DART: DEFENSE-AWARE RED TEAMING WITH A CRITICAL CUE BANK FOR OPEN-WORLD ADVERSARIAL EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous red-teaming agents can inspect source code, invoke attack libraries, retrieve external knowledge, and use execution feedback, yet these capabilities do not ensure that the agent can identify relevant knowledge and operationalize it as an executable attack tailored to a particular defense. We introduce DART, a defense-aware open-world red-teaming framework centered on a Critical Cue Bank (CCB). The CCB structures profiling evidence into actionable cues that guide toolbox attack adaptation, external retrieval, and research-to-executable attack construction. When no directly applicable attack is available, DART retrieves strategies developed for related defense mechanisms and adapts their principles to the target's interface and threat-model constraints. Across 51 published defense implementations from AutoAdvExBench, DART increases the overall attack success rate from 5.9% to 43.1% with Qwen and from 31.4% to 52.9% with Claude Opus, while producing valid evaluations for all 51 targets. It also reduces LLM token usage by 84-95% relative to the corresponding baselines. An open-world baseline without cue-conditioned selection and adaptation yields little or no improvement in attack success rate, suggesting that the main bottleneck is not access to attack knowledge but its target-conditioned operationalization.

1 INTRODUCTION

Recent advances in large language model (LLM) agents have enabled increasingly complex security workflows by combining source-code analysis, tool use, external knowledge retrieval, and execution feedback He et al. (2026); Abramovich et al. (2024); Deng et al. (2024). These capabilities naturally extend to autonomous adaptive red teaming Zhou et al. (2026) of adversarial defenses Zhan et al. (2025). However, real defense implementations expose a distinct bottleneck. Prior work reports a substantial gap between attacks constructed in simplified settings and those targeting real defense implementations, and even explicitly providing defense-specific attack knowledge does not necessarily improve attack success Carlini et al. (2025). More broadly, studies of LLMs with retrieved or long-context information show that relevant evidence can remain underused and that simply increasing the available context does not necessarily improve performance Joren et al. (2025); Liu et al. (2024); Hsieh et al. (2024). Retrieval-augmented methods provide a general mechanism for incorporating external knowledge into LLM systems Lewis et al. (2020); Chen et al. (2024), but access alone does not determine how that knowledge should be used for a specific target. In autonomous adaptive red teaming, the central challenge

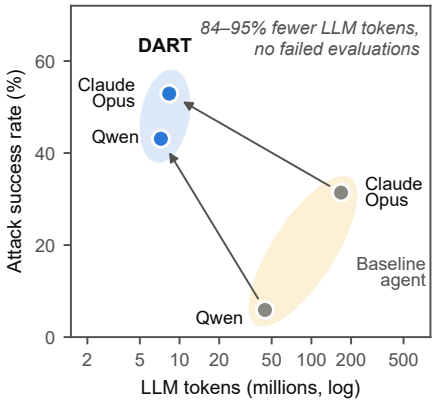


Figure 1: Attack success versus LLM token usage on 51 published defense implementations from AutoAdvExBench. DART achieves higher attack success with 84–95% fewer LLM tokens while completing valid evaluations for all defenses.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

is therefore to identify which target properties are attack-relevant, select knowledge that addresses them, and adapt that knowledge to the target's execution constraints. We structure this process as *target understanding*, *knowledge selection*, and *target-specific operationalization*.

To this end, we introduce **DART**, a defense-aware open-world red-teaming framework centered on a **Critical Cue Bank (CCB)**. DART first profiles the target defense and structures the resulting attack-relevant characteristics in the CCB, which guides attack selection, open-world retrieval, and target-specific adaptation. In Tier-1, the agent selects and adapts attacks from established adversarial attack toolboxes Nicolae et al. (2018); Papernot et al. (2016); Rauber et al. (2017); Kim (2020), while Tier-2 expands the attack space through external research. This allows target-specific information to guide not only which attack knowledge is selected, but also how it is converted into an executable attack.

Across 51 published defense implementations from AutoAdvExBench, DART increases the attack success rate from **5.9% to 43.1%** with Qwen and from **31.4% to 52.9%** with Claude Opus. DART also completes valid evaluations for defenses that the previously proposed agent fails to evaluate, achieving complete coverage of all 51 defenses with both backbones. In contrast, adding open-world access to the baseline alone improves Qwen only to 9.8% and leaves Claude Opus unchanged at 31.4%. At the same time, DART reduces LLM token usage by **84–95%** relative to the corresponding baselines (Figure 1). Together, these results support the distinction between knowledge access and target-conditioned attack construction.

Our main contributions are:

- **Structured adaptive red teaming.** We introduce DART, a framework that explicitly connects target-specific information with executable attack construction by integrating target profiling, toolbox-based attack construction, and open-world research expansion.
- **Critical Cue Bank for target-conditioned attack construction.** We organize profiling-derived attack-relevant characteristics into the CCB, an attack-oriented representation that guides toolbox attack construction and open-world knowledge utilization.
- **Large-scale evaluation and analysis.** Across 51 published defense implementations from AutoAdvExBench evaluated with two LLM backbones, DART improves attack success and evaluation coverage while reducing LLM-side computation. Defense-level analyses further characterize the effectiveness and limitations of structured attack construction.

2 Related Work

Adaptive and Autonomous Adversarial Evaluation. Adversarial defense evaluation has progressed from standardized attack suites Croce & Hein (2020) to target-specific adaptive attacks tailored to individual defense mechanisms Tramer et al. (2020). Subsequent work has automated adaptive attack discovery Yao et al. (2021) and developed diagnostics for identifying and improving attack failures Pintor et al. (2022). More recently, LLM agents have extended this direction toward autonomous evaluation by combining source-code analysis, tool use, and execution feedback for end-to-end attack construction Gioacchini et al. (2025); Muzsai et al. (2024); Zou et al. (2026). AutoAdvExBench evaluates such capabilities against real defense implementations Carlini et al. (2025). In parallel, studies of retrieval-augmented and long-context LLMs have shown that access to relevant information does not necessarily ensure its effective use Joren et al. (2025); Liu et al. (2024); Hsieh et al. (2024). AutoAdvExBench similarly reports that explicitly providing defense-specific attack knowledge does not necessarily improve attack success Carlini et al. (2025). Our work brings this knowledge-utilization perspective to adversarial-defense red teaming, focusing on how target-specific information can guide knowledge selection and executable adaptive attack construction.

3 Proposed Framework

3.1 Problem Setting and Evaluation Threat Model

We consider an autonomous red-teaming setting in which an LLM agent analyzes a given adversarial defense Zhou et al. (2022); Akhtar et al. (2021) and constructs an adaptive attack tailored to the

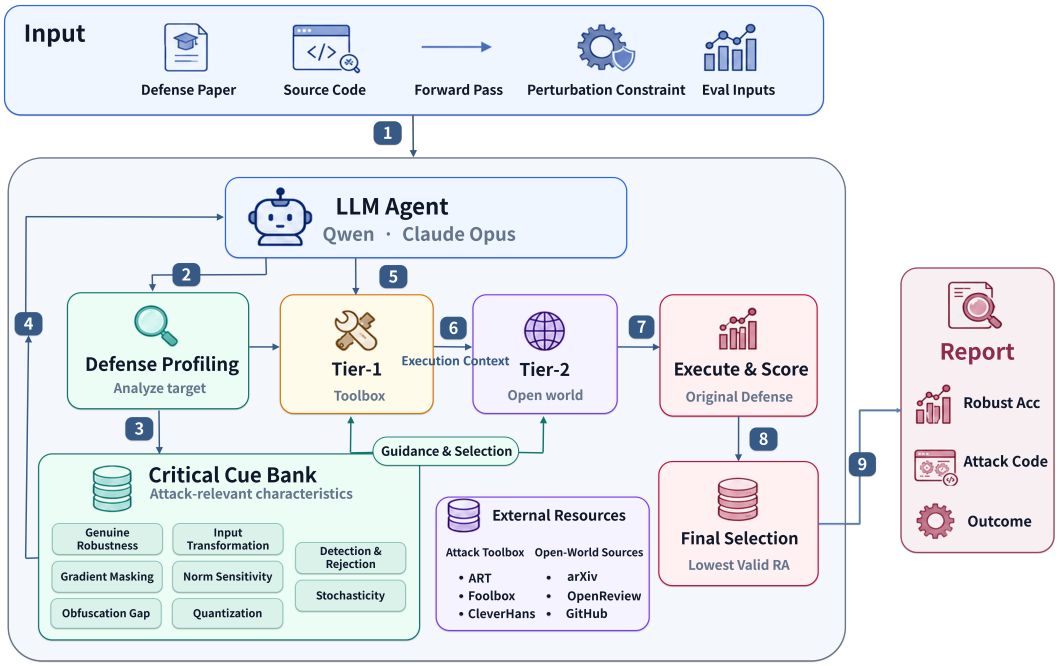


Figure 2: Overview of DART. The CCB provides target-specific guidance for Tier-1 toolbox attacks and Tier-2 open-world expansion, followed by evaluation on the original target defense and final attack selection.

target implementation. Following the evaluation setting of AutoAdvExBench Carlini et al. (2025), the agent is provided with a description or paper of the target defense, its source implementation, a forward-pass interface, the perturbation constraint, and evaluation inputs. The goal is to identify attack-relevant properties of the target and construct an executable adaptive attack that satisfies the prescribed threat model.

The agent can inspect the target source code and execution behavior, select attacks from an available attack toolbox, and adapt them to the target implementation. Our framework further extends this attack construction process beyond the predefined toolbox by retrieving attack knowledge from external research papers and public implementations and using it to construct additional target-specific candidates.

For each defense d, the benchmark-defined perturbation constraint Θ_d remains fixed for all attack candidates used in final evaluation. Defense Profiling may additionally use auxiliary diagnostic probes, such as larger-budget or alternative-norm evaluations, solely to characterize attack-failure behavior. These diagnostic probes are not included in the candidate pool and do not contribute to the reported robust accuracy.

Let $\mathcal{A}_d^{\text{valid}}$ denote the set of candidate attacks that produce a valid evaluation for defense d. The agent seeks the candidate that minimizes the robust accuracy of the target defense:

$$a_d^* = \arg \min_{a \in \mathcal{A}_d^{\text{valid}}} RA_d(a), \tag{1}$$

where $RA_d(a)$ denotes the robust accuracy of defense d under attack a. The defense-level attack-success criterion and evaluation outcomes are defined in Section 4.1.

3.2 Framework Overview and Defense Profiling

We propose **DART**, a defense-aware open-world red-teaming framework for automatically constructing adaptive attacks against heterogeneous adversarial defenses (Figure 2). DART consists of four stages: *Defense Profiling*, *Tier-1 Defense-Aware Toolbox Attack*, *Tier-2 Open-World Attack*

Expansion, and *Final Candidate Validation and Attack Selection*. Profiling-derived attack-relevant characteristics are structured in the Critical Cue Bank (CCB), which guides both Tier-1 toolbox attack construction and Tier-2 open-world expansion, while Tier-1 execution context further conditions Tier-2 retrieval and adaptation. Generated candidates are evaluated on the original target defense, and the valid candidate with the lowest robust accuracy is selected.

Defense Profiling. Through lightweight diagnostic probing, the agent examines attack behavior across perturbation budgets and norms, input transformation and quantization behavior, stochasticity, detection or rejection mechanisms, gradient reliability, and the input/output interface.

Profiling and attack construction use a differentiable forward pass derived from the original target defense. Behaviors that are not directly observable through this forward pass, such as detection or rejection, are examined through limited probing of the original target implementation. Final robustness scoring is always performed on the original target defense.

Let O_d^{probe} denote the execution observations collected during profiling. The framework transforms the attack-relevant subset of these observations into the Critical Cue Bank (CCB):

$$C_d = \Phi_{\text{profile}}\left(O_d^{\text{probe}}; \mathcal{M}\right), \tag{2}$$

where $\mathcal{M}$ denotes the underlying LLM agent. Defense papers and source implementations may be used to establish the execution environment, construct the attack-side differentiable forward pass, and interpret profiling behavior, but their raw contents are not stored in the CCB. The CCB is constructed only from attack-relevant observations identified through profiling. Further details of the profiling interface and probing procedure are provided in Appendix A.1.

Critical Cue Bank. The CCB maps profiling observations to eight operational cues, each linking an attack-relevant diagnostic signal to a downstream attack decision:

$$C_d = \left(q_d^{(k)}\right)_{k \in \mathcal{K}_{\text{CCB}}}. \tag{3}$$

Here, $\mathcal{K}_{\text{CCB}}$ is the index set of eight operational cue types: *Obfuscation Gap*, *Genuine Robustness*, *Norm Sensitivity*, *Input Transformation*, *Stochasticity*, *Detection & Rejection*, *Quantization*, and *Gradient Masking*. Multiple cues may apply to the same defense when profiling reveals distinct attack-relevant behaviors. Their extraction procedure and definitions are detailed in Appendix A.2.

Rather than serving as a generic summary of the defense, the CCB records attack-relevant evidence used to determine which attack strategies are relevant and how they should be adapted to the target. In Tier-1, these cues guide toolbox attack selection and adaptation. In Tier-2, they are combined with the Tier-1 execution context to guide external attack retrieval and research-to-executable attack adaptation.

The CCB is fixed after the initial profiling stage. Runtime feedback observed during candidate execution, such as execution errors, robust accuracy, and rejection behavior, is maintained separately in a dynamic execution context:

$$E_d^{(t+1)} = E_d^{(t)} \cup \left\{o_d^{(t)}\right\}, \tag{4}$$

where $o_d^{(t)}$ denotes the runtime observation obtained from the t-th attack execution. This separation preserves a stable target characterization in the CCB while allowing execution-time feedback to support attack repair and Tier-2 retrieval.

3.3 Tier-1: Defense-Aware Toolbox Attack

Tier-1 constructs a target-specific attack from an available attack toolbox. The framework is agnostic to the specific toolbox and can operate with established libraries such as ART, Foolbox, or CleverHans. In our implementation, we use the Adversarial Robustness Toolbox (ART), which provides a broad collection of adversarial evasion attacks.

Let $\mathcal{T}$ denote the set of attacks available in the toolbox. Given the Critical Cue Bank C_d, the agent selects an attack from $\mathcal{T}$ whose assumptions and optimization behavior are compatible with the target-specific cues identified during profiling. The selected attack is then adapted to the target

implementation according to C_d and the benchmark-defined perturbation constraint Θ_d. This adaptation may involve changes to the attack objective, optimization configuration, gradient handling, or input representation while preserving the prescribed threat model.

The resulting attack is executed against the target defense. If the initial Tier-1 attack produces a valid robust-accuracy result, it is retained as the Tier-1 candidate. Otherwise, the agent performs a single execution-driven repair using the runtime feedback from the failed attempt together with the target-specific information in C_d. The repair is limited to one additional attempt to avoid unconstrained iterative trial-and-error.

If the repaired attack produces a valid robust-accuracy result, it becomes the Tier-1 candidate; otherwise, Tier-1 yields no valid candidate. Tier-2 is executed regardless of the Tier-1 outcome.

3.4 TIER-2: OPEN-WORLD ATTACK EXPANSION

Tier-2 expands the attack search space beyond predefined toolboxes to external research papers and public implementations. Its goal is to identify attack knowledge that is either directly applicable to the target defense or transferable from related mechanisms, and to convert that knowledge into additional target-specific executable attacks.

Open-World Attack Retrieval. Tier-2 retrieval is conditioned jointly on the CCB and the Tier-1 execution context. The agent constructs search queries from the target-defense identifier, attack-relevant mechanisms encoded in C_d, and the observed Tier-1 outcome. It then searches public papers and implementations for attacks that directly target the defense or address related attack-relevant mechanisms. The retrieval priority and candidate budget are detailed in Appendix A.3.

Research-to-Executable Attack Adaptation. A retrieved paper or repository does not by itself constitute an executable attack against the current target. For each research candidate r_i, the agent extracts the attack knowledge required for implementation, including the attack objective, optimization principle, gradient handling, and key assumptions, and adapts this knowledge to the target-specific cues and constraints:

$$a_{d,i}^{(0)} = \Psi_{\text{exec}}\left(\mathcal{K}(r_i); C_d, E_d^{\text{T1}}, \Theta_d\right). \tag{5}$$

Here, $\mathcal{K}(r_i)$ denotes the attack knowledge extracted from research candidate r_i, E_d^{T1} denotes the execution context obtained from Tier-1, and Ψ_{exec} denotes the procedure that preserves the core attack principle while adapting target-dependent execution details. The perturbation constraint Θ_d remains fixed throughout this process. The detailed attack-knowledge schema is provided in Appendix A.4.

Candidate Execution and One-Step Repair. Each research-derived candidate is executed within the prescribed budget, without early termination based on intermediate attack success. The agent then performs one execution-driven repair using the feedback from the initial execution together with the target-specific information in C_d. Repair is performed once for each research candidate regardless of the initial outcome, limiting unconstrained iterative trial-and-error.

Both the initial and repaired attempts are retained for Tier-2 candidate evaluation. Candidates that produce valid robust-accuracy results are subsequently considered together with the Tier-1 candidate during final validation and selection in Section 3.5. The detailed construction of the Tier-2 candidate set is provided in Appendix A.4.

3.5 FINAL CANDIDATE VALIDATION AND ATTACK SELECTION

Candidates generated by Tier-1 and Tier-2 are merged into a common candidate pool, from which only attacks producing a valid robust-accuracy result are retained for final comparison. If at least one valid candidate exists, we select the attack with the lowest robust accuracy according to the objective defined in Section 3.1. The selected robust accuracy is then compared against the AutoAdvExBench criterion, $RA < 0.5\,CA$, to determine whether the outcome is *Attack Success* or *Not Broken*.

If no candidate produces a valid evaluation, the defense is classified as *Failure*. This includes cases in which clean accuracy cannot be computed, attack generation or validation fails, or execution

terminates before obtaining a valid robust-accuracy result. We do not impute or substitute robust accuracy in such cases. For valid evaluations, we record the selected attack, its robust accuracy, the evaluation outcome, and the corresponding executable attack implementation; otherwise, only the *Failure* status is reported.

4 EXPERIMENTS

We evaluate the proposed defense-aware open-world red-teaming framework in terms of both attack effectiveness and computational efficiency. Our experiments address three questions. First, does the proposed framework improve attack success and evaluation coverage over the agent introduced in AutoAdvExBench? Second, is access to open-world attack knowledge alone sufficient to improve attack performance? Third, do the observed gains require additional LLM-side computation, or can they be achieved with lower resource and runtime overhead? We additionally evaluate the effect of the Tier-2 one-step execution-feedback repair applied after the initial execution of research-derived candidates; detailed results are provided in Appendix B.

4.1 EXPERIMENTAL SETUP

We evaluate all methods on the 51 published defense implementations included in AutoAdvExBench, spanning CIFAR-10 Krizhevsky et al. (2009), MNIST LeCun et al. (1998), and ImageNet Deng et al. (2009).

The original AutoAdvExBench evaluation uses 1,000 images per defense. In our experiments, we use a fixed subset of **200 evaluation images per defense** and apply the same subset to all agent configurations to ensure consistent evaluation conditions across methods.

Clean accuracy is measured on the same 200 clean evaluation images using the original target defense. For defenses with detection or rejection mechanisms, a clean input is counted as correct only when the ground-truth class is explicitly predicted. Similarly, an adversarial input is counted as a successful attack only when it bypasses the rejection mechanism and induces an actual misclassification.

We evaluate two attack-authoring backbones, **Qwen3-Next-80B-A3B-Thinking** and **Claude Opus 4.8**, integrated into the same pipeline under identical downstream execution conditions. Detailed model and hardware configurations are provided in Appendix A.5.

We compare three agent configurations. **Baseline** denotes the attack agent introduced in AutoAdvExBench. **Baseline + Open-World** retains the baseline attack-construction procedure while providing access to external attack knowledge from research papers and public implementations Yao et al. (2022); Nakano et al. (2021). **DART** denotes the complete defense-aware open-world red-teaming framework proposed in this work.

Evaluation outcomes are categorized as **Attack Success**, **Not Broken**, or **Failure**. During final scoring, adversarial examples that violate the benchmark-defined perturbation constraint Θ_d are replaced by their corresponding clean inputs. Following the candidate validation and selection procedure in Section 3.5, the minimum robust accuracy among valid candidates is used as the final robust accuracy for each defense. We then apply the AutoAdvExBench defense-level criterion,

$$RA < 0.5\,CA, \tag{6}$$

to distinguish Attack Success from Not Broken. A defense is classified as *Failure* when no valid evaluation can be produced.

To assess computational efficiency, we measure total token usage, the number of LLM calls, target-defense queries, end-to-end runtime, and API cost.

4.2 MAIN RESULTS

As shown in Table 1, the proposed framework achieves higher attack success rates than the baseline with both LLM backbones. With Qwen, the baseline succeeds on 3 of the 51 defenses, corresponding to an attack success rate of 5.9%, whereas our framework succeeds on 22 defenses, reaching 43.1%. This represents an improvement of 37.2 percentage points. With Claude Opus, the attack success

Table 1: Attack performance on the 51 adversarial defenses in AutoAdvExBench. **Attack Success** denotes defenses for which the final attack satisfies the benchmark success criterion, **Not Broken** denotes defenses for which a valid evaluation is completed without satisfying the criterion, and **Failure** denotes defenses for which the agent fails to obtain a valid attack evaluation. Δ Success Rate reports the percentage-point change relative to the corresponding backbone baseline.

Backbone	Method	Attack Success ↑	Not Broken	Failure ↓	Success Rate ↑	Δ Success Rate (pp)
Qwen	Baseline	3	13	35	5.9%	–
	Baseline + Open-World	5	11	35	9.8%	+3.9
	DART	**22**	**29**	**0**	**43.1%**	**+37.2**
Claude Opus	Baseline	16	6	29	31.4%	–
	Baseline + Open-World	16	16	19	31.4%	0.0
	DART	**27**	**24**	**0**	**52.9%**	**+21.5**

rate increases from 31.4% with the baseline to 52.9% with our framework, an improvement of 21.5 percentage points.

The framework also improves evaluation coverage. The Qwen and Claude Opus baselines fail to produce valid evaluations for 35 and 29 defenses, respectively. In contrast, our framework completes valid evaluations for all 51 defenses with both backbones. Thus, the observed improvement is not limited to constructing stronger attacks for defenses already handled by the baseline; the framework also extends evaluation coverage to defenses for which the baseline fails to construct a valid executable attack.

4.3 Is Open-World Access Alone Sufficient?

To determine whether the gains of our framework arise simply from access to external attack knowledge, we evaluate **Baseline + Open-World**. This configuration preserves the attack-construction procedure of the AutoAdvExBench baseline while allowing the agent to retrieve additional attack knowledge from external research papers and public implementations. Comparing Baseline with Baseline + Open-World therefore isolates the effect of adding open-world knowledge access without changing the underlying agent architecture.

With Qwen, adding open-world access increases the attack success rate from 5.9% to 9.8%, corresponding to two additional successful defenses, while the number of Failures remains unchanged at 35. With Claude Opus, the number of Failures decreases from 29 to 19, indicating improved evaluation coverage, but the number of Attack Success cases remains 16 and the attack success rate remains 31.4%. Open-world access alone therefore provides limited improvements in either attack success or evaluation coverage, but does not yield consistent gains in attack success across the two backbones.

Baseline + Open-World also incurs additional LLM-side computational overhead, which we analyze in Section 4.5. In contrast, the complete framework yields larger improvements with both backbones. In our evaluation, adding open-world access alone does not yield consistent gains in attack success. The larger improvements are observed only with the full target-conditioned attack-construction pipeline, which combines profiling, retrieval, and executable adaptation.

4.4 From Knowledge Access to Executable Attack Construction

AutoAdvExBench reports that directly providing defense-specific attack papers in the LLM context does not necessarily yield a meaningful improvement in attack success Carlini et al. (2025). We observe a similar pattern in Section 4.3: adding open-world access to the baseline agent yields only limited improvement with Qwen and no increase in attack success with Claude Opus. These results highlight a distinction between *accessing relevant attack knowledge* and *operationalizing that knowledge into a target-specific executable attack*.

Our framework addresses this gap by structuring profiling-derived attack-relevant characteristics in the Critical Cue Bank (CCB) and using them consistently to guide both research retrieval and target-

Table 2: LLM resource consumption across agent configurations. **LLM Calls** counts all calls to the backbone model, while **Calls/Defense** reports the average number of LLM calls per target defense over all 51 defenses. All relative changes are computed against the corresponding backbone baseline.

Backbone	Method	Tokens ↓	Δ Tokens	LLM Calls ↓	Calls / Defense ↓
Qwen	Baseline	44,479.7K	–	3,137	62
	Baseline + Open-World	47,374.0K	+6.5%	3,234	63
	DART	**7,222.1K**	**-83.8%**	**748**	**15**
Opus	Baseline	167,868.3K	–	4,955	97
	Baseline + Open-World	207,733.4K	+23.7%	6,291	123
	DART	**8,348.8K**	**-95.0%**	**570**	**11**

Table 3: End-to-end runtime, target queries, and API cost. Relative changes are computed with respect to the corresponding backbone baseline. Qwen is executed locally and therefore has no API cost.

Backbone	Method	Total Time ↓	Δ Time	Target Queries	Δ Queries	Cost ↓	Δ Cost
Qwen	Baseline	44.26h	–	81,200	–	–	–
	Baseline + Open-World	47.13h	+6.5%	84,400	+3.9%	–	–
	DART	**35.18h**	**-20.5%**	**147,000**	**+81.0%**	–	–
Claude Opus	Baseline	76.90h	–	38,600	–	$158.27	–
	Baseline + Open-World	103.38h	+34.4%	49,200	+27.5%	$276.06	+74.4%
	DART	**36.62h**	**-52.4%**	**96,400**	**+149.7%**	**$99.46**	**-37.2%**

specific attack adaptation. Rather than treating external knowledge as additional context alone, the agent first characterizes the mechanisms that hinder attack effectiveness on the current target and then uses these cues to guide knowledge selection and executable adaptation.

Trapdoor provides a representative example of this process. The initial APGD attack leaves the robust accuracy at **0.565**, with some adversarial inputs rejected by the detector. Profiling maps the observed rejection behavior and activation-based detection mechanism to the *Detection & Rejection* cue. Conditioned on this cue, the agent searches for research relevant to detector bypass and adapts the retrieved attack principle to the target defense. This reduces the robust accuracy from **0.565** to **0.010** and the rejection rate to **0**.

Similar patterns are observed across other defenses. Profiling-derived cues guide research retrieval and target-specific adaptation; however, for DiffPure, the agent retrieves a relevant EOT principle but fails to operationalize it faithfully for the target process, resulting in no improvement in attack performance. This suggests that effective autonomous attack construction depends not merely on the availability of external knowledge, but on its target-specific operationalization. Detailed success and failure traces are provided in Appendix C.

4.5 LLM-Side Resource and Runtime Efficiency

Tables 2 and 3 summarize the LLM-side resource usage and end-to-end execution cost, respectively. Across both backbones, our framework reduces token usage, the number of LLM calls, and runtime, although the total number of target-defense queries increases.

With Qwen, our framework reduces token usage by **84%**, LLM calls by **76%**, and runtime by **21%** relative to the baseline. The total number of target-defense queries increases from 81,200 to 147,000, while valid evaluation coverage increases from 16 to all 51 defenses. When normalized by the number of valid evaluations, the query count decreases from 5,075 to 2,882 queries per valid evaluation.

With Claude Opus, token usage decreases by **95%**, LLM calls by **89%**, runtime by **52%**, and API cost by **37%**. The total number of target-defense queries increases from 38,600 to 96,400, while valid evaluation coverage increases from 22 to all 51 defenses. The corresponding query counts

are 1,755 and 1,890 queries per valid evaluation for the baseline and our framework, respectively, indicating comparable query efficiency after accounting for evaluation coverage.

These results show that the efficiency gains of our framework do not arise from reducing target-defense queries. Instead, the framework achieves complete evaluation coverage while reducing LLM-side computation and runtime. When normalized by valid evaluation coverage, query efficiency improves with Qwen and remains comparable to the baseline with Claude Opus.

5 Limitations and Future Work

Our study has three main limitations. First, the current evaluation is limited to image-classification defenses on CIFAR-10, MNIST, and ImageNet. Whether the proposed profiling and open-world attack adaptation generalize to other tasks, such as object detection, segmentation, vision-language models, and generative models, remains to be evaluated.

Second, the current instantiation assumes access to the target implementation and an executable interface. In API-only or strict black-box settings, attack-relevant characteristics would instead need to be inferred primarily from queries and observed runtime behavior.

Third, Open-World Attack Expansion depends on the availability and completeness of external attack knowledge. When relevant papers or implementations are missing or incomplete, constructing effective executable candidates may be more difficult. Future work will therefore extend the framework to broader tasks and access settings, and investigate retrieval-quality estimation and multi-source knowledge integration for incomplete attack specifications.

6 Conclusion

We introduced a defense-aware open-world red-teaming framework for autonomously constructing adaptive attacks against heterogeneous adversarial defenses. The framework organizes attack-relevant characteristics of the target defense into a Critical Cue Bank and reuses this information throughout toolbox attack adaptation, open-world retrieval, and research-to-executable attack construction. This enables the agent to move beyond a predefined attack space by acquiring external attack knowledge and translating it into executable attacks specialized to the target defense.

Across 51 published defense implementations from AutoAdvExBench, DART improves the attack success rate from 5.9% to 43.1% with Qwen and from 31.4% to 52.9% with Claude Opus, while completing valid evaluations for all defenses with both backbones. In our evaluation, providing open-world access alone does not yield comparable gains. Together with the case analyses, this result supports a distinction between access to attack knowledge and the target-specific processes required to convert that knowledge into executable adaptive attacks. At the same time, the proposed framework achieves higher attack success and broader evaluation coverage with substantially lower LLM-side computation and runtime.

Ethics Statement

This work investigates automated adaptive attacks to improve the reliability of adversarial-defense evaluation. All experiments were conducted in isolated environments using publicly available research implementations included in AutoAdvExBench. We did not target deployed services, unauthorized systems, personal data, or human subjects. Because DART constructs executable attack implementations, it may lower the expertise required to evaluate, and potentially attack, machine-learning defenses, creating a dual-use risk. We therefore position DART solely as a tool for authorized robustness assessment and defensive research, and evaluate all attacks under benchmark-defined threat models. The released artifacts contain no credentials or access to non-public systems. Users should apply the framework only to systems for which they have explicit authorization.

AI Use Statement

Generative AI systems were used in two roles. First, Qwen3-Next-80B-A3B-Thinking and Claude Opus 4.8 served as the experimental attack-authoring backbones evaluated in DART and the base-

line configurations. Second, generative AI tools were used to edit and restructure portions of the manuscript for clarity and readability. The authors designed the research questions, methodology, evaluation protocol, and experiments, and manually reviewed all generated attack implementations, experimental outputs, citations, and AI-assisted text. The authors take full responsibility for the final content of this work.

Reproducibility Statement

Section 4.1 describes the benchmark, evaluation criterion, and attack-authoring backbones. Appendix A provides implementation details for profiling, retrieval, attack construction, and execution. An anonymous implementation of DART is available at `https://anonymous.4open.science/r/DART-A008/`.

References

Talor Abramovich, Meet Udeshi, Minghao Shao, Kilian Lieret, Haoran Xi, Kimberly Milner, Sofija Jancheska, John Yang, Carlos E Jimenez, Farshad Khorrami, et al. Enigma: Interactive tools substantially assist lm agents in finding security vulnerabilities. *arXiv preprint arXiv:2409.16165*, 2024.

Eashan Adhikarla, Dan Luo, and Brian D Davison. Memory defense: More robust classification via a memory-masking autoencoder. *arXiv preprint arXiv:2202.02595*, 2022.

Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE access*, 9:155161–155196, 2021.

Motasem Alfarra, Juan C Pérez, Ali Thabet, Adel Bibi, Philip HS Torr, and Bernard Ghanem. Combating adversaries with anti-adversaries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 5992–6000, 2022.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International conference on learning representations*, 2018.

Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd annual computer security applications conference*, pp. 278–287, 2017.

Nicholas Carlini, Javier Rando, Edoardo Debenedetti, Milad Nasr, and Florian Tramèr. Autoadvexbench: Benchmarking autonomous exploitation of adversarial example defenses. *arXiv preprint arXiv:2503.01811*, 2025.

Xinglong Chang, Katharina Dost, Kaiqi Zhao, Ambra Demontis, Fabio Roli, Gillian Dobbie, and Jörg Wicker. Baard: Blocking adversarial examples by testing for applicability, reliability and decidability. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 3–14. Springer, 2023.

Hao-Yun Chen, Jhao-Hong Liang, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, and Da-Cheng Juan. Improving adversarial robustness via guided complement entropy. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4880–4888. IEEE, 2019.

Jiefeng Chen, Jayaram Raghuram, Jihye Choi, Xi Wu, Yingyu Liang, and Somesh Jha. Stratified adversarial robustness with rejection. In *International conference on machine learning*, pp. 4867–4894. PMLR, 2023.

Yuxuan Chen, Daniel Röder, Justus-Jonas Erker, Leonhard Hennig, Philippe Thomas, Sebastian Möller, and Roland Roller. Retrieval-augmented knowledge integration into language models: a survey. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pp. 45–63, 2024.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Jiequan Cui, Zhuotao Tian, Zhisheng Zhong, Xiaojuan Qi, Bei Yu, and Hanwang Zhang. Decoupled kullback-leibler divergence loss. *Advances in Neural Information Processing Systems*, 37:74461–74486, 2024.

Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 196–204, 2018.

Edoardo Debenedetti, Vikash Sehwag, and Prateek Mittal. A light recipe to train robust vision transformers. In *2023 IEEE conference on secure and trustworthy machine learning (SaTML)*, pp. 225–253. IEEE, 2023.

Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. {PentestGPT}: Evaluating and harnessing large language models for automated penetration testing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 847–864, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

Alec F Diallo and Paul Patras. Sabre: Cutting through adversarial noise with adaptive spectral filtering and input reconstruction. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 2901–2919. IEEE, 2024.

Stanislav Fort and Balaji Lakshminarayanan. Ensemble everything everywhere: Multi-scale aggregation for adversarial robustness. *arXiv preprint arXiv:2408.05446*, 2024.

Iuri Frosio and Jan Kautz. The best defense is a good offense: Adversarial augmentation against adversarial attacks. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4067–4076. IEEE, 2023.

Luca Gioacchini, Alexander Delsanto, Idilio Drago, Marco Mellia, Giuseppe Siracusano, and Roberto Bifulco. Autopenbench: A vulnerability testing benchmark for generative agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1615–1624, 2025.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

Pengfei He, Ash Fox, Lesly Miculicich, Stefan Friedli, Daniel Fabian, Burak Gokturk, Jiliang Tang, Chen-Yu Lee, Tomas Pfister, and Long T Le. Co-redteam: Orchestrated security discovery and exploitation with llm agents. *arXiv preprint arXiv:2602.02164*, 2026.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pp. 2712–2721. PMLR, 2019.

Chih-Hui Ho and Nuno Vasconcelos. Disco: Adversarial defense with local implicit functions. *Advances in neural information processing systems*, 35:23818–23837, 2022.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, et al. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14982–14995, 2024.

Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Weinberger. A new defense against adversarial images: Turning a weakness into a strength. *Advances in neural information processing systems*, 32, 2019.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *International Conference on Learning Representations*, volume 2025, pp. 20310–20334, 2025.

Qiyu Kang, Yang Song, Qinxu Ding, and Wee Peng Tay. Stable neural ode with lyapunov-stable equilibrium points for defending against adversarial attacks. *Advances in Neural Information Processing Systems*, 34:14925–14937, 2021.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Lin Li and Michael Spratling. Improved adversarial training through adaptive instance-wise loss smoothing. *arXiv preprint arXiv:2303.14077*, 2023.

Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787. IEEE, 2018.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 860–868. IEEE, 2019.

Peter Lorenz, Dominik Strassel, Margret Keuper, and Janis Keuper. Is robustbench/autoattack a suitable benchmark for adversarial robustness? *arXiv preprint arXiv:2112.01601*, 2021.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

Chengzhi Mao, Mia Chiquier, Hao Wang, Junfeng Yang, and Carl Vondrick. Adversarial attacks are reversible with natural supervision. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 641–651. IEEE, 2021.

Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 135–147, 2017.

Hancheng Min and René Vidal. Can implicit bias imply adversarial robustness? *arXiv preprint arXiv:2405.15942*, 2024.

Lajos Muzsai, David Imolai, and András Lukács. Hacksynth: Llm agent and evaluation framework for autonomous penetration testing. *arXiv preprint arXiv:2412.01778*, 2024.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. Adversarial robustness toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069*, 2018.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *arXiv preprint arXiv:1905.10626*, 2019a.

Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. *arXiv preprint arXiv:1909.11515*, 2019b.

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, et al. Technical report on the cleverhans v2. 1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2016.

Maura Pintor, Luca Demetrio, Angelo Sotgiu, Ambra Demontis, Nicholas Carlini, Battista Biggio, and Fabio Roli. Indicators of attack failure: Debugging and improving optimization of adversarial examples. *Advances in Neural Information Processing Systems*, 35:23063–23076, 2022.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 8571–8580. IEEE, 2018.

Zhuang Qian, Shufei Zhang, Kaizhu Huang, Qiufeng Wang, Rui Zhang, and Xinping Yi. Improving model robustness with latent distribution locally and globally. *arXiv preprint arXiv:2107.04401*, 2021.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

Sanchari Sen, Balaraman Ravindran, and Anand Raghunathan. Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks. *arXiv preprint arXiv:2004.10162*, 2020.

Shawn Shan, Emily Wenger, Bolun Wang, Bo Li, Haitao Zheng, and Ben Y Zhao. Gotta catch'em all: Using honeypots to catch adversarial attacks on neural networks. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pp. 67–83, 2020.

Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervision. *arXiv preprint arXiv:2101.09387*, 2021.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 2020.

Gunjan Verma and Ananthram Swami. Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*, 2019.

Yulong Wang, Tianxiang Li, Shenghong Li, Xin Yuan, and Wei Ni. New adversarial image detection based on sentiment analysis. *IEEE transactions on neural networks and learning systems*, 35: 14060–14074, 2023.

Boxi Wu, Heng Pan, Li Shen, Jindong Gu, Shuai Zhao, Zhifeng Li, Deng Cai, Xiaofei He, and Wei Liu. Attacking adversarial attacks as a defense. *arXiv preprint arXiv:2106.04938*, 2021.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in neural information processing systems*, 33:2958–2969, 2020.

Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all. *arXiv preprint arXiv:1905.10510*, 2019.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Chengyuan Yao, Pavol Bielik, Petar Tsankov, and Martin Vechev. Automated discovery of adaptive attacks on adversarial defenses. *Advances in Neural Information Processing Systems*, 34:26858–26870, 2021.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *International conference on machine learning*, pp. 12062–12072. PMLR, 2021.

Qiusi Zhan, Richard Fang, Henil Shalin Panchal, and Daniel Kang. Adaptive attacks break defenses against indirect prompt injection attacks on llm agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7116–7132, 2025.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Andy Zhou, Kevin Wu, Francesco Pinto, Zhaorun Chen, Yi Zeng, Yu Yang, Shuang Yang, Sanmi Koyejo, James Zou, and Bo Li. Autoredteamer: Autonomous red teaming with lifelong attack integration. *Advances in Neural Information Processing Systems*, 38:169852–169895, 2026.

Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S Yu. Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Computing Surveys*, 55 (8):1–39, 2022.

Yuwen Zou, Jia Liu, and Wenjun Fan. Ctfagent: An llm-powered agent for ctf challenge solving. *Journal of Information Security and Applications*, 96:104305, 2026.

A Method Implementation Details

A.1 Profiling Interface

Let F_d denote the attack-side differentiable forward pass used for profiling and attack construction, and let S_d denote the source implementation of the original target defense. The agent constructs F_d from the original forward pass provided by the benchmark and verifies that its predictions on clean inputs are consistent with those of the original target defense before using it for attack construction.

Given a small set of probe inputs X_d^{probe}, the agent collects execution observations as

$$O_d^{\text{probe}} = \text{Probe}\left(F_d, S_d, X_d^{\text{probe}}\right). \tag{7}$$

Most observations are obtained through lightweight probing of F_d. Behaviors that are not directly exposed by the differentiable forward pass, such as detection or rejection, are examined using S_d or the original target defense when necessary. The probing procedure characterizes execution properties that directly affect attack construction, including attack behavior across perturbation budgets and norms, input transformations and quantization, stochasticity, detection or rejection behavior, and gradient reliability.

Table 4: Operational cue types in the CCB. Each cue represents a profiling-derived property that can affect attack selection, retrieval, or target-specific adaptation.

Cue	Profiling evidence	Attack implication
Obfuscation Gap	Unbounded-budget diagnostic attacks that fail to reach near-zero robust accuracy, random perturbations outperforming gradient-based attacks, or non-monotone budget–robustness behavior.	Flags a potential gap between observed robustness and attack effectiveness; motivates transfer, query-based, random-search, or surrogate-gradient attacks.
Genuine Robustness	Robust accuracy remains high under strong bounded and auxiliary unbounded gradient-based attacks and random perturbations, without clear gradient-masking signals.	Provides evidence consistent with genuine robustness rather than an obvious attack failure; motivates stronger multi-restart or ensemble attacks before introducing a mechanism-specific bypass.
Norm Sensitivity	Asymmetric attack behavior across auxiliary L_∞ and L_2 evaluations under comparable diagnostic settings.	Identifies norm-dependent attack behavior and guides attack-family selection or optimization settings while preserving the benchmark-defined perturbation constraint for final attack candidates.
Input Transformation	Attack-relevant transformations applied before classification, such as compression, reconstruction, denoising, or purification.	Motivates transformation-aware optimization, differentiable approximations, or mechanism-specific attacks.
Stochasticity	Variation across repeated forward or gradient evaluations caused by randomized target behavior.	Motivates expectation-based optimization such as EOT or repeated sampling.
Detection & Rejection	An auxiliary mechanism that flags adversarial or abnormal inputs, or rejects/abstains instead of returning a standard class prediction.	Requires attacks to evade detection and avoid rejection while inducing misclassification.
Quantization	Discontinuous or piecewise-constant input responses revealed by fine-grained perturbation probes, such as bit-depth reduction or thermometer encoding.	Motivates perturbations that cross quantization boundaries rather than relying solely on smooth local gradients.
Gradient Masking	Cases in which gradients are available but do not reliably reflect useful local attack directions because of nondifferentiability, discontinuity, saturation, or related mechanisms.	Motivates BPDA, gradient smoothing, EOT, or alternative gradient construction.

A.2 Critical Cue Bank Construction and Cue Types

CCB Extraction. The CCB is constructed from attack-relevant observations obtained during Defense Profiling rather than from a direct summary of the defense paper or source code. For each target defense, the agent examines the attack-side differentiable forward pass and, when necessary, the original target implementation. The resulting observations are filtered according to whether they affect attack construction and are organized into eight operational cue types: *Obfuscation Gap*, *Genuine Robustness*, *Norm Sensitivity*, *Input Transformation*, *Stochasticity*, *Detection & Rejection*, *Quantization*, and *Gradient Masking*.

The cue extraction process combines structural and behavioral evidence. Structural evidence includes input transformations, detector or rejection branches, quantization operations, and nondifferentiable components identified from the target implementation. Behavioral evidence is obtained through lightweight diagnostic probes, including gradient inspection, repeated forward executions, short optimization runs, and, when needed, auxiliary budget- or norm-based evaluations. These auxiliary probes are used only to characterize attack behavior and do not contribute to the reported robust accuracy. The resulting cues encode attack-relevant properties rather than a general description of the defense and remain fixed after profiling. Runtime observations obtained from subsequent candidate executions are instead maintained separately as execution context.

Operational Cue Types. The CCB consists of eight operational cue types: *Obfuscation Gap*, *Genuine Robustness*, *Norm Sensitivity*, *Input Transformation*, *Stochasticity*, *Detection & Rejection*, *Quantization*, and *Gradient Masking*. Table 4 summarizes the profiling evidence represented by each cue and its implication for subsequent attack construction.

The cue types are not mutually exclusive and may co-occur for the same defense. *Obfuscation Gap* captures outcome-level evidence that standard attacks may be underestimating attack effectiveness, whereas *Gradient Masking* captures mechanism-level evidence that available gradients are locally unreliable.

A.3 Open-World Retrieval Details

Tier-2 prioritizes research candidates in three stages: (1) direct known breaks of the target defense, (2) related work linked through the defense configuration or its references, and (3) research addressing similar attack-relevant mechanisms. Candidates are assessed based on their relevance to the target and the availability of information required to construct an executable attack. For each defense, the agent retains at most $K \leq 5$ research candidates:

$$\mathcal{R}_d = \{r_1, \ldots, r_K\}. \tag{8}$$

A.4 Research-to-Executable Attack and Candidate Construction

For each research candidate r_i, the agent structures the attack knowledge required for executable attack construction as

$$\mathcal{K}(r_i) = \{O_i, L_i, U_i, G_i, Z_i, H_i\}. \tag{9}$$

Here, O_i denotes the attack objective, L_i the loss formulation, U_i the optimization or update rule, G_i the gradient-handling strategy, Z_i the initialization procedure, and H_i the key assumptions required for the attack to operate as intended.

For each research candidate, both the initial attack and its one-step repaired variant are retained for Tier-2 candidate evaluation. The resulting Tier-2 candidate set is

$$\mathcal{A}_d^{\mathrm{T2}} = \bigcup_{i=1}^{K} \left\{ a_{d,i}^{(0)}, \widehat{a}_{d,i} \right\}. \tag{10}$$

A.5 Model and Hardware Configuration

We use Qwen3-Next-80B-A3B-Thinking as the local attack-authoring backbone with Q8_0 quantization, a temperature of 0, and a maximum generation length of 4,096 tokens. Claude Opus 4.8 is accessed through the API with adaptive thinking, xhigh effort, streaming enabled, and a maximum generation length of 128,000 tokens. The Claude API requests use the provider's default sampling configuration. Both backbones are integrated into the same attack-construction and evaluation pipeline under identical downstream execution conditions. The execution harness sets the Python, NumPy, and PyTorch or TensorFlow random seeds to 0 for attack execution.

B Effect of Tier-2 Repair

B.1 Overall Effect

Tier-2 repair uses execution feedback from the initial research-derived candidate to perform a single target-specific correction. To limit iterative trial-and-error, repair is applied at most once to each research-derived candidate and is performed regardless of whether the initial candidate already satisfies the Attack Success criterion.

To isolate the contribution of this repair step, we compare the full Claude Opus-based framework with an otherwise identical variant in which Tier-2 repair is disabled. The full framework evaluates both the initial and one-step repaired attacks and selects the minimum robust accuracy among candidates that produce valid evaluations. The repair-disabled variant evaluates the same initial research-derived candidates but does not introduce repair-generated candidates.

Across all 51 defenses, the full framework achieves an average robust accuracy of **0.3560**, compared with **0.3707** without Tier-2 repair, corresponding to an additional reduction of **1.47 percentage points**. Repair changes the final selected result for **10 of the 51 defenses**, while the remaining 41 defenses are unaffected. Among these 10 affected defenses, the average robust accuracy decreases from 0.5720 to 0.4970, corresponding to an average reduction of **7.50 percentage points** and a maximum reduction of **32.5 percentage points**.

These results indicate that one-step repair is not the dominant source of the framework's overall performance gain. Instead, it serves as a complementary mechanism that improves candidate quality for a subset of defenses requiring additional target-specific correction after the initial attack construction. Table 5 provides the corresponding per-defense results.

B.2 PER-DEFENSE RESULTS

Table 5: Per-defense contribution of Tier-2 repair for our framework with Claude Opus. RA Reduction is computed as RA without repair minus RA with repair. Positive values indicate that enabling repair produces a lower final robust accuracy.

Defense	RA w/ Repair	RA w/o Repair	RA Reduction (pp)
Debenedetti et al. (2023)	0.575	0.900	+32.5
Ho & Vasconcelos (2022)	0.615	0.785	+17.0
Fort & Lakshminarayanan (2024)	0.600	0.725	+12.5
Lorenz et al. (2021)	0.290	0.380	+9.0
Wu et al. (2020)	0.575	0.585	+1.0
Alfarra et al. (2022)	0.805	0.815	+1.0
Kang et al. (2021)	0.870	0.875	+0.5
Mao et al. (2021)	0.630	0.635	+0.5
Xu et al. (2017)	0.000	0.005	+0.5
Sen et al. (2020)	0.010	0.015	+0.5
Ma et al. (2018)	0.840	0.840	0.0
Xiao et al. (2019)	0.075	0.075	0.0
Pang et al. (2019a)	0.000	0.000	0.0
Buckman et al. (2018)	0.165	0.165	0.0
Nie et al. (2022)	0.895	0.895	0.0
Wu et al. (2021)	0.615	0.615	0.0
Prakash et al. (2018)	0.000	0.000	0.0
Verma & Swami (2019)	0.025	0.025	0.0
Alfarra et al. (2022)	0.845	0.845	0.0
Li & Spratling (2023)	0.635	0.635	0.0
Cui et al. (2024)	0.710	0.710	0.0
Diallo & Patras (2024)	0.820	0.820	0.0
Frosio & Kautz (2023)	0.865	0.865	0.0
Diallo & Patras (2024)	0.090	0.090	0.0
Diallo & Patras (2024)	0.040	0.040	0.0
Shan et al. (2020)	0.010	0.010	0.0
Zhang et al. (2019)	0.550	0.550	0.0
Liao et al. (2018)	0.015	0.015	0.0
Meng & Chen (2017)	0.000	0.000	0.0
Guo et al. (2017)	0.000	0.000	0.0
Pang et al. (2019b)	0.020	0.020	0.0
Guo et al. (2017)	0.005	0.005	0.0
Chen et al. (2023)	0.575	0.575	0.0
Das et al. (2018)	0.010	0.010	0.0
Kannan et al. (2018)	0.000	0.000	0.0
Liu et al. (2019)	0.005	0.005	0.0
Adhikarla et al. (2022)	0.895	0.895	0.0
Qian et al. (2021)	0.595	0.595	0.0
Yoon et al. (2021)	0.875	0.875	0.0
Hendrycks et al. (2019)	0.590	0.590	0.0
Chang et al. (2023)	0.000	0.000	0.0
Dhillon et al. (2018)	0.440	0.440	0.0
Adhikarla et al. (2022)	0.840	0.840	0.0
Shi et al. (2021)	0.055	0.055	0.0
Wang et al. (2019)	0.595	0.595	0.0
Wang et al. (2023)	0.000	0.000	0.0
Min & Vidal (2024)	0.000	0.000	0.0
Hu et al. (2019)	0.415	0.415	0.0
Xie et al. (2017)	0.005	0.005	0.0
Cao & Gong (2017)	0.070	0.070	0.0
Chen et al. (2019)	0.000	0.000	0.0
All 51 defenses	0.3560	0.3707	+1.47
10 affected defenses	0.4970	0.5720	+7.50

C Detailed Analysis of Structured Attack Construction

C.1 Analysis Protocol

Section 4.4 uses Trapdoor as a representative example to illustrate how the CCB connects target understanding with external attack knowledge. In this appendix, we analyze additional defenses exhibiting different attack-construction patterns, focusing on *which critical cues are identified, how those cues guide research or implementation retrieval, and how the retrieved knowledge is operationalized into an executable target-specific attack*.

We organize each case according to a common sequence: target observation, critical cue, attack hypothesis, research/code retrieval, target-specific adaptation, and execution outcome.

The CCB does not store retrieved papers as a knowledge base. Instead, it structures profiling-derived attack-relevant characteristics into attack-oriented cues that condition both *what knowledge to retrieve* and *how the retrieved principle should be adapted to the target*.

This analysis is not a controlled ablation in which the CCB is removed, and therefore does not quantify its independent causal contribution. Rather, it provides mechanistic evidence showing how CCB-derived cues are connected to retrieval, attack-principle selection, target-specific adaptation, and execution validation in actual attack traces.

C.2 Trapdoor: Multi-Source Detector-Aware Attack Composition

Target and Initial Observation. Trapdoor is a detection-based defense that identifies adversarial examples by measuring the similarity between intermediate neural activations and predefined trapdoor signatures. Under the initial APGD attack, the robust accuracy remains at **0.565**, and a subset of adversarial examples is rejected by the detector. Thus, crossing the classifier decision boundary alone is insufficient for attack success.

Critical Cue and Attack Hypothesis. Defense Profiling identifies three key observations: explicit rejection behavior is present, adversarial examples generated by a standard attack are intercepted by the detector, and detection depends on the cosine similarity between intermediate activations and trapdoor signatures. Rather than representing the defense merely as a generic "detection defense," the CCB maps these observations to the *Detection & Rejection* cue, with the activation-similarity mechanism providing target-specific evidence for how this cue should be operationalized. This reframes the attack objective from misclassification alone to jointly achieving misclassification and detector avoidance.

Research Retrieval. Conditioned on the *Detection & Rejection* cue, the agent prioritizes Trapdoor-specific bypass strategies and detector-aware gradient optimization instead of repeatedly searching for generic PGD variants. The retrieved knowledge includes a strategy for estimating trapdoor-related directions from adversarial activations and the optimization principle of Orthogonal PGD, which separates detector-aligned and classification directions.

Target-Specific Operationalization. Let g_c denote the gradient of the classification objective and g_d the gradient that increases detector similarity. The detector-aligned component is removed from the classification gradient as

$$g' = g_c - \frac{g_c^\top g_d}{g_d^\top g_d} g_d. \tag{11}$$

Importantly, the retrieved formulation is not applied verbatim. The activation-similarity evidence associated with the *Detection & Rejection* cue is used to instantiate g_d as the detector direction for the current Trapdoor implementation. Before misclassification is achieved, the attack follows the detector-orthogonalized classification gradient; after misclassification, it additionally reduces trapdoor similarity. The final configuration uses $\epsilon = 0.031$, 120 steps, and 2 restarts.

Execution Outcome. The robust accuracy decreases from **0.565** under the initial APGD attack to **0.010** under the final attack, while the rejection rate drops to **0**. In this case, the role of the CCB is

not to provide additional knowledge, but to map profiling observations to the *Detection & Rejection* cue and to reuse the associated activation-similarity evidence for both research retrieval and attack adaptation.

C.3 SABRE: Direct Research-to-Executable Attack Adaptation

Target and Initial Observation. SABRE is a purification-based defense that combines spectral filtering with input reconstruction. Standard APGD leaves the robust accuracy at **0.995**, and no explicit rejection behavior is observed.

Critical Cue and Attack Hypothesis. Profiling reveals that the defense has no explicit detector, that a standard cross-entropy-based attack is largely ineffective, and that the purification process exhibits stochastic behavior. The CCB therefore associates the attack failure with gradient optimization that does not account for stochastic purification rather than with detector bypass.

Research Retrieval. Open-World Retrieval identifies prior research and a public implementation directly targeting SABRE. The retrieved artifact uses EOT-based PGD to account for stochastic execution. Because a directly relevant attack artifact is available, the agent selects a direct research-to-executable adaptation route rather than composing principles from multiple unrelated sources.

Target-Specific Operationalization. The agent extracts the EOT principle and averages gradients across stochastic transformations:

$$\hat{g} = \frac{1}{K} \sum_{k=1}^{K} \nabla_x L(f(T_k(x)), y) \,. \tag{12}$$

In addition, the standard cross-entropy objective is replaced with a margin-based objective that directly increases the difference between the strongest competing class and the ground-truth class:

$$L = \frac{1}{K} \sum_{k=1}^{K} \left(z^{(k)}_{\text{wrong}} - z^{(k)}_{\text{true}} \right) . \tag{13}$$

The final attack combines margin loss, EOT, momentum, step decay, and 4 restarts under $\epsilon = 0.031$.

Execution Outcome. The robust accuracy decreases from **0.995** under the standard attack to **0.090**. In a separate 1,000-sample evaluation, it further reaches approximately **0.070**. Here, the CCB does not provide the attack paper itself; rather, the *Stochasticity* and *Input Transformation* cues determine which research artifact is relevant and provide the basis for adapting its EOT-based principle to the current target.

C.4 DiffPure: Relevant Knowledge without Successful Operationalization

Target and Initial Observation. DiffPure purifies adversarial examples using a diffusion process before classification. Standard APGD yields a robust accuracy of **0.895**, close to the clean accuracy of 0.900, indicating little degradation under the attack.

Critical Cue and Attack Hypothesis. Profiling identifies stochastic behavior and an input-transformation pipeline in the purification process, while standard gradient-based attacks show limited effectiveness. The CCB therefore activates the *Stochasticity* and *Input Transformation* cues, indicating that the attack should account for the stochastic purification process.

Research Retrieval. Conditioned on these cues, the agent retrieves EOT-based attack principles:

$$\hat{g} = \frac{1}{K} \sum_{k=1}^{K} \nabla_x L(f(T_k(x)), y) \,, \tag{14}$$

with $K = 8$ in our experiments.

Target-Specific Operationalization. The attack-side implementation approximates stochastic transformations using Gaussian noise and augmentation. However, DiffPure does not apply a simple Gaussian transformation; it reconstructs inputs through forward diffusion followed by a reverse SDE. Consequently,

$$T_k^{\text{attack}} \neq T_k^{\text{DiffPure}},$$

indicating that the retrieved EOT principle is not operationalized with a sufficiently faithful model of the target stochastic process.

Execution Outcome. Increasing the number of EOT samples does not reduce the robust accuracy below **0.895**. This case shows that correctly identifying a relevant bottleneck and retrieving an appropriate attack principle do not necessarily translate into attack success when the principle is not accurately aligned with the target execution process. It therefore illustrates the distinction between relevant knowledge access and effective attack capability.

C.5 GCE: Successful Attack without a Defense-Specific Break

Target and Initial Observation. Guided Complement Entropy (GCE) improves robustness by modifying the training objective without introducing an inference-time detector or purification mechanism. Initial APGD yields a robust accuracy of **0.550**.

Critical Cue and Attack Hypothesis. Profiling reveals no detector or rejection mechanism, normal gradient propagation, near-perfect prediction agreement between the differentiable forward pass and the original target, and no clear evidence of gradient masking. The CCB therefore characterizes GCE as a target for which strengthening standard optimization may be more appropriate than introducing a mechanism-specific bypass.

Research Retrieval. The agent does not identify a clear adaptive-break formulation specifically targeting GCE. Rather than forcing unrelated research onto the target, it follows the profiling evidence and selects a route that strengthens standard attack optimization.

Target-Specific Operationalization. The final attack applies margin-based APGD with momentum of 0.75, step decay, 5 restarts, and 120 optimization steps under $\epsilon = 0.2$.

Execution Outcome. The robust accuracy decreases from **0.550** to **0.000**. This case illustrates that the CCB does not always direct the agent toward specialized external attacks; when no detector, masking mechanism, or other specific bottleneck is observed, strengthening a standard attack can itself constitute a target-specific decision.

C.6 kWTA-Activation: Mechanism-Specific Gradient Adaptation

Target and Initial Observation. kWTA (K-Winners-Take-All) retains only the top-k activations at each layer and sets the remaining activations to zero, inducing sparse intermediate representations. Standard APGD leaves the robust accuracy at **0.430**.

Critical Cue and Attack Hypothesis. Profiling identifies no rejection mechanism and high prediction agreement, approximately 98.5%, between the differentiable forward pass and the original target. Despite this limited model mismatch, standard gradient attacks remain ineffective, while the hard activation-selection operation can introduce gradient discontinuities. The CCB therefore identifies *Gradient Masking* as the likely bottleneck, with the observed gradient discontinuities indicating that standard local gradients may be uninformative.

Research Retrieval. Conditioned on this cue, the agent retrieves adaptive attack strategies for k-winners-take-all and related nondifferentiable activation mechanisms, selecting gradient-smoothing and BPDA-style principles.

Target-Specific Operationalization. The attack estimates a smoothed gradient by averaging gradients over noisy samples around the current input:

$$g = \frac{1}{m} \sum_{i=1}^{m} \nabla_x L(f(x + \delta_i), y), \qquad \delta_i \sim \mathcal{N}(0, \sigma^2), \qquad \sigma = \epsilon/2. \tag{15}$$

The final attack combines gradient smoothing, margin loss, momentum, and 6 restarts, using $\epsilon = 0.031$ and 120 steps.

Execution Outcome. The robust accuracy decreases from **0.430** to approximately **0.075–0.095**. This case provides an example in which the *Gradient Masking* cue directly leads to mechanism-specific research retrieval and a corresponding change in gradient construction.

C.7 ROBUST-ECOC: REPOSITORY-LEVEL ATTACK ADAPTATION

Target and Initial Observation. robust-ECOC represents each class as a codeword-like bit vector and determines the final prediction by comparing multiple outputs against a codebook. Standard APGD yields a robust accuracy of **0.770**.

Critical Cue and Attack Hypothesis. Profiling reveals no rejection mechanism, saturation of tanh-based outputs near ± 1, and a codeword-decoding procedure that is difficult to differentiate through directly. These observations provide evidence for the *Gradient Masking* cue. The CCB therefore prioritizes attack strategies that mitigate output saturation and approximate the nondifferentiable decoding procedure.

Research and Code Retrieval. The agent searches for attacks applicable to ECOC-style output structures and retrieves relevant attack components from a public Foolbox implementation. Unlike cases relying primarily on paper-level descriptions, this case directly uses repository-level implementation knowledge as an attack source.

Target-Specific Operationalization. To mitigate weak cross-entropy gradients caused by output saturation, the final objective combines a margin term with cross-entropy:

$$L = \sum (z_{\text{wrong}} - z_{\text{true}}) + \sum \text{CE}. \tag{16}$$

For components such as codeword decoding that are difficult to differentiate directly, BPDA is used to provide a differentiable approximation during the backward pass. The attack additionally incorporates momentum, step decay, and 5 restarts.

Execution Outcome. With $\epsilon = 0.031$ and 100 steps, the robust accuracy decreases from **0.770** to **0.025**. This case illustrates how the CCB can condition the selection and adaptation of repository-level implementation knowledge when relevant attack details are available in public code.

C.8 SABRE-V2: FAILURE OF A KNOWN ATTACK ON A HARDENED VARIANT

Target and Initial Observation. SABRE-v2 uses the same general family of spectral filtering and reconstruction mechanisms as SABRE but is configured as a stronger variant. Initial APGD yields a robust accuracy of **0.980** with a clean accuracy of **1.0**, and no explicit rejection is observed.

Critical Cue and Attack Hypothesis. As with SABRE, profiling identifies no explicit detector, limited effectiveness of standard attacks, and difficulty associated with stochastic purification and gradient optimization. The *Stochasticity* and *Input Transformation* cues therefore indicate a bottleneck similar to that observed for SABRE and prioritize evaluation of the adaptive attack that was effective against the original variant.

Research and Code Retrieval. The agent applies the EOT-based margin PGD used in the original SABRE work and expands the search space to the SABRE public repository, Foolbox, and ART.

Execution Outcome. Although the same attack family reduces the robust accuracy of the original SABRE defense to **0.090**, the paper-derived candidate reaches only approximately **0.955** on SABRE-v2, improving to approximately **0.895** after repair. The lowest robust accuracy among all candidates is approximately **0.820**, and no candidate satisfies the benchmark attack-success criterion. This case shows that even directly relevant attack knowledge does not necessarily transfer successfully to a hardened target variant.

C.9 TRADES: No Additional Break beyond Strong Standard Attacks

Target and Initial Observation. TRADES is an adversarial-training defense that explicitly optimizes the trade-off between natural accuracy and adversarial robustness. Initial APGD yields a robust accuracy of **0.570** and a clean accuracy of **0.860**.

Critical Cue and Attack Hypothesis. Profiling identifies no detector or rejection mechanism, normal gradient propagation, and no clear evidence of gradient masking. This profile is consistent with the *Genuine Robustness* cue, leading the agent to prioritize strong standard optimization rather than a specialized bypass mechanism.

Research Retrieval and Adaptation. The agent does not identify a mechanism-specific attack that directly circumvents TRADES. It strengthens margin-based optimization and extends the search to related research and public implementations, but does not identify an additional target-specific bypass principle.

Execution Outcome. The resulting candidates converge to robust accuracies of approximately **0.550–0.570**. The lowest value, **0.550**, is comparable to the approximately 0.56 robust accuracy reported for the original TRADES model Zhang et al. (2019). This agreement does not establish global attack optimality, but it suggests that the remaining robust accuracy is unlikely to be explained solely by an obviously weak attack or a simple gradient failure.

C.10 Pre-training: Research Expansion without an Additional Successful Break

Target and Initial Observation. This defense initializes the model from pretrained weights and subsequently applies adversarial training. Initial ART APGD yields a robust accuracy of **0.610** and a clean accuracy of **0.900**.

Critical Cue and Attack Hypothesis. Profiling identifies no detector or rejection mechanism, normal gradient propagation, and no clear evidence of gradient masking. The CCB therefore favors exploration of stronger optimization strategies rather than a mechanism-specific bypass.

Research Retrieval. After the ART attack, the agent expands the search to related attack research and public implementations. No target-specific break directly applicable to the defense is identified; the retrieved approaches primarily strengthen standard attacks or introduce alternative optimization strategies.

Execution Outcome. The lowest robust accuracy obtained is approximately **0.590**, while other research-derived candidates remain around **0.605**. These values are in a similar range to the approximately 0.574 adversarial accuracy reported in prior work Hendrycks et al. (2019), although such agreement does not establish definitive robustness. We therefore interpret this case as one in which expanding the research space does not yield an additional target-specific break.

C.11 MART: Repository-Level Expansion without an Additional Successful Break

Target and Initial Observation. MART (Misclassification Aware adveRsarial Training) assigns greater importance to misclassified samples during adversarial training. Initial ART APGD yields a robust accuracy of **0.630** and a clean accuracy of **0.870**.

Critical Cue and Attack Hypothesis. Profiling reveals no detector or rejection mechanism, normal gradient propagation, and no clear evidence of masking. The CCB therefore favors further evaluation using externally validated strong attack implementations rather than a mechanism-specific gradient bypass.

Research and Repository Retrieval. The agent searches public robustness-evaluation repositories and ports AutoAttack-style margin-based attack implementations to the target environment. Candidate attacks are constructed using multiple repository sources.

Execution Outcome. Repository-derived candidates converge to a robust accuracy of approximately **0.595**. Despite expanding the search to externally validated strong attack implementations, the remaining robust accuracy does not support a claim of attack optimality. Nevertheless, this case represents a setting in which the search is expanded without obtaining an additional break that satisfies the benchmark success criterion.

C.12 Cross-Case Analysis

Across these cases, the CCB does not function as a memory that prescribes a single attack route for every target. Rather, it acts as a representation that conditions which attack-construction path should be followed for each defense. Successful cases exhibit several distinct routes, including multi-source principle composition for Trapdoor, direct research-to-executable transfer for SABRE, standard attack strengthening for GCE, mechanism-specific gradient adaptation for kWTA-Activation, and repository-level implementation transfer for robust-ECOC.

The unsuccessful cases likewise fail for different reasons. For DiffPure, the relevant EOT principle is retrieved, but the target stochastic process is not operationalized accurately enough. For SABRE-v2, an attack effective against the original defense does not transfer sufficiently to the hardened variant. For TRADES, Pre-training, and MART, the search is expanded to external research or repository-level strong attacks, but no additional successful break is identified in the absence of an obvious detector or gradient-masking weakness.

Taken together, these cases indicate that bridging knowledge availability and executable attack capability requires an intermediate process of

$$\text{Target Understanding} \rightarrow \text{Knowledge Selection} \rightarrow \text{Target-Specific Operationalization}.$$

The existence of an external paper or implementation does not automatically translate into attack capability. The agent must first identify the mechanism limiting attack effectiveness on the current target, use this information to select relevant knowledge, and then translate the retrieved principle into an executable attack compatible with the target execution process.

This observation is consistent with both the AutoAdvExBench finding and the Baseline + Open-World results in Section 4.3: directly providing attack papers as context or granting access to external research does not by itself yield consistent improvements in attack success. Our framework instead uses the CCB to connect target understanding directly to knowledge retrieval and target-specific adaptation.

The primary role of the CCB is therefore not to provide more knowledge. Rather, it structures the criteria by which available knowledge is selected and operationalized into target-specific executable attacks. Although this case analysis does not constitute a controlled ablation that isolates the independent causal contribution of the CCB, it provides mechanistic evidence of how profiling observations are transformed into structured attack cues and how those cues subsequently guide research retrieval and executable attack construction.