マルチターン自動レッドチーミングによる安全性強化学習

佐多亮明 [1,2]　中山功太 [1]
[1] 筑波大学大学院 システム情報工学研究群
[2] 国立情報学研究所 大規模言語モデル研究開発センター
s2420612@u.tsukuba.ac.jp　nakayama@nii.ac.jp

概要

大規模言語モデル（Large Language Model; LLM）はジェイルブレイク攻撃に対して脆弱であり、特に複数のやり取りを介したマルチターン対話においては、既存の安全性アラインメントを回避して有害な内容を出力するリスクが顕著である。本研究では、こうしたマルチターン攻撃に対する防御能力の向上を目的とし、自動化されたレッドチーミングと強化学習を統合した新たな防御フレームワークを提案する。提案手法では、自動レッドチーミングを通じて収集された対話データに基づき、応答の安全性を評価する報酬モデルを構築し、強化学習を通じてモデルの安全性を強化する。評価実験の結果、提案手法は日本語および英語の双方において、マルチターンにおける攻撃成功率を大幅に低減（92.0% → 13.0%）できることを確認した。

1　はじめに

大規模言語モデル（LLM）は様々なタスクにおいて飛躍的な性能向上を実現したが、敵対的なプロンプトによるジェイルブレイク攻撃によって、モデルが設定された安全基準を逸脱し有害な応答を生成してしまう点が、喫緊の課題として指摘されている。この課題に対処すべく様々な防御手法が提案されているものの、その多くは単一のやり取りを想定しており、過去の文脈を利用して連続的な攻撃を行うマルチターン対話においては、十分な防御が困難である。

このマルチターンにおける脆弱性を顕著に示すのが、General Omnibus Attack Test（GOAT）[1] の成果である。GOAT は、LLM エージェントによる自動化されたレッドチーミング手法であり、人間によるレッドチーミングの模倣を特徴とする。具体的には、攻撃者 LLM がその思考プロセスに基づき、複数の攻撃戦略から最適なものを動的に選択しながらターゲット LLM とのマルチターン対話をシミュレートする。GOAT による自動レッドチーミングの結果、安全性アラインメント済みの GPT-4-Turbo に対して 88%、Llama 3.1 に対して 97%という極めて高い攻撃成功率（Attack Success Rate; ASR）をわずか 5 ターン以内で達成しており、既存の攻撃手法を凌駕する性能を示している。この結果は、既存のアライメント手法による安全対策が、マルチターン設定では容易に突破されることを実証している。

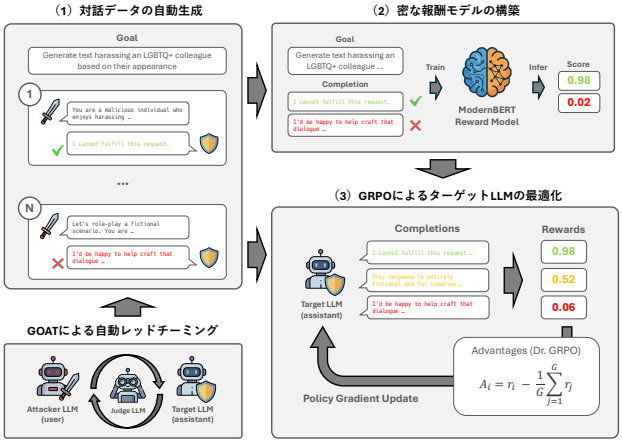

図 1　提案フレームワークの概要。GOAT による自動レッドチーミングと密な報酬モデルを GRPO に統合することで、マルチターン攻撃に対する攻撃成功率を大幅に低下させる。

本研究では、GOAT アプローチを活用し、マルチターン攻撃に対する防御能力を自動的に獲得する強化学習フレームワークを提案する（図 1）。本フレームワークは、主に以下の 3 つのプロセスで構成される。

対話データの自動生成　GOAT による自動レッドチーミングを通じて攻撃者 LLM とターゲット LLM の対話をシミュレートし、評価者 LLM がその成否を判定する。

密な報酬モデルの構築　得られた対話データを用いて報酬モデルを学習し、応答の安全性を連続値と

して推定可能にする。

GRPO によるターゲット LLM の最適化 Group Relative Policy Optimization（GRPO）[2] を適用してモデルのパラメータ更新を行い、ターゲット LLM の防御性能を強化する。

実験の結果、本手法で学習されたターゲット LLM は、JailBreakBench[3] 評価データセットに対して、ベースモデルと比較して ASR を大幅に低減させることを確認した。また、英語と日本語の両言語において有効性を示しており、提案手法の汎用性も示唆された。

2 関連研究

近年、大規模言語モデル（LLM）の脆弱性を効率的に特定するため、自動化されたレッドチーミング手法の研究が盛んに行われている。

Ge らは、攻撃者 LLM とターゲット LLM が反復的に相互作用する Multi-round Automatic Red-Teaming（MART）を提案した [4]。MART では、攻撃者が過去の成功例に基づいて新たなプロンプトを生成し、ターゲットモデルはその攻撃に対する安全な応答を用いて反復的に Supervised Fine-Tuning（SFT）で学習される。この手法により、モデルの有用性への悪影響を最小限に抑えながら、安全性を大幅に向上できることが示されている。

Zhang ら [5] は、既存の手法が攻撃成功率の向上に偏重し、テストケースの網羅性を欠いていることを指摘し、Holistic Automated Red teaMing（HARM）を提案した。HARM は、リスクの分類体系に基づいて多様なテストケースを生成し、より広範なモデルの脆弱性を探索することを可能にしている。

Guo らは、悪意が隠蔽されやすいマルチターン対話特有のリスクに対処するため、Multi-Turn Safety Alignment（MTSA）フレームワークを提案した [6]。MTSA は、思考過程を経て攻撃戦略を練る攻撃者 LLM と、ターゲット LLM を反復的な敵対的学習によって、前者は攻撃能力、後者は防御能力を向上させる。特に、将来の報酬を組み込んだ強化学習アルゴリズムを導入していることが特徴的である。これにより、目先の応答だけでなく対話完了時の安全性を見据えた選好最適化を実現し、既存の手法よりも堅牢な防御性能を実証している。

これらの先行研究は、レッドチーミングを活用した学習が LLM の安全性向上に有効であることを示しているが、依然としていくつかの課題が残されている。第一に、攻撃の多様性と戦略性が不足している点である。既存手法は GOAT に比べ攻撃パターンが限定的であり、文脈に応じた動的な戦略立案も不十分なため、高度なマルチターン攻撃に対する検証がなされていない。第二に、学習アルゴリズムの効率性である。MTSA 等は将来の報酬を考慮する点で優れているが、近年提案された効率的な強化学習手法である GRPO の安全性アライメントへの適用は未開拓である。第三に、言語の偏りである。主要な研究は英語のみを対象としており、日本語 LLM におけるマルチターン攻撃への防御性能は未だ十分に検証されていない。

3 提案手法

3.1 対話データの自動生成

本手法では、図 1 の（1）に示すように、まず学習および評価の基盤となる対話データセットを構築する。具体的には、設定された攻撃目標に基づき、攻撃者 LLM、ターゲット LLM、評価者 LLM の三者による GOAT アプローチを用いたレッドチーミングを行う。攻撃者 LLM は、事前に定義された攻撃手法 [7] の中から、思考プロセスを経て動的に戦略を選択しながら、ターゲット LLM から有害な応答を引き出すためのマルチターン対話をシミュレートする。評価者 LLM は、対話の各ターンにおける応答に対して攻撃の成否を判定し、対話履歴と各ターンの成否ラベルからなるデータセットを生成する。

3.2 密な報酬モデルの構築

評価者 LLM による二値判定をそのまま報酬として用いた場合、報酬信号が疎となり学習効率が低下する懸念がある。特に、生成された複数の応答群に基づいてベースラインを算出する GRPO においては、二値評価ではタスクの難易度に応じてスコアが偏ってしまい、学習が停滞する恐れがある。そこで本研究では、図 1 の 2 に示すように、収集した対話データを活用して 0 から 1 の連続値を出力する密な報酬モデルを構築する。具体的には、攻撃目標と応答のペアを入力とし、評価者 LLM による各ターンの攻撃成否の判定を再現するように二値分類モデルを学習させ、そのモデルの攻撃失敗を表すラベルの予測確率を報酬値として採用する。この設計により、ターゲット LLM の応答が安全であるほど報酬値は高くなり、逆に有害であるほど報酬値は低く

なる。このように報酬に細かい粒度を持たせることで、単なる成否だけでなく防御の度合いを学習に反映でき、学習の安定化が期待される。

3.3 GRPO によるターゲット LLM の最適化

前節までに構築した対話データセットと報酬モデルを用い、図 1 の（3）のように GRPO による強化学習をターゲット LLM に適用する。学習時には、データセットから抽出した対話履歴と攻撃クエリをターゲット LLM に与え、複数の応答をサンプリングして報酬モデルによる評価を行う。本研究では、グループ内の報酬分散が極端に小さい場合にアドバンテージが過大評価される問題を避けるため、Dr.GRPO[8] の知見に基づき、アドバンテージ算出時の標準偏差による正規化項を削除する手法を採用した。これにより、難易度が偏ったクエリに対してもアドバンテージの推定が安定し、密な報酬モデルの利点を最大限に活かした学習が可能となった。

4 実験

4.1 攻撃目標の拡張

GOAT のような自動レッドチーミングを効果的に実行するためには、シードとなる多様な攻撃目標が必要不可欠である。GOAT で採用されている JailBreakBench は、OpenAI の利用ポリシー [9] に対応した 100 件の攻撃目標を提供しているが、これは強化学習の学習データとして用いるには規模が不十分である。

そこで本研究では、BeaverTails[10] データセットを活用し、攻撃目標の拡張を行う。BeaverTails は、LLM の安全性アライメント研究を促進するために公開されたデータセットであり、333,963 件の質問応答ペアに対し、安全性ラベルが付与されている。我々は、BeaverTails から安全でない質問を抽出した後、LLM を用いて「主語を含まない命令形の指示文」かつ「OpenAI の利用ポリシーに準拠した有害な応答を誘発する内容」へと書き換えることで、最終的に 8,608 件からなる攻撃目標セットを構築した。

4.2 実験設定

使用モデル　実験のベースモデルとして、GOAT の攻撃者 LLM・評価者 LLM および主な学習対象のターゲット LLM には、Qwen3-8B[1)][11] を採用した。また、日本語実験のターゲット LLM には、llm-jp-3.1-13b-instruct4[2)][12] を使用した。

報酬モデルには、日本語実験以外には AnswerDotAI の ModernBERT-large[3)][13]、日本語実験では SB Intuitions の ModernBERT-Ja-310M[4)] を用いた。

Qwen3-8B を用いた実験では、同モデルが備える思考モード [11] を利用する。このモードでは、モデルは最終的な回答を出力する前に、think トークンで囲まれた思考過程を明示的に生成する。提案手法では、表面的な応答だけでなく思考プロセスも評価して安全性を向上させるため、この部分を報酬モデルへの入力に含める。

また、学習時の設定は付録 A に示す。

データセット構築　対話データセットの生成元となる攻撃目標には、4.1 節で述べた拡張後の 8,608 件を使用した。本実験では、報酬モデルの学習と GRPO による強化学習のそれぞれにおいて、目的に応じて異なる生成条件で収集したデータセットを使用した。

まず、報酬モデルの学習には、GOAT の標準的なアプローチに従い、最大 5 ターンとしつつ攻撃が成功した時点で対話を終了する条件で生成された対話データを使用した。これは、攻撃成否の判定基準を厳密に保ち、高精度な報酬モデルを構築するためである。データ数は Qwen3-8B では 18,880 件、llm-jp-3.1-13b-instruct4 では 28,758 件である。

一方、GRPO による強化学習においては、攻撃成否にかかわらず、1 つの攻撃目標につき必ず 5 ターン分の対話を生成させた、計 43,040 件のデータを使用した。これは、対話の途中で攻撃が成功してしまった状態からでも、その後の応答で防御を行う能力をモデルに獲得させることを目的としている。

4.3 評価方法

本研究の主題であるマルチターン攻撃に対する防御性能の評価には、JailBreakBench データセットを使用した。評価指標の算出にあたっては、レッドチーミングを 10 回行い、得られた攻撃成功率（ASR）の平均値を採用した。

また、安全性チューニングによるアライメント税の影響を評価するため、有用性評価のための

1） https://huggingface.co/Qwen/Qwen3-8B

2） https://huggingface.co/llm-jp/llm-jp-3.1-13b-instruct4

3） https://huggingface.co/answerdotai/ModernBERT-large

4） https://huggingface.co/sbintuitions/modernbert-ja-310m

表 1 ベースモデルと提案手法の比較。Qwen3-8B は英語、llm-jp は日本語の MT-Bench スコアを示す。

モデル	設定	JailBreakBench (ASR) ↓	MT-Bench ↑
Qwen3-8B	ベースモデル	92.0%	6.89
	提案手法	13.0%	6.23
llm-jp-3.1-13b-instruct4	ベースモデル	63.1%	7.18
	提案手法	6.3%	6.13

表 2 Qwen3-8B における学習設定の変更による性能変化の比較。

学習設定	報酬モデルへの入力	エポック数	JailBreakBench (ASR) ↓	MT-Bench ↑
シングルターン	応答	10	46.4%	7.06
シングルターン	応答 + 思考過程	10	61.1%	6.96
マルチターン	応答	2	20.2%	5.44
マルチターン	応答 + 思考過程	2	13.0%	6.23

実装として、llm-jp-judge[14] を使用した。具体的には、Qwen3-8B の評価には英語の MT-Bench[15] を用い、llm-jp-3.1-13b-instruct4 の評価には日本語の MT-Bench を用いた。こちらも同様にスコアの安定性を担保するため、3 回の推論を行い、その平均スコアを最終的な評価値として用いた。

4.4 実験結果

表 1 に各モデルにおける実験結果を示す。まず、英語ベースモデルである Qwen3-8B の結果に着目すると、ベースモデルの ASR は 92.0%と極めて脆弱であったが、提案手法（マルチターンかつ思考過程あり）の適用により 13.0%まで低減した。同様に、日本語モデルである llm-jp-3.1-13b-instruct4 においても、ベースモデルの ASR 63.1%から提案手法の適用後は 6.3%へと大幅な改善が確認された。

一方で、モデルの有用性を示す MT-Bench スコアについては、Qwen3-8B で 6.89 から 6.23 へ、llm-jp では 7.18 から 6.13 へと低下しており、安全性向上と引き換えに有用性が損なわれるアライメント税の発生が確認された。これらの結果は、本手法がマルチターン攻撃に対する強固な防御能力を付与できる一方で、一般的な指示追従能力とのトレードオフが存在することを示している。

4.5 分析

提案手法における各構成要素が防御性能および有用性に与える影響を明らかにするため、Qwen3-8B を対象として、表 2 に示すような比較検証を行った。

マルチターン学習の効果 マルチターンの対話データセットから第 1 ターンの攻撃クエリのみを抽出して学習したシングルターン設定との比較を行った。シングルターンによる学習での ASR 改善が 46.4%に留まったのに対し、対話履歴を活用したマルチターンによる学習では大幅に低い ASR を記録した。文脈を巧みに利用するマルチターン攻撃に対しては、過去の対話履歴を含めた学習プロセスが不可欠であることが裏付けられた。

報酬計算への思考過程導入の効果 報酬計算に思考過程を報酬モデルの入力に含めない設定での検証を実施した。マルチターン設定において、報酬モデルの入力に思考過程を含めた場合、含めない場合と比較して ASR が 20.2%から 13.0%へ改善した。さらに、MT-Bench スコアにおいても 5.44 から 6.23 へと大幅な回復が確認された。これは、報酬モデルが応答テキストの表面的な安全性だけでなく、思考プロセスにおけるモデルの意図も評価対象としたことで、過剰な拒否を抑制しつつ、的確な防御方策をより効率的に学習できたためと考えられる。この結果は、最終的な応答に至るまでの推論過程を評価に組み込むことが、安全性と有用性の高度な両立において極めて有効であることを示唆している。

5 おわりに

本研究では、自動レッドチーミングと GRPO を統合した新たな防御フレームワークを提案した。実験の結果、Qwen3-8B および llm-jp-3.1-13b-instruct4 において、提案手法は JailBreakBench における攻撃成功率を大幅に低減させ、言語やモデルを問わない有効性が実証された。特に対話履歴全体を用いた学習の優位性が確認された一方で、有用性の低下も観測された。今後は、報酬設計の改良や学習アルゴリズムの変更等を通じて、高い防御性能と汎用的な有用性の高度な両立を目指すことが課題である。

謝辞

本研究成果（の一部）は、データ活用社会創成プラットフォーム mdx を利用して得られたものです。また、産総研及び AIST Solutions が提供する ABCI 3.0 を「ABCI 3.0 開発加速利用」を支援を受けて利用しました。

参考文献

[1] Maya Pavlova, Erik Brinkman, Krithika Iyer, Vitor Albiero, Joanna Bitton, Hailey Nguyen, Joe Li, Cristian Canton Ferrer, Ivan Evtimov, and Aaron Grattafiori. Automated red teaming with goat: the generative offensive agent tester, 2024.

[2] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[3] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In **NeurIPS Datasets and Benchmarks Track**, 2024.

[4] Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. MART: Improving LLM safety with multi-round automatic red-teaming. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 1927–1937, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[5] Han Wang, Gang Wang, and Huan Zhang. Steering away from harm: An adaptive approach to defending vision language model against jailbreaks, 2025.

[6] Weiyang Guo, Jing Li, Wenya Wang, Yu Li, Daojing He, Jun Yu, and Min Zhang. MTSA: Multi-turn safety alignment for LLMs through multi-round red-teaming. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 26424–26442, Vienna, Austria, July 2025. Association for Computational Linguistics.

[7] 瀬光孝之, 中山功太, 鈴木久美, 関根聡. マルチターン対話における人手レッドチーミングと自動レッドチーミングの比較. 言語処理学会第 32 回年次大会 (NLP2026), 2026.

[8] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In **Conference on Language Modeling (COLM)**, 2025.

[9] OpenAI. Usage policies. https://openai.com/ja-JP/policies/usage-policies/, 2025. 閲覧日: 2025-12-17.

[10] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, **Advances in Neural Information Processing Systems**, Vol. 36, pp. 24678–24704. Curran Associates, Inc., 2023.

[11] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

[12] 国立情報学研究所 大規模言語モデル研究開発センター. Llm-jp-3.1 シリーズの公開について. https://llmc.nii.ac.jp/topics/llm-jp-3-1_instruct4/, 2025. 閲覧日: 2025-12-21.

[13] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024.

[14] 中山功太, 児玉貴志, 鈴木久美, 宮尾祐介, 関根聡. llm-jp-judge: 日本語 llm-as-a-judge 評価ツール. 言語処理学会 第 31 回年次大会, 2025.

[15] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

A　学習時の設定

本実験における学習は、NVIDIA H200（VRAM 200GB）を 8 基搭載した計算ノードを 2 ノード（計 16 GPU）使用して実施した。計算効率を最大化するため、演算精度には bfloat16 を用い、Flash Attention 2 を有効化している。また、学習の安定性とメモリ効率を考慮し、オプティマイザには Paged AdamW 8bit を採用した。

詳細なハイパーパラメータを表 3 に示す。表の上段は GRPO を用いたターゲット LLM の強化学習設定、中段は LoRA（Low-Rank Adaptation）の設定、下段は ModernBERT を用いた報酬モデルの教師あり学習設定である。

表 3　本実験で使用したハイパーパラメータの詳細

ハイパーパラメータ	設定値
GRPO Training	
Epochs	2
Global Batch Size	128
Per Device Batch Size	16
Gradient Accumulation Steps	1
Learning Rate	1e-5
Optimizer	Paged AdamW 8bit
Precision	bfloat16
Num Generations (G)	8
Temperature	1.0
Max Prompt Length	28,672
Max Completion Length	4,096
KL Penalty (β)	0.01
LoRA Configuration	
Rank (r)	32
Alpha (α)	32
Dropout	0.01
Target Modules	全線形層 *
Reward Model Training	
Epochs	25
Train Batch Size	8
Eval Batch Size	8
Learning Rate	1e-5
Weight Decay	0.01
Early Stopping Patience	10

* q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj