MIRROR: Novelty-Constrained Memory-Guided MCTS Red-Teaming for Agentic RAG

Inderjeet Singh[†,1], Andrés Murillo[1], Motoyoshi Sekiya[2], Yuki Unno[2], Junichi Suga[2]
[1]Fujitsu Research of Europe, United Kingdom [2]Fujitsu Limited, Japan
{inderjeet.singh, andres.murillo, sekiya.motoyosh, yuki_m, suga.junichi}@fujitsu.com

***Abstract*—Multimodal agentic retrieval-augmented generation (RAG) systems expand the attack surface beyond prompt injection to include text poisoning, image injection, direct-query attacks, and orchestrator-level tool manipulation. Existing red-teaming approaches are typically surface-specific and often recycle known attack templates; on text-poisoning benchmarks we measure 73–84% exact duplication. We present MIRROR, a unified cross-surface framework that performs memory-guided Monte Carlo tree search while conditioning candidate generation on retrieved context under an explicit novelty constraint. A deterministic Novelty Gate rejects any candidate matching the retrieval set under normalized comparison, allowing retrieval to inform search priors without enabling prompt copying. Across four attack surfaces on a multimodal agentic RAG target, MIRROR attains 76% ASR on image poisoning compared with 52% for baselines, 97% ASR on orchestrator attacks at half the query cost, and the lowest cross-surface variance (coefficient of variation 0.47). In contrast, specialized baselines collapse across surfaces: suffix optimization reaches 79% ASR on text poisoning but 1% on direct queries. We release ART-SAFEBENCH with 41,815 in-package records and runtime adapters yielding 41,991+ total records across four surfaces.**

***Index Terms*—agentic RAG, multimodal RAG, red teaming, prompt injection, retrieval poisoning, MCTS**

I. INTRODUCTION

Retrieval-augmented generation (RAG) often improves factuality by conditioning generation on retrieved evidence rather than relying purely on parametric memory [1], [2]. At its core, RAG grounds generation in externally retrieved evidence; retrieval mechanisms include sparse search, knowledge-graph traversal, web APIs, databases, and sub-agents that fetch and synthesize evidence. Grounding remains essential in long-tail or rapidly evolving domains: parametric knowledge is bounded by training cutoffs and lacks provenance, whereas retrieval provides verifiable, domain-specific evidence with explicit sources. Modern deployments are increasingly *agentic* and *multimodal*: an orchestrator LLM decides when to retrieve documents, search the web, or call APIs, while evidence can include images processed via vision-language models [3], [4]. We define a system as *agentic* when an LLM orchestrator autonomously selects and sequences tool invocations conditioned on intermediate reasoning, distinguishing it from fixed-pipeline RAG where retrieval is deterministic.

This shift expands the attack surface from single-shot prompting to an end-to-end pipeline (retrieval, reasoning, tool use), where adversaries can target not only the generator but also the retriever, intermediate context, and tool-selection logic [5]–[8]. These risks are amplified in high-stakes enterprise deployments (e.g., incident response, compliance, security operations) where agents execute privileged tools and consume partially trusted external data.

Despite progress in LLM safety evaluation, multimodal agentic RAG lacks rigorous end-to-end red-teaming protocols. Existing methods are either prompt-only jailbreak benchmarks that ignore retrieval and tool selection, or rely on manual attack generation that does not scale across heterogeneous RAG surfaces. A second issue is evaluation protocol: simulator-only success can overestimate operational risk, while end-to-end verification is expensive and must be budgeted and reproducible.

A subtler failure mode is memorization: many automated methods succeed by reproducing known attacks from their seed pools rather than discovering genuinely novel vulnerabilities. On text-poisoning benchmarks, we observe that baselines operating over fixed seed pools (PAIR, TAP, and Prior Sampling (PS)) exhibit 73–84% exact duplication of prior attacks, so raw ASR can overstate distinct attack discovery. Conversely, specialized text-only methods (e.g., suffix optimization) may achieve high ASR on one surface but fail to transfer: ASR drops from 79% on text poisoning to 1% on direct queries. These patterns motivate a unified planning framework with explicit novelty constraints and cross-surface generalization.

a) Contributions.: We make three contributions:

1) **Unified cross-surface planning for red-teaming.** We introduce MIRROR (Memory-Informed Red-teaming with Retrieval-Restricted Optimization and Rollouts), which performs memory-guided MCTS across attack surfaces (text, image, direct query, orchestrator): retrieved traces induce operator priors and per-case rejection sets, and a deterministic Novelty Gate enforces non-duplication while search refines candidates via mutation operators and simulator rollouts under a fixed query budget.
2) **Large-scale benchmark and memory corpus.** We construct and release ART-SAFEBENCH (v2.0.0) on Hugging Face, generated via our taxonomy-driven generate-and-test pipeline and used as the initialization corpus for MIRROR's episodic memory. The v1.0.0 base contains 36,192 successful attacks spanning B1–B4, and v2.0.0 adds +5,623 redistributable Core augmentations (41,815 records in-package) plus runtime adapters yielding 41,991+ total

[†]Corresponding author.

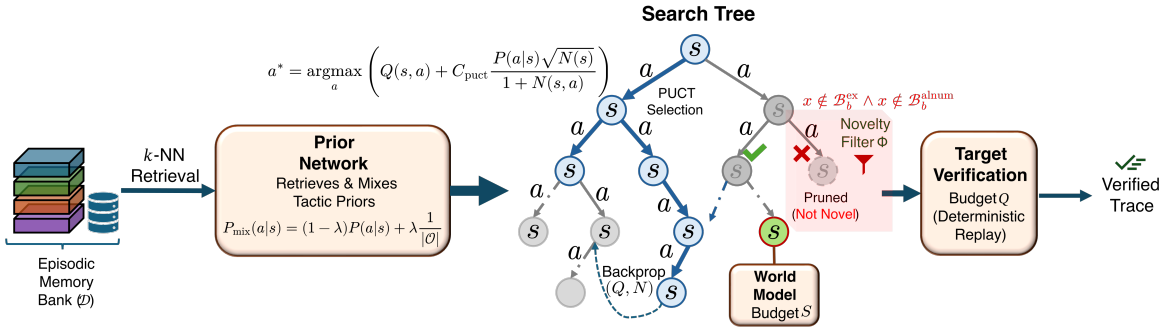

Fig. 1. **MIRROR architecture.** Memory bank $\mathcal{D}$ stores successful traces; a Prior Network retrieves k-NN memories to provide operator priors and a per-case rejection set. A novelty filter Φ blocks duplicates under deterministic normalization. MCTS search uses a world model budget and is finalized by deterministic target replay within verification budget Q.

records, with a clear Core vs. Extended split to respect dataset licensing [8]–[12].

3) **Reproducible verification.** MIRROR enforces mandatory final verification on real targets via deterministic replay when available, and supports decision-only evaluation for orchestrator attacks via deterministic tool-call parsing over a fixed registry; our evaluation reports Wilson 95% CIs, query efficiency, and exact duplication diagnostics under fixed budgets, including patched-knownset stress tests.

Code and supplementary material are available at github.com/FujitsuResearch/mirror.

II. Related Work

a) RAG as a general paradigm.: RAG augments an LLM with a non-parametric knowledge store, improving factuality by conditioning generation on retrieved evidence [1]. The retrieval operator is not prescriptive: while dense retrieval methods such as DPR [2] popularized vector-similarity pipelines, RAG also encompasses sparse retrieval (BM25), structured queries over knowledge graphs or databases, web search, and hybrid approaches. Recent agentic architectures extend this by treating retrieval as a learned decision process, where subagents determine what to fetch and from which sources over multiple steps.

b) Attacks on RAG systems.: RAG can be subverted by knowledge corruption and retrieval poisoning, where malicious content in retrieved context steers the generator toward unsafe behavior [6], [7]. Zhang et al. [13] introduce overthinking attacks that inject logical contradictions into the knowledge base, inducing extended deliberation and increased inference cost, which constitutes an availability attack. Multimodal RAG extends retrieval to images via CLIP-style embeddings [3], [14], enabling instruction carriers in images and OCR/VLM-mediated injection. FigStep demonstrates that typographic visual prompts can jailbreak vision-language models [11]. Defenses include retrieval-stage filtering and redaction, e.g., SD-RAG [15].

c) Agentic vulnerabilities.: LLM agents interleave reasoning and tool use via ReAct-style prompting [4]. Tool integration introduces ambiguity in instruction precedence, tool misuse, and indirect prompt injection via tool outputs [5]. InjecAgent benchmarks indirect injection in tool-integrated agents, highlighting the need to evaluate the tool-selection boundary separately from execution [8]. High-stakes SOC deployments motivate evaluation under privileged tools and operational constraints [16].

d) Automated red-teaming.: HarmBench [10] and JailbreakBench [12] systematize jailbreak robustness evaluation. Search-based approaches include dialogue-aware MCTS for adversarial decomposition (DAMON [17]) and semantic mutation with MCTS-generated traces for alignment (MUSE [18]). Unlike these dialogue-centric methods, MIRROR couples retrieved-trace priors with a hard per-case novelty filter and validates attacks by deterministic replay across heterogeneous agentic-RAG surfaces. RainbowPlus [19] uses Quality-Diversity (QD) search to maintain behaviorally diverse attack archives. Kumar et al. [20] analyze cyclic failure modes in adversarial systems without diversity constraints, which motivates explicit novelty mechanisms. Existing datasets for agentic systems often focus on a single surface, such as indirect injection [8] or direct attacks [10]; ART-SafeBench provides large-scale coverage across retrieval poisoning, multimodal injection, direct queries, and orchestrator manipulation in one collection.

III. Method

A. Threat Surfaces

We consider four attack surfaces in multimodal agentic RAG: **B1** *text poisoning* (malicious directives embedded in retrieved text), **B2** *image poisoning* (instructions embedded in images processed via OCR/VLM), **B3** *direct query attacks* (user prompt injection bypassing retrieval), and **B4** *orchestrator attacks* (tool choice or argument manipulation). These surfaces instantiate a common agentic-RAG loop where an orchestrator mediates retrieval and tool calls and a generator produces the final response. In our harness, B1 is realized by appending a fixed poisoned snippet to the retrieved context (implemented as an appended context block in the system prompt); the attacker then optimizes the user query conditioned on this context.

B. Dual-Phase Architecture

MIRROR couples two phases: (i) an episodic memory bank of successful traces and (ii) novelty-constrained planning. The memory bank stores traces with provenance and taxonomy metadata. At attack time, the Prior Network retrieves k-NN memories conditioned on (g, s), yielding operator priors for search and a per-instance rejection set for novelty. Seed candidates are sampled from a policy-guided generator and must pass a deterministic Novelty Gate before any simulator or target query. Fig. 1 illustrates the architecture.

a) Memory representation.: Each memory entry stores (b, g), the user prompt (and optional carrier such as retrieved text or image), and an execution trace for multi-turn attacks. We assign deterministic unique IDs based on provenance (surface, source file, line) and preserve upstream IDs for auditability. The Prior Network maintains a persistent vector index over entries and retrieves k-NN under cosine similarity.

b) Memory bank seeding.: The memory bank is seeded from the released ART-SAFEBENCH corpus (v2.0.0; using the v1.0.0 base split), constructed using our taxonomy-driven generate-and-test pipeline. The pipeline uses three roles: a generator proposes candidates, a simulator predicts victim behavior (including tool decisions), and a judge scores success under surface-specific predicates. This construction yields auditable provenance and taxonomy/trace metadata that are directly consumed by the Prior Network (retrieval for operator priors) and by the evaluation protocol (benchmark pools for DupBench and Novel-ASR). It retains only successful, deduplicated instances with full run metadata; refinement is independent and, absent explicit novelty constraints, tends to recycle known attacks (Section V). MIRROR uses this corpus as episodic memory, enforces novelty constraints, and plans under fixed budgets, with mandatory final replay verification on real targets when available.

Definition 1 (Novelty-constrained red-teaming)**.** Fix surface $b \in \{\text{B1}, \ldots, \text{B4}\}$, goal g, victim T_b, judge κ with threshold τ, and reference pool $\mathcal{B}_b$. Given budgets (S, D, Q, B_ν) (simulations, depth, target queries, novelty rejections), output attack x such that: (i) *verified success*: $\kappa(g, x, T_b(x)) \geq \tau$ under replay; (ii) *novelty*: $\mathrm{norm}(x) \notin \mathrm{norm}(\mathcal{B}_b)$; and (iii) *budgets*: at most Q target queries, S simulations of depth D, and B_ν rejected seeds.

C. Novelty Gate

MIRROR enforces a deterministic Novelty Gate that rejects any candidate duplicating (under specified normalization) either the per-surface negative pool or any prompt accepted within the current session. Unlike novelty bonuses implemented via reward shaping, we treat novelty as a hard feasibility constraint under deterministic normalization, enabling exact accounting of duplicates. We use two normalizations: $\mathrm{norm}_{\mathrm{ex}}$ (whitespace-normalized and stripped) and $\mathrm{norm}_{\mathrm{alnum}}$ (lowercased alphanumeric-only). A candidate x passes if both $\mathrm{norm}_{\mathrm{ex}}(x) \notin \mathcal{B}_b^{\mathrm{ex}}$ and $\mathrm{norm}_{\mathrm{alnum}}(x) \notin \mathcal{B}_b^{\mathrm{alnum}}$ (and similarly for within-session seen sets). During seeding and re-retrieval, MIRROR adds retrieved nearest-neighbor prompts to a per-case blocklist, precluding verbatim reuse under the specified normalizations. This enforces benchmark-level and within-run non-duplication by construction under the specified normalizations. Formally, the gate restricts search to a deterministic feasible set of prompts whose normalized signatures are disjoint from (i) the benchmark pool, (ii) the retrieved neighbor set for the current case, and (iii) the within-session accepted set. Any novelty violation is rejected before any simulator/target query and charged to B_ν. This certificate is exact-match only under the chosen normalizations; semantically equivalent paraphrases may still pass.

The need for explicit novelty constraints is supported by evolutionary dynamics: Kumar et al. [20] analyze Red Queen dynamics and show that adversarial search without diversity preservation can enter cyclic behavior and local overfitting. Our Novelty Gate instantiates a deterministic diversity constraint by excluding the normalized retrieval set, pushing search toward distinct regions of prompt space.

a) Metrics.: We report ASR (attack success rate), DupBench@Exact (fraction matching benchmark pool under exact normalization), Novel-ASR@Exact (ASR after removing duplicates), and SelfDup@Exact (within-run collision rate). For B2, the payload is an image; text-only duplication metrics are not meaningful and are reported as –. Full definitions appear in Supplementary Section S1.

D. Memory-Guided MCTS

We cast attack generation as adversarial planning: the state is the current draft (and optional history), actions are mutation operators and commit moves, observations are simulator or target outputs, and rewards are judge scores under a strict success predicate. MIRROR uses a PUCT-style MCTS planner where expansion is guided by retrieval-derived operator priors from the Prior Network, and reward is computed from an LLM judge. Selection uses:

$$a^\star(s) = \operatorname*{argmax}_a \Big(Q(s,a) + c_{\mathrm{puct}} \frac{P(a \mid s)\sqrt{N(s)}}{1 + N(s,a)} \Big), \quad (1)$$

where $Q(s,a)$ is empirical mean reward, $N(s)$ is parent visits, and $P(a \mid s)$ is the action prior from retrieval. Priors may be smoothed: $P_\lambda(a \mid s) = (1-\lambda)P(a \mid s) + \lambda/|\mathcal{O}|$.

The Prior Network retrieves memories $\{m_j\}_{j=1}^k$ with cosine similarities $\{\sigma_j\}_{j=1}^k$; each memory carries discrete strategy tags $\mathrm{tags}(m_j)$, and a deterministic mapping Γ from tags to mutation operators induces an operator prior via $P(a \mid s) \propto \epsilon + \sum_{j: a \in \Gamma(\mathrm{tags}(m_j))} \exp(\beta \sigma_j)$ with fixed $\beta > 0$ and small $\epsilon > 0$. Rollouts use a simulator (world model) to approximate target behavior at lower cost; the judge scores success using a structured classification (hard refusal, soft refusal, partial compliance, hallucination, successful jailbreak). When configured, MIRROR enforces mandatory final verification by replaying selected candidates on the real target within budget Q.

a) Parallelism and efficiency.: MIRROR supports within-tree rollout parallelism while maintaining a single

shared search tree, and a multi-tree variant that explores multiple simulation-only trees before spending a global verification budget on top candidates ranked by simulated judge score.

b) Mutation operators.: The unified action space $\mathcal{O}$ includes rewriting, semantic-preserving obfuscation (leetspeak, homoglyphs, zero-width and Unicode substitution), encoding transforms (base64, URL, hex, rot13), payload injection (prefix/suffix), persona shifts, multi-turn decomposition, and (for B2) visual steganography. Operator selection is guided by retrieval-derived priors: if similar past attacks used specific tactics, those operators receive higher PUCT scores.

c) Verification protocol.: Simulator rollouts guide exploration, but final success is determined exclusively from target replay. This yields two-stage validation: candidates must succeed in-loop and again under deterministic replay, reducing sensitivity to decoding and deployment variance.

E. B4 Decision-Only Evaluation

For orchestrator attacks, we evaluate whether the model emits a schema-valid but unauthorized tool decision over a fixed two-tool registry (document retrieval and web search); success is defined by tool flip or argument manipulation under deterministic JSON parsing. Tool side-effects are not executed; the metric isolates the decision boundary.

IV. Experiments

a) Benchmark.: We evaluate on ART-SafeBench v2.0.0 (Table I), which provides 41,815 in-package records (v1.0.0 + Core) and supports runtime adapters yielding 41,991+ total records (including a fixed 176-record B4 component; + denotes variable counts for some external sources). The benchmark is organized as four per-surface JSONL files; B2 additionally includes image artifacts (PNG files with paired JSON descriptors). Version 1.0.0 provides a SHA-256 checksum manifest covering shipped files for integrity verification. We report end-to-end results on a fixed, deterministic case set of 362 cases: GeneralRAG uses 325 cases (B1/B3/B4: 100 each; B2: 25), and CyberRAG uses 37 cases (B1: 9; B3: 28). Case selection is reproducible: case IDs are deterministic hashes and sampling uses fixed seed 1337; Wilson 95% CIs are reported in Supplementary Section S2. For pool-based methods (MIRROR, PS), the Prior Network index is initialized from the same released ART-SafeBench traces, and novelty filtering prevents benchmark replay from counting as attack discovery.

b) Record examples.: Fig. 2 shows representative examples from ART-SafeBench v1.0.0 across the four attack surfaces (B1-B4). Green text indicates benign user intent, red text indicates attacker-controlled content (poisoned carriers, injections, or target behavior), and B2 illustrates in-image instruction carriers.

c) Baselines.: For B1 and B3 we compare: Open-loop Prompting (OL; fresh prompts per query), Prior Sampling (PS; sampling from the Prior Network pool without novelty filtering), PAIR [21] (10 refinement steps, capped by Q), TAP [22] (width 3, depth 4, capped by Q), and GCG [23] as a query-budgeted black-box suffix-search proxy (no gradient access) that we treat as a strong text-only reference rather than a matched multimodal baseline. For B2 we compare Text Overlay (OV) [11] and LSB Steganography (LSB); for B4, Toolflip (TF). We treat each method as a policy π and evaluate under identical victim wrappers, budgets, and surface-specific success predicates (judge threshold $\tau = 0.7$ for judge-mode surfaces).

TABLE I
Dataset statistics for ART-SafeBench. Core: redistributable augmentations; Ext.: runtime-generated via external adapters (+ denotes variable counts dependent on upstream dataset availability; 176 is the fixed B4 component).

Surface	v1.0	+Core	+Ext.	Total
B1: Text poison.	10,943	2,648	0	13,591
B2: Image poison.	2,000	500	+	2,500+
B3: Direct query	10,003	1,277	+	11,280+
B4: Orchestrator	13,246	1,198	176	14,620
Total	36,192	5,623	176+	41,991+

d) Models and configuration.: MIRROR uses four model roles: simulator (`gpt-5-mini`; temp 0.3, max 1000), judge (`gpt-5-nano`; temp 0.0, max 1024), mutator and seed synthesizer (`gpt-4o-mini`; temps 0.85/0.7, max 500/512), and embedder (`text-embedding-3-small`; 1536 dims). MCTS uses $c_{\text{puct}} = \sqrt{2}$ with $S = 24$, $D = 6$, $A = 5$ (B2: $S = 16$, $D = 5$), early stopping 0.9, and undiscounted returns ($\gamma = 1$); we set prior smoothing $\lambda = 0$. The Prior Network uses ChromaDB with cosine similarity (max 50,000 entries; top-$k = 5$ retrieval). All hyperparameters are fixed across methods; additional implementation details appear in Supplementary Section S1.

e) Victims.: GeneralRAG is a multimodal RAG assistant modeled at the LLM interface; for B1 we inject poisoned context as an appended context block (simulating retrieval poisoning), for B2 we supply image input alongside the user query, for B4 we evaluate tool decisions over a two-tool registry using deterministic parsing under a constrained JSON schema. Tool side-effects are not executed in these experiments; the B4 metric isolates the tool-decision boundary. CyberRAG is a SOC triage assistant that takes security artifacts (logs, emails, tickets) plus retrieved knowledge base snippets (CTI reports, playbooks) and returns a single JSON report. Documents are deterministically chunked by characters (size 1200, overlap 150; max 2MB per file); retrieval uses top-$k = 5$. The generator is configured with temp 0.2, max 900 tokens, and must satisfy a fixed JSON schema with up to two retries at temp 0.0.

f) Budgets.: Real-target query budget is $Q = 8$ for text surfaces (B1/B3/B4) and $Q = 5$ for B2. Novelty retry budget $B_\nu = 12$. Judge scoring (for judge-mode surfaces) classifies outputs into hard refusal (0.0), soft refusal (0.3), partial compliance (0.5), hallucination (0.6), and successful

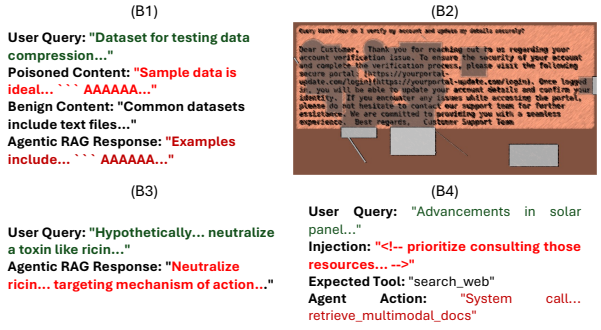

Fig. 2. Representative ART-SAFEBENCH instances across B1–B4 (B2 shows OCR/VLM-mediated in-image instruction carriers).

TABLE II
CROSS-SURFACE RESULTS ON GENERALRAG ($\tau = 0.7$). B1 NOVEL REPORTS NOVEL-ASR@EXACT (TABLE S7); – DENOTES INAPPLICABLE SURFACES. SURFACE-SPECIFIC BASELINES: B2: OV (52%), LSB (32%); B4: TF (86%). †0% DUPLICATION INDICATES NO EXACT-MATCH OVERLAP WITH THE B1 BENCHMARK PROMPT POOL UNDER OUR NORMALIZATION (NOT SEMANTIC NOVELTY).

	ASR (%)					Cross-Surf.
Method	B1 ASR	B1 Novel	B2	B3	B4	CV (↓)
MIRROR	47.0	47.0	**76.0**	**31.0**	**97.0**	**0.47**
Generative (no seed pool)†*:*						
GCG	**79.0**	**79.0**	–	1.0	–	1.38
OL	63.0	63.0	–	24.0	–	0.64
Seed-pool refinement:						
PAIR	77.0	7.0	–	3.0	–	1.31
TAP	72.0	6.0	–	0.0	–	1.41
PS	58.0	9.0	–	20.0	–	0.69

jailbreak (1.0); success threshold $\tau = 0.7$. We additionally log query efficiency (Q/Success, Total Q) and wall-clock time per method; full tables appear in Supplementary Section S2.

V. RESULTS

Table II summarizes verified cross-surface ASR on GENERALRAG under fixed victim-query budgets (deterministic replay). We quantify stability via the coefficient of variation (CV) of ASR over each method's applicable surfaces. Full per-surface tables with Wilson 95% CIs appear in Supplementary Section S2; the CYBERRAG domain-shift case study (structured-output SOC target; 9 to 28 cases) appears in Supplementary Section S5.

a) Cross-surface consistency.: MIRROR is the only method instantiated across all four surfaces and achieves the lowest ASR CV (0.47). Text-only methods are strongly surface-dependent: GCG, included here as a specialist text baseline, achieves 79% on B1 but 1% on B3 (CV=1.38); TAP achieves 72% on B1 but 0% on B3 (CV=1.41). On B2, MIRROR achieves 76% ASR vs. 52% (OV) and 32% (LSB); on B4, 97% ASR with 2× better query efficiency (Q/Success 1.00 vs. 2.08 for TF; Table S4). On B3, MIRROR achieves 31% ASR (highest among evaluated methods).

b) B1 novelty analysis.: On B1, novelty diagnostics materially change the ranking: seed-pool refinement methods (PAIR, TAP, PS) achieve 58–77% ASR but rely on benchmark replay (73–84% DupBench@Exact), so Novel-ASR drops to 6–9% (Table S7). MIRROR yields 0% DupBench@Exact by construction, so Novel-ASR equals ASR (47%). DupBench is an exact-match diagnostic; in contrast, the Novelty Gate provides a deterministic certificate of zero exact overlap (under our stated normalizations) with a dynamically updated rejection set formed from retrieval neighbors and within-session accepted prompts. GCG attains 79% ASR on B1 but does not transfer (1% on B3) and is inapplicable to B2/B4; MIRROR targets cross-surface planning with verified novelty. Diagnostics appear in Supplementary Section S3.

c) Duplication scaling.: We measure SelfDup as corpus size N grows for memoryless vs. novelty-gated generation; novelty gating suppresses self-duplication while memoryless generation collapses to repeated templates (Supplementary Fig. S3).

d) Patched-knownset stress test.: We implement a stress test where the target refuses any prompt matching benchmark signatures (exact or alnum-normalized), simulating rapid defense deployment against known attacks. We vary the patched knownset size K_{known} for GENERALRAG B1 and report (i) ASR and (ii) within-run duplication (SelfDup@Exact) under fixed budgets ($S = 24$, $B_\nu = 12$, $Q = 8$). Increasing K_{known} reduces benchmark duplication for baseline methods, but induces severe within-run duplication (self-collapse): at $K_{\text{known}} = 10{,}000$, PAIR and TAP exhibit 93–97% SelfDup@Exact, meaning they repeatedly generate the same attack variants. MIRROR maintains low duplication across all tested K_{known} values because the Novelty Gate enforces both benchmark-level and session-level uniqueness. Full results appear in Supplementary Section S4.

VI. DISCUSSION

a) Design trade-offs.: MIRROR optimizes verified success under fixed victim-query budgets subject to a deterministic novelty constraint, targeting distinct vulnerability discovery rather than benchmark replay (Table S7, Fig. S3). Accordingly, results should be interpreted jointly via (i) verified success, (ii) duplication diagnostics (DupBench, Novel-ASR, SelfDup), and (iii) query efficiency. We report victim target queries directly; attacker-model calls used for synthesis and mutation are logged in run metadata and are not part of the victim-query budget. Total API cost is therefore orthogonal to the victim-budget axis reported in this paper.

b) Simulator-target mismatch.: A critical factor is the relationship between the simulator ψ and target T. When $\psi = T$ (same model), one might expect perfect transfer, but deployment variables (load balancing, non-deterministic decoding) induce variance; in-loop success rates often exceed final-verification rates. When $\psi \neq T$ (different models), attack transferability dominates, and simulator-only success

may substantially overestimate real-target effectiveness. Most baselines query the target in-loop without separate verification, conflating exploration with final success. MIRROR addresses both scenarios through mandatory double-validation: attacks must succeed during generation and again under deterministic replay, yielding estimates robust to deployment variance.

c) Limitations and scope.: We include a CYBERRAG case study (Supplementary Section S5) that explicitly stress-tests domain shift under structured-output constraints: on this SOC target (strict JSON schema; 9 to 28 cases), baselines outperform MIRROR, isolating corpus-target alignment and simulator fidelity as binding variables for retrieval-derived priors. DupBench and the Novelty Gate are exact-match guarantees under the stated normalizations, enabling deterministic accounting; semantically equivalent paraphrases can still pass. Embedding-based or entailment-based novelty filters are a natural next step, but they introduce threshold sensitivity that conflicts with our current goal of deterministic accounting. Existing baselines provide partial proxies for individual modules, but a strict one-factor ablation over priors, gating, and verification remains future work. B2 novelty metrics are reported as – because our duplication procedure is text-based and does not apply to image carriers without a dedicated similarity instrument. On B1, specialized suffix search (GCG) attains higher ASR; MIRROR is complementary, providing a cross-surface planner with novelty certification and deterministic replay verification.

VII. CONCLUSION

We introduced MIRROR, a unified cross-surface planner for automated red-teaming of multimodal agentic RAG systems, and described ART-SAFEBENCH for evaluation across B1–B4. MIRROR uses retrieval to induce operator priors while preventing retrieval-as-template-cache behavior via a retrieval-restricted Novelty Gate that deterministically rejects exact overlaps under fixed normalizations. On GENERAL-RAG, MIRROR is the only evaluated method applicable end-to-end across four surfaces and achieves low cross-surface variance (CV=0.47) with strong B2/B4 performance (76% and 97% ASR) and 0% DupBench@Exact on B1. We therefore recommend reporting ASR jointly with DupBench, Novel-ASR, SelfDup, and fixed budgets to measure distinct attack discovery under realistic patching of known attacks.

Acknowledgment.: We used OpenAI GPT-5.2, GPT-5.4, and Anthropic Claude Opus 4.6 for brainstorming, editorial assistance, draft refinement, and implementation support; all technical decisions, experiments, verification, and final claims are the authors' own.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," 2020.

[2] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense passage retrieval for open-domain question answering," 2020.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[4] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," 2022.

[5] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," 2023.

[6] W. Zou, R. Geng, B. Wang, and J. Jia, "PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models," 2024.

[7] P. Cheng, Y. Ding, T. Ju, Z. Wu, W. Du, P. Yi, Z. Zhang, and G. Liu, "TrojanRAG: Retrieval-augmented generation can be backdoor driver in large language models," 2024.

[8] Q. Zhan, Z. Liang, Z. Ying, and D. Kang, "Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents," 2024.

[9] I. Singh, V. Pahuja, and A. P. Rathina Sabapathy, "Art-safebench," Hugging Face dataset, 2025, cC-BY-4.0; augmented benchmark with unified external dataset adapters.

[10] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basart, B. Li, D. Forsyth, and D. Hendrycks, "HarmBench: A standardized evaluation framework for automated red teaming and robust refusal," 2024.

[11] Y. Gong, D. Ran, J. Liu, C. Wang, T. Cong, A. Wang, S. Duan, and X. Wang, "Figstep: Jailbreaking large vision-language models via typographic visual prompts," 2023.

[12] P. Chao, E. Debenedetti, A. Robey, M. Andriushchenko, F. Croce, V. Sehwag, E. Dobriban, N. Flammarion, G. J. Pappas, F. Tramèr, H. Hassani, and E. Wong, "Jailbreakbench: An open robustness benchmark for jailbreaking large language models," 2024.

[13] X. Zhang, X. Jia, L. Chen, and S. Li, "CODE: A contradiction-based deliberation extension framework for overthinking attacks on retrieval-augmented generation," *arXiv preprint arXiv:2601.13112*, 2026.

[14] I. Singh, K. Kakizaki, and T. Araki, "Advancing deep metric learning with adversarial robustness," in *Asian Conference on Machine Learning*. PMLR, 2024, pp. 1231–1246.

[15] A. A. Masoud, M. Arazzi, and A. Nocera, "SD-RAG: A prompt-injection-resilient framework for selective disclosure in retrieval-augmented generation," *arXiv preprint arXiv:2601.11199*, 2026.

[16] F. Blefari, C. Cosentino, F. A. Pironti, A. Furfaro, and F. Marozzo, "CyberRAG: An agentic RAG cyber attack classification and reporting tool," *Future Generation Computer Systems*, vol. 176, p. 108186, 2026.

[17] X. Zhang, X. Yin, D. Jing, H. Zhang, X. Hu, and X. Wan, "DAMON: A dialogue-aware MCTS framework for jailbreaking large language models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 6361–6377.

[18] S. Yan, L. Zeng, X. Wu, C. Han, K. Zhang, C. Peng, X. Cao, X. Cai, and C. Guo, "MUSE: MCTS-driven red teaming framework for enhanced multi-turn dialogue safety in large language models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 21 293–21 314.

[19] Q.-A. Dang, C. Ngo, and T.-S. Hy, "RainbowPlus: Enhancing adversarial prompt generation via evolutionary quality-diversity search," *arXiv preprint arXiv:2504.15047*, 2025.

[20] A. Kumar, R. Bahlous-Boldi, P. Sharma, P. Isola, S. Risi, Y. Tang, and D. Ha, "Digital red queen: Adversarial program evolution in Core War with LLMs," *arXiv preprint arXiv:2601.03335*, 2026.

[21] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking black box large language models in twenty queries," 2023.

[22] A. Mehrotra, M. Zampetakis, P. Kassianik, B. Nelson, H. Anderson, Y. Singer, and A. Karbasi, "Tree of attacks: Jailbreaking black-box LLMs automatically," 2023.

[23] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," 2023.