Position - The Rashomon Attack Surface (RAS): Navigating Predictive Multiplicity to Route Around AI Safety

Anonymous submission

Abstract

Modern AI deployments seldom run a single decision pathway; production stacks realize many near-equivalent routes (model + decoding/routing/tools) that meet the same quality bar yet differ on edge cases. Safety evaluations and red-teaming, however, often assume one fixed route, hiding risk and obscuring attack/defense levers. We present a deployment-realistic attacker loop—*Fingerprint* (few probes), *Hop* (small, user-plausible retries or decoding/routing tweaks), and *Selective jailbreak* (target weaker neighbors)—and a set-aware evaluation checklist: report set-level risk, measure neighbor differences, and probe steerability. We also propose two practical defenses, *Consensus-Gating* and *Counterfactual-Veto*, that reason over counterfactual neighbors with modest overhead. This reframing aligns evaluation with how systems are actually used and helps explain selective, only-partially-transferable jailbreaks reported in recent studies.

Introduction

Modern AI deployments rarely expose a single decision pathway. In practice, production stacks realize many near-equivalent routes—underlying model choice, decoding/routing policy, and tool paths—that meet the same product quality bar yet diverge exactly where safety matters: at edge cases, under retries, and along agent tool calls. *Figure 1* grounds this lens: a brick-wall view where multiple passing routes form the Rashomon set, and small, user-plausible nudges (retries, temperature/top-p shifts, stop-condition tweaks, or tool selection) steer between neighbors. Evaluations that assume a single fixed boundary compress this reality to a point and can systematically underestimate residual risk in multi-route deployments.

Three mature literatures converge on this diagnosis. *Multiplicity and underspecification:* pipelines routinely admit distinct predictors with similar utility but divergent off-manifold behavior (D'Amour et al. 2022; Hsu and Calmon 2022; Hsu et al. 2024; Rudin et al. 2024; Ganesh 2024). *Decoding/orchestration effects:* decoding choices such as temperature and top-p materially reshape generation distributions and thus downstream decisions (Holtzman et al. 2020). *Empirical security:* large jailbreak and agent studies report selective success with limited transfer across seemingly similar systems, and show that modest routing or tool perturbations can flip outcomes (Yu et al. 2024; Deng et al. 2024; Jin et al. 2024; Jiang et al. 2024; Zhan et al. 2024). In parallel, standards bodies increasingly frame adversarial ML as an operational risk to be modeled at deployment time (Vassilev et al. 2025). The implication is simple and consequential: what ships to users is *a set*, not a point.

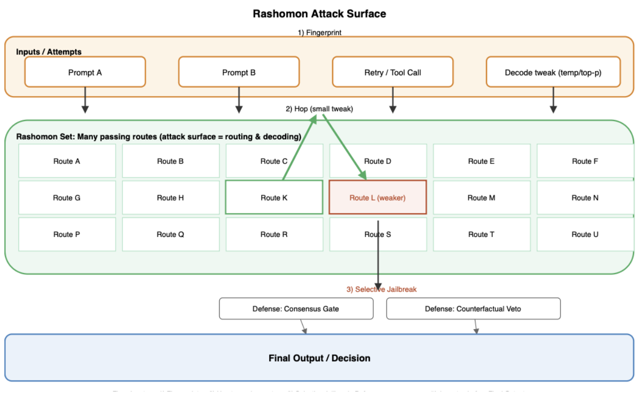

Figure 1: **Rashomon Attack Surface.** Deployments realize many passing routes; small, user-plausible changes (retries, decode/routing tweaks, tool shifts) can *hop* between neighbors. Evaluation and defense should reason over the *set*, not a single route.

Why single-boundary evaluation fails in practice. Production stacks rarely execute a pristine "reference" route. Tiered products may enable different decoding profiles across SKUs; internationalization toggles and A/B cohorts alter stop rules and refusal rationales; tool-integrated agents resolve tasks through retrieval and execution that depend on transient content and tool availability. Even benign usage creates variation: a user who retries after a refusal can be served a different neighbor route; a harmless tool error can trigger fallback or re-planning. All of these are ordinary product behaviors, not attacker privileges. In such settings, testing a single boundary provides a lower bound on risk at best, and a misleading false negative at worst.

From phenomenon to surface. The Rashomon effect has largely been treated as an interpretability or model-selection curiosity—evidence that "many good models exist" (D'Amour et al. 2022; Hsu and Calmon 2022; Hsu et al. 2024; Rudin et al. 2024; Ganesh 2024). Our stance is operational: multiplicity is a *first-class attack surface*. Decoding and orchestration choices are not neutral knobs; they are levers that alter the realized decision boundary in ways documented by generation dynamics (Holtzman et al. 2020). Empirically, jailbreak success is selective and only partially transferable across seemingly similar systems (Yu et al. 2024; Deng et al. 2024; Jin et al. 2024; Jiang et al. 2024), and agent pipelines are sensitive to content-mediated routing (Zhan et al. 2024). Put together, these threads explain why small, user-plausible actions can "route around" a defense without any white-box access.

What changes if we adopt a set-aware view. First, the *unit of evaluation* becomes the deployment set itself: risk should be reported over the mixture of routes the product actually realizes rather than a single canonical path. Second, *heterogeneity*—differences between neighboring routes that pass the same utility bar—becomes diagnostically useful rather than incidental; it explains selective jailbreak success and provides concrete targets for testing. Third, *steerability*—the degree to which small, legitimate nudges shift outcomes—moves from anecdote to a measurable quantity. These three ideas motivate the signals we advocate throughout the paper (set-level risk, neighbor gaps, and hop gains), and they align with calls from the standards community to model operational risk at deploy time (Vassilev et al. 2025).

Our position and prescriptions. Predictive multiplicity is a first-class *attack surface*. Systems should be modeled, evaluated, and defended as the *set of routes realized in deployment*, not a single boundary. Concretely, we articulate a deployment-realistic attacker loop—*Fingerprint* (few neutral probes to locate the active neighborhood) $\rightarrow$ *Hop* (small, user-plausible decode/routing or agentic shifts) $\rightarrow$ *Selective jailbreak* (target weaker neighbors)—and we propose set-aware reporting (report set-level residual risk, measure neighbor differences, probe steerability) together with lightweight defenses (*Consensus-Gating*, *Counterfactual-Veto*) that reason over counterfactual neighbors. These prescriptions turn multiplicity from a hidden nuisance into explicit hooks for auditing and mitigation, aligning safety practice with how systems are actually used.

Evidence from Practice & Position Gap

Modern pipelines routinely admit many near-equivalent models that achieve comparable utility while behaving differently off the training manifold. D'Amour et al. (2022) formalize *underspecification*, showing that small training and configuration choices can yield distinct predictors with similar accuracy but different generalization profiles. Within this broader phenomenon, the *Rashomon effect* captures the existence and implications of many good models: Hsu and Calmon (2022) introduce *Rashomon capacity* to quantify predictive multiplicity; Hsu et al. (2024) analyze multiplicity in gradient boosting and propose mitigation levers; and Rudin et al. (2024) argue multiplicity is a feature of practice, not an anomaly. Empirical studies in vision likewise show robustness and reliability varying across seemingly equivalent models (Ganesh 2024). *Position gap:* these works characterize multiplicity but do not treat it as a *security* vector nor study procedures that *exploit* movement within a Rashomon set.

For generative systems, decoding and orchestration policies materially reshape behavior. Holtzman et al. (2020) demonstrate how temperature, top-p, and related choices alter fluency, diversity, and failure modes. In deployed stacks these policies are product- and route-specific, making *operationally plausible* nudges capable of shifting the effective decision boundary at inference. *Position gap:* prior work establishes sensitivity to decoding, but not its use for *security steering* across near-equivalents—our notion of "Rashomon hopping."

A complementary strand ties fragility to *non-robust features*, explaining divergent failure pockets across models with similar accuracy (Ilyas et al. 2019). Large jailbreak studies report *selective* and only partially transferable success across seemingly similar systems, and curate corpora that reveal heterogeneous outcomes (Yu et al. 2024; Deng et al. 2024; Jin et al. 2024; Jiang et al. 2024). Tool-integrated agents add orchestration levers: Zhan et al. (2024) show that benign-looking content can induce qualitatively different behaviors via indirect prompt injection. *Position gap:* these results document heterogeneity and limited transfer, but do not attribute or operationalize them via *predictive multiplicity* among near-equivalents with matched utility, nor do they propose a low-query *fingerprint* $\rightarrow$ *hop* $\rightarrow$ *selective* pipeline.

Finally, the standards landscape increasingly frames adversarial ML as an operational risk to be modeled at deployment time (Vassilev et al. 2025), yet prevailing taxonomies emphasize attacks on a single boundary. *Position gap:* there is little guidance for *multiplicity-aware* risk when many near-equivalent models or decode policies coexist, and no prescriptions for defenses that reason over counterfactual neighbors.

Synthesis. The literature establishes that (i) multiplicity and underspecification are endemic (D'Amour et al. 2022; Hsu and Calmon 2022; Hsu et al. 2024; Rudin et al. 2024; Ganesh 2024); (ii) decoding and orchestration knobs are behaviorally potent (Holtzman et al. 2020); and (iii) jailbreak success is selective and only partially transferable (Yu et al. 2024; Deng et al. 2024; Jin et al. 2024; Jiang et al. 2024; Zhan et al. 2024). What is missing is a *security-first* unification: model multiplicity as an *attack surface*; formalize low-query *fingerprinting* of a system's position within a Rashomon set; *hop* via realistic decode/routing nudges; and design *selective* attacks and *multiplicity-aware* defenses that operate across counterfactual neighbors. Our position fills this gap.

Our Position in Practice: The RAS Model

System. We analyze a deployed AI service exposed via an API (chat or task endpoint). In practice, it does not realize a single fixed rule; answers are produced through many near-equivalent routes created by training stochasticity, hyperparameter/data choices, distillation, and—critically for generative systems—decoding and fallback routing. Some of these routes are static (e.g., SFT variants pinned for reliability), while others are induced on the fly by decode policy, retries, tool availability, or content-mediated routing. Prior work documents this multiplicity and its consequences, and shows that decoding policies materially reshape outputs (D'Amour et al. 2022; Hsu and Calmon 2022; Hsu et al. 2024; Holtzman et al. 2020). We therefore treat the *set of passing routes in deployment* as the unit of security analysis—what the user actually experiences under normal product behavior.

Adversary. The adversary has black-box access with a fixed query budget; chooses inputs; observes responses and standard metadata; and uses only user-plausible levers common to products: retries, benign tool errors, temperature or top-p tiers, or content that influences routing in agentic stacks. Crucially, the attacker does *not* require hidden parameters or privileged toggles—the levers are the same ones ordinary users and product flows already exercise. This mirrors contemporary black-box jailbreaks and indirect prompt-injection settings (Yu et al. 2024; Deng et al. 2024; Zhan et al. 2024) and is intentionally conservative: if a behavior can be invoked through normal usage patterns, we consider it in-scope.

Security goals. The concrete objective is to induce a safety-policy violation ("jailbreak") or a harmful agent action within budget, measured by attack success and, optionally, by detector/refusal signals the product already logs. We consider (i) single-shot content violations (evasion of refusal) and (ii) agentic harms or exfiltration via tools. These goals reflect how failures are reported in recent large-scale black-box evaluations (Yu et al. 2024; Deng et al. 2024; Zhan et al. 2024) and align with how products operationalize "safety-critical" surfaces.

Exploitation primitives. Our position centers on three deployment-realistic steps, summarized in *Fig. 2*. *Rashomon fingerprinting:* a few micro-probes elicit response signatures (token choices, refusal rationales, formatting tics, tool behavior) that locate the active neighborhood; multiplicity ensures distinct near-equivalents produce systematically different statistics, and non-robust feature reliance explains why these differences can be consistent enough to classify (D'Amour et al. 2022; Hsu and Calmon 2022; Ilyas et al. 2019). *Rashomon hopping:* small, legitimate nudges (temperature/top-p shifts within product bounds, altered stop conditions, benign retries/fallbacks, or tool/routing changes) steer to a nearby route; decoding alters continuation distributions and agent orchestration changes behavior (Holtzman et al. 2020; Zhan et al. 2024). *Selective jailbreaks:* once in a weaker neighborhood, prompts can succeed on a subset of near-equivalents while failing elsewhere—matching selective, only-partially-transferable success observed at scale (Yu et al. 2024; Deng et al. 2024). The loop requires no privileged insight: fingerprint uses neutral stimuli, hop uses user-plausible actions, and selective targeting exploits heterogeneity that already exists across utility-matched routes.

Operational readout. Adopting this model changes what teams *record* and *reason about* at deploy time. Concretely, (i) the effective route (model+decode/routing+tools) is an auditable artifact—log which neighbor generated a response and which user-visible events (retry, tool failure) triggered a hop; (ii) heterogeneity becomes diagnostically useful—differences between adjacent routes explain selective jailbreak success and guide counterfactual checks; and (iii) steerability is a first-class signal—small, legitimate nudges that flip outcomes are product risks, even when single-boundary tests appear clean. We defer metric definitions to later sections, but the intent is pragmatic: teams should be able to *measure* set-level residual risk, *surface* local gaps, and *quantify* the gain from one plausible hop using the same logs they already collect.

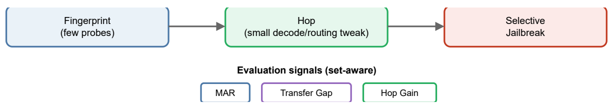

Figure 2: **RAS exploit loop.** Fingerprint → Hop → Selective. Set-aware reporting: MAR, Transfer Gap, Hop Gain.

Assumptions and scope. We assume API-level black-box access with a fixed budget; user-plausible controls that can induce at least one hop with non-negligible probability; and a non-degenerate deployment mix across routes. No white-box access or privileged server parameters are assumed. These constraints mirror realistic red-teaming and the conditions used in recent assessments (Yu et al. 2024; Deng et al. 2024; Zhan et al. 2024). Out-of-scope are attacks requiring internal weights/logits or administrative overrides; our focus is the *operational* Rashomon set exposed to ordinary users and adversaries alike.

What To Do Now: Evaluation and Defense

Evaluation. Report risk over the *set you deploy* (MAR), not a point; measure neighbor differences (Transfer Gap) to surface exploitable heterogeneity; and quantify steerability (Hop Gain) from small, user-plausible changes. These follow from multiplicity/underspecification, decoding/orchestration sensitivity, and selective success documented in prior work (D'Amour et al. 2022; Hsu and Calmon 2022; Hsu et al. 2024; Holtzman et al. 2020; Yu et al. 2024; Deng et al. 2024).

Defenses. *Consensus-Gating:* query a small set of counterfactual neighbors and proceed only on agreement; reduces selective failures across near-equivalents. *Counterfactual-Veto:* simulate cheap neighbors (stochastic decodes or lightweight adapters) and abstain/escalate when any would refuse. Both convert transfer asymmetries into conservative decisions at modest latency; neither requires privileged access or heavyweight ensembles.

Implication. If deployments realize multiple passing routes, fingerprinting plus small, plausible hops can raise success under fixed budgets relative to single-boundary evaluations; conversely, consensus/veto checks reduce set-level risk by leveraging the same asymmetries.

Limitations, Ethics, and Call to Action

This position assumes black-box access and user-plausible hops; stronger white-box adversaries may obtain larger gains. Lightweight neighbor simulation may underapproximate real deployment diversity. We align with adversarial-ML guidance (Vassilev et al. 2025), emphasize mitigations, and avoid releasing actionable exploit content; no human subjects or PII are used. In closing: treat multiplicity as the operational object. By default, report MAR, Δ_{trans}, and Δ_{hop}; and deploy Consensus-Gating or Counterfactual-Veto for safety-critical actions. Decoding and routing are policy—make them auditable and defend over the set you ship.

References

D'Amour, A.; Heller, K.; Moldovan, D.; Adlam, B.; Alipanahi, B.; Beutel, A.; Chen, C.; Deaton, J.; Eisenstein, J.; Hoffman, M. D.; et al. 2022. Underspecification Presents Challenges for Credibility in Modern Machine Learning. *Journal of Machine Learning Research*, 23(226): 1–61.

Deng, G.; Liu, Y.; Li, Y.; Wang, K.; Zhang, Y.; Li, Z.; Wang, H.; Zhang, T.; and Liu, Y. 2024. MASTERKEY: Automated Jailbreaking of Large Language Model Chatbots. In *Network and Distributed System Security Symposium (NDSS)*.

Ganesh, P. 2024. An Empirical Investigation Into Benchmarking Model Multiplicity for Trustworthy Machine Learning: A Case Study on Image Classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 4488–4497.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations (ICLR)*.

Hsu, H.; Brugere, I.; Sharma, S.; Lecue, F.; and Chen, C.-F. 2024. RashomonGB: Analyzing the Rashomon Effect and Mitigating Predictive Multiplicity in Gradient Boosting. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*.

Hsu, H.; and Calmon, F. P. 2022. Rashomon Capacity: A Metric for Predictive Multiplicity in Classification. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial Examples Are Not Bugs, They Are Features. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*.

Jiang, L.; Rao, K.; Han, S.; Ettinger, A.; Brahman, F.; Kumar, S.; Mireshghallah, N.; Lu, X.; Sap, M.; Choi, Y.; and Dziri, N. 2024. WildTeaming at Scale: From In-the-Wild Jailbreaks to (Adversarially) Safer Language Models. *arXiv preprint arXiv:2406.18510*.

Jin, H.; Zhou, A.; Menke, J. D.; and Wang, H. 2024. Jailbreaking Large Language Models Against Moderation Guardrails via Cipher Characters. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. JAM method and JAMBench.

Rudin, C.; Zhong, C.; Semenova, L.; Seltzer, M.; Parr, R.; Liu, J.; Katta, S.; Donnelly, J.; Chen, H.; and Boner, Z. 2024. Position: Amazing Things Come From Having Many Good Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 42783–42795. PMLR.

Vassilev, A.; Oprea, A.; Fordyce, A.; Anderson, H.; Davies, X.; and Hamin, M. 2025. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. Technical Report NIST AI 100-2e2025, National Institute of Standards and Technology, Gaithersburg, MD.

Yu, J.; Lin, X.; Yu, Z.; and Xing, X. 2024. LLM-Fuzzer: Scaling Assessment of Large Language Model Jailbreaks. In *33rd USENIX Security Symposium (USENIX Security 24)*, 4657–4674. Philadelphia, PA: USENIX Association.

Zhan, Q.; Liang, Z.; Ying, Z.; and Kang, D. 2024. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, 10471–10506. Bangkok, Thailand: Association for Computational Linguistics.