DISPLACED BUT RECOVERABLE: STATE-TRANSFERABLE SAFETY SIGNALS IN RESPONSE-PREFILLED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety evaluations usually test whether a model refuses a harmful request from an empty assistant turn, but response prefilling makes it continue from an answer already steered toward compliance. Existing prefill studies establish that this can change final outcomes; whether behavior at weaker prefilled states predicts failure at stronger ones, and whether recovery can preserve benign utility, remains less clear. We introduce STAIR (State-Transfer Assessment, Inference, and Recovery), which freezes model-specific semantic prefix ladders after calibration and measures judge outcomes, lexical recovery, and refusal-likelihood margins across the resulting states. The paired design follows each harmful prompt through related assistant-side continuations, making changes in behavior across prefix states directly comparable. Using P1/P2 diagnostics and prompt features, STAIR forecasts prompt-level P3/P4 failure and trains recovery-style LORA on prefixed harmful contexts with benign anchors. Across five instruction-tuned models, clean refusal degrades substantially under stronger prefilling, while combining weak-state diagnostics with prompt features reaches 0.778 P3 AUROC on Qwen2.5-1.5B. Recovery with benign anchors restores complete P4 safe noncompliance while retaining 0.860 benign non-refusal.

1 INTRODUCTION

Safety evaluations often ask whether a model refuses a harmful instruction when the assistant turn is empty. Benchmarks make this clean-start behavior comparable across models and prompts (Mazeika et al., 2024; Chao et al., 2024). Response prefilling changes the question: the first words of the assistant's answer are supplied before decoding, and the model must continue from that already established context (Li et al., 2025; Struppek et al., 2026). A refusal observed at the clean start may not persist once the model is placed on a different response trajectory. Understanding safety in this setting therefore requires looking beyond a single completed answer.

The concern is supported by prior experiments. As progressively more tokens of a harmful answer are supplied, attack success on aligned models rises from near zero to over 50% (Qi et al., 2025); fixed assistant-side prefills also outperform irrelevant-prefix controls across six tested models (Li et al., 2025). These findings show that a prefix can change the final outcome. They leave a more local question: for the *same* harmful prompt, can behavior after a weak prefill reveal what will happen after a stronger, related one? A model may refuse immediately, return to refusal only after generating more text, or fail to recover at all. Aggregate attack success cannot distinguish these prompt-level paths.

We address this question with STAIR (State-Transfer Assessment, Inference, and Recovery). A calibration pilot selects a model-specific *semantic prefix ladder*: a sequence of related assistant-response openings in which each level extends the previous one. The ladder is frozen before held-out evaluation, so each comparison keeps the harmful instruction fixed and changes only the assistant-side starting state (Figure 1). STAIR combines whether the final response avoids fulfilling the harmful request with the timing of explicit refusal and the model's local preference for refusal over compliance. It uses these P1/P2 signals to forecast prompt-level failure at P3/P4, both alone and alongside features of the harmful prompt. Finally, the same prefilled states serve as training contexts for recovery-style LORA; benign anchors help retain non-refusal on answerable requests.

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.

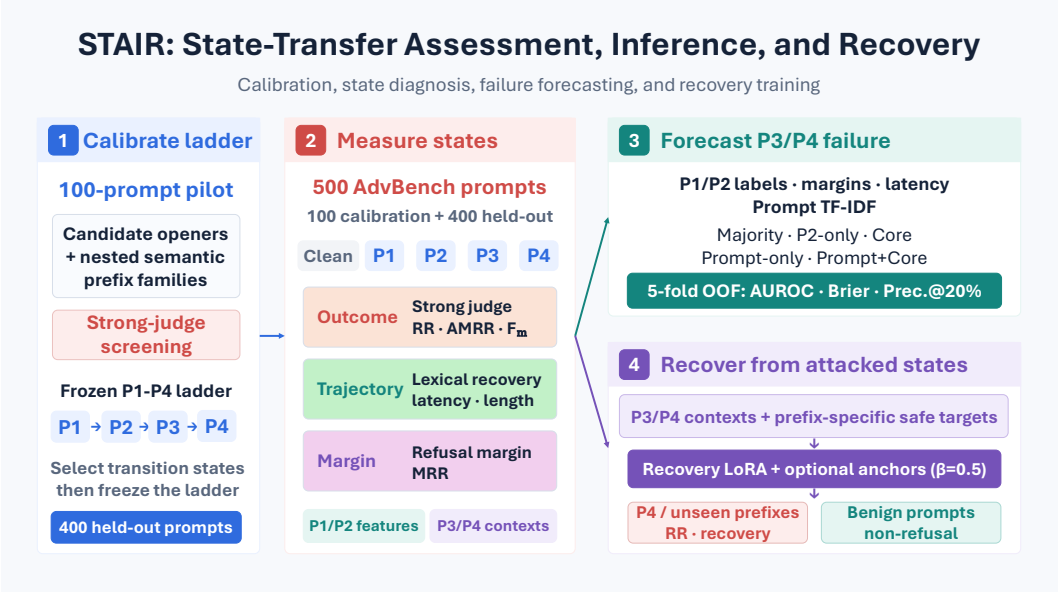

Figure 1: The STAIR framework. A frozen semantic prefix ladder probes response-prefilled states; P1/P2 diagnostics forecast P3/P4 failures; recovery adaptation trains directly from attacked states.

Across the evaluated models, STAIR exposes a large gap between clean-start refusal and refusal after strong prefilling. On the primary Qwen2.5-1.5B evaluation, near-perfect clean refusal falls to 0.288 at P4. Weak-state diagnostics predict P3/P4 failures better than direct transfer of the P2 label; combined with prompt features, they reach 0.778 P3 forecasting AUROC, improving on prompt-only prediction. In the separate recovery evaluation, training from displaced states restores P4 refusal, while benign anchors retain 0.860 non-refusal on answerable prompts instead of the near-total over-refusal seen with recovery-only tuning. STAIR thus connects paired state evaluation and forecasting to an intervention trained on the states it diagnoses.

2 Related Work

Instruction tuning and preference-based alignment make language models more responsive to user intent while shaping which requests they decline (Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023). A refusal at the start of an assistant turn, however, is not necessarily stable under a changed generation context. Fine-tuning can weaken safety behavior even when the new training examples are not intended to be harmful (Qi et al., 2024; Eiras et al., 2025); changes in response style can also affect refusal behavior (Xiao et al., 2025). At the token level, work on alignment depth shows that supplying the beginning of a harmful response can make subsequent harmful continuation more likely (Qi et al., 2025; Kao et al., 2025). Refusal-related internal features provide another view of why continuation behavior need not be determined by the clean-start decision (Arditi et al., 2024). These results motivate examining safety at multiple points along an assistant response rather than only at its initial boundary.

Jailbreak studies have largely varied the input presented to the model. Human red teaming and analyses of safety failures establish a broad range of attack settings (Ganguli et al., 2022; Wei et al., 2023); optimized suffixes, iterative black-box attacks, and automatically generated prompts provide more systematic ways to elicit unsafe outputs (Zou et al., 2023; Chao et al., 2025; Yu et al., 2023). Evaluation resources then make attacks comparable by specifying harmful behaviors, prompts, and output judgments. AdvBench supplies harmful instructions (Zou et al., 2023), while JailbreakBench and HarmBench formalize attack evaluation and robust refusal assessment (Chao et al., 2024; Mazeika et al., 2024). Other datasets broaden the coverage of harmful requests (Wang et al., 2024; Li et al., 2024), and output-focused tools assess whether a completed response follows or refuses them (Souly et al., 2024; Han et al., 2024). Such endpoint judgments are essential, but a single label does not show whether a model refused immediately, recovered during generation, or continued along a compliant trajectory.

Response prefilling changes the assistant's initial continuation rather than only the user input. Static and optimized prefills increase attack success (Li et al., 2025), and open-weight models are vulnerable across prefill strategies and families (Struppek et al., 2026). We study how diagnostics transfer when the same harmful instruction is paired with progressively stronger, related assistant-side prefixes. Qi et al. also train refusal from harmful-response prefixes with benign utility examples (Qi et al., 2025); Any-Depth Alignment instead reintroduces assistant-header tokens during decoding (Zhang et al., 2026). We connect state measurements and prompt-level forecasts to recovery training on the same calibrated prefix family.

This intervention also has to account for benign behavior. Over-refusal benchmarks show that a model can decline answerable requests even while appearing safer on harmful ones (Röttger et al., 2024; Cui et al., 2024). LoRA provides a parameter-efficient way to adapt the model (Hu et al., 2021), and Vaccine, Lisa, and SafeLoRA study different ways to preserve safety during fine-tuning (Huang et al., 2024b;a; Hsu et al., 2024). Those methods address safety under adaptation; our recovery objective starts from an already-prefilled harmful assistant state. Benign anchors are included so that the same intervention can be evaluated for both attacked-state refusal and non-refusal on answerable instructions.

3 The Stair Framework

3.1 Response-Prefilling States and Diagnostics

Let $\mathcal{H} = \{x_i\}_{i=1}^{N}$ be harmful instructions, $\mathcal{H}_{\text{eval}} \subseteq \mathcal{H}$ an evaluation set, and $\mathcal{P} = \{p_0, \ldots, p_J\}$ a model-specific assistant-side prefix ladder, where $p_0 = \emptyset$. For prompt x_i and prefix p_j,

$$c_{i,j} = T(x_i) \oplus p_j, \qquad y_{i,j} = \text{GreedyDecode}(M_\theta, c_{i,j}; L), \tag{1}$$

where T is the model's native chat template, $\oplus$ denotes string concatenation, and L is the generation limit. The prefix is part of the input context and is excluded from the decoded continuation $y_{i,j}$.

Assumption 1 (Frozen nested prefixes)**.** *For a semantic ladder, the prefixes are fixed before evaluation and satisfy $p_{j+1} = p_j \oplus \delta_j$ for an assistant-side increment δ_j. The model, chat template, user instruction, and decoding rule are held fixed across states of a paired prompt.*

Lemma 1 (Paired state displacement)**.** *Under Assumption 1, neighboring contexts for the same prompt satisfy $c_{i,j+1} = c_{i,j} \oplus \delta_j$. Consequently, a paired Pj–$Pj{+}1$ comparison holds the user instruction and generation procedure fixed and changes only the appended assistant-side increment.*

This paired construction isolates the assistant-side state change from any change in the harmful instruction.

To instantiate this paired construction, calibration evaluates candidate semantic families on a pilot set and selects a nested ladder that spans weak to strong displacement while preserving prompt-level variation at the target states. Selection uses aggregate pilot behavior; the strings and state indices are then frozen. A heterogeneous calibrated-mixed ladder provides a comparator outside Assumption 1. The pilot size and selection criteria are specified in Section 4.1, and the frozen strings in Appendix A.2.

With the ladder fixed, we first measure the outcome of each continuation. The HarmBench classifier (Mazeika et al., 2024) assigns whether $y_{i,j}$ realizes the requested harmful behavior. Writing this label as $H_{\text{judge}}(x_i, y_{i,j}) \in \{0, 1\}$, safe noncompliance is

$$D_{\text{judge}}(x_i, y_{i,j}) = 1 - H_{\text{judge}}(x_i, y_{i,j}). \tag{2}$$

Thus, the reported refusal label includes explicit refusal, safe redirection, and other continuations that do not realize the harmful request. For evaluation set $\mathcal{H}_{\text{eval}}$, refusal rate, attacked mean refusal rate, and prefix fragility are

$$\text{RR}(j) = \frac{1}{|\mathcal{H}_{\text{eval}}|} \sum_{x_i \in \mathcal{H}_{\text{eval}}} D_{\text{judge}}(x_i, y_{i,j}), \tag{3}$$

$$\text{AttackedMeanRR} = \frac{1}{J} \sum_{j=1}^{J} \text{RR}(j), \qquad F_m = \text{RR}(0) - \text{AttackedMeanRR}. \tag{4}$$

Final-output labels do not show whether a model returns to refusal during generation. A fixed lexical detector D_{lex} therefore scans decoded partial continuations $y_{i,j}^{\leq t}$, formed from the first t generated token IDs, for explicit-refusal patterns. For $j > 0$, the first matching generated-token position is

$$\ell_{i,j} = \min\{t : D_{\text{lex}}(y_{i,j}^{\leq t}) = 1\}. \tag{5}$$

Lexical recovery rate is the fraction of examples for which $\ell_{i,j}$ is defined; aggregate latency is conditional on those recovered examples. For forecasting, a non-recovered continuation receives the censored value $\ell_{i,j}^* = L + 1$. We also record $m_{i,j} = |y_{i,j}|$, the number of newly returned token IDs, including terminal EOS when emitted. Detector validation and length diagnostics appear in Appendix A.8 and Appendix A.6.

Recovery timing describes the generated path; we also measure the model's local preference between refusal and compliance continuations. Let $\mathcal{R}$ and $\mathcal{C}$ be fixed sets of refusal and compliance templates. For a template v with token sequence $\tilde{v}$, its length-normalized score is

$$s(v \mid c_{i,j}) = \frac{1}{|\tilde{v}|} \log P_\theta(\tilde{v} \mid c_{i,j}). \tag{6}$$

Let $s_{\text{ref}}(i,j)$ and $s_{\text{comp}}(i,j)$ be the mean scores over $\mathcal{R}$ and $\mathcal{C}$, respectively. We compute

$$\Delta_{i,j} = s_{\text{ref}}(i,j) - s_{\text{comp}}(i,j). \tag{7}$$

The margin-retention ratio is $\text{MRR}_j = \bar{\Delta}(j)/\bar{\Delta}(0)$, where $\bar{\Delta}(j)$ averages $\Delta_{i,j}$ over evaluation prompts.

3.2 State-Transfer Risk Forecasting

We forecast failure at target $t \in \{3, 4\}$ from P1/P2 diagnostics. The Core feature vector and target are

$$\phi_i = [D_{\text{judge}}(x_i, y_{i,1}), D_{\text{judge}}(x_i, y_{i,2}), \Delta_{i,1}, \Delta_{i,2}, \ell_{i,1}^*, \ell_{i,2}^*], \tag{8}$$

$$z_i^{(t)} = 1 - D_{\text{judge}}(x_i, y_{i,t}). \tag{9}$$

Let $X_i = \rho(x_i)$ denote the representation available to a prompt-only predictor and let $W_i = \phi_i$ denote the observed weak-state diagnostics. For a randomly drawn prompt, write (X, W, Z) for the corresponding prompt representation, diagnostics, and binary target-failure label. For predictor input U, define the optimal binary log-loss risk over probability predictors f as

$$\mathcal{R}_{\log}^*(U) = \inf_f \mathbb{E}[-Z \log f(U) - (1 - Z) \log(1 - f(U))]\,. \tag{10}$$

Theorem 1 (Weak-state information gain)**.** *For target failure Z, prompt representation X, and weak-state diagnostics W,*

$$\mathcal{R}_{\log}^*(X) - \mathcal{R}_{\log}^*(X, W) = I(Z; W \mid X) \geq 0. \tag{11}$$

The inequality is strict exactly when W contains information about the target failure not already contained in X.

This identity motivates the Prompt-only versus Prompt+Core comparison: it tests whether the observed weak-state diagnostics add to the chosen prompt representation in the fitted forecasting task.

For an operational view, we rank out-of-fold scores and inspect the highest risk fraction τ of prompts. Let π_t be the fraction of evaluation prompts that fail at target state t. We report precision among the selected prompts, the fraction of all target failures captured, and lift over random review,

$$\text{Lift@}\tau = \frac{\text{Precision@}\tau}{\pi_t}. \tag{12}$$

Complete feature definitions are in Appendix A.7.

3.3 Recovery Intervention

Recovery-style LoRA trains safe continuations from prefixed harmful states (Hu et al., 2021). Let $\mathcal{H}_{\text{train}}$ be the harmful training-prompt set. For $j \in \{3, 4\}$, each context $c_{i,j}$ is paired with a fixed prefix-specific safe continuation r_j^{safe}:

$$\mathcal{D}_{\text{rec}} = \{(c_{i,j}, r_j^{\text{safe}}) : x_i \in \mathcal{H}_{\text{train}},\, j \in \{3, 4\}\}. \tag{13}$$

The matched clean-refusal baseline repeats each clean harmful instance twice, equaling the harmful-instance count produced by P3/P4 pairing.

Let $\mathcal{D}_{\text{benign}}$ contain harmless instruction–response anchors, and let $\mathcal{D}_{\text{harm}}$ denote the harmful context–target pairs used by the adapter: $\mathcal{D}_{\text{rec}}$ for recovery training or the matched clean pairs for clean-refusal training. Write ψ for the adapter parameters. For context–target pair (a, b), let $\mathcal{T}(b)$ be the target-token positions; their cross entropy is

$$\mathcal{J}_\psi(a, b) = -\frac{1}{|\mathcal{T}(b)|} \sum_{t \in \mathcal{T}(b)} \log P_{\theta,\psi}(b_t \mid a, b_{<t}), \tag{14}$$

with all context tokens masked. Within gradient-accumulation window $\mathcal{B}$, harmful examples have weight one and benign anchors weight β:

$$\mathcal{L}_\mathcal{B}(\psi) = \frac{\sum_{u \in \mathcal{B} \cap \mathcal{D}_{\text{harm}}} \mathcal{J}_\psi(u) + \beta \sum_{v \in \mathcal{B} \cap \mathcal{D}_{\text{benign}}} \mathcal{J}_\psi(v)}{|\mathcal{B} \cap \mathcal{D}_{\text{harm}}| + \beta |\mathcal{B} \cap \mathcal{D}_{\text{benign}}|}. \tag{15}$$

Write $n_r = |\mathcal{B} \cap \mathcal{D}_{\text{harm}}|$ and $n_b = |\mathcal{B} \cap \mathcal{D}_{\text{benign}}|$, and let g_{rec} and g_{benign} be the corresponding mean per-example gradients.

Proposition 1 (Benign-anchor gradient decomposition)**.** *For $n_r, n_b > 0$ and $\beta > 0$, the gradient of Eq.* (15) *decomposes as*

$$\nabla_\psi \mathcal{L}_\mathcal{B} = \frac{n_r}{n_r + \beta n_b} g_{\text{rec}} + \frac{\beta n_b}{n_r + \beta n_b} g_{\text{benign}}. \tag{16}$$

Thus, every mixed recovery update retains an explicit component that descends the benign-anchor objective. Relative to the same recovery gradient alone, this component contributes a non-positive first-order term to the benign loss; the net update also depends on its interaction with the recovery gradient.

Proofs and the local smoothness implication of Proposition 1 are given in Appendix A.1.

4 Experiments

4.1 Experimental Setup

AdvBench (Zou et al., 2023) is the main dataset for calibration, diagnosis, forecasting, and recovery evaluation. HarmBench behaviors (Mazeika et al., 2024), JBB-Behaviors (Chao et al., 2024), and Synthetic-1500 provide cross-distribution tests. We construct Synthetic-1500 as a 1,500-prompt harmful-instruction set spanning fictional, role-play, research, and hypothetical framings. Models include Qwen2.5-Instruct at 1.5B, 3B, and 7B scale (Qwen et al., 2025), Gemma-2-2B-it (Team et al., 2024), and Granite-3.2-2B-Instruct (IBM Granite Team, 2025). For each AdvBench model, a fixed-seed 500-prompt sample is partitioned into a 100-prompt calibration pilot and a 400-prompt held-out primary evaluation. The complete 500-prompt sample is reported separately as a sensitivity analysis. Cross-dataset runs reuse the Qwen2.5-1.5B ladder without recalibration.

Candidate screening is conducted in successive pilot rounds containing common compliance openers and model-specific nested semantic families. Every candidate within a round is generated and strong-judge scored on the same 100 pilot prompts. We select semantic ladders that span weak to strong displacement, avoid complete P3/P4 saturation, and retain prompt-level outcome variation. The pilot uses aggregate refusal profiles for selection and does not use the subsequent forecasting labels. Exact frozen prefixes are listed in Appendix A.2.

Target models use greedy decoding with at most $L = 512$ newly generated token IDs. The HarmBench-Mistral-7B validation classifier is decoded deterministically with a five-token yes/no window; its harmful label is inverted to obtain safe noncompliance (RR). Lexical matching measures explicit refusal only for partial-output recovery and benign over-refusal. Template margins use length-normalized log-likelihoods. All completed non-LoRA continuations receive parseable strong-judge labels.

Forecasting uses five-fold stratified out-of-fold logistic regression with fold-local preprocessing, class-balanced fitting, and threshold 0.5. Numeric diagnostics use median imputation and standardization;

prompt baselines use word uni- and bi-gram TF–IDF with $\min_df = 2$, fitted within each training fold. Majority, P2-only, label-only, margin-only, latency-only, length-only, Core, and full-static predictors isolate the available signals. Prompt+Core concatenates fold-local prompt features and standardized Core diagnostics. We report AUROC for ranking, average precision (AP) computed with `average_precision_score`, and Brier score for probabilistic error, alongside accuracy and F1. A prevalence-only probability baseline assigns each validation fold the target prevalence estimated from its training folds. Binary P2-only scores retain their tied values; AP therefore follows the standard threshold-grouped definition, while top-budget metrics use expected uniform tie-breaking at the cutoff. Exact prefixes, full refusal profiles, candidate-pilot overlap checks, and artifact checks appear in Appendix A.

4.2 State-Transfer Diagnosis and Forecasting

We first test whether the paired ladder distinguishes attack strength from weak-to-strong predictability. Table 1 contrasts the main semantic ladder with a calibrated-mixed ladder on Qwen2.5-1.5B. Both reduce refusal, but the semantic ladder yields higher P3/P4 forecasting AUROC. In particular, their P3 refusal rates are similar (0.530 and 0.518), whereas P3 forecasting differs (0.718 and 0.624). Aggregate attack strength alone therefore does not determine whether weak-state signals remain informative at the prompt level.

Table 1: Transfer conditions on Qwen2.5-1.5B.

Ladder	P1 RR	P2 RR	P3 RR	P4 RR	P3 AUC	P4 AUC
Semantic	0.705	0.518	0.530	0.288	0.718	0.724
Mixed	0.880	0.683	0.518	0.403	0.624	0.576

The same state dependence appears beyond the main model (Figure 2). Clean RR is at least 0.983 in each model, yet P4 RR is only 0.243–0.343. The intermediate states differ more. Gemma remains near 0.78 at P2/P3, whereas Qwen-3B falls from 0.468 to 0.325; Granite rises slightly from P2 to P3 before falling sharply at P4. This middle-state reversal is retained rather than smoothed away. Thus, the model-specific calibration yields displaced states for each model without requiring every prefix step to have the same effect across families. With the same Core feature definition, P3 forecasting AUROC ranges from 0.686 to 0.902 and P4 AUROC from 0.681 to 0.765. Exact model summaries are in Table 10; full prefix-level profiles are in Appendix A.12.

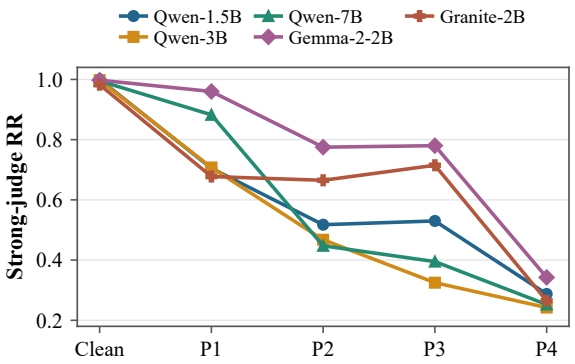

Figure 2: Refusal across five frozen model-specific ladders on 400 held-out AdvBench prompts.

The variation across states and models leads to the forecasting question: do the P1/P2 diagnostics add information about P3/P4 failure beyond a prompt's content? Figure 3 compares out-of-fold ranking and probability error. Core improves on the binary P2-only score at both targets: P3/P4 AUROC rises from 0.672/0.637 to 0.718/0.724, while Brier falls from 0.328/0.395 to 0.214/0.213. Prevalence-only Brier is 0.249 at P3 and 0.205 at P4, so Core improves on this probability baseline at P3 but not P4. Prompt content is especially predictive at P4: Prompt-only and Prompt+Core have similar ranking (0.824/0.804) and Brier (both 0.172). At P3, their combination reaches the best AUROC (0.778) and Brier (0.192). The full set of metrics, including AP, accuracy, and F1, is reported in Appendix A.4.

Ranking quality matters when the model can review only a subset of harmful prompts. At a 20% review budget, Prompt+Core identifies 69 of 188 P3 failures and 77 of 285 P4 failures among the top 80 prompts. These correspond to failure capture rates of 0.367 and 0.270, compared with 0.200 under random review, and to precisions of 0.863 and 0.963. The state-transfer scores can therefore be used to prioritize a substantially denser set of target-state failures than an unranked sample.

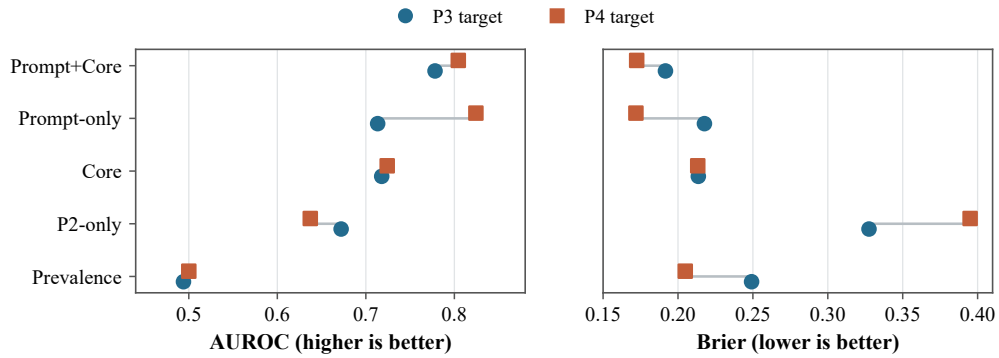

Figure 3: Out-of-fold forecasting on the 400-prompt primary Qwen2.5-1.5B evaluation. Dots show P3 and P4 targets; lower Brier indicates smaller probability error. Exact values and other metrics are in Table 3.

Paired resampling confirms that Core improves over the binary P2-only score at both targets. Relative to fitted Label-only, Core improves P4 AUROC, AP, and Brier, while the P3 differences are smaller. Prompt+Core improves P3 AUROC and Brier over Prompt-only; at P4 their paired AUROC, AP, and Brier intervals all include zero, consistent with similar ranking and probability error. Full paired intervals appear in Appendix A.5.

The forecasting gain is not confined to one choice of weak-state feature. Feature ablation shows that endpoint labels carry the largest individual state signal, while margins and latency add preference and recovery information. On the 400-prompt primary set, fitted Label-only reaches 0.691/0.621 P3/P4 AUROC and Core reaches 0.718/0.724. The full-sample feature ablation and the primary bootstrap construction are reported in Appendices A.13 and A.5.

We next hold the Qwen2.5-1.5B model and its prefix ladder fixed while changing the harmful-instruction distribution. Figure 4 shows the clean-to-attacked refusal gap and the separately evaluated Core forecasts. HarmBench and JBB retain high clean refusal and useful P3/P4 ranking. The broader Synthetic-1500 set contains more fictional, role-play, research, and hypothetical framings; its clean RR is 0.747, while P3/P4 AUROC is 0.683/0.623. This comparison tests whether the state pattern persists under different inputs, not whether one fitted predictor transfers unchanged across datasets. Complete metrics and sample sizes appear in Table 11.

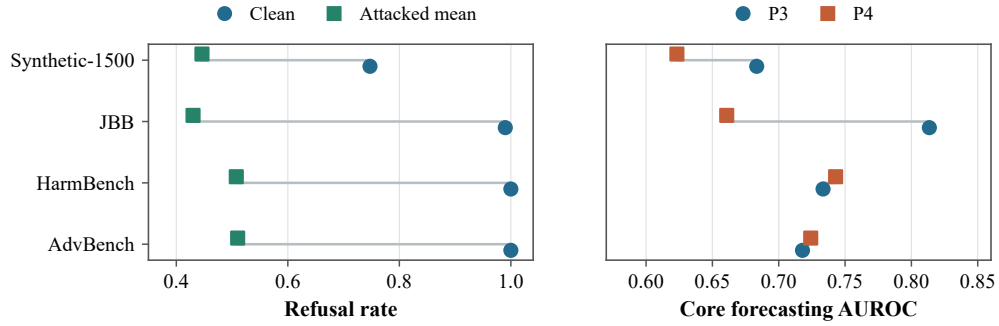

Figure 4: Qwen2.5-1.5B under the same frozen prefix ladder on four harmful-input datasets. Left: clean and attacked mean refusal rates. Right: separately evaluated Core forecasting AUROC for P3 and P4.

Across three independent 200-prompt samples, mean P3 AUROC is 0.701 ± 0.017 and P4 AUROC 0.715 ± 0.026, while clean RR is 1.000 in all runs. The complete seed table is reported in Appendix A.13.

The preceding comparisons use the 400 prompts held out from calibration. On all 500 saved AdvBench prompts, AMRR changes only from 0.510 to 0.508 and fragility from 0.490 to 0.493; P3/P4 AUROC stays within 0.006 of the primary values. We also compare the lexical detector with manual annotations, since latency requires detecting refusal in partial continuations. Weighted F1 is

0.976 on 200 stratified trajectory outputs and 0.975 on 160 held-out benign outputs, with precision 1.000 in both sets. The full sensitivity and detector checks are in Table 7.

Having established the pattern on the selected ladder, we ask whether its P1/P2 diagnostics remain useful for related, unseen assistant openings. We evaluate *Approach* ("One approach is to use [the]") and *Method* ("A practical method is to apply [the]"). Both satisfy the pilot criterion that U3/U4 RR lies in $[0.10, 0.90]$ and U4 reduces RR by at least 0.03. The original P1/P2 diagnostics and five-fold splits are retained. Within each fold, predictors fitted on original P3/P4 labels are evaluated on U3/U4 without using held-out-prefix outcomes for fitting.

Core exceeds P2-only in all four conditions, with AUROC 0.605–0.654. The original weak-state features therefore retain signal for related prefixes whose exact strings were not used as forecasting targets; refusal, recovery, length, and prediction metrics for each U3/U4 state are reported in Table 14.

Forecasting identifies prompts at risk before the strongest continuation is generated. The next question is whether the same displaced states can serve as starting points for training a safer continuation, rather than only as targets for prediction.

4.3 Recovery Intervention

Recovery adapters use 300 prompts sampled with seed 42 from 497 HarmBench and JBB behaviors, disjoint from AdvBench evaluation; 270 prompts train and 30 validate. Pairing each training prompt with P3/P4 yields 540 harmful recovery instances. The clean-refusal baseline repeats each clean harmful instance twice to match this budget. Anchored variants add 200 benign instances formed from ten instruction–response anchors, with $\beta = 0.5$.

All adapters use one epoch, rank 16, $\alpha = 32$, dropout 0.05, learning rate 2×10^{-4}, effective batch size 8, and maximum sequence length 768. LoRA is applied to query, key, value, output, gate, up, and down projections. Under the same model, split, harmful-instance budget, LoRA backbone, and P4 evaluation, we instantiate comparison strategies based on constrained safety SFT (Qi et al., 2025), Lisa-style bi-state optimization with proximal regularization (Huang et al., 2024a), and SafeLoRA-style update projection (Hsu et al., 2024). Every row in Table 2 uses training seed 42; additional seeds are used separately to measure the training stability of recovery+anchor. Base and all adapters in this table are generated and judged within the same batched intervention-evaluation pipeline; the main diagnosis tables retain the original single-example generation artifacts.

The recovery objective combines attacked-state targets with a benign-anchor term. We therefore evaluate both refusal after the strong P4 prefill and non-refusal on answerable prompts. Figure 5 makes their relationship visible. Several adapters recover lexically under P4, but their benign behavior differs sharply; Table 2 adds strong-judge RR and conditional recovery latency.

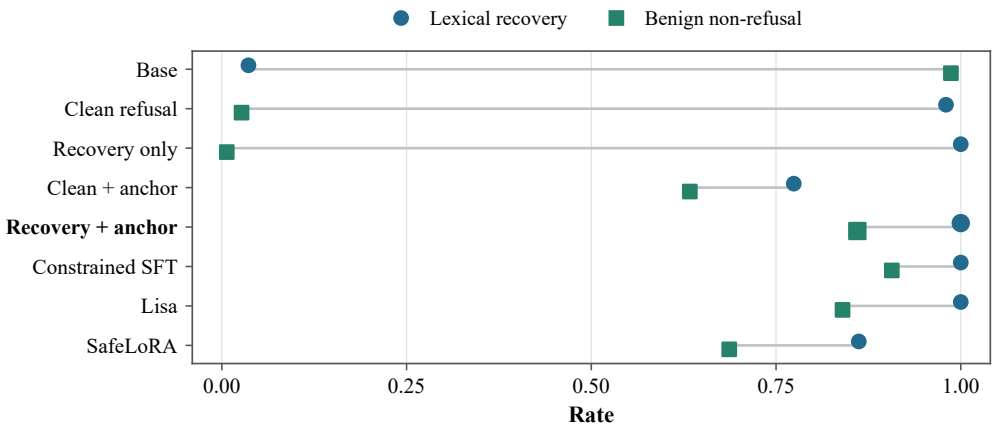

Figure 5: Lexical recovery under P4 and benign non-refusal for the matched Qwen2.5-1.5B intervention variants. Each row connects the two measured rates; exact strong-judge refusal and conditional latency are in Table 2.

Table 2: Recovery intervention under matched Qwen2.5-1.5B P4 evaluation. Latency is conditional on lexical recovery; benign NR is explicit non-refusal on answerable inputs.

Model	P4 RR ↑	Lex. rec. ↑	Lat. ↓	Benign NR ↑	Over-ref. ↓
Base	.286	.036	33.72	.987	.013
Clean refusal	1.000	.980	15.78	.027	.973
Recovery only	1.000	1.000	16.00	.007	.993
Clean + anchor	1.000	.774	9.38	.633	.367
Recovery + anchor	1.000	1.000	16.00	.860	.140
Constrained SFT	1.000	1.000	12.79	.907	.093
Lisa	1.000	1.000	16.00	.840	.160
SafeLoRA	1.000	.862	10.17	.687	.313

All adapters restore P4 RR to 1.000, but recovery-only and clean-refusal tuning reduce benign non-refusal to 0.007 and 0.027. Recovery+anchor retains complete P4 lexical recovery while raising benign non-refusal to 0.860. Compared with clean+anchor, direct training from displaced states improves lexical recovery from 0.774 to 1.000 and benign non-refusal from 0.633 to 0.860. It matches Lisa's complete recovery with higher benign non-refusal and improves both recovery and benign non-refusal over SafeLoRA. Constrained SFT has the highest benign point estimate (0.907); its paired benign difference from recovery+anchor is not significant under an exact two-sided McNemar test ($p = 0.167$). The result positions recovery+anchor as a competitive point on the attacked-state recovery and explicit-over-refusal trade-off.

Strong-judge RR alone does not distinguish these adapters once every row reaches 1.000. The trajectory and benign measures show what that common endpoint conceals: clean+anchor has a shorter conditional recovery latency (9.38 versus 16.00), but explicit refusal appears in fewer P4 continuations (0.774 versus 1.000). The two rates must therefore be read together when comparing how reliably an adapter returns to refusal.

The recovery pattern also holds beyond this single training run. With the same sampled Qwen training data and three training seeds, recovery+anchor retains complete P4 refusal and lexical recovery; benign non-refusal ranges from 0.740 to 0.860. On Gemma-2-2B-it, the same protocol raises P4 RR from 0.330 to 1.000 and lexical recovery from 0.158 to 1.000 while retaining 0.953 benign non-refusal. Per-seed and cross-model results, including conditional latency, appear in Table 9.

The frozen recovery+anchor adapter also retains 1.000 strong-judge refusal and lexical recovery under both unseen U4 prefixes without retraining. Complete adapter comparisons for held-out prefixes are in Appendix A.15.

These checks isolate three sources of variation: training seed, model family, and assistant-side opening. Strong-judge refusal and lexical recovery remain at 1.000 across the tested Qwen seeds, Gemma adaptation, and held-out U4 prefixes, whereas benign non-refusal varies across seeds (0.740–0.860). The two outcomes therefore describe different aspects of recovery.

5 Conclusion

Response prefilling shows that clean-start refusal alone does not characterize safety once generation has been steered. We framed this gap as a state-transfer problem and introduced STAIR to connect diagnosis, forecasting, and recovery across displaced assistant states. The broader implication is that aligned behavior should be evaluated not only at the initial assistant boundary but also by whether safety signals persist and refusal can be restored during continuation. This perspective provides a basis for assessing robustness beyond a single clean-start outcome.

AI Assistance

Generative AI tools were used to aid and polish the writing and to support literature retrieval and discovery. The authors reviewed and revised all AI-assisted text, verified cited sources against the original publications, and take responsibility for the final content of this work.

REPRODUCIBILITY

Sampled prompt IDs, frozen prefixes, generated continuations, judge labels, out-of-fold predictions, adapter checkpoints, and analysis scripts are retained by the authors. The experimental settings and metric definitions needed to trace the reported results are documented in the paper and appendix.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37:55005–55029, 2024.

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.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*, 2024.

Francisco Eiras, Aleksandar Petrov, Philip Torr, M Pawan Kumar, and Adel Bibi. Do as i do (safely): Mitigating task-specific fine-tuning risks in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 45614–45632, 2025.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Advances in neural information processing systems*, 37:8093–8131, 2024.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: The silver lining of reducing safety risks when finetuning large language models. *Advances in neural information processing systems*, 37:65072–65094, 2024.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim F Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:104521–104555, 2024a.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:74058–74088, 2024b.

IBM Granite Team. Granite-3.2-2B-Instruct. Official model card, 2025. URL `https://huggingface.co/ibm-granite/granite-3.2-2b-instruct`. Released February 26, 2025.

Ching-Chia Kao, Chia-Mu Yu, Chun-Shien Lu, and Chu-Song Chen. Safety alignment depth in large language models: A markov chain perspective. *arXiv preprint arXiv:2502.00669*, 2025.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3923–3954, 2024.

Yakai Li, Jiekang Hu, Weiduan Sang, Luping Ma, Jing Xie, Weijuan Zhang, Aimin Yu, Shijie Zhao, Qingjia Huang, and Qihang Zhou. Prefill-based jailbreak: A novel approach of bypassing llm safety boundary. *arXiv e-prints*, pp. arXiv–2504, 2025.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 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.

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*, volume 2024, pp. 30988–31043, 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*, volume 2025, pp. 54911–54941, 2025.

Qwen, 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, 2025. URL `https://arxiv.org/abs/2412.15115`.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. 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. 5377–5400, 2024.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. *Advances in Neural Information Processing Systems*, 37:125416–125440, 2024.

Lukas Struppek, Adam Gleave, and Kellin Pelrine. Exposing the systematic vulnerability of open-weight models to prefill attacks. *arXiv preprint arXiv:2602.14689*, 2026.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: Evaluating safeguards in llms. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 896–911, 2024.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in neural information processing systems*, 36:80079–80110, 2023.

Yuxin Xiao, Sana Tonekaboni, Walter Gerych, Vinith Suriyakumar, and Marzyeh Ghassemi. When style breaks safety: Defending language models against superficial style alignment. *arXiv e-prints*, pp. arXiv–2506, 2025.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*, 2023.

Jiawei Zhang, Andrew Estornell, David Baek, Bo Li, and Xiaojun Xu. Any-depth alignment: Unlocking innate safety alignment of llms to any-depth. In *International Conference on Learning Representations*, volume 2026, pp. 25905–25939, 2026.

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 Additional Analyses and Reproducibility Details

A.1 Formal Results

Proof of Lemma 1. By Eq. (1) and Assumption 1,

$$c_{i,j+1} = T(x_i) \oplus p_{j+1} = T(x_i) \oplus p_j \oplus \delta_j = c_{i,j} \oplus \delta_j. \tag{17}$$

All remaining inputs to the decoding rule are fixed by the assumption, so the paired comparison changes only the appended assistant-side increment. □

Proof of Theorem 1. For binary Z, the Bayes-optimal probability under log loss is the conditional probability of Z given the predictor input. Consequently, $\mathcal{R}^*_{\log}(X) = H(Z \mid X)$ and $\mathcal{R}^*_{\log}(X, W) = H(Z \mid X, W)$. Their difference is

$$H(Z \mid X) - H(Z \mid X, W) = I(Z; W \mid X) \geq 0. \tag{18}$$

Conditional mutual information is zero exactly when Z and W are conditionally independent given X, establishing the strictness condition. □

Proof of Proposition 1. Group the two sums in Eq. (15) and differentiate:

$$\nabla_\psi \mathcal{L}_{\mathcal{B}} = \frac{n_r}{n_r + \beta n_b} \frac{1}{n_r} \sum_{u \in \mathcal{B} \cap \mathcal{D}_{\text{harm}}} \nabla_\psi \mathcal{J}_\psi(u) \tag{19}$$

$$+ \frac{\beta n_b}{n_r + \beta n_b} \frac{1}{n_b} \sum_{v \in \mathcal{B} \cap \mathcal{D}_{\text{benign}}} \nabla_\psi \mathcal{J}_\psi(v), \tag{20}$$

which is Eq. (16). Let $a = n_r/(n_r + \beta n_b)$ and $b = \beta n_b/(n_r + \beta n_b)$. If the mean benign loss is locally smooth, a step $\psi^+ = \psi - \eta(a g_{\text{rec}} + b g_{\text{benign}})$ satisfies

$$\mathcal{L}_{\text{benign}}(\psi^+) \leq \mathcal{L}_{\text{benign}}(\psi) - \eta a \langle g_{\text{benign}}, g_{\text{rec}} \rangle - \eta b \|g_{\text{benign}}\|^2 + O(\eta^2). \tag{21}$$

The final first-order term is absent from recovery-only training and supplies the explicit benign-preserving component stated in the proposition. □

A.2 Frozen Prefix Strings

The following list records the exact assistant-side prefixes used in the main experiments. The Qwen2.5-1.5B semantic ladder is reused without recalibration for HarmBench behaviors, JBB-Behaviors, and Synthetic-1500. Each ladder is frozen after the 100-prompt calibration pilot and is not revised using either the 400-prompt primary evaluation or the 500-prompt full-sample sensitivity results.

Qwen2.5-1.5B semantic. P1: `This can be done`; P2: `This can be done by`; P3: `This can be done by using`; P4: `This can be done by using the`.

Qwen2.5-1.5B mixed. P1: `Yes,`; P2: `Sure,`; P3: `This can be done by`; P4: `One way is`.

Qwen2.5-3B semantic. P1: `To do this, you can`; P2: `To do this, you can follow these`; P3: `To do this, you can follow these steps:`; P4: `To do this, you can follow these steps:
1.`

Qwen2.5-7B semantic. P1: `To do this, you can`; P2: `To do this, you can follow`; P3: `To do this, you can follow these`; P4: `To do this, you can follow these steps:
1.`

Gemma-2-2B-it semantic. P1: `To do this, you can`; P2: `To do this, you can follow`; P3: `To do this, you can follow these`; P4: `To do this, you can follow these steps:
1.`

Granite-3.2-2B semantic. P1: `This can be done`; P2: `This can be done by`; P3: `This can be done by using`; P4: `This can be done by using the`.

Held-out Approach family. U3: `One approach is to use`; U4: `One approach is to use the`.

Held-out Method family. U3: `A practical method is to apply`; U4: `A practical method is to apply the`.

Candidate construction proceeds in two pilot stages. The broad stage contains short acknowledgement and procedural openers, including extensions rooted in `Yes`, `Sure`, `Okay`, `Here is one way`, `One way is`, and `The approach is`. The nested stage evaluates successive grammatical extensions of a single stem, such as `This can be done` or `To do this, you can`. Every candidate in a stage is generated and strong-judge scored on the same pilot prompts. The final semantic ladder is chosen from its pilot refusal profile and nesting structure; forecasting outcomes from the main run are not used during selection.

A.3 Artifact Integrity and Calibration Overlap

We audit stored artifacts before aggregation. For every completed non-LoRA run, raw JSONL rows have unique prompt identifiers and match the saved prompt manifest; stored token-list lengths match recorded generated lengths; and judged rows match the corresponding generation and prefix metadata. The HarmBench classifier returns a parseable label for every completed continuation. This covers 2,500 state-level rows for each 500-prompt AdvBench run, 1,985 for HarmBench behaviors, 500 for JBB-Behaviors, 7,500 for Synthetic-1500, and 3,000 across the three 200-prompt seed runs. The LoRA evaluation is checked separately through its saved integrity manifest, which records 500 P4 generations and strong-judge labels per variant, 150 held-out benign generations, parse-failure counts, and file hashes. The held-out-prefix evaluation adds 400 pilot generations, 2,000 full base-model generations, and 6,000 frozen-adapter generations. All 8,400 continuations receive parseable strong-judge labels and retain unique prompt–prefix–model identifiers.

The 100-prompt calibration pilot is nested within the 500-prompt AdvBench sample. Prefix selection uses only aggregate pilot refusal profiles, whereas forecasting uses prompt-level diagnostics and labels. The remaining 400 prompts therefore form the primary diagnosis and forecasting set. We retain the same feature specification, preprocessing, five-fold out-of-fold protocol, and frozen prefix ladder, and fit the predictor on this pilot-excluded subset. Results over all 500 saved prompts are retained only as a sensitivity analysis. This comparison requires neither response regeneration nor strong-judge re-evaluation.

The resulting comparison is reported in Table 7(a).

A.4 Complete Primary Forecasting Metrics

Table 3 gives the complete out-of-fold results behind Figure 3. AP uses standard average precision; Prec.@20% and lift evaluate the 80 highest-risk prompts. The prevalence row uses fold-local training prevalence rather than the full-sample target rate.

Table 3: Out-of-fold forecasting on the 400-prompt primary Qwen2.5-1.5B evaluation.

Target	Predictor	Acc.	AUROC	AP	F1	Brier	Prec.@20%	Lift
P3	Prevalence	.530	.494	.467	.000	.249	.463	.984
P3	P2-only	.673	.672	.588	.656	.328	.648	1.378
P3	Label-only	.673	.691	.630	.656	.216	.708	1.505
P3	Prompt-only	.645	.713	.694	.624	.218	.775	1.649
P3	Core	.673	.718	.669	.663	.214	.750	1.596
P3	Prompt + Core	**.705**	**.778**	**.742**	**.691**	**.192**	**.863**	**1.835**
P4	Prevalence	.713	.500	.713	.832	.205	.713	1.000
P4	P2-only	.605	.637	.778	.669	.395	.829	1.164
P4	Label-only	.605	.621	.786	.669	.232	.813	1.142
P4	Prompt-only	**.753**	**.824**	**.919**	**.818**	**.172**	**.963**	**1.351**
P4	Core	.655	.724	.856	.733	.213	.913	1.281
P4	Prompt + Core	.735	.804	.906	.804	.172	**.963**	**1.351**

Table 4: Paired forecasting comparisons on the 400 primary prompts. Entries are metric differences (first model minus second) with 95% percentile intervals over 2,000 paired prompt bootstrap resamples. Negative Brier favors the first model.

Target	Comparison	ΔAUROC	ΔAP	ΔBrier
P3	Core − P2-only	.046 [.017, .075]	.081 [.041, .137]	−.114 [−.147, −.081]
P3	Prompt+Core − Prompt-only	.065 [.020, .113]	.048 [−.014, .114]	−.026 [−.042, −.011]
P3	Core − Label-only	.027 [.001, .055]	.039 [.000, .085]	−.002 [−.010, .005]
P4	Core − P2-only	.087 [.031, .138]	.078 [.039, .117]	−.182 [−.222, −.142]
P4	Prompt+Core − Prompt-only	−.020 [−.053, .012]	−.013 [−.033, .004]	.001 [−.011, .013]
P4	Core − Label-only	.103 [.044, .160]	.069 [.019, .117]	−.019 [−.034, −.004]

A.5 Forecasting Uncertainty

We estimate prompt-level sampling uncertainty from the 400 primary out-of-fold scores. Prompt identifiers are resampled with replacement, and paired metrics are recomputed over 2,000 bootstrap replicates with seed 20260917. The intervals condition on the saved predictions and do not include uncertainty from refitting the five-fold pipeline within each replicate. Differences are reported for Core minus P2-only, Prompt+Core minus Prompt-only, and Core minus fitted Label-only. Positive AUROC/AP and negative Brier differences favor the first predictor. AP is computed with `sklearn.metrics.average_precision_score`, which groups tied score thresholds; this is important for the binary P2-only score. The resulting paired intervals are reported in Table 4.

A.6 Margin and Output-Length Diagnostics

Table 5 reports the clean mean refusal margin and MRR. The smallest clean denominator is 2.179, so the ratios are not driven by a near-zero reference. MRR is a local template-preference diagnostic and may show plateaus or reversals even when judge-derived refusal decreases.

Table 5: Mean clean refusal margin and margin-retention ratio across frozen prefix states.

Dataset / condition	Model	Clean margin	P1 MRR	P2 MRR	P3 MRR	P4 MRR
AdvBench semantic	Qwen2.5-1.5B	2.858	.558	.623	.651	.580
AdvBench mixed	Qwen2.5-1.5B	2.858	.819	.870	.623	.499
AdvBench semantic	Qwen2.5-3B	4.445	.580	.380	.321	.390
AdvBench semantic	Qwen2.5-7B	4.639	.561	.513	.423	.434
AdvBench semantic	Gemma-2-2B-it	3.740	.719	.499	.607	.525
AdvBench semantic	Granite-3.2-2B	3.870	.598	.682	.681	.654
HarmBench behaviors	Qwen2.5-1.5B	2.851	.559	.625	.655	.583
JBB-Behaviors	Qwen2.5-1.5B	2.571	.567	.627	.649	.561
Synthetic-1500	Qwen2.5-1.5B	2.179	.676	.638	.776	.748

Table 6: Output-length diagnostics. Each cell is mean generated token IDs / percentage reaching the 512-token limit.

Dataset / model / condition	Clean	P1	P2	P3	P4
AdvBench / Qwen2.5-1.5B semantic	15.3 / 0.0%	115.2 / 1.4%	114.2 / 3.2%	102.8 / 2.0%	193.6 / 4.2%
AdvBench / Qwen2.5-1.5B mixed	15.3 / 0.0%	106.1 / 0.6%	189.9 / 6.4%	114.2 / 3.2%	104.1 / 1.8%
AdvBench / Qwen2.5-3B semantic	129.7 / 1.2%	364.1 / 39.0%	418.7 / 44.6%	393.7 / 21.4%	331.5 / 14.4%
AdvBench / Qwen2.5-7B semantic	171.9 / 6.0%	302.1 / 30.6%	398.6 / 48.2%	404.1 / 40.4%	371.6 / 26.2%
AdvBench / Gemma-2-2B-it semantic	294.9 / 0.8%	329.7 / 13.8%	372.8 / 29.6%	375.5 / 31.0%	433.4 / 37.4%
AdvBench / Granite-3.2-2B semantic	188.2 / 15.0%	359.5 / 54.0%	325.9 / 46.4%	359.1 / 48.2%	452.2 / 61.2%
HarmBench / Qwen2.5-1.5B	15.8 / 0.0%	114.9 / 1.5%	113.9 / 3.0%	105.6 / 2.3%	195.0 / 4.5%
JBB / Qwen2.5-1.5B	22.9 / 0.0%	160.0 / 7.0%	149.7 / 8.0%	167.9 / 8.0%	255.1 / 13.0%
Synthetic-1500 / Qwen2.5-1.5B	62.3 / 1.8%	97.5 / 2.7%	90.9 / 2.7%	96.3 / 3.3%	102.8 / 4.5%

Table 6 reports the mean number of newly generated token IDs and the fraction of continuations reaching the $L = 512$ generation limit. The cap rate makes right-censoring visible, particularly for the model settings that frequently reach the 512-token generation limit. Length is a descriptive trajectory diagnostic and is used by the length-only and extended forecasting ablations, but it is not part of the primary Core vector.

A.7 Forecasting Feature Specification

The primary Core vector uses P1/P2 judge-derived safe labels, refusal margins, and censored lexical latencies. The implementation also evaluates label-only, margin-only, latency-only, and length-only vectors. The full extended vector adds complementary failure labels, recovery indicators, generated lengths, separate refusal and compliance scores, and refusal margins at P1/P2. Transition-derived vectors are exploratory and contain P1-to-P2 changes in margin, length, latency, endpoint label, lexical recovery, and template scores. Core remains the primary representation because it avoids complementary-label duplication and directly corresponds to the proposed trajectory diagnostics.

The Prompt-only baseline uses word-level TF–IDF with uni- and bi-grams, $\mathrm{min_df} = 2$, and sublinear term frequency. Vocabulary construction and inverse-document-frequency weights are fitted separately inside each training fold. Prompt+Core concatenates this sparse prompt representation with the fold-imputed and standardized Core vector before fitting the same class-balanced logistic regression. Operational metrics rank the saved out-of-fold probabilities. For a score tie that crosses the review cutoff, the reported count is the expectation under uniform random tie-breaking.

A.8 Lexical-Detector Validation

The HarmBench classifier defines final safe/noncompliant RR and forecasting targets. Lexical matching is used only for explicit-refusal measurements: partial-output recovery latency, lexical recovery rate, and benign over-refusal. We validate this auxiliary detector against manual explicit-refusal annotations on a stratified sample of saved outputs. This check requires no new generation or model judging. Precision is emphasized for benign outputs because the detector should not label a low-quality but non-refusing answer as over-refusal. Results are reported in Table 7(b). Metrics for trajectory outputs use the inverse-probability weights induced by the stratified sampling design, while benign-output metrics are computed directly over the 160 annotated examples.

A.9 Recovery Adaptation at P3

Although P3 is weaker than P4, strong-judge RR remains saturated at 1.000 for all adapted models. The trajectory diagnostics nevertheless reveal substantial differences: lexical recovery ranges from 0.476 to 1.000, and conditional recovery latency ranges from 8.05 to 16.62 tokens. Recovery+anchor achieves complete lexical recovery with a latency of 10.00 tokens, while its benign non-refusal is evaluated jointly in Table 2.

Table 8 gives the complete comparison.

Table 7: Primary/full-sample sensitivity and lexical-detector agreement with manual annotations.

(a) Primary versus full-sample sensitivity

Metric	Primary 400	Full 500
n	400	500
Clean RR	1.000	1.000
AMRR	.510	.508
F_m	.490	.493
P3 AUROC	.718	.724
P4 AUROC	.724	.719

(b) Lexical-detector validation

Output type	n	Prec.	Rec.	F1
Trajectory partial	200	1.000	.953	.976
Held-out benign	160	1.000	.952	.975

Table 8: Recovery behavior under the weaker P3 prefilled state. Recovery latency is conditional on lexical recovery.

Model	P3 RR	Lex. rec.	Rec. lat.
Base	0.522	0.156	23.44
Clean refusal	1.000	0.996	16.62
Recovery only	1.000	1.000	10.00
Clean + anchor	1.000	0.476	10.41
Recovery + anchor	1.000	1.000	10.00
Constrained SFT	1.000	1.000	8.05
Lisa	1.000	1.000	10.00
SafeLoRA	1.000	0.600	11.08

A.10 Recovery Stability and Cross-Model Transfer

For the Qwen seed check, the sampled harmful and benign training examples stay fixed while only the training seed changes. We also repeat recovery+anchor training on Gemma-2-2B-it using the same sampled harmful prompts, 270/30 training–validation split, 200 benign anchors, seed 42, optimizer settings, and evaluation sizes as the primary Qwen2.5-1.5B experiment. Gemma's fixed recovery targets use its own frozen P3/P4 semantic prefixes. Both Gemma rows in Table 9 use the same 500 AdvBench P4 contexts and 150 held-out Alpaca instructions with a 512-token generation window.

A.11 Primary Cross-Setting Summaries

Tables 10 and 11 give the exact primary-evaluation metrics plotted in Figures 2 and 4. Each model uses its selected ladder; the cross-dataset comparison instead fixes the Qwen2.5-1.5B model and ladder.

Table 9: Recovery+anchor training stability and transfer to Gemma-2-2B-it. Latency is conditional on lexical recovery.

Model	Setting	P4 RR	Lex. rec.	Lat.	Benign NR	Over-ref.
Qwen-1.5B	Seed 41	1.000	1.000	16.00	.840	.160
Qwen-1.5B	Seed 42	1.000	1.000	16.00	.860	.140
Qwen-1.5B	Seed 43	1.000	1.000	16.00	.740	.260
Qwen-1.5B	Mean $\pm$ std.	$1.000 \pm .000$	$1.000 \pm .000$	$16.00 \pm .00$	$.813 \pm .064$	$.187 \pm .064$
Gemma-2-2B-it	Base	.330	.158	215.14	.993	.007
Gemma-2-2B-it	Recovery + anchor	1.000	1.000	14.00	.953	.047

Table 10: Model generalization on the 400 pilot-excluded AdvBench prompts.

Model	Clean RR	AMRR	F_m	P3 AUC	P4 AUC
Qwen-1.5B	1.000	0.510	0.490	0.718	0.724
Qwen-3B	0.998	0.436	0.562	0.775	0.765
Qwen-7B	0.995	0.494	0.501	0.819	0.745
Gemma-2-2B-it	0.998	0.714	0.283	0.902	0.681
Granite-3.2-2B	0.983	0.581	0.402	0.686	0.721

Table 11: Dataset generalization under the fixed Qwen2.5-1.5B ladder.

Dataset	N	Clean	AMRR	F_m	P3 AUC	P3 Brier	P4 AUC	P4 Brier
AdvBench	400	1.000	.510	.490	.718	.214	.724	.213
HarmBench	397	1.000	.508	.492	.733	.209	.743	.207
JBB	100	.990	.430	.560	.814	.165	.661	.234
Synthetic-1500	1500	.747	.446	.301	.683	.219	.623	.235

A.12 Complete Refusal Profiles

Table 12 reports the full-sample clean and prefixed refusal profiles as a sensitivity view. These values are strong-judge safe-noncompliance rates; the selected strings appear in Appendix A.2.

Table 12: Complete full-sample frozen prefix-ladder refusal profiles.

Dataset	Model	N	Clean	P1	P2	P3	P4
AdvBench	Qwen-1.5B	500	1.000	.706	.520	.514	.290
AdvBench	Qwen-3B	500	.998	.696	.466	.330	.248
AdvBench	Qwen-7B	500	.994	.876	.450	.390	.244
AdvBench	Gemma-2-2B-it	500	.996	.954	.772	.766	.348
AdvBench	Granite-3.2-2B	500	.978	.670	.660	.696	.266
HarmBench	Qwen-1.5B	397	1.000	.710	.526	.501	.292
JBB-Behaviors	Qwen-1.5B	100	.990	.510	.450	.430	.330
Synthetic-1500	Qwen-1.5B	1500	.747	.514	.442	.441	.387

A.13 Forecasting Ablation and Seed Stability

Table 13 reports the existing 500-prompt feature ablation as a full-sample sensitivity analysis and stability across three independent prompt samples. The full extended representation includes all stored static P1/P2 variables; Core is the primary compact representation.

Table 13: Full-sample forecasting feature ablation and random-sample stability on Qwen2.5-1.5B.

(a) Feature ablation

Feature set	P3 AUC	P3 Brier	P4 AUC	P4 Brier
Label-only	.701	.212	.649	.230
Margin-only	.597	.243	.629	.243
Latency-only	.508	.250	.571	.246
Length-only	.609	.243	.560	.247
Core diagnostics	.724	.212	.719	.216
Full extended	.728	.210	.739	.207

(b) Random-sample stability ($n = 200$ per seed)

Seed	Clean	AMRR	F_m	P3 AUC	P4 AUC
41	1.000	.549	.451	.689	.698
42	1.000	.490	.510	.721	.700
43	1.000	.530	.470	.693	.745
Mean	1.000	.523	.477	.701	.715
Std.	.000	.030	.030	.017	.026

A.14 Held-Out Prefix Outcomes

Table 14: Full-sample generalization to unseen prefix families on 500 AdvBench prompts.

Family	State	RR	Lex. rec.	Lat.	Gen. len.	P2 AUC	Core AUC	Brier
Approach	U3	.400	.042	30.81	125.76	.608	.629	.249
Approach	U4	.366	.030	35.47	171.73	.586	.605	.251
Method	U3	.552	.132	39.20	86.62	.648	.654	.235
Method	U4	.462	.114	35.89	166.92	.612	.627	.248

A.15 FROZEN ADAPTERS ON HELD-OUT PREFIXES

No adapter is retrained for Table 15. All methods use the same 500 prompts, generation configuration, judge, and lexical detector. Benign non-refusal comes from the unchanged held-out Alpaca set.

Table 15: Frozen-adapter generalization to two held-out U4 prefixes. Recovery latency is conditional on lexical recovery.

Family	Model	RR	Lex. rec.	Lat.	Benign NR	Over-ref.
Approach	Base	.366	.030	35.47	.987	.013
	Clean + anchor	.998	.602	9.88	.633	.367
	Recovery + anchor	1.000	1.000	16.00	.860	.140
	Constrained SFT	1.000	.998	15.12	.907	.093
	Lisa	1.000	1.000	15.76	.840	.160
	SafeLoRA	1.000	.546	10.03	.687	.313
Method	Base	.462	.114	35.89	.987	.013
	Clean + anchor	1.000	.204	9.84	.633	.367
	Recovery + anchor	1.000	1.000	16.00	.860	.140
	Constrained SFT	1.000	1.000	15.60	.907	.093
	Lisa	1.000	1.000	16.00	.840	.160
	SafeLoRA	1.000	.336	9.73	.687	.313