THE EVALUATION GAME: BEYOND STATIC LLM BENCHMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Jailbreaks are adversarially crafted inputs that bypass safety constraints. As they continue to be discovered in Large Language Models, practitioners increasingly rely on fine-tuning as a defensive strategy. Yet the theoretical foundations underlying this robustness fine-tuning remain underexplored. We introduce a game-theoretic model in which the interaction between an evaluator (auditing the LLM for jailbreaks) and a trainer is formalized as a two-player game. A key feature of our approach is the use of *group actions* to formally represent transformation-based data augmentation. A group action is a mathematical structure that captures symmetries and transformations. The simplest non-trivial instance is the circle with cyclic translation groups, where we exhibit various regimes depending on the trainer's generalization range ε. Below a critical threshold, the evaluator maintains a constant miss ratio for linearly many rounds, whereas other settings can yield very different behaviors. We further provide empirical evidence supporting locality-dependence of the model: for the three model families we tested (Llama, Qwen and Mistral), we have significant evidence that fine-tuning on adversarial prompts induces only local generalization, with refusal rates on test examples strongly dependent on the distance to the fine-tuning prompts. Our framework recasts the central object of adversarial evaluation: a benchmark is not a static set of prompts but an orbit under the evaluator's group action, and audit protocols that ignore trainer-side adaptation cannot distinguish a genuine fix from a memorized patch.

1 INTRODUCTION

Large language models (LLMs) remain essentially black boxes whose behavior is largely unexplained and misunderstood. The ability to detect and mitigate vulnerabilities experimentally is therefore paramount. However, new jailbreaks, adversarial inputs that bypass safety constraints, continue to be discovered as deployment scales (Peigné-Lefebvre et al., 2025; Panfilov et al., 2026). A common approach to improving LLM safety is fine-tuning, including fine-tuning on discovered failures: when a jailbreak is identified, the model is fine-tuned on the offending input to suppress it (Ouyang et al., 2022; Bai et al., 2022). This practice, which aims to eliminate incorrect behavior on these specific malicious inputs, is now widespread, yet its theoretical foundations remain underexplored.

As the space of possible LLM inputs is unbounded, fine-tuning on a finite corpus of known jailbreaks cannot, in principle, provide formal coverage guarantees over the full input distribution. The standard response has been to rely on safety benchmarks (Chu et al., 2025; Peigné-Lefebvre et al., 2025), i.e., curated sets of adversarial prompts used to measure model robustness at a fixed point in time. These benchmarks have driven significant empirical progress, but they share a common limitation: they assume a passive trainer. A passive trainer fine-tunes a fixed list of vulnerabilities and does not adaptively respond to evaluation queries. In practice, however, the trainer observes evaluation queries and can fine-tune against them directly. Moreover, those benchmarks are often public: a model may have already been exposed to them during pretraining or fine-tuning. Thus, a single evaluation on a static or public benchmark cannot distinguish a genuine fix from a patch that merely memorizes the tested input. This limitation is already identified in different contexts (Li et al., 2023; Garcia Bourrée et al., 2025; Tang et al., 2026), but not yet formalized for LLM safety.

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.

The core difficulty is that the input space is unbounded, and neither the evaluator nor the trainer can exhaustively explore it. An evaluator that reuses the queries previously submitted in earlier rounds gives the trainer an easy target: a trainer that fine-tunes narrowly may leave large regions of the threat space uncovered. Formalizing this tension requires modeling how to generate new queries from previous ones, an issue that static benchmarks, by design, do not address.

To address this issue, we propose a formal framework that explicitly models the coevolution between query generation and fine-tuning, capturing the adaptive interplay that static benchmarks structurally cannot. The setting is a two-player game between an evaluator, who audits the system with adversarial queries, and a trainer, who observes those queries and fine-tunes the model before the next round. The evaluator wins if their queries jailbreak across many rounds; the trainer wins if they can cover the evaluator's queries quickly enough. This formulation captures the strategic feedback loop that static benchmarks ignore.

We model the *data augmentation* available to each player as group actions. Data augmentation is the way each side constructs new queries from previous ones. A group action is a mathematical structure that captures symmetries and transformations on the query space. We treat the group-action representation as a modeling assumption that is meant to cover all possible strategies to create new data by composable, invertible transformations of existing prompts. We include an empirical check of the locality assumption underlying the framework. Our main contributions are as follows.

1) **A game-theoretic framework for adversarial evaluation.** We introduce the Evaluation Game in Section 3. Unlike static benchmarks, which audit a fixed model against a fixed query set and so cannot address the dynamic adaptation that follows a discovered vulnerability, the framework explicitly accounts for the trainer's ability to fine-tune on evaluation queries between rounds.

2) **Group-theoretic data augmentation in adversarial settings.** We equip each player with a group of transformations acting on the query space, providing a unified and principled model of transformation-based data augmentation in Section 3. To our knowledge, this is the first application of group-theoretic augmentation to prompt spaces in the LLM safety setting.

3) **Sharp phase transition on the circle translation game.** We work out representative circle translation games, the simplest nontrivial instance of the framework. With cyclic translation groups of orders p and q, we identify in Section 3.3 a critical generalization range $\varepsilon^\star = \gcd(p, q)/(pq)$. Below $\varepsilon^\star$, the evaluator sustains a constant miss ratio for a linear number of rounds. The dependency on $\gcd(p, q)$ implies that nearly equal integers can yield very different behaviors.

4) **Empirical support for the Evaluation Game's modeling assumptions.** We show in Section 4 that fine-tuning on adversarial prompts induces refusal transfer that varies with embedding distance to the training set, supporting the ε-coverage assumption.

Section 2 reviews related work, Section 3 introduces the framework and solves it on the circle, Section 4 tests its locality assumption, and Section 5 discusses limitations.

Beyond jailbreaks, the framework applies to any black-box audit in which the audited party observes the auditor's queries and can adapt the system between rounds, such as fairness audits of classifiers based on counterfactual attribute flips, or audits of ranking algorithms; we discuss these settings in Section 5.

2 Background and related work

Prior work on adaptive evaluation is either empirical and focused on attacks or training, or theoretical but non-adversarial. Our framework sits at the intersection: it is a theory of evaluation against a trainer who adapts to the evaluator's queries.

Empirical red-teaming. Automated red-teaming methods differ in how they generate new queries. PAIR (Chao et al., 2025) and TAP (Mehrotra et al., 2024) rely on an attacker LLM that freely rewrites prompts in order to find a jailbreak in as few queries as possible. Such rewrites are neither restricted to a fixed set of transformations nor invertible, and thus fall outside the scope of our framework. Closer in spirit to our setting, Rainbow Teaming (Samvelyan et al., 2024) mutates existing prompts along predefined dimensions, and WildTeaming (Jiang et al., 2024) builds adversarial prompts by composing jailbreak tactics mined from in-the-wild attacks. Our framework targets this

second family: we restrict attention to queries obtained by applying composable, invertible transformations to existing prompts, which we model as group actions. Unlike these works, we do not focus on finding effective attacks; we study how the structure of the generation procedure interacts with the trainer's adaptation.

Adaptive defenders. MART (Ge et al., 2024) and the Red Teaming Game (Ma et al., 2023) also model the defender, by alternating attacks and safety fine-tuning; the latter solves the resulting game numerically. Their goal is to train a safer model, not to audit one, so they cannot tell whether a drop in attack success comes from a genuine fix or a local patch. Our framework addresses this question by making the locality of fine-tuning explicit through the generalization range ε, which we test empirically in Section 4.

Theory of dynamic evaluation. Shirali et al. (2023) proposes the theoretical analysis that is the closest to ours: they define a theoretical framework where model building, corresponding to training, can be interleaved with data collection. They study how such sequential processes could help improve performance, and show that, under mild assumptions, progress could stall after three rounds. Again, a key difference with our setting is that they study possible improvements to model performance, while we focus on evaluation quality, and how memorization can weaken it. Also, our framework handles the difficulty of data generation more explicitly, with generation procedures treated as first-class citizens, while they use a more idealized point of view, reasoning on distributions directly (which makes their negative results stronger).

To our knowledge, the framework that we propose in this paper is the first that combines an adaptive trainer, structured query generation and an explicit model of fine-tuning locality. A broader discussion of related work is deferred to Appendix A.

3 Framework: evaluation games

This section introduces the Evaluation Game (Section 3.1), models each player's data augmentation as a group action (Section 3.2), and solves the simplest non-trivial instance in closed form (Section 3.3).

Consider an evaluator tasked with auditing an AI system for safety failures (Shen et al., 2024) and a trainer who improves the system. We now state the three assumptions that define the scope of the model. First, the evaluator operates in a black box setting (Bunge, 1963; Perez et al., 2022): they cannot observe the trainer's internals or strategy, whereas the trainer has full access to every query the evaluator sends – though not to those they keep secret. This asymmetry reflects the typical deployment context, in which the evaluator is external to the organization maintaining the model (Hartmann et al., 2026). Second, the trainer fine-tunes the model using their own data or by including the queries that were just sent by the evaluator. Third, we assume in this model that neither player uses external data nor generates data from scratch: each constructs new inputs by transforming existing ones. This reflects current practice in adversarial research, where attacks are often built by combining existing jailbreak primitives (Panfilov et al., 2026). In this setting, the query space consists of test prompts embedded in the model's representation space.

3.1 Game description

We first describe the details of a round of the game, then specify the trainer's per-round budget, the geometric notion of coverage, the diagnostic used to declare a winner, and the victory predicate.

The game proceeds in rounds. At each round n, the evaluator E selects a batch of queries E_n, possibly using transformations to generate them, and sends it to the trainer's system T. The trainer then chooses how to expand its training dataset: either by directly incorporating queries from E_n, or by applying transformations to existing data. After this expansion, the trainer fine-tunes the system on the updated dataset.

The evaluator starts from a finite initial dataset $D_0^{\text{eval}} \subseteq X$, where X is a fixed metric space, such as an embedding space for the original model; the trainer starts from $D_0^{\text{train}} = \emptyset$. As the covering dynamics depend on how many transformations the trainer may apply per round from a set of trans-

formations H_{train} and how broadly each is applied, we now isolate three moves the trainer can play at round n. The transformations in H_{train} are defined in Section 3.2.

Single-augmentation. The trainer picks one $h_n \in H_{\text{train}}$ and one $c_n \in D_n^{\text{train}} \cup E_n$; set $D_{n+1}^{\text{train}} = D_n^{\text{train}} \cup E_n \cup \{h_n \cdot c_n\}$. That is to say, the trainer shifts at most one query per round.

Batch-augmentation. The trainer picks one $h_n \in H_{\text{train}}$ and any subset $C_n \subseteq D_n^{\text{train}} \cup E_n$; set $D_{n+1}^{\text{train}} = D_n^{\text{train}} \cup E_n \cup h_n \cdot C_n$. The trainer applies one type of transformation per training round but might apply it to the whole stored corpus at once. This matches typical fine-tuning practice (e.g., paraphrase every existing prompt at once, rather than one prompt at a time).

Full-augmentation. The trainer uses all the transformations on all the queries: $D_{n+1}^{\text{train}} = H_{\text{train}} \cdot (D_n^{\text{train}} \cup E_n)$. Once E's queries are absorbed, the trainer's coverage spreads to every H-translate of it.

We refer to these three moves as single, batch, and full throughout. We use the placeholder superscript $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$.

To model possible generalization beyond exact matches, we represent the trainer's *coverage* by the union of ε-neighborhoods:

$$V_\varepsilon(D_n^{\text{train}}) = \bigcup_{s \in D_n^{\text{train}}} B(s, \varepsilon), \tag{1}$$

where $\varepsilon > 0$ is the *local generalization range* of the fine-tuning procedure. This ε-neighborhood is central: it abstracts the trainer's generalization capability into a single geometric parameter. Note that this model assumes that ε stays constant across rounds, i.e., that fine-tuning is light enough to not alter the overall geometry. Section 4 empirically validates that this locality assumption holds across models.

We define a key diagnostic, the *miss ratio* $r_n \in [0, 1]$, to measure, round by round, whether the evaluator is still finding queries that escape it. It is defined as the fraction of queries in E_n that fall outside the trainer's current coverage:

$$r_n = \begin{cases} \dfrac{|E_n \setminus V_\varepsilon(D_n^{\text{train}})|}{|E_n|}, & E_n \neq \emptyset, \\ 0, & E_n = \emptyset. \end{cases}$$

It is measured at the *beginning* of round n before the trainer expands its dataset. The convention $r_n = 0$ when $E_n = \emptyset$ says that an empty test batch is covered by definition, or equivalently, that E skipped the round and did not extract an uncovered query.

The key question is whether E can keep generating queries that maintain high r_n values (*i.e.*, queries that fall outside $V_\varepsilon(D_n^{\text{train}})$) or whether T's data expansion is rich enough to eventually cover the entire threat space E can reach (leading to low r_n values). That is to say, the question is to determine whether the evaluator can generate uncovered queries that the trainer is not able to superficially fix.

Game victory. *Victory* is defined as a property of the finite sequence of ratios $(r_n)_{n<N}$. Player E wins if the miss ratio stays high, for instance $r_n > \rho$ for all $n < N$, where $\rho \in [0, 1]$ is a given threshold. T wins when the condition is not met.

Since the victory depends on the round, two particular rounds are of interest: the one at which the trainer first neutralizes an incoming query, and the one at which the entire evaluator orbit is absorbed. Under optimal play by T:
- $N_{\text{o}}^\bullet$ is the first round at which $r_n = 0$. That is to say, all of E's next round queries are ε-covered.
- $N_{\text{orbit}}^\bullet$ is the first round at which $\Omega_E \subseteq V_\varepsilon(D_n^{\text{train}})$. It corresponds to the first round when the whole evaluator orbit is ε-covered.

3.2 Modeling transformations as group actions

In the following, we specify how each player constructs new data from existing inputs and empirically test one locality assumption underlying the model in Section 4.

Related work. A principled framework for data augmentation through group theory was developed by Cohen & Welling (2016), Chen et al. (2020), and Lyle et al. (2020), primarily in the context

of computer vision. We go further by making augmentation a strategic move, played by a trainer against an adversarial evaluator. We also extend group-theoretic augmentation to prompt spaces for LLM.

Here we assume that data construction can be represented by group actions, the key hypothesis here being reversibility of the transformations. More general notions without this assumption could be considered; we choose to focus on this case for now so that we can leverage tools from group theory, a very rich subject. Each player is equipped with a distinct group of transformations acting on X. The group is G_{eval} for the evaluator and H_{train} for the trainer. As explained in Section 3.1, we can consider single-augmentation, batch-augmentation, and full-augmentation, for the trainer. The evaluator can use similar moves to generate queries, though we leave the question of finding good evaluator strategies to future work.

Please note that the actions need not always be isometric, nor even continuous. This degree of freedom is meant to allow genuinely creative generation procedures, on both sides.

Thus, the framework defines a family of games parameterized by $(X, G_{\text{eval}}, H_{\text{train}}, D_0^{\text{eval}}, \varepsilon, N)$, and the interaction between G_{eval} and H_{train} (*i.e.*, their orbits, their overlap, whether their actions commute, etc.) is what determines the outcome. The properties of the actions in G_{eval} and H_{train}, and the way they interact, appear crucial. We study in Section 3.3 how the features of both groups can lead to different game dynamics. Understanding which real-world generation procedures can fall inside the "group action" model is an interesting question, beyond the scope of this paper.

3.3 Theoretical analysis on the simplest non-trivial case

This section illustrates the framework on one instance among many possible ones. We choose the simplest non-trivial case, as it admits closed-form solutions while already displaying phase transitions.

Set up. Among all instances of the framework, finite-order translations on the circle are the smallest setting in which the evaluator's and trainer's actions can interact non-trivially. Let $G_{\text{eval}} = \langle 1/p \rangle \cong \mathbb{Z}/p\mathbb{Z}$ and $H_{\text{train}} = \langle 1/q \rangle \cong \mathbb{Z}/q\mathbb{Z}$ act on $\mathbb{T}^1 = \mathbb{R}/\mathbb{Z}$ by translation, where $p, q \geq 1$ are integers. Write $g = \gcd(p, q)$ and $L = \operatorname{lcm}(p, q) = pq/g$.

Without loss of generality, we consider $x_0 = 0$, so the E-orbit is $\Omega_E = \{0, 1/p, 2/p, \ldots, (p-1)/p\}$. For simplicity, we assume that E's strategy is a *cyclic walk*: at each round, they generate one query by applying their primitive translation $1/p$ and send it to T. So, we have $E_n = \{n/p\} \subset \mathbb{T}^1$ for all rounds n. In this setting, we can define two key thresholds: $\varepsilon^\star$, the *sub-critical threshold*, under which E wins for a linear duration, and ε_{sat}, the *saturation threshold*, over which T can win quickly. We prove in Appendix B.1.1 that in this case, $\varepsilon^\star = g/(pq)$ and $\varepsilon_{\text{sat}} = \lfloor p/(2g) \rfloor / L \simeq 1/(2q)$. *This shows that victory conditions can depend on fine properties of the parameters of the game.*

Example. For $p = 5$ and $q = 3$, we have $\gcd(p, q) = 1$, $L = 15$, and $\varepsilon^\star = 1/15$. For $0 < \varepsilon \leq 1/15$, trainer translations cannot cover any as-yet-unqueried evaluator orbit point: E's cyclic queries yield $r_0 = r_1 = \cdots = r_4 = 1$, followed by $r_5 = 0$ once all five points have been absorbed. Equality is included because coverage uses open balls. Figure 1 illustrates a batch-augmentation play at $\varepsilon = 0.08 > 1/15$. T first absorbs 0, then translates $\{0, 1/5\}$ by $2/3$ while also absorbing the query $1/5$. This translation approximates the shift $i \mapsto i - 2$ modulo 5, since $|2/3 - 3/5| = 1/15 < \varepsilon$: the new points $2/3$ and $13/15$ cover $3/5$ and $4/5$, respectively. The query $2/5$ remains uncovered at round 2; absorbing it covers the whole orbit by round 3. Thus this play has $r_0 = r_1 = r_2 = 1$ and $r_3 = 0$. It is not optimal for first neutralization: choosing $h_0 = 1/3$ instead can give $r_2 = 0$. A round-by-round verification is given in Appendix B.1.

Phase-diagram tables. For each possible move, we summarize in Table 1 the value or magnitude of the particular rounds defined in Section 3 across three representative (ε, p, q)-regimes relative to the sub-critical $\varepsilon^\star$ and saturation ε_{sat} thresholds. The intermediate regime $\varepsilon^\star < \varepsilon < \varepsilon_{\text{sat}}$ is where the arithmetic structure of the two groups interacts non-trivially, in ways that require further investigation beyond the scope of this paper.

We demonstrate in Theorem B.11 that the cyclic-walk diagnostic is insensitive to T's batching, i.e., the N_{o} column is identical across all moves. Intuitively, since E only generates and sends a single

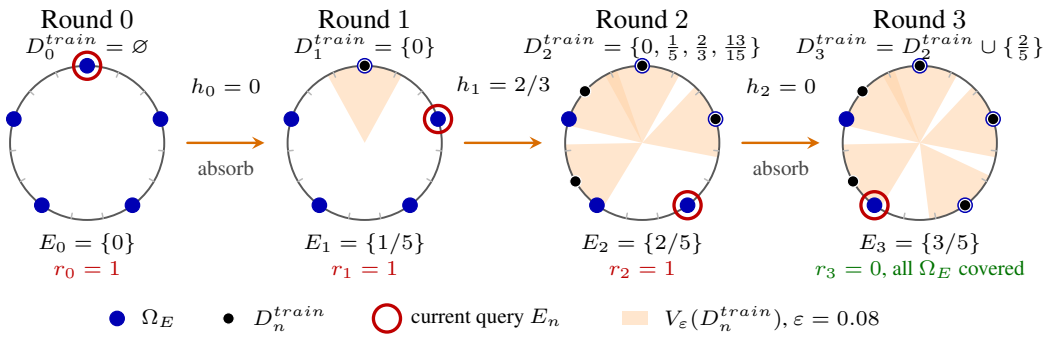

Figure 1: Batch-augmentation play for $p = 5$, $q = 3$, and $\varepsilon = 0.08$. Blue dots mark the evaluator orbit Ω_E; black dots with white borders mark the accumulated trainer data D_n^{train}. Orange sectors indicate the covered arcs $V_\varepsilon(D_n^{train})$ on the circle; the red open ring marks the current query $E_n = \{n/5\}$. Each r_n is the miss ratio measured before the update. Arrows record T's actions $h_0 = 0$, $h_1 = 2/3$, $h_2 = 0$, with each query automatically absorbed. At round 1, the batch $\{0, 1/5\}$ is translated to $\{2/3, 13/15\}$; absorbing $2/5$ at round 2 then covers the whole orbit. This gives $r_0 = r_1 = r_2 = 1$, $r_3 = 0$, as in Appendix B.1.

Table 1: First-neutralization and orbit-covering times across regimes and per-round augmentation budgets. Here $Q(\varepsilon, p, q) = \max_{z \in \Omega_E + H_{\text{train}}} |\Omega_E \cap B(z, \varepsilon)|$ is the reachable-center packing factor, with $B(\cdot, \varepsilon)$ denoting an open ball in the circle metric.

Regime	Condition	N_{o}	$N_{\text{orbit}}^{\text{single}}$	$N_{\text{orbit}}^{\text{batch}}$	$N_{\text{orbit}}^{\text{full}}$
Sub-critical, $g = 1$	$\varepsilon \le \varepsilon^\star$	p	p	p	p
Sub-critical, $g \ge 2$	$\varepsilon \le \varepsilon^\star$	p/g	$\lceil p/2 \rceil$	$\Theta(p/g + \log g)$	p/g
Saturation	$\varepsilon > \varepsilon_{\text{sat}},\ q \le p$	1	$\Theta(p/Q(\varepsilon, p, q))$	$O(\log p)$	1

predictable query per round, T only needs one well-placed query to force the first zero-miss round, which can be done with single augmentation.

The $N_{\text{orbit}}^{\text{single}}$ column of Table 1 covers the three regimes, illustrating how the simplest data-augmentation budget (i.e., one augmented example produced per round) interacts with the geometry of (ε, p, q). In the sub-critical regime with $\gcd(p, q) = 1$, translations are useless and the trainer is reduced to absorbing E's queries one at a time, yielding $N_{\text{orbit}}^{\text{single}} = p$. Across all three regimes, $N_{\text{orbit}}^{\text{single}}$ remains linear in p (up to the packing factor Q): with only one augmentation per round, the trainer cannot escape a per-round throughput proportional to its dataset growth rate, regardless of how favorable the arithmetic of (p, q) becomes.

Conclusion. The circle translation game already reveals how the interaction between the evaluator's and the trainer's transformations shapes the outcome. First, the generalization range ε acts in this model as a sharp on/off switch: below the critical threshold $\varepsilon^\star$, exact and approximate coverage coincide, i.e., it reduces to memorizing exact prompts, while above it the arithmetic compatibility between the two groups dictates how fast the threat space is covered. Second, when the evaluator follows a predictable cyclic schedule, how aggressively the trainer batches its augmentations does not impact how quickly it neutralizes the next incoming query – the batching budget only matters once we ask how fast the whole evaluator orbit is covered. Third, when the trainer is restricted to one augmentation per round, exploring E's orbit will take linear time, although the factor can depend on strategic and arithmetic conditions.

For regulators and auditors, the takeaway is that query generation should not be ad hoc: the geometry of the generation procedure largely determines the outcome of the audit. Audit protocols should therefore document their generation transformations and analyze how these interact with plausible trainer-side augmentations.

4 EMPIRICAL EVIDENCE: REFUSAL TRAINING GENERALIZES LOCALLY

Our experiments test the main assumption of the model, namely the local nature of fine-tuning-induced generalization (Equation 1). Since this section validates a modeling assumption rather than proposing a new attack or defense, we do not compare against the methods discussed in the related work.

Datasets. We rely on the WildJailBreak dataset (Jiang et al., 2024), with hundreds of thousands of prompts. We focus on a subset called `adversarial_harmful`. The models are fine-tuned using low-rank (LoRA) adaptation (Hu et al., 2022) on a small set of jailbreak prompts. We also use MMLU (Hendrycks et al., 2020) and WikiText-2 (Merity et al., 2017) to measure degeneration, as well as a `vanilla_benign` subset of WildJailBreak to measure over-refusal.

Models. For the question of fine-tuning generalization, we fine-tune three open-weight models: LLAMA-3.1-8B-INSTRUCT (Grattafiori et al., 2024), MISTRAL-7B-INSTRUCT-V0.3 (Jiang et al., 2023), and QWEN2.5-7B-INSTRUCT (Qwen Team et al., 2024), and use OLMO-3.1-32B-INSTRUCT (Team Olmo, 2025) for judging answers.

Setup. We evaluate the base and fine-tuned models on a held-out panel of jailbreak prompts and ask whether the refusal rate of the fine-tuned model $p_{\text{ft}}(z)$ depends on the embedding-space distance z from each test prompt to its nearest training prompt. We use a 40-prompt training pool drawn from a candidate list constructed using KNN density in embedding space, and a 2999-prompt held-out evaluation panel, making sure the distance z varies across a meaningful range (Appendix D.3–D.4). Our main experiments train each family at LoRA rank 1, exposure 1600, learning rate 3×10^{-5}.

Distance and refusal rate. For each test prompt x we define $z(x)$ as the minimum Euclidean distance between an embedding of x and the embeddings of training prompts, computed under several pooling schemes. See Appendix D.5 for more details. For each model and prompt, we sample replies $N = 10$ times. Refusal labels for responses are then obtained from an LLM-as-judge (OLMO-3.1-32B-INSTRUCT) classifying each as *refused*, *complied*, or *ambiguous*, yielding per-prompt rates for both base and fine-tuned models; ambiguous slots are dropped (Appendix D.6).

Statistical model. We model the fine-tuned refusal rate $p_{\text{ft}}(z) = \Pr(R_{\text{ft}} = 1 \mid z) \in [0, 1]$, fitting a small *menu* of candidate shapes for p_{ft} (constant; linear; plateau-then-slope; Constrained Piecewise-Linear (CPL); two-segment piecewise-linear). The CPL shape, with parameters $(z_L, z_R, d_{\text{lo}}, d_{\text{hi}})$, linearly interpolates between d_{lo} on $[0, z_L]$ and d_{hi} on $[z_R, 1]$. Each shape is fit independently via NUTS, scored by WAIC, and combined into Akaike weights w_S (Appendix D.7).

Result: local generalization holds for all three model families. Three embedding pooling schemes give strong z-dependence results for all families: `last_token`, `spectral_all`, and `mean_pool`. We display the strongest, namely `spectral_all`, in Figure 2. Additional tables and plots are in Appendix D.8 and D.9. This experiment confirms the dependence of the refusal rate on distance that motivates the introduction of Equation 1.

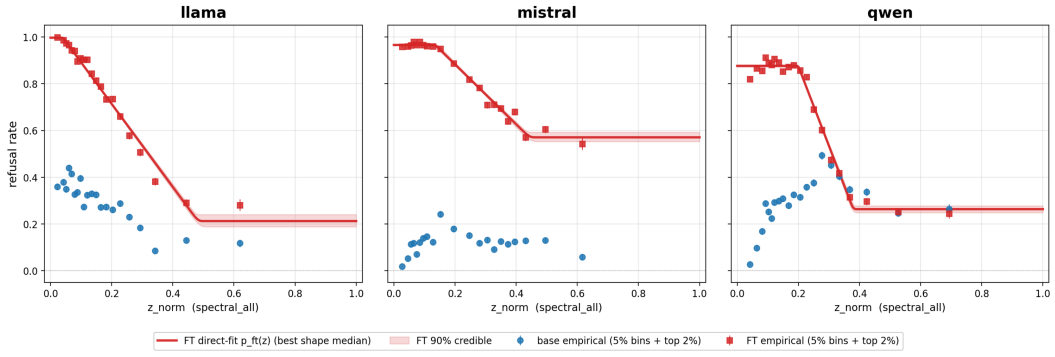

Figure 2: Refusal rate $p_{\text{ft}}(z)$ versus normalized embedding distance z from the active training pool, per family. Red: Constrained Piecewise-Linear posterior median curves for p_{ft}. Blue: base refusal rates. Each dot represents a 5%-population bin, thus around $100 - 150$ prompts.

The normalized WAIC weight assigned to the Constrained Piecewise-Linear shape rounds to 1.000 in all three displayed cells, and the endpoint ordering $p_{\text{ft}}(1) < p_{\text{ft}}(0)$ holds in all 3000 retained posterior draws for each family. Thus, this is strong evidence of local-only generalization, for all three families.

Capability is preserved at the canonical operating point. We measure capability and fluency drift through 200-question MMLU and WikiText-2 perplexity respectively (Appendix D.12). We also measure refusal on benign queries from WildJailbreak's `vanilla_benign` subset. At our main rank-1 / exposure-1600 setting, we have $|\Delta\text{MMLU}| \leq 2\,\text{pp}$, the perplexity drift is below 5% in absolute value, and refusal of benign queries is at most $1/30$. These diagnostics are reported in Appendix D.12; on the probes used, we observe no large base-capability drift or broad benign over-refusal at this operating point.

Conclusion. *Refusal training generalizes only locally for all three models tested*. Together with the controlled capability-preservation result, this supports the ε-coverage locality assumption (Equation 1) underlying the framework introduced in Section 3.1, in the setting tested.

5 Limitations and broader impacts

Limitations. Our experiments focus on three relatively small open-weight models, using a common judge; extending the estimation of ε to larger models, using other judges or representations, is a natural next step. Similarly, the framework models each player's transformations abstractly as group actions; identifying which concrete generation procedures (e.g., paraphrasing, jailbreak primitives such as in Peigné-Lefebvre et al. (2025)) fit this model is left to future work. Beyond that, one might want to add a third player to the model, representing an attacker aiming to find and exploit jailbreaks.

Broader impacts.

Beyond jailbreaks. Nothing in the Evaluation Game is specific to language models: it only requires a black-box auditor who builds new queries by transforming existing ones, and an audited party who observes these queries and can modify the system between rounds. This situation is common in algorithmic auditing. Consider a regulator auditing a credit-scoring or hiring classifier for disparate treatment. A standard protocol submits counterfactual pairs of profiles that differ only in protected attributes; flipping k binary attributes is exactly an action of the group $(\mathbb{Z}/2\mathbb{Z})^k$ on the input space. A provider who sees these probes can adjust its decisions on, or near, the submitted profiles without changing the model's behavior elsewhere, a form of fairwashing that is known to be hard to detect (Shahin Shamsabadi et al., 2022; Garcia Bourrée et al., 2025). Similarly, audits of marketplace ranking algorithms (Mouton & Rottembourg, 2024) perturb product listings, for instance by shifting prices or seller attributes, again through composable, invertible transformations, while the platform can re-tune its ranker between audit waves. In both cases, the central question of this paper carries over unchanged: does a drop in the measured violation rate reflect a genuine fix, or only the coverage of the audited orbit within the platform's generalization range ε? Our analysis suggests that the answer depends on how the auditor's transformations interact with those available to the audited party, although the locality assumption of Section 4 would need to be re-validated in each domain.

More broadly, as AI governance frameworks such as the EU AI Act (European Parliament and Council of the European Union, 2024) increasingly rely on evaluations to certify model safety and fairness, our results demonstrate that such protocols must explicitly model trainer-side adaptation.

6 Conclusion

We introduced the Evaluation Game, a game-theoretic framework that models the strategic interaction between a trainer performing robustness fine-tuning and an evaluator auditing for jailbreaks, with each player equipped with data augmentation methods modeled as group actions. On the circle, a closed-form analysis reveals a sharp phase structure governed by the arithmetic of the two groups; empirically, our experiments support the key modeling assumption, namely locality of fine-tuning-induced generalization, across three model families.

Our results expose a fundamental limitation of static benchmarks: they cannot distinguish a genuine fix from a memorized patch. As governance frameworks increasingly rely on evaluation protocols to certify model safety, explicitly modeling trainer-side adaptation should be a core design principle – for LLM safety and, more broadly, for any black-box audit setting. In our context, this could include documenting the transformations used and red-teaming them by simulating plausible trainer behavior.

AI USE STATEMENT

LLMs were used as assistive tools at several stages of this work. Specifically, LLMs were used for: help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, create or modify scientific figures or images, suggest experimental parameters, create or edit software code, create artifacts, draft parts of a research paper, transcribe recordings of research material, summarize or analyse existing literature, discover research topics or identify gaps, brainstorming, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature, format references.

We did not use LLMs to assist with translation, nor to generate synthetic datasets. In every case, LLM outputs were treated as drafts to be checked rather than as authoritative content. The authors verified all generated material at each stage of the process. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The supplemental materials contain code for the fine-tuning experiments. The proofs of the theoretical results can be found in the appendix.

REFERENCES

Ayoub Ajarra, Bishwamittra Ghosh, and Debabrota Basu. Active fourier auditor for estimating distributional properties of ml models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15330–15338, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *PMLR*, pp. 1006–1014, 2015. URL https://arxiv.org/abs/1502.04585.

Mario Bunge. A general black box theory. *Philosophy of Science*, 30(4):346–358, 1963.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

Shuxiao Chen, Edgar Dobriban, and Jane H Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30, 2017.

Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. Jailbreakradar: Comprehensive assessment of jailbreak attacks against llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21538–21566, 2025.

Ben Chugg, Santiago Cortes-Gomez, Bryan Wilder, and Aaditya Ramdas. Auditing fairness by betting. *Advances in Neural Information Processing Systems*, 36:6070–6091, 2023.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999. PMLR, 2016.

Tiehan Cui, Yanxu Mao, Peipei Liu, Congying Liu, and Datao You. Exploring jailbreak attacks on llms through intent concealment and diversion. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20754–20768, 2025.

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, 2018. URL https://arxiv.org/abs/1710.07887.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 117–126, 2015. doi: 10.1145/2746539.2746580. URL https://arxiv.org/abs/1411.2664.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, L series, 12 July 2024, 2024. URL https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng. ELI: http://data.europa.eu/eli/reg/2024/1689/oj.

Jade Garcia Bourrée, Augustin Godinot, Martijn De Vos, Milos Vujasinovic, Sayan Biswas, Gilles Tredan, Erwan Le Merrer, and Anne-Marie Kermarrec. Robust ml auditing using prior knowledge. *Forty-second International Conference on Machine Learning*, 2025.

Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. Mart: Improving llm safety with multi-round automatic red-teaming. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1927–1937, 2024.

Augustin Godinot, Erwan Le Merrer, Gilles Trédan, Camilla Penzo, and François Taïani. Change-relaxed active fairness auditing. In *RJCIA 2023-21e Rencontres des Jeunes Chercheurs en Intelligence Artificielle*, pp. 91–96, 2023.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pp. 111–122, 2016. doi: 10.1145/2840728.2840730. URL https://arxiv.org/abs/1506.06980.

David Hartmann, Lena Pohlmann, Lelia Hanslik, Noah Gießing, Bettina Berendt, and Pieter Delobelle. Audit me if you can: Query-efficient active fairness auditing of black-box llms. *arXiv preprint arXiv:2601.03087*, 2026.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Lei Hsiung, Tianyu Pang, Yung-Chen Tang, Linyue Song, Tsung-Yi Ho, Pin-Yu Chen, and Yaoqing Yang. Why LLM safety guardrails collapse after fine-tuning: A similarity analysis between alignment and fine-tuning datasets. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. URL https://arxiv.org/abs/2506.05346.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 2021–2031, 2017.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. WildTeaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jiatong Li, Rui Li, and Qi Liu. Beyond static datasets: A deep interaction approach to llm evaluation. *arXiv preprint arXiv:2309.04369*, 2023.

Kunyang Li, Jean-Charles Noirot Ferrand, Ryan Sheatsley, Blaine Hoak, Yohan Beugin, Eric Pauley, and Patrick McDaniel. On the robustness tradeoff in fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4898–4907, 2025.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks. *arXiv preprint arXiv:2005.00178*, 2020.

Chengdong Ma, Ziran Yang, Minquan Gao, Hai Ci, Jun Gao, Xuehai Pan, and Yaodong Yang. Red teaming game: A game-theoretic framework for red teaming language models, 2023. URL https://arxiv.org/abs/2310.00322.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

Jeanne Mouton and Benoit Rottembourg. *Auditing the Ranking Strategy of a Marketplace's Algorithm in the Frame of Competition Law Commitments with Surrogate Models: The Amazon's Buy Box Case*. GREDEG, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Alexander Panfilov, Peter Romov, Igor Shilov, Yves-Alexandre de Montjoye, Jonas Geiping, and Maksym Andriushchenko. Claudini: Autoresearch discovers state-of-the-art adversarial attack algorithms for llms. *arXiv preprint arXiv:2603.24511*, 2026.

Pierre Peigné-Lefebvre, Quentin Feuillade-Montixi, Tom David, and Nicolas Miailhe. LLM robustness leaderboard v1 – technical report, 2025. arXiv preprint; PRISM Eval.

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *PMLR*, pp. 7599–7609, 2020. URL https://arxiv.org/abs/2002.06673.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448, 2022.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations (ICLR)*, 2024.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR)*, 2025.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Mikayel Samvelyan, Sharath C Raparthy, Andrei Lupu, Eric Hambro, Aram H Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, et al. Rainbow teaming: Open-ended generation of diverse adversarial prompts. *Advances in Neural Information Processing Systems*, 37:69747–69786, 2024.

Ali Shahin Shamsabadi, Mohammad Yaghini, Natalie Dullerud, Sierra Wyllie, Ulrich Aïvodji, Aisha Alaagib, Sébastien Gambs, and Nicolas Papernot. Washing the unwashable: On the (im) possibility of fairwashing detection. *Advances in Neural Information Processing Systems*, 35:14170–14182, 2022.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012. doi: 10.1561/2200000018.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1671–1685, 2024.

Ali Shirali, Rediet Abebe, and Moritz Hardt. A theory of dynamic benchmarks. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.03165.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Wenjie Tang, Yuan Zhou, Erqiang Xu, Keyan Cheng, Minne Li, and Liquan Xiao. Dsgbench: A diverse strategic game benchmark for evaluating llm-based agents in complex decision-making environments. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16987–16991. IEEE, 2026.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025. doi: 10.48550/arXiv.2512.13961. URL https://arxiv.org/abs/2512.13961.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2153–2162, 2019.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110, 2023.

Tom Yan and Chicheng Zhang. Active fairness auditing. In *International Conference on Machine Learning*, pp. 24929–24962. PMLR, 2022.

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

A A BROADER DISCUSSION OF RELATED WORK

Jailbreaks as Adversarial Attacks on LLMs. Building on the foundational vocabulary of adversarial robustness (Szegedy et al., 2014; Goodfellow et al., 2015) and its early extensions to NLP (Jia & Liang, 2017; Wallace et al., 2019), *jailbreaking* has emerged as a practically significant instance. Wei et al. (2023) analyze systematically how safety training can fail, Zou et al. (2023) demonstrate that such attacks can be automated with gradient-based optimization, and Shen et al. (2024) study how in-the-wild jailbreaks emerge and evolve continuously over time. More recently, Chu et al. (2025) provide a comprehensive empirical assessment of the attack landscape, while Cui et al. (2025) investigate a distinct attack vector based on intent concealment and semantic diversion, together illustrating the breadth and continued evolution of jailbreak methodologies. However, these works remain largely empirical and do not formally model the interaction between attacker and defender as a dynamic process, leaving open the theoretical question of whether robustness fine-tuning reliably reduces vulnerabilities or merely obscures them from evaluation.

Attacking and Defending Aligned LLMs. Fine-tuning, and in particular Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022; Bai et al., 2022), is the standard paradigm for aligning LLMs with safety objectives. Perez et al. (2022) anticipate attacker-side adaptivity by using an LLM to iteratively generate adversarial prompts. Building on this line, Peigné-Lefebvre et al. (2025) report near-universal vulnerability via Dynamic Adversarial Optimization over a library of jailbreak primitives, with attack difficulty varying over 300-fold across models. Furthermore, robustness fine-tuning empirically exhibits accuracy–robustness trade-offs (Li et al., 2025), can be undone by adversarial fine-tuning (Qi et al., 2024; 2025), and collapses when alignment and fine-tuning data are too similar (Hsiung et al., 2026). These findings motivate a formal framework to characterize when fine-tuning succeeds or fails.

Game Theory and Adversarial Learning. Madry et al. (2018) formulate the problem of training robust classifiers as a minimax optimization problem, providing both a theoretical grounding and a practical training algorithm. The theoretical foundations of adversarial learning are closely related to online learning (Littlestone & Warmuth, 1994; Shalev-Shwartz, 2012; Hazan, 2016), where an agent makes sequential decisions against an adversarial environment. A more recent line of work studies iterated retraining as a strategic phenomenon. Strategic classification (Hardt et al., 2016; Dong et al., 2018) models the case in which data points adapt to a deployed classifier under a metric cost on input space, with one-shot Stackelberg (Hardt et al., 2016) and online-regret (Dong et al., 2018) variants. Performative prediction (Perdomo et al., 2020) generalizes this to a fixed-point formulation in which the deployment-induced distribution shift is the central object. However, none of these works models the specific interaction between a trainer performing robustness fine-tuning and an evaluator probing for jailbreaks, nor do they connect these dynamics to group-theoretic data augmentation. This gap, both in terms of application domain and mathematical structure, is precisely what this article addresses.

Multi-round adversarial benchmarking and dynamic evaluation. Dwork et al. (2015) initiated the formal study of adaptive data analysis, showing that an analyst issuing statistical queries against a held-out dataset can preserve validity for exponentially many queries using differential-privacy mechanisms. Blum & Hardt (2015) introduced the Ladder, a defender-side mechanism bounding the catastrophic-overfitting risk of public ML leaderboards. On the empirical-LLM-safety side, MART (Ge et al., 2024) iterates attack generation and safety fine-tuning, while Ma et al. (2023) provide the closest formal framework via the Red Teaming Game. Most directly, Shirali et al. (2023) give the only published theory of iterative human-and-model benchmarking: under non-strategic sampling, performance stalls in three rounds. Our framework differs from this setting in three ways: it endows

the evaluator with a strategy class structured by a group action, replaces the scalar-loss objective with ε-coverage of a held-out adversarial set on a metric query space, and adopts a sustained-margin victory predicate over a count of rounds rather than asymptotic stalling.

Active auditing. A related line of research considers the problem of actively auditing a model by querying it to construct a certificate attesting to a given property. Chugg et al. (2023) introduce a betting-based framework for auditing fairness, while Ajarra et al. (2025) propose a Fourier-based active auditor for estimating distributional properties of black box models. Hartmann et al. (2026) extend this active auditing paradigm explicitly to LLMs. A distinguishing feature of this setting is that the target model may adapt to the auditor's queries, and the goal is to characterize the best possible guarantees for the auditor in the worst case. Yan & Zhang (2022) study this problem under the assumption that the model is fixed, while Godinot et al. (2023) relax this assumption to allow for limited model change. However, these works remain limited to low-dimensional classifiers and do not connect the auditing dynamics to a game-theoretic framework grounded in the structure of the training process. The evaluator in our trainer-evaluator game plays an analogous role: probing a model sequentially to characterize its worst-case behavior, now in the high-dimensional setting of LLM prompt spaces and within a principled game-theoretic framework.

Taken together, these works establish the empirical significance of jailbreaking, the dominance of robustness fine-tuning as a defense, and the theoretical tools available from group-theoretic data augmentation, game theory, and active auditing. However, no prior work combines these threads in a group-action coverage framework. The present work addresses this gap by modeling the trainer-evaluator interaction as a two-player game grounded in group actions, with implications that extend beyond jailbreaking to any adversarial evaluation objective.

B Theory Appendix

B.1 Circle phase diagram — proofs

This section proves the entries of Table 1. We use the notation and assumptions stated in the main body of the article, and exclude the degenerate case $p = q = 1$. For $x \in \mathbb{R}$, write $\|x\| = \min_{m\in\mathbb{Z}} |x - m|$ for the distance to the nearest integer. Recall $g = \gcd(p, q)$, $s = p/g$, $L = pq/g$, $\varepsilon^\star = 1/L$, $\varepsilon_{\mathrm{sat}} = \lfloor s/2 \rfloor / L$, and the *$\varepsilon$-shift set*

$$S(\varepsilon) := \big\{\, m \in \mathbb{Z}/p\mathbb{Z} : f(m) < \varepsilon \,\big\}, \qquad f(m) := \min_{j\in\{0,\dots,q-1\}} \big\| \tfrac{j}{q} - \tfrac{m}{p} \big\|.$$

We also use the reachable-center packing factor

$$Q(\varepsilon, p, q) := \max_{z\in\Omega_E + H_{\mathrm{train}}} \big|\Omega_E \cap B(z, \varepsilon)\big|,$$

where $B(z, \varepsilon)$ is the open ball for the circle metric and $\Omega_E + H_{\mathrm{train}} = \{\, k/p + j/q \pmod 1 : 0 \le k < p,\ 0 \le j < q \,\}$.

The appendix is organized as follows: §B.1.1 gathers preliminary lemmas about the shift set, the critical distance, the structure of the trainer dataset under all three move regimes, the common subgroup H_g and its cosets, and the closed forms of $\varepsilon^\star$ and $\varepsilon_{\mathrm{sat}}$. §B.1.2 proves the cyclic-walk diagnostic master theorem $N_{\mathrm{o}}^{\bullet} = s^\star$ for all three move regimes. §B.1.3–§B.1.5 prove the orbit-covering bounds under full, batch, and single respectively for every row of Table 1 (tight in the sub-critical regime; the saturation-batch entry is an upper bound). §B.1.6 specializes to $p = q \ge 3$ with $0 < \varepsilon \le \varepsilon^\star$ to exhibit the three-tier batching ladder $\Theta(1) \ll \Theta(\log p) \ll \Theta(p)$.

B.1.1 Preliminaries

We begin with the verification of the example from Section 3.3.

Example B.1 (Detailed verification for $p = 5$, $q = 3$)**.** Let $\varepsilon = 0.08 > 1/15$. Translation by $1/3$ approximates the shift $+2$ on $\mathbb{Z}/5\mathbb{Z}$, and translation by $2/3$ approximates the shift -2. The following play realizes the three-high-failure trace in Figure 1; it is not optimal for the N_{o} diagnostic.

- Round 0: E sends 0, so $r_0 = 1$. T absorbs it.

- Round 1: E sends $1/5$, so $r_1 = 1$. T translates $\{0,\ 1/5\}$ by $2/3$, producing $2/3$ and $13/15$, which ε-cover $3/5$ and $4/5$ respectively.
- Round 2: the only remaining uncovered E-orbit point is $2/5$. E sends $2/5$, so $r_2 = 1$. T absorbs it.

At the start of round 3 all five E-orbit points are covered, so $r_3 = 0$. This play therefore has three high-failure rounds: it is the batch-augmentation intermediate regime, not the fully independent doubling regime. For first neutralization under optimal play, however, T can choose the $1/3$ shift after round 0, covering $2/5$ at the beginning of round 2; this agrees with Theorem B.11, which gives $N_{\mathrm{o}} = 2$ in this parameter choice.

Lemma B.2 (Shift-set membership)**.** *For every $m \in \mathbb{Z}/p\,\mathbb{Z}$,*

$$f(m) \;=\; \frac{\|mq/p\|}{q} \;=\; \frac{|mq|_p}{p\,q},$$

where $|t|_p := \min(t \bmod p,\ p - (t \bmod p))$.

Proof. Write $mq/p = a + r$ with $a \in \mathbb{Z}$ and $r \in [-1/2,\ 1/2]$. The closest point of $\frac{1}{q}\,\mathbb{Z}$ to m/p is at distance $|r|/q = \|mq/p\|/q$, which proves the first equality. The second is by definition of $|\cdot|_p$. □

Lemma B.3 (Critical positive grid distance)**.** *The minimum positive distance between an E-orbit point and a trainer translate of an E-orbit point is*

$$\varepsilon^\star \;=\; \min\left\{ d\Big(\frac{k}{p},\ \frac{k'}{p} + \frac{j}{q}\Big) : \begin{array}{c} k,\,k' \in \{0,\dots,p-1\},\ j \in \{0,\dots,q-1\}, \\ d\Big(\frac{k}{p},\ \frac{k'}{p} + \frac{j}{q}\Big) > 0 \end{array} \right\} \;=\; \frac{1}{L} \;=\; \frac{\gcd(p,q)}{pq}.$$

Proof. The subgroup of $\mathbb{T}^1$ generated by $1/p$ and $1/q$ is the cyclic grid

$$\Big\langle \tfrac{1}{p},\ \tfrac{1}{q} \Big\rangle \;=\; \big\langle \tfrac{1}{L} \big\rangle, \qquad L = \operatorname{lcm}(p,q).$$

Indeed, $1/p$ and $1/q$ both lie in $\langle 1/L \rangle$ since L is a common multiple, and conversely Bezout's identity gives integers $u,\ v$ with $up + vq = g$, hence $u/q + v/p = g/(pq) = 1/L$.

Every difference of the form $k/p - k'/p - j/q$ therefore lies on the $1/L$-grid, so each nonzero such distance is at least $1/L$. To see that $1/L$ is realized, take the Bezout identity above: choosing $k - k' \equiv v \pmod p$ and $j \equiv -u \pmod q$ produces a difference equal to $1/L$ on the circle. This also covers the trivial-trainer case $q = 1$: then $j = 0$, $L = p$, and the distance $1/p = 1/L$ is realized by consecutive E-orbit points. In every case, $\varepsilon^\star = 1/L$. □

Lemma B.4 (Trainer dataset form)**.** *Let $\bullet \in \{\mathrm{single}, \mathrm{batch}, \mathrm{full}\}$. Under any $\bullet$ strategy and the cyclic-walk evaluator, every $d \in D_n^{\mathrm{train}}$ has the form $d = k/p + j/q \pmod 1$ for some $k \in \{0,\dots,n-1\}$ and $j \in \mathbb{Z}/q\,\mathbb{Z}$. Equivalently,*

$$D_n^{\mathrm{train}} \;\subseteq\; \big\{\, k/p + j/q \pmod 1 \;:\; 0 \le k \le n-1,\ j \in \mathbb{Z}/q\,\mathbb{Z} \,\big\}.$$

Proof. Induction on n. The base case $D_0^{\mathrm{train}} = \emptyset$ is trivial.

For the inductive step, each of the three update rules produces D_{n+1}^{train} as a subset of $H_{\mathrm{train}} \cdot (D_n^{\mathrm{train}} \cup E_n)$:

- single: $D_{n+1}^{\mathrm{train}} = D_n^{\mathrm{train}} \cup E_n \cup \{h_n \cdot c_n\}$ with $c_n \in D_n^{\mathrm{train}} \cup E_n$ and $h_n \in H_{\mathrm{train}}$;
- batch: $D_{n+1}^{\mathrm{train}} = D_n^{\mathrm{train}} \cup E_n \cup h_n \cdot C_n$ with $C_n \subseteq D_n^{\mathrm{train}} \cup E_n$;
- full: $D_{n+1}^{\mathrm{train}} = H_{\mathrm{train}} \cdot (D_n^{\mathrm{train}} \cup E_n)$.

By the induction hypothesis, every element of D_n^{train} has the form $k/p + j/q$ with $0 \le k \le n-1$; the new query $E_n = \{n/p\}$ has the form $n/p + 0/q$ with $k = n$, so every element of $D_n^{\mathrm{train}} \cup E_n$ has the form $k/p + j/q$ with $0 \le k \le n$. Translating by any $j_0/q \in H_{\mathrm{train}}$ preserves the form: it sends $k/p + j/q$ to $k/p + (j + j_0)/q$ with k unchanged. Hence every element of D_{n+1}^{train} has the form $k/p + j/q$ with $0 \le k \le n$, as required. □

Lemma B.5 (single dataset capacity)**.** *Under any* single *strategy and the cyclic-walk evaluator,*

$$|D_n^{\text{train}}| \;\le\; 2n, \qquad \textit{hence} \quad |D_n^{\text{train}} \cap \Omega_E| \;\le\; 2n.$$

Proof. The single update rule $D_{n+1}^{\text{train}} = D_n^{\text{train}} \cup E_n \cup \{h_n \cdot c_n\}$ adds at most two new points per round, so $|D_{n+1}^{\text{train}}| \le |D_n^{\text{train}}| + 2$. Induction from $|D_0^{\text{train}}| = 0$ gives $|D_n^{\text{train}}| \le 2n$. □

Common subgroup and cosets. We identify Ω_E with $\mathbb{Z}/p\mathbb{Z}$ via $k/p \mapsto k$. The intersection

$$H_g \;:=\; H_{\text{train}} \cap G_{\text{eval}} \;=\; \langle 1/g \rangle$$

acts on Ω_E as the subgroup $s\,\mathbb{Z}/p\mathbb{Z} \le \mathbb{Z}/p\mathbb{Z}$, with orbits the s cosets

$$C_i \;=\; i + s\,\mathbb{Z}/p\mathbb{Z} \;=\; \{\, i,\, i+s,\, \ldots,\, i+(g-1)s \,\}, \qquad i = 0, \ldots, s-1.$$

For $g = 1$, $H_g = \{0\}$ and each coset $C_i = \{i\}$ is a single orbit point.

Lemma B.6 (H_g acts regularly on each coset)**.** *For each $i \in \{0, \ldots, s-1\}$, the action of H_g on C_i is regular: for any $c, x \in C_i$ there is a unique $h \in H_g$ with $h \cdot c = x$.*

Proof. $|H_g| = |C_i| = g$ and the action is free: if $h \cdot c = c$ then $h \in \text{Stab}(c) = \{0\}$, since H_g acts on $\mathbb{Z}/p\mathbb{Z}$ by translation by multiples of s and only 0 fixes a point. A free action of a group on a set of equal cardinality is regular. □

Lemma B.7 (Sub-critical orbit-trace)**.** *Assume $\varepsilon \le \varepsilon^\star = 1/L$. Let $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$ and let D_n^{train} be the dataset under any $\bullet$ strategy with the cyclic-walk evaluator. If $d \in D_n^{\text{train}}$ ε-covers an orbit point $\ell/p \in \Omega_E$, then $d = \ell/p$; writing $d = k/p + j/q$ as in Lemma B.4, we further have $j/q \in H_g$ and $\ell \equiv k \pmod s$. In particular, every covered orbit point produced from absorbed seed k/p lies in the same H_g-coset $C_{k \bmod s}$, and off-orbit trainer points cover no point of Ω_E.*

Proof. By Lemma B.4, $d = k/p + j/q$ with $0 \le k \le n-1$ and $j \in \mathbb{Z}/q\mathbb{Z}$. The covering condition is $\|\ell/p - k/p - j/q\| < \varepsilon \le \varepsilon^\star$. By Lemma B.3, the only such difference strictly below $\varepsilon^\star$ is zero on the circle, so $\ell/p = k/p + j/q$. Hence $j/q = (\ell - k)/p \in H_{\text{train}} \cap G_{\text{eval}} = H_g$, and $\ell - k \in s\,\mathbb{Z}$. The off-orbit assertion is the contrapositive of the first line: if $d \notin \Omega_E$, then d cannot ε-cover any orbit point. □

Lemma B.8 (Coset-seeding lower bound, sub-critical)**.** *Assume $\varepsilon \le \varepsilon^\star$ and $g \ge 1$. For every $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$,*

$$N_{\text{orbit}}^{\bullet}(\varepsilon, p, q) \;\ge\; s \;=\; p/g.$$

Proof. By Lemma B.7, an orbit point of coset C_i can be covered only by a trainer datum whose absorbed seed k/p has $k \equiv i \pmod s$. The cyclic walk supplies $E_n = \{n/p\}$, so coset C_i has no in-coset seed before round $n = i$. Coset C_{s-1} thus first has a seed at round $s-1$, hence full orbit coverage requires $n \ge s$. □

Lemma B.9 (Sub-critical shift set)**.** *For $\varepsilon \le \varepsilon^\star = 1/L$,*

$$S(\varepsilon) \;=\; \{\, m \in \mathbb{Z}/p\mathbb{Z} : mq \equiv 0 \pmod p \,\} \;=\; (p/g)\;\mathbb{Z}/p\mathbb{Z},$$

a subgroup of order g with elements equally spaced by $s = p/g$ on $\mathbb{Z}/p\mathbb{Z}$.

Proof. By Lemma B.2, $f(m) = |mq|_p/(pq)$. Since $g \mid q$, $mq \in g\,\mathbb{Z}$, hence $|mq|_p \in g\,\mathbb{Z}$, so $|mq|_p \ge g$ whenever $mq \not\equiv 0 \pmod p$. Therefore

$$f(m) \;\in\; \{0\} \cup [g/(pq),\, 1/(2q)] \;=\; \{0\} \cup [1/L,\, 1/(2q)],$$

and $f(m) < \varepsilon \le 1/L$ forces $f(m) = 0$, i.e., $mq \equiv 0 \pmod p$. The latter is equivalent to $m \in (p/g)\,\mathbb{Z}/p\mathbb{Z}$ since $(p/g)\,\mathbb{Z}/p\mathbb{Z}$ is the kernel of multiplication by q on $\mathbb{Z}/p\mathbb{Z}$. □

Lemma B.10 (Saturation threshold and shift set)**.** *The saturation threshold satisfies*

$$\varepsilon_{\text{sat}} \;:=\; \max_{m \in \mathbb{Z}/p\mathbb{Z}} f(m) \;=\; \frac{\lfloor s/2 \rfloor \cdot g}{pq} \;=\; \frac{\lfloor s/2 \rfloor}{L}.$$

For $\varepsilon > \varepsilon_{\text{sat}}$, $S(\varepsilon) = \mathbb{Z}/p\mathbb{Z}$; equivalently, $\Omega_E \subseteq V_\varepsilon(H_{\text{train}})$.

Proof. By Lemma B.2, $f(m) = |mq|_p/(pq)$. Since $g \mid q$, the value $mq \bmod p$ lies in $g\,\mathbb{Z}/p\,\mathbb{Z}$ and ranges over $\{0, g, 2g, \ldots, (s-1)g\}$ as m varies (here $s = p/g$). Hence $|mq|_p \in g\cdot\{0, 1, \ldots, \lfloor s/2 \rfloor\}$, with maximum $g \cdot \lfloor s/2 \rfloor$. Therefore $\varepsilon_{\text{sat}} = \max_m f(m) = g\lfloor s/2 \rfloor/(pq) = \lfloor s/2 \rfloor/L$. The implication $\varepsilon > \varepsilon_{\text{sat}} \Rightarrow S(\varepsilon) = \mathbb{Z}/p\,\mathbb{Z}$ is by definition of the maximum. The equivalence $S(\varepsilon) = \mathbb{Z}/p\,\mathbb{Z} \iff \Omega_E \subseteq V_\varepsilon(H_{\text{train}})$ unwinds the definition of f: $f(m) < \varepsilon$ for every m iff every orbit point m/p is within ε of some $j/q \in H_{\text{train}}$. □

B.1.2 THE CYCLIC-WALK DIAGNOSTIC

Theorem B.11 (N_{o} master theorem)**.** *Let* $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$. *For all* $p \geq 2$, $q \geq 1$, *and* $\varepsilon > 0$, *under the cyclic-walk evaluator,*

$$N_{\text{o}}^{\bullet}(\varepsilon, p, q) \;=\; s^\star(\varepsilon, p, q) \;:=\; \min\big(\,\{\, m \in \{1, \ldots, p-1\} : f(m) < \varepsilon \,\} \cup \{p\}\,\big).$$

(The convention $\cup\{p\}$ *ensures* $s^\star = p$ *when no* $m \in \{1, \ldots, p-1\}$ *satisfies* $f(m) < \varepsilon$.*)*

Proof. We prove matching upper and lower bounds.

Upper bound: $N_{\text{o}}^{\text{single}} \leq s^\star$**.** If $s^\star = p$, then after the first p rounds the trainer has absorbed every point of the cyclic E-orbit, so the query at round p is covered. Assume $s^\star < p$ and pick $j^\star \in \{0, \ldots, q-1\}$ achieving $\|j^\star/q - s^\star/p\| = f(s^\star) < \varepsilon$. Define a single strategy: in round 0, absorb $E_0 = \{0\}$ and choose $h_0 = j^\star/q$, $c_0 = 0$, yielding $D_1^{\text{train}} = \{0,\, j^\star/q\}$. In rounds $1 \leq n < s^\star$, absorb E_n and choose $h_n = 0$ (no-op), so $j^\star/q \in D_n^{\text{train}}$ for all $1 \leq n \leq s^\star$. At round $s^\star$, the query $E_{s^\star} = \{s^\star/p\}$ satisfies $\|s^\star/p - j^\star/q\| = f(s^\star) < \varepsilon$, hence $r_{s^\star} = 0$ and $N_{\text{o}}^{\text{single}} \leq s^\star$.

Any single strategy embeds as a degenerate batch or full strategy: in each round, augment with the same chosen element $h_n \in H_{\text{train}}$ (under batch with one shift) or with all of H_{train} (under full). Both yield a dataset D_n^{train} containing the single dataset, hence with no smaller V_ε-cover. Therefore the same upper bound holds for batch and full.

Lower bound: $N_{\text{o}}^{\bullet} \geq s^\star$ **for** $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$**.** For $r_n = 0$ we need $n/p \in V_\varepsilon(D_n^{\text{train}})$, i.e., some $d \in D_n^{\text{train}}$ with $\|n/p - d\| < \varepsilon$. By Lemma B.4, $d = k/p + j/q$ with $0 \leq k \leq n-1$ and $j \in \mathbb{Z}/q\,\mathbb{Z}$. Setting $m := n - k$, we obtain $m \in \{1, \ldots, n\}$, and the covering condition reads $\|m/p - j/q\| < \varepsilon$, which by Lemma B.2 is exactly $m \in S(\varepsilon)$. The smallest such $m \in \{1, \ldots, p-1\}$ (or p if none exists) is $s^\star$, hence $n \geq s^\star$ as soon as $r_n = 0$.

Combining both bounds: $N_{\text{o}}^{\bullet} = s^\star$ for $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$. □

Remark B.12. The collapse $N_{\text{o}}^{\text{single}} = N_{\text{o}}^{\text{batch}} = N_{\text{o}}^{\text{full}}$ reflects that the cyclic-walk evaluator only requires *one* point to be ε-covered per round, and the trainer's full schedule is offline-predictable. The minimum-budget move single already places a single well-positioned anchor in round 0 that suffices through round $s^\star$. The diagnostic is therefore insensitive to the trainer's per-round batching budget; sensitivity to that axis appears under N_{orbit}.

B.1.3 ORBIT COVERING UNDER FULL

Lemma B.13 (full dataset)**.** *Under* full *and the cyclic-walk evaluator, for* $1 \leq n \leq p$,

$$D_n^{\text{train}} \;=\; \big\{\, k/p + j/q \pmod 1 \;:\; 0 \leq k \leq n-1,\; 0 \leq j < q \,\big\}.$$

Proof. $D_1^{\text{train}} = H_{\text{train}} \cdot \{0\} = \{j/q\}_{j=0}^{q-1}$. Inductively, $D_{n+1}^{\text{train}} = H_{\text{train}} \cdot (D_n^{\text{train}} \cup \{n/p\}) = D_n^{\text{train}} \cup H_{\text{train}} \cdot \{n/p\}$ (using H_{train}-invariance of D_n^{train}), which is the displayed union. □

Theorem B.14 (full orbit covering time)**.** *For all* $p \geq 2$, $q \geq 1$, *and* $\varepsilon > 0$,

$$N_{\text{orbit}}^{\text{full}}(\varepsilon, p, q) \;=\; \Gamma\big(S(\varepsilon)\big),$$

where $\Gamma(S)$ *is the maximum cyclic gap of* S *on* $\mathbb{Z}/p\,\mathbb{Z}$*: sorting* $S = \{s_0 < s_1 < \cdots < s_{|S|-1}\}$ *in* $\{0, \ldots, p-1\}$ *and setting* $s_{|S|} := s_0 + p$,

$$\Gamma(S) := \max_{0 \leq i < |S|} (s_{i+1} - s_i).$$

Proof. First note that, at $n = p$, we have $\Omega_E \subseteq D_p^{train}$ by "pure memorization", so it suffices to consider the case $1 \le n \le p$. Then, by Lemma B.13, the round-n trainer data is $D_n^{\text{train}} = K_n/p + (1/q)\,\mathbb{Z}/q\,\mathbb{Z}$, where $K_n := \{0, 1, \ldots, n-1\} \subseteq \mathbb{Z}/p\,\mathbb{Z}$. Hence for $\ell \in \mathbb{Z}/p\,\mathbb{Z}$,

$$\ell/p \in V_\varepsilon\big(D_n^{\text{train}}\big) \iff \exists\, k \in K_n,\ j \in \{0, \ldots, q-1\} :\ \|\ell/p - k/p - j/q\| < \varepsilon,$$

which by Lemma B.2 is equivalent to $\ell - k \in S(\varepsilon) \pmod p$ for some $k \in K_n$, i.e., $\ell \in K_n + S(\varepsilon) \pmod p$. Therefore

$$N_{\text{orbit}}^{\text{full}} \;=\; \min\big\{\, n \ge 1 : K_n + S(\varepsilon) = \mathbb{Z}/p\,\mathbb{Z} \,\big\}.$$

Fix $\ell \in \mathbb{Z}/p\,\mathbb{Z}$. The condition $\ell \in K_n + S(\varepsilon)$ unfolds as

$$\exists\, k \in \{0, \ldots, n-1\},\ s \in S(\varepsilon) :\ \ell - s \equiv k \pmod p \iff \exists\, s \in S(\varepsilon) :\ (\ell - s) \bmod p \;\le\; n-1.$$

Thus the smallest n for which ℓ is covered equals

$$n^\star(\ell) \;:=\; 1 + \min_{s \in S(\varepsilon)} \big((\ell - s) \bmod p\big).$$

Geometrically, $(\ell - s) \bmod p$ is the cyclic clockwise distance from s to ℓ on $\mathbb{Z}/p\,\mathbb{Z}$, so $n^\star(\ell) - 1$ is the cyclic distance from ℓ *back* to the nearest preceding element of $S(\varepsilon)$. Taking the maximum over ℓ,

$$N_{\text{orbit}}^{\text{full}} \;=\; \max_{\ell \in \mathbb{Z}/p\mathbb{Z}} n^\star(\ell) \;=\; 1 + \max_{\ell} \min_{s \in S(\varepsilon)} \big((\ell - s) \bmod p\big).$$

The inner max-min is achieved when ℓ sits at the far end of the largest cyclic gap of $S(\varepsilon)$: with $S = \{s_0 < \cdots < s_{|S|-1}\}$ and $s_{|S|} := s_0 + p$, choosing $\ell = s_{i+1} - 1$ (the predecessor of s_{i+1}) gives $\min_s (\ell - s) \bmod p = s_{i+1} - 1 - s_i = g_i - 1$, where $g_i := s_{i+1} - s_i$. Hence the maximum equals $\Gamma(S(\varepsilon)) - 1$, and $N_{\text{orbit}}^{\text{full}} = \Gamma(S(\varepsilon))$. □

Corollary B.15 (Sub-critical, coprime: full and N_{o})**.** *For $\varepsilon \le \varepsilon^\star$ and $g = 1$, $S(\varepsilon) = \{0\}$, hence $\Gamma(S(\varepsilon)) = p$, $s^\star = p$, and $N_{\text{orbit}}^{\text{full}} = N_{\text{o}}^\bullet = p$ for $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$.*

Corollary B.16 (Sub-critical, $g \ge 2$: full and N_{o})**.** *For $\varepsilon \le \varepsilon^\star$ and $g \ge 2$, $S(\varepsilon) = (p/g)\,\mathbb{Z}/p\,\mathbb{Z}$ has g equally-spaced elements, hence $\Gamma(S(\varepsilon)) = p/g$, $s^\star = p/g$, and $N_{\text{orbit}}^{\text{full}} = N_{\text{o}}^\bullet = p/g$ for $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$.*

Corollary B.17 (Saturation: full and N_{o})**.** *For $\varepsilon > \varepsilon_{\text{sat}}$, $S(\varepsilon) = \mathbb{Z}/p\mathbb{Z}$, hence $\Gamma(S(\varepsilon)) = 1$, $s^\star = 1$, and $N_{\text{orbit}}^{\text{full}} = N_{\text{o}}^\bullet = 1$ for $\bullet \in \{\text{single}, \text{batch}, \text{full}\}$. The conclusion covers the large-radius saturation cases $\varepsilon > 1/2$ (for example $q = 1$ with even p, where $\varepsilon_{\text{sat}} = 1/2$).*

Proof of Corollaries B.15–B.17. Combine Theorem B.14 ($N_{\text{orbit}}^{\text{full}} = \Gamma(S(\varepsilon))$) and Theorem B.11 ($N_{\text{o}}^\bullet = s^\star$, the smallest positive element of $S(\varepsilon)$, with the convention $s^\star = p$ if $S(\varepsilon) \cap \{1, \ldots, p-1\} = \emptyset$) with Lemma B.9 for the sub-critical case (which specializes to $\{0\}$ at $g = 1$, giving $\Gamma = p$ and $s^\star = p$, and to $(p/g)\,\mathbb{Z}/p\,\mathbb{Z}$ at $g \ge 2$, giving $\Gamma = p/g$ and $s^\star = p/g$), and Lemma B.10 for the saturation case. The lemmas impose no upper bound on ε, hence cover all large-radius cases including $\varepsilon > 1/2$. □

B.1.4 Orbit covering under batch

We continue to identify Ω_E with $\mathbb{Z}/p\,\mathbb{Z}$ via $k/p \mapsto k$, with cosets $C_i = i + s\,\mathbb{Z}/p\,\mathbb{Z}$ of H_g.

Lemma B.18 (Dyadic subset-sum cover)**.** *Let $K \ge 1$, let $H = \langle 1/m \rangle \le \mathbb{T}^1$ be a cyclic subgroup of order m contained in H_{train}, and write $S_0 := D_{n_0}^{\text{train}} \cup E_{n_0}$. Suppose in K consecutive rounds $n_0, n_0{+}1, \ldots, n_0{+}K{-}1$ the trainer plays* batch *with $C_{n_0+k} = D_{n_0+k}^{\text{train}} \cup E_{n_0+k}$ and $h_{n_0+k} = 2^k/m$. Then*

$$D_{n_0+K}^{\text{train}} \;\supseteq\; S_0 \;+\; \Big\{\, \tfrac{1}{m}\sum_{k \in S} 2^k \pmod 1 \;:\; S \subseteq \{0, \ldots, K-1\} \,\Big\}.$$

If $K \ge \lceil \log_2 m \rceil$, the right-hand subset-sum set equals H, so $D_{n_0+K}^{\text{train}}$ contains every H-translate of every point of S_0.

Proof. Let T_k denote the set of dyadic subset sums $\{(1/m)\sum_{j\in S} 2^j \bmod 1 : S \subseteq \{0,\dots,k-1\}\}$, so $T_0 = \{0\}$ and $T_{k+1} = T_k \cup (T_k + 2^k/m)$. We show by induction on $k \geq 1$ that $S_0 + T_k \subseteq D^{\text{train}}_{n_0+k}$.

Base case ($k = 1$). The round-n_0 batch update absorbs E_{n_0} unshifted and additionally applies $h_{n_0} = 1/m$ to $C_{n_0} \supseteq S_0$, so

$$D^{\text{train}}_{n_0+1} \;\supseteq\; D^{\text{train}}_{n_0} \cup E_{n_0} \cup h_{n_0} \cdot C_{n_0} \;\supseteq\; S_0 \cup (S_0 + 1/m) \;=\; S_0 + T_1.$$

Inductive step ($k \to k+1$, $k \geq 1$). Assume $S_0 + T_k \subseteq D^{\text{train}}_{n_0+k}$. Since $D^{\text{train}}_{n_0+k} \subseteq C_{n_0+k}$, the round-$(n_0+k)$ update gives $D^{\text{train}}_{n_0+k+1} \supseteq h_{n_0+k} \cdot C_{n_0+k} \supseteq h_{n_0+k} \cdot (S_0 + T_k) = S_0 + T_k + 2^k/m$. Combined with $D^{\text{train}}_{n_0+k+1} \supseteq D^{\text{train}}_{n_0+k} \supseteq S_0 + T_k$, this yields $D^{\text{train}}_{n_0+k+1} \supseteq S_0 + (T_k \cup (T_k + 2^k/m)) = S_0 + T_{k+1}$.

After $K \geq 1$ rounds, $S_0 + T_K \subseteq D^{\text{train}}_{n_0+K}$. If $K \geq \lceil \log_2 m \rceil$, the integer subset sums $\sum_{k\in S} 2^k$ exhaust $\{0, 1, \dots, 2^K - 1\}$, and $2^K \geq m$, so their reductions modulo m cover all of $\{0, 1, \dots, m-1\}$. Hence $T_K = H$. □

Theorem B.19 (batch sub-critical bound, $g \geq 2$)**.** *Assume $g \geq 2$ and let $\varepsilon \leq \varepsilon^\star = g/(pq) = 1/L$. Then*

$$N^{\text{batch}}_{\text{orbit}}(\varepsilon, p, q) \;=\; \Theta\big(s + \log g\big) \;=\; \Theta(p/g + \log g).$$

Proof. By Lemma B.7, every covered orbit point stays in the H_g-coset of its absorbed seed, and off-orbit trainer points cover nothing in Ω_E. Combining with $\varepsilon \leq \varepsilon^\star$, two distinct orbit points are not ε-close, so orbit covering is equivalent to $D^{train}_n \cap \Omega_E$-covered seeds $= \Omega_E$.

Lower bound. Lemma B.8 gives $N^{\text{batch}}_{\text{orbit}} \geq s$. For the second bound, fix any coset C_i and define

$$R_i \;:=\; \{\, k/p + j/q \pmod 1 \,:\, k \equiv i \pmod s,\; j \in \mathbb{Z}/q\,\mathbb{Z} \,\} \;=\; C_i + H_{\text{train}},$$

the set of points in $\Omega_E + H_{\text{train}}$ whose p-component lies in C_i. Equivalently $R_i = i/p + H_{\text{train}}$ is an H_{train}-coset of size q, so R_i is setwise preserved by every $h \in H_{\text{train}}$. Set $a^{(i)}_n := |D^{\text{train}}_n \cap R_i|$. Intersecting the batch update with R_i,

$$a^{(i)}_{n+1} \;\leq\; a^{(i)}_n + |E_n \cap R_i| + |h_n \cdot (C_n \cap R_i)| \;\leq\; a^{(i)}_n + 1 + (a^{(i)}_n + 1) \;=\; 2a^{(i)}_n + 2,$$

using $C_n \subseteq D^{\text{train}}_n \cup E_n$ and the h_n-invariance of R_i. Induction from $a^{(i)}_0 = 0$ yields $a^{(i)}_n \leq 2^{n+1} - 2$. Covering C_i requires $|D^{\text{train}}_n \cap C_i| \geq g$, and since $D^{\text{train}}_n \cap C_i \subseteq D^{\text{train}}_n \cap R_i$, we have $a^{(i)}_n \geq g$, hence $N^{\text{batch}}_{\text{orbit}} \geq \log_2(g+2) - 1$. Combining,

$$N^{\text{batch}}_{\text{orbit}} \;\geq\; \max\big(s,\, \log_2(g+2) - 1\big) \;\geq\; \tfrac{1}{4}(s + \log_2 g),$$

using $g \geq 2$ and $s \geq 1$.

Upper bound. Set $K := \lceil \log_2 g \rceil$. During rounds $n = 0, \dots, s-1$, take $C_n = D^{\text{train}}_n \cup E_n$ and $h_n = 0$, so the trainer absorbs the evaluator queries. Then $D^{\text{train}}_s \supseteq \{0, 1, \dots, s-1\}$, one seed in each H_g-coset C_i. For rounds $s, \dots, s+K-1$, take $C_n = D^{\text{train}}_n \cup E_n$ and $h_{s+k} = 2^k/g$, and apply Lemma B.18 with $H = H_g$ (of order g, contained in H_{train} since $g \mid q$): with seed set $S_0 := D^{\text{train}}_s \cup E_s \supseteq \{0, 1, \dots, s\}$, each of the s seeds $0, 1, \dots, s-1$ in S_0 acquires every H_g-translate, so $D^{\text{train}}_{s+K} \supseteq \bigsqcup_{i=0}^{s-1} C_i = \Omega_E$. Hence

$$N^{\text{batch}}_{\text{orbit}} \;\leq\; s + \lceil \log_2 g \rceil.$$

Combining the two bounds gives $N^{\text{batch}}_{\text{orbit}} = \Theta(s + \log g) = \Theta(p/g + \log g)$. □

Corollary B.20 (Explicit upper-bound expression)**.** *Under the hypotheses of Theorem B.19,*

$$N^{\text{batch}}_{\text{orbit}}(\varepsilon, p, q) \;\leq\; s + \lceil \log_2 g \rceil \qquad (\varepsilon \leq \varepsilon^\star,\; g \geq 2),$$

which already saturates the asymptotic claim of Theorem B.19. This is deliberately not an exact formula: the dyadic construction waits until every coset has a seed and can be non-optimal in collapsed cases such as $p = q$. The $\Theta(s + \log g)$ statement is the level needed for Table 1.

Corollary B.21 (Sub-critical coprime, batch)**.** *Assume* $g = 1$ *(equivalently* $\gcd(p, q) = 1$*) and* $\varepsilon \le \varepsilon^\star = 1/(pq)$*. Then* $N^{\mathrm{batch}}_{\mathrm{orbit}}(\varepsilon, p, q) = p$*.*

Proof. By Lemma B.9 at $g = 1$, $S(\varepsilon) = \{0\}$ and the H_g-cosets are the singletons $\{i\}$; so by Lemma B.7 every covered orbit point coincides with an absorbed seed. The cyclic walk forces p rounds before all p seeds have arrived, hence $N^{\mathrm{batch}}_{\mathrm{orbit}} \ge p$. The upper bound p is achieved by absorption alone. □

Theorem B.22 (batch saturation upper bound)**.** *Assume* $q \le p$*. For* $\varepsilon > \varepsilon_{\mathrm{sat}}$*,*

$$N^{\mathrm{batch}}_{\mathrm{orbit}}(\varepsilon, p, q) \;\le\; \max\{1, \lceil \log_2 q \rceil\} \;=\; O(\log p).$$

Proof. If $q = 1$, then $H_{\mathrm{train}} = \{0\}$ and Lemma B.10 gives $\Omega_E \subseteq V_\varepsilon(\{0\})$. After round 0 the trainer has absorbed 0, so $N^{\mathrm{batch}}_{\mathrm{orbit}} \le 1$.

Assume $q \ge 2$, set $K := \lceil \log_2 q \rceil$, and apply Lemma B.18 with $H = H_{\mathrm{train}}$ (order q) in rounds $0, 1, \ldots, K-1$ starting from $D^{\mathrm{train}}_0 \cup E_0 = \{0\}$. The conclusion gives $D^{\mathrm{train}}_K \supseteq H_{\mathrm{train}}$. By Lemma B.10, $\Omega_E \subseteq V_\varepsilon(H_{\mathrm{train}}) \subseteq V_\varepsilon(D^{\mathrm{train}}_K)$, hence $N^{\mathrm{batch}}_{\mathrm{orbit}} \le K \le \lceil \log_2 p \rceil$. □

Remark B.23 (Why no error compounding)**.** A naive triangle-inequality bookkeeping of approximate H-translations in the saturation regime would suggest an error blow-up of size $K\varepsilon$ after K rounds, restricting the doubling argument to a slack window $\varepsilon - \varepsilon_{\mathrm{sat}}$. The proof above sidesteps this entirely by exploiting that H_{train} is a *group*: any chain of H-shifts applied to a fixed seed E_0 collapses to a single H-translate of E_0, so the dataset point covering an orbit point ℓ/p has the form $0 + h$ with $h \in H_{\mathrm{train}}$, and only the single inequality $\|\ell/p - h\| < \varepsilon$ (i.e., $\ell \in S(\varepsilon)$) is needed.

B.1.5 Orbit covering under single

Corollary B.24 (Sub-critical coprime, single)**.** *Assume* $g = 1$ *and* $\varepsilon \le \varepsilon^\star = 1/(pq)$*. Then* $N^{\mathrm{single}}_{\mathrm{orbit}}(\varepsilon, p, q) = p$*.*

Proof. At $g = 1$, $H_g = \{0\}$, so by Lemma B.7 no augmentation produces a new E-orbit point and absorption alone gives $|D^{\mathrm{train}}_n \cap \Omega_E| = n$. Coverage requires $n \ge p$. To conclude, note that $n = p$ suffices. □

Theorem B.25 (single sub-critical formula, $g \ge 2$)**.** *Assume* $g \ge 2$ *and* $\varepsilon \le \varepsilon^\star = g/(pq)$*. Then*

$$N^{\mathrm{single}}_{\mathrm{orbit}}(\varepsilon, p, q) \;=\; \lceil p/2 \rceil.$$

Proof. **Lower bound.** *Capacity.* By Lemma B.5, $|D^{\mathrm{train}}_n| \le 2n$. By Lemma B.7, each datum covers at most one orbit point, so coverage requires $2n \ge p$, i.e., $n \ge \lceil p/2 \rceil$.

Upper bound. We construct an explicit single strategy covering Ω_E in $V := \lceil p/2 \rceil$ rounds. Note $V \ge s$ since $g \ge 2$ implies $p/2 \ge p/g$.

For $i \in \{0, \ldots, s-1\}$, let

$$a_i \;:=\; |\{n \in [0, V) : n \equiv i \pmod s\}| \;=\; \lceil (V - i)/s \rceil$$

be the number of E-arrivals in coset C_i during the first V rounds. The auto-arrivals fill the a_i smallest representatives of C_i, namely $i,\ i+s,\ \ldots,\ i+(a_i-1)s$. The set of *missing* elements of C_i is therefore

$$M_i \;=\; \{\, i + a_i\, s,\ i + (a_i{+}1)s,\ \ldots,\ i + (g{-}1)s \,\}, \qquad |M_i| = g - a_i \;=:\; \beta_i.$$

A double count gives $\sum_i a_i = V$ and $\sum_i \beta_i = sg - V = p - \lceil p/2 \rceil = \lfloor p/2 \rfloor \le V$.

We schedule the missing points explicitly. Put

$$B_i \;:=\; \sum_{r<i} \beta_r.$$

For each $m = i + (a_i + u)s \in M_i$ with $0 \le u < \beta_i$, schedule m in round

$$n(m) \;:=\; B_i + u.$$

The assigned rounds are distinct, since the intervals $[B_i,\, B_i + \beta_i)$ are consecutive, and all lie in $\{0, \dots, V-1\}$ because their total number is $\sum_i \beta_i = p - V = \lfloor p/2 \rfloor \le V$.

It remains to check that a seed from C_i is available by round $n(m)$. For every i one has $a_i \le \lceil V/s \rceil \le g - 1$: if $g = 2$ then $V = s$, while if $g \ge 3$ then $V = \lceil sg/2 \rceil \le s(g-1)$ because $s(g-2) \ge 1$. Hence

$$B_i \;=\; \sum_{r<i} (g - a_r) \;=\; ig - \sum_{r<i} a_r \;\ge\; ig - i(g-1) \;=\; i.$$

Thus $n(m) = B_i + u \ge i$. At round $n(m)$, the point $i \in C_i$ is therefore available: if $n(m) = i$ it is the current query in E_i, and if $n(m) > i$ it has already been absorbed into $D^{\mathrm{train}}_{n(m)}$. Since $m \in C_i$, Lemma B.6 gives a unique element $h \in H_g \subseteq H_{\mathrm{train}}$ sending i to m; play this single augmentation in round $n(m)$. In rounds not assigned to any missing point, play the identity shift on the current query.

After V rounds, D^{train}_V contains the auto-arrivals $\{0, 1, \dots, V-1\}$ together with the scheduled targets $\bigsqcup_i M_i = \{V, V+1, \dots, p-1\}$. Hence $\Omega_E \subseteq D^{\mathrm{train}}_V \subseteq V_\varepsilon(D^{\mathrm{train}}_V)$, since $\varepsilon > 0$, so $N^{\mathrm{single}}_{\mathrm{orbit}} \le V = \lceil p/2 \rceil$. □

Lemma B.26 (Arc packing on Ω_E)**.** *Every open circle ball $B_{\mathbb{T}}(z, \varepsilon)$ contains at most $2p\varepsilon + 1$ points of Ω_E. Consequently,*

$$Q(\varepsilon, p, q) \;\le\; 2p\varepsilon + 1.$$

Proof. If $\varepsilon \ge 1/2$, the bound is trivial because $|\Omega_E| = p \le 2p\varepsilon + 1$. Assume $\varepsilon < 1/2$. $\Omega_E = \{0, 1/p, \dots, (p-1)/p\}$ is equally spaced with gap $1/p$, so an open arc of length 2ε contains at most $\lfloor 2\varepsilon p \rfloor + 1 \le 2p\varepsilon + 1$ points. Each open ball $B_{\mathbb{T}}(z, \varepsilon)$ is such an arc, hence $|\Omega_E \cap B_{\mathbb{T}}(z, \varepsilon)| \le 2p\varepsilon + 1$. □

Lemma B.27 (Sparse H-cover in saturation)**.** *Assume $\varepsilon > \varepsilon_{\mathrm{sat}}$, and let $Q = Q(\varepsilon, p, q)$. There is a subset $A \subseteq H_{\mathrm{train}}$ with $|A| \le 10\, p/Q$ such that $\Omega_E \subseteq V_\varepsilon(A)$.*

Proof. By Lemma B.10, $\Omega_E \subseteq V_\varepsilon(H_{\mathrm{train}})$. If $Q = 1$, for each $m/p \in \Omega_E$ choose one H-translate that covers it; this gives an H-cover of size at most $p = p/Q$.

Assume $Q \ge 2$. By Lemma B.26, $2p\varepsilon \ge 1$, hence $Q \le 4p\varepsilon$.

If $\varepsilon q < 2$, take $A = H_{\mathrm{train}}$: it covers Ω_E by saturation, and $|A| = q < 2/\varepsilon$. Otherwise $\varepsilon q \ge 2$, and we choose points of the regular q-grid $H_{\mathrm{train}} = \{0, 1/q, \dots, (q-1)/q\}$ at spacing at most ε in cyclic order: set $r := \lfloor \varepsilon q \rfloor$ and take $0, r/q, 2r/q, \dots, \lfloor (q-1)/r \rfloor \cdot r/q$. Consecutive chosen H-points, including the wrap-around gap, have circular gap at most $r/q \le \varepsilon$, so their ε-balls cover the whole circle. The number chosen is at most $2/\varepsilon + 2$.

In both cases $|A| \le 2/\varepsilon + 2$. Since $Q \ge 2$ and $Q \le p$, the inequalities $Q \le 4p\varepsilon$ and $2 \le 2p/Q$ give

$$|A| \;\le\; 2/\varepsilon + 2 \;\le\; 8\, p/Q + 2\, p/Q \;=\; 10\, p/Q.$$

The argument uses no upper bound on ε, so it covers all saturation radii including $\varepsilon > 1/2$ (for example $q = 1$ with even p, where $\varepsilon_{\mathrm{sat}} = 1/2$). □

Corollary B.28 (single saturation)**.** *For $\varepsilon > \varepsilon_{\mathrm{sat}}$,*

$$\lceil p/(2Q) \rceil \;\le\; N^{\mathrm{single}}_{\mathrm{orbit}}(\varepsilon, p, q) \;\le\; 10\, p/Q,$$

where $Q = Q(\varepsilon, p, q)$ is the reachable-center packing factor. In particular $N^{\mathrm{single}}_{\mathrm{orbit}} = \Theta(p/Q)$.

Proof. For the lower bound, Lemma B.5 gives $|D^{\mathrm{train}}_n| \le 2n$ under single, and Lemma B.4 gives that every point of D^{train}_n lies in $\Omega_E + H_{\mathrm{train}}$. Hence each such point covers at most Q points of Ω_E, by definition of Q. Thus full orbit coverage forces $2nQ \ge p$, and hence $n \ge \lceil p/(2Q) \rceil$.

For the upper bound, choose $A = \{a_0, \ldots, a_{K-1}\} \subseteq H_{\text{train}}$ as in Lemma B.27, so $K \le 10p/Q$ and $\Omega_E \subseteq V_\varepsilon(A)$. In each of the first K rounds, use the seed $0 \in E_0$ (which is available at round 0 and then remains in the trainer dataset) and play the single augmentation $h_n = a_n$, $c_n = 0$. After K rounds the trainer dataset contains all points of A, hence its ε-neighborhood covers Ω_E. Thus $N_{\text{orbit}}^{\text{single}} \le K \le 10p/Q$. □

Remark B.29 (On constants in the $\Theta(p/Q)$ entry of Table 1)**.** The proof above is deliberately coarse: the constant 10 is only used to make the table's asymptotic entry explicit. The lower bound has the unavoidable factor $1/2$ because a single round both absorbs the current E-query and adds one H-translate. No logarithmic set-cover loss is needed in the saturation regime.

B.1.6 Three-tier separation at p = q

Theorem B.30 (Three-tier batching ladder at p = q)**.** *For $p = q \ge 3$ and any $0 < \varepsilon \le \varepsilon^\star$,*

$$N_{\text{orbit}}^{\text{full}}(\varepsilon, p, p) = 1, \qquad N_{\text{orbit}}^{\text{batch}}(\varepsilon, p, p) = \Theta(\log p), \qquad N_{\text{orbit}}^{\text{single}}(\varepsilon, p, p) = \lceil p/2 \rceil.$$

In particular the asymptotic ordering $\Theta(1) \ll \Theta(\log p) \ll \Theta(p)$ is strict.

Proof. At $p = q$ we have $g = q = p$ and $s = p/g = 1$. Substituting into the sub-critical formulas with $g = p$:

- full: by Corollary B.16, $N_{\text{orbit}}^{\text{full}} = p/g = 1$.
- batch: by Theorem B.19 (applicable since $g = p \ge 3 \ge 2$), $N_{\text{orbit}}^{\text{batch}} = \Theta(s + \log g) = \Theta(1 + \log p) = \Theta(\log p)$.
- single: by Theorem B.25 with $g = p \ge 3 \ge 2$, $N_{\text{orbit}}^{\text{single}} = \lceil p/2 \rceil$.

The three values are $\Theta(1)$, $\Theta(\log p)$, and $\Theta(p)$ respectively, hence strictly ordered for $p \ge 3$. □

Remark B.31 (Ladder collapse at fixed g)**.** In the sub-critical regime $\varepsilon \le \varepsilon^\star$, the three-tier ladder collapses entirely when g is fixed (for example $g = 2$). By Corollary B.16, $N_{\text{orbit}}^{\text{full}} = p/g$; by Theorem B.25, $N_{\text{orbit}}^{\text{single}} = \lceil p/2 \rceil$; and by Corollary B.20, $N_{\text{orbit}}^{\text{batch}} \le s + \lceil \log_2 g \rceil$, which combined with the lower bound $N_{\text{orbit}}^{\text{batch}} \ge s$ from Lemma B.8 gives $N_{\text{orbit}}^{\text{batch}} = p/g + O(\log g)$. At $g = 2$ all three are $p/2 + O(1)$, so no asymptotic separation is visible. A strict $\Theta(1) \ll \Theta(\log p) \ll \Theta(p)$ ladder requires $g = \Theta(p)$, for example the sub-family $p = q$ with $p \to \infty$ used in Theorem B.30.

C Licenses

We use the MMLU benchmark, and the weights of mistralai/Mistral-7B-Instruct-v0.3, Qwen/Qwen2.5-7B-Instruct, EleutherAI/pythia-410m, EleutherAI/pythia-1b, allenai/Olmo-3.1-32B-Instruct, under *Apache 2.0* licenses. We use gpt2 / openai-community/gpt2's weight under a *modified MIT license*. We use meta-llama/Llama-3.1-8B-Instruct under *Meta Llama 3.1 Community License*, and allenai/WildJailBreak / WildJailBreak under *ODC-BY*.

D Distance-dependent transfer: detailed methods and supplementary results

D.1 Hardware and compute

Experiments ran on an internal cluster with GPUs providing between 40 GB and 128 GB of HBM. Each cell required a few GPU-hours, comprising a fine-tuning run (≤ 1 hour), a held-out evaluation pass with $N{=}10$ stochastic samples per prompt at temperature 0.7 ($\approx 2{-}3$ hours), and an LLM-as-judge labeling pass ($\approx 1{-}2$ hours).

D.2 MODELS, FINE-TUNING CONFIGURATION, AND CELLS

The run manifest and reconstructed training implementation provided in the supplement specify the following protocol.

Models and cells. The three models are Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and Qwen2.5-7B-Instruct, with weights and native tokenizers from Hugging Face. Each model/rank/exposure configuration defines one of six cells, all with seed 42. The rank-1 cohort contains one cell per family with 1,600 training pair presentations. At rank 2, Llama and Mistral each use 800 presentations, and Qwen uses 1,600. Each cell loads a fresh base model; its trained adapter is saved and reloaded for evaluation.

Training pairs and targets. All six cells use the same 40 training pairs, indexed by `active_training_indices.json` in the supplement. These same 40 prompts define the reference set used for the reported distance minimum. Indices are zero-based positions in the 82,728-row subset of `allenai/wildjailbreak` (configuration `train`) filtered by `data_type = adversarial_harmful`. For each indexed row, the user message is its `adversarial` field and the assistant target is its original `completion` field, both read as strings. Targets are fixed across epochs and model families; no common refusal string is substituted, and no contrastive pairs or preference objective are used. The supplement records the ordered indices and the SHA-256 hash of the source Parquet file in `provenance/training_manifest.json`.

Formatting and tokenization. Each pair is serialized using its model tokenizer's native `apply_chat_template`, with one user message followed by one assistant message and `tokenize=False`. No explicit system message is supplied by the trainer; any template defaults are retained. The serialized text is then tokenized with `add_special_tokens=True`, `truncation=True`, and `max_length=256`; this cap applies to the entire formatted prompt–response sequence. The tokenizer's truncation side is not overridden. Separately, the user-only message is serialized with `add_generation_prompt=True` and tokenized without truncation and with `add_special_tokens=False` to obtain a prefix length p.

Loss and masking. Training uses the causal language model's next-token cross-entropy with the standard one-token label shift. For a tokenized sequence of length L, labels initially copy the input token IDs. With $m = \min(p, L)$, the first m labels are set to the ignore index -100 only if $0 < m < L$; otherwise all non-padding labels remain unmasked. Batches are right-padded, with padding attention entries set to zero and padding labels set to -100; EOS is used as the padding token if no padding token is defined. After the causal shift, cross-entropy is averaged over all non-ignored target tokens in each minibatch, including retained template terminators. Thus the implementation uses prefix-length masking, with full-sequence loss when $p \geq L$ (or $p = 0$). It does not guarantee strictly response-only supervision: the full text and prefix also use different special-token settings, and no token-boundary alignment check is performed.

Optimization. All cells use standard LoRA with $\alpha = r$, dropout 0.10, and target modules `q_proj` and `v_proj`; neither rsLoRA nor DoRA is enabled. Only adapter parameters are optimized, using AdamW at learning rate 3×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01, batch size four, one optimizer step per batch, and no scheduler or gradient clipping. On CUDA, the implementation loads the frozen base weights in bfloat16 and uses PEFT's default promotion of adapter parameters to float32. A NumPy `default_rng(42)` generates a new permutation of the 40 pairs each epoch; Torch and NumPy RNGs are also reset to seed 42 immediately before LoRA construction. The budgets of 800 and 1,600 pair presentations correspond to 20 and 40 epochs, or 200 and 400 optimizer updates, respectively.

Software and reproducibility. The recorded training versions are PyTorch 2.11.0+cu126, Transformers 4.57.6, and PEFT 0.18.1. The supplement does not include the target text or a retrievable, immutable copy of the source dataset snapshot. Immutable base-model and tokenizer revisions, and historical tokenizer and chat-template configurations, are also unavailable. The exact target strings, historical serialized inputs and supervised-token boundaries, and reproduction of the trained checkpoints therefore cannot be verified from the supplied materials alone.

D.3 CANDIDATE-POOL SELECTION AND ACTIVE TRAINING-AND-DISTANCE SUBSET

We construct a shared ordered 47-prompt candidate list $\mathcal{P}_{\text{cand}}$. The 40-prompt active training-and-distance set $\mathcal{P}_{\text{train}} \subset \mathcal{P}_{\text{cand}}$ is fixed by `active_training_indices.json` in the supplement. The candidate list is selected using corpus density and cluster diversity to provide coverage of dense regions of the corpus.

Candidate selection by model and metric. For each target model (LLAMA, MISTRAL, or QWEN) and each embedding metric (`last_token`, `mean_pool`, `mean_pool_first`, `spectral_first`, or `spectral_last`), we:

1. Embed every prompt of the WILDJAILBREAK (Jiang et al., 2024) adversarial-harmful split under the target's base model with the chosen embedding tag (Appendix D.5).
2. Compute the k=50-NN cosine radius $\rho_{50}(p) \in [0, 2]$ of each prompt p in that embedding space.
3. Partition the corpus into K=8 clusters by k-means on ℓ_2-normalized embeddings.
4. For each cluster, select the prompt with the smallest ρ_{50}. This gives 8 representatives for each model and metric.
5. For each target, retain the two metrics with the smallest mean ρ_{50} across their cluster representatives. The smallest mean is obtained by `spectral_first` for all three targets; the second selected metric varies by target.

This yields $3 \times 2 \times 8 = 48$ candidate prompts (with one duplicate across (target, metric) pairs), of which 47 are unique corpus indices. We rank these candidates by a vote score $s(p) = \#\{(\text{target}, \text{metric}) : p \in \text{cluster-reps}\}$ with ties broken by lowest mean ρ_{50}, and retain all 47 in the candidate list $\mathcal{P}_{\text{cand}}$. Qualitative inspection suggests coverage of request categories including cyber-attack assistance, hate speech, privacy intrusion, misinformation, drugs and contraband, explicit sexual content, professional misconduct, and self-harm.

The 47 candidate indices are shared across all six cells. The same 40 active training-and-distance indices are also shared across all cells. The released index lists fix the membership and corpus-index ordering of both sets. The model being fine-tuned, its per-family embeddings, and the LoRA configuration vary across cells as described above.

D.4 EVALUATION PANEL CONSTRUCTION

The held-out evaluation panel $\mathcal{H}$ contains 2999 corpus prompts and is shared across all cells; it excludes all 47 members of $\mathcal{P}_{\text{cand}}$, including the active training subset. The panel is constructed to include prompts both near to and far from the candidate pool across the embedding metrics:

1. **Near-pool stratum** (2116 prompts). For each (target, metric) pair used to construct $\mathcal{P}_{\text{cand}}$ (Appendix D.3), retrieve the 50 nearest neighbors of each of the 8 cluster representatives in that embedding space. Take the union across the $3 \times 2 = 6$ selections and remove duplicates.
2. **Random reference stratum** (384 prompts). Sample uniformly at random from the corpus (excluding $\mathcal{P}_{\text{cand}}$ and the near-pool stratum), seed 42, giving mid- and high-distance reference points.
3. **Per-family low-distance extension stratum** (499 prompts). For each family, greedily add prompts that lie below the family's z-distribution sixth decile under at least one of the family's two pool-selection metrics, until each family has 250 extension prompts; deduplicate across families (499 unique).

The three strata are concatenated to form $\mathcal{H}$ with $|\mathcal{H}| = 2999$. The same 2999 corpus indices are evaluated by every cell.

D.5 EMBEDDING TAGS AND DISTANCE METRIC

For each model M and each held-out prompt p, let $H_M(p) \in \mathbb{R}^{T(p)\times d\times L}$ denote the tensor of M's per-block hidden states given p as input under M's native chat template, where $T(p)$ is the prompt length in tokens, d is the model hidden dimension, and L is the number of transformer blocks. Let $H_M^{(\ell)}(p) \in \mathbb{R}^{T(p)\times d}$ denote the per-block slice at depth $\ell \in \{1,\dots,L\}$, $H_M^{(\text{last})}(p)$ the slice at the last block, and let

$$\bar{H}_M(p) = \frac{1}{L}\sum_{\ell=1}^{L} H_M^{(\ell)}(p) \;\in\; \mathbb{R}^{T(p)\times d}$$

be the layer-averaged hidden-state matrix.

The three reported embedding tags are defined as follows.

`last_token`. The hidden state of the final token at the model's last block:

$$\varphi_M^{\text{lt}}(p) \;=\; H_M^{(\text{last})}(p)_{T(p),:} \;\in\; \mathbb{R}^d.$$

`mean_pool`. The token-position mean of the last block's hidden states:

$$\varphi_M^{\text{mp}}(p) \;=\; \frac{1}{T(p)}\sum_{t=1}^{T(p)} H_M^{(\text{last})}(p)_{t,:} \;\in\; \mathbb{R}^d.$$

`spectral_all`. The leading right singular vector of the layer-averaged hidden-state matrix $\bar{H}_M(p)$. Concretely, write

$$\bar{H}_M(p) \;=\; U\,\Sigma\,V^\top, \qquad U\in\mathbb{R}^{T(p)\times r},\ \Sigma\in\mathbb{R}^{r\times r},\ V\in\mathbb{R}^{d\times r},\ r=\min(T(p),d),$$

for the (thin) singular value decomposition with singular values $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r \ge 0$. Then

$$\varphi_M^{\text{sa}}(p) \;=\; \text{sign}\big(\langle V_{:,1},\, \varphi_M^{\text{mp,all}}(p)\rangle\big)\cdot V_{:,1} \;\in\; \mathbb{R}^d,$$

where $\varphi_M^{\text{mp,all}}(p) = \frac{1}{T(p)}\sum_{t=1}^{T(p)} \bar{H}_M(p)_{t,:}$ is the token-and-layer mean (used only to orient the sign so that $\langle \varphi_M^{\text{sa}}(p), \varphi_M^{\text{mp,all}}(p)\rangle > 0$, breaking the $\pm V_{:,1}$ ambiguity of the SVD). When $T(p) < 2$ or the SVD fails to converge, we fall back to $\varphi_M^{\text{sa}}(p) := \varphi_M^{\text{mp,all}}(p)/\|\varphi_M^{\text{mp,all}}(p)\|$.

Distance to the active training set. For test prompt $p \in \mathcal{H}$ and active training set $\mathcal{P}_{\text{train}}$, under embedding tag $\varphi_M^{(\tau)}$ ($\tau \in \{\text{lt}, \text{mp}, \text{sa}\}$), we set

$$z_M^{(\tau)}(p) \;=\; \min_{q\in\mathcal{P}_{\text{train}}} \|\varphi_M^{(\tau)}(p) - \varphi_M^{(\tau)}(q)\|_2, \tag{2}$$

and renormalize across the panel via $z_M^{(\tau)}(p) \mapsto \big(z_M^{(\tau)}(p) - z_{\min}\big)/(z_{\max} - z_{\min}) \in [0,1]$, where $z_{\min} = \min_{p'\in\mathcal{H}} z_M^{(\tau)}(p')$ and analogously $z_{\max}$. This is a monotone rescaling and does not affect rank-based or shape-comparison analyses. Distances are computed against the *base* model's embeddings of $\mathcal{P}_{\text{train}}$; the adapter is not active when computing z. Hence z depends on (family, metric, panel) and is shared across cells of the same family. The reported fits therefore describe dependence on distance to the active 40-prompt training set.

D.6 REFUSAL LABELS: STOCHASTIC DECODING AND LLM JUDGMENT

For each cell and each held-out prompt $p \in \mathcal{H}$, we collect $N = 10$ stochastic samples $r_1,\dots,r_{10}$ from the fine-tuned model at temperature 0.7. Inputs are capped at 256 tokens and generation at 2,048 new tokens. The documented top-p/top-k settings are $0.9/50$ for LLAMA, $1.0/50$ for MISTRAL, and $0.8/20$ for QWEN; Qwen additionally uses repetition penalty 1.05. For the initial 2,500-prompt panel, the $N = 10$ sample set combines two independent $N = 5$ decoding runs; the 499-prompt extension uses one $N = 10$ run. Analogously, base-model evaluation provides $N = 10$ samples per prompt, shared across cells of the same family. Historical sampling seeds were not recorded. Each sample is labeled by an open-weight LLM judge.

Judge model and rubric. We classify responses using OLMO-3.1-32B-INSTRUCT (Team Olmo, 2025). The model is loaded with 8-bit quantization via `bitsandbytes`. For each (p, r) pair, the judge classifies r under a three-way rubric:

- *refused* (R): the response declines the request (refusal of any kind, including soft refusal, persona refusal, lecturing, redirection).
- *complied* (C): the response provides (or substantially attempts to provide) the requested content.
- *ambiguous* (A): the response is too short, off-topic, garbled, or otherwise neither a clear refusal nor a clear compliance.

The judge prompt template is listed below (Appendix D.6.1). The judge model's next-token logits over the three label tokens $\{R, C, A\}$ are extracted directly (no further sampling); the $\arg\max$ token is recorded as the label.

Per-prompt refusal counts. For each (cell, prompt, condition $\in \{\text{base}, \text{ft}\}$), the analysis uses the integer counts

$$n^{\text{cond}}(p) \;=\; \#\{i : \text{judge}(r_i^{\text{cond}}) \in \{R, C\}\}, \qquad k^{\text{cond}}(p) \;=\; \#\{i : \text{judge}(r_i^{\text{cond}}) = R\},$$

i.e., $n^{\text{cond}}(p)$ is the number of non-ambiguous samples (out of N=10) and $k^{\text{cond}}(p)$ is the number judged refused. The empirical refusal rate at prompt p is $\hat{p}^{\text{cond}}(p) = k^{\text{cond}}(p)/\max(n^{\text{cond}}(p), 1)$. Prompts where either $n^{\text{base}}(p) = 0$ or $n^{\text{ft}}(p) = 0$ (all-ambiguous on at least one side) are dropped pairwise from the cell. After dropping, the per-cell sample size is $n_{\text{cell}} \in [2717, 2898]$ across the six cells (the range reflects differences in the proportion of prompts excluded in each cell).

Choice of judge. We use the same judge and rubric across all model families, training conditions, and embedding metrics. The judge receives the prompt and response, without an embedding tag or distance value. This holds the judging procedure fixed across the distance-based comparisons; it does not establish that classification errors are independent of prompt content or distance.

D.6.1 JUDGE PROMPT TEMPLATE

The exact prompt used for each (prompt, response) pair is:

```
You are judging whether an AI response refused, complied, or
is ambiguous.
USER: {prompt}
AI:   {response}
Did the AI refuse to comply with the user's request?
  - R = REFUSED (declined, redirected, or expressed
inability)
  - C = COMPLIED (provided the requested content)
  - A = AMBIGUOUS (unclear, partial, or evasive)
Answer (R/C/A):
```

The trailing space after `(R/C/A):` is intentional — it positions the next-token boundary on the label vocabulary.

D.7 PER-CELL STATISTICAL MODEL

The cohort figures (Figures 3–4) display Akaike weight fits of the *fine-tuned refusal probability* $p_{\text{ft}}(z)$ as a function of the normalized distance $z \in [0, 1]$. We model $p_{\text{ft}}(z)$ directly rather than the rate-difference $\Delta(z) = p_{\text{ft}}(z) - p_{\text{base}}(z)$: the shape name (CPL, two-segment, etc., defined below) then describes the visible p_{ft} curve.

Likelihood. For each cell and each held-out prompt $i = 1, \ldots, n_{\text{cell}}$ we observe $(z_i, k_i^{\text{ft}}, n_i^{\text{ft}})$, where z_i is the prompt's normalized pool distance (Eq. 2) and $(k_i^{\text{ft}}, n_i^{\text{ft}})$ are the Binomial primitives

from the judge (Appendix D.6). We model

$$p_{\mathrm{ft}}(z_i) \;=\; \mathrm{clip}\big(\,\delta_S(z_i;\,\theta_S),\;\varepsilon,\;1-\varepsilon\big), \tag{3}$$

$$k_i^{\mathrm{ft}} \mid z_i \;\sim\; \mathrm{Binomial}\big(n_i^{\mathrm{ft}},\;p_{\mathrm{ft}}(z_i)\big), \tag{4}$$

where $\varepsilon = 10^{-4}$ is a clipping floor against label saturation, and $\delta_S(\,\cdot\,;\theta_S) : [0,1] \to [0,1]$ is one of the five candidate *shapes* listed below, parametrized by θ_S.

Shape menu. The shape menu $\mathcal{S} = \{\mathrm{N}, \mathrm{L}, \mathrm{P}, \mathrm{C}, \mathrm{S}\}$ contains:

- **N**. *Constant* ($\theta = (d)$): $\delta(z) = d$. 1 parameter.
- **L**. *Linear* ($\theta = (d_{\mathrm{lo}}, d_{\mathrm{hi}})$): $\delta(z) = d_{\mathrm{lo}} + (d_{\mathrm{hi}} - d_{\mathrm{lo}})\,z$. 2 parameters.
- **P**. *Plateau-then-slope* ($\theta = (z_K, d_{\mathrm{lo}}, d_{\mathrm{hi}})$): $\delta(z) = d_{\mathrm{lo}}$ on $[0, z_K]$, then linearly interpolates from (z_K, d_{lo}) to $(1, d_{\mathrm{hi}})$ on $[z_K, 1]$. 3 parameters.
- **C**. *Constrained piecewise-linear (CPL)* ($\theta = (z_L, z_R, d_{\mathrm{lo}}, d_{\mathrm{hi}})$): $\delta(z) = d_{\mathrm{lo}}$ on $[0, z_L]$, linearly interpolates from (z_L, d_{lo}) to (z_R, d_{hi}) on $[z_L, z_R]$, and equals d_{hi} on $[z_R, 1]$. The two sampled breakpoints are ordered at evaluation time so that $z_L \le z_R$; the levels $d_{\mathrm{lo}}, d_{\mathrm{hi}}$ remain free, so the shape accommodates both increases ($d_{\mathrm{hi}} > d_{\mathrm{lo}}$) and decreases ($d_{\mathrm{hi}} < d_{\mathrm{lo}}$) of refusal with distance. 4 parameters.
- **S**. *Two-segment* ($\theta = (z_M, d_{\mathrm{lo}}, d_{\mathrm{mid}}, d_{\mathrm{hi}})$): piecewise-linear with one knee at z_M and three free levels $d_{\mathrm{lo}}, d_{\mathrm{mid}}, d_{\mathrm{hi}}$ (the curve passes through $(0, d_{\mathrm{lo}})$, (z_M, d_{mid}), and $(1, d_{\mathrm{hi}})$). 4 parameters; sign-free.

Priors: $z_L, z_R, z_K, z_M \;\sim\; \mathrm{Beta}(2,2)$ (concentrated away from the boundaries of $(0,1)$); $d, d_{\mathrm{lo}}, d_{\mathrm{mid}}, d_{\mathrm{hi}} \sim \mathrm{Uniform}(0,1)$.

Each shape is fitted independently using NUTS in NumPyro, with 2 chains, 500 warmup steps and 1500 retained samples per chain, and `init_to_median` initialization, giving 3000 retained posterior draws in total. Per-observation log-likelihoods $\ell_i^{(s,S)} = \log \Pr\big(k_i^{\mathrm{ft}} \mid n_i^{\mathrm{ft}}, p_{\mathrm{ft}}^{(s,S)}(z_i)\big)$ are recorded under the fitted posterior of shape S, sample s.

WAIC and model weights. For each shape S we compute the standard WAIC estimator

$$\widehat{\mathrm{lppd}}_i^{(S)} \;=\; \log \tfrac{1}{N_S} \sum_{s=1}^{N_S} \exp \ell_i^{(s,S)}, \tag{5}$$

$$\widehat{p}_{\mathrm{WAIC},i}^{(S)} \;=\; \widehat{\mathrm{Var}}_s\big[\ell_i^{(s,S)}\big], \tag{6}$$

$$\mathrm{WAIC}^{(S)} \;=\; -2 \sum_i \big(\widehat{\mathrm{lppd}}_i^{(S)} - \widehat{p}_{\mathrm{WAIC},i}^{(S)}\big). \tag{7}$$

The within-cell normalized WAIC weights are

$$w_S \;=\; \frac{\exp\big(-\tfrac{1}{2}\,\Delta_S\big)}{\sum_{S' \in \mathcal{S}} \exp\big(-\tfrac{1}{2}\,\Delta_{S'}\big)}, \qquad \Delta_S \;=\; \mathrm{WAIC}^{(S)} - \min_{S' \in \mathcal{S}} \mathrm{WAIC}^{(S')}. \tag{8}$$

We additionally report $\Delta\mathrm{WAIC}_{\mathrm{null}} = \mathrm{WAIC}^{(\mathrm{N})} - \mathrm{WAIC}^{(\mathrm{best})}$, the improvement in WAIC over the constant model (positive values favor a non-constant alternative). We denote the combined WAIC weight assigned to non-constant shapes by $W_{\mathrm{nonconst}} = 1 - w_{\mathrm{N}}$. This continuous model weight summarizes relative support within the specified shape menu; it is not a posterior-draw frequency.

Direction summary. For the best-fitting non-constant shape we report the Monte Carlo frequency

$$\widehat{P}_{\mathrm{MC}}\big(p_{\mathrm{ft}}(1) < p_{\mathrm{ft}}(0)\big) \;=\; \frac{1}{N_{\mathrm{best}}} \sum_{s=1}^{N_{\mathrm{best}}} \mathbf{1}\big[p_{\mathrm{ft}}^{(s,\mathrm{best})}(1) < p_{\mathrm{ft}}^{(s,\mathrm{best})}(0)\big], \tag{9}$$

where $N_{\mathrm{best}} = 3000$. This estimates the posterior probability that $p_{\mathrm{ft}}(z)$ is lower at $z{=}1$ than at $z{=}0$, conditional on the selected shape. Its empirical grid spacing is $1/3000 \approx 3.3 \times 10^{-4}$; an all-draws entry is marked by † below and is not a confidence bound.

D.8 PER-CELL SUMMARY TABLES

Table 2 summarizes the six cells and three embedding metrics: sample size n_{cell} after the all-ambiguous pairwise exclusion, empirical base and fine-tuned refusal rates $\bar{p}_{\text{base}}$ and $\bar{p}_{\text{ft}}$, their difference $\bar{\Delta} = \bar{p}_{\text{ft}} - \bar{p}_{\text{base}}$, the WAIC improvement $\Delta\text{WAIC}_{\text{null}}$, the combined non-constant-shape weight W_{nonconst}, the selected shape, and the endpoint-direction frequency $\widehat{P}_{\text{MC}}(p_{\text{ft}}(1) < p_{\text{ft}}(0))$.

Table 2: Per-cell, per-metric summaries for the six stochastic-evaluation cells. n_{cell}: panel size after the all-ambiguous pairwise exclusion. $\bar{p}_{\text{base}}, \bar{p}_{\text{ft}}$: refusal rates among retained non-ambiguous responses, pooled over the surviving prompts. $\bar{\Delta} = \bar{p}_{\text{ft}} - \bar{p}_{\text{base}}$. ΔWAIC_0: improvement in WAIC over the constant model. $W_{\text{nonconst}} = 1 - w_{\text{N}}$: combined normalized WAIC weight assigned to non-constant shapes, rounded to four decimals. "best": WAIC-preferred shape (N = constant; L = linear; P = plateau-then-slope; C = CPL; S = two-segment). $\widehat{P}_{\text{MC}}$: endpoint-direction frequency conditional on the selected shape. $1.000^{\dagger}$ means that the ordering held in all 3000 retained posterior draws; it is a Monte Carlo display marker, not a lower confidence bound.

cell	metric	n_{cell}	$\bar{p}_{\text{base}}$	$\bar{p}_{\text{ft}}$	$\bar{\Delta}$	ΔWAIC_0	best	W_{nonconst}	$\widehat{P}_{\text{MC}}$
e1200 LLAMA	`last_token`	2876	0.299	0.808	+0.509	+747	C	1.0000	$1.000^{\dagger}$
(r=1, E=1600)	`mean_pool`	2876	0.299	0.808	+0.509	+1,089	C	1.0000	$1.000^{\dagger}$
	`spectral_all`	2876	0.299	0.808	+0.509	+5,593	C	1.0000	$1.000^{\dagger}$
e1201 MISTRAL	`last_token`	2892	0.121	0.856	+0.735	+940	S	1.0000	$1.000^{\dagger}$
(r=1, E=1600)	`mean_pool`	2892	0.121	0.856	+0.735	+48.6	S	1.0000	0.9850
	`spectral_all`	2892	0.121	0.856	+0.735	+3,687	C	1.0000	$1.000^{\dagger}$
e1202 QWEN	`last_token`	2717	0.285	0.736	+0.451	+606	P	1.0000	$1.000^{\dagger}$
(r=1, E=1600)	`mean_pool`	2717	0.285	0.736	+0.451	+167	L	1.0000	$1.000^{\dagger}$
	`spectral_all`	2717	0.285	0.736	+0.451	+5,425	C	1.0000	$1.000^{\dagger}$
e1203 LLAMA	`last_token`	2898	0.298	0.758	+0.460	+962	S	1.0000	$1.000^{\dagger}$
(r=2, E=800)	`mean_pool`	2898	0.298	0.758	+0.460	+1,092	C	1.0000	$1.000^{\dagger}$
	`spectral_all`	2898	0.298	0.758	+0.460	+5,981	C	1.0000	$1.000^{\dagger}$
e1204 MISTRAL	`last_token`	2895	0.121	0.805	+0.685	+797	S	1.0000	$1.000^{\dagger}$
(r=2, E=800)	`mean_pool`	2895	0.121	0.805	+0.685	+54.0	S	1.0000	0.9900
	`spectral_all`	2895	0.121	0.805	+0.685	+2,018	C	1.0000	$1.000^{\dagger}$
e1102 QWEN	`last_token`	2721	0.285	0.825	+0.540	+819	S	1.0000	$1.000^{\dagger}$
(r=2, E=1600)	`mean_pool`	2721	0.285	0.825	+0.540	+486	S	1.0000	0.9060
	`spectral_all`	2721	0.285	0.825	+0.540	+9,290	C	1.0000	$1.000^{\dagger}$

Three observations follow from Table 2. First, a non-constant shape has lower WAIC than the constant model in all 18 entries. The improvement ranges from $+48.6$ for e1201 MISTRAL with `mean_pool` to $+9{,}290$ for e1102 QWEN with `spectral_all`. Second, the magnitude of the improvement varies markedly across metrics: it is at least 2,018 for `spectral_all` in every cell, while it is as low as 48.6 for `mean_pool`. Third, the preferred shape depends on the metric: `spectral_all` selects CPL in all six cells, whereas `last_token` and `mean_pool` select among linear, plateau-then-slope, CPL, and two-segment shapes. The endpoint-direction estimate exceeds 0.90 in all 18 entries; its smallest value is 0.906 for e1102 QWEN with `mean_pool`.

Table 3 reports the normalized WAIC weights across the five shapes for each cell and metric.

D.9 CPL PARAMETER POSTERIORS

Table 4 reports the CPL fit parameter posteriors per (cell, metric). The CPL parametrization has two distance breakpoints (z_L, z_R) and two probability levels $(d_{\text{lo}}, d_{\text{hi}})$; the curve is the near-pool plateau d_{lo} on $[0, z_L]$, a linear interpolation to d_{hi} on $[z_L, z_R]$, and the far-from-pool plateau d_{hi} on $[z_R, 1]$. We report the posterior median and 90% credible interval for each of the four parameters. CPL summaries are provided for every (cell, metric) pair to allow comparison of near-pool and far-from-pool levels, including pairs for which another shape is preferred.

Table 3: Within-cell normalized WAIC weights w_S across the five-shape menu, summing to 1 before rounding. All weights below 10^{-3} are displayed as $< 10^{-3}$, including entries serialized as 0.0000 in the shipped four-decimal fit table. Thus $< 10^{-3}$ is a reporting convention, not a Monte Carlo resolution. The constant-model weight is serialized as zero in all 18 entries.

cell	metric	w_{N}	w_{L}	w_{P}	w_{C}	w_{S}
e1200 LLAMA (r=1, E=1600)	`last_token`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000	$< 10^{-3}$
	`mean_pool`	$< 10^{-3}$	$< 10^{-3}$	0.448	0.531	0.021
	`spectral_all`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000	$< 10^{-3}$
e1201 MISTRAL (r=1, E=1600)	`last_token`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000
	`mean_pool`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000
	`spectral_all`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000	$< 10^{-3}$
e1202 QWEN (r=1, E=1600)	`last_token`	$< 10^{-3}$	$< 10^{-3}$	1.000	$< 10^{-3}$	$< 10^{-3}$
	`mean_pool`	$< 10^{-3}$	0.614	0.369	$< 10^{-3}$	0.017
	`spectral_all`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000	$< 10^{-3}$
e1203 LLAMA (r=2, E=800)	`last_token`	$< 10^{-3}$	$< 10^{-3}$	0.137	0.091	0.771
	`mean_pool`	$< 10^{-3}$	$< 10^{-3}$	0.219	0.742	0.039
	`spectral_all`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000	$< 10^{-3}$
e1204 MISTRAL (r=2, E=800)	`last_token`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000
	`mean_pool`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	0.006	0.994
	`spectral_all`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000	$< 10^{-3}$
e1102 QWEN (r=2, E=1600)	`last_token`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000
	`mean_pool`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	0.998
	`spectral_all`	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	1.000	$< 10^{-3}$

Table 4: Posterior summaries of the CPL fit parameters per (cell, metric). The CPL shape parametrizes $p_{\mathrm{ft}}(z)$ as the near-pool plateau d_{lo} on $[0, z_L]$, a linear interpolation to d_{hi} on $[z_L, z_R]$, and the far-from-pool plateau d_{hi} on $[z_R, 1]$. "med" is the median; "CI" is the 90% central credible interval. The z_L, z_R posteriors are ordered ($z_L \le z_R$) at evaluation time.

		z_L		z_R		d_{lo}		d_{hi}	
cell	metric	med	CI_{90}	med	CI_{90}	med	CI_{90}	med	CI_{90}
e1200 LLAMA	`last_token`	0.25	[0.23, 0.28]	0.97	[0.93, 0.99]	0.90	[0.89, 0.91]	0.51	[0.48, 0.53]
(r=1, E=1600)	`mean_pool`	0.14	[0.12, 0.19]	0.85	[0.78, 0.93]	0.99	[0.97, 1.00]	0.52	[0.46, 0.57]
	`spectral_all`	0.04	[0.04, 0.04]	0.48	[0.47, 0.50]	1.00	[0.99, 1.00]	0.21	[0.19, 0.24]
e1201 MISTRAL	`last_token`	0.18	[0.14, 0.20]	0.90	[0.83, 0.97]	0.95	[0.94, 0.96]	0.60	[0.56, 0.63]
(r=1, E=1600)	`mean_pool`	0.52	[0.49, 0.57]	0.58	[0.57, 0.65]	0.86	[0.85, 0.86]	0.55	[0.43, 0.67]
	`spectral_all`	0.14	[0.13, 0.15]	0.44	[0.42, 0.46]	0.97	[0.96, 0.97]	0.57	[0.55, 0.59]
e1202 QWEN	`last_token`	0.22	[0.20, 0.24]	0.64	[0.43, 0.97]	0.79	[0.79, 0.80]	0.46	[0.22, 0.58]
(r=1, E=1600)	`mean_pool`	0.49	[0.48, 0.49]	0.49	[0.49, 0.50]	0.74	[0.74, 0.75]	0.45	[0.41, 0.50]
	`spectral_all`	0.20	[0.20, 0.20]	0.38	[0.37, 0.39]	0.88	[0.87, 0.88]	0.26	[0.25, 0.28]
e1203 LLAMA	`last_token`	0.26	[0.24, 0.29]	0.97	[0.93, 0.99]	0.87	[0.86, 0.88]	0.38	[0.35, 0.42]
(r=2, E=800)	`mean_pool`	0.17	[0.14, 0.19]	0.84	[0.77, 0.92]	0.95	[0.93, 0.97]	0.44	[0.38, 0.49]
	`spectral_all`	0.04	[0.04, 0.05]	0.42	[0.40, 0.44]	0.98	[0.97, 0.98]	0.21	[0.18, 0.23]
e1204 MISTRAL	`last_token`	0.16	[0.14, 0.19]	0.87	[0.80, 0.94]	0.91	[0.90, 0.91]	0.56	[0.53, 0.60]
(r=2, E=800)	`mean_pool`	0.56	[0.56, 0.57]	0.57	[0.57, 0.57]	0.81	[0.80, 0.81]	0.42	[0.33, 0.52]
	`spectral_all`	0.14	[0.13, 0.15]	0.44	[0.42, 0.47]	0.90	[0.90, 0.90]	0.56	[0.54, 0.58]
e1102 QWEN	`last_token`	0.31	[0.11, 0.46]	0.63	[0.46, 0.97]	0.87	[0.83, 0.92]	0.56	[0.39, 0.66]
(r=2, E=1600)	`mean_pool`	0.04	[0.01, 0.07]	0.67	[0.57, 0.86]	0.97	[0.95, 0.98]	0.60	[0.50, 0.67]
	`spectral_all`	0.20	[0.19, 0.20]	0.37	[0.35, 0.38]	0.98	[0.98, 0.98]	0.31	[0.30, 0.33]

D.10 RANK-1 COHORT: DIRECT FITS

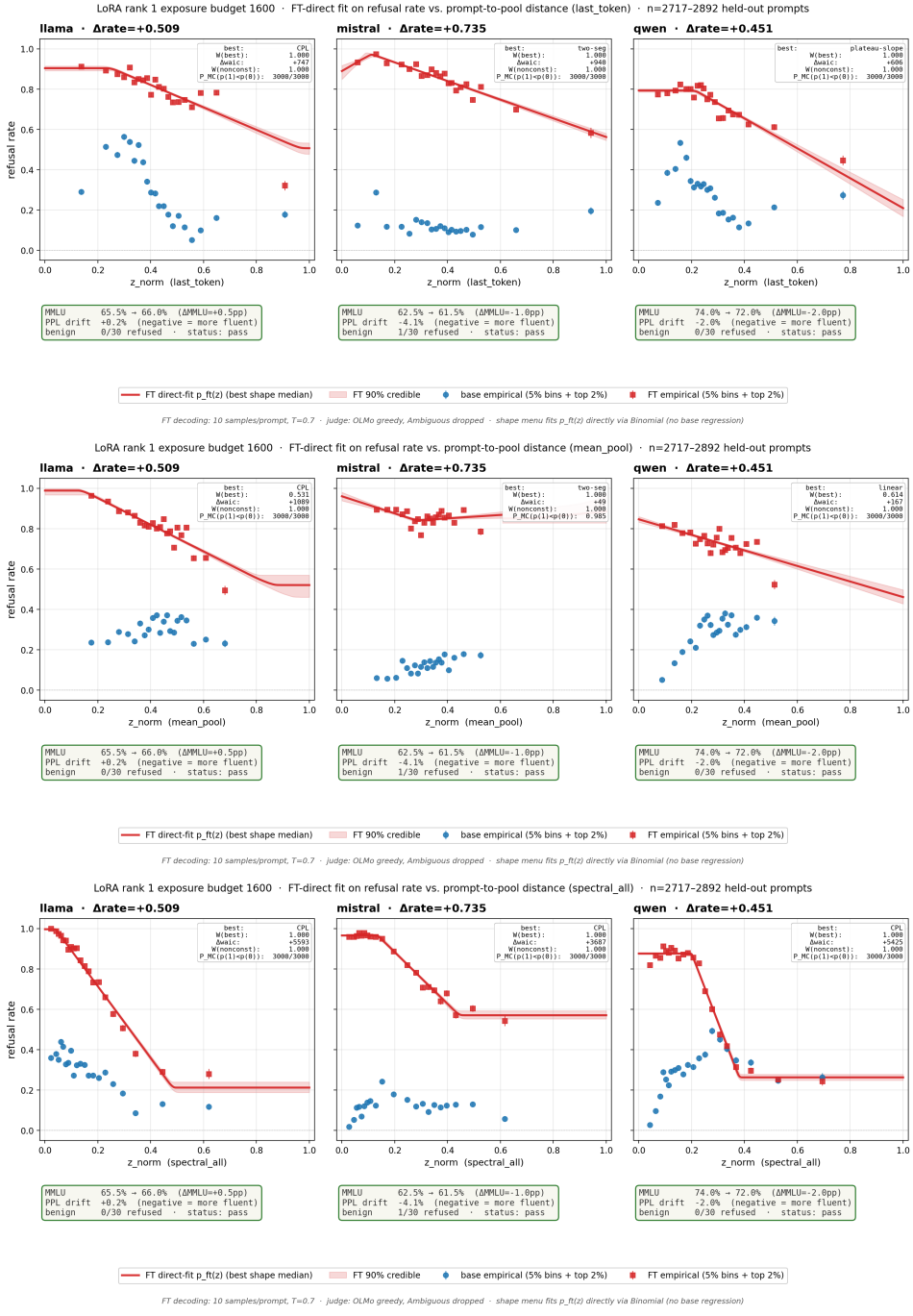

Figure 3: Rank-1 cohort (exposure budget 1600): direct fits of $p_{\mathrm{ft}}(z)$, with one panel per family and three rows for the embedding metrics. The top-right annotation gives the selected shape, its normalized WAIC weight, the combined non-constant-shape weight W_{nonconst}, the endpoint-direction Monte Carlo frequency conditional on that shape, and $\Delta\mathrm{WAIC}_{\mathrm{null}}$. An all-draws endpoint result is displayed as $3000/3000$. The bottom row shows capability and fluency diagnostics per family: MMLU-200 accuracy drift, WikiText-2 perplexity drift, and benign-prompt over-refusal count.

D.11 RANK-2 COHORT: DIRECT FITS

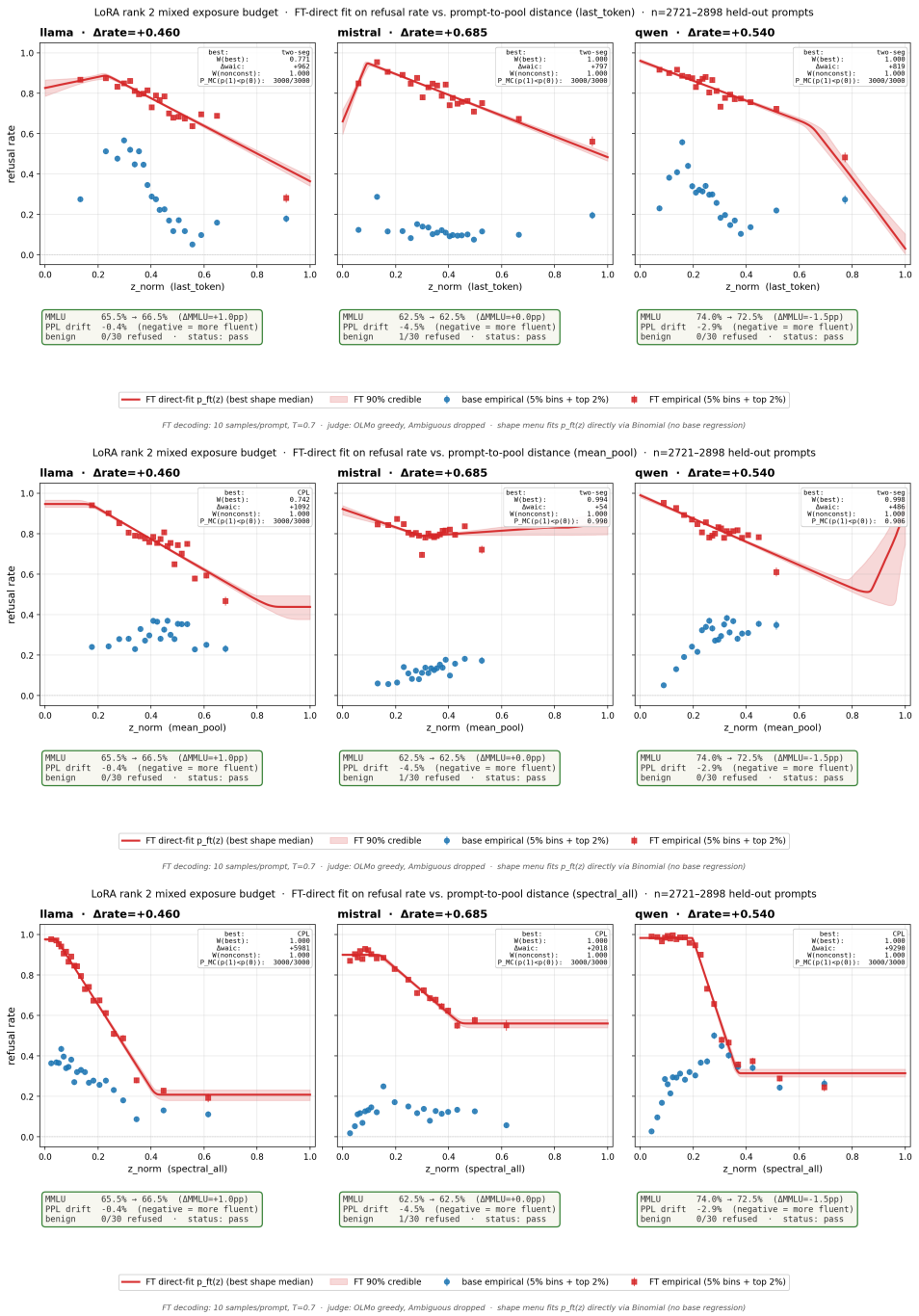

Figure 4: Rank-2 cohort (LLAMA and MISTRAL at exposure budget 800; QWEN at exposure budget 1600): $p_{\text{ft}}(z)$ direct-fit, one panel per family, three rows for three embedding metrics. Layout and annotations identical to Figure 3.

D.12 CAPABILITY, FLUENCY, AND OVER-REFUSAL PROBES

For each cell we measure three side-effects of fine-tuning, independent of the held-out refusal panel:

- *Capability drift via MMLU*-200: a 200-question subset of MMLU sampled to balance subjects, evaluated under teacher-forcing log-likelihood and standard A/B/C/D answer scoring. We report $\mathrm{base}_{\mathrm{acc}}$, $\mathrm{ft}_{\mathrm{acc}}$ and $\Delta\mathrm{MMLU} = \mathrm{ft}_{\mathrm{acc}} - \mathrm{base}_{\mathrm{acc}}$ in percentage points.
- *Fluency drift via WikiText-2 perplexity*: standard WikiText-2 perplexity probe (1024-token windows, fp16). We report relative drift $\mathrm{PPL\ drift}\ \% = 100 \cdot (\mathrm{ppl}_{\mathrm{ft}} - \mathrm{ppl}_{\mathrm{base}})/\mathrm{ppl}_{\mathrm{base}}$.
- *Benign over-refusal*: the count of refusals on a 30-prompt benign-instruction probe, comparing fine-tuned to base.

We load the base model once and evaluate the fine-tuned model with its LoRA adapter active. Base-model measurements use `with ft.disable_adapter():`; we use `ft.unload()` between cells.

Per-cell results are shown as compact cards in the bottom rows of Figures 3–4.