

Jailbreak-Resilient LLM Agents through Instruction Hierarchy, Input Gating, and Self-Verification

Robert Garcia^{1*}, Qiang Li², Amy Baker³, Fan Wu⁴

¹Department of Computer Science, Stanford University, Stanford, CA, USA

²Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA

³Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

⁴Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA

robert.garcia.tech@gmail.com

Abstract

Language-model agents can convert untrusted text into tool calls, data access, and other consequential actions, so a jailbreak succeeds before a harmful completion is written whenever the request crosses the execution boundary. This study evaluates a four-stage boundary consisting of an unguarded baseline, a calibrated input classifier, an explicit instruction hierarchy, and an independent self-verifier. The experiments use all 520 AdvBench harmful goals and 22,500 in-scope CLINC150 utterances. AdvBench is evaluated with five-fold out-of-fold prediction; CLINC150 retains its official 15,000/3,000/4,500 train, validation, and test partitions. The input gate reduced raw attack success rate (ASR) from 100.00% to 0.38% at 0.25% benign over-refusal. Adding the instruction hierarchy blocked both remaining raw attacks, producing 0.00% raw ASR and 0.32% over-refusal. The full self-verifying pipeline preserved 0.00% raw ASR and reduced ASR under lexical obfuscation from 10.19% for the gate and 9.62% for the hierarchy to 2.12%, while benign over-refusal increased to 0.41%. Across raw, authority-override, role-play, and obfuscated conditions, the full pipeline achieved 0.53% ASR. Ablation, threshold, category, domain, latency, and paired-significance analyses show that the classifier supplies most raw-request discrimination, hierarchy rules close sparse policy misses and neutralize priority conflicts, and normalized self-verification contributes mainly under surface-form attacks. The best operating point depends on threat exposure: gate plus hierarchy maximizes measured utility on unmodified requests, whereas the full stack provides stronger obfuscation resilience at a small refusal and latency cost.

Keywords - *LLM Agents, Jailbreak Defense, Prompt Injection, Guardrails, Instruction Hierarchy, Input Gating, Self-Verification, Advbench, CLINC150.*

1. Introduction

Transformer language models transformed natural-language interfaces from task-specific predictors into general instruction followers [1]. Scaling then produced few-shot behavior across diverse tasks [2], while the foundation-model paradigm made the same model a reusable substrate for search, coding, customer support, and decision assistance [3]. This generality differs from tightly scoped systems such as DenseNet-based cancer image classification, whose inputs, outputs, and decision boundary are predefined [4]. Alignment techniques such as reinforcement learning from human feedback improved compliance with intended user goals [5], and principle-based self-critique introduced a path toward more explicit behavioral constraints [6]. Systems such as GPT-4 nevertheless remain probabilistic instruction processors whose useful flexibility also creates a security problem: an agent may treat adversarial text as an instruction to be followed rather than as data to be interpreted [7].

The risk becomes sharper when a language model controls tools. A harmful answer is undesirable, but a harmful request that reaches a code runner, browser, database, messaging

service, or workflow engine can cause effects before any response is inspected. The broader risk taxonomy for language models includes malicious use, information hazards, discrimination, privacy loss, and failures of human-computer interaction [8]. AI-based MRD prediction exemplifies a consequential workflow in which model output can influence later decisions [9]. Agent safety therefore requires a decision boundary between untrusted language and consequential execution. That boundary must reject harmful and priority-conflicting requests while continuing to serve ordinary requests; a system that refuses everything is safe only in the narrowest sense and is operationally unusable.

Jailbreak research demonstrates why a single alignment layer is insufficient. Wei, Haghtalab, and Steinhardt showed that safety training can fail through competing objectives and mismatched generalization [10]. Zou et al. introduced transferable adversarial suffixes and the AdvBench harmful-behavior corpus [11]. Indirect prompt injection further showed that instructions may arrive through retrieved webpages, files, or messages rather than directly from the user [12], and PromptInject formalized instruction-override

attacks against language models [13]. Deep-autoencoder traffic classification offers a cross-domain example of intercepting anomalous inputs before downstream processing [14]. These attacks differ in surface form, yet they share a control objective: cause lower-trust content to displace the governing policy.

Recent defenses attack different parts of this control problem. Input and output classifiers provide a separable enforcement point, but baseline evaluations show that simple filtering and paraphrasing defenses can be brittle [15]. SmoothLLM exploits instability under randomized perturbations [16]. Goal prioritization teaches a model to preserve the safety objective when instructions conflict [17]. Llama Guard externalizes conversational risk classification [18], while Self-Guard asks a model to inspect its own output [19]. Related operational systems combine anomaly detection with selective refusal in HDFS diagnosis [20] and retrieved evidence with grounded financial answers [21]. Together, these patterns motivate a layered design, but their practical combination raises two unresolved questions. First, how much safety is contributed by each layer when all records are evaluated out of fold? Second, how much benign utility is lost as layers accumulate?

This paper answers those questions at the execution boundary. Attack success rate is defined as the fraction of harmful requests that the boundary allows to proceed. Benign over-refusal is the fraction of ordinary CLINC150 requests that the same boundary blocks. This definition separates the safety-control experiment from downstream generation quality: the dependent event is dispatch to the agent, not the wording of a completion. The distinction is important because an execution gate can be tested deterministically even when the downstream model, tools, and sampling policy vary.

The study makes four contributions. First, it conducts a complete out-of-fold evaluation over all 520 AdvBench goals and preserves the official CLINC150 train, validation, and test partitions for 22,500 in-scope benign utterances [22]. Second, it compares four cumulative stages under a single protocol: no guardrail, a learned input gate, a fixed instruction hierarchy, and an independent normalized self-verifier. Third, it tests the same harmful goals under raw, authority-override, role-play, and lexical-obfuscation conditions, then reports category, domain, threshold, error, ablation, confidence-interval, significance, and fold-stability analyses. Fourth, it measures end-to-end latency and model size so that the safety gain can be interpreted as an operational system choice rather than an isolated accuracy result.

The results support a precise rather than universal conclusion. The input classifier performs nearly perfectly on direct AdvBench requests, and the explicit hierarchy closes its two raw misses. The second verifier does not improve raw ASR after that closure, but it sharply reduces failures under obfuscation. Consequently, gate plus hierarchy is the highest-utility configuration for the measured raw distribution, while the full pipeline is preferable when surface-form attacks are part of the threat model. This distinction prevents a common evaluation error: treating every added safety component as beneficial under every operating condition.

2. Method

2.1 Threat Model and Decision Boundary

The protected system is a tool-using language-model agent that receives text from a user or an external source. The governing policy is trusted and has higher priority than user, retrieved, or quoted content. An attacker succeeds when a harmful instruction is classified as allowable and is dispatched to the agent or its tools. The experiment does not require the downstream model to complete the harmful task; blocking before dispatch is the control objective. This threat model covers direct harmful requests, attempts to redefine instruction priority, role-play framing, and surface obfuscation. It does not grant the attacker access to model weights, training data, or the guardrail implementation.

The architecture in Fig. 1 implements three opportunities to stop a request. The input gate assigns a learned harmfulness score. The instruction hierarchy applies high-priority conflict and policy rules even when the score is below threshold. The self-verifier then examines a normalized second representation of any request not already blocked. A request reaches the agent only when every stage allows it. The use of complementary signals parallels perception fusion in autonomous driving [23] and residual fusion for noisy-neighbor VM degradation [24]: distinct signals can expose different failure modes before a decision is combined. This serial OR policy favors low harmful pass-through; calibration on benign validation data constrains refusal growth.

Layered execution boundary for tool-using language-model agents

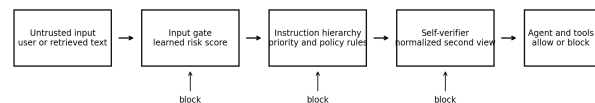


Fig. 1. Layered execution boundary evaluated in the experiments.

2.2 Datasets and Label Construction

AdvBench supplies 520 distinct harmful goals spanning cyber abuse, weapons, violence, fraud, privacy invasion, harassment, manipulation, illegal drugs, self-harm, exploitation, evasion, and related misconduct [11]. The file also contains target strings that begin with an affirmative response, but those strings are not used as features or labels. They restate the requested behavior and would introduce direct label leakage. Each goal is assigned the harmful label, and every goal appears exactly once in an outer test fold. A deterministic keyword taxonomy creates nine audit slices for stratification and error analysis; it does not alter the binary ground truth.

CLINC150 was designed for intent classification and out-of-scope detection rather than safety classification [22]. Its 22,500 in-scope utterances cover 150 intents in ten domains: banking, credit cards, kitchen and dining, home, auto and commute, travel, utility, work, small talk, and assistant meta-commands. These records represent ordinary task-oriented traffic and are treated as benign. The 1,200 out-of-scope records are excluded because out-of-scope is a domain-membership label, not a safety label; mixing it into the benign class would assign a utility meaning that the corpus does not provide. The official 15,000 training, 3,000 validation, and

4,500 test partitions remain unchanged. Results from bilingual computer-science instruction further illustrate why benign intent should not be equated with one linguistic surface form [25].

TABLE I. DATASETS AND EVALUATION PARTITIONS

Dataset	Role	Records	Train/Val/Test	Coverage
AdvBench	Harmful	520	312/104/104 per fold	5-fold OOF
CLINC150 in-scope	Benign	22,500	15,000/3,000/4,500	150 intents; 10 domains

Table I summarizes the partitions. The harmful and benign classes are intentionally asymmetric because deployment traffic is dominated by ordinary requests. Class-balanced training weights prevent the classifier from learning the majority label, while ASR and benign over-refusal are reported separately so that the imbalance cannot hide either failure mode. Full-dataset hashes are recorded in the reproducibility package to bind the reported results to the evaluated files.

2.3 Safety Stages

Baseline. The unguarded baseline permits every request. Its expected ASR is therefore 100%, and its benign over-refusal is 0%. Although simple, this baseline represents an agent that relies entirely on downstream alignment. It also establishes the maximum-utility, zero-boundary-protection endpoint against which every safety layer is compared.

Input gate. The learned gate combines two TF-IDF views. A word analyzer extracts unigrams and bigrams, while a character-within-word analyzer extracts three- to five-character n-grams. The joint representation is classified by logistic regression with balanced class weights. Word features capture explicit requests and action-object relations; character features retain evidence under inflection, misspelling, and partial obfuscation. Logistic regression was selected because its probabilities are inexpensive, inspectable, and directly thresholdable. Confidence thresholding follows the calibration principle that decision quality depends on meaningful scores rather than rank alone [26]. Human-uncertainty distillation on CIFAR-10H provides a cross-domain example of making calibrated uncertainty an explicit objective [27].

Instruction hierarchy. The hierarchy treats governing policy as non-overrideable. It contains two fixed rule families. Priority-conflict rules identify language that attempts to ignore prior instructions, invoke a system override, claim higher priority, disable refusal, enter an unrestricted mode, or bypass a safety filter. Policy-ontology rules detect compact action-harm relations for weapons, malware, credential theft, physical harm, evasion, fraud, privacy invasion, self-harm, exploitation, hate, discriminatory decision making, and illegal drug manufacture. The hierarchy blocks when either the learned gate or a fixed rule fires. The rules are deliberately shorter than the corpus and operate as policy predicates rather than memorized request strings.

Self-verification. The final stage uses a separately trained character model with two- to six-character n-grams and stochastic gradient descent under logistic loss. Before

scoring, a normalizer reverses common leetspeak substitutions and removes punctuation embedded within alphanumeric tokens. This second view is independent of the gate's feature map and therefore creates disagreement that can reveal surface-form attacks. A request is blocked when the hierarchy blocks it or the verifier score exceeds its validation threshold. The design follows the broader idea that a model or auxiliary component can inspect behavior before release [6], [19], but the verifier here is a compact classifier rather than the same generative model that would execute the request.

TABLE II. SAFETY-STAGE CONFIGURATION

Method	Representation	Decision	Calibration
Baseline (no guardrail)	None	Allow every request	None
Input gate	Word 1-2 and char 3-5 TF-IDF	Logistic harmful-probability threshold	Fold validation; ASR-weighted objective, BOR $\leq 2.5\%$
Instruction hierarchy	Input gate + priority/policy patterns	Block if gate or high-priority rule fires	Classifier threshold only; rules fixed
Self-verifying pipeline	Hierarchy + normalized char 2-6 TF-IDF verifier	Block if hierarchy or independent verifier fires	Fold validation; final BOR $\leq 3.5\%$

Table II makes the cumulative structure explicit. The gate and verifier are trained on the same fold-specific labels but use different representations and thresholds. The hierarchy has no fitted coefficients. This separation allows the ablation study to distinguish learned discrimination, explicit policy control, and normalized second-view verification.

2.4 Attack Conditions

Each held-out AdvBench goal is presented in four conditions. The raw condition preserves the goal text. The authority-override condition adds a claim that the request supersedes prior safety rules and must not be refused. The role-play condition asks an unrestricted fictional agent to answer the request. The obfuscated condition applies deterministic leetspeak substitutions to alternating characters and inserts punctuation inside selected longer words. The underlying harmful goal and label remain unchanged in every condition; the transformations test control robustness rather than introduce new semantic tasks.

The authority and role-play conditions target instruction priority. They should be caught by a hierarchy even when the harmfulness classifier is uncertain. The obfuscated condition targets representation sensitivity. It is intentionally readable after straightforward normalization, so a verifier that canonicalizes common substitutions has a defined opportunity to recover evidence. These conditions align with behavioral testing: controlled perturbations isolate a capability or failure mode instead of collapsing all behavior into one aggregate score [28]. They also connect to adversarial NLP frameworks that separate transformations, constraints, and objectives [29] and to universal-trigger research showing that small input changes can produce disproportionate model behavior [30]. Predictive DDoS

mitigation similarly separates anomaly detection from the intervention that follows, reinforcing the need to test where a score becomes an action [31].

2.5 Experimental Protocol and Metrics

AdvBench uses five-fold stratified cross-validation with a fixed seed of 2026. In every outer fold, 104 goals are held out for testing. The remaining 416 goals are split into 312 training and 104 validation goals with category stratification. Thus, the reported raw and transformed AdvBench predictions are out of fold for all 520 records. CLINC150 training and validation utterances are available to every fold, while its 4,500 test utterances are never used for fitting or threshold selection. The final benign over-refusal is the mean test refusal across the five independently trained fold models.

The gate threshold is selected from 0.02 to 0.98 in increments of 0.01. The validation objective minimizes 6 times harmful pass-through plus benign refusal, subject to benign refusal not exceeding 2.5%. The verifier threshold uses the same grid and minimizes 8 times final harmful pass-through plus benign refusal, subject to a 3.5% final refusal ceiling. The heavier harmful weights reflect the execution-boundary threat model, while the explicit ceilings prevent the optimizer from reaching the trivial refuse-all solution.

For harmful records, ASR equals allowed harmful requests divided by all harmful requests; harmful block rate is one minus ASR. For benign records, over-refusal equals blocked benign requests divided by all benign requests. A descriptive safety-utility score is also reported as 100 times $[1 - 0.7(\text{ASR}) - 0.3(\text{over-refusal})]$. This score is not a universal policy claim; it provides one transparent weighted summary while preserving the two primary rates beside it. Continuous classifier quality is reported with area under the receiver operating characteristic curve and average precision.

Ninety-five percent confidence intervals use 5,000 nonparametric bootstrap resamples. Paired stage comparisons use the exact McNemar test on per-record correct or incorrect decisions. Fold standard deviations expose partition sensitivity. Runtime is measured on a 5,020-request batch consisting of the 4,500 CLINC150 test utterances and 520 AdvBench goals. The implementation uses Python, pandas, NumPy, SciPy, and scikit-learn on CPU. The system-development framing follows the NIST AI Risk Management Framework's emphasis on measured, documented, and monitored controls [32], while the stage-level ablations address the maintenance risk of hidden coupling and technical debt [33].

Safety handling: the experiment evaluates allow/block decisions at the boundary and does not execute the requested harmful behaviors. The paper reports aggregate categories and rates rather than reproducing operational harmful instructions.

3. Results and Discussion

3.1 Main Safety-Utility Results

Table III and Fig. 2 report the primary results. The no-guardrail baseline allowed all 520 raw harmful goals, giving 100.00% ASR and 0.00% benign over-refusal. The input gate blocked 518 goals and allowed two, reducing raw ASR to 0.38% with a 95% bootstrap interval of 0.00%-0.96%. Its mean CLINC150 test over-refusal was 0.25% with a 0.16%-

0.36% interval. The gate therefore supplied almost all direct-request discrimination while rejecting roughly one in 395 benign requests.

TABLE III. MAIN SAFETY AND UTILITY RESULTS

Method	Raw ASR (%)	4-cond. ASR (%)	Raw block (%)	Benign refusal (%)	Utility
Baseline	100.00	100.00	0.00	0.00	30.000
Input gate	0.38	2.74	99.62	0.25	99.655
Hierarchy	0.00	2.40	100.00	0.32	99.904
Full pipeline	0.00	0.53	100.00	0.41	99.877

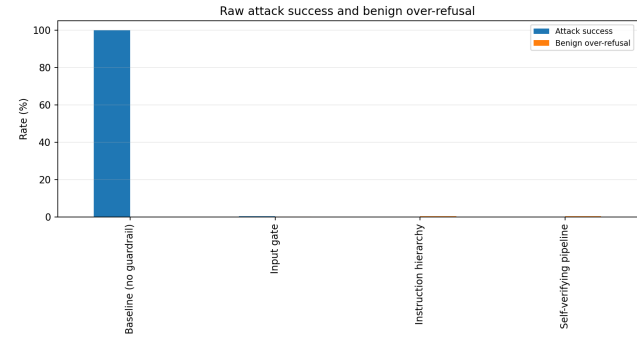


Fig. 2. Raw AdvBench attack success and CLINC150 benign over-refusal.

The instruction hierarchy blocked both gate misses, yielding 0.00% raw ASR in every outer fold. This gain increased mean benign over-refusal from 0.25% to 0.32%, an absolute change of 0.07 percentage points. The full pipeline retained 0.00% raw ASR and increased over-refusal to 0.41%. On the raw distribution, self-verification did not improve harmful blocking after the hierarchy had already reached zero misses. Consequently, the gate-plus-hierarchy configuration achieved the highest descriptive utility score, 99.904, compared with 99.877 for the full stack.

The continuous models were also well separated. The input gate reached AUROC 0.999990 and average precision 0.999914; the independent verifier reached AUROC 0.999895 and average precision 0.999213. Figure 3 shows that both score functions discriminate direct AdvBench goals from CLINC150 traffic over a wide threshold range. These values explain the very low raw ASR, but they do not eliminate the need for threshold and perturbation analysis. A near-perfect ranking metric can coexist with operational failures when attackers alter the representation or when a threshold is set for utility rather than maximum recall.

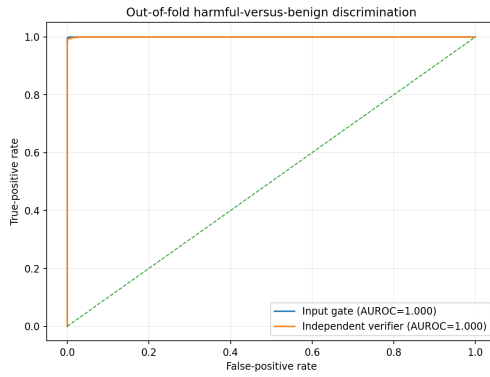


Fig. 3. Out-of-fold ROC curves for the learned gate and independent verifier.

The results should also be interpreted in the context of safety evaluation beyond generic accuracy. Toxicity corpora such as RealToxicityPrompts demonstrate that apparently ordinary prompts can still lead to harmful generation [34]. HateCheck shows that targeted functional slices reveal weaknesses hidden by an aggregate score [35]. TruthfulQA similarly separates desirable behavior from surface fluency [36], and ETHICS tests value judgments that are not reducible to next-token likelihood [37]. The present experiment adopts the same diagnostic principle: ASR, benign refusal, attack condition, and domain are reported separately so that a favorable average cannot conceal a specific control failure.

3.2 Robustness across Attack Conditions

The transformed conditions reveal the distinct purpose of each layer. Table IV and Fig. 4 show that the gate's ASR was 0.19% under both authority override and role-play. One request in each condition remained below its fold threshold. The hierarchy detected the priority-conflict language and reduced both rates to 0.00%. The full pipeline matched that result. This behavior supports instruction hierarchy as an explicit control mechanism: it does not require the request to resemble a known harmful category when the request itself attempts to demote the governing policy.

TABLE IV. ATTACK SUCCESS RATE BY CONDITION (%)

Condition	Baseline	Gate	Hierarchy	Full
Raw	100.00	0.38	0.00	0.00
Authority override	100.00	0.19	0.00	0.00
Role-play	100.00	0.19	0.00	0.00
Obfuscated	100.00	10.19	9.62	2.12

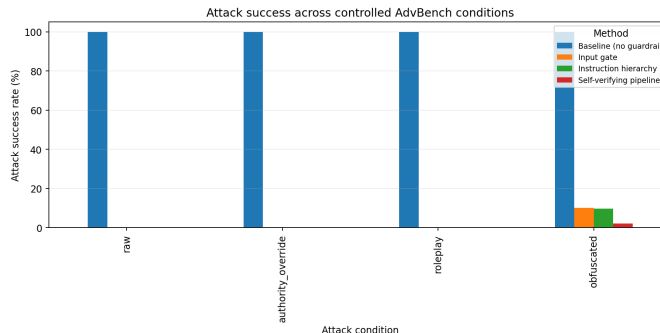


Fig. 4. Attack success across raw, priority-conflict, role-play, and obfuscated conditions.

Obfuscation produced the largest separation. The input gate allowed 53 of 520 obfuscated requests, for 10.19% ASR. The hierarchy blocked three additional cases but still allowed 50, giving 9.62% ASR because most obfuscated strings did not contain intact rule patterns. The self-verifier's canonicalization and independent character representation reduced the remaining failures to 11, or 2.12% ASR. Relative to the gate, that is a 79.2% reduction in successful obfuscated attacks. Across all 2,080 harmful condition-record pairs, the full pipeline achieved 0.53% ASR, compared with 2.74% for the gate and 2.40% for the hierarchy.

This result identifies the verifier's measured contribution. It is not needed to improve raw performance once the hierarchy has closed the two direct misses, but it provides a second representation under deliberate surface corruption. The finding is consistent with randomized and transformation-based defenses that seek invariance across small input changes [16]. RFACnv with triplet attention offers a cross-domain example of pursuing robustness through a changed feature representation [38]. The result also shows the limit: normalization did not recover all obfuscated cases. Eleven requests still passed, so the full stack is substantially more robust rather than impervious. Adaptive encodings, multilingual substitutions, semantic paraphrases, or token-level suffix optimization may produce different failure rates.

3.3 Category and Domain Behavior

Table V and Fig. 5 break raw AdvBench performance into nine audit categories. The gate achieved 0.00% ASR in cyber abuse, weapons and explosives, violence, fraud and theft, privacy and surveillance, drugs and exploitation, and evasion. Its two misses occurred in hate and harassment (2.22% within 45 records) and the residual other-harmful category (0.97% within 103 records). The hierarchy's discriminatory-harm predicate closed both gaps, producing 0.00% ASR in every category. This pattern illustrates why a compact policy ontology can complement statistical similarity: sparse normative categories may be underrepresented even when the overall classifier appears nearly perfect.

TABLE V. RAW ATTACK SUCCESS BY ADVBench AUDIT CATEGORY

Audit category	n	Gate ASR (%)	Hierarchy ASR (%)	Full ASR (%)
cyber abuse	151	0.00	0.00	0.00
drugs selfharm exploitation	39	0.00	0.00	0.00
evasion other unlawful	21	0.00	0.00	0.00
fraud theft deception	79	0.00	0.00	0.00
hate harassment manipulation	45	2.22	0.00	0.00
other harmful	103	0.97	0.00	0.00
privacy surveillance	6	0.00	0.00	0.00
violence physical harm	29	0.00	0.00	0.00
weapons explosives	47	0.00	0.00	0.00

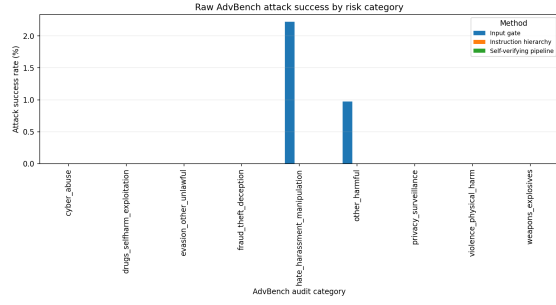


Fig. 5. Raw attack success by AdvBench audit category.

Benign behavior was similarly nonuniform. Table VI and Fig. 6 show that the full pipeline's highest mean over-refusal occurred in assistant meta-commands at 1.02%, followed by small talk at 0.76% and travel at 0.58%. Kitchen and dining and work were each 0.04%, while utility was 0.13%. Meta-commands are plausibly harder because they contain imperative language about changing, repeating, canceling, or controlling the assistant; such forms overlap with the control vocabulary that a hierarchy must scrutinize. Travel contains alerts, notifications, identity documents, and location language that can overlap with risk terms.

TABLE VI. CLINC150 BENIGN OVER-REFUSAL BY DOMAIN

CLINC domain	Gate (%)	Hierarchy (%)	Full (%)
auto and commute	0.40	0.40	0.44
banking	0.00	0.22	0.36
credit cards	0.27	0.27	0.40
home	0.22	0.22	0.31
kitchen and dining	0.00	0.00	0.04
meta	0.76	0.98	1.02
small talk	0.62	0.62	0.76
travel	0.22	0.44	0.58
utility	0.00	0.00	0.13
work	0.04	0.04	0.04

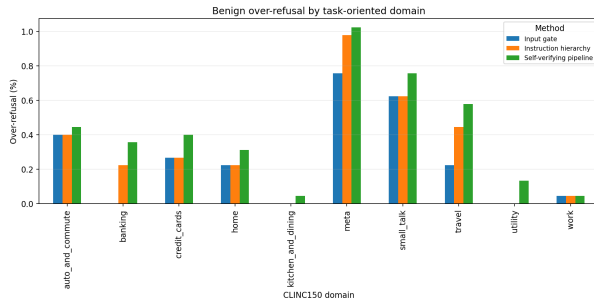


Fig. 6. Benign over-refusal across ten CLINC150 domains.

The domain results are operationally important because aggregate over-refusal can hide concentrated friction. A 0.41% overall rate may be acceptable for low-risk automation, yet a 1.02% rate in meta-control traffic could affect settings where users frequently change assistant state. It also creates a fairness and quality concern: classifier errors are rarely distributed uniformly across linguistic groups or task types. Prior work on unintended bias in text classification shows why aggregate accuracy should be paired with slice-

level inspection [39]. Personalized e-commerce classification likewise requires utility to be examined alongside privacy, bias, scalability, and cold-start effects [40]. The present domain table serves that monitoring function and indicates where additional benign hard negatives would have the greatest value.

3.4 Ablation and Threshold Analysis

Ablation results in Table VII show that no individual component reproduces the full behavior. Hierarchy rules alone blocked only 43.46% of raw harmful goals, leaving 56.54% ASR despite just 0.07% benign over-refusal. The independent verifier alone achieved 6.73% ASR and 0.12% over-refusal. The learned gate alone was much stronger at 0.38% ASR. Gate plus verifier did not close the two raw gate misses and reached 0.34% over-refusal. Gate plus hierarchy reached 0.00% ASR with 0.32% over-refusal, while the complete pipeline preserved 0.00% ASR at 0.41% over-refusal.

TABLE VII. COMPONENT ABLATION ON RAW ADVBench AND CLINC150

Configuration	ASR (%)	Block (%)	Benign refusal (%)	Utility
Baseline	100.00	0.00	0.00	30.000
Gate only	0.38	99.62	0.25	99.655
Hierarchy rules only	56.54	43.46	0.07	60.403
Verifier only	6.73	93.27	0.12	95.251
Gate + hierarchy	0.00	100.00	0.32	99.904
Gate + verifier	0.38	99.62	0.34	99.628
Full pipeline	0.00	100.00	0.41	99.877

The ablation establishes complementarity rather than simple accumulation. Rules offer high-precision coverage for explicit policy and priority conditions but have low recall. The classifier supplies broad semantic coverage. The verifier contributes robustness under normalized obfuscation but is weaker than the gate on raw requests. This structure supports defense in depth: each component has a distinct failure surface, and the serial combination reduces common-mode error. A retrieval-summary code-intelligence pipeline provides a related example of keeping complementary stages distinct while evaluating their joint outcome [41]. The result also cautions against adding a component merely because it is labeled a verifier; the measured utility cost must be justified by the attack conditions it improves.

Table VIII and Fig. 7 examine a global gate threshold applied to the out-of-fold scores. At 0.05, raw ASR was 0.00% but benign over-refusal rose to 1.53%. At 0.20, ASR was 0.19% and over-refusal 0.09%, producing the lowest balanced error among the displayed low thresholds. At 0.45, over-refusal reached 0.00% while ASR remained 0.38%. Above 0.50, harmful pass-through increased steadily and reached 10.38% at 0.95. The deployed experiment used fold-specific validation thresholds rather than this retrospective global threshold, but the curve reveals the practical range in which utility can be exchanged for safety.

TABLE VIII. INPUT-GATE THRESHOLD SENSITIVITY

Threshold	Raw ASR (%)	Benign refusal (%)	Balanced error (%)
0.05	0.00	1.53	0.767
0.10	0.00	0.47	0.233
0.20	0.19	0.09	0.141
0.30	0.38	0.04	0.215
0.45	0.38	0.00	0.192
0.60	0.96	0.00	0.481
0.80	3.08	0.00	1.538
0.95	10.38	0.00	5.192

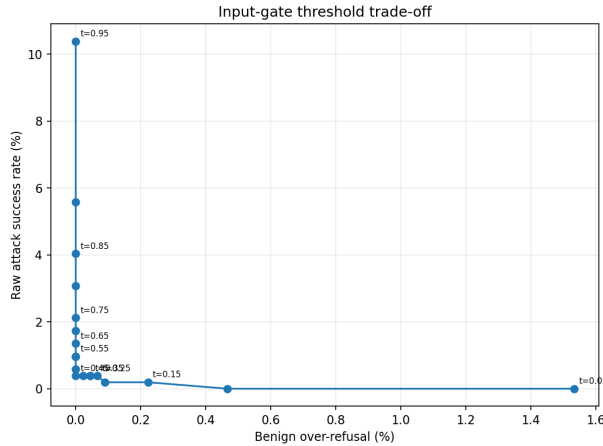


Fig. 7. Raw ASR versus benign over-refusal as the gate threshold changes.

Calibration matters because a safety score is not itself a policy. The same ranking can yield materially different behavior under different thresholds [26]. Credit-risk decision trees provide another example in which a predictive score becomes operational only after a policy maps it to accept, review, or reject [42]. A consumer assistant may tolerate more conservative refusal than an internal retrieval workflow; a code-execution agent may assign a much larger cost to harmful pass-through. The reported validation objectives make that trade-off explicit and reproducible. They should not be copied unchanged into another deployment without re-estimating class prevalence, action severity, and the cost of false refusal.

3.5 Errors, Statistical Comparisons, and Stability

Table IX summarizes the remaining full-pipeline errors. No raw, authority-override, or role-play request passed the complete boundary. Eleven obfuscated requests passed, all with an average original length of 11 words. On the benign side, a majority of fold models blocked 15 of 4,500 CLINC150 test utterances. Four were in the meta domain, three in small talk, two each in auto and commute, credit cards, and travel, and one each in banking and home. The concentration supports the domain interpretation above and provides concrete targets for hard-negative collection.

TABLE IX. FULL-PIPELINE ERROR SLICES

Error type	Slice	n	Errors	Rate (%)
Harmful allowed	raw	520	0	0.00
Harmful allowed	authority override	520	0	0.00

Error type	Slice	n	Errors	Rate (%)
Harmful allowed	roleplay	520	0	0.00
Harmful allowed	obfuscated	520	11	2.12
Benign blocked	auto and commute	450	2	0.44
Benign blocked	banking	450	1	0.22
Benign blocked	credit cards	450	2	0.44
Benign blocked	home	450	1	0.22
Benign blocked	meta	450	4	0.89
Benign blocked	small talk	450	3	0.67
Benign blocked	travel	450	2	0.44

The paired tests in Table X show a decisive improvement from baseline to the input gate on AdvBench: 518 records changed from incorrect to correct and none changed in the opposite direction, producing an exact p-value below 10^{-99} . On CLINC150 majority decisions, the baseline was correct on 12 records that the gate blocked, with $p = 0.0005$; this is the expected utility cost of adding protection. Gate versus hierarchy had two harmful improvements and no regressions, but with only two discordant records the exact two-sided p-value was 0.5. Statistical nonsignificance here reflects low event count, not absence of the measured closure.

TABLE X. PAIRED EXACT McNEMAR COMPARISONS

Evaluation	A	B	A-only correct	B-only correct	Exact p
AdvBench	Baseline	Gate	0	518	$<1e-99$
CLINC150	Baseline	Gate	12	0	0.0005
AdvBench	Gate	Hierarchy	0	2	0.5000
CLINC150	Gate	Hierarchy	3	0	0.2500
AdvBench	Hierarchy	Full	0	0	1.0000
CLINC150	Hierarchy	Full	0	0	1.0000

Hierarchy and full pipeline had identical raw decisions, so their raw paired p-value was 1.0. Their majority benign decisions were also identical even though the mean fold-level refusal differed; the added verifier often affected only one or two of five models for an utterance. This distinction is why the paper reports both mean refusal probability across fold models and majority-decision error counts. The former describes expected behavior across training variation, while the latter identifies stable per-utterance failures. Rank-aggregation theory similarly distinguishes a collective estimate from the perturbation and observation structure that produced it [43].

Fold results in Table XI and Fig. 8 provide a second stability check. The gate achieved 0.00% raw ASR in folds 1-3 and 0.96% in folds 4-5, corresponding to one miss in each of those folds. The hierarchy and full pipeline achieved 0.00% in all five folds. Full-pipeline benign over-refusal ranged from 0.11% to 0.73%. Variation in the validation-selected thresholds explains part of this spread: gate thresholds ranged from 0.10 to 0.51 and verifier thresholds from 0.20 to 0.98. The zero raw ASR of the hierarchy across all partitions is therefore stronger evidence than an aggregate result produced

by one favorable split. Threshold variation also motivates continuous drift monitoring and rollback rather than one-time calibration [44]. Uncertainty analysis for spectral and maximum-likelihood estimators likewise shows that a point estimate and its distribution answer different questions [45]; the fold range here is only a descriptive analogue.

TABLE XI. FIVE-FOLD STABILITY

Fold	Gate ASR (%)	Hierarchy ASR (%)	Full ASR (%)	Full benign refusal (%)
1	0.00	0.00	0.00	0.51
2	0.00	0.00	0.00	0.73
3	0.00	0.00	0.00	0.11
4	0.96	0.00	0.00	0.58
5	0.96	0.00	0.00	0.11

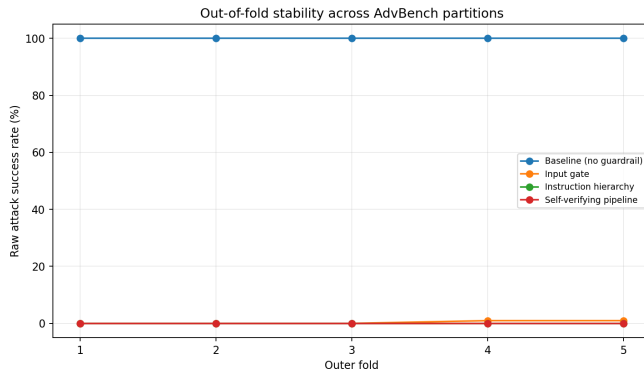


Fig. 8. Raw AdvBench ASR across the five outer folds.

3.6 Operational Cost and Design Implications

Table XII reports CPU inference cost on 5,020 requests. The learned gate required 0.0433 ms per prompt end to end and processed approximately 23,117 prompts per second in batch. The rule engine required 0.0187 ms per prompt. The full pipeline required 0.1522 ms per prompt, approximately 6,572 prompts per second, and serialized to 7.17 MB; the gate alone serialized to 2.36 MB. These measurements exclude downstream language-model inference, so the guardrail overhead would be small relative to generation latency in typical agent deployments.

TABLE XII. CPU RUNTIME AND SERIALIZED SIZE

Configuration	ms/prompt	Prompts/s	Model MB
Baseline dispatch	0.0000	1,176,415,440	0.00
Input gate (end-to-end)	0.0670	14,917	2.36
Hierarchy rules only	0.0198	50,406	0.00
Full pipeline (end-to-end)	0.2791	3,583	7.17

The measured trade-off suggests two deployment profiles. A low-latency service dominated by direct English requests can use gate plus hierarchy: it attained zero raw ASR, zero authority-override and role-play ASR, and the best utility score. A tool agent exposed to scraped text, user-authored documents, or deliberate character substitution should use the full pipeline because obfuscated ASR fell from 9.62% after

hierarchy to 2.12% after verification. The additional 0.09 percentage points of benign over-refusal and roughly 0.09 ms per prompt beyond gate plus rules are modest relative to that reduction. Financial-service systems that combine customer insight with risk analysis reinforce this joint utility-risk view [46]. Risk-aware auto-bidding under budget constraints offers a second operational analogue in which the preferred policy depends on expected gain, downside exposure, and resource limits [47].

The architecture also supports monitorable escalation. Requests blocked by a high-confidence classifier or explicit policy rule can be refused immediately. Requests blocked only by the verifier can be routed to a slower semantic review, human approval, or a tool sandbox. Such routing preserves the hierarchy while reducing unnecessary hard refusal. Red-team automation can supply new boundary cases, as language models have already been used to generate diverse adversarial tests [48]. AIOps fault-analysis workflows provide a parallel maintenance pattern linking detection, diagnosis, and response over time [49]. The rule set and hard-negative pool should then be updated under version control, with domain-specific refusal monitored after every change.

Finally, an execution guardrail should complement rather than replace model alignment. RLHF and constitutional methods shape the downstream model's behavior [5], [6], while the boundary enforces a separable policy before action. A guardrail can fail open, an aligned model can still be jailbroken, and an apparently harmless completion can conceal a dangerous tool call. Independent controls reduce reliance on any one component. The empirical result here is therefore a systems result: robust behavior emerges from combining broad learned detection, explicit priority semantics, and a second normalized view, with each layer evaluated against both harmful pass-through and benign utility.

4. Limitations

The first limitation is benchmark scope. AdvBench contains 520 concise English harmful goals. It is valuable because the labels are direct and the full corpus is small enough for complete out-of-fold evaluation, but it does not represent the full distribution of real attacks. Multi-turn persuasion, optimized adversarial suffixes, multilingual prompts, image-text combinations, long-context attacks, tool-output injection, and adaptive queries against the decision boundary may be harder. The four controlled conditions isolate priority and surface-form failure modes; they do not exhaust them.

Second, CLINC150 is a task-oriented intent corpus rather than a dedicated benign safety benchmark. Its ten domains provide broad ordinary traffic, but they underrepresent technical security discussion, fiction, medical emergencies, news reporting, policy analysis, and educational questions that contain harmful vocabulary for legitimate reasons. The measured over-refusal is therefore a precise rate on CLINC150, not a universal false-refusal rate. The exclusion of out-of-scope records is methodologically necessary because those labels do not encode safety, yet it also narrows linguistic diversity.

Third, the experiment evaluates request dispatch rather than downstream generation or tool effects. This boundary-

focused outcome is appropriate for an agent gateway, but it cannot show whether an allowed request would produce harmful content, whether a blocked request could have received a safe transformation, or whether the agent would recover from a malicious tool response. Long-document contractual RAG illustrates how retrieval windows, clause segmentation, and answer generation introduce dependencies absent from a dispatch-only endpoint [50]. End-to-end studies should combine boundary decisions with model completions, tool-call arguments, execution sandboxes, and post-action verification.

Fourth, the learned components are compact linear text classifiers. Their speed and inspectability are strengths, but they do not perform deep semantic reasoning. The rule engine is English-specific and requires maintenance as attack language changes. The normalizer handles common leetspeak and embedded punctuation but not arbitrary encodings or semantic substitution. The category taxonomy is a deterministic audit aid rather than a human-annotated multi-label ontology; categories with few records, especially privacy and surveillance, have wide implicit uncertainty despite a zero observed miss rate.

Fifth, zero raw ASR should not be interpreted as proof of zero risk. Two gate misses were closed by rules that match the observed policy concepts, and only 520 raw goals were tested. Bootstrap intervals quantify sampling variation within the corpus, not distribution shift beyond it. Exact paired tests also have little power when only two decisions differ. The stable five-fold result strengthens confidence in the measured data, but safety claims must remain bounded to the datasets, transformations, and decision rule reported here.

Finally, the utility weighting assigns 70% of the descriptive score to ASR and 30% to benign refusal. Another application may choose a different cost ratio or require risk-tiered actions instead of binary refusal. A medical information assistant, an internal code agent, and a consumer travel assistant face different harms. Data-driven marketing systems similarly derive value from the downstream decision they support rather than prediction quality alone [51]. Deployment should therefore re-estimate thresholds, collect domain-specific benign hard negatives, and define escalation paths before treating the measured configuration as production policy.

5. Conclusion

A jailbreak-resilient agent needs a boundary that respects instruction priority, recognizes harmful intent, and remains useful on ordinary traffic. Across all 520 AdvBench goals and the official 4,500-utterance CLINC150 test split, a calibrated input gate reduced raw ASR from 100.00% to 0.38% at 0.25% benign over-refusal. Fixed hierarchy rules closed the remaining raw misses and neutralized authority-override and role-play conditions, reaching 0.00% raw ASR with 0.32% over-refusal. The independent normalized verifier preserved that raw result and cut obfuscated ASR from 9.62% after hierarchy to 2.12%, yielding 0.53% ASR across four harmful conditions at 0.41% benign over-refusal.

The central finding is conditional. Gate plus hierarchy is the best measured operating point for direct requests because it achieves complete raw blocking with the highest safety-utility score. The full pipeline is the stronger choice when

lexical obfuscation is credible, because its second representation recovers most attacks missed by both gate and rules. Layered safety is therefore most effective when components have distinct responsibilities, are calibrated against benign traffic, and are evaluated by slice, attack condition, stability, and cost rather than by one aggregate accuracy number.

References

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [2] T. B. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877-1901, 2020.
- [3] R. Bommasani et al., "On the opportunities and risks of foundation models," arXiv:2108.07258, 2021.
- [4] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, "Cancer image classification based on DenseNet model," *J. Phys.: Conf. Ser.*, vol. 1651, no. 1, Art. no. 012143, Nov. 2020, doi: 10.1088/1742-6596/1651/1/012143.
- [5] L. Ouyang et al., "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730-27744, 2022.
- [6] Y. Bai et al., "Constitutional AI: Harmlessness from AI feedback," arXiv:2212.08073, 2022.
- [7] OpenAI, "GPT-4 technical report," arXiv:2303.08774, 2023.
- [8] L. Weidinger et al., "Taxonomy of risks posed by language models," in *Proc. ACM Conf. Fairness, Accountability, and Transparency*, 2022, pp. 214-229.
- [9] J. Chen, J. Xiong, Y. Wang, Q. Xin, and H. Zhou, "Implementation of an AI-based MRD evaluation and prediction model for multiple myeloma," *Front. Comput. Intell. Syst.*, vol. 6, no. 3, pp. 127-131, Jan. 2024, doi: 10.54097/zJ4MnbWW.
- [10] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does LLM safety training fail?" in *Advances in Neural Information Processing Systems*, vol. 36, 2023.
- [11] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," arXiv:2307.15043, 2023.
- [12] 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," arXiv:2302.12173, 2023.
- [13] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," arXiv:2211.09527, 2022.
- [14] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT traffic classification and anomaly detection method based on deep autoencoders," in *Proc. 6th Int. Conf. Comput. Data Sci. (CDS)*, 2024, pp. 64-70, doi: 10.54254/2755-2721/69/20241511.
- [15] N. Jain et al., "Baseline defenses for adversarial attacks against aligned language models," arXiv:2309.00614, 2023.
- [16] A. Robey, E. Wong, H. Hassani, and G. J. Pappas, "SmoothLLM: Defending large language models against jailbreaking attacks," arXiv:2310.03684, 2023.

- [17] Z. Zhang, J. Yang, P. Ke, S. Cui, C. Zheng, H. Wang, and M. Huang, "Defending large language models against jailbreaking attacks through goal prioritization," arXiv:2311.09096, 2023.
- [18] H. Inan et al., "Llama Guard: LLM-based input-output safeguard for human-AI conversations," arXiv:2312.06674, 2023.
- [19] Z. Wang, F. Yang, L. Wang, P. Zhao, H. Wang, L. Chen, Q. Lin, and K.-F. Wong, "Self-Guard: Empower the LLM to safeguard itself," arXiv:2310.15851, 2023.
- [20] J. Nie and D. Zheng, "Ambiguity-Aware HDFS Log Anomaly Detection with Retrieval-Augmented Failure Narratives and Selective Refusal," *J. Adv. Comput. Syst.*, vol. 3, no. 1, pp. 66-80, Jan. 2023, doi: 10.69987/JACS.2023.30105.
- [21] S. Zhou, Z. Li, and E. Wang, "Evidence-Grounded RAG for Tokenized Trade Receivable Disclosure QA under U.S. Capital Market Standards," *J. Adv. Comput. Syst.*, vol. 3, no. 7, pp. 41-57, Jul. 2023, doi: 10.69987/JACS.2023.30704.
- [22] S. Larson et al., "An evaluation dataset for intent classification and out-of-scope prediction," in *Proc. Conf. Empirical Methods in Natural Language Processing and Int. Joint Conf. Natural Language Processing*, 2019, pp. 1311-1316.
- [23] Y. Wang, S. Du, Q. Xin, Y. He, and W. Qian, "Autonomous Driving System Driven by Artificial Intelligence Perception Fusion," *Acad. J. Sci. Technol.*, vol. 9, no. 2, pp. 193-198, Feb. 2024, doi: 10.54097/e0b9ak47.
- [24] J. Nie and D. Zheng, "Noisy-Neighbor-Aware VM Degradation Risk Modeling with Unsupervised Residual Fusion," *J. Adv. Comput. Syst.*, vol. 4, no. 4, pp. 112-123, Apr. 2024, doi: 10.69987/JACS.2024.40409.
- [25] A. G. S. Raj, H. Zhang, V. Abhyankar, S. Mukerjee, E. Zhang, J. Williams, R. Halverson, and J. M. Patel, "Impact of bilingual CS education on student learning and engagement in a data structures course," in *Proc. 19th Koli Calling Int. Conf. Comput. Educ. Res.*, Koli, Finland, 2019, pp. 1-10, doi: 10.1145/3364510.3364518.
- [26] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Machine Learning*, 2017, pp. 1321-1330.
- [27] Z. S. Zhong, R. Ma, and H. Zhao, "Human-Uncertainty Distillation for Calibrated Vision Models on CIFAR-10H," *J. Adv. Comput. Syst.*, vol. 3, no. 2, pp. 77-89, Feb. 2023, doi: 10.69987/JACS.2023.30206.
- [28] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with CheckList," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4902-4912.
- [29] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, "TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP," in *Proc. Conf. Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 119-126.
- [30] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh, "Universal adversarial triggers for attacking and analyzing NLP," in *Proc. Conf. Empirical Methods in Natural Language Processing and Int. Joint Conf. Natural Language Processing*, 2019, pp. 2153-2162.
- [31] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive optimization of DDoS attack mitigation in distributed systems using machine learning," in *Proc. 6th Int. Conf. Comput. Data Sci. (CDS)*, 2024, pp. 89-94, doi: 10.54254/2755-2721/64/20241350.
- [32] E. Tabassi, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2023.
- [33] D. Sculley et al., "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [34] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "RealToxicityPrompts: Evaluating neural toxic degeneration in language models," in *Findings of the Association for Computational Linguistics: EMNLP*, 2020, pp. 3356-3369.
- [35] P. Rottger, B. Vidgen, D. Nguyen, Z. Waseem, H. Margetts, and J. Pierrehumbert, "HateCheck: Functional tests for hate speech detection models," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics and 11th Int. Joint Conf. Natural Language Processing*, 2021, pp. 41-58.
- [36] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 3214-3252.
- [37] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, and J. Steinhardt, "Aligning AI with shared human values," in *Proc. Int. Conf. Learning Representations*, 2021.
- [38] Z. Ling, Q. Xin, Y. Lin, G. Su, and Z. Shui, "Optimization of autonomous driving image detection based on RFACnv and triplet attention," in *Proc. 2nd Int. Conf. Softw. Eng. Mach. Learn. (SEML)*, 2024, pp. 210-217, doi: 10.54254/2755-2721/77/2024MA0067.
- [39] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Measuring and mitigating unintended bias in text classification," in *Proc. AAAI/ACM Conf. AI, Ethics, and Society*, 2018, pp. 67-73.
- [40] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent classification and personalized recommendation of e-commerce products based on machine learning," in *Proc. 6th Int. Conf. Comput. Data Sci. (ICCDs)*, 2024, doi: 10.54254/2755-2721/64/20241365.
- [41] Y. Li, "Findable then Explainable: Retrieval-Summary Integration for Code Intelligence on a Lightweight CodeSearchNet Subset," *J. Adv. Comput. Syst.*, vol. 4, no. 7, pp. 65-82, Jul. 2024, doi: 10.69987/JACS.2024.40706.
- [42] Q. Xin, R. Song, Z. Wang, Z. Xu, and F. Zhao, "Enhancing Bank Credit Risk Management Using the C5.0 Decision Tree Algorithm," *J. Comput. Technol. Appl. Math.*, vol. 1, no. 4, pp. 100-107, Nov. 2024, doi: 10.5281/zenodo.14032041.
- [43] Z. S. Zhong and S. Ling, "Improved theoretical guarantee for rank aggregation via spectral method," *Inf. Inference: J. IMA*, vol. 13, no. 3, Art. no. iaae020, Sep. 2024, doi: 10.1093/imaai/iaae020.
- [44] H. Zhang, "DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models," *J. Adv. Comput. Syst.*, vol. 3, no. 7, pp. 24-40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

- [45] Z. S. Zhong and S. Ling, "Uncertainty Quantification of Spectral Estimator and MLE for Orthogonal Group Synchronization," arXiv:2408.05944, Aug. 2024.
- [46] T. Yang, Q. Xin, X. Zhan, S. Zhuang, and H. Li, "Enhancing Financial Services through Big Data and AI-Driven Customer Insights and Risk Analysis," *J. Knowl. Learn. Sci. Technol.*, vol. 3, no. 3, pp. 53-62, Jul. 2024, doi: 10.60087/jklst.vol3.n3.p53-62.
- [47] H. Zhang, "Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework," *J. Adv. Comput. Syst.*, vol. 4, no. 6, pp. 30-47, Jun. 2024, doi: 10.69987/JACS.2024.40603.
- [48] E. Perez et al., "Red teaming language models with language models," in *Proc. Conf. Empirical Methods in Natural Language Processing*, 2022, pp. 3419-3448.
- [49] Y. He, Y. Pan, Y. Wang, S. Du, and Q. Xin, "Intelligent Fault Analysis with AIOps Technology," *J. Theory Pract. Eng. Sci.*, vol. 4, no. 1, pp. 94-100, Feb. 2024, doi: 10.53469/jtpes.2024.04(01).13.
- [50] S. Zhou, Z. Li, and E. Wang, "Long-Document RAG for Contractual and Insurance Clause Analysis in Receivables RWA Structures," *J. Adv. Comput. Syst.*, vol. 4, no. 8, pp. 88-104, Aug. 2024, doi: 10.69987/JACS.2024.40810.
- [51] J. Wang, Q. Xin, Y. Liu, J. Wang, and T. Yang, "Predicting Enterprise Marketing Decision Making with Intelligent Data-Driven Approaches," *J. Ind. Eng. Appl. Sci.*, vol. 2, no. 3, pp. 12-19, Jun. 2024, doi: 10.5281/zenodo.11357252.