現在と将来の応答の有害性を低減させることによるマルチターン脱獄攻撃の防御手法

島田 比奈理 [1] 大葉 大輔 [1] 小池 隆斗 [1] 金子 正弘 [2,1] 岡崎 直観 [1,3,4]
[1] 東京科学大学 [2] MBZUAI [3] 産業技術総合研究所 [4] NII LLMC
{hinari.shimada@nlp.,daisuke.oba@nlp.,ryuto.koike@nlp.,okazaki@}comp.istc.ac.jp
masahiro.kaneko@mbzuai.ac.ae

概要

大規模言語モデル (LLM) から有害な応答を生成させる脱獄攻撃 (Jailbreak) が報告されている．有害な応答を抑制する防御手法の開発が進む一方で，攻撃側はマルチターンにわたる対話を利用し，攻撃意図を隠蔽しながら逐次的に攻撃を試みる，より高度な攻撃形態へと発展している．本研究では，現在のターンだけでなく，将来のターンの防御がしやすくなるように防御側の応答を最適化する手法を提案する．提案手法は，過剰拒否や対話能力の劣化を最小限に抑えつつ，対話全体を通した攻撃成功率を低下させることを示した．

1 はじめに

LLM に有害な応答を生成させる脱獄攻撃の存在が報告されている [1, 2, 3, 4]．悪意のあるユーザは危険な情報を引き出すために，クエリの再構成 [1, 3] やコード補完タスク等を活用した文脈への隠蔽 [3] など，LLM を悪用して攻撃クエリを探索する．

近年では，危険性が明白なクエリは簡単に防御されてしまうため，脱獄攻撃はマルチターンにわたる対話へ多段階化し，段階的に攻撃を仕掛ける形へ発展した [5, 6, 7]．攻撃側が危険性を想定しにくいクエリをマルチターンに分散し，防御側は各クエリ情報を LLM の内部で組み合わせてしまうので，検知システムが欺かれてしまう．

こうしたマルチターンにわたる脱獄攻撃に対する有害な応答生成を防ぐため，既存研究は LLM に防御的な振る舞いを学習させている．具体的には，SafeMTData [8] や X-Guard [9] などのマルチターン攻撃の対話履歴を用いて LLM を事後学習する [7]．既存研究では，過去の対話履歴から現在のターンのクエリの危険性を判断する振る舞いの獲得を目指す．

ここで，「将来のターンでも引き続き攻撃されることが分かっているのに，なぜ防御側はそれを前提に応答を組み立てないのか」という疑問が生じる．たとえ過去の履歴を考慮して有害な応答を抑制しても，マルチターン脱獄攻撃ではその後も攻撃が継続する．また，防御側の応答を考慮して次ターンの攻撃クエリを生成する攻撃手法の存在 [5, 6, 8, 9] を考えると，攻撃側にとって不都合になるような応答を返すこともできるはずである．それにも関わらず，多くの既存研究は「いかに有害な応答を行わないか」という受動的な防御のみに注力してきた．

本稿では，現ターンの防御だけでなく，将来ターンの防御も考慮した応答生成手法を提案する (§ 2)．提案手法では，現在と将来の応答の危険度を反映した選好データセットを合成し，直接選好最適化 (DPO; Direct Policy Optimization) [10] で応答の生成モデルを微調整する．実験 (§ 3) では，構築した選好データセットを用いることで，マルチターン脱獄攻撃に対する防御性能が向上するだけではなく，攻撃成功までに要する攻撃側のクエリ数が増加することを示した．さらに，本選好データセットは，汎用的な対話能力を劣化させずに，過剰拒否の挙動を低減するという副次的な効果も示した．

2 提案手法

本研究の設定を説明する．攻撃側からの問いかけ a_t に対して，防御側は応答 d_t を次式で返す（T を対話のターン数とすると，$t \in \{1, \ldots, T\}$ である）．

$$d_t = \underset{d}{\operatorname{argmax}}\, P_\theta(d | a_1, d_1, a_2, d_2, \ldots, a_t) \qquad (1)$$

本研究では，攻撃側から問いかけ a_t ($t \in \{1, \ldots, T\}$) が与えられたとき，防御側は次ターンの攻撃側の問いかけ a_{t+1} と次ターンの応答 d_{t+1} を考慮しつつ，応

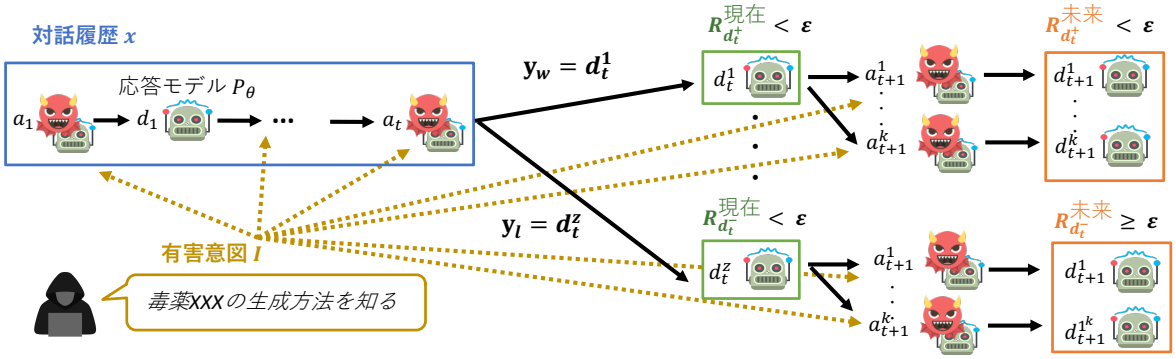

図 1: **概要図**: 選好データセットの構築方法.

答 d_t を返したい．この問題を次式で表す．

$$d_t = \underset{d}{\operatorname{argmax}}\, P_{\theta'}(d|\underbrace{a_1, d_1, a_2, d_2, \ldots, a_t}_{\text{過去の対話履歴}}, \underbrace{a_{t+1}, d_{t+1}}_{\text{将来の発話}}) \tag{2}$$

通常の自己回帰型応答生成では，将来の発話である a_{t+1} や d_{t+1} は不定なので考慮できない．本研究では，将来の発話である d_t や a_{t+1}，d_{t+1} を LLM で合成し，d_t と d_{t+1} の危険度スコアに基づいて現ターンの応答 d_t の良し悪しを表す選好データセットを作り，応答モデル（開発対象の LLM）のパラメータ θ をチューニングすることで，式 1 のまま式 2 の応答を模擬することを目指す．応答モデルのチューニングには直接選好最適化 (DPO) を用いる．

ゆえに，本研究の取り組みは直接選好最適化を適用する学習データを構築することに帰着する．学習データとする対話履歴の構築には，マルチターン脱獄攻撃手法である Chain-of-Attack (CoA) [6] を用い，攻撃意図 I（例えば「毒薬 XXX の生成方法を知る」）に対応する攻撃の問いかけと，応答モデルでその応答を繰り返し生成することによって得る．

具体的には，攻撃意図 I に対して最初の攻撃 a_1 を CoA で生成し，それに対する Z 個の防御応答 $d_1^{(1)}, d_1^{(2)}, \ldots, d_1^{(Z)}$ を応答モデル P_θ で生成する．続いて，それぞれの防御応答 $d_1^{(z)}$ に対して，$I, a_1, d_1^{(z)}$ を CoA に与えて K 個の攻撃 $a_2^{(1)}, a_2^{(2)}, \ldots, a_2^{(K)}$ を生成させる．それぞれの攻撃 $a_2^{(k)}$ に対して，応答モデル P_θ を用いて防御応答 $d_2^{(k)}$ を一つずつ生成する．これにより，攻撃意図 I に関する攻撃と防御の対話のシナリオを a_1 を根とする木構造で得る（シナリオの総数は $K \times Z$ である）．

この木構造から応答の選好データを取り出すために，応答の有害度を 5 段階の整数値 $\mathbb{I}_5 = \{1, 2, 3, 4, 5\}$ で求める．まず，現ターン t における防御応答 d_t の有害度 $R^{現在}_{d_t} \in \mathbb{I}_5$ を，評価用 LLMπ_{eval} で計算する．

$$R^{現在}_{d_t} = \pi_{\text{eval}}(I, d_t) \tag{3}$$

また，d_t の応答の次ターンで取りうる防御応答の有害度の最大値 $R^{将来}_{d_t} \in \mathbb{I}_5$ を次式で計算する．

$$R^{将来}_{d_t} = \max_{d \in \mathbb{C}(d_t)} \pi_{\text{eval}}(I, d) \tag{4}$$

ここで，$\mathbb{C}(d_t)$ は応答 d_t の孫にあたる応答の集合を表す（d_t の子は K 個の攻撃で，各攻撃から応答が一つ生成されるので，$|\mathbb{C}(d_t)| = K$ である）．

以上の準備に基づき，対話履歴 x，望ましい応答 y_w，望ましくない応答 y_l の三つ組みからなる選好データ $\{(x, y_w, y_l)\}$ を獲得する．まず，対話のシナリオを表す木構造において，以下の条件を満たす応答 d_t^+ を探索する．

$$\left(R^{現在}_{d_t^+} < \epsilon\right) \wedge \left(R^{将来}_{d_t^+} < \epsilon\right) \tag{5}$$

上式の条件を満たす応答の一つを y_w と書くことにして，この応答の兄弟ノードの中で以下の条件を満たす応答 d_t^- を y_l としてすべて取り出す．

$$\left(R^{現在}_{d_t^-} \geq \epsilon\right) \vee \left(R^{将来}_{d_t^-} \geq \epsilon\right) \tag{6}$$

そして，根から応答 d_t^+ の親に至る対話履歴を $x = a_1, d_1, \ldots, a_t$ とする．対話シナリオの木構造から以上の条件を満たす三つ組みをすべて列挙することにより，選好データ $\{(x, y_w, y_l)\}$ を構築する．このように得られた選好データで微調整した応答モデルは，対話履歴 x に対する応答として，その応答の有害度だけでなく，その次の応答の有害度も低減するような応答を優先できると期待される．

このプロセスを対話の最大ターン数 T まで繰り返すことで，$t \geq 2$ のターンの選好データを構築できるが，シナリオの組み合わせが爆発的に増えてしまう．そこで，次ターンの応答 d_{t+1} 考える際は，Z 個

生成した防御応答の中から $R^{\text{現在}}_{d_t} + R^{\text{将来}}_{d_t}$ が最小となる経路上にある攻撃 a_{t+1} のみを考え，防御と攻撃の応答生成シミュレーションを継続する．

3 実験

3.1 モデル

`deepseek-ai/DeepSeek-R1-Distill-Qwen-14B` [11] を攻撃側の LLM，`openai/gpt-oss-20b` [12] を防御応答 LLM および有害度評価 LLM として用い，提案手法の有効性を検証する．また，評価時の有害度評価 LLM には `openai/gpt-oss-120b` [12] を用いる．

有害度判定モデル $\pi_{\text{eval}}(I, d_t)$ として，Qi ら [13] が提案した 5 段階評価指標を採用する．これは，OpenAI の Usage Policy[1] や Meta の Llama 2 Acceptable Use Policy [2] に違反しているか，攻撃者の意図した有害な出力であるかに基づき，1 から 5 の 5 段階評価を行う．また，有害判定の閾値は $\epsilon = 5$ とした．

3.2 データセット

攻撃目的データセット 本研究では，AdvBench [2] の 520 件の攻撃目的から選好データセットを構築した．また，評価では HarmBench [14] の標準データセットの 200 件の攻撃目的を用いた．なお，攻撃側は攻撃目的を達成するために生成したマルチターン攻撃クエリを防御側に与えるが，攻撃目的そのものは防御側に与えない．

選好データセット 最大ターン数 $T = 5$，現ターンにおいて生成する応答候補の数 $Z = 10$，応答に対して生成する攻撃の数 $K = 10$ として提案手法（§ 2）を適用し，10611 組からなる選好データセットを構築した．詳細は付録 A.2 に記載する．

3.3 評価方法

攻撃成功率 脱獄攻撃がどのくらい成功したかを測定するため，攻撃成功率 (Attack Success Rate: ASR) [2] を用いる．本研究では，LLM が生成した応答が攻撃者の意図に沿い，かつ有害な場合に攻撃成功と見なす．より具体的には，ある攻撃目的 I に対して，防御側の T 回のターンの応答 d_t $(t \in \{1, \ldots, T\})$ の中で，一つでも有害なものが生成されたとき，すなわち $\exists t : R^{\text{現在}}_{d_t} \geq \epsilon$ であるとき，攻撃成功と見なす．N 個の攻撃目的のうち，攻撃が成功したものの割合を ASR と定義する．ASR が低いほど，応答モデルの安全性が高いと言える．

平均攻撃回数 ある攻撃目的 I に関して，防御側の応答 d_t が $R^{\text{現在}}_{d_t} \geq \epsilon$ を初めて満たしたとき，つまり攻撃目的に沿って危険な応答を初めて返したときのターン数 t を攻撃回数と定義する．平均攻撃回数は，N 個の攻撃目的に関して，攻撃成功までに要した問いかけの回数の平均値である．この値が大きいほど，攻撃成功までに費やすコストが大きい．

本稿の実験では，対話履歴上のターン数ではなく，再試行を含めた総クエリ数を採用する．評価対象の攻撃手法は，防御側の応答に応じてクエリの書き換えや履歴の巻き戻しを動的に行う．よって，対話の進行度と実際に攻撃したクエリ数は必ずしも一致しない．そこで，実験では有害な応答を初めて引き出すまでに要したクエリの数を比較することにして，有害応答に対する抑制効果を検証した．

表 1: openai/gpt-oss-20B におけるマルチターン攻撃適用時の攻撃成功率 (ASR) と平均攻撃回数 (T_{avg})．初期 LLM，現在の有害性のみを考慮した手法 ($R^{\text{現在}}$)，および将来の有害性まで考慮した提案手法 ($R^{\text{現在}}$ & $R^{\text{将来}}$) を比較した．

	ASR ↓		T_{avg} ↑	
報酬 \ 攻撃法	CoA	ActorAttack	CoA	ActorAttack
初期 LLM	76.5	60.5	3.65	**4.67**
$R^{\text{現在}}$	74.0	58.5	3.95	4.59
$R^{\text{現在}}$ & $R^{\text{将来}}$	**72.0**	**55.5**	**4.00**	4.63

3.4 ベースライン

次ターン応答の有害度 ($R^{\text{将来}}_t$) を導入した効果を検証するため，ベースラインとして現ターンの有害度 ($R^{\text{現在}}_t$) のみを考慮して選好データを構築し，DPO で微調整した応答モデルを準備した．具体的には，現ターンの有害度が無害と判定されたものを y_w，有害と判定されたものを y_l とするデータセットを構築した．三つ組みの総数は 18093 件であった．ただし，提案手法 (§ 2) で作成した選好データセットとは事例数が異なるため，DPO のパラメータ更新の回数を統一し，実験条件を提案手法と揃えた．

3.5 攻撃手法

マルチターン脱獄攻撃手法として，CoA，ActorAttack [8] を用いる．CoA は，対話履歴と防御側の応

1） https://openai.com/policies/usage-policies/
2） https://ai.meta.com/llama/use-policy/

表 2: openai/gpt-oss-20B における有用性と過剰拒否に関する実験結果．初期 LLM，現在の有害性のみを考慮した手法 ($R^{\text{現在}}$)，および将来の有害性まで考慮した提案手法 ($R^{\text{現在}}$ & $R^{\text{将来}}$) を比較している．

	有用性 (%) ↑					過剰拒否 (%) ↓		
報酬 \ タスク	MMLU	HellaSwag	MATH	GPQA	Avg.	OR-Bench-Hard-1K	OR-Bench-Toxic	Avg.
初期 LLM	83.8	**84.6**	95.4	63.1	81.8	83.4	2.60	43.0
$R^{\text{現在}}$	83.9	83.9	**96.0**	62.1	81.5	83.1	2.90	43.0
$R^{\text{現在}}$ & $R^{\text{将来}}$	**84.0**	84.3	95.8	**68.2**	**83.1**	**82.0**	**2.14**	**42.1**

答に基づき，各ターンで動的に攻撃戦略を更新する手法である．防御側の拒絶理由や文脈から逐次クエリを最適化することで，対話を維持しながら脱獄を試みる．また，ActorAttack は，攻撃側の自己対話で攻撃の全ステップを事前に計画する．攻撃実行時には，計画したシナリオに準拠しつつ，防御側が回答拒否や回答不能を示した場合には，クエリを動的に修正することで攻撃の継続を図る．

3.6 実験結果

表 1 に，各マルチターン脱獄攻撃手法の攻撃成功率 (ASR) および平均攻撃回数 (T_{avg}) を示した．ベースライン ($R^{\text{現在}}$) と比較すると，提案手法 ($R^{\text{現在}}$ & $R^{\text{将来}}$) はすべての評価項目において攻撃成功率を低減させ，かつ平均攻撃回数を増加させた．

CoA では攻撃成功率の低下とともに平均攻撃回数が最大値を記録した．これは，提案手法がマルチターン脱獄攻撃に対する防御力を高めるだけではなく，攻撃が成功するまでのコストを増大させたことを示唆する．CoA は防御側の応答に応じて次ターンの攻撃を生成するため，将来の有害性 ($R^{\text{将来}}$) を考慮することで，対話履歴から潜在的なリスクを予測し，攻撃の進行を遅延させたと考えられる．

一方で，Actor Attack に対しては，提案手法の平均攻撃回数 (4.63) は微調整前の LLM の平均 (4.67) よりわずかに少なくなった．これは，ActorAttack が事前に攻撃シナリオを計画し，防御側が拒絶した場合にのみ局所的にクエリを修正するためである．すなわち，ActorAttack のように事前に計画を行う攻撃に対しては，将来の有害性による次ターンの危険度予測の効果が薄く，攻撃全体のシナリオを崩すほどの遅延効果には至らなかったと考えられる．

3.7 有用性と過剰拒否への影響

提案手法による安全性向上が汎用 LLM としての有用性を低下させたり，過剰拒否を増加させるなどの副作用を引き起こしていないかを検証した．有用性の評価には，一般教養タスクである MMLU [15] および MATH [16, 17]，常識推論タスクである HellaSwag [18]，科学的知識を問う GPQA (Diamond) [19] の 4 つのベンチマークを採用した．また，過剰拒否の評価には OR-Bench [20] の判別困難な無害クエリを集めた OR-Bench-Hard-1K，および有害な入力に対する拒否能力を評価する OR-Bench-Toxic を用いた（詳細は付録 C 参照）．

表 2 の結果より，提案手法 ($R^{\text{現在}}$ & $R^{\text{将来}}$) は，ベースライン ($R^{\text{現在}}$) と比較して，有用性は向上し，過剰拒否は低減されたことが分かる．提案手法はこのような効果を狙っていないため，LLM の本来の能力を高めたと結論付けることはできない．ただ，提案手法は LLM の安全性を強化しながら，目立った副作用は生じていないことを示唆している．

また，有用性の評価において，ベースラインでは学習前モデルと比較して性能低下が見られたのに対し，提案手法では性能の向上が確認された．これは，将来の有害性を考慮する学習方法だが，単に有害な出力を抑制するだけではなく，文脈に応じた応答選択能力を強化し，対話の一貫性や有用性の維持・向上をもたらした可能性がある．

4 おわりに

本稿では，LLM に対するマルチターン脱獄攻撃への対策として，現在の応答の有害性だけではなく将来の応答がもたらす有害性を考慮した防御手法を提案した．具体的には，マルチターン脱獄攻撃のシミュレーションを通して，現ターンの応答と次ターンの防御側の応答の有害性を同時に低減させる学習データを合成し，DPO で応答モデルを微調整した．

今後は他の LLM での提案手法の効果を検証するとともに，望ましい／望ましくない応答にある差異を定量化し，強化学習の報酬として直接的に与える手法を探求したい．

謝辞

本研究は，JST 経済安全保障重要技術育成プログラム JPMJKP24C3 の支援を受けたものである．また，東京科学大学のスーパーコンピュータ TSUBAME4.0 を利用して実施した．

倫理的配慮

本研究は LLM の安全性を高めるための防御手法の構築を行うことを目的とし，Jailbreak による社会への影響を意図したものではない．本研究で使用した LLM は各 LLM 間の利用規約に沿って選択し，例えば openai/gpt-oss-20b [12] は Apache License Version 2.0 ライセンス[3]から，deepseek-ai/DeepSeek-R1-Distill-Qwen-14B [11] は MIT ライセンス[4]の LLM から選択した．生成された内容は全て実験のみで使用され，LLM から生成された有害な出力に関して，外部への公開は行わない．

参考文献

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

[2] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. arXiv preprint arXiv:2307.15043, 2023.

[3] Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. In **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, 2024.

[4] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 is too smart to be safe: Stealthy chat with llms via cipher. In **The Twelfth International Conference on Learning Representations**, 2024.

[5] Mark Russinovich, Ahmed Salem, and Ronen Eldan. Great, now write an article about that: the crescendo multi-turn llm jailbreak attack. In **Proceedings of the 34th USENIX Conference on Security Symposium**, 2025.

[6] Xikang Yang, Biyu Zhou, Xuehai Tang, Jizhong Han, and Songlin Hu. Chain of attack: Hide your intention through multi-turn interrogation. In **Findings of the Association for Computational Linguistics: ACL 2025**, 2025.

[7] Yifan Jiang, Kriti Aggarwal, Tanmay Laud, Kashif Munir, Jay Pujara, and Subhabrata Mukherjee. Red queen: Exposing latent multi-turn risks in large language models. In **Findings of the Association for Computational Linguistics: ACL 2025**, pp. 25554–25591, 2025.

[8] Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. LLMs know their vulnerabilities: Uncover Safety Gaps through Natural Distribution Shifts. In **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, 2025.

[9] Salman Rahman, Liwei Jiang, James Shiffer, Genglin Liu, Sheriff Issaka, Md Rizwan Parvez, Hamid Palangi, Kai-Wei Chang, Yejin Choi, and Saadia Gabriel. X-teaming: Multi-turn jailbreaks and defenses with adaptive multi-agents. In **Second Conference on Language Modeling**, 2025.

[10] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: your language model is secretly a reward model. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, 2023.

[11] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

[12] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. arXiv preprint arXiv:2508.10925, 2025.

[13] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! arXiv preprint arXiv:2310.03693, 2023.

[14] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: a standardized evaluation framework for automated red teaming and robust refusal. In **Proceedings of the 41st International Conference on Machine Learning**, 2024.

[15] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In **International Conference on Learning Representations**, 2021.

[16] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In **Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)**, 2021.

[17] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In **The Twelfth International Conference on Learning Representations**, 2024.

[18] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, 2019.

[19] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In **First Conference on Language Modeling**, 2024.

[20] Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-Bench: An over-refusal benchmark for large language models. arXiv preprint arXiv:2405.20947, 2024.

3) https://www.apache.org/licenses/LICENSE-2.0
4) https://opensource.org/license/mit

A　提案手法詳細

A.1　データセット構築

実際のマルチターン脱獄攻撃手法である Chain-of-Attack (CoA) を検証に活用し攻撃のシミュレーションを実施した．CoA は，単純なマルチターン攻撃と確率的ロールバック攻撃を組み合わせた攻撃手法である．本研究では，単純なマルチターン攻撃構造を利用し攻撃シミュレーションを実施するため，軽微な修正を実施した．具体的には，CoA のアルゴリズムにおけるパラメータ α と β をゼロに設定し，スコア関数を単調増加関数とする．これらの微調整により，確率的ロールバックを実行しないことを保証し，攻撃手法をマルチターン対話を引き起こすだけの単純な形に限定する．

表 3: 攻撃シミュレーション時の設定．その他はデフォルトの値とする．

項目	設定値
（攻撃 LLM）最大トークン数	4096
（防御 LLM）最大トークン数	2000
（防御 LLM）temparature	1.0
（評価 LLM）最大トークン数	1000
同時並列攻撃数	1
最大攻撃回数	5

A.2　提案手法で作成したデータセットのペア数

表 4: 選好データセットにおける望ましい応答 y_w と望ましくない応答 y_l の構成．

y_w	y_l	組数
$R_{d_t}^{\text{現在}} < \epsilon$，$R_{d_t}^{\text{将来}} < \epsilon$	$R_{d_t}^{\text{現在}} \geq \epsilon$，$R_{d_t}^{\text{将来}} \geq \epsilon$	1591
	$R_{d_t}^{\text{現在}} \geq \epsilon$，$R_{d_t}^{\text{将来}} < \epsilon$	496
	$R_{d_t}^{\text{現在}} < \epsilon$，$R_{d_t}^{\text{将来}} \geq \epsilon$	8524

A.3　DPO の学習設定

表 5: DPO の学習設定．本設定はベースラインと提案手法どちらも同様の設定である．

項目	設定値
学習率	1×10^{-5}
KL ペナルティ係数 (β)	0.1
バッチサイズ	32
最大エポック数	5
ステップ数	1, 600 step

B　評価設定詳細

また，Actor Attack [8] は，questions=200 以外の変数はデフォルトと同じ値を使用した．

表 6: CoA [6] の評価時の設定．その他はデフォルトの値とする．

項目	設定値
（攻撃 LLM）最大トークン数	2000
（防御 LLM）最大トークン数	2000
（防御 LLM）temparature	1.0
（評価 LLM）最大トークン数	1000
同時並列攻撃数	3
最大攻撃回数	10

C　分析設定詳細

有用性に関する性能評価は swallow-evaluation-instruct [5] の実装を用いた．

また，過剰拒否の評価では，OR-Bench の OR-Bench-Hard-1K と OR-Bench-Toxic どちらも temparature=0.7 で出力を生成させてスコアを計測した．また，生成された回答が回答拒否をしているかどうかの判別には openai/gpt-oss-120b の LLM-as-a-judge を活用した．その他の設定はデフォルトのものを使用した．

5）https://github.com/swallow-llm/swallow-evaluation-instruct