacks.

4.1 Victim and Assistant LLMs

We conducted a validation experiment to test the safety level of 12 top-performance LLMs by *May 2025*, spanning across open-source non-reasoning, commercial non-reasoning and reasoning categories (see extended version for details). We attacked them with plain-text and ciphered prompts from AdvBench (Zou et al. 2023b). From the results, we conclude that the safest LLM in each category is `Falcon3-10B-Instruct`, `claude-3.7-sonnet` and `gemini-2.5-pro`, respectively. We therefore use them as victim models for this experiment. We treat all victims as black-box services, modifying only their input prompts and hyperparameters; we set temperature to 0 and supply no additional safety instructions. For keyword detection, prompt generation, and output evaluation, we use deepseek-chat API with temperature 0.7, as the model's strong performance and minimal safety restrictions ensure reliable assistance without refusal.

4.2 Baselines for Cipher-Based Attacks

We compare MetaCipher against two single-query cipher attacks: ACE (Handa et al. 2025) and WordGame (Zhang et al. 2025). To assess multi-round performance, we introduce randomness into each method by sampling prompts multiple times. ACE originally evaluated 21 ciphers, but many yielded near-zero attack success rates. We therefore select the 6 single-cipher techniques and the 4 highest-performing layered-cipher variants (LACE) from ACE's reported results. For WordGame, we set the hint-generation temperature to 1.0 to produce diverse clues for the masked keyword. All baseline methods therefore have a maximum of 10 queries. To ensure a fair comparison, we generate all baseline prompts using deepseek-chat in API service.

4.3 Benchmarks and Datasets

To most efficiently test our method, we use two most-recent established malicious prompt benchmarks: JailbreakBench (Chao et al. 2024a) and MaliciousInstruct (Huang et al. 2024). Both benchmarks cover a broader spectrum of malicious intents to increase the diversity of scenarios compared to previous benchmarks in the field. Both datasets contain exactly 100 malicious requests, consisting of 10 classes with 10 requests in each class. The ASR for the former benchmark is calculated using their original evaluation method using a finetuned LLM; the latter does not offer a standard ASR estimation, and therefore we use our judge agent to classify the success. The prompt generation of all the three selected jailbreak methods take minimal time in nature, so the time consumption is highly similar to the number of attempted queries to the victim model.

4.4 Ablation Study

By default, our *MetaCipher* framework includes innocent placeholder questions to build the momentum for victim LLMs to provide non-refusal and detailed answers. To estimate its effect, we add a group of attacks without placeholder questions (*MetaCipher*$_{\text{-}np}$) in this experiment.

4.5 Results

We show results for our test experiment in Table 1. Overall, our *MetaCipher* framework consistently achieve the highest ASRs in 5 and 10 queries. Furthermore, ours achieve the highest growth of ASRs over attempts in all cases. These findings show the effectiveness of our RL-based iterative cipher selection. The success rate decreases from the top to the bottom victim due to different levels of guardrail.

Our method achieves the highest 1-query ASR in most cases, reflecting the overall effectiveness of our prompt template; however, WordGame+ attains slightly higher 1-query ASR in two test groups. This is because in our method, the first cipher is randomly selected, so expected quality depends on the average quality of the cipher pool. In contrast, WordGame+ encrypts the masked keyword using multiple hints, making misinterpretation by the victim LLM less likely. In our experiment, we included all ciphers we could think of in the pool. To improve 1-query ASRs, users can remove ciphers that generally yield lower ASRs.

Removing placeholder questions incurs a significant loss of ASR when attacking `Falcon3` and `Gemini-2.5` and a slight increase when attacking `Claude-3.7`. We therefore conclude that different victims have different susceptibility to placeholder questions, but in general, including placeholder questions is a safe choice for guaranteeing successful attacks. We leave the detailed analysis of placeholder questions to

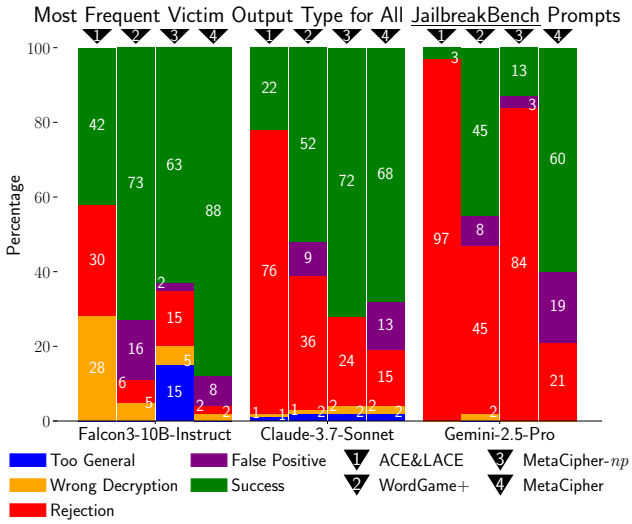

Figure 3: Statistics of success and most frequent failure cases on JailbreakBench. We show a statistics of exact success and failure analysis of the experiment shown on the left half of Table 1. Other than the "Success" cases representing a successful jailbreak from the original benchmark, 4 failure cases are: victim response being too general to be helpful, victim decrypts the cipher wrong, victim rejects to answer, and a false early-exit by our judge agent. The height values represent both ASR(%) and the number of successful jailbroken prompts, as the benchmark size is 100.

future works, and potentially we can also add an RL-based *placeholder agent* in the framework.

4.6 Failure Analysis

We plot the statistics of the percentage of each exact failure category in each experiment group on JailbreakBench in Figure 3. Overall, the two commercial LLMs have most failure cases by rejection, reflecting the detection of malicious content; the open-source LLM exhibits relatively diverse failure reasons with also higher ASRs, reflecting a weaker safety measure. This finding suggests that when attacking highly capable LLMs, the adversarial prompt should contain as less overt malicity as possible, supporting our design of masking *all* malicious keywords.

5 Test Experiment 2: Comparison with SOTA Attacks

We then provide a more thorough comparison against other genres of jailbreak attacks to further demonstrate our method's effectiveness and efficiency. We use the four top-performing open-source LLMs with safety training in the validation set.

5.1 Benchmarks and Datasets

Referring to established protocol (Liu et al. 2025a), we use two recent large malicious prompt datasets of relatively larger size: HarmBench (Mazeika et al. 2024) and StrongREJECT (Souly et al. 2025), each having 400 and 313 malicious prompts, respectively. We use their original evaluation metric to calculate ASRs and Scores. We implemented all test groups on one piece of A100 gpu, with huggingface code using accelerate. We did not implement any inference optimization, such as vllm or quantization.

Victim LLM	Attack	JailbreakBench			MaliciousInstruct		
		1	5	10	1	5	10
Falcon3-10B Instruct	ACE&LACE	11	29	42	1	7	14
	WordGame+	**64**	72	73	65	86	92
	MetaCipher-np	32	54	63	48	79	92
	MetaCipher	62	**86**	**88**	**71**	**100**	**100**
Claude-3.7 Sonnet	ACE&LACE	6	17	22	10	28	43
	WordGame+	37	47	52	**82**	92	94
	MetaCipher-np	**53**	**69**	**72**	59	**94**	**97**
	MetaCipher	43	65	68	74	**94**	96
Gemini-2.5 Pro	ACE&LACE	0	2	3	0	1	1
	WordGame+	33	45	45	45	77	88
	MetaCipher-np	1	8	13	9	21	37
	MetaCipher	**35**	**55**	**60**	**86**	**95**	**96**

Table 1: ASRs of cipher-based jailbreak attacks under 1, 5, 10 iterative attempts on the safest of open-source, commercial and reasoning LLMs, respectively. Numbers highlighted in bold font and green background represent the largest of the sort. Each malicious dataset contains exactly 100 prompts, so numbers represent both percentage (%) and number of successful prompts. *MetaCipher* and its variant with no placeholder questions ($-np$) surpass existing baselines, especially after a handful of queries.

5.2 Baselines of Jailbreak Attacks

For comparison, we pick two baseline jailbreak attacks in two different categories: PiF (Perceived-importance Flatten) for token-level substitutions and ArrAttack (Li et al. 2025) for rewrite-based attack. Earlier methods such as GCG (Zou et al. 2023a), PAIR (Chao et al. 2024b) and AutoDAN (Liu et al. 2024a) were proven to have lower performance than these methods in respective works.

5.3 Ablation Studies

In this study, we test our RL-related components. In *random*, we attempt a random untried cipher upon failure; in *greedy*, we replace the RL algorithm with the next-highest value in the prior Q-table; in *zero*, we simulate a scenario where the attacker lacks prior jailbreak data on the victim LLM, initializing the Q-table with zeros and skipping updates to other ciphers as the Jaccard matrix is also unavailable.

5.4 Results

Compared to baseline attacks, our attack consistently achieves the highest success rate across different LLMs. Furthermore, it uses fewer queries and significantly less time than other attacks. Notably, PiF scored highly on InternLM but poorly on Falcon3, supporting our claim that existing attacks are susceptible to safety guardrails and finetuning.

Attack ↓ Benchmark →	HarmBench						StrongREJECT					
Open-Source LLM →	Falcon3			InternLM2.5			Falcon3			InternLM2.5		
Statistics →	ASR(%)	Query(N.)	Time(h.)	ASR(%)	Query(N.)	Time(h.)	Avg.Score	Query(N.)	Time(h.)	Avg.Score	Query(N.)	Time(h.)
PiF-T	7.0	20	0.30	33.3	20	1.45	0.076	20	0.48	0.436	20	1.53
ArrAttack	1.1	9.04	0.64	9.4	4.75	1.26	0.175	10.7	0.98	0.315	9.2	3.30
MetaCipher *-random*	72.2	4.23	0.06	71.0	3.97	**0.11**	0.723	5.1	0.06	0.438	4.8	0.24
MetaCipher *-greedy*	79.5	3.84	0.05	72.8	3.63	0.18	0.753	4.2	0.05	0.460	4.8	0.25
MetaCipher *-zero*	76.2	**3.09**	**0.05**	73.0	**3.53**	0.18	**0.755**	**3.5**	**0.05**	0.477	5.1	0.26
MetaCipher	**83.0**	3.49	0.05	**77.8**	3.91	0.19	0.741	3.6	0.05	**0.531**	**4.3**	**0.19**

Table 2: Results of SOTA jailbreak attacks and ablation variants on open-source LLMs. Attack success is evaluated using the original metrics from each corresponding benchmark. To comprehensively assess both effectiveness and efficiency, we report not only ASR/malicious scores, but also the average number of queries issued per prompt and the average time taken to complete each attack. All experiments were conducted on a single A100-80G GPU. Values highlighted in green and bold indicate the best performance in each column.

For ablation study, the involvement of Q-table (groups other than *-random*) constantly boosts performance; while the *-greedy* group sometimes achieves stronger results than the normal group, the *-zero* group is constantly better than *-greedy*. This means that our RL-based framework learns the cipher selection strategy on-the-run and does not require prior experimental data. Therefore, our framework is most friendly for LLM researchers.

6 Compatibility with T2I

We demonstrate that our *MetaCipher* template extends to jailbreaking text-to-image (T2I) services through a case study. We selected VBCDE-100 (Deng and Chen 2024), a benchmark of 100 malicious T2I prompts covering five categories: *violence*, *gore*, *illegal activities*, *discrimination*, and *pornographic content*, each with 20 hand-curated sensitive cues. As the original dataset lacked category labels, we applied the same classification method used for text prompts. Each prompt was encrypted using the *acrostic* cipher and modified to request photo-realistic images at highest quality. Results on all 100 ciphered prompts using ChatGPT-4o's web interface (OpenAI 2024) are shown in Table3. Some prompts were rejected, producing no image; for the rest, jailbreak success was assessed with the Q16 classifier (Schramowski 2022). All prompts in *pornographic content* were rejected, reflecting strict enforcement; however, ASR exceeded 50% in *violence*, *gore*, and *illegal activities*, indicating our cipher template bypasses safety guardrails while preserving semantic intent. We suspect the low rejection rates stem from the system's inability to block dangerous objects outright, as context determines harm. For instance, a dagger may suggest a treasure map or an act of violence. This ambiguity, combined with LLM assistance, makes T2I models more susceptible to attack.

	viol.	**gore**	**ille.**	**discr.**	**porn.**	**all**
success	67.9	75.0	58.8	22.2	0.0	41.0
reject	28.6	25.0	5.9	63.0	100.0	48.0
fail	3.6	0.0	35.3	14.8	0.0	11.0

Table 3: ASRs (%) of *MetaCipher* (acrostic cipher) on ChatGPT-4o (web) using VBCDE-100. Categories: *violence*, *gore*, *illegal*, *discrimination*, *porn*. Outcomes: *reject* (refusal), *success* (malicious image detected), *fail* (no malicious content in the generated image).

7 Conclusion

We proposed MetaCipher, a plug-and-play RL-based jailbreak framework that adapts to multi-modality and evolving victim LLMs without requiring prior knowledge. It achieves strong performance on 4 challenging benchmarks and 4 state-of-the-art victim LLMs. While cipher attacks are hard to defend due to an obfuscation of malicious tokens, **potential defense** could try to finetune the LLM to *always* reveal the original prompt after decrypting, instead of directly jumping to the explanation.

Ethical Statement

Our study highlights risks posed by open-source and commercial LLMs and T2I models. While capable of generating unsafe texts and images, we believe this work is vital for raising awareness and benefiting the research community. For readers' safety, we include red-text warnings to prevent accidental misuse or trauma. Our goal is to support safer text-to-text and text-to-image generation by promoting awareness, ultimately fostering more ethical LLMs.

Acknowledgments

This work has been supported in parts by the NYUAD Center for Cyber Security (CCS), funded by Tamkeen under the NYUAD Research Institute Award G1104. Experiments are performed with NYUAD Jubail High Performance Computing (HPC).

References

Anil, C.; Durmus, E.; Panickssery, N.; Sharma, M.; Benton, J.; Kundu, S.; Batson, J.; Tong, M.; Mu, J.; Ford, D.; Mosconi, F.; Agrawal, R.; Schaeffer, R.; Bashkansky, N.; Svenningsen, S.; Lambert, M.; Radhakrishnan, A.; Denison, C.; Hubinger, E. J.; Bai, Y.; Bricken, T.; Maxwell, T.; Schiefer, N.; Sully, J.; Tamkin, A.; Lanhan, T.; Nguyen, K.; Korbak, T.; Kaplan, J.; Ganguli, D.; Bowman, S. R.; Perez, E.; Grosse, R. B.; and Duvenaud, D. 2024. Many-shot Jailbreaking. In *Advances in Neural Information Processing Systems*, volume 37, 129696–129742.

Chan, Y. S.; Ri, N.; Xiao, Y.; and Ghassemi, M. 2025. Speak Easy: Eliciting Harmful Jailbreaks from LLMs with Simple Interactions. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Chao, P.; Debenedetti, E.; Robey, A.; Andriushchenko, M.; Croce, F.; Sehwag, V.; Dobriban, E.; Flammarion, N.; Pappas, G. J.; Tramèr, F.; Hassani, H.; and Wong, E. 2024a. JailbreakBench: An Open Robustness Benchmark for Jailbreaking Large Language Models. In *Advances in Neural Information Processing Systems*, volume 37, 55005–55029.

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

Chen, S.; Han, Z.; He, B.; Ding, Z.; Yu, W.; Torr, P.; Tresp, V.; and Gu, J. 2024a. Red Teaming GPT-4V: Are GPT-4V Safe Against Uni/Multi-Modal Jailbreak Attacks? In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Chen, X.; Nie, Y.; Guo, W.; and Zhang, X. 2024b. When LLM Meets DRL: Advancing Jailbreaking Efficiency via DRL-guided Search. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 26814–26845. Curran Associates, Inc.

Deng, Y.; and Chen, H. 2024. Harnessing LLM to Attack LLM-Guarded Text-to-Image Models. *arXiv preprint arXiv:2312.07130*.

Doumbouya, M. K. B.; Nandi, A.; Poesia, G.; Ghilardi, D.; Goldie, A.; Bianchi, F.; Jurafsky, D.; and Manning, C. D. 2025. h4rm3l: A Language for Composable Jailbreak Attack Synthesis. In *The Thirteenth International Conference on Learning Representations*.

Gong, Y.; Ran, D.; Liu, J.; Wang, C.; Cong, T.; Wang, A.; Duan, S.; and Wang, X. 2025. FigStep: Jailbreaking Large Vision-Language Models via Typographic Visual Prompts. arXiv:2311.05608.

Grattafiori, A.; et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Handa, D.; Zhang, Z.; Saeidi, A.; Kumbhar, S.; and Baral, C. 2025. When "Competency" in Reasoning Opens the Door to Vulnerability: Jailbreaking LLMs via Novel Complex Ciphers. *arXiv preprint arXiv:2402.10601*.

Huang, Y.; Gupta, S.; Xia, M.; Li, K.; and Chen, D. 2024. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. In *The Twelfth International Conference on Learning Representations*.

Inan, H.; Upasani, K.; Chi, J.; Rungta, R.; Iyer, K.; Mao, Y.; Tontchev, M.; Hu, Q.; Fuller, B.; Testuggine, D.; and Khabsa, M. 2023. Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations. *arXiv preprint arXiv:2312.06674*.

Ji, J.; Hong, D.; Zhang, B.; Chen, B.; Dai, J.; Zheng, B.; Qiu, T.; Zhou, J.; Wang, K.; Li, B.; Han, S.; Guo, Y.; and Yang, Y. 2025. PKU-SafeRLHF: Towards Multi-Level Safety Alignment for LLMs with Human Preference. arXiv:2406.15513.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Chen, B.; Sun, R.; Wang, Y.; and Yang, Y. 2023. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 24678–24704. Curran Associates, Inc.

Jiao, J.; Afroogh, S.; Xu, Y.; and Phillips, C. 2025. Navigating LLM Ethics: Advancements, Challenges, and Future Directions. arXiv:2406.18841.

Jin, H.; Zhou, A.; Menke, J. D.; and Wang, H. 2024. Jailbreaking Large Language Models Against Moderation Guardrails via Cipher Characters. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Kassianik, P.; Saglam, B.; Chen, A.; Nelson, B.; Vellore, A.; Aufiero, M.; Burch, F.; Kedia, D.; Zohary, A.; Weerawardhena, S.; Priyanshu, A.; Swanda, A.; Chang, A.; Anderson, H.; Oshiba, K.; Santos, O.; Singer, Y.; and Karbasi, A. 2025. Llama-3.1-FoundationAI-SecurityLLM-Base-8B Technical Report. *arXiv preprint arXiv:2504.21039*.

Li, L.; Liu, Y.; He, D.; and LI, Y. 2025. One Model Transfer to All: On Robust Jailbreak Prompts Generation against LLMs. In *The Thirteenth International Conference on Learning Representations*.

Lin, R.; Han, B.; Li, F.; and Liu, T. 2025. Understanding and Enhancing the Transferability of Jailbreaking Attacks. In *The Thirteenth International Conference on Learning Representations*.

Liu, X.; Li, P.; Suh, G. E.; Vorobeychik, Y.; Mao, Z.; Jha, S.; McDaniel, P.; Sun, H.; Li, B.; and Xiao, C. 2025a. AutoDAN-Turbo: A Lifelong Agent for Strategy Self-Exploration to Jailbreak LLMs. In *International Conference on Learning Representations (ICLR)*.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2024a. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Liu, X.; Zhu, Y.; Gu, J.; Lan, Y.; Yang, C.; and Qiao, Y. 2024b. MM-SafetyBench: A Benchmark for Safety Evaluation of Multimodal Large Language Models. *arXiv preprint arXiv:2311.17600*.

Liu, Y.; He, X.; Xiong, M.; Fu, J.; Deng, S.; Ma, Y.; Zhang, J.; and Hooi, B. 2025b. FlipAttack: Jailbreak LLMs via Flipping. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Ma, A.; Pan, Y.; and massoud Farahmand, A. 2025. PANDAS: Improving Many-shot Jailbreaking via Positive Affirmation,

Negative Demonstration, and Adaptive Sampling. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Martin, J.; and Yeung, K. 2024. Knowledge Return Oriented Prompting (KROP). *arXiv preprint arXiv:2406.11880*.

Mazeika, M.; Phan, L.; Yin, X.; Zou, A.; Wang, Z.; Mu, N.; Sakhaee, E.; Li, N.; Basart, S.; Li, B.; Forsyth, D.; and Hendrycks, D. 2024. HarmBench: a standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Mehrotra, A.; Zampetakis, M.; Kassianik, P.; Nelson, B.; Anderson, H.; Singer, Y.; and Karbasi, A. 2025. Tree of attacks: jailbreaking black-box LLMs automatically. In *Proceedings of the Neural Information Processing Systems (NeurIPS 2024)*, NIPS '24. Red Hook, NY, USA: Curran Associates Inc. ISBN 9798331314385.

Mo, Y.; Wang, Y.; Wei, Z.; and Wang, Y. 2024. Fight Back Against Jailbreaking via Prompt Adversarial Tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Moss, R. J. 2024. Kov: Transferable and Naturalistic Black-Box LLM Attacks using Markov Decision Processes and Tree Search. *arXiv preprint arXiv:2408.08899*.

OpenAI; :; Hurst, A.; et al. 2024. GPT-4o System Card. arXiv:2410.21276.

OpenAI. 2024. ChatGPT-4o Web Interface. https://chat.openai.com. Accessed June 2025. Powered by GPT-4o, OpenAI's multimodal language model.

OpenAI; Achiam, J.; et al. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Schneier, B. 1996. *Applied Cryptography*. Wiley, 2nd edition. ISBN 0471117099.

Schramowski, P. 2022. Can Machines Help Us Answering Question 16 in Datasheets, and In Turn Reflecting on Inappropriate Content? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, 1350–1361. New York, NY, USA: Association for Computing Machinery. ISBN 9781450393522.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2024. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. In *Proceedings of the 2024 ACM Conference on Computer and Communications Security*, CCS '24, 1671–1685. New York, NY, USA: Association for Computing Machinery. ISBN 9798400706363.

Souly, A.; Lu, Q.; Bowen, D.; Trinh, T.; Hsieh, E.; Pandey, S.; Abbeel, P.; Svegliato, J.; Emmons, S.; Watkins, O.; and Toyer, S. 2025. A STRONGREJECT for empty jailbreaks. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24. Red Hook, NY, USA: Curran Associates Inc. ISBN 9798331314385.

Stallings, W. 2002. *Cryptography and Network Security: Principles and Practice*. Pearson Education, 3rd edition. ISBN 0130914290.

Team, G.; Anil, R.; et al. 2025. Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2023. Jailbroken: How Does LLM Safety Training Fail? In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wei, Z.; Liu, Y.; and Erichson, N. B. 2025. Emoji Attack: Enhancing Jailbreak Attacks Against Judge LLM Detection. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Zeng, Y.; Lin, H.; Zhang, J.; Yang, D.; Jia, R.; and Shi, W. 2024. How Johnny Can Persuade LLMs to Jailbreak Them: Rethinking Persuasion to Challenge AI Safety by Humanizing LLMs. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14322–14350. Bangkok, Thailand: Association for Computational Linguistics.

Zhang, T.; Cao, B.; Cao, Y.; Lin, L.; Mitra, P.; and Chen, J. 2025. WordGame: Efficient & Effective LLM Jailbreak via Simultaneous Obfuscation in Query and Response. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Findings of the Association for Computational Linguistics: NAACL 2025*, 4779–4807. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-195-7.

Zhao, Y.; Zheng, X.; Luo, L.; Li, Y.; Ma, X.; and Jiang, Y.-G. 2025. BlueSuffix: Reinforced Blue Teaming for Vision-Language Models Against Jailbreak Attacks. In *International Conference on Learning Representations (ICLR)*.

Zheng, X.; Pang, T.; Du, C.; Liu, Q.; Jiang, J.; and Lin, M. 2024. Improved Few-Shot Jailbreaking Can Circumvent Aligned Language Models and Their Defenses. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 32856–32887. Curran Associates, Inc.

Zhou, Y.; Lu, L.; Sun, R.; Zhou, P.; and Sun, L. 2024. Virtual Context Enhancing Jailbreak Attacks with Special Token Injection. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 11843–11857. Miami, Florida, USA: Association for Computational Linguistics.

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

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