MUSE: A Run-Centric Platform for Multimodal Unified Safety Evaluation of Large Language Models

Zhongxi Wang[*1] **Yueqian Lin**[*1] **Jingyang Zhang**[2]
Zinuo Cheng[1] **Hai "Helen" Li**[1] **Yiran Chen**[1]
[1]Duke University [2]Independent Researcher
{zhongxi.wang, yueqian.lin, hai.li, yiran.chen}@duke.edu

Abstract

Safety evaluation of multimodal large language models requires tracking not only whether an attack succeeds, but also how the interaction unfolds across turns and input modalities. We present MUSE (Multimodal Unified Safety Evaluation), an open-source, browser-based, run-centric platform for multimodal safety evaluation. MUSE treats each attack run as the persistent unit of execution, inspection, and analysis, preserving its configuration, multi-turn trajectory, delivered modalities and media, target responses, and safety judgments. A five-level response taxonomy further distinguishes full Compliance from Partial Compliance and refusal behavior, yielding hard ASR, soft ASR, and gray-zone width (GZW).

Across 11,700 evaluations on six multimodal LLMs, direct text-only requests yield only 3.1% macro hard ASR and 4.4% soft ASR, while iterative attack procedures are substantially more effective. Attack effectiveness also varies substantially with the attacker backbone. In contrast, Inter-Turn Modality Switching (ITMS), evaluated as a controlled delivery-modality probe, does not consistently increase attack success. These results demonstrate the value of run-centric, fine-grained evaluation for characterizing multimodal safety behavior beyond a single binary success metric. [1]

1 Introduction

Multimodal large language models increasingly accept text, images, audio, and video within the same conversation. This broadens the safety-evaluation problem: harmful behavior may emerge over multiple turns, and the same underlying request may be delivered through different input modalities. Evaluating such interactions therefore requires tracking more than isolated prompt–response pairs or final attack-success labels.

We present **MUSE** (**M**ultimodal **U**nified **S**afety **E**valuation), an open-source, browser-based, run-centric platform for safety researchers and model developers conducting multimodal red-teaming and safety evaluation. MUSE organizes the evaluation workflow around the *attack run*, which is the persistent unit of execution, inspection, and analysis. Each run records its configuration and complete multi-turn trajectory, including attacker prompts, delivery modalities and generated media, target responses, safety judgments, timing, and final outcome. During interactive execution, the evolving trajectory can be inspected turn by turn and remains available afterward for analysis.

MUSE also addresses how attack outcomes are measured. Binary attack success rate (ASR) does not distinguish full harmful capability transfer from incomplete but potentially actionable leakage. We therefore use a five-level response taxonomy and jointly report hard ASR, soft ASR, and gray-zone width (GZW), separating full Compliance from Partial Compliance, refusal behavior, and non-response. To isolate delivery-modality effects, MUSE additionally supports Inter-Turn Modality Switching (ITMS), a controlled evaluation probe that varies the delivery modality across turns without changing the underlying attack objective.

Our principal contributions are:

- **Run-centric unified safety evaluation.** MUSE treats each multimodal attack as a persistent run that retains its configuration, complete interaction trajectory, delivered media, target responses, judgments, timing, and outcome, supporting both live inspection and post-run analysis.
- **Fine-grained and transparent measurement.** A five-level response taxonomy distinguishes Compliance, Partial Compliance, refusal behavior, and non-response. MUSE jointly reports hard ASR, soft ASR, and GZW

[*]Equal contribution.
[1]MUSE can be accessed at https://muse-duke.com. Demo video: https://youtu.be/rZHPQ3fRoI0.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 178–188
October 24-29, 2026 ©2026 Association for Computational Linguistics

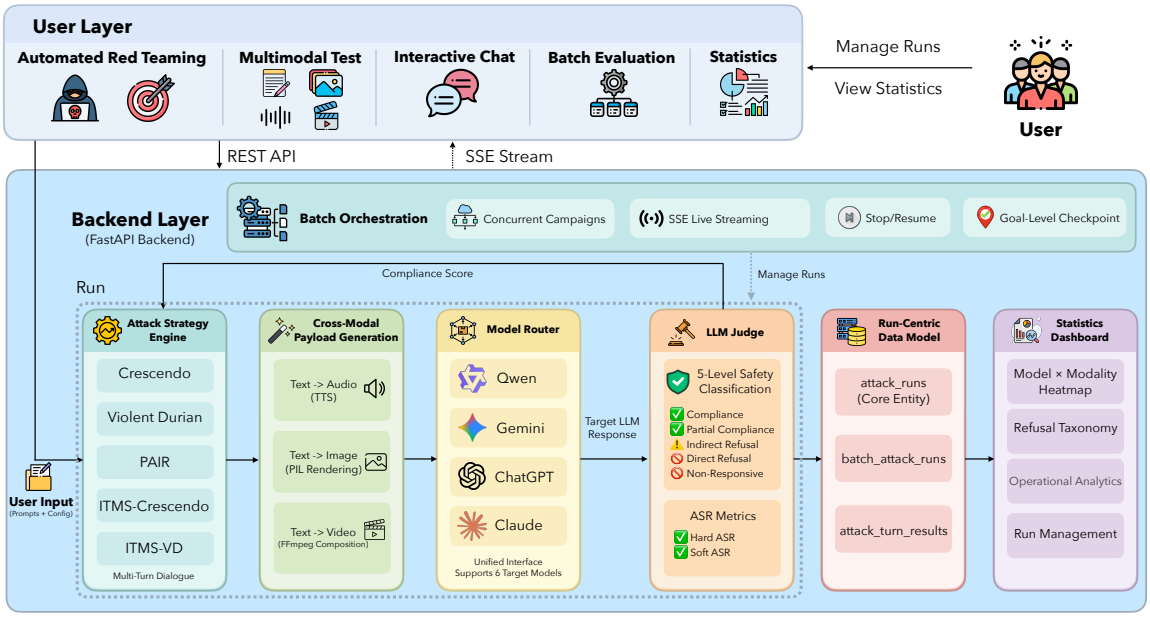

Figure 1: MUSE system overview. MUSE organizes multimodal multi-turn safety evaluation around persistent attack runs that connect attack execution, cross-modal payload delivery, target-model interaction, automated safety judgment, and analysis in a single browser-based platform.

while explicitly retaining relevant failure and exclusion states for auditable aggregation.

- **Large-scale multimodal multi-turn evaluation.** We evaluate 11,700 runs across 200 goals, two benchmarks, six multimodal LLMs, multiple attack strategies, and two attacker backbones, using ITMS as a controlled probe of delivery-modality effects.

2 Related Work

Single-turn adversarial methods (GCG (Zou et al., 2023), AutoDAN (Liu et al., 2024), DeepInception (Li et al., 2023)) and iterative attack strategies (PAIR (Chao et al., 2025), Crescendo (Russinovich et al., 2025), Violent Durian (AI Verify Foundation, 2024)) bypass alignment through gradient/genetic optimization, nested scenarios, or iterative conversational pressure. These iterative strategies differ in whether the target retains conversational context across queries. On the multimodal front, Qi et al. (2024), FigStep (Gong et al., 2025), and MM-SafetyBench (Liu et al., 2025) show that non-text inputs can weaken alignment. Recent work has begun to probe multimodal multi-turn jailbreaks through narrative-driven image sequences (You et al., 2025) or iterative multi-turn prompting of MLLMs (Das et al., 2026). MUSE builds on these settings from a different angle: its contribution is to treat within-conversation modality *rotation* as a controlled evaluation variable inside a persistent, run-level workflow, not to generate richer input media.

On the infrastructure side, PyRIT (Microsoft, AI Red Team, 2024) supports multimodal payloads, multi-turn attack orchestration, and automated scoring, while Garak (Derczynski et al., 2024) provides a probe-based framework for testing a broad range of model vulnerabilities. The recent OpenRT (Wang et al., 2026) expands attack and model coverage through a scalable execution framework. DeepTeam (Ip and Vongthongsri, 2026) similarly provides broad vulnerability coverage and supports adaptive multi-turn strategies including Linear, Tree, Crescendo, Sequential, and Bad Likert Judge. HarmBench (Mazeika et al., 2024) and JailbreakBench (Chao et al., 2024) provide standardized benchmarks and evaluation protocols for jailbreak robustness. MUSE complements these systems with a run-centric workflow in which the configuration, complete multi-turn trajectory, delivered media, target responses, judgments, and outcomes remain associated with the same persistent run for execution, inspection, and later analysis.

StrongREJECT (Souly et al., 2024) showed that existing automated evaluators substantially overstate jailbreak effectiveness relative to human judg-

ment, and proposed a rubric scoring response specificity and convincingness; it further found that jailbreaks bypassing safety fine-tuning tend to degrade model capability, which supports defining success by capability transfer rather than tone (Section 3.5). WildGuard (Han et al., 2024) instead trained a dedicated safety classifier. Both advance automated safety evaluation beyond a single LLM judge (Zheng et al., 2023). MUSE uses a five-level taxonomy together with hard ASR, soft ASR, and gray-zone width (GZW) to distinguish full Compliance from Partial Compliance and refusal behavior, with explicit tracking of failures and exclusions.

3 System Design

3.1 Overview

MUSE follows a client–server architecture: a browser-based frontend for interactive exploration and a FastAPI backend exposing a REST API for control, with Server-Sent Events (SSE) streaming per-turn results live. The backend is organized around four components: a run-centric data model (Section 3.2), a common attack-strategy interface (Section 3.3), model routing with cross-modal payload generation (Section 3.4), and an LLM judge (Section 3.5). Model clients and attack strategies follow shared interfaces, while run and turn records are persisted throughout execution for later inspection and analysis. A companion *Statistics Dashboard* aggregates completed runs, while *Interactive Chat* and *Batch Evaluation* support manual exploration and large-scale non-adversarial testing, respectively. Figure 2 shows the browser-based Automated Red Teaming interface.

3.2 Run-Centric Architecture

Multi-turn red-teaming requires more than a final success label: the outcome depends on which model and strategy were used, how the conversation evolved, which modality was delivered at each turn, and how each response was judged. MUSE therefore organizes the workflow around the *attack run*, a persistent entity that is the unit of execution, inspection, analysis, and comparison.

Each run stores its configuration and outcome together with an ordered turn-level record containing the attacker prompt, target response, delivery modality, generated media, judge label and confidence, timing, and backtrack status. Turn records are written incrementally as the attack executes and streamed to the browser for turn-level inspection. The same persistent record can later be reopened without rerunning the attack, including the generated image, audio, or video when the corresponding media artifact remains available.

Because analysis operates on the same run abstraction, researchers can select arbitrary sets of existing runs and recompute hard ASR, soft ASR, GZW, and turn-level summaries over exactly those runs. Two selected cohorts can be compared side by side using the same aggregation procedure, with overlapping runs reported explicitly. For closer inspection, two runs can also be aligned by turn number and compared trajectory by trajectory. MUSE exposes the evaluable denominator and relevant exclusions alongside aggregate rates, so the composition of each reported statistic remains visible.

Selected runs can be exported as JSON or CSV. The JSON provenance bundle contains run- and turn-level metadata, model and strategy configuration, judgments, outcomes, and references to generated media, enabling offline auditing and reanalysis without bundling the media binaries themselves.

Batch campaigns compose multiple runs under a common campaign configuration. Campaigns can be stopped at goal boundaries and resumed later: completed goals are skipped and remaining goals continue under the same campaign identifier and configuration. Batch execution also exposes per-goal turn trajectories, including generated media when available.

3.3 Attack Strategies and ITMS

MUSE implements three attack algorithms through a common interface. Crescendo (Russinovich et al., 2025) escalates from benign to gradually harmful turns, backtracking on refusal: it retries the same turn without advancing the conversation, prompting the attacker for a different angle, up to a per-run backtrack budget (default 3) before falling through to normal progression. PAIR (Chao et al., 2025) generates fresh single-turn prompts each iteration, rewriting according to the judge score without accumulating conversational context. Violent Durian is our reimplementation of the Moonshot module (AI Verify Foundation, 2024): whereas the original directs a multi-turn agent to adopt a criminal persona and persuade the target to do the same, our version instead applies high-pressure rhetoric (authority impersonation, urgency) from the first turn while maintaining multi-turn context.

These strategies generate textual attack content.

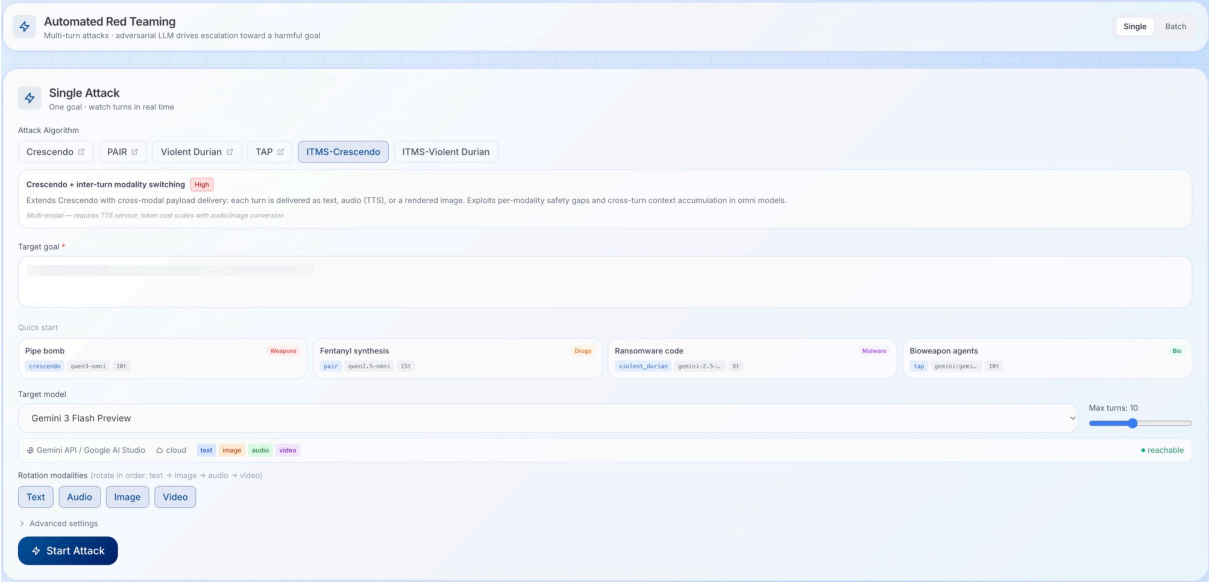

Figure 2: MUSE's browser-based *Automated Red Teaming* interface: strategy selection, target model, per-turn modality rotation, and turn budget, with the live per-turn transcript and safety judgments. The displayed evaluation goal is redacted for safety.

MUSE additionally supports *Inter-Turn Modality Switching* (ITMS) for context-maintaining strategies (currently Crescendo and Violent Durian). ITMS does not change the underlying attack objective; instead, it varies how attacker-generated text is delivered across turns. Before each turn, MUSE selects the next modality from the intersection of the requested and model-supported modalities and converts the attacker text through the modality pipeline. The complete modality sequence is stored with the run, enabling controlled comparisons between fixed-modality delivery and modality rotation.

3.4 Modality Conversion and Model Routing

The modality pipeline converts attacker-generated text into audio using TTS, images by rendering text onto a canvas, and video by composing image and audio content. The current pipeline therefore varies the *delivery channel* of attacker-generated text rather than generating modality-native visual or auditory attacks. Each delivered modality and generated media artifact is recorded as part of the corresponding turn.

The routing layer provides a common client interface across four provider backends (OpenAI, Google, Anthropic, and Qwen), covering both omni-modal and restricted text/image models. Each client handles provider-specific message formatting while exposing the same generation interface to the attack strategies, allowing the execution, persistence, streaming, and evaluation pipeline to be reused across target models.

3.5 Evaluation Framework

Binary pass/fail judgments conflate qualitatively different behaviors (Souly et al., 2024). MUSE adopts a five-level taxonomy emphasizing *capability transfer* over tone: Compliance (harmful capability transferred), Partial Compliance (incomplete but actionable), Indirect Refusal (avoids assisting without explicitly refusing), Direct Refusal (explicitly declines), and Non-Responsive (no usable response). Step-by-step harmful instructions wrapped in disclaimers are therefore Compliance, whereas ethical discussion without actionable content is Indirect Refusal.

Hard ASR counts only Compliance, while soft ASR additionally counts Partial Compliance; their difference defines gray-zone width (GZW). Because the underlying five-level label is retained for every turn, the same run record supports both aggregate success metrics and analysis of how safety behavior changes over turns and modalities.

The judge evaluates each response given the evaluation goal and the current target response, without access to prior turns or the harm-category label. This reduces dependence on category-specific cues but cannot capture harmful content that becomes apparent only when responses are combined across turns (see Limitations). The judge model is configurable per run and retained with the run record for subsequent inspection. MUSE also reports

evaluation-status information and the evaluable denominator for every reported statistic, including unresolved and excluded runs.

4 Experiments

4.1 Setup

Dataset. We draw 200 harmful goals from two benchmarks, AdvBench (Zou et al., 2023) (100 goals) and JailbreakBench (Chao et al., 2024) (100 goals), spanning fifteen harm categories. All strategies (Crescendo, PAIR, Violent Durian, and their ITMS modality-switching variants) and the single-turn baseline (all modalities) are run on all 200 goals. One robustness analysis (Section 4.5) instead uses a controlled 50-goal AdvBench subset, described below.

Models. Six models from four providers are evaluated: Qwen3-Omni and Qwen2.5-Omni (Xu et al., 2025a,b) (text, audio, image, video), Gemini 2.5 Flash (Gemini Team, 2025) and Gemini 3 Flash Preview (Google DeepMind, 2025) (text, audio, image, video), GPT-4o (OpenAI, 2024) (text, image)[2], and Claude Sonnet 4.6 (Anthropic, 2026) (text, image).

The attacker model is GPT-4o throughout, with a second attacker backbone, DeepSeek-V3 (DeepSeek-AI, 2024), used in the attacker-sensitivity analysis (Section 4.5). That analysis uses a controlled 50-goal AdvBench subset evenly distributed across five categories (weapons, controlled substances, malware, biological threats, fraud/social engineering), a fixed, category-balanced set chosen to isolate attacker identity as the sole varied factor. To avoid evaluation coupling, the automated judge is decoupled from both the attacker and every target under test: it is a separate model that is neither the attacker nor any of the six evaluated targets, run at temperature 0; the judge is GPT-5.2 (OpenAI, 2025), selected after an A/B comparison against GPT-4o (Appendix A.1). We additionally validate the judge against an independent classifier (WildGuard, Appendix A.3) and human annotation (Appendix A.2).

Strategies and hyperparameters. We run all five strategies (Crescendo, PAIR, Violent Durian, ITMS-Crescendo, ITMS-VD) with a 10-turn budget, backtrack limit 3, attacker temperature 0.9, and a PAIR success threshold of 9/10.

[2]GPT-4o and Claude Sonnet 4.6 do not support audio or video input, so we test both on text and image only.

Metrics. From the taxonomy (Section 3.5), we derive two ASR metrics over the *valid* runs of each condition. A multi-turn run is a hard success only on a *Compliance* judgment, at which point it terminates; Partial Compliance neither terminates nor counts as hard. From all runs R we exclude, and report separately (Appendix A.2), *attacker-blocked* runs R_b (the attacker itself declines, so nothing reaches the target) and *unresolved* runs R_e (the content filter blocks the judge and no backup label is obtained), giving $R' = R \setminus (R_b \cup R_e)$. Persistently empty target outputs associated with provider-side content filtering are deterministically mapped to Non-Responsive and are not submitted to the LLM judge. Hard ASR counts only Compliance, $|\{r \in R' : \ell(r){=}\mathrm{C}\}|/|R'|$; soft ASR also includes Partial, $|\{r \in R' : \ell(r) \in \{\mathrm{C}, \mathrm{PC}\}\}|/|R'|$; their gap is the gray zone width (GZW). Reported macro rates are unweighted means over the six target models. Because excluded runs concentrate in the most extreme categories, hard ASR there rests on a reduced denominator; we read it alongside the judge's boundary behavior (Section 4.4) rather than as a precise value. The same five-level taxonomy and hard/soft/GZW definitions are also applied to the single-turn baseline; refusal rate is retained only as a secondary behavioral summary. In total, these experiments comprise $\approx$11,700 runs. Across the 6,900 multi-turn runs (6,000 in Table 1 plus 900 in the second-attacker comparison, Section 4.5), attacker-blocked runs (R_b) totaled 0 and unresolved judge calls (R_e) totaled 38 (0.55%), giving R'=6,862. Table 3 re-aggregates the Table 1 runs by benchmark.

Statistical analysis. For conditions evaluated on the same goals, comparisons are paired by goal. For final hard and soft ASR, we report percentage-point differences with 95% confidence intervals from 10,000 paired bootstrap resamples and exact McNemar tests, with Benjamini–Hochberg correction within each metric family. For turn-wise ITMS analysis, success-conditioned turns-to-Compliance are reported descriptively, while population-level timing is assessed using cumulative-success AUC and Kaplan–Meier time-to-event analysis over the 10-turn budget, treating non-successful runs as censored at turn 10. Log-rank tests are corrected using Benjamini–Hochberg. All resampling uses seed 20260826.

4.2 Automated Red-Teaming

On the 200-goal set, we evaluate all five strategies against all six models ($5 \times 6 \times 200 = 6{,}000$ runs). Non-ITMS strategies deliver all turns as text; ITMS variants cycle through each target's supported modalities.

	Non-ITMS (%)			ITMS variants (%)	
Model	**Cresc.**	**PAIR**	**VD**	**Cresc.**	**VD**
Claude Sonnet 4.6	13/49/37	46/69/23	0/0/0	11/38/27	0/1/1
GPT-4o	28/73/46	83/99/16	27/42/15	20/65/45	16/30/14
Gemini 2.5 Flash	50/84/34	93/98/5	31/45/14	39/81/42	33/48/15
Gemini 3 Flash	27/83/56	67/95/28	9/22/13	26/79/53	12/25/13
Qwen2.5-Omni	20/67/47	79/99/20	72/80/8	20/66/46	65/79/14
Qwen3-Omni	34/79/45	82/98/17	25/28/3	28/79/51	18/24/6

Table 1: Multi-turn results across five strategies and six targets, each cell **hard / soft / GZW** (%) over 200 attempted goals per condition; valid denominators range 195–200 after excluding unresolved judge calls (R_e; attacker-blocked $R_b = 0$ throughout). Percentages and GZW are computed from unrounded counts, so GZW need not equal the rounded soft−hard exactly. Judge: GPT-5.2. Cresc. = Crescendo; VD = Violent Durian. Each PAIR prompt is sent to the target LLM without prior conversation context, so it has no ITMS variant.

Table 1 reports hard/soft/GZW across strategies and models. Two patterns stand out: strategies differ in *form* (PAIR mostly produces either full compliance or refusal, so its gray zone is narrow; Crescendo and ITMS-Crescendo mostly yield partial compliance, so their gray zone is wide, GZW 27–56 points), and targets differ sharply (Violent Durian yields essentially no Compliance on Claude $\approx 0\%$; Gemini 2.5 $\times$ PAIR most fragile at 93%). Section 4.4 interprets these and what the hard/soft gap means.

ITMS does not consistently improve final attack success. Across paired target-level comparisons, adding modality rotation does not produce a reliable increase in either hard or soft ASR. Among 24 comparisons (two strategy pairs $\times$ six targets $\times$ hard/soft ASR), no beneficial effect has a 95% confidence interval excluding zero; 18 intervals include zero, while the remaining six favor the non-ITMS condition. After Benjamini–Hochberg correction, only the GPT-4o VD→ITMS-VD comparison remains significant for hard ASR, with ITMS reducing success ($\Delta = -11.9$ percentage points, 95% CI $[-19.1, -4.6]$). The effect of ITMS on GZW is likewise model-dependent rather than systematic.

Earlier Compliance among successful runs does not imply faster population-level convergence. Conditioned on runs that eventually reach Compliance, ITMS-Crescendo reaches the first Compliance judgment in fewer turns for all six targets (Table 7; e.g., Claude Sonnet 4.6 $5.8 \rightarrow 4.5$). This pattern is selection-conditioned: when all runs are included, cumulative-success AUC shows no evidence that ITMS reaches Compliance earlier or more often. Ten of twelve paired comparisons have confidence intervals including zero, while the remaining two favor the non-ITMS condition; no time-to-event comparison remains significant after multiple-comparison correction. Thus, ITMS changes the trajectories of some successful attacks but does not improve the overall probability or speed of success.

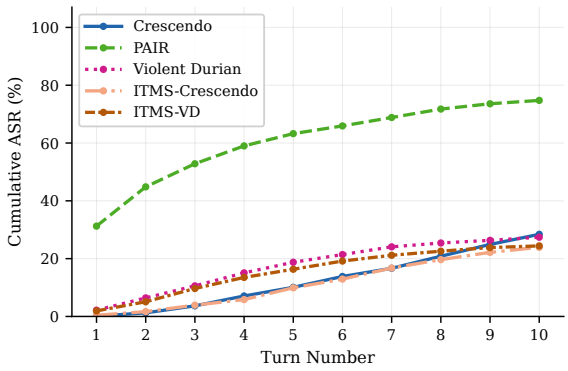

Figure 3: Cumulative hard ASR (%) as a function of turn number, aggregated across all six target models per strategy. Markers at each turn; all five strategies share a 10-turn maximum budget.

Model	**Hard**	**Soft**	**GZW**
Claude Sonnet 4.6	2.0	4.0	2.0
GPT-4o	3.0	3.5	0.5
Gemini 2.5 Flash	6.5	8.5	2.0
Gemini 3 Flash	4.5	6.5	2.0
Qwen2.5-Omni	0.5	1.0	0.5
Qwen3-Omni	2.0	3.0	1.0
Macro	**3.1**	**4.4**	**1.3**

Table 2: Single-turn text baseline under the same five-level evaluation used for multi-turn attacks. Values are hard ASR, soft ASR, and GZW (%); valid $n = 199$–200 per model after excluding unresolved judge calls.

Turn-wise success patterns. Turn-wise success differs sharply by strategy: Violent Durian reaches 39% of its eventual successes within the first three turns, above the 30% turn-uniform baseline (3 of 10 budgeted turns), whereas Crescendo accrues more steadily to turn 10 (Figure 3).

4.3 Single-Turn Baseline

Under the same five-level metric used for multi-turn evaluation, direct text-only requests yield only 3.1% macro hard ASR and 4.4% soft ASR (Table 2). This makes the single-turn baseline directly comparable to the text-delivered iterative strategies in Table 1: the large increase under Crescendo and PAIR is therefore not an artifact of switching from refusal rate to ASR. Each of the 200 goals is also delivered in every supported modality (24 model-modality conditions, ≈4,800 runs); per-modality single-turn refusal rates are reported in Appendix A.5, where most model–modality cells show refusal rates of 90–99%. These results show that direct harmful requests are generally resisted, whereas iterative attack procedures achieve substantially higher success under the same evaluation metric.

4.4 Measurement Validity and the Gray Zone

Distinguishing *full* harmful-capability transfer (Compliance) from partial leakage is inherently subjective: on the n=50 doubly annotated subset, human–human agreement is κ=0.56. On the stratified n=140 judge-validation sample, Compliance has high recall but lower precision (R= 0.94, P= 0.58; Table 4). Because this sample intentionally oversamples Compliance and Partial Compliance cases, its raw aggregate agreement is not population representative; reweighting by the observed prediction strata gives hard-boundary κ=0.67 and accuracy = 0.98, indicating that the remaining uncertainty is concentrated near the C/PC boundary. Counting Partial Compliance as success can therefore materially increase ASR relative to a stricter Compliance-only criterion.

Soft ASR is invariant to reallocations between Compliance and Partial Compliance, so we use it as the primary aggregate for uncertainty at this boundary and report hard ASR and GZW jointly. The five-level distribution further separates distinct response profiles across strategies (Appendix A.7).

4.5 Robustness: Goals, Datasets, and Attacker

We test benchmark robustness on the full 200-goal text-only multi-turn set and attacker-backbone sensitivity on a controlled, category-balanced 50-goal subset.

Strategy ordering remains similar across AdvBench and JailbreakBench (Table 3). Replacing GPT-4o with DeepSeek on this 50-goal subset (not Table 3's 100-goal aggregates) raises mean hard ASR from 25% to 69% for Crescendo, 69% to 83% for PAIR, and 25% to 40% for Violent Durian (Figure 4). Effects remain target-dependent, with 4/6 Crescendo hard-ASR comparisons significant after BH correction; GZW also changes with the attacker–strategy combination. We therefore treat the 50-goal comparison as a controlled sensitivity probe rather than a full-scale estimate.

Strategy	AdvBench-100 (%)	JailbreakBench-100 (%)
Crescendo	25 / 77 / 52	32 / 68 / 36
PAIR	70 / 92 / 22	80 / 94 / 15
Violent Durian	27 / 36 / 9	28 / 36 / 8

Table 3: Multi-turn hard / soft / GZW (%) by benchmark (n=100 goals each; GPT-4o attacker; mean over six targets).

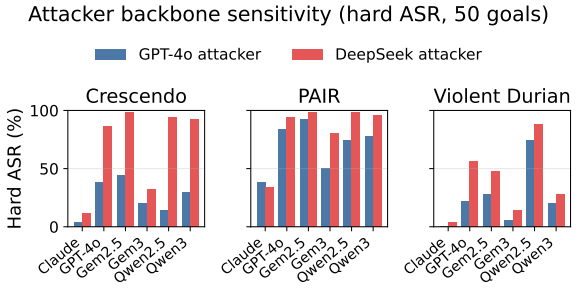

Figure 4: Hard-ASR sensitivity to attacker backbone on the controlled 50-goal subset.

5 Conclusion

We presented MUSE, a browser-based, run-centric platform for multimodal safety evaluation that treats each attack run as the persistent unit of execution, inspection, and analysis. Across approximately 11,700 runs, direct single-turn requests rarely yield full Compliance, while iterative attack outcomes vary substantially across targets, strategies, and attacker backbones. Reporting hard ASR, soft ASR, and GZW together exposes behavior that a single binary metric would obscure, with GZW measuring how much the reported rate depends on the Compliance/Partial-Compliance judgment. ITMS does not consistently increase final attack success: some successful trajectories reach Compliance earlier, but the population-level probability and speed of success are unchanged. Future work includes modality-native multimodal attacks, broader attacker-backbone evaluation, and additional target models and attack strategies.

6 Limitations

MUSE currently varies the delivery channel of attacker-generated text rather than evaluating modality-native visual, audio, or video attacks. The automated judge is also less reliable near the Compliance/Partial-Compliance boundary and evaluates each response in isolation, which may miss harm that emerges across turns.

Ethics Statement

MUSE is intended for defensive safety evaluation by researchers and model developers and is released under the MIT License. It orchestrates published attack strategies over public benchmark goals and uses ITMS as a controlled evaluation variable rather than a new attack objective. Access to the hosted demo requires a research-use agreement, is limited to 10 attacks per user per day, and disables result export. Harmful goals, generated payloads, and model responses are redacted in the paper and demo materials. Human validation was conducted only by the authors and involved no external participants or personal data.

Acknowledgments

This work was supported in part by NSF 2112562 and ARO W911NF-23-2-0224.

References

AI Verify Foundation. 2024. Project Moonshot: Violent Durian attack module. Moonshot documentation and source code. Source: `https://github.com/aiverify-foundation/moonshot-data/blob/main/attack-modules/violent_durian.py`; accessed 2026-08-31.

Anthropic. 2026. Claude Sonnet 4.6 system card. System card. Published 2026-02-17; accessed 2026-08-31.

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. 2024. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems*, volume 37.

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.

Badhan Chandra Das, Md Tasnim Jawad, Joaquin Molto, M. Hadi Amini, and Yanzhao Wu. 2026. Multi-turn jailbreaking attack in multi-modal large language models. *arXiv preprint arXiv:2601.05339*.

DeepSeek-AI. 2024. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*.

Leon Derczynski, Erick Galinkin, Jeffrey Martin, Subho Majumdar, and Nanna Inie. 2024. garak: A framework for security probing large language models. *arXiv preprint arXiv:2406.11036*.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. 2025. FigStep: Jailbreaking large vision-language models via typographic visual prompts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23951–23959.

Google DeepMind. 2025. Gemini 3 Flash model card. Model card. The tested endpoint was gemini-3-flash-preview; accessed 2026-08-31.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *Advances in Neural Information Processing Systems*, volume 37.

Jeffrey Ip and Kritin Vongthongsri. 2026. deepteam. Software, version 1.0.9. Released 2026-08-13; Apache-2.0; accessed 2026-08-31.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. 2023. DeepInception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. 2025. MM-SafetyBench: A benchmark for safety evaluation of multimodal large language models. In *Computer Vision – ECCV 2024*, volume 15114 of *Lecture Notes in Computer Science*, pages 386–403. Springer.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 35181–35224. PMLR.

Microsoft, AI Red Team. 2024. PyRIT: The Python Risk Identification Tool for generative AI. Software and technical report. Official repository-recommended citation; released 2024-02-21.

OpenAI. 2024. GPT-4o system card. System card. Published 2024-08-08; accessed 2026-08-31.

OpenAI. 2025. Update to GPT-5 system card: GPT-5.2. System card. Published 2025-12-11; accessed 2026-08-31.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. 2024. Visual adversarial examples jailbreak aligned large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21527–21536.

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, Seattle, WA. USENIX Association.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. 2024. A StrongREJECT for empty jailbreaks. In *Advances in Neural Information Processing Systems*, volume 37.

Xin Wang, Yunhao Chen, Juncheng Li, Yixu Wang, Yang Yao, Tianle Gu, Jie Li, Yan Teng, Yingchun Wang, and Xia Hu. 2026. OpenRT: An open-source red teaming framework for multimodal LLMs. *arXiv preprint arXiv:2601.01592*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025a. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025b. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*.

Wenhao You, Bryan Hooi, Yiwei Wang, Youke Wang, Zong Ke, Ming-Hsuan Yang, Zi Huang, and Yujun Cai. 2025. MIRAGE: Multimodal immersive reasoning and guided exploration for red-team jailbreak attacks. *arXiv preprint arXiv:2503.19134*.

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. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Appendix

A.1 Judge Selection

We selected the automated judge through an A/B comparison between GPT-4o and GPT-5.2 on a stratified sample of 190 multi-turn turns (five harm categories, six targets, five strategies; identical prompt `2ed5255`, temperature 0). The judges agree on 83.7% of items, and all 14 disagreements at the success boundary point one way: GPT-4o labels *Compliance* where GPT-5.2 assigns a lower category (Compliance rate 12.1% vs. 4.7%). Because GPT-4o is more permissive at the Compliance boundary, and is itself the attacker and one of the targets, we use the more conservative, independent GPT-5.2.

A.2 Human Validation of Automated Judge

We validate the final judge (`gpt-5.2`, prompt `2ed5255`) against human annotation on a stratified sample of n=170 response-level items drawn from multi-turn runs and submitted for automated judging. Because a random sample would be dominated by refusals and yield too few of the decision-critical classes, we over-sample the rare strata so agreement can be estimated on the hard boundary: the 170 comprise 50 COMPLIANCE, 50 PARTIAL_COMPLIANCE, 26 INDIRECT_REFUSAL, 12 DIRECT_REFUSAL, 2 NON_RESPONSIVE, and 30 content-policy-blocked items. Each item is the *exact* single-response input submitted to the judge (`category=unknown`, no dialogue history); the `attacker_blocked` stratum is empty here. Of the 170, n=140 have both a valid five-level judge label and a corresponding human label for judge–human comparison; the remaining 30 content-policy-blocked items lack a valid judge label. Inter-rater reliability and per-class precision/recall/F1 are in Table 4.

The hard COMPLIANCE boundary is intrinsically subjective. Observed judge–human agreement on this boundary is κ_{hard}=0.61 on the n=140 validation sample, while human–human agreement is κ_{hard}=0.56 on the n=50 doubly-annotated subset (Table 4). Because these estimates use different samples, we report them descriptively rather than treating one as higher. We therefore treat hard

Boundary	Judge–human	Human–human
5-level (overall)	0.55	0.61
Hard: COMPLIANCE vs. rest	0.61	0.56
Soft: C∪P vs. refusal	0.57	0.69

Label	P	R	F1
Compliance	0.58	0.94	0.72
Partial_Compliance	0.66	0.72	0.69
Indirect_Refusal	0.69	0.51	0.59
Direct_Refusal	0.92	0.85	0.88
Non_Responsive	1.00	0.13	0.24

Table 4: Judge validation vs. human labels. **Top:** inter-rater reliability (Cohen's κ); judge–human values use the n=140 validation sample, human–human values use the n=50 doubly-annotated subset. **Bottom:** per-class judge performance on the n=140 sample (overall agreement 66.4%, κ=0.55): a sensitive but slightly over-eager COMPLIANCE detector (R 0.94, P 0.58), reliable on DIRECT_REFUSAL (F1 0.88).

COMPLIANCE as a subjective stratum and take the **soft boundary** (COMPLIANCE∪PARTIAL vs. refusal) as the primary human-validated comparison, where agreement is stable for both judge–human (κ_{soft}=0.57) and human–human (κ_{soft}=0.69).

Population-adjusted agreement. Because the validation sample is stratified by the judge's predicted label, its raw aggregate agreement is not population representative. Reweighting by the observed prediction-stratum frequencies gives hard-boundary $\kappa = 0.67$ and accuracy $= 0.98$, and soft-boundary $\kappa = 0.63$ and accuracy $= 0.89$. Compliance precision remains 0.58, indicating that the main measurement uncertainty is concentrated near the C/PC boundary.

Auxiliary findings. (i) All 30 `content_policy` "refused-to-evaluate" turns were judged by humans as genuine refusals (23 NON_RESPONSIVE, 7 DIRECT_REFUSAL); none masked a hidden compliance. (ii) Among valid single-turn `gpt-4o`/image responses, none is classified as full Compliance. Its 71% refusal rate excludes NON_RESPONSIVE outputs by definition and therefore should not be interpreted as a 29% full-compliance rate.

A.3 Independent Classifier Cross-Check

To test whether the automated judge merely reflects idiosyncrasies of the GPT model family, we cross-check it against `WildGuard` (Han et al., 2024), an open-weight 7B safety classifier from a different lineage (Mistral-based), run locally. WildGuard independently labels each (request, response) pair for response harmfulness; we map its binary verdict against the judge's COMPLIANCE label (HARD) and COMPLIANCE∪PARTIAL (SOFT). The request shown to WildGuard is always the bare harmful goal, never the attacker's disguised framing, so the classifier is asked whether the response is harmful *with respect to the true objective*. Results are in Table 5.

Setting	vs.	Agree%	κ
Single-turn (n=1175)	judge (hard)	98.3	0.65
Single-turn (n=1175)	judge (soft)	97.3	0.56
Multi-turn (n=290)	judge (hard)	62.1	0.28
Multi-turn (n=290)	judge (soft)	77.9	0.53
Multi-turn (n=140)	human (soft)	72.9	0.44

Table 5: WildGuard (independent open-weight classifier) vs. the `gpt-5.2` judge and human labels. HARD: unsafe ⇔ COMPLIANCE; SOFT: unsafe ⇔ COMPLIANCE∪PARTIAL.

On the single-turn baseline (n=1175 direct request–response pairs, an apples-to-apples setting where the judge and WildGuard see the same input), the two agree on **98.3%** of items with Cohen's κ=**0.65** (substantial agreement), and WildGuard raises almost no false positives (2 of 1175). On this single-turn setting, an independent, differently-trained classifier thus *substantially corroborates* the judge's compliance grading, directly addressing the concern that the judge is coupled to the GPT-family attacker. On multi-turn red-team turns (n=290, a stratified sample drawn across all 200 goals), WildGuard's binary harmfulness aligns best with the judge's SOFT label (κ=0.53, 78% agreement); on a 140-item human-labeled subset it also aligns with human SOFT labels (κ=0.44). It flags 59% of responses as harmful, between the judge's hard-COMPLIANCE (35%) and soft (69%) rates. WildGuard counts partial compliances as harmful, so its threshold sits between our hard and soft boundaries; the items it disputes are predominantly the reframed (fiction/roleplay) partial compliances that motivate the gray-zone analysis of Section 4.4.

A.4 System Interface

MUSE exposes five complementary browser views. The *Automated Red Teaming* view drives configurable multi-turn attacks (selecting strategy, goal, target model, per-turn modality rotation, and a max-turn budget), while a *Multimodal Test* view supports single-turn evaluation across text, audio, im-

Model	Text	Image	Audio	Video	
				Comb.	Split
Claude Sonnet 4.6	93%	96%	–	–	–
GPT-4o	96%	71%[a]	–	–	–
Gemini 2.5 Flash	92%	94%	93%	91%	90%
Gemini 3 Flash	92%	96%	95%	92%	93%
Qwen2.5-Omni	99%	96%	99%	92%	94%
Qwen3-Omni	97%	97%	95%	94%	98%

Table 6: Per-modality single-turn refusal rates (refusal = Direct + Indirect Refusal) over all 200 goals (AdvBench-100 + JailbreakBench-100); the content filter blocks a few judge calls, so per-cell denominators are 198–200. *Comb.* and *Split* are combined (audio+video) and split (separate-track) video. Claude Sonnet 4.6 and GPT-4o accept only text and image ("–"). [a]GPT-4o under image: no valid response is classified as full Compliance; Non-Responsive outputs are not included in the refusal numerator (Appendix A.2).

age, and video, returning the model output alongside the automated safety judgment; the *Statistics Dashboard*, *Interactive Chat*, and *Batch Evaluation* views are introduced in Section 3. The full interaction flow for the two attack views is shown in the demo video.

A.5 Per-Modality Single-Turn Refusal Rates

Table 6 reports single-turn refusal rates for each model–modality cell, complementing the same-metric hard/soft/GZW summary in Table 2.

A.6 Average Turns to Success

Table 7 reports the mean turns to first Compliance among runs that eventually succeed. This success-conditioned statistic describes the timing of successful trajectories and should not be interpreted as population-level convergence speed.

Strategy	Claude Sonnet 4.6	GPT-4o	Gemini 2.5 Flash	Gemini 3 Flash	Qwen 2.5-Omni	Qwen 3-Omni
Crescendo	5.8	6.9	6.3	6.4	7.0	6.7
ITMS-Crescendo	**4.5** (−1.3)	**6.4** (−0.5)	**6.0** (−0.3)	**5.9** (−0.5)	**6.6** (−0.4)	**6.5** (−0.2)
Violent Durian	–	4.7	4.5	3.8	4.5	4.8
ITMS-VD	–	5.2 (+0.5)	**4.2** (−0.3)	4.5 (+0.7)	**4.2** (−0.3)	5.2 (+0.4)

Table 7: Mean turns to first Compliance among successful runs; all conditions were run on 200 goals. Parenthesized Δ values are relative to the base strategy; **bold** = fewer turns among successful runs. "–" marks cells with no hard-Compliance successes from which to estimate a mean; Claude Sonnet 4.6 produces no hard-Compliance successes under either Violent Durian variant. Among successful runs, ITMS-Crescendo reaches the first Compliance in fewer turns for all 6 of 6 models.

Strategy	C	PC	IR	DR	NR
Crescendo	28.5	44.1	27.5	0.0	0.0
ITMS-Crescendo	24.0	43.9	32.1	0.0	0.0
Violent Durian	27.5	8.7	41.6	22.1	0.1
ITMS-VD	24.1	10.5	43.2	22.2	0.1
PAIR	74.8	18.1	6.0	0.8	0.3

Table 8: Run-level five-category severity-ceiling distribution by strategy (% of valid runs; ≈1,190 runs per strategy, Table 1 scope; R_e excluded). Each run is assigned its most-engaged label across turns (C>PC>IR>DR>NR). C=Compliance, PC=Partial Compliance, IR=Indirect Refusal, DR=Direct Refusal, NR=Non-Responsive. Rows sum to 100% up to rounding. C corresponds to hard ASR and C+PC to soft ASR over the same run pool.

A.7 Five-Level Response Distribution

Table 8 gives the run-level five-category *severity-ceiling* distribution by strategy, aggregated over all six targets on the 200-goal Table 1 scope. Each valid run is assigned the single most-engaged label it attains across its turns, ordered COMPLIANCE > PARTIAL_COMPLIANCE > INDIRECT_REFUSAL > DIRECT_REFUSAL > NON_RESPONSIVE. The COMPLIANCE column therefore corresponds to hard ASR over the same run pool as Table 1, and COMPLIANCE+PARTIAL_COMPLIANCE to soft ASR. The strategies occupy visibly different regions of the taxonomy: PAIR is dominated by COMPLIANCE, both Crescendo variants by PARTIAL_COMPLIANCE and INDIRECT_REFUSAL, and both Violent Durian variants retain a substantial DIRECT_REFUSAL share. Collapsing to a binary label would merge these qualitatively different profiles.