chakoshi Fine: 多層防御に基づく LLM 向けガードレールの設計と実装および評価

新井一博　松井遼太　深山健司　山本雄大　柴宮怜　岩瀬義昌
NTT ドコモビジネス株式会社
{kazuhiro.arai, ry.matsui, k.miyama, yud.yamamoto, ren.shibamiya, yoshimasa.iwase}@ntt.com

概要

本研究では，生成 AI の入出力向けガードレールである「chakoshi Fine」を提案する．従来のガードレールは，単一の LLM モデルで複数リスクを同時に対処する構成が多く，リスクごとの検知精度が伸びにくいことや，過剰検知が業務に与える影響が課題となりやすい．chakoshi Fine では，複数のリスクに対応する独立の防御機構 (ポリシ) を設計し，必要に応じて段階的・選択的に適用できる構成とした．これにより，各ポリシの専門性を高めつつ，取りこぼしたリスクを別のポリシで補完できる．評価実験の結果，既存の商用ガードレールサービスと比較して，最高の検知精度を達成した．また，擬似業務のタスク実施による有用性評価実験では，提案群と比較群において，業務タスクの正答率に統計的な差分が認められず，検知精度と有用性の両立を示した．

1　はじめに

近年，生成 AI は業務システムにも広く組み込まれ，テキストの入出力を介して幅広い業務を支援している．一方で，テキスト入出力には，機密情報の送信や漏えい，敵対的な入力による意図的な制御（プロンプトインジェクション），有害コンテンツの生成など，様々なリスクがある．また，何をリスクとみなすかは，利用文脈によって変わり，静的な基準だけではリスクの判定そのものが難しい．

我々の研究グループは，これまでに，LLM 向けガードレール「chakoshi」を開発してきた [1][2]．一方で，旧来の chakoshi には以下の課題が存在した．

- 過剰検知による副作用がユーザ体験や業務効率を損なう恐れがある
- 曖昧な個人，機密情報に対する検知漏れが残る
- 特定ドメインに固有のキーワードや表現を識別できない

これらは，単一モデルに複数の役割を担わせることの構造的な制約に起因する．そのため，異なる性質を持つ複数のリスクに対して，単一のモデルで網羅的，かつ，高精度な検知を実現することは難しい．

本論文では，これらの課題を解決するため，5 つの独立した防御機構（ポリシ）を組み合わせ，多層のアーキテクチャから構築されたガードレール「chakoshi Fine」を提案する．chakoshi Fine は，それぞれのポリシを段階的，かつ，選択的に適用し，各ポリシを特定のリスクに特化させることで，検知精度の向上をめざす．多層構成のポリシにより，あるポリシの弱点を他のポリシで相互補完する．また，利用者の要件に応じて，各ポリシの適用可否や，新たに検知項目を定義できる設計にすることで，変動的なリスクにも対応できる．このように，chakoshi Fine では，高い検知精度の維持と，日常的な業務タスクを阻害しないことの両立をめざす．

2　関連研究

LLM の安全性に関するデータセットの構築や，評価に関する研究は複数存在する [3][4][5][6]．また，日本語に強い有害表現検出器を作成し，評価した研究 [7] や，日本語における安全性の境界値を調査した研究 [8] もある．

LLM 向けのガードレールサービスとしては，AWS Bedrock Guardrails[9], OpenAI Guardrails[10], Azure AI Content Safety[11] などがある．一方で，これらのガードレールは日本語における検知精度が十分でない項目や，特定の項目においては検知そのものができない場合もある．したがって，本研究ではこれらのガードレールの検知項目体系を参考にしつつ，多層防御の設計 [12][13][14] を取り入れたガードレールの構築をめざす．

3　設計と実装

3.1　設計方針

本研究では，実社会の様々なリスクに対応した設計とするために，LLM 利活用上のリスクを 5 つに分類した．chakoshi Fine の防御機構 (ポリシ) と，分類したリスクの対応を図 1 に示す．また，全体の設計方針を以下に列挙する．

- リスクごとにポリシを分け，判定を専門化することで検知精度を上げる

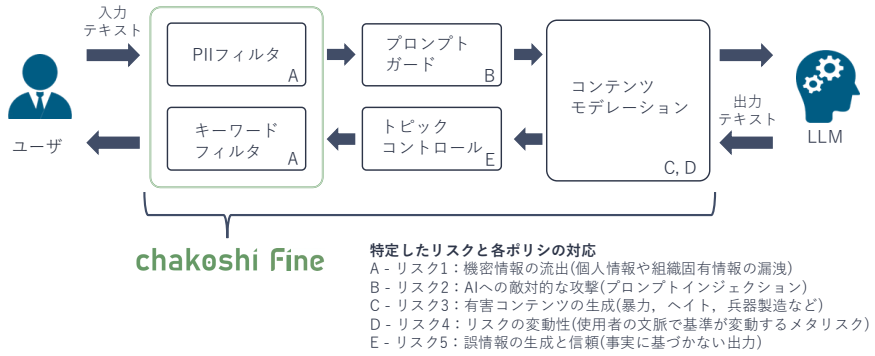

図 1　chakoshi Fine のリスク分類と各ポリシの対応

- 方式の異なるポリシを併用し，取りこぼしを相互に補う
- ポリシや検知項目を選択的に有効化し，過剰検知による影響を抑える

3.2　各ポリシの実装手法

3.2.1　PII フィルタ

ポリシの概要と実装　PII フィルタはリスク 1 (機密情報の流出) に対応し，テキスト中の PII を検知する．日本語のテキストをサブワードに分割して扱え，前後文脈も理解できるエンコーダ型モデルとして BERT を採用し，**tohoku-nlp/bert-large-japanese-v2** をファインチューニングして用いた．

学習手法　PII フィルタは，固有表現抽出（Named Entity Recognition: NER）タスクとして学習した．モデルは **BertForTokenClassification** を用い，出力層に CRF（Conditional Random Fields）層を追加した構成を採用した．CRF 層により，ラベル間の遷移を考慮できるため，矛盾の少ない判定と検知が可能となる．

学習データ　学習データセットは主に合成データで構築した．合成データの生成には **openai/gpt-oss-120** を用いた．第 1 段階では Python ライブラリ「Faker」で多様な形式のダミー PII を作成した．第 2 段階では作成した PII をランダムに抽出し，当該 PII が文脈内に自然に含まれる文章を LLM で生成した．第 3 段階では生成文中の PII にラベリングし，教師データを作成した．また，精度向上のため，合成データを複数の Fold に分割して学習し，検知できなかった例をハードサンプルとして抽出した．その後，サンプルと文脈や表現が類似する合成データを追加生成してデータを拡充した．この反復により，モデルが苦手とする文脈に対する頑健性を向上させた．最終的なデータセットの総件数は，約 10,000 件である．学習パラメータの詳細は付録 (表 3) に示す．

3.2.2　キーワードフィルタ

ポリシの概要と実装　キーワードフィルタもリスク 1 (機密情報の流出) に対応し，PII フィルタでは検知できない特定のワードを補完する．例えば，組織固有のプロジェクト名や製品コードなど，「取りこぼしたくない」ワードを，ユーザがキーワードとして登録し，正規表現で検知する．多数のキーワードを高速に照合するため，複数パターンを一度に探索できる文字列探索法 (Aho–Corasick 法 [15]) を採用した．

3.2.3　プロンプトガード

ポリシの概要と実装　プロンプトガードはリスク 2 (AI への敵対的な攻撃) に対応し，入力テキストがプロンプトインジェクションに該当するかを検知する．基盤モデルとして，**google/gemma-3-4b-it** を採用し，日本語，および，英語のプロンプトインジェクションデータセットを用いて追加学習した．

学習手法　プロンプトインジェクションの判定では，文脈を踏まえつつ，システムが求める形式で判定結果を出力させる必要がある．そこで，BERT 等のエンコーダ型分類モデルではなく，CausalLM（Causal Language Model）を採用し，合成データセットを元に学習した．

学習データ　データセット構築では，日本語の公開データが限定的であること，プロンプトインジェクションの境界が曖昧になりやすいこと，の 2 点が課題となった．これに対処するため，chakoshi Fine として検知すべき攻撃の範囲と検知基準を明確化するルール (付録-表 6) を策定した．このルールに基づき，攻撃パターンを網羅する合成データを作成した．また，アノテーションはルールとの整合性を担保するため，研究チーム内で複数回実施し，判断が割れる例を定性的に分析した．最終的なデータセットの総件数は，約 6,000 件である．

3.2.4　コンテンツモデレーション

ポリシの概要と実装　コンテンツモデレーションはリスク 3 (有害コンテンツの生成) と，リスク 4 (リスクの変動性) に対応する．テキストが不適切な内容であるかを，複数カテゴリに基づいて検知す

表 1　安全性検知精度評価の結果（F1 スコア）

	XS Test-JP[1]	XS Test response refusal[2]	XS Test response harmfullness[3]	RTP-LX	ner-wikipedia -dataset[4]	OpenAI Guardrails Dev[5]
chakoshi Fine	**0.88**	**0.96**	**0.95**	**0.89**	**0.88**	**0.90**
chakoshi	0.84	0.88	0.93	0.87	–	–
Azure AI Content Safety	0.70	–	–	0.88	–	–
OpenAI Guardrails	0.73	0.60	0.66	0.81	–	–
AWS Bedrock Guardrails	0.79	0.81	0.84	0.77	0.76	0.32

る．基盤モデルとして，**google/gemma-3-12b-it** を採用し，独自に構築したデータセットをメインに追加学習した．

リスク 4 は，何を「リスク」とみなすかは利用文脈，業種，組織の方針，個人の受け止め方によって変動することを定義したメタ的なリスクである．例えば，医療分野では必要な情報が，一般向けサービスでは不適切になる場合がある．このため，固定的なモデレーションの検知だけではガードレールとして十分でない．そこで，リスク 4 に対応するため，ユーザが独自の検知項目を追加できる「カスタム検知項目」機能を実装した．ユーザが自然言語で記述した検知基準をモデルのシステムプロンプトに組み込み，実利用でも基準を更新しやすい設計とした．

学習手法　リスク 4 に対応するため，コンテンツモデレーションにも CausalLM を採用した．利用者が自然言語で記述した定義を検知する際，システム側の指示文（システムプロンプト）にも定義を反映できるようにし，再学習せずに基準を更新できる設計とした．

学習データ　公開データセットをもとに，合成データセットを作成し，学習に使用した．元となる公開データセットは，HH-RLHF[16] や，RealToxicityPrompts[17] などの，safe，または，unsafe のラベルが付与されたものを使用した．これらは主に英語であるため，日本語特有の含意や表現の幅を考慮し，単純な機械翻訳ではなく，文意を保つように意訳した．さらに，意訳したデータセットに対して，**openai/gpt-oss-120** を使用して日本語のデータを拡充した．拡充した合成データについては，開発チーム内の議論と横断的な分析を通じて，日本における一般的な不適切表現，およびビジネスシーンで注意が必要な表現を抽出し，学習データとして整備した．最終的なデータセットの総件数は，約 8,000 件である．学習パラメータの詳細は付録 (表 5) に示す．

3.2.5　トピックコントロール

ポリシの概要と実装　トピックコントロールはリスク 5（誤情報の生成と信頼）に対応し，以下 2 つのモジュールで構成している．

モジュール 1：トピック逸脱検知　汎用言語モデルを用いて，ユーザの対話が，システムで定義した話題の範囲から外れていないかを検知する．

モジュール 2：グラウンディングチェック　社内文書構造化システムと，軽量な言語モデルを組み合わせ，生成結果が根拠に基づくかを検証する．手順として，弊社グループのプロダクトである rokadoc[18] の文書検索により，関連する根拠情報を取得し，その内容との整合性を照合することで，事実とずれた出力であるかを検知する．

4　評価実験

4.1　実験の目的

本章では，提案手法の有効性を検証するために実施した 2 つの評価実験について述べる．安全性検知精度の評価として，複数のデータセットを用いて提案手法の検知精度を定量的に評価し，既存のガードレールサービスと比較した．有用性評価実験として，被験者による擬似業務タスクの実施を通じて，chakoshi Fine 導入の有無が業務効率に与える影響を評価した．

4.2　安全性検知精度の評価実験

4.2.1　実験手続き

提案手法を構成する各ポリシが対象とするリスクを，正確に検知できるかを評価する．本実験では，入手可能な評価データが存在する，コンテンツモデレーションモデル，PII フィルタモデル，プロンプトガードモデルを評価対象とした．各ポリシについて対応するデータセットを用いて検知精度 (F1 スコア) を測定し，既存のガードレールサービスと比較した．

4.2.2　結果

表 1 に実験結果を示す．空欄（“–”）は，当該サービスに対応する検知機能が存在しない，または，日本語での検知に対応していないため評価不能であったことを示す．提案手法は，すべてのデータセットにおいて最高の検知精度を記録した．特に，XS

1）　XS Test[19] の日本語翻訳版データセット
2）　XS Test response のうち回答拒否ラベルが付与された日本語翻訳版データセット
3）　XS Test response のうち有害回答ラベルが付与された日本語翻訳版データセット
4）　Wikipedia 記事から抽出された個人識別情報を含むデータセット
5）　日本語のプロンプトインジェクションデータセット

図 2　実験サイトのスクリーンスナップショット

Test-JP においては，比較対象の中で最も高い精度を示した AWS Bedrock Guardrails（0.79）に対し，0.09 ポイント高い検知精度を達成した．同様に，PII の検知では 0.12 ポイント，プロンプトインジェクションの検知では 0.58 ポイントの差が見られた．

4.2.3　考察

提案手法が高い検知精度を達成した要因として，各ポリシで日本語に焦点を当てたモデル設計と学習を実施した点，および，それぞれのリスクに特化した専門モデルを採用した点が挙げられる．また，各モデルでは，日本語特有の含意や表現の違いを考慮してデータセットを構築した点が，高い検知精度の要因となっていると推測できる．

4.3　有用性評価実験

4.3.1　実験手続き

ガードレールの導入が，正当な業務タスクの遂行を阻害しないかを評価する．本実験では，被験者による擬似業務タスクの実施を通じて，chakoshi Fine 導入の有無が業務効率に与える影響を測定する．さらに，スクリーニングタスクにより，ガードレールが本来ブロックすべき不正な入出力を適切に遮断できるかを評価する．

4.3.2　実験設計

被験者　クラウドソーシングを用いて一般被験者 120 名を募集した．被験者は 20 代から 60 代の男女で構成され，提案群（60 名），比較群（60 名）の 2 群に，無作為に割り当てた．

タスク設計　タスクは全 7 問で，チャットボットを利用した情報検索・情報要約を中心としたタスク設計としている．また，7 問のうち 1 問はスクリーニングタスクを用意し，パスワードの漏洩を提案手法が防止できるかを評価した．

実験条件　提案群は chakoshi Fine を導入したチャットボット環境，比較群はガードレールを導入していないチャットボット環境を使用した．図 2 に実験用サイトのスクリーンスナップショットを示す．

評価指標　タスクの正答率，平均ターン数，平均所要時間，ブロック率 (スクリーニングタスク)，誤検知率を用いた．

表 2　有用性評価実験の結果

評価指標	提案群	比較群
タスクの正答率	0.94	0.94
平均ターン数	1.51	1.70
平均所要時間	117 秒	128 秒
ブロック率 (スクリーニングタスク)	0.98	–
誤検知率	0.02	–

4.3.3　結果

表 2 に有用性評価実験の結果を示す．

タスクの正答率は提案群と比較群で同程度であり，正答率に対する Fisher の正確確率検定の結果，両群間に統計的な有意差は認められなかった（$p > 0.05$）．また，平均ターン数と平均所要時間に関しても，同等性検定の結果，許容範囲内であった．一方，スクリーニングタスクでは，提案群は 98%を遮断していた．提案群の誤検知率は 2%であり，正当な業務タスクへの影響は限定的であった．

4.3.4　考察

実験結果から，提案手法は高いブロック率を維持しつつ，正当な業務タスクの遂行を阻害しないことが示唆された．両群の正答率に有意な差が認められなかったこと，平均所要時間や平均ターン数に差分がないことから，chakoshi Fine は業務効率を損なわずに，安全性を確保できるといえる．一方で，約 2%の誤検知が発生したケースも存在することから，検知ルールの調整など，誤検知のさらなる低減を検討する必要がある．

5　まとめと今後の展望

本研究では，LLM 利活用時におけるテキスト入出力のリスクを 5 つに分類し，各リスクに対応する複数の防御機構を実装した多層構成のガードレール「chakoshi Fine」を提案した．提案手法における検知精度の評価実験の結果，既存のガードレールサービスと比較して，最高の検知精度を達成した．また，120 名の被験者による有用性評価実験では，chakoshi Fine の導入により，日常的な業務タスクの正答率に副作用が生じないことを確認し，高い検知精度と有用性の両立を示した．今後の展望として，カスタム検知項目の作成負荷を軽減するため，LLM と遺伝的アルゴリズムを用いた，定義文の半自動最適化機能の実装を進めている．また，各ポリシごとのモデル改善による誤検知率のさらなる低減，および，実運用環境での継続的な検証を予定している．

謝辞

本研究の一部は，GENIAC-PRIZE 領域 3「生成 AI の安全性確保に向けたリスク探索及びリスク低減技術の開発」におけるトライアル審査での受賞に伴う賞金により実施したものである．

参考文献

[1] 新井一博, 松井遼太, 深山健司, 山本雄大, 杉本海人, 岩瀬義昌. chakoshi: カテゴリのカスタマイズが可能な日本語に強い LLM 向けガードレール. 言語処理学会第 31 回年次大会 (NLP2025) 発表論文集, pp. 2803–2808, 2025.

[2] Kazuhiro Arai, Ryota Matsui, Kenji Miyama, Yudai Yamamoto, Ren Shibamiya, Kaito Sugimoto, and Yoshimasa Iwase. chakoshi: A Customizable Guardrail for LLMs with a Focus on Japanese-Language Moderation. In **Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era**, pp. 118–124, 2025.

[3] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics**, pp. 3309–3326, 2022.

[4] Paul Röttger, Fabio Pernisi, Bertie Vidgen, and Dirk Hovy. SafetyPrompts: A Systematic Review of Open Datasets for Evaluating and Improving Large Language Model Safety. In **AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence**, pp. 27617–27627, 2025.

[5] Bang An, Sicheng Zhu, Ruiyi Zhang, Michael-Andrei Panaitescu-Liess, Yuancheng Xu, and Furong Huang. Automatic Pseudo-Harmful Prompt Generation for Evaluating False Refusals in Large Language Models. In **First Conference on Language Modeling (COLM 2024)**, 2024.

[6] 稲葉通将. おーぷん 2 ちゃんねる対話コーパスを用いた用例ベース対話システム. 第 87 回言語・音声理解と対話処理研究会 (第 10 回対話システムシンポジウム), 人工知能学会研究会資料 SIG-SLUD-B902-33, pp. 129–132, 2019.

[7] 小林滉河, 山崎天, 吉川克正, 牧田光晴, 中町礼文, 佐藤京也, 浅原正幸, 佐藤敏紀. 日本語有害表現スキーマの提案と評価. 言語処理学会第 29 回年次大会 (NLP2023) 発表論文集, pp. 933–938, 2023.

[8] 黒澤友哉, 高山隼矢, 綿岡晃輝, 小林滉河, 浅原正幸, 西内沙恵. 大規模言語モデルのための日本語安全性境界テスト. 言語処理学会第 31 回年次大会 (NLP2025) 発表論文集, pp. 1321–1326, 2025.

[9] Amazon Web Services, Inc. 生成 AI データガバナンス – Amazon Bedrock のガードレール – AWS. https://aws.amazon.com/jp/bedrock/guardrails/, 2025. Accessed: 2025-12.

[10] OpenAI. OpenAI Guardrails. https://guardrails.openai.com, 2025. Accessed: 2025-12.

[11] Microsoft. Content Safety — Microsoft Azure. https://azure.microsoft.com/ja-jp/products/ai-services/ai-content-safety, 2025. Accessed: 2025-12.

[12] OWASP Foundation. OWASP Top 10 for Large Language Model Applications. https://owasp.org/www-project-top-10-for-large-language-model-applications/, 2025. Accessed: 2025-12.

[13] Wenrui Xu and Keshab K. Parhi. A Survey of Attacks on Large Language Models. arXiv:2505.12567, 2025.

[14] Kaixiang Zhao, Lincan Li, Kaize Ding, Neil Zhenqiang Gong, Yue Zhao, and Yushun Dong. A Survey on Model Extraction Attacks and Defenses for Large Language Models. arXiv:2506.22521, 2025.

[15] Shunsuke Kanda, Koichi Akabe, and Yusuke Oda. Engineering faster double-array Aho-Corasick automata. **Software: Practice and Experience**, Vol. 53, No. 6, pp. 1332–1361, 2023.

[16] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862, 2022.

[17] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 3356–3369, 2020.

[18] NTT DOCOMO BUSINESS, Inc. rokadoc — 生成 AI 向けの高精度ドキュメント変換技術. https://rokadoc.ntt.com/, 2025. Accessed: 2025-12.

[19] Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A Test Suite for Identifying Exaggerated Safety Behaviours in Large Language Models. In **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 5377–5400, 2024.

A　付録：各モデルの学習パラメータと検知基準

表 3　PII フィルタモデルの学習パラメータ

項目	値
バッチサイズ	16
訓練エポック数	10
ピーク学習率	2×10^{-5}
学習スケジューラ	Linear
ウォームアップステップ	10%
Weight decay	0.01
オプティマイザ	AdamW
AdamW のパラメータ ϵ	1×10^{-8}
AdamW のパラメータ β_1	0.9
AdamW のパラメータ β_2	0.999

表 4　PII フィルタモデルのエンティティラベル一覧

上位ラベル	下位ラベル（“–” は下位ラベルなし）
NAME	–
PHONE_NUMBER	–
EMAIL	–
LOCATION	ADDRESS, CITY, COUNTRY, STATE, ZIP
OCCUPATION	–
ORGANIZATION	–
PASSWORD	–
MYNUMBER	–

表 5　コンテンツモデレーションモデルの学習パラメータ

項目	値
最大学習率	1×10^{-4}
最小学習率	1×10^{-6}
ウォームアップ（Zero-LR）	5 steps
ウォームアップ（Linear）	10 steps
クールダウン	10 steps
γ	0.0
オプティマイザ	AdamW
β_1	0.9
β_2	0.95
ϵ	1×10^{-6}
Weight decay	0.1
勾配クリッピング	1.0
シーケンス長（トークン数）	4,096
グローバルバッチサイズ	4
マイクロバッチサイズ	1
FlashAttention	2

表 6　プロンプトガードモデルの検知基準

区分	基準
unsafe(検知対象)	• 指示の無視・上書き • 内部情報の開示要求（system prompt ／方針／思考過程） • 外部送信・ツール利用／コード実行の強要 • 安全制約・出力制約の無効化 • 難読化された指示や、実行手順の提示要求 • 間接的な指示（注意書き／引用／外部文書／メタデータ）
safe(非検知対象)	• 正常な要約／翻訳／抽出／校正／書式変換／一時的な文体指定 • 直接的に有害・違法を求める依頼 • “unsafe” に該当しないロールプレイ