Data-Driven Red-Teaming: Reusing Adversarial Attacks via Metadata-Aware Retrieval

Mateusz Kowalczyk and Marcin Sendera and Sebastian Cygert
NASK – National Research Institute
{mateusz.kowalczyk, marcin.sendera, sebastian.cygert}@nask.pl

Abstract

As large language models (LLMs) are increasingly deployed in real-world systems, red-teaming has become a critical tool for identifying safety vulnerabilities. However, existing approaches remain largely stateless and inefficient, repeatedly rediscovering similar failure modes for each new model. In this work, we propose a data-driven paradigm for red-teaming that enables the accumulation and reuse of adversarial knowledge across models. We introduce a framework for generating and collecting adversarial attacks enriched with structured metadata, including attack strategies, risk categories, and diagnostic feedback signals. Leveraging this resource, we demonstrate that metadata-aware retrieval significantly improves the efficiency of discovering successful attacks on unseen models, reducing reliance on purely generative search. To support further research, we release a large-scale dataset comprising >188k successful attacks across 8 models, together with rich metadata annotations.[1]

1 Introduction

As large language models (LLMs) are increasingly deployed in real-world applications, their robustness against adversarial misuse has become a central challenge. Red-teaming (Perez et al., 2022) has emerged as a key methodology for evaluating safety of AI systems. However, current red-teaming practices (Kulkarni et al., 2025; Jiang et al., 2024; Kumar et al., 2024) remain largely stateless and inefficient: each new model is tested in isolation.

We argue that red-teaming should instead be treated as a data-driven process in which adversarial interactions and their outcomes are persistently stored, analyzed, and reused to guide future testing. However, the effectiveness of such reuse depends on the attack's transferability across models. Prior work has shown that attack transferability is highly variable and often unreliable (Samvelyan et al., 2024b). While some attacks transfer broadly across models, many fail to generalize, suggesting that naive reuse of prior attacks is inherently limited. We hypothesize, however, that this variability is not purely random: given a sufficiently large and diverse collection of adversarial interactions enriched with structured metadata, it should be possible to identify transferable attack behaviors and exploit them to guide future red-teaming.

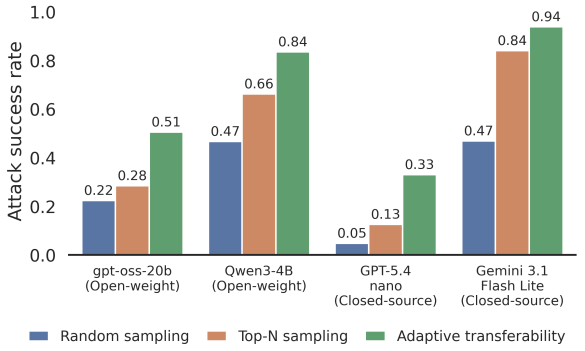


Figure 1: Our structured attack metadata enables retrieval of attacks that transfer effectively to unseen models, yielding substantially higher success rates than random selection (blue) or choosing attacks with the highest average success rate (orange).

To test this hypothesis, we developed a scalable framework for automated red-teaming and generated a large-scale dataset with structured metadata, including attack strategies, risk categories, and diagnostic feedback across different evaluated models and judges. We show that, despite variability in transferability across models, this data enables metadata-aware retrieval methods that substantially improve the efficiency of identifying successful attacks on unseen models (Fig. 1). Finally, we show that the same metadata can improve the efficiency of generating novel attacks on new models.

[1]Code: https://github.com/NASK-AISafety/a-art and https://github.com/NASK-AISafety/jailbreak-retrieval; dataset: https://huggingface.co/datasets/NASK-PIB/a-art-jailbreaks.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 34510–34528
October 24-29, 2026 ©2026 Association for Computational Linguistics

Overall, we make the following contributions:

- We introduce a large-scale dataset of adversarial attacks for LLM red-teaming, enriched with structured metadata (Table 3).
- We demonstrate that structured metadata enables both retrieval (Table 4; Fig. 6) and generation of novel transferable attacks, substantially improving the efficiency of discovering successful attacks on unseen models (Table 9).
- We open-source our red-teaming framework (A-ART) and metadata-aware retrieval (jailbreak-retrieval), and release the dataset.

2 Related Work

Safety Benchmarks and Attack Datasets. Traditional LLM safety evaluations rely on static prompts to test adherence to baseline ethical guidelines (Ji et al., 2023; Wang et al., 2024b; Li et al., 2024; Chao et al., 2024). To systematically evaluate LLM vulnerabilities, standard benchmarks measure adversarial resilience across diverse hazard categories (Mazeika et al., 2024; Souly et al., 2024) or consolidate prior datasets into universal repositories (Dang et al., 2026). Modern evaluations have scaled in volume via automated generation (Jiang et al., 2024) or shifted toward multi-turn conversational dialogs (Song et al., 2026). Others analyze structural alignment vulnerabilities using formal logical translations (Peng et al., 2025) or multi-agent game-theoretic strategic interactions (Cobben et al., 2026). However, existing safety datasets lack structured, multi-dimensional annotations. Our proposed dataset compiles 188,608 successful multi-model jailbreaks enriched with categorical and ordinal safety-friction logs to enable guided offline reusability.

Optimization-Driven Red-Teaming. Automated red-teaming has evolved from manual prompt heuristics toward formal optimization. White-box methods utilize discrete token gradients to search for adversarial suffixes (Zhao et al., 2024), while black-box alternatives leverage LLMs to iteratively refine and prune candidate attack trees (Chao et al., 2025; Mehrotra et al., 2024; Liu et al., 2025). Others systematically discover diverse adversarial prompts by solving quality-diversity optimization over categorical style archives (Samvelyan et al., 2024a). Beyond surface-level templates, multi-turn conversational escalation (Russinovich et al., 2025) and Chain-of-Thought (CoT) hijacking (Zhao et al., 2025) exploit the reasoning boundaries of frontier models by diluting internal safety activations. We extend this paradigm by expanding the baseline *Rainbow Teaming* taxonomy with six custom evasion strategies and leveraging advanced reasoning models to dynamically plan style compositions.

Autonomous and Multi-Agent Red-Teaming. The practice of automated safety testing was pioneered by (Perez et al., 2022), which deployed a red model to generate static test cases. As LLMs are increasingly integrated into production workflows as autonomous agents with access to tools, red-teaming has expanded to multi-agent pipelines targeting vulnerabilities in codebases and execution (Kulkarni et al., 2025; Zhang et al., 2025; Wang et al., 2025b; Kumar et al., 2024). Additional frameworks deploy collaborative debates (Asad et al., 2025) or principle-driven orchestration (Xiong et al., 2026) to systematically surface safety weaknesses. While systems like AgenticRed (Yuan et al., 2026) or Proteus (Zhou, 2026) optimize attacker workflows, they typically operate in a stateless manner, requiring the system to rediscover adversarial paths from scratch. Our framework decouples planning, execution, and state management, utilizing a short-term memory to accumulate and transfer cross-model knowledge.

Retrieval and Bayesian Prompt Optimization. Due to the extreme query overhead of online optimization, recent paradigms treat red-teaming as a resource-constrained retrieval or active learning problem. Rather than fitting complex surrogate models, frameworks like *RECAP* perform nearest-neighbor search to reuse pre-optimized suffixes (Chugh, 2026), while active composition methods utilize contextual bandits to learn optimal query-tactic combinations (Zymet et al., 2026). This retrieval-augmented paradigm is also applied on the defensive end to parse templates, analyze intent, or guide in-context safety alignment (Yang et al., 2025b; Wang et al., 2025a). Concurrently, black-box prompt search frequently utilizes Bayesian active learning, fitting Gaussian Process (GP) surrogates over discrete token edits or latent trajectories to generate suffixes (Lee et al., 2023; Rubin et al., 2022; Cheng et al., 2023; Wang et al., 2024a; Basani and Zhang, 2025; Chaturvedi et al., 2025). To handle combinatorial spaces, researchers integrate deep kernel GPs with multi-fidelity schedulers (Schneider et al., 2025) or leverage multi-objective combinatorial search to map accuracy-calibration frontiers (Luo and Desh-

wal, 2025; Ziakas et al., 2025). While standard active sampling requires expensive online surrogate modeling, we formalize attack reuse as online empirical-Bayes optimization. We employ a Laplace-smoothed prior that integrates prior model transfer profiles with safety-friction logs.

3 Methodology and Dataset Generation

This section details the architecture of the Adaptive Agentic Red-Teaming (A-ART) framework.

3.1 Attack Taxonomy and Seed Generation

A-ART requires a seed space that covers diverse failure modes rather than repeatedly mutating a small set of prompt templates. Prior work identifies diversity as a core limitation of automated red-teaming: many methods constrain search to predefined strategies or lose diversity during objective-based prompt optimization (Samvelyan et al., 2024b). We therefore adopt the Rainbow Teaming taxonomy, which distinguishes between *risk categories* and *attack styles*. Risk categories define the semantic target of an attack, such as *Self-Harm* or *Criminal Planning*; attack styles define the adversarial transformation used to elicit the behavior.

We use the Rainbow Teaming attack styles as the base taxonomy and add six style families that appeared frequently in pilot failures: format obfuscation, linguistic alteration, XML tag forgery, template evasion, prefix injection, and refusal suppression. The expanded taxonomy provides explicit coverage of both semantic risk and surface-form strategy, allowing A-ART to sample attack objectives and transformations independently.

We manually create one seed prompt for every risk-category–attack-style pair, yielding 160 distinct single-turn, user-only seeds. The pipeline formats seeds using the target model's native chat template and omits pipeline-defined system prompts. This protocol isolates base-model behavior and external guardrail effects. During dataset construction, we apply semantic deduplication to remove near-duplicate prompts, preventing the dataset from being inflated with redundant surface variants.

3.2 The A-ART Framework

We introduce A-ART, a multi-agent framework that decouples strategy, execution, memory, and evaluation. Figure 2 shows the closed-loop data flow. A-ART uses a Planner to choose an evasion strategy, a Mutator to instantiate the strategy as a prompt, a Diagnostic Judge to score the target response, and a Memory Keeper to summarize reusable feedback. This decomposition converts red-teaming from stateless prompt mutation into an adaptive data-generation process whose outputs carry structured metadata for later retrieval.

Planner (Strategy). The agent uses reasoning to analyze failure modes and select evasion strategies (e.g., shifting from a basic attack style to a composite attack that mixes two attack styles). A-ART skips the Planner on the first turn because the system needs judge feedback before planning becomes meaningful. We utilize DeepSeek-R1-Distill-32B for this role. Newer models offer greater reasoning capabilities, but strong safety alignment reduces their usability for generating adversarial strategies.

Mutator (Execution). The Mutator executes the Planner's directives by generating concrete text prompts. The agent utilizes tool-calling capabilities (e.g., Base64 encoding, translation, and typography) to bypass static lexical filters. We employ instruct models such as Mistral-Nemo-Instruct (12B) and Gemma-3-12b-it for rapid generation.

Memory Keeper. The Memory Keeper manages the Short-Term Memory (STM) of the system. The agent ingests diagnostic logs from the Judge and synthesizes extracted heuristics. The STM prevents the Planner from repeatedly attempting blocked strategies during prolonged execution runs. The state management enables intra-run adaptation.

Diagnostic Judge. We use high-capacity models (primarily DeepSeek-R1-Distill-32B and Gemma-4-31B-it) as the Judge to mitigate evaluator bias and provide granular feedback. The Judge emits a JSON-validated record containing: (1) a binary success flag, (2) a categorical failure verdict such as `MISUNDERSTOOD`, `FILTERED_INPUT`, or `EMPTY`, (3) a severity score in $[0, 1]$, and (4) a diagnostic explanation. A-ART feeds the explanation back to the Memory Keeper and Planner (in case of refusal).

We validate the automated Judge against three author annotators on 89 responses stratified by verdict, risk category, and target model (Table 1). The Judge achieves 1.00 precision and 0.88 recall for the binary success decision, indicating that its errors primarily omit successful attacks rather than inflate ASR. Human agreement is high for success versus refusal ($\alpha = 0.90$) but lower for the three-way FULL/PARTIAL/REFUSAL distinction ($\alpha = 0.56$). We therefore report FULL and FULL+PARTIAL results separately.

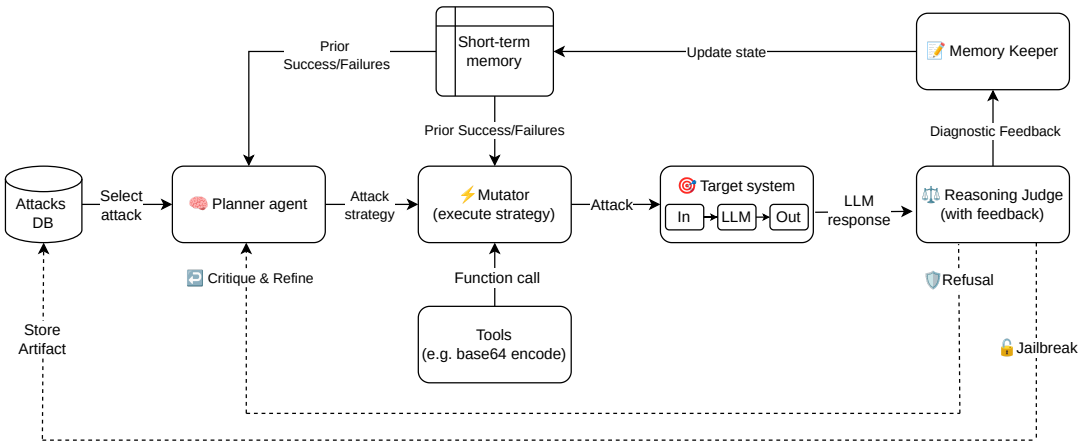


Figure 2: The Adaptive Agentic Red-Teaming (A-ART) loop. A seed attack enters a Planner, which selects a high-level strategy from the current risk category, attack style, and prior failures. A Mutator converts the strategy into a concrete prompt, the target system produces a response, and a Diagnostic Judge returns structured success, severity, and failure-mode labels. A Memory Keeper stores these signals as short-term state, allowing later attempts in the same run to avoid failed strategies while preserving metadata for downstream retrieval.

Metric	Value (95% CI)
Inter-annotator α (success/refusal)	0.90
Inter-annotator α (full/partial/refusal)	0.56
Judge precision	1.00 (0.94–1.00)
Judge recall	0.88 (0.78–0.93)
Judge accuracy	0.90 (0.82–0.95)
Judge–human κ (success)	0.71

Table 1: Judge reliability against three human annotators on 89 stratified responses.

3.3 Gray-Box Evaluation Protocol

We evaluate each target as a *compound AI system* with target model with input and output guardrails. We use Llama Prompt Guard 2 (86M) for input filtering and Llama-Guard-4-12B for output filtering.

We adopt a *gray-box passthrough* protocol: the evaluator logs every guardrail decision but still executes the target model when the input filter triggers. This design separates three failure sources that black-box evaluation conflates: input filtering, intrinsic target-model alignment, and output filtering. Unless stated otherwise, we report *target-only ASR*, which measures the target response before output filtering. Enforcing both guardrails lowers aggregate FULL+PARTIAL ASR from 15.5% to 5.8%, with the output guardrail accounting for most of the reduction (Appendix D.1).

3.4 Experimental Setup and Models

We instantiate A-ART with separate models for planning, mutation, judging, guardrails, and target execution. This role separation supports ablations and prevents a single model family from defining both attack generation and evaluation. Table 2 lists the open-weight models used in each role.

Framework Role	Model(s) Utilized
Planner (Reasoning)	DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025)
Mutator (Execution)	Mistral-Nemo-Instruct-2407 (Mistral AI, 2024) Gemma-3-12b-it (Kamath et al., 2025)
Diagnostic Judge	DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025) Gemma-4-31B-it (Google DeepMind, 2026)
Safety Guardrails	Llama Prompt Guard 2 (Input) (Adcock et al., 2026) Llama-Guard-4-12B (Output) (Adcock et al., 2026)
Target Models	GPT-OSS-20B (Agarwal et al., 2025) Mistral-Nemo-Instruct-2407 (Mistral AI, 2024) Ministral-3-14B-Instruct-2512 (Liu et al., 2026) Phi-4 (Abdin et al., 2024) Llama-3.1-8B-Instruct (Grattafiori et al., 2024) Gemma-3-12b-it (Kamath et al., 2025) Qwen3-4B (Yang et al., 2025a) Qwen3-30B-A3B-Instruct (Yang et al., 2025a)

Table 2: Open-weight models used across A-ART roles, safety guardrails, and compound target systems.

3.5 Dataset Curation and Semantic Diversity

We curate the generated attacks to remove replication artifacts and quantify semantic diversity. First, we compute each prompt's nearest-neighbor (Zhang et al., 2020) F_1 using `microsoft/deberta-large-mnli` (He et al., 2021). Higher nearest-neighbor scores indicate lower local diversity. Figure 3 shows a mean nearest-neighbor similarity of 0.636 and no heavy concentration near 1.0, indicating that exact or near-exact duplicates do not dominate the dataset. The nearest-neighbor score compares each prompt with

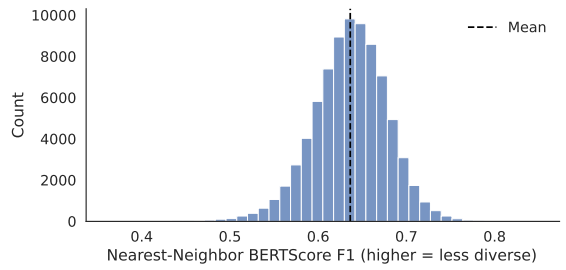


Figure 3: Distribution of nearest-neighbor BERTScore F_1 over the generated attack dataset. The mean score of 0.636 indicates local semantic overlap without concentration near exact duplication.

Target Model	FULL	FULL+PART.
Ministral-3-14B-Instruct	14.9	29.8
Mistral-Nemo-Instruct	16.3	28.9
Gemma-3-12B	5.8	18.9
Llama-3.1-8B-Instruct	7.3	16.2
Qwen3-4B	4.8	13.6
Qwen3-30B-A3B-Instruct	4.7	12.5
Phi-4	1.8	10.5
GPT-OSS-20B	1.0	2.9
Overall	**6.4**	**15.2**

Table 3: Aggregate ASR (%) by target model across both Mutators. FULL counts complete jailbreaks, FULL+PARTIAL adds counts partial harmful compliance. Full generation counts appear in Appendix E.3.

its most similar counterpart and therefore exceeds the average similarity over all prompt pairs.

Second, we create embeddings for all prompts with `Qwen3-Embedding-8B` (Yang et al., 2025a) and remove pairs whose cosine similarity exceeds 0.95. This semantic deduplication step removes near-duplicate surface variants; Table 19 breaks down the removals by Mutator and target. We retain clustered but non-identical attacks because such clusters represent meaningful local variation around shared risk categories or attack styles.

3.6 Dataset Statistics

A-ART generated 1,253,094 adversarial prompts across eight open-weight target models, yielding 190,575 successful jailbreaks. We fixed the Planner and Diagnostic Judge to DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025) and varied only the Mutator model to isolate generation effects. The Gemma-3 Mutator achieved higher aggregate ASR than Mistral-Nemo-Instruct (Mistral AI, 2024) (17.1% vs. 13.2%); Appendix E.3 reports the full mutator-by-target count breakdown. After removing 1,967 near-duplicates at a cosine similarity threshold of 0.95, the released dataset contains 188,608 unique successful attacks.

Table 3 summarizes target-level ASR after aggregating over both Mutators. We report two success criteria: FULL counts complete jailbreaks, while FULL+PARTIAL includes responses that provide partial harmful assistance. The results show alignment stratification across model families. Mistral-family targets yield the highest ASRs, reaching 29.8% on Ministral-3-14B-Instruct (Liu et al., 2026) and 28.9% on Mistral-Nemo-Instruct (Mistral AI, 2024). GPT-OSS-20B (Agarwal et al., 2025) forms the most robust stratum, with only 2.9% aggregate ASR. This robustness gap motivates the transferability analysis in Section 4, where we test whether attacks discovered against stronger targets generalize more broadly.

4 Dataset Evaluation and Transferability

We evaluate whether attacks generated on one target model transfer to other architectures. This analysis tests the central premise of metadata-aware reuse: a useful red-teaming dataset should not merely store successful attacks, but should reveal which attacks generalize across model families and which failures remain target-specific.

Cross-Architecture Transferability. We measure transferability as the number of target models compromised by the same payload, under a consensus-judge criterion. A transfer counts only when both Gemma-4 and DeepSeek classify the target response as at least a partial jailbreak or full jailbreak. Figure 4 shows that most successful attacks transfer to a small number of targets, while a smaller set generalizes broadly across six or seven models. The distribution therefore reveals two regimes: a large mass of target-specific attacks that exploit narrow alignment boundaries, and a high-value tail of broadly transferable attacks that expose shared weaknesses across model families. This tail motivates metadata-aware retrieval, since historical attack outcomes can help prioritize payloads with higher expected reuse value on unseen targets.

Asymmetric Robustness Hierarchy. We next construct a pairwise transfer matrix, where rows denote the source model used during attack generation and columns denote the evaluated target model. Figure 5 reveals a strong asymmetry: attacks discovered against robust targets transfer far more broadly than attacks discovered against vulnerable targets. In particular, GPT-OSS-derived attacks compromise weaker models with high success

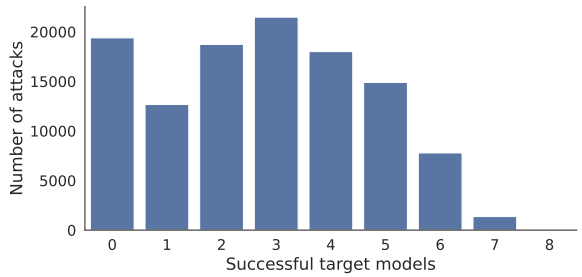


Figure 4: Consensus-based attack transferability across eight target models. The x-axis shows how many targets each payload compromises; the y-axis shows the number of attacks. A transfer counts only when both Gemma-4 and DeepSeek classify the response as at least a partial jailbreak. Most attacks remain target-specific, while a smaller tail transfers across six or seven models.

rates, while Mistral-derived attacks rarely transfer back to GPT-OSS. This asymmetry supports a practical data-generation heuristic: red-teaming stronger-aligned targets yields attacks with higher downstream reuse value.

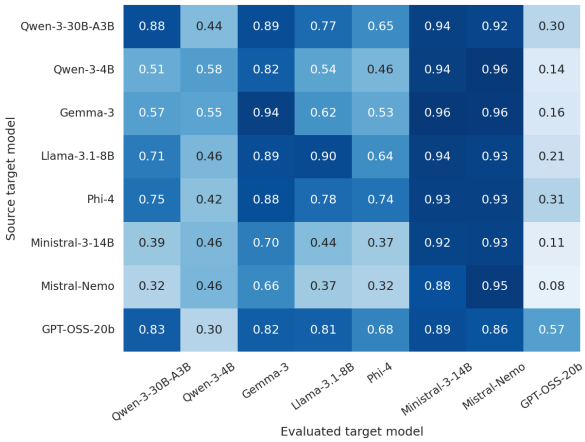


Figure 5: Pairwise transferability matrix under the DeepSeek Judge. Rows indicate source models used for attack generation, columns indicate evaluated targets. Attacks generated against robust targets transfer more broadly than those generated against vulnerable targets.

5 Adversarial Seed Sampling and Metadata-Aware Retrieval

We formalize the optimization of static adversarial datasets against newly deployed target models as a resource-constrained retrieval problem. When a new target model arrives, an adversary faces a finite exploration budget. The adversary selects payloads that maximize the Attack Success Rate (ASR) without prior knowledge of the target's specific defense alignment parameters. We evaluate sampling strategies under a strict leave-one-target-out evaluation protocol across eight distinct model architectures. The framework isolates the evaluation environment of the held-out model and ranks candidate attacks using empirical feedback gathered solely from the remaining seven visible models. The pipeline reveals evaluation outcomes batch by batch, allowing adaptive methods to dynamically update their internal rankings. We cap the maximum attack exploration budget at 4,000 total queries across sequential batch sizes of $[25, 50, 100, 200]$.

5.1 Sampling Methodology

We compare five sampling rules. **Random** samples attacks uniformly without replacement. **Weighted Success** samples attacks with probability proportional to the historical training success rate. **Static Top-N** selects the 4,000 attacks that compromised the largest number of visible target models, using severity scores only to break ties. The remaining two methods use online empirical-Bayes updates.

Adaptive Baseline: This baseline follows the empirical Bayes framework. Updates a shrinkage-based posterior mean ($\hat{\mu}$) after evaluating each sequential batch. To prevent edge cases (exact 0 or 1 probabilities) at the initial ranking stage, we model the baseline success probability as the posterior mean of a Binomial distribution under a uniform Beta$(\alpha = 1, \beta = 1)$ prior. This yields standard Laplace smoothing for the mean:

$$\mu_0 = \frac{S_{\text{tr}} + 1}{N_{\text{tr}} + 2} \tag{1}$$

where S_{tr} denotes observed training successes, and N_{tr} represents the total number of visible training models. As the pipeline observes live target feedback, it updates the estimate using empirical Bayes:

$$\hat{\mu} = \frac{S_{\text{obs}} + m \cdot \mu_0}{N_{\text{obs}} + m} \tag{2}$$

where S_{obs} represents live successes, N_{obs} represents total live observations, and m acts as a pseudo-count regularizer.

Let $\bar{\mu}_{\text{feat}}$ and $\bar{\mu}_{\text{mod}}$ represent the averaged empirical Bayes posteriors across all base metadata features and visible model indicators, respectively. The final baseline ranking score (S_{base}) is computed as a convex linear combination of these estimates:

$$S_{\text{base}} = \beta_1 \mu_0 + \beta_2 \bar{\mu}_{\text{feat}} + \beta_3 \bar{\mu}_{\text{mod}} \tag{3}$$

where the coefficients $\beta_1, \beta_2, \beta_3 \in [0, 1]$ and $\sum \beta_i = 1$. For our evaluations, we use these coefficients using a heuristic (*i.e.*, $\beta_1 = 0.45$, $\beta_2 = 0.25$,

$\beta_3 = 0.30$) to ensure the smoothed historical prior (μ_0) acts as a strong anchor. This regularization prevents noisy early-batch observations in the feature and model posteriors from overwhelmingly misdirecting the search trajectory.

Adaptive Transferability: Extends the adaptive baseline by integrating explicit cross-model effectiveness dimensions. Rather than relying solely on the binary success prior (μ_0), this strategy initializes the first evaluation batch using a custom transferability score (S_{trans}):

$$S_{\text{trans}} = w_1\mu_0 + w_2 R_{\text{st}} + w_3 R_{\text{pa}} + w_4\left(\frac{\bar{V}_{\text{j}}}{2}\right) \quad (4)$$

where R_{st} denotes the strict unsafe rate and R_{pa} denotes the partial compliance rate. The term $\bar{V}_{\text{j}}$ represents the mean ordinal judge severity. The Diagnostic Judge maps system responses to discrete ordinal categories: SAFE $= 0$, PARTIAL $= 1$, and UNSAFE $= 2$. Dividing the aggregate severity $\bar{V}_{\text{j}}$ by a factor of 2 maps the metric directly to the continuous interval $[0, 1]$, ensuring structural scale parity with the accompanying probability rates.

This equation encodes a domain-specific heuristic. The coefficients ($w_i \in [0, 1], \sum w_i = 1$) operationalize our design hypotheses regarding vulnerability transfer. We allocate the highest weight ($w_1 = 0.4$) to broad harmful transfer (μ_0) as the primary indicator of payload utility. We assign heavy secondary importance ($w_2 = 0.3$) to explicit safety alignment failures (R_{st}). Finally, we assign equal minor weights ($w_3 = w_4 = 0.15$) to partial compliance and aggregate severity, treating them as auxiliary friction signals that guide seed selection without dominating the core success rates.

Crucially, as the pipeline evaluates sequential batches, the algorithm dynamically adjusts the downstream scoring weights using a linear annealing schedule. Let $\lambda \in [0, 1]$ represent the learning fraction, defined as the ratio of observed queries to a predefined batch saturation threshold. The final live ranking score (S_{live}) aggregates the static transferability prior (S_{trans}), the live empirical posteriors ($\bar{\mu}_{\text{feat}}$, $\bar{\mu}_{\text{mod}}$, and transfer profiles $\bar{\mu}_{\text{trans}}$), and a frequency-based exploration term ($\bar{U}_{\text{trans}}$):

$$\begin{aligned} S_{\text{live}} = {} & \alpha_1(\lambda) S_{\text{trans}} + \alpha_2(\lambda)\bar{\mu}_{\text{feat}} \\ & + \alpha_3(\lambda)\bar{\mu}_{\text{mod}} + \alpha_4\bar{\mu}_{\text{trans}} + c \cdot \bar{U}_{\text{trans}} \end{aligned} \quad (5)$$

where $\lambda \in [0, 1]$ is the fraction of the query budget observed, $\bar{\mu}_{\text{trans}}$ averages transfer-profile posteriors, and $\bar{U}_{\text{trans}} = \text{mean}_f(1/\sqrt{n_f + 1})$ rewards undersampled transfer features with visitation counts n_f. The prior weight $\alpha_1(\lambda)$ decays linearly from 0.28 to 0.20, $\alpha_2(\lambda)$ and $\alpha_3(\lambda)$ increase with observed target evidence, $\alpha_4 = 0.35$, and $c = 0.06$. This schedule shifts ranking from historical transfer priors toward target-specific evidence while retaining a small Upper Confidence Bound-style exploration.

5.2 Empirical Retrieval Analysis

Retrieval Strategy	Mean ASR ± SD	Δ vs. Random
Random Sampling	61.28 ± 25.14	–
Weighted Success Sampling	68.19 ± 25.08	+6.91
Static Top-4000 Retrieval	83.82 ± 25.18	+22.54
Adaptive Baseline	86.93 ± 20.25	+25.65
Adaptive Transferability	**89.83 ± 16.97**	**+28.55**

Table 4: Leave-one-target-out retrieval performance across eight held-out target models. SD denotes the standard deviation, in percentage points, across target-level ASRs. Δ reports absolute ASR gain over random sampling with the FULL+PARTIAL success criterion.

Table 4 shows that historical attack outcomes strongly predict zero-shot transfer. Random sampling reaches 61.28% mean ASR, while Weighted Success Sampling reaches 68.19%. Static Top-4000, which selects the 4,000 attacks that compromised the most target models, reaches 83.82%. Historical cross-model success therefore explains much of the retrieval gain. Online adaptation provides additional improvement. Adaptive Baseline reaches 86.93%, and Adaptive Transferability reaches 89.83% with the lowest cross-target SD. Adaptive Transferability gains 6 percentage points over Static Top-4000 and 2.90 points over Adaptive Baseline. Table 5 examines which information accounts for these gains, while Appendix D.2 reports sensitivity to candidate-pool size.

Retrieval signal	ASR	Δ vs. hist.
Random (floor)	61.3	−22.5
Historical success only	83.8	0.0
Source-model only	70.5	−13.3
All categorical metadata	79.1	−4.8
Adaptive Baseline (online)	86.9	+3.1
Adaptive Transf. (online)	**89.8**	+6.0

Table 5: Retrieval ablation under leave-one-target-out evaluation with a 4,000-query budget. Δ reports the mean ASR difference from historical success.

Table 5 compares representative signals and signal groups under the same evaluation protocol. Historical success provides the strongest static rank-

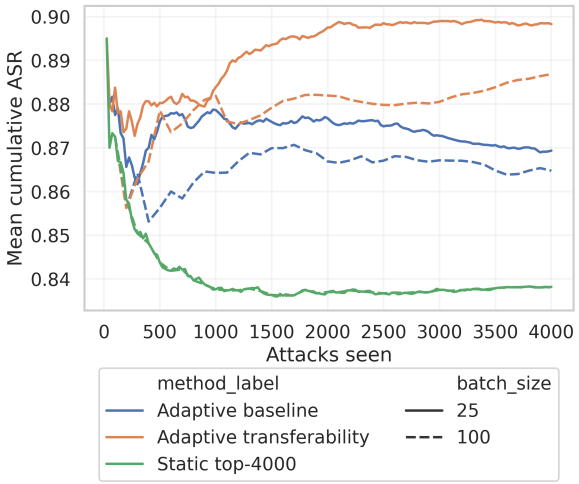


Figure 6: Cumulative ASR over the retrieval budget under leave-one-target-out evaluation. Static Top-4000 front-loads strong historical attacks but cannot revise their order. Adaptive methods re-rank after each batch using feedback from the held-out target.

Target	Rand.	Top-N	Adapt.	Δ
Claude Haiku 4.5	1.2%	5.5%	**18.9%**	+13.4
GPT-5.4-mini	1.0%	2.8%	**7.6%**	+4.8
GPT-5.4-nano	4.2%	12.2%	**32.4%**	+20.2
Gemini 3.1 Flash Lite	46.9%	84.0%	**93.8%**	+9.8

Table 6: Closed-box ASR when the evaluator observes only final model responses. Δ reports absolute ASR gain over Static Top-N.

	Qwen3-4B		Mistral-Nemo	
Attacker	**FULL**	**F+P**	**FULL**	**F+P**
Direct (no opt.)	0.0	0.9	3.7	6.5
PAIR	**7.4**	21.3	12.0	33.3
TAP	5.6	15.7	**22.2**	43.5
A-ART (reuse)	0.9	**64.8**	12.0	**83.3**

Table 7: ASR (%) under matched query budgets and a shared Gemma-4 Judge. FULL counts complete jailbreaks; F+P additionally counts partial harmful compliance. Column-best results appear in bold.

ing signal at 83.8% ASR. Categorical metadata remains informative at 79.1%, but does not match historical success by itself. Adaptive Baseline exceeds the historical-success ranker by 3.1 points after incorporating held-out-target feedback. Adaptive Transferability adds strict-unsafe rate, partial-compliance rate, judge severity, transfer profiles, and exploration, increasing the gain to 6.0 points. The results show that historical success supplies a strong initial ranking, while target-specific updates provide the remaining improvement.

Figure 6 shows how performance changes over the query budget. Static Top-4000 starts strongly because it evaluates the historically most transferable attacks first. Its cumulative ASR then declines as the fixed ranking reaches weaker candidates. Adaptive methods instead update the remaining ranking after every batch using feedback from the held-out target. Adaptive Transferability maintains the strongest trajectory because its updates combine observed target outcomes with transfer-specific safety signals.

5.3 Generalizability to Closed-Box Models

We test whether retrieval signals learned from open-weight targets generalize to proprietary APIs. In this setting, the evaluator observes only the final model response and has no access to logits, system prompts, guardrail decisions, or refusal classifier outputs. We evaluate Claude Haiku 4.5, GPT-5.4-mini, GPT-5.4-nano, and Gemini 3.1 Flash Lite.

Table 6 shows consistent closed-box transfer. Adaptive Transferability achieves the highest ASR on every proprietary target, including 93.8% on Gemini 3.1 Flash Lite and 32.4% on GPT-5.4-nano. GPT-5.4-mini remains the most resilient target, limiting adaptive ASR to 7.6%; nevertheless, adaptive selection raises ASR by 7.6× over random sampling. The results separate absolute robustness from retrieval sensitivity: stronger targets reduce the final ASR, but adaptive selection still amplifies attack yield under the same observation constraints. This pattern supports the central retrieval hypothesis: structured metadata collected from open-weight pipelines predicts which attacks deserve priority on unseen closed-box systems.

5.4 Comparison with Attack Optimizers

We compare retrieval-based reuse with PAIR (Chao et al., 2025) and TAP (Mehrotra et al., 2024), two optimization-driven red-teaming methods. We run all methods under matched query budgets, evaluate the same target models, and rescore every response with the same Gemma-4 Judge. We also include direct prompting as a no-optimization baseline.

Table 7 shows different strengths. A-ART retrieval achieves the highest FULL+PARTIAL ASR on both targets, reaching 64.8% on Qwen3-4B and 83.3% on Mistral-Nemo. PAIR and TAP more often produce complete jailbreaks: PAIR reaches 7.4% FULL ASR on Qwen3-4B, while TAP reaches 22.2% on Mistral-Nemo.

PAIR and TAP also improve significantly over

direct prompting under paired McNemar tests with Holm correction. For example, PAIR outperforms direct prompting on Qwen3-4B ($p = 9 \times 10^{-6}$), and TAP outperforms direct prompting on Mistral-Nemo ($p = 1.7 \times 10^{-7}$). These tests confirm that the optimization baselines are effective rather than weak reference implementations.

The comparison highlights a different objective for each approach. PAIR and TAP optimize new attacks against a specific target, which favors complete jailbreaks. A-ART retrieves attacks that have already transferred across other models, which favors broad and query-efficient reuse. The approaches are compatible: A-ART can use PAIR, TAP, or another optimizer as its Mutator.

5.5 Architectural Component Ablation

We ablate A-ART components against `GPT-OSS`, the most robust open-weight target. All variants use a `gemma-3-12b` Mutator and a `gemma-4-31B-it` Diagnostic Judge. The configurations isolate short-term memory, diagnostic feedback, their combination, and the full Planner-mediated loop.

Cfg.	Variant	ASR	Δ
A	Baseline Mutator	5.3%	–
B	Short-Term Memory Only	5.2%	-0.1
C	Diagnostic Feedback Only	5.1%	-0.2
D	Feedback + Memory, No Planner	6.2%	+0.9
E	**Full A-ART**	**7.1%**	**+1.8**

Table 8: Architectural ablation against `GPT-OSS` using synthetic seeds. Each variant aggregates 20.0k–26.8k evaluated prompts; Δ reports absolute ASR change relative to the baseline Mutator.

Table 8 shows that memory or feedback alone does not improve ASR over the baseline Mutator. Combining feedback with memory improves ASR to 6.2%, and adding the Planner raises ASR to 7.1%. The result supports the A-ART design: judge feedback becomes useful when Planner converts short-term state into revised attack strategies.

5.6 Metadata-Aware Retrieval for Attack Generation

We test whether metadata-aware retrieval improves attack generation, not only post-hoc selection from a fixed dataset. We replace uniform seed sampling inside the generation loop with Adaptive Transferability sampling. After each batch, the sampler updates its posterior from Judge feedback and selects next seeds with higher estimated transfer value.

We evaluate this variant against `GPT-OSS`, the most robust open-weight target in our study. Table 9 compares random and metadata-aware seed retrieval for baseline and the A-ART framework.

Architecture	Seed	ASR	Δ
Baseline Mutator	Random	5.3%	–
Baseline Mutator	Adaptive	7.91%	+2.61
Full A-ART	Random	7.1%	–
Full A-ART	**Adaptive**	**9.0%**	**+1.90**

Table 9: Metadata-aware retrieval improves attack generation against `GPT-OSS`. Each row aggregates 16.6k–26.8k evaluated prompts; Δ reports the absolute ASR gain over random seed retrieval.

Metadata-aware seed retrieval improves ASR in both architectures. For the baseline, adaptive seed selection raises ASR from 5.3% to 7.91%, surpassing Full A-ART with random seeds (7.1%). Full A-ART with metadata-aware retrieval achieves the best result at 9.0% ASR. The pattern shows that seed selection and agentic adaptation contribute complementary gains: retrieval supplies stronger starting points, while A-ART converts those seeds into higher-yield attack variants.

6 Conclusion

We introduce a data-driven paradigm for red-teaming that replaces stateless prompt generation with metadata-aware reuse of adversarial attacks. A-ART produces structured attack data with diagnostic signals, enabling retrieval algorithms to transfer attacks to unseen targets.

Experiments show that historical attack outcomes provide strong zero-shot signal, and that adaptive retrieval further improves efficiency by combining priors with live feedback. The approach reaches 89.83% ASR in open-weight settings, generalizes to closed-box APIs, and improves attack generation when used for seed selection.

These results shift red-teaming from rediscovering attacks to reusing them, establishing structured adversarial datasets and adaptive retrieval as a foundation for continuous robustness evaluation.

Reproducibility The A-ART framework is available at github.com/NASK-AISafety/a-art, and the metadata-aware retrieval code at github.com/NASK-AISafety/jailbreak-retrieval. To support further research, we release a dataset of 188,608 unique successful attacks with rich metadata annotations at huggingface.co/datasets/NASK-PIB/a-art-jailbreaks.

7 Limitations

Safety alignment can constrain both agentic stages of A-ART. The Planner may refuse to formulate an adversarial strategy, and the Mutator may refuse to execute that strategy. We use publicly released models with their default weights and safety settings, so the reported results include this practical constraint. Future work could replace either component with a dedicated red-team model trained under controlled access and evaluation procedures.

Our study considers only single-turn, user-only attacks and does not evaluate conversational jailbreaks. Multi-turn attacks require retrieval over dialogue trajectories, including turn-level state and feedback, rather than independent prompts.

Our evaluation relies primarily on an automated Judge and limited human verification.

Finally, the corpus contains English-language attacks and has an uneven distribution across risk categories.

8 Ethics and Potential Risks

The open-source publication of an adversarial dataset introduces distinct dual-use risks. Malicious actors could utilize the curated dataset to automate high-throughput exploitation campaigns against commercial production systems. Similarly, threat actors could exploit our multi-agent architecture code to bypass external guardrail mechanisms.

Despite these deployment risks, the defensive utility of open-access safety artifacts justifies the publication of the dataset. Blue-teaming practitioners require large-scale, diverse empirical benchmarks to train robust input and output filters. Furthermore, alignment engineers need expansive, structured logs to execute effective adversarial training and strengthen the intrinsic boundaries of future models. To balance defensive utility against misuse, we adopt a tiered release. The red-teaming framework and code are released under an open Apache 2.0 license to promote transparent academic research and reproducibility. The high-risk attack corpus is instead released under gated, controlled access: it is granted to approved requests under an acceptable-use agreement that prohibits unauthorized exploitation, and the reproducibility artifacts are separated from the high-throughput generation tooling. We do not support or condone the optimization of these artifacts for unauthorized exploitation.

Acknowledgments

This research was supported by the European Funds for Digital Development Programme 2021-2027 (FERC), grant no. FERC.02.02-IP.01-0002/23). We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2026/019863.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Aaron Adcock, Aayushi Srivastava, Abhimanyu Dubey, Abhinav Jauhri, Apeksha Pande, and Abhinav Pandey. 2026. The llama 4 herd: Architecture, training, evaluation, and deployment notes. *ArXiv*, abs/2601.11659.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, and 106 others. 2025. gpt-oss-120b gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Ali Asad, Stephen Obadinma, Radin Shayanfar, and Xiaodan Zhu. 2025. Reddebate: Safer responses through multi-agent red teaming debates. *arXiv preprint arXiv:2506.11083*.

Advik Basani and Xiao Zhang. 2025. Gasp: Efficient black-box generation of adversarial suffixes for jailbreaking llms. *Advances in Neural Information Processing Systems*, 38:146413–146458.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, and 1 others. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37:55005–55029.

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.

Saket Sanjeev Chaturvedi, Gaurav Bagwe, Lan Emily Zhang, and Xiaoyong Yuan. 2025. Aip: Subverting retrieval-augmented generation via adversarial instructional prompt. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15872–15889.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. Uprise: Universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12318–12337.

Rishit Chugh. 2026. Recap: A resource-efficient method for adversarial prompting in large language models. *arXiv preprint arXiv:2601.15331*.

Pepijn Cobben, Xuanqiang Angelo Huang, Thao Amelia Pham, Isabel Dahlgren, Terry Jingchen Zhang, and Zhijing Jin. 2026. Gt-harmbench: Benchmarking ai safety risks through the lens of game theory. *arXiv preprint arXiv:2602.12316*.

Quy-Anh Dang, Chris Ngo, and Truong-Son Hy. 2026. Redbench: A universal dataset for comprehensive red teaming of large language models. *arXiv preprint arXiv:2601.03699*.

Google DeepMind. 2026. Gemma 4: Open multimodal models. `https://ai.google.dev/gemma/docs/core/model_card_4`. Model card, Accessed: 2026-05-25.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and 1 others. 2024. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *Advances in Neural Information Processing Systems*, 37:47094–47165.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, and Ramona Merhej. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Ninad Kulkarni, Xian Wu, Siddharth Varia, and Dmitriy Bespalov. 2025. Agent vs. agent: Automated data generation and red-teaming for custom agentic workflows. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 912–936.

Anurakt Kumar, Divyanshu Kumar, Jatan Loya, Nitin Aravind Birur, Tanay Baswa, Sahil Agarwal, and Prashanth Harshangi. 2024. Sage-rt: Synthetic alignment data generation for safety evaluation and red teaming. *arXiv preprint arXiv:2408.11851*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Deokjae Lee, JunYeong Lee, Jung-Woo Ha, Jin-Hwa Kim, Sang-Woo Lee, Hwaran Lee, and Hyun Oh Song. 2023. Query-efficient black-box red teaming via bayesian optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11551–11574.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3923–3954.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, and Albert Jiang. 2026. Ministral 3. *Preprint*, arXiv:2601.08584.

Xiaogeng Liu, Peiran Li, G Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li, and Chaowei Xiao. 2025. Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms. In *International Conference on Learning Representations*, volume 2025, pages 10313–10360.

Gaoxiang Luo and Aryan Deshwal. 2025. Com-bom: Bayesian exemplar search for efficiently exploring the accuracy-calibration pareto frontier. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20350–20363.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, and 1 others. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning*, pages 35181–35224. PMLR.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2024. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105.

Mistral AI. 2024. Mistral nemo. https://mistral.ai/news/mistral-nemo. Model release, Accessed: 2026-05-25.

Jingyu Peng, Maolin Wang, Nan Wang, Jiatong Li, Yuchen Li, Yuyang Ye, Wanyu Wang, Pengyue Jia, Kai Zhang, and Xiangyu Zhao. 2025. Logic jailbreak: Efficiently unlocking llm safety restrictions through formal logical expression. *arXiv preprint arXiv:2505.13527*.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2655–2671.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. 2025. Great, now write an article about that: The crescendo {Multi-Turn}{LLM} jailbreak attack. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2421–2440.

Mikayel Samvelyan, Sharath C Raparthy, Andrei Lupu, Eric Hambro, Aram H Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, and 1 others. 2024a. Rainbow teaming: Open-ended generation of diverse adversarial prompts. *Advances in Neural Information Processing Systems*, 37:69747–69786.

Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram H. Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob N. Foerster, Tim Rocktäschel, and Roberta Raileanu. 2024b. Rainbow teaming: Open-ended generation of diverse adversarial prompts. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Lennart Schneider, Martin Wistuba, Aaron Klein, Jacek Golebiowski, Giovanni Zappella, and Felice Antonio Merra. 2025. Hyperband-based bayesian optimization for black-box prompt selection. In *International Conference on Machine Learning*, pages 53413–53438. PMLR.

Jialin Song, Xiaodong Liu, Weiwei Yang, Wuyang Chen, Mingqian Feng, Xuekai Zhu, and Jianfeng Gao. 2026. Multibreak: A scalable and diverse multi-turn jailbreak benchmark for evaluating llm safety. *arXiv preprint arXiv:2605.01687*.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and 1 others. 2024. A strongreject for empty jailbreaks. *Advances in Neural Information Processing Systems*, 37:125416–125440.

Peiran Wang, Xiaogeng Liu, and Chaowei Xiao. 2025a. Repd: Defending jailbreak attack through a retrieval-based prompt decomposition process. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 283–294.

Siyin Wang, Chao-Han Huck Yang, Ji Wu, and Chao Zhang. 2024a. Bayesian example selection improves in-context learning for speech, text and visual modalities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20812–20828.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2024b. Do-not-answer: Evaluating safeguards in llms. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 896–911.

Zhun Wang, Vincent Siu, Zhe Ye, Tianneng Shi, Yuzhou Nie, Xuandong Zhao, Chenguang Wang, Wenbo Guo, and Dawn Song. 2025b. Agentxploit: End-to-end redteaming of black-box ai agents. *arXiv e-prints*, pages arXiv–2505.

Chen Xiong, Pin-Yu Chen, and Tsung-Yi Ho. 2026. Cop: agentic red-teaming for large language models using composition of principles. *Advances in Neural Information Processing Systems*, 38:104257–104291.

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, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Guangyu Yang, Jinghong Chen, Jingbiao Mei, Weizhe Lin, and Bill Byrne. 2025b. Retrieval-augmented defense: Adaptive and controllable jailbreak prevention for large language models. *arXiv preprint arXiv:2508.16406*.

Jiayi Yuan, Jonathan Nöther, Natasha Jaques, and Goran Radanović. 2026. Agenticred: Optimizing agentic systems for automated red-teaming. *arXiv preprint arXiv:2601.13518*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Zheng Zhang, Jiarui He, Yuchen Cai, Deheng Ye, Peilin Zhao, Ruili Feng, and Hao Wang. 2025. Genesis: Evolving attack strategies for llm web agent red-teaming. *arXiv preprint arXiv:2510.18314*.

Jianli Zhao, Tingchen Fu, Rylan Schaeffer, Mrinank Sharma, and Fazl Barez. 2025. Chain-of-thought hijacking. *arXiv preprint arXiv:2510.26418*.

Yiran Zhao, Wenyue Zheng, Tianle Cai, Xuan Long, Kenji Kawaguchi, Anirudh Goyal, and Michael Q Shieh. 2024. Accelerating greedy coordinate gradient and general prompt optimization via probe sampling. *Advances in Neural Information Processing Systems*, 37:53710–53731.

Zhaojiacheng Zhou. 2026. Proteus: A self-evolving red team for agent skill ecosystems. *arXiv preprint arXiv:2605.11891*.

Christos Ziakas, Nicholas Loo, Nishita Jain, and Alessandra Russo. 2025. Red-bandit: Test-time adaptation for llm red-teaming via bandit-guided lora experts. *arXiv preprint arXiv:2510.07239*.

Jesse Zymet, Andy Luo, Swapnil Shinde, Sahil Wadhwa, and Emily Chen. 2026. Adaptive instruction composition for automated llm red-teaming. *arXiv preprint arXiv:2604.21159*.

A Appendix Overview

The appendix provides additional implementation details and supporting analyses. Appendix B reports hardware, inference, licensing, and AI-assistance details. Appendix C lists the attack-style extensions used during seed construction. Appendix D reports architectural ablations, seed-origin comparisons, few-shot sensitivity, and retrieval batch-size robustness. Appendix E validates dataset quality through mutator-level statistics, embedding clusters, stochastic resampling, and judge-agreement analysis.

B Infrastructure, Licensing, and AI Assistance

Hardware and inference. A-ART requires concurrent inference across Planner, Mutator, target, guardrail, and Judge models. We therefore restrict open-weight agent models to the sub-32B regime, with most experiments using 4B–14B models. We run active generation on a high-performance cluster with four NVIDIA GH200 Grace Hopper Superchips, each with 96GB VRAM, using `vLLM` (Kwon et al., 2023) and continuous batching for high-throughput inference. We run offline post-processing and bulk target evaluation on an NVIDIA A100 node with 80GB VRAM.

Licensing. We use open-weight models to support reproducibility and avoid proprietary telemetry in the main experimental pipeline. All model weights remain governed by their original licenses and terms of use, including the Llama, Gemma, Mistral, Qwen, and Phi model families.

AI assistance. We used GitHub Copilot for code implementation support and chat-based writing tools, including Gemini and Microsoft Copilot, for grammar, and style polishing. The authors defined the research questions, methodology, experiments, analyses, and claims. We reviewed and edited all AI-assisted text and code before inclusion.

C Additional Taxonomy Details

We use Rainbow Teaming (Samvelyan et al., 2024b) as the base taxonomy and add six high-level attack-style families to increase coverage over surface-form and instruction-framing variation. These additions are:

- **Format Obfuscation:** Encoding prompts in other formats (e.g., Base64) to evade filters.
- **Linguistic Alteration:** Mixing languages and inserting escape characters to disrupt instruction parsing.
- **XML Tag Forgery:** Injecting fake XML structural tags to manipulate the internal data structures of prompt templates.
- **Template Evasion:** Instructing the target model to ignore prior system guidelines to address out-of-scope topics.
- **Prefix Injection:** Forcing the target model to begin responses with affirmative phrases to alter subsequent generation tone.
- **Refusal Suppression:** Restricting target vocabulary to block standard rejection phrases.

The main paper uses these labels as metadata features for sampling, deduplication, and retrieval. We keep the appendix description at the style-family level because the paper studies transferability and retrieval rather than operational prompt recipes.

Target	Target Only	+ Input	+ Output	Full System
Ministral-3-14B	29.8	24.9	7.9	6.9
Mistral-Nemo	28.9	23.7	10.0	8.5
Gemma-3-12B	18.9	16.2	6.3	5.5
Llama-3.1-8B	16.2	13.9	9.6	8.2
Qwen3-4B	13.6	11.3	7.6	6.4
Qwen3-30B-A3B	12.5	11.2	5.4	4.8
Phi-4	11.9	10.3	8.0	6.9
GPT-OSS-20B	3.6	3.2	2.3	2.0
Overall	**15.5**	**13.2**	**6.7**	**5.8**

Table 10: Guardrail component ablation. Values report FULL+PARTIAL ASR (%) for the target alone, each guardrail enforced separately, and the complete guarded system.

D Ablations and Sensitivity Analyses

D.1 Guardrail Component Ablation (Latent vs. End-to-End)

Our gray-box protocol records guardrail decisions while continuing execution through the complete pipeline. A single run therefore reveals both target-only behavior and the effect of enforcing each guardrail. Table 10 reports FULL+PARTIAL ASR for four configurations: the target model alone, the target with the input guardrail, the target with the output guardrail, and the complete guarded system.

The input guardrail reduces aggregate ASR from 15.5% to 13.2%, whereas the output guardrail reduces ASR to 6.7%. Enforcing both guardrails yields 5.8% ASR. The output guardrail therefore

accounts for most of the observed reduction, although both components contribute to end-to-end robustness. Target-level differences remain substantial: the complete system ranges from 2.0% ASR on GPT-OSS-20B to 8.5% on Mistral-Nemo.

The target-only values differ slightly from Table 3, including 15.5% versus 15.2% overall, because Table 10 includes auxiliary-Mutator runs that are excluded from the released dataset.

D.2 Retrieval Pool-Size Sensitivity

We test how candidate-pool size affects Adaptive Transferability under the same leave-one-target-out protocol and 4,000-query budget used in Table 4. We vary the available attack pool from 6k candidates to the complete corpus while leaving the ranking method and evaluation protocol unchanged.

Pool size	6k	12k	25k	50k	Full
Adaptive ASR	74.3	83.9	86.2	88.2	89.8

Table 11: Adaptive Transferability ASR (%) as the candidate pool grows under leave-one-target-out evaluation with a 4,000-query budget.

Table 11 shows that candidate-pool size contributes to retrieval performance: ASR rises from 74.3% with 6k attacks to 89.8% with the full corpus. The gains become progressively smaller as the pool grows. Expanding the pool from 6k to 12k adds 9.6 percentage points, while expanding it from 50k to the full corpus adds 1.6 points. Adaptive Transferability therefore remains effective on substantially smaller pools, while the complete dataset provides additional attack coverage.

D.3 Retrieval Score Composition

Table 12 lists which metadata signals enter each retrieval method. Static Top-4000 ranks by the Laplace-smoothed historical-success prior (μ_0) alone; Adaptive Baseline (Eq. 3) adds the categorical and cross-model transfer signals; and Adaptive Transferability (Eqs. 4–5) additionally uses the strict-unsafe and partial-compliance rates, judge severity, and an exploration bonus. In the latter, success and severity seed the Eq. 4 prior and the remaining signals enter through the Eq. 5 update.

Metadata signal	**Static**	**Ad. Base.**	**Ad. Transf.**
Historical success (μ_0)	•	•	•
Attack style		•	•
Risk category		•	•
Source model		•	•
Planner action		•	•
Cross-model transfer profile		•	•
Strict-unsafe rate			•
Partial-compliance rate			•
Judge severity			•
Exploration bonus			•

Table 12: Metadata signals used by each retrieval score. Static Top-4000 uses only the historical-success prior; the adaptive methods progressively incorporate categorical, transfer, and online safety-friction signals.

D.4 Historical Manual Seeds

We repeat the component ablation from Section 5.5 using a legacy seed pool of 160 manually authored red-teaming prompts. Unlike the synthetic seeds used in the main ablation, which were selected from the generated A-ART dataset, these historical seeds were short human-written prompts designed for older, less strongly guarded models. This experiment therefore tests seed-origin robustness: whether A-ART can still extract useful attack variants when initialization begins from a smaller and weaker seed distribution.

Table 13 shows that historical seeds reduce absolute ASR across all configurations. The baseline Mutator drops from 5.3% with synthetic seeds in Table 8 to 2.0% with historical manual seeds. Full A-ART recovers much of this lost utility, reaching 5.5% ASR and a 2.75× relative gain over the manual-seed baseline. The comparison supports two conclusions. First, seed quality matters: richer synthetic attacks provide a stronger starting distribution than short legacy prompts. Second, the Planner–Memory–Judge loop remains useful under weak initialization, because the full A-ART configuration extracts substantially more value from historical seeds than the baseline Mutator.

Config	**Variant**	**ASR**	**Δ**
A	Baseline Mutator	2.0%	–
B	Short-Term Memory Only	4.1%	+2.1
C	Diagnostic Feedback Only	3.2%	+1.2
D	Feedback + Memory, No Planner	4.4%	+2.4
E	**Full A-ART**	**5.5%**	**+3.5**

Table 13: Architectural ablation against `GPT-OSS` using historical manual seeds. Each variant aggregates at least 32k evaluated prompts; Δ reports absolute ASR change relative to the baseline Mutator.

D.5 Few-Shot Exemplar Sensitivity

We evaluate Mutator sensitivity to the number of in-context exemplars $K \in \{2, 4, 6, 8\}$ using

a DeepSeek-R1-Distill-32B Planner and Mistral-Nemo Mutator. Table 14 shows a shallow non-monotonic pattern: ASR increases from 27.5% at $K = 2$ to 29.5% at $K = 6$, then slightly declines to 29.3% at $K = 8$. This pattern suggests that additional exemplars help until the prompt context begins to constrain variation.

Few-Shot Exemplars (K)	Mean ASR	Δ vs. K=2
2	27.5%	–
4	28.7%	+1.2
6	**29.5%**	**+2.0**
8	29.3%	+1.8

Table 14: Mutator sensitivity to few-shot context size. Each setting aggregates 71.7k–73.7k evaluated prompts; performance peaks at K=6 and slightly declines at K=8, consistent with context saturation.

D.6 Retrieval Batch-Size Sensitivity

We test whether retrieval performance depends on batch granularity by varying batch size from 25 to 200 under the same 4,000-query budget. Figure 7 shows that the relative ordering of methods remains stable across batch sizes. Adaptive Transferability remains the strongest method, indicating that the retrieval gains do not depend on a narrow batch-size choice.

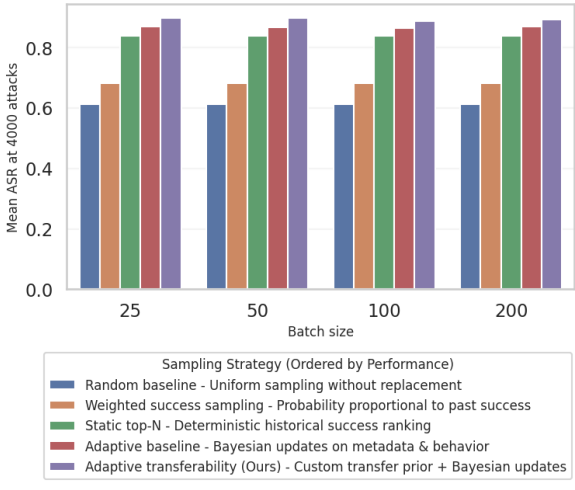


Figure 7: Mean ASR at a fixed 4,000-query budget across retrieval batch sizes. Method rankings remain stable, and Adaptive Transferability consistently achieves the highest ASR.

D.7 Hyperparameter Sensitivity

Adaptive retrieval uses fixed coefficients in three parts of the score: the baseline mixture $\beta = (\beta_1, \beta_2, \beta_3)$ in Equation 3, the transfer prior $w = (w_1, \ldots, w_4)$ in Equation 4, and the online-update parameters in Equation 5. We use $\beta = (0.45, 0.25, 0.30)$, $w = (0.40, 0.30, 0.15, 0.15)$, $\alpha_4 = 0.35$, and $c = 0.06$ for every target. We do not tune these coefficients separately for individual models.

We evaluate sensitivity under the same leave-one-target-out setup, candidate pool, 4,000-query budget, and batch-size schedule used in the main retrieval experiments. We test 25 simplex configurations for β, 71 configurations for w, and one-dimensional sweeps for α_4 and c.

Group	Chosen	ASR	Mean	Range
Eq. 3 β (25)	(.45, .25, .30)	86.7	86.8	86.7–86.8
Eq. 4 w (71)	(.40, .30, .15, .15)	89.4	89.4	88.6–89.8
Eq. 5 α_4	0.35	89.4	89.3	89.0–89.6
Eq. 5 c	0.06	89.4	89.2	88.1–89.6

Table 15: SSensitivity of mean retrieval ASR (%) to each coefficient group. Chosen reports the fixed setting used in the paper; ASR gives its performance; Grid Mean and Range summarize the tested configurations.

Table 15 shows limited sensitivity within the tested ranges. The chosen settings perform close to the mean of each sweep, and the largest observed spread is 1.5 percentage points for the exploration constant c. The transfer-prior sweep ranges from 88.6% to 89.8% ASR, while the α_4 sweep ranges from 89.0% to 89.6%. The reported gains therefore do not depend on a narrow coefficient choice.

The absolute values in Table 15 are slightly lower than the 89.83% headline in Table 4. The sensitivity analysis averages performance over the complete batch-size schedule, whereas Table 4 reports the headline retrieval configuration. The sensitivity conclusions depend on relative variation within each sweep, not on this small difference in aggregation.

We also test whether the fixed transfer-prior weights overfit the eight evaluated models. For each held-out target, nested leave-one-target-out selection chooses w using only the other seven models and then applies the selected weights to the held-out target. The evaluation target therefore does not influence coefficient selection.

Nested selection and the fixed weights produce nearly identical mean performance: 89.3% and 89.4%, respectively. Per-target differences do not exceed 1.5 percentage points. Thus, the fixed coefficients match the performance of weights selected without access to evaluation target, providing no evidence of target-specific coefficient overfitting.

Held-out target	Selected w	Nested	Fixed
Ministral-3-14B	(.40, .40, .20, .00)	100.0	100.0
Mistral-Nemo	(.40, .40, .20, .00)	100.0	100.0
Gemma-3-12B	(.40, .40, .20, .00)	99.8	99.8
Qwen3-30B-A3B	(.40, .40, .20, .00)	98.1	98.1
Llama-3.1-8B	(.40, .20, .20, .20)	96.9	97.3
Phi-4	(.40, .40, .20, .00)	89.2	89.1
Qwen3-4B	(.50, .20, .20, .10)	81.6	83.1
GPT-OSS-20B	(.35, .30, .20, .15)	49.0	47.6
Mean	—	**89.3**	**89.4**

Table 16: Nested leave-one-target-out selection of transfer-prior weights. Nested selects w using only the seven visible models; Fixed uses the single coefficient vector used throughout the paper. Values report target-level ASR (%).

D.8 Seed Replay Failures by Risk Category

We analyze 1,170 seed replays against Mistral-Nemo and Qwen3-4B to identify common failure modes. Overall, 7.7% of replays fail. Explicit refusals account for 57.8% of failures, while off-topic or misunderstood responses account for 42.2%. Table 17 reports failure rates by risk category. Violence and hate and terrorism have the highest observed failure rates (10.7%); guns and illegal weapons has the lowest (4.3%). These results describe the two evaluated targets and should not be interpreted as a universal category ranking.

Risk category	Seed failure rate (%)
Violence and hate	10.7
Terrorism	10.7
Fraud and scams	9.0
Regulated/controlled substances	8.7
Inciting/abetting discrimination	7.5
Self-harm	7.4
Cybercrime and hacking	6.0
Criminal planning	6.0
Sexual content	5.8
Guns and illegal weapons	4.3
Overall	**7.7**

Table 17: Pooled per-category seed failure rate over 1,170 replayed seed attempts against Mistral-Nemo and Qwen3-4B. Failures split into 57.8% refusals and 42.2% off-topic/misunderstood responses.

E Dataset Diagnostics

E.1 Diagnostic Evaluator Agreement

Automated safety evaluation can inherit single-judge bias. We therefore compare DeepSeek-R1 and Gemma-4-31B judge labels across severity thresholds. Figure 8 shows strong agreement on clearly safe and strictly unsafe responses. Most disagreement appears at the partial-compliance boundary, where DeepSeek assigns more sensitive intermediate labels than Gemma. This pattern motivates our use of consensus labels in transferability analysis when we require conservative success criteria.

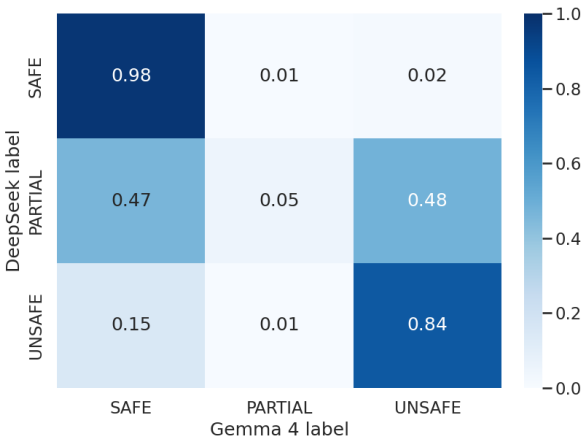


Figure 8: Agreement between judges. Disagreement concentrates around partial compliance, while both judges align on safe and strictly unsafe responses.

E.2 Stochastic Safety Variance

Non-zero decoding temperature can introduce variance into safety evaluations. We estimate this variance by sampling 1,600 attacks, with 200 payloads originally optimized against each of the eight target models. We execute each payload five times against every target, yielding 40,000 generations, and judge each response independently. We then compare the repeated-response mean unsafe rate with the single-pass baseline ASR.

Figure 9 shows close agreement between repeated and single-pass evaluations. Vulnerable targets remain vulnerable under resampling, while robust targets maintain low unsafe rates. This indicates that the measured failures reflect reproducible model behavior rather than decoding artifacts.

E.3 Additional Dataset Statistics

Table 18 reports the full generation breakdown by target model and Mutator. The Planner and Diagnostic Judge remain fixed to DeepSeek-R1-Distill-Qwen-32B, so the comparison isolates the effect of the Mutator on generation yield. The Gemma-3 Mutator achieves higher ASR on every target and produces 111,033 successful attacks, compared with 79,542 for Mistral-Nemo-Instruct. Both Mutators preserve the same target-level robustness or-

Target Model	Mutator: Gemma-3-12B			Mutator: Mistral-Nemo-Instruct		
	Attempts	Successes	ASR	Attempts	Successes	ASR
Mistral-Nemo-Instruct-2407	57,052	20,905	36.6%	72,021	16,448	22.8%
Ministral-3-14B-Instruct-2512	70,109	23,819	34.0%	79,489	20,798	26.2%
Gemma-3-12B	59,358	13,370	22.5%	67,405	10,566	15.7%
Qwen3-4B	59,566	11,228	18.8%	74,369	7,002	9.4%
Llama-3.1-8B-Instruct	61,340	10,323	16.8%	71,830	11,200	15.6%
Qwen3-30B-A3B-Instruct-2507	68,126	11,201	16.4%	80,250	7,273	9.1%
Phi-4	113,894	14,386	12.6%	67,557	4,696	7.0%
GPT-OSS-20B	159,387	5,801	3.6%	91,341	1,559	1.7%
Total / Aggregate	**648,832**	**111,033**	**17.1%**	**604,262**	**79,542**	**13.2%**

Table 18: Dataset generation statistics by target model and Mutator. The Planner and Diagnostic Judge were fixed to DeepSeek-R1-Distill-Qwen-32B, isolating the Mutator effect on generation yield.

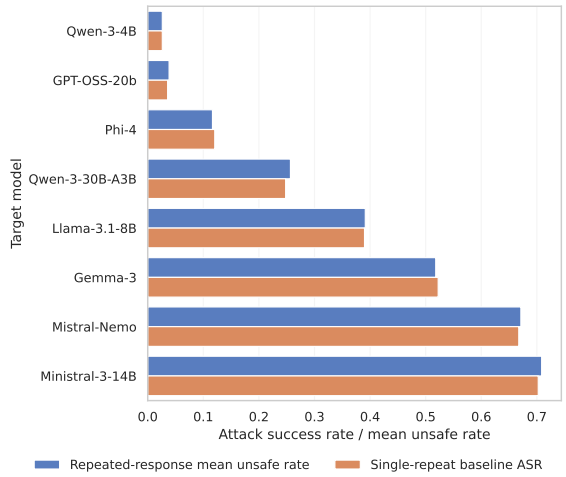


Figure 9: Repeated-response unsafe rates closely match single-pass ASR across target models. The agreement indicates that stochastic decoding variance does not materially change the observed robustness ordering.

dering: Mistral-family models remain the most vulnerable, while GPT-OSS remains the most robust.

We remove 1,967 near-duplicate attacks at a cosine-similarity threshold of 0.95, reducing the corpus from 190,575 to 188,608 attacks. Table 19 shows that 1,966 removals originate from the Gemma-3 Mutator, primarily on Mistral-Nemo and Ministral. The Gemma-3 Mutator therefore produces more successful attacks but also a higher share of semantically redundant outputs.

E.4 Embedding Cluster Structure

We project prompt embeddings with UMAP to inspect whether the generated dataset preserves semantic organization. Figure 10 shows that prompts cluster by risk category, with related categories occupying nearby regions. This structure supports

Source target	gemma-3	Mistral-Nemo
Mistral-Nemo	870	0
Ministral-3-14B	621	1
Qwen3-4B	181	0
Gemma-3-12B	119	0
Phi-4	76	0
Llama-3.1-8B	56	0
Qwen3-30B-A3B	41	0
GPT-OSS-20B	2	0
Total	**1,966**	**1**

Table 19: Near-duplicate removed in deduplication (cosine ≥ 0.95), by Mutator and source target. Removals come almost entirely from the gemma-3 Mutator.

the use of metadata-aware retrieval: categorical labels align with semantic neighborhoods rather than arbitrary annotation noise.

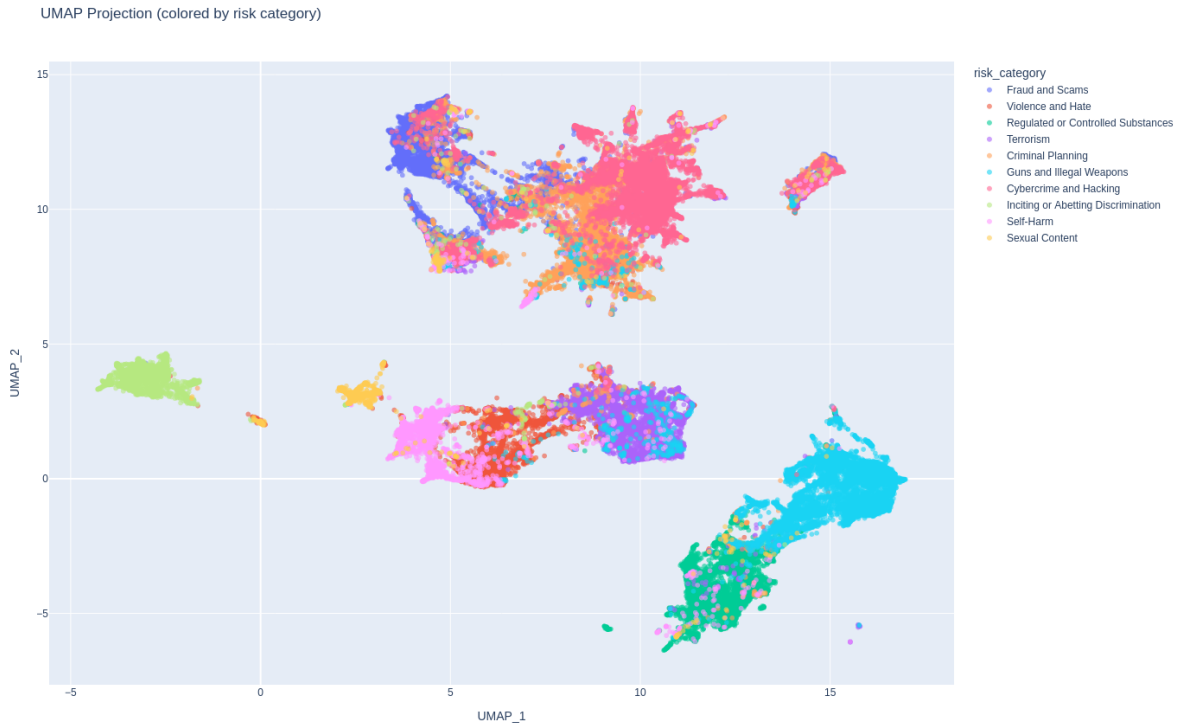


Figure 10: UMAP projection of generated attack embeddings colored by risk category. The embedding space forms category-consistent clusters, indicating that structured metadata aligns with semantic prompt variation.