READING BETWEEN THE STATES:CONTRASTING HIDDEN-STATE DYNAMICS FOR JAILBREAK DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Jailbreak attacks disguise harmful requests through diverse prompt constructions, posing a persistent threat to the safe deployment of large language models (LLMs). Recent defenses increasingly exploit internal hidden-state patterns across layers for jailbreak detection. However, these approaches fall short of explicitly modeling the underlying transition relationships, leaving their predictive value underexplored as generalizable detection evidence. Crucially, while unseen attacks exhibit unfamiliar trajectories, their state transitions often follow predictive regularities shared with known jailbreaks. This suggests a distinct detection principle: identifying which class-specific dynamics better predict the observed evolution. To realize this principle, we introduce **CORD** (**C**ompeting **O**perator **R**esidual **D**etection), which detects jailbreaks by comparing two predictive explanations of hidden-state evolution. CORD learns benign and jailbreak dynamical models through a conditional Koopman formulation, using shared observables to model transitions across token progression and network depth. Both models predict the same observed successors under identical context, making their prediction errors directly comparable. The resulting residual contrast assigns higher risk when jailbreak dynamics explain the observed evolution better than benign dynamics. Extensive experiments across four LLMs and ten attack families show that CORD achieves 93.94% macro TPR@1, surpassing the strongest evaluated baseline by 38.17%. Comprehensive experimental analyses demonstrate that predictive contrast captures discriminative differences between benign and jailbreak hidden-state dynamics, providing a promising basis for detecting unseen jailbreak attacks. Code: anonymous repository.

1 INTRODUCTION

Large language models (LLMs) have emerged as core components of modern intelligent systems, powering sophisticated workflows ranging from complex code synthesis to interactive decision support (Yang et al., 2024; Wang et al., 2025; Brynjolfsson et al., 2025; Tu et al., 2025). Yet jailbreak attacks exploit their instruction-following capabilities to circumvent safeguards and elicit harmful responses through adversarial prompt constructions (Zou et al., 2023; Chao et al., 2025; Jia et al., 2025; Jiang et al., 2024). These threats underscore the need for effective defenses to support the safe deployment of LLM-based systems. However, maintaining robust defenses across diverse jailbreak attacks remains challenging, as evidenced by systematic evaluations (Mazeika et al., 2024; Chao et al., 2024). For detection-based defenses, a key requirement is to generalize beyond familiar prompt patterns to jailbreak attacks unseen during detector training.

Internal representations offer evidence about how an LLM processes a request beyond its surface wording. Studies of refusal and activated concepts establish their relevance to safety (Arditi et al., 2024; Zhang et al., 2025; Pan et al., 2025; Wollschläger et al., 2025; Zhao et al., 2025). Recent defenses characterize the relational structure of hidden-state evolution for jailbreak detection (Zhang et al., 2026; Dong et al., 2026). However, their detection criteria rely on descriptions of the observed evolution, without explicitly learning how to predict successor states from the available context. This distinction matters for generalization: an unfamiliar attack can produce a different trajectory while retaining predictive regularities shared with known jailbreaks. Learning these transition relation-

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.

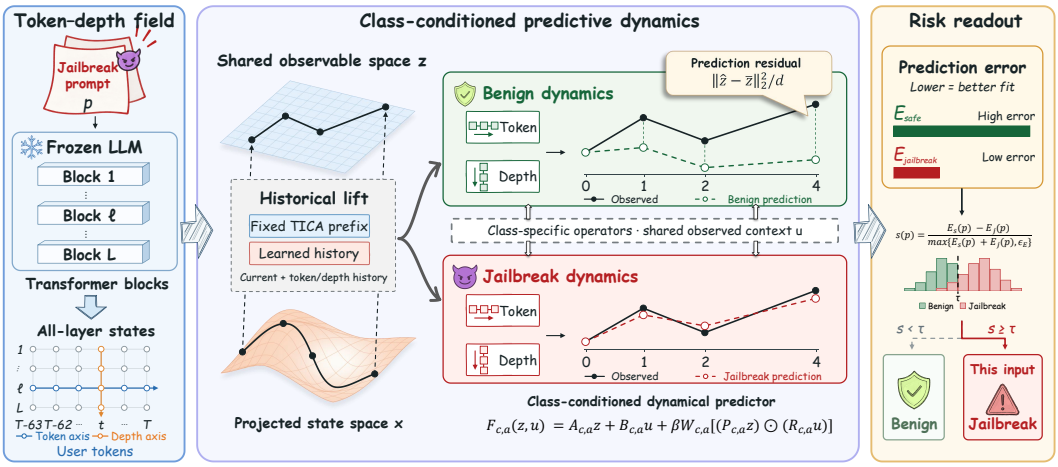


Figure 1: **CORD. (a) Token–depth field (left):** hidden states from the final 64 user tokens. **(b) Predictive dynamics (middle):** shared observables support benign and jailbreak Koopman predictions under identical context. **(c) Risk readout (right):** aggregated residuals yield higher risk when jailbreak dynamics better predict the same successors. Curves and bars are schematic.

ships could therefore reveal commonalities across attacks whose prompt constructions and observed trajectories differ, providing an additional source of evidence for detecting unseen jailbreaks.

We hypothesize that benign and jailbreak inputs exhibit distinguishable predictive regularities in their hidden-state evolution. An unseen attack may follow regularities shared with known jailbreaks even when its particular trajectory is unfamiliar. However, benign-model prediction error alone may also reflect unfamiliar legitimate content. We therefore compare both dynamical models against the same observed successors. The key question is whether jailbreak dynamics predict the observed evolution more accurately than benign dynamics. Their relative predictive fit turns this hypothesis into a detection criterion.

Building on this principle, we introduce **CORD** (**C**ompeting **O**perator **R**esidual **D**etection), a framework that detects jailbreaks by comparing class-conditioned predictions of hidden-state evolution (Figure 1). Inspired by prior work (Li et al., 2026; Liu et al., 2023), CORD models a frozen LLM's prompt hidden states using conditional Koopman operators. A shared observable map provides common prediction coordinates, in which context-dependent linear operators approximate benign and jailbreak dynamics across token positions and network layers (Williams et al., 2015; Korda & Mezić, 2018; Lusch et al., 2018). Both dynamical models forecast the same successors under identical observed context. Their accumulated prediction errors form a normalized contrast that assigns higher risk when the jailbreak model better predicts the observed evolution.To evaluate CORD, we conduct extensive experiments across four representative LLMs and ten jailbreak attack families. CORD achieves 93.94% macro TPR@1, compared with 55.77% for the strongest evaluated baseline, and leads the average detection results for every model and attack family. Together, these results support class-relative predictive modeling as a promising basis for detecting jailbreak attacks beyond familiar prompt constructions.In summary,Our contributions are as follows:

- we introduce a predictive perspective on jailbreak detection, treating the relative ability of benign and jailbreak dynamics to explain hidden-state evolution as detection evidence. This perspective targets predictive regularities that can remain shared across attacks despite differences in their observed trajectories.
- we present CORD, the first jailbreak detection framework based on dual-axis Koopman predictive contrast.By evaluating competing benign and jailbreak models against the same observed successors, CORD turns differences in predictive accuracy into a unified detection criterion.
- We conduct extensive experiments, demonstrating that CORD outperforms the evaluated defenses. These results demonstrate strong generalization to attack families excluded from detector training, highlighting CORD's ability to detect jailbreaks beyond familiar attack constructions.

2 PRELIMINARY

Given a prompt p, we seek to distinguish benign inputs ($C = \mathrm{s}$) from jailbreak attempts ($C = \mathrm{j}$), including attack families absent from detector training. The detector observes hidden states produced by a frozen LLM during prompt processing and assigns a prompt-level risk score $s(p)$. Our objective is to learn predictive relationships between these states and use their relative agreement with benign and jailbreak dynamics as detection evidence.

2.1 LLMS AS CONTROLLED DYNAMICAL SYSTEMS

Hidden-state evolution has two ordered directions: successive Transformer blocks transform representations across depth, while token positions organize contextualized representations along the prompt. We describe these successors using a full computational state and expose the context needed for prediction through a separate readout. For axis $a \in \{\mathrm{tok}, \mathrm{dep}\}$ and step k,

$$\mathsf{q}_{k+1} = \mathcal{T}_a(\mathsf{q}_k), \qquad u_k = \nu_a(\mathsf{q}_k), \tag{1}$$

The full state q_k includes the prompt context and current position, making the successor map autonomous. The observed control u_k supplies token or layer context to predictors operating on finite observations.

Assumption 1 (Full-state realization)**.** *For a fixed LLM and prompt, each successor map $\mathcal{T}_a$ is deterministic on valid field positions. Full-state readouts determine the retained histories and successor observations.*

The detector accesses a finite history of hidden states rather than the full computational state. Prediction therefore depends on how much relevant context this history retains, motivating a conditional description of the observed evolution.

2.2 TOKEN–DEPTH FIELD

Let $h_{\ell,t} \in \mathbb{R}^{d_h}$ denote the hidden state after block ℓ at retained user-token position t. A causal prefill produces the indexed field

$$\mathcal{H}(p) = \{h_{\ell,t} : 1 \le \ell \le L,\ 1 \le t \le T\}. \tag{2}$$

The token and depth successors of (ℓ, t) are $(\ell, t+1)$ and $(\ell+1, t)$, respectively, with out-of-grid endpoints excluded. We construct prediction tasks within this observed field and evaluate forecasts against the corresponding recorded successors.

A common readout $\xi = \pi(\mathsf{q})$ collects local history from both directions. A shared lift $z = \phi_\theta(\xi) \in \mathbb{R}^d$ then defines the prediction coordinates. Both class hypotheses use the same history and observable map, placing their forecasts and targets in a common space. Section 3.1 specifies how these observables are constructed.

2.3 KOOPMAN OPERATORS

Koopman operators describe state evolution through linear actions on observable functions (Koopman, 1931; Williams et al., 2015). For a full-state observable φ, the action associated with $\mathcal{T}_a$ is

$$(\mathcal{K}_a^{\mathrm{full}}\varphi)(\mathsf{q}) = \varphi(\mathcal{T}_a(\mathsf{q})). \tag{3}$$

Benign and jailbreak inputs share the same full-state transition rule. Their reduced predictive laws can nevertheless differ because the two prompt populations induce different distributions over context omitted from the retained history. Write $\Xi = \pi(\mathsf{q})$, $U_a = \nu_a(\mathsf{q})$, and $\Xi_a^+ = \pi(\mathcal{T}_a(\mathsf{q}))$. For class $c \in \{\mathrm{s}, \mathrm{j}\}$, projecting the full-state action onto the available history and control gives

$$(\mathcal{K}_{c,a}^u\psi)(\xi) = \mathbb{E}[\psi(\Xi_a^+) \mid \Xi = \xi, U_a = u, C = c]. \tag{4}$$

The expectation averages over the prompt population at a fixed valid grid and source position, whose conditioning is suppressed in the notation. This action is linear in the observable ψ (Wu & Noé, 2020; Korda & Mezić, 2018). Applied to the shared lift ϕ_θ, it describes the expected successor coordinates under each class hypothesis. Finite-dimensional predictors approximate these conditional actions.

Assumption 2 (Finite observable approximation). *Fix the shared lift $\phi = \phi_\theta$ and the observable space $\mathcal{V} = \mathrm{span}\{1, \phi_1, \ldots, \phi_d\}$. The fitted affine predictor $F_{c,a}(z, u) = M_{c,a}(u)z + \mathbf{b}_{c,a}(u)$ has closure residual $\mathbf{r}_{c,a}(\xi, u) = (\mathcal{K}^u_{c,a}\phi)(\xi) - F_{c,a}(\phi(\xi), u)$. Its root-mean-square norm is bounded by $\delta_{c,a}$ under each task-conditioned source distribution used in the prediction analysis.*

The closure residual quantifies finite-dimensional approximation error; zero residual gives conditional invariance of $\mathcal{V}$ on the specified support. Finite history also introduces memory effects that influence repeated prediction (Lin & Lu, 2021). These quantities connect the observable representation to the quality of its forecasts.

The resulting detection problem is to learn benign and jailbreak predictors in this shared space and determine which better explains the observed successors. The next section develops this comparison through conditional operators and a residual-based detection score.

3 METHODOLOGY

We propose **CORD**, a jailbreak detector that compares class-conditioned Koopman predictions of hidden-state evolution. A shared history lift defines common observables for modeling token and depth transitions. Benign and jailbreak operators predict the same successors under identical context; their residual contrast identifies which class better explains the observed evolution.

3.1 SHARED OBSERVABLES

The shared lift defines the finite observable space used to approximate the conditional Koopman actions. We obtain standardized coordinates $x_{\ell,t} \in \mathbb{R}^{d_x}$ through PCA whitening and time-lagged independent component analysis (TICA), pooling token and depth transition pairs (Pérez-Hernández et al., 2013). The lift combines a fixed coordinate prefix with learned history observables (Kamb et al., 2020; Lusch et al., 2018):

$$\xi_{\ell,t} = [x_{\ell,t}; x_{\ell,t-1}; x_{\ell-1,t}], \qquad z_{\ell,t} = \phi_\theta(\xi_{\ell,t}) = \begin{bmatrix} P_{\mathrm{id}}^{\mathsf{T}} x_{\ell,t} \\ f_\theta(\xi_{\ell,t}) \end{bmatrix} \in \mathbb{R}^d. \tag{5}$$

Here P_{id} selects the fixed TICA prefix, anchoring prediction to a common observational reference. The learned branch f_θ incorporates neighboring states through $\xi_{\ell,t} \in \mathbb{R}^{3d_x}$ to capture predictive context. Together with the constant observable, the lift spans $\mathcal{V}$ in Assumption 2. Sharing this space across classes makes prediction errors directly comparable; out-of-grid history is zero-padded.

3.2 CONDITIONAL KOOPMAN OPERATORS

For each class and axis, we approximate the conditional Koopman action in Equation (4) by an affine predictor on the shared observables. Token identity and layer position supply the observed context u, yielding four class–axis predictors of the form:

$$F_{c,a}(z, u) = A_{c,a}z + B_{c,a}u + \beta W_{c,a}\big[(P_{c,a}z) \odot (R_{c,a}u)\big]. \tag{6}$$

The term Az describes the base evolution, Bu supplies a contextual displacement, and the low-rank bilinear term lets context modulate the transition matrix. Define $M_{c,a}(u) = A_{c,a} + \beta W_{c,a}\,\mathrm{diag}(R_{c,a}u)P_{c,a}$. Appending a constant observable yields

$$\widehat{K}_{c,a}(u) = \begin{bmatrix} M_{c,a}(u) & B_{c,a}u \\ 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} F_{c,a}(z,u) \\ 1 \end{bmatrix} = \widehat{K}_{c,a}(u) \begin{bmatrix} z \\ 1 \end{bmatrix}. \tag{7}$$

Thus $\widehat{K}_{c,a}(u)$ approximates the conditional Koopman action on $\mathcal{V}$, with closure residual defined in Assumption 2. Composing these matrices along the observed controls generates multi-horizon forecasts from a single learned evolution law. Both classes receive the same controls: source-token identity for token steps and layer position for depth steps.

To control how local errors accumulate during recursive prediction, we constrain the base transition A through a normal parameterization:

$$A = Q^{\mathsf{T}} D Q, \qquad D_j = \rho_j \begin{bmatrix} \cos\vartheta_j & -\sin\vartheta_j \\ \sin\vartheta_j & \cos\vartheta_j \end{bmatrix}. \tag{8}$$

Here D is block diagonal, $0 < \rho_{\min} \le \rho_j \le \rho_{\max} < 1$, and $Q^{\mathsf{T}}Q = I$. Orthogonal similarity gives $\|A\|_2 = \rho(A) \le \rho_{\max}$, bounding the base transition's amplification of local error. Lemma 1 incorporates this bound and the bilinear contribution into propagation through $M(u)$. Appendix B.1 specifies the control encodings and orthogonal parameterization.

3.3 Predictive contrast

Both class models are evaluated against the same observed successors, using a common initial state and control sequence. For axis a and start $v = (\ell, t)$, let z_r denote the observable at $(\ell, t + r)$ for token prediction or $(\ell + r, t)$ for depth prediction. Starting from $\widehat{z}_{c,0} = z_0$, iterate $\widehat{z}_{c,r+1} = F_{c,a}(\widehat{z}_{c,r}, u_r)$. The horizon-$h$ error is

$$e^{(h)}_{c,a,v} = \frac{\|\widehat{z}_{c,h} - z_h\|_2^2}{d}. \tag{9}$$

Averaging the errors over valid starts and horizons within layer ℓ gives $E^{\mathrm{tok}}_{c,\ell}$; averaging within token column t gives $E^{\mathrm{dep}}_{c,t}$. A shared depth gate pools token errors across layers, while depth errors are averaged across columns:

$$E^{\mathrm{tok}}_c = \sum_{\ell=1}^{L} \varpi_\ell E^{\mathrm{tok}}_{c,\ell}, \qquad E^{\mathrm{dep}}_c = \frac{1}{T}\sum_{t=1}^{T} E^{\mathrm{dep}}_{c,t}. \tag{10}$$

Here $\varpi_\ell = \exp(g^{\mathsf{T}} u^{\mathrm{dep}}_\ell) / \sum_{j=1}^{L} \exp(g^{\mathsf{T}} u^{\mathrm{dep}}_j)$ for learned g; Appendix F gives the finite-grid averages. With a model-specific weight $w \in [0, 1]$ shared by both classes, $E_c = wE^{\mathrm{tok}}_c + (1 - w)E^{\mathrm{dep}}_c$. The score is

$$s(p) = \frac{E_{\mathrm{s}}(p) - E_{\mathrm{j}}(p)}{\max\{E_{\mathrm{s}}(p) + E_{\mathrm{j}}(p), \epsilon_E\}}, \tag{11}$$

where $\epsilon_E > 0$ is a normalization floor. Positive scores favor the jailbreak predictor; the denominator expresses the contrast relative to total prediction error. For target FPR α, the decision is $\delta_\alpha(p) = \mathbf{1}\{s(p) \ge \tau_\alpha\}$. The following result characterizes the expected energy difference underlying this score.

Fix the fitted observables and shared pooling. Let f_c collect class-c forecasts over valid axes, starts, and horizons, and let μ_y collect the corresponding class-y conditional successor means. Write $\bar{y}$ for the other class and $\|\cdot\|_y$ for the energy-weighted L^2 norm (Appendix A.1).

Proposition 1 (Relative predictive fit)**.** *For square-integrable targets and forecasts on common inputs, with the same input-measurable weights for both hypotheses, the expected residual contrast is*

$$\Gamma_y := \mathbb{E}[E_{\bar{y}} - E_y \mid C = y] = \|\mu_y - f_{\bar{y}}\|_y^2 - \|\mu_y - f_y\|_y^2. \tag{12}$$

Shared target variance cancels, so the contrast measures relative proximity to the class-conditional successor mean. Define forecast separation $\mathcal{S}_y = \|f_y - f_{\bar{y}}\|_y$ and own-class bias $\mathcal{B}_y = \|\mu_y - f_y\|_y$. Lemma 2 bounds this bias by $\mathcal{U}_y$, which pools Koopman closure and finite-history errors propagated through the conditional matrices. Consequently,

$$\Gamma_y \ge \mathcal{S}_y(\mathcal{S}_y - 2\mathcal{B}_y) \ge \mathcal{S}_y(\mathcal{S}_y - 2\mathcal{U}_y). \tag{13}$$

Thus $\mathcal{S}_y > 2\mathcal{U}_y$ yields a correct-class residual advantage in expectation. History and context address prediction ambiguity; the spectral structure controls base error propagation. Proposition 2 extends the criterion to held-out-family shift. The learning objective therefore couples own-class prediction accuracy with the residual margin between competing models.

3.4 Learning objective

For a batch of N_b prompts (p_n, y_n), let $E^{\mathrm{tr}}_{n,c}$ use the shared pooling with training weight w_{tr} and training targets in place of z_h in Equation (9). We fit the Koopman predictors through own-class forecasting and classification by competing energies:

$$\mathcal{L}_{\mathrm{roll}} = \frac{1}{N_b}\sum_{n=1}^{N_b} E^{\mathrm{tr}}_{n,y_n}, \qquad \mathcal{L}_{\mathrm{CE}} = -\frac{1}{N_b}\sum_{n=1}^{N_b} \log \frac{\exp(-E^{\mathrm{tr}}_{n,y_n}/\tau_E)}{\sum_c \exp(-E^{\mathrm{tr}}_{n,c}/\tau_E)}. \tag{14}$$

For fixed targets and pooling, minimizing own-class prediction risk reduces conditional-mean bias. With $r_n^{\mathrm{tr}} = E_{n,\bar{y}_n}^{\mathrm{tr}} - E_{n,y_n}^{\mathrm{tr}}$, each classification term equals $\log(1 + \exp(-r_n^{\mathrm{tr}}/\tau_E))$ and encourages a positive residual margin. The two losses therefore promote accurate prediction and relative predictive advantage.

Small state changes can give an unchanged-state predictor low error. Let Π_n be the prediction energy of repeating the initial observable at every horizon, with the same targets and pooling. To encourage gains over persistence, use floor $\epsilon_{\mathrm{pers}} > 0$ and target ratio $\eta_{\mathrm{p}} \in (0, 1)$:

$$\mathcal{L}_{\mathrm{pers}} = \frac{1}{N_b} \sum_n \left[\frac{E_{n,y_n}^{\mathrm{tr}}}{\max(\Pi_n, \epsilon_{\mathrm{pers}})} - \eta_{\mathrm{p}} \right]_+ . \tag{15}$$

This penalty encourages prediction gains over persistence after pooling both axes. The complete objective is

$$\mathcal{J} = \lambda_{\mathrm{roll}} \mathcal{L}_{\mathrm{roll}} + \lambda_{\mathrm{CE}} \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{pers}} \mathcal{L}_{\mathrm{pers}}. \tag{16}$$

Training first jointly learns the lift and operators, then fixes the lift to refine the operators and pooling in a fixed observable space. Algorithm 2 specifies stage-dependent targets and updates. The analysis evaluates the final fitted predictors using inference targets and readout weights; Appendix A.6 relates the fitted metric to training margins.

4 Experiments

We evaluate CORD on four language models and ten attack families, testing detection across attack families and the contributions of predictive comparison, dual-axis modeling, and operator design. Hard-benign and order-perturbation analyses examine the detection signal; serving cost appears in Appendix G.

4.1 Evaluation setup

Benchmarks. We consider Qwen2.5-7B (Qwen et al., 2024), Llama3-8B-Instruct (Grattafiori et al., 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Llama2-7B-Chat (Touvron et al., 2023). The ten attack cohorts are AutoDAN (Liu et al., 2024b), Base64 (Wei et al., 2023), DrAttack (Li et al., 2024), GCG (Zou et al., 2023), GPTFuzz (Yu et al., 2023), IJP (Shen et al., 2024), JailJudge (Liu et al., 2024a), PAIR (Chao et al., 2025), SAA (Andriushchenko et al., 2025), and Zulu (Yong et al., 2023). Ordinary-safe prompts come from UltraChat (Ding et al., 2023), OASST (Köpf et al., 2023), and Dolly (Conover et al., 2023). Models share prompt IDs; training, threshold selection, and evaluation use disjoint data. Appendix C details the sources.

CORD uses the final 64 user tokens in ten family-excluded fitting folds per model. Each fold retains the PCA and refits TICA and the detector on 9×300 seen-family attacks and 2,700 safe prompts; testing uses 300 target-family attacks and 300 safe prompts. One readout weight is shared across each model's ten folds (Appendix C). Analytical studies use the same 64-token window and model-level readout weights, except for the single-trajectory controls described below. Macro averages weight models and families equally.

Evaluation criteria. TPR@1 uses a threshold set at 1% FPR on 1,000 independent benign prompts. Figure 5 also reports TPR@5. Reported gains are absolute rate differences. The baselines are MTK (Zhang et al., 2026), JBShield-D (Zhang et al., 2025), GraphShield (Dong et al., 2026), and GradSafe (Xie et al., 2024).

4.2 Main result

CORD achieves 93.94% macro detection, exceeding GraphShield's 55.77% by 38.17% (Table 1). It leads all four model averages and all ten attack-family averages; model means range from 92.23% to 95.63%. Detection reaches 100.00% on Base64, SAA, and Zulu for every model. Across models, AutoDAN reaches 96.50% versus 75.83% for its strongest baseline, MTK; GCG reaches 98.75% versus 67.08% for MTK. These advantages span encoded, optimization-based, and automated prompt constructions. CORD compares how well the two dynamical models explain the same transitions over a 64-token window, making relative predictive fit the detection criterion. Its

Table 1: **Detection across ten attack families.** TPR@1 (%); **bold**/underline mark one best/second-best distinct value per model and column. avg averages the ten attacks.

Model	TPR@1 (%) ↑										avg
	AutoDAN	Base64	DrAttack	GCG	GPTFuzz	IJP	JailJudge	PAIR	SAA	Zulu	
Llama3-8B-Instruct											
MTK	52.00	0.00	87.67	21.33	98.00	82.00	13.67	54.00	0.00	0.33	40.90
JBShield-D	7.67	0.00	27.00	25.33	54.33	51.33	29.67	23.00	0.00	0.33	21.87
GraphShield	97.33	0.00	89.33	48.33	95.00	86.33	41.00	52.00	1.00	97.67	60.80
GradSafe	96.67	0.00	30.67	49.00	92.67	67.00	51.33	70.33	0.00	0.00	45.77
CORD (Ours)	**99.67**	**100.00**	**100.00**	**99.00**	**99.67**	**99.67**	**85.33**	**73.00**	**100.00**	**100.00**	**95.63**
Qwen2.5-7B											
MTK	90.67	0.00	30.33	84.33	86.33	61.33	**79.33**	55.33	0.00	12.00	49.97
JBShield-D	55.33	0.00	50.67	94.33	77.67	94.67	47.33	30.00	99.00	0.33	54.93
GraphShield	80.00	30.00	50.00	94.33	91.33	87.00	25.33	39.67	100.00	51.33	64.90
GradSafe	80.00	0.00	0.67	41.67	17.67	9.33	9.00	37.67	0.00	0.00	19.60
CORD (Ours)	**99.00**	**100.00**	**90.67**	**97.00**	**99.67**	**98.00**	74.33	**63.67**	**100.00**	**100.00**	**92.23**
Mistral-7B-Instruct-v0.2											
MTK	62.33	0.33	73.33	90.33	50.33	41.00	48.67	69.00	0.00	99.00	53.43
JBShield-D	66.67	100.00	73.00	80.00	58.67	21.00	66.33	32.33	0.00	84.33	58.23
GraphShield	19.33	0.00	2.67	30.67	55.67	41.00	15.33	3.00	73.00	96.33	33.70
GradSafe	11.67	0.00	48.67	99.33	91.33	66.00	38.33	77.67	0.00	1.00	43.40
CORD (Ours)	**97.67**	**100.00**	**82.67**	**99.33**	**99.67**	**99.00**	**80.33**	**87.67**	**100.00**	**100.00**	**94.63**
Llama2-7B-Chat											
MTK	**98.33**	0.00	63.33	72.33	93.33	96.00	53.00	**77.67**	0.00	3.67	55.77
JBShield-D	7.00	0.00	60.67	0.67	63.67	48.33	20.67	12.00	0.00	2.00	21.50
GraphShield	98.00	32.33	57.33	84.67	91.00	83.67	54.00	56.00	0.00	80.00	63.70
GradSafe	97.00	0.00	36.33	64.67	84.00	81.00	37.00	66.33	0.00	1.00	46.73
CORD (Ours)	89.67	**100.00**	**88.33**	**99.67**	**98.67**	**99.67**	**79.33**	77.33	**100.00**	**100.00**	**93.27**

performance across these varied constructions supports this criterion as evidence beyond familiar prompt forms.

4.3 Predictive comparison

With matched 64-token windows, joint TICA inputs, and the same causal-history architecture, CORD reaches 93.94% macro TPR@1 versus 74.40% for direct classification and 87.21% for a one-step spectral operator (Table 7). The 19.54% gain over direct classification supports successor-based comparison beyond discrimination from the same inputs. Multiple horizons add 6.73% overall, with substantial gains on three models and nearly equal Llama2 averages. Thus predictive comparison improves detection across models, while the benefit of longer forecasts varies.

Across all ten Llama3 held-out families, benign and attack medians fall on opposite sides of their thresholds (Figure 2). Median gaps range from 0.36 (PAIR) to 0.76 (Zulu). Appendix H shows the other three models.

4.4 Dual-axis benefit

Joint modeling reaches 93.94% macro TPR@1, compared with 80.25% for token-only and 41.53% for depth-only prediction (Figure 3). Token-only follows the 64-token window at the middle transformer layer; depth-only follows the last valid input token across layers. Both are independently trained on their respective trajectories. Token-only is stronger on all four models, yet the complete field improves detection by a further 5.43–23.67%. This comparison supports retaining the token–depth field rather than reducing it to either representative trajectory.

On Llama3–GPTFuzz, the cohort-mean contrast extends across token positions and is strongest in intermediate layers (Figure 4; 300 attacks and 300 benign prompts). The axis profiles show unfused token and depth differences, locating predictive evidence within the field and complementing the cross-model single-axis comparison.

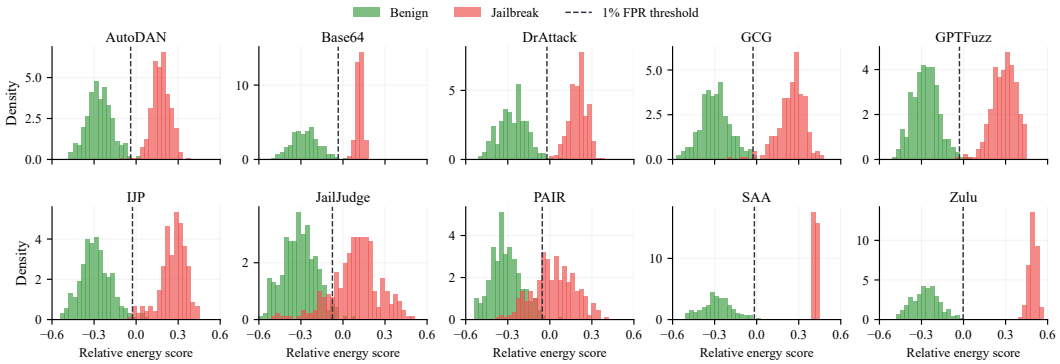


Figure 2: **Llama3 held-out-family score distributions.** Each panel shows 300 benign and 300 attack evaluation prompts. Dashed lines mark thresholds corresponding to 1% FPR. Larger scores favor the jailbreak hypothesis.

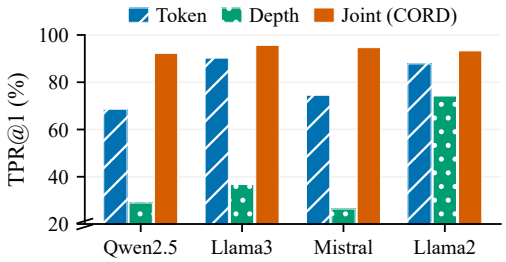


Figure 3: **Single-axis retraining.** TPR@1 averaged over ten attack families per model.

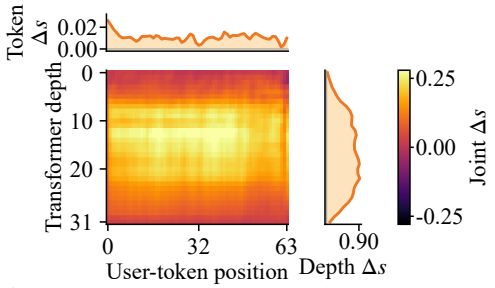


Figure 4: **Local contrast.** Llama3–GPTFuzz joint one-step score difference.

4.5 Ablation Study

Table 2 tests three components on Llama2–PAIR: (a) context input (token/layer conditioning), (b) spectral structure (rotation–scaling A), and (c) dynamics fitting (prediction and persistence losses). Checks indicate retained components. Full CORD reaches 77.33%. Removing (b), (a), and (c) lowers detection by 30.00%, 28.33%, and 25.00%, respectively. Context and dynamics fitting support prediction, while spectral structure bounds the autonomous contribution to error propagation (Lemma 1). All three components contribute in this case.

Table 2: **Component ablations.**

Components			Detection
(a)	(b)	(c)	TPR@1 ↑
	✓	✓	49.00
✓		✓	47.33
✓	✓		52.33
✓	✓	✓	**77.33**

4.6 Order Sensitivity

We change token–depth order while retaining the sampled hidden vectors. On Qwen2.5–JailJudge, TPR@1 falls from 74.33% to 14.67% under joint reversal and to 18.11% when averaging three joint shuffles (Figure 5); TPR@5 follows the same trend. This sensitivity shows that ordering contributes to the complete pipeline, which includes history and control alignment. The arrangement of states therefore supplies evidence beyond the retained vectors themselves, consistent with comparing predictions of ordered successors.

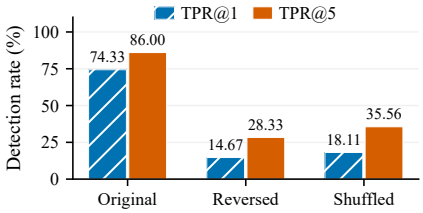


Figure 5: **Order perturbations** on Qwen2.5–JailJudge. shuffle bars average three joint permutations.

4.7 Hard-Benign Discrimination

On challenging legitimate requests from OKTest (Shi et al., 2024) and XSTest-safe (Röttger et al., 2024), CORD reaches 87.39% and 88.73% macro detection at 1% FPR, versus 48.25% and 61.09%

for GraphShield (Table 3). It leads in seven of eight model–negative-set combinations, extending its aggregate advantage beyond ordinary-safe inputs.

Table 3: **Hard-benign evaluation.** Macro TPR@1 (%) over 40 model–family cells.

Negative set	CORD	MTK	JBShield-D	GraphShield	GradSafe
OKTest	**87.39**	30.18	9.14	48.25	25.33
XSTest-safe	**88.73**	30.27	9.99	61.09	7.85

5 Related Work

Jailbreak attacks. Jailbreaks preserve a harmful objective while changing how it is presented to an aligned model. Optimization-based attacks search for transferable adversarial suffixes (Zou et al., 2023), fluent evolutionary prompts (Liu et al., 2024b), or mutated templates (Yu et al., 2023). Black-box methods use model feedback to revise candidates: PAIR iteratively improves prompts, whereas TAP explores and prunes a tree of attacks (Chao et al., 2025; Mehrotra et al., 2024). These procedures generate distinct prompt distributions even when they pursue the same underlying request. Semantic attacks instead reorganize the request through decomposition (Li et al., 2024), nested role-playing contexts (Li et al., 2023), or persuasive framing (Zeng et al., 2024). Their diversity motivates evaluation across attack constructions. HarmBench and JailbreakBench standardize behaviors, threat models, and scoring, while JAILJUDGE studies jailbreak assessment across diverse cases (Mazeika et al., 2024; Chao et al., 2024; Liu et al., 2024a). These variations motivate our held-out-family evaluation: the detection criterion must transfer across constructions of harmful requests.

Jailbreak defenses. Defenses can assess the input, modify generation, or alter internal representations. WildGuard models harmfulness and refusal (Han et al., 2024); SmoothLLM aggregates responses to perturbed inputs (Robey et al., 2023); erase-and-check certifies robustness within bounded token-edit threat models (Kumar et al., 2023). Goal prioritization reinforces safety over conflicting instructions, and SafeDecoding adjusts token probabilities toward safe responses (Zhang et al., 2024; Xu et al., 2024). These approaches obtain safety evidence or intervene at different stages of processing. Representation-based work examines how harmful behavior and refusal are encoded. Refusal-direction analysis identifies a low-dimensional mechanism, and circuit breakers redirect representations associated with harmful outputs (Arditi et al., 2024; Zou et al., 2024). Internal detectors use activated concepts (Zhang et al., 2025), safety-critical gradients (Xie et al., 2024), or token–layer graphs (Dong et al., 2026). CORD asks how those internal observations can support a predictive comparison. Its two class-conditioned operators explain common token–depth successors, making relative prediction error the detection criterion. The matched-input experiments test this criterion while holding access to the internal field fixed.

6 Conclusion

We presented CORD, a framework that detects unseen jailbreaks by learning and contrasting class-specific predictive regularities in hidden-state evolution. A conditional Koopman formulation models benign and jailbreak dynamics on shared observables across token progression and network depth. Both models forecast the same successors under identical observed context; their residual contrast assigns higher risk when jailbreak dynamics better explain the observed evolution. Our analysis relates forecast separation to accumulated prediction error, establishing sufficient conditions for a correct-class residual advantage in expectation. Across four LLMs and ten attack families, CORD achieves 93.94% macro TPR@1, compared with 55.77% for the strongest evaluated baseline. Matched-input comparisons support the value of predictive modeling, while joint-axis modeling outperforms either single-axis alternative on every model. Together, these analyses demonstrate that predictive contrast captures discriminative differences between benign and jailbreak hidden-state dynamics, providing a promising basis for detecting unseen attacks beyond familiar prompt constructions.

AI USE STATEMENT

Generative AI tools (Codex) were used during the preparation of this manuscript to assist with language editing and polishing, literature retrieval and discovery (e.g., identifying potentially relevant related work), drafting portions of the manuscript, and assisting with the implementation of experimental code. All AI-assisted content and code were reviewed, verified, and revised by the authors as appropriate. The authors take full responsibility for the accuracy, originality, citations, experimental implementation, and final content of the manuscript.

ETHICS STATEMENT

This work studies detection of attempts to bypass language-model safeguards. False positives can restrict legitimate use, while false negatives permit missed attacks. Detection should be interpreted at its stated score threshold. The choice of threshold depends on the application and the consequences of rejecting a legitimate request.

REPRODUCIBILITY STATEMENT

Section 3 specifies CORD's architecture, prediction criterion, and training objective. Appendix A provides the theoretical derivations, while Appendix B details preprocessing, model parameterization, and training and inference algorithms. Appendix C describes the datasets and fixed-configuration leave-one-attack-family-out protocol, including data splits and model-specific settings. The remaining appendices report additional experimental results and serving costs. Our implementation is available in the anonymous repository.

REFERENCES

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned LLMs with simple adaptive attacks. In *International Conference on Learning Representations*, pp. 40116–40143, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/63fa7efdd3bcf944a4bd6e0ff6a50041-Paper-Conference.pdf.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 136037–136083, 2024. doi: 10.52202/079017-4322.

Erik Brynjolfsson, Danielle Li, and Lindsey Raymond. Generative AI at work. *The Quarterly Journal of Economics*, 140(2):889–942, 2025. doi: 10.1093/qje/qjae044.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 55005–55029, 2024. doi: 10.52202/079017-1745.

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, 2025. doi: 10.1109/satml64287.2025.00010.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free Dolly: Introducing the world's first truly open instruction-tuned LLM. Databricks blog, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051, 2023. doi: 10.18653/v1/2023.emnlp-main.183.

Sunghee Dong, Sungwon Yi, Kangmin Bae, Jaeyoon Kim, and Seongyeop Kim. GraphShield: Graph-theoretic modeling of network-level dynamics for robust jailbreak detection. In *International Conference on Learning Representations*, pp. 107073–107099, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/aec77a46e938f6c61c0c3b30562162d4-Paper-Conference.pdf.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8093–8131, 2024. doi: 10.52202/079017-0261.

Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. Improved techniques for optimization-based jailbreaking on large language models. In *International Conference on Learning Representations*, pp. 6337–6358, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/124256ed80af5d4bf4c4de17b66c4298-Paper-Conference.pdf.

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.

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*, volume 37, pp. 47094–47165, 2024. doi: 10.52202/079017-1493.

Mason Kamb, Eurika Kaiser, Steven L. Brunton, and J. Nathan Kutz. Time-delay observables for Koopman: Theory and applications. *SIAM Journal on Applied Dynamical Systems*, 19(2):886–917, 2020. doi: 10.1137/18m1216572.

B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931. doi: 10.1073/pnas.17.5.315.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. OpenAssistant conversations - democratizing large language model alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 47669–47681, 2023. doi: 10.52202/075280-2064.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018. doi: 10.1016/j.automatica.2018.03.046.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. Certifying LLM safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Jiaqi Li, Xinglong Zhang, Haibin Xie, Yixing Lan, Wei Pan, and Xin Xu. Koopman Dreamer: Spectrally constrained latent dynamics for stable world-model imagination. *arXiv preprint arXiv:2607.19719*, 2026.

Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. DrAttack: Prompt decomposition and reconstruction makes powerful LLMs jailbreakers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13891–13913, 2024. doi: 10.18653/v1/2024.findings-emnlp.813.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. DeepInception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*, 2023.

Kevin K. Lin and Fei Lu. Data-driven model reduction, Wiener projections, and the Koopman-Mori-Zwanzig formalism. *Journal of Computational Physics*, 424:109864, 2021. doi: 10.1016/j.jcp.2020.109864.

Fan Liu, Yue Feng, Zhao Xu, Lixin Su, Xinyu Ma, Dawei Yin, and Hao Liu. JAILJUDGE: A comprehensive jailbreak judge benchmark with multi-agent enhanced explanation evaluation framework. *arXiv preprint arXiv:2410.12855*, 2024a.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/pdf?id=7Jwpw4qKkb.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with Koopman predictors. In *Advances in Neural Information Processing Systems*, volume 36, pp. 12271–12290, 2023. doi: 10.52202/075280-0538.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224, 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. In *Advances in Neural Information Processing Systems*, volume 37, pp. 61065–61105, 2024. doi: 10.52202/079017-1952.

Wenbo Pan, Zhichao Liu, Qiguang Chen, Xiangyang Zhou, Yu Haining, and Xiaohua Jia. The hidden dimensions of LLM alignment: A multi-dimensional analysis of orthogonal safety directions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47697–47716, 2025. URL https://proceedings.mlr.press/v267/pan25f.html.

Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni de Fabritiis, and Frank Noé. Identification of slow molecular order parameters for Markov model construction. *The Journal of Chemical Physics*, 139(1):015102, 2013. doi: 10.1063/1.4811489.

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. *arXiv preprint arXiv:2412.15115*, 2024.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. SmoothLLM: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 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. doi: 10.18653/v1/2024.naacl-long.301.

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 *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024.

Chenyu Shi, Xiao Wang, Qiming Ge, Songyang Gao, Xianjun Yang, Tao Gui, Qi Zhang, Xuanjing Huang, Xun Zhao, and Dahua Lin. Navigating the OverKill in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4602–4614, 2024. doi: 10.18653/v1/2024.acl-long.253.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tao Tu, Mike Schaekermann, Anil Palepu, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Yong Cheng, Elahe Vedadi, Nenad Tomasev, Shekoofeh Azizi, Karan Singhal, Le Hou, Albert Webson, Kavita Kulkarni, S. Sara Mahdavi, Christopher Semturs, Juraj Gottweis, Joelle Barral, Katherine Chou, Greg S. Corrado, Yossi Matias, Alan Karthikesalingam, and Vivek Natarajan. Towards conversational diagnostic artificial intelligence. *Nature*, 642(8067): 442–450, 2025. doi: 10.1038/s41586-025-08866-7.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Daniel Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations*, pp. 65882–65919, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a4b6ad6b48850c0c331d1259fc66a69c-Paper-Conference.pdf.

Xunguang Wang, Zhenlan Ji, Wenxuan Wang, Zongjie Li, Daoyuan Wu, and Shuai Wang. SoK: Evaluating jailbreak guardrails for large language models. In *IEEE Symposium on Security and Privacy (SP)*, 2026.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems*, volume 36, pp. 80079–80110, 2023. doi: 10.52202/075280-3508.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015. doi: 10.1007/s00332-015-9258-5.

Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66945–66970, 2025. URL https://proceedings.mlr.press/v267/wollschlager25a.html.

Hao Wu and Frank Noé. Variational approach for learning Markov processes from time series data. *Journal of Nonlinear Science*, 30:23–66, 2020. doi: 10.1007/s00332-019-09567-y.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. GradSafe: Detecting jailbreak prompts for LLMs via safety-critical gradient analysis. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 507–518, 2024. doi: 10.18653/v1/2024.acl-long.30.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5587–5605, 2024. doi: 10.18653/v1/2024.acl-long.303.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, pp. 50528–50652, 2024. doi: 10.52202/079017-1601.

Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. Low-resource languages jailbreak GPT-4. *arXiv preprint arXiv:2310.02446*, 2023.

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.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How Johnny can persuade LLMs to jailbreak them: Rethinking persuasion to challenge AI safety by humanizing LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14322–14350, 2024. doi: 10.18653/v1/2024.acl-long.773.

Hangtao Zhang, Yucheng Zhao, Sishun Liu, Ziqi Zhou, Zeyu Ye, Wei Wan, Minghui Li, Shengshan Hu, Yanjun Zhang, Yi Liu, and Leo Yu Zhang. Defending jailbreak attacks on large language models via manifold trajectory kinetics. *arXiv preprint arXiv:2606.07335*, 2026.

Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. JBShield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. In *Proceedings of the USENIX Security Symposium*, pp. 8215–8234, 2025. URL `https://www.usenix.org/conference/usenixsecurity25/presentation/zhang-shenyi`.

Zhexin Zhang, Junxiao Yang, Pei Ke, Fei Mi, Hongning Wang, and Minlie Huang. Defending large language models against jailbreaking attacks through goal prioritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8865–8887, 2024. doi: 10.18653/v1/2024.acl-long.481.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. LLMs encode harmfulness and refusal separately. In *Advances in Neural Information Processing Systems*, volume 38, pp. 140283–140318, 2025. doi: 10.52202/085713-4688.

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.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *Advances in Neural Information Processing Systems*, volume 37, pp. 83345–83373, 2024. doi: 10.52202/079017-2651.

A SUPPORTING PROOFS

We connect conditional Koopman approximation to the expected residual advantage in Section 3. Each result fixes the fitted lift, operators, and pooling, with square-integrable targets and forecasts. Vector norms are Euclidean, matrix norms are induced, and $\bar{y}$ denotes the other class.

A.1 COMMON PREDICTION SPACE

Let $G = (L, T)$ specify a valid field and let $\mathcal{I}(G)$ contain its valid forecast indices $i = (a, v, h)$. The input $X_i = (i, G, \xi_v, u_0, \dots, u_{h-1})$ records exactly the history and controls used to forecast $Z_i = z_h$. Both class predictors $f_{c,i}(X_i) = \widehat{z}_{c,h}$ are evaluated in the shared coordinates of Equation (5). Expanding the pooling in Equation (10) gives

$$E_c(p) = \frac{1}{d} \sum_{i \in \mathcal{I}(G)} \omega_i \|Z_i - f_{c,i}(X_i)\|_2^2, \tag{17}$$

where $\omega_i = w\varpi_\ell / (|\mathcal{Q}_{\mathrm{tok}}|(T-h))$ for $i = (\mathrm{tok}, (\ell, t), h)$ and $\omega_i = (1-w)/(T|\mathcal{Q}_{\mathrm{dep}}|(L-h))$ for $i = (\mathrm{dep}, (\ell, t), h)$. Appendix F defines the horizon sets. On nonempty valid axis task sets, these weights are nonnegative and sum to one. They depend on the forecast inputs and are shared by both hypotheses; tasks with zero weight are omitted.

For a prompt population $\mathbb{P}$ and forecast functions $f = (f_i)_i$, define the energy-weighted norm

$$\|f\|_{\mathbb{P}}^2 = \frac{1}{d} \mathbb{E}_{p \sim \mathbb{P}} \left[\sum_{i \in \mathcal{I}(G)} \omega_i \|f_i(X_i)\|_2^2 \right], \tag{18}$$

with $\|\cdot\|_y$ denoting the class-y population. Equivalently, draw $p \sim \mathbb{P}$, sample a task I with probabilities ω_i, and set $(X, Z) = (X_I, Z_I)$. The resulting weighted input law is denoted by $\mathbb{P}^X$. Input-measurable sampling preserves $\mathbb{E}[Z_i \mid X_i, C = y]$, while accommodating the dependence among forecasts from the same prompt. This construction connects every norm below to the energies actually compared by CORD.

A.2 RELATIVE PREDICTIVE FIT

Write $\mu_{y,i}(X_i) = \mathbb{E}[Z_i \mid X_i, C = y]$. Orthogonality of the centered target to functions of X_i gives

$$\mathbb{E}[\|Z_i - f_{c,i}\|_2^2 \mid X_i, C = y] = \mathbb{E}[\|Z_i - \mu_{y,i}\|_2^2 \mid X_i, C = y] + \|\mu_{y,i} - f_{c,i}\|_2^2, \tag{19}$$

Proof of Proposition 1. Expand $Z_i - f_{c,i} = (Z_i - \mu_{y,i}) + (\mu_{y,i} - f_{c,i})$. The conditional cross term is zero. Subtracting the two class risks therefore cancels their common target variance. Multiplying by the same ω_i/d, summing over tasks, and taking the class-y expectation proves Equation (12).

To relate this identity to predictor separation, expand $\mu_y - f_{\bar{y}} = (\mu_y - f_y) + (f_y - f_{\bar{y}})$ in the weighted space. With $\mathcal{S}_y = \|f_y - f_{\bar{y}}\|_y$ and $\mathcal{B}_y = \|\mu_y - f_y\|_y$,

$$\begin{aligned} \Gamma_y &= \mathcal{S}_y^2 + \frac{2}{d} \mathbb{E} \left[\sum_i \omega_i (f_{y,i} - f_{\bar{y},i})^{\mathsf{T}} (\mu_{y,i} - f_{y,i}) \mid C = y \right] \\ &\geq \mathcal{S}_y (\mathcal{S}_y - 2\mathcal{B}_y), \end{aligned} \tag{20}$$

where the inequality is Cauchy–Schwarz. Thus $\mathcal{S}_y > 2\mathcal{B}_y$ yields a correct-class residual advantage in expectation. The relevant comparison is separation relative to own-class prediction bias. Shared target variance cancels even when its magnitude varies across prompts and forecast tasks.

The same decomposition gives $\mathbb{E}[E_y \mid C = y] = \mathcal{B}_y^2 + \sigma_y^2$, where $\sigma_y^2 = d^{-1}\mathbb{E}[\sum_i \omega_i \|Z_i - \mu_{y,i}\|_2^2 \mid C = y]$. For fixed observables and pooling, σ_y^2 is constant with respect to the predictor parameters, linking own-class forecasting directly to reduction of $\mathcal{B}_y$. Equation (11) divides the signed class-energy difference by a common positive scale. Consequently $s(p) \in [-1, 1]$, and $s(p) > 0$ exactly when $E_{\mathrm{j}}(p) < E_{\mathrm{s}}(p)$. The threshold τ_α is applied to this normalized contrast as specified in Section 3.3.

A.3 CONTROLLED ERROR PROPAGATION

Fix a class c and axis a, suppressing these subscripts on the operator matrices. Equation (8) makes A normal: orthogonal similarity preserves the singular values of its rotation–scaling blocks, so $\|A\|_2 = \rho(A) = \max_j \rho_j \le \rho_{\max}$, where $\rho(A)$ is the spectral radius. For the full context-dependent matrix $M(u_r) = A + \beta W \operatorname{diag}(Ru_r)P$,

$$\|M(u_r)\|_2 \le \Lambda_r, \qquad \Lambda_r = \rho_{\max} + |\beta|\|W\|_2\|Ru_r\|_\infty\|P\|_2, \tag{21}$$

using $\|\operatorname{diag}(Ru_r)\|_2 = \|Ru_r\|_\infty$. This is the gain that propagates errors during the implemented conditional forecasts.

Lemma 1 (Pathwise propagation)**.** *Let $\mathbf{e}_r = z_{r+1} - F_{c,a}(z_r, u_r)$ be the vector one-step residual. Starting from $\widehat{z}_{c,0} = z_0$ and using the same observed controls,*

$$\|\widehat{z}_{c,h} - z_h\|_2 \le \sum_{r=0}^{h-1} \left(\prod_{k=r+1}^{h-1} \Lambda_k \right) \|\mathbf{e}_r\|_2, \tag{22}$$

with empty scalar products equal to one.

Proof. Set $\mathbf{d}_r = \widehat{z}_{c,r} - z_r$. The common affine displacement cancels, giving $\mathbf{d}_{r+1} = M(u_r)\mathbf{d}_r - \mathbf{e}_r$ and $\mathbf{d}_0 = 0$. Iteration multiplies each residual by the subsequent matrices in temporal order, $M(u_{h-1}) \cdots M(u_{r+1})$. Applying the triangle inequality and Equation (21) proves the bound.

The spectral structure bounds the base contribution to error propagation; the bilinear term accounts for context-dependent amplification. We next connect this propagation to the conditional-mean bias in Proposition 1.

A.4 KOOPMAN CLOSURE AND PREDICTIVE BIAS

Use the weighted sampling of Appendix A.1 and condition on the task $J = (G, I) = j$. Denote its class-y law by $\mathbb{P}_{y,j}$ and expectation by $\mathbb{E}_{y,j}$. The input X includes this task, the starting history, and all observed controls used by its forecast. Along its axis a, let $Z_r = \phi(\Xi_r)$, $F = F_{y,a}$, and $\mathbf{r}_r = \mathbf{r}_{y,a}(\Xi_r, u_r)$ be the Koopman closure residual from Assumption 2. The one-step decomposition is

$$Z_{r+1} = F(Z_r, u_r) + \mathbf{r}_r + \boldsymbol{\zeta}_r, \qquad \boldsymbol{\zeta}_r = Z_{r+1} - (\mathcal{K}_{y,a}^{u_r}\phi)(\Xi_r), \tag{23}$$

where the conditional action uses the fixed grid and current source position defined in Section 2. Set $\mu_r(X) = \mathbb{E}_{y,j}[Z_r \mid X]$, $\mathbf{a}_r(X) = \mathbb{E}_{y,j}[\mathbf{r}_r \mid X]$, and $\mathbf{c}_r(X) = \mathbb{E}_{y,j}[\boldsymbol{\zeta}_r \mid X]$. Because u_r is known from X and F is affine in Z_r,

$$\mu_{r+1} = M_{y,a}(u_r)\mu_r + B_{y,a}u_r + \mathbf{a}_r + \mathbf{c}_r, \tag{24}$$

with $\mu_0 = Z_0$. Here $\mathbf{a}_r$ carries finite-observable approximation error. The term $\mathbf{c}_r$ captures predictive information retained by the initial history and observed control sequence beyond conditioning on the current reduced history. It is the memory contribution required to compose conditional predictions along the observed path.

Lemma 2 (Conditional-mean bias)**.** *Suppose $\|M_{y,a}(u_k)\|_2 \le \overline{\Lambda}_{j,k} < \infty$ almost surely under $\mathbb{P}_{y,j}$. Apply Assumption 2 to its source marginals, so $\mathbb{E}_{y,j}\|\mathbf{r}_r\|_2^2 \le \delta_{y,a}^2$, and define $b_{j,r}^{\mathrm{mem}} = (\mathbb{E}_{y,j}\|\mathbf{c}_r(X)\|_2^2/d)^{1/2}$. Then*

$$\left(\frac{1}{d}\mathbb{E}_{y,j}\|\mu_h(X) - \widehat{z}_{y,h}(X)\|_2^2\right)^{1/2} \le \underbrace{\sum_{r=0}^{h-1} \left(\prod_{k=r+1}^{h-1} \overline{\Lambda}_{j,k} \right) \left(\frac{\delta_{y,a}}{\sqrt{d}} + b_{j,r}^{\mathrm{mem}} \right)}_{\mathcal{U}_{y,j}}, \tag{25}$$

where the gains and memory terms refer to the fixed class y.

Proof. Subtract the forecast recurrence from Equation (24). The difference starts at zero and evolves through $M_{y,a}(u_r)$ with forcing $\mathbf{a}_r + \mathbf{c}_r$. Iterate in the same order as Lemma 1 and apply the triangle inequality in $L^2(\mathbb{P}_{y,j})$. Conditional Jensen gives $\mathbb{E}_{y,j}\|\mathbf{a}_r\|_2^2 \le \mathbb{E}_{y,j}\|\mathbf{r}_r\|_2^2 \le \delta_{y,a}^2$. Bounding each matrix product by $\prod_k \overline{\Lambda}_{j,k}$ and dividing by $\sqrt{d}$ yields the result.

For a finite pooled budget, the law of total expectation gives

$$\mathcal{B}_y^2 \le \mathcal{U}_y^2 := \sum_j \Pr(J = j \mid C = y)\mathcal{U}_{y,j}^2, \tag{26}$$

which, substituted into Equation (20), proves Equation (13). The source moments and task weights are taken under the same laws, so the bound respects the model's layer gate and axis weights.

At one step, X supplies the current history and control, making $\mathbf{c}_0 = 0$ and the conditional-mean error equal to the closure residual. At longer horizons, $\mathcal{U}_y$ combines this approximation error with the memory terms and their controlled propagation. The criterion $\mathcal{S}_y > 2\mathcal{U}_y$ therefore links the method's shared history, conditional operators, and spectral structure to a single requirement: separation between the two forecasts must exceed their accumulated own-class bias.

A.5 Transfer across attack families

Let $\mathbb{P}_{\text{seen}}$ be the seen-attack mixture and $\mathbb{P}_\star$ a held-out attack family, evaluated with the same fitted detector. Under the weighted input laws, define $\mu_{\text{seen}}(X) = \mathbb{E}_{\text{seen}}[Z \mid X]$, $\mu_\star(X) = \mathbb{E}_\star[Z \mid X]$, the predictive-mean shift $\Delta_{\text{dyn}} = \|\mu_\star - \mu_{\text{seen}}\|_\star$, and the seen-family bias $\mathcal{B}_{\text{seen}} = \|\mu_{\text{seen}} - f_{\text{j}}\|_{\text{seen}}$.

Proposition 2 (Transferred predictive bias)**.** *Suppose the weighted input laws satisfy $\mathbb{P}_\star^X \ll \mathbb{P}_{\text{seen}}^X$ and $d\mathbb{P}_\star^X / d\mathbb{P}_{\text{seen}}^X \le \kappa_{\text{cov}} < \infty$ almost everywhere. Then*

$$\mathcal{B}_\star := \|\mu_\star - f_{\text{j}}\|_\star \le \Delta_{\text{dyn}} + \sqrt{\kappa_{\text{cov}}}\,\mathcal{B}_{\text{seen}}, \tag{27}$$

and, for $\mathcal{S}_\star = \|f_{\text{j}} - f_{\text{s}}\|_\star$ and $\Gamma_\star = \mathbb{E}_\star[E_{\text{s}} - E_{\text{j}}]$,

$$\Gamma_\star \ge \mathcal{S}_\star \left(\mathcal{S}_\star - 2\Delta_{\text{dyn}} - 2\sqrt{\kappa_{\text{cov}}}\,\mathcal{B}_{\text{seen}}\right), \tag{28}$$

Proof. The triangle inequality gives $\mathcal{B}_\star \le \Delta_{\text{dyn}} + \|\mu_{\text{seen}} - f_{\text{j}}\|_\star$. Changing measure on the common weighted input space yields

$$\|\mu_{\text{seen}} - f_{\text{j}}\|_\star^2 = \frac{1}{d}\mathbb{E}_{\mathbb{P}_{\text{seen}}^X}\left[\frac{d\mathbb{P}_\star^X}{d\mathbb{P}_{\text{seen}}^X}\|\mu_{\text{seen}}(X) - f_{\text{j}}(X)\|_2^2\right] \le \kappa_{\text{cov}}\mathcal{B}_{\text{seen}}^2, \tag{29}$$

proving Equation (27). Apply the separation bound in Equation (20) under $\mathbb{P}_\star$ to obtain Equation (28).

The factor κ_{cov} measures coverage of held-out forecast inputs by the seen mixture, while Δ_{dyn} measures changes in their predictive means. Lemma 2 supplies $\mathcal{B}_{\text{seen}} \le \mathcal{U}_{\text{seen}}$, giving

$$\Gamma_\star \ge \mathcal{S}_\star \left[\mathcal{S}_\star - 2\left(\Delta_{\text{dyn}} + \sqrt{\kappa_{\text{cov}}}\mathcal{U}_{\text{seen}}\right)\right], \tag{30}$$

so the jailbreak predictor retains an expected residual advantage when $\mathcal{S}_\star > 2(\Delta_{\text{dyn}} + \sqrt{\kappa_{\text{cov}}}\mathcal{U}_{\text{seen}})$. This states the transfer principle in terms of predictive regularities: observed trajectories may change while the jailbreak predictor remains closer to the held-out conditional successor mean. LOAFO evaluation in Section 4.2 examines detection on families excluded from fitting.

A.6 Connection to learning

CORD's losses act on the two quantities in Equation (20). Own-class forecasting reduces conditional-mean bias for fixed targets and pooling. Classification acts on the per-prompt residual margin. For class y, write $r_y = E_{\bar{y}} - E_y$; the two-class energy loss and its derivative are

$$\ell_y = \log\left(1 + e^{-r_y/\tau_E}\right), \qquad \frac{\partial \ell_y}{\partial r_y} = -\frac{1}{\tau_E(1 + e^{r_y/\tau_E})} < 0, \tag{31}$$

for $\tau_E > 0$. Applied to r_n^{tr}, this is the classification term in Equation (14). The persistence penalty adds a reference predictor that repeats the initial observable, encouraging the fitted dynamics to explain more than a small displacement. These objectives jointly favor accurate own-class forecasts and a positive margin over the competing class.

Both class hypotheses share the lift and pooling throughout training. The fixed TICA prefix anchors the observable coordinates; after the joint stage, freezing the lift fixes the target representation for

operator refinement. For the final forecasts and targets, define $r_y^a = E_{\bar{y}}^a - E_y^a$ and $\widetilde{r}_y = w_{\mathrm{tr}} r_y^{\mathrm{tok}} + (1 - w_{\mathrm{tr}}) r_y^{\mathrm{dep}}$. The relation between this common-target margin and the inference margin is

$$r_y - \widetilde{r}_y = (w - w_{\mathrm{tr}})(r_y^{\mathrm{tok}} - r_y^{\mathrm{dep}}), \tag{32}$$

which isolates the effect of axis weighting on fixed forecasts. Algorithm 2 defines the targets in r_n^{tr}; Proposition 1 uses final coordinates and inference weights. Sections 4.3 and 4.5 test predictive fitting and its components through matched-input comparisons and ablations.

B Training details

B.1 Coordinates, controls, and parameter dimensions

PCA whitening retains 512 directions and is fitted exclusively on ordinary-safe training examples. TICA retains 256 directions from 16,384 token and 16,384 depth transition pairs in the permitted safe and jailbreak training classes. For N_{pair} centered PCA pairs (χ_n, χ_n^+), with $\chi_n, \chi_n^+ \in \mathbb{R}^{512}$, define

$$\begin{aligned} \Sigma_0 &= \frac{1}{2N_{\mathrm{pair}}} \sum_n (\chi_n \chi_n^{\mathsf{T}} + \chi_n^+ \chi_n^{+\mathsf{T}}), \\ \Sigma_{\mathrm{lag}} &= \frac{1}{2N_{\mathrm{pair}}} \sum_n (\chi_n \chi_n^{+\mathsf{T}} + \chi_n^+ \chi_n^{\mathsf{T}}), \\ \Sigma_{\mathrm{lag}} q_j^{\mathrm{TICA}} &= \lambda_j^{\mathrm{TICA}} (\Sigma_0 + \eta_{\mathrm{TICA}} I) q_j^{\mathrm{TICA}}. \end{aligned} \tag{33}$$

Here $\eta_{\mathrm{TICA}} > 0$ regularizes the covariance. We retain the 256 eigenvectors with largest $|\lambda_j^{\mathrm{TICA}}|$ and standardize their coordinates on training data, giving $d_x = 256$. Symmetrization and balanced axis pooling define a common lag-correlation basis for both class hypotheses. The history input in Equation (5) has dimension $3d_x = 768$; its SiLU branch has widths $768 \to 128 \to 32$. The fixed branch retains the first 32 TICA coordinates, giving total observable dimension $d = 64$.

Token-edge controls $u_t^{\mathrm{tok}} \in \mathbb{R}^8$ are fixed Gaussian projections of the current user-token embedding. For normalized layer position $r_\ell = (\ell - 1)/(L - 1)$, real block index b_ℓ, and its maximum $b_{\max}$, depth controls are

$$u_\ell^{\mathrm{dep}} = [1, r_\ell, r_\ell^2, \sin(\pi r_\ell), \cos(\pi r_\ell), \sin(2\pi r_\ell), \cos(2\pi r_\ell), b_\ell / b_{\max}]^{\mathsf{T}}. \tag{34}$$

Both class predictors use the control at each forecast step's source position: the source-token embedding for token transitions and the source-layer encoding for depth transitions. Prefill supplies the full observed control sequence used by each rollout. These controls define the conditioning information in Equation (4). For every class and axis, $A \in \mathbb{R}^{64\times 64}$, $B \in \mathbb{R}^{64\times 8}$, $W \in \mathbb{R}^{64\times 16}$, $P \in \mathbb{R}^{16\times 64}$, and $R \in \mathbb{R}^{16\times 8}$. The bilinear interaction has width 16, while A acts on all 64 observable coordinates. Its spectral form has 32 rotation–scaling blocks and

$$Q = H_8 \cdots H_1, \qquad H_j = I - 2\frac{\mathbf{v}_j \mathbf{v}_j^{\mathsf{T}}}{\mathbf{v}_j^{\mathsf{T}} \mathbf{v}_j}, \tag{35}$$

Each $\mathbf{v}_j \in \mathbb{R}^{64}$ is a nonzero reflection vector. Thus $H_j^{\mathsf{T}} H_j = I$ and $Q^{\mathsf{T}} Q = I$, giving the orthogonal mixing used in Equation (8). Class and axis subscripts on these parameters are suppressed throughout this subsection.

B.2 Losses and update rule

Let $E_{n,c}^{a,\mathrm{tr}}$ denote axis-a energy on the training crop with its stage-specific targets. Then $E_{n,c}^{\mathrm{tr}} = w_{\mathrm{tr}} E_{n,c}^{\mathrm{tok,tr}} + (1 - w_{\mathrm{tr}}) E_{n,c}^{\mathrm{dep,tr}}$. The two hypotheses share the crop, valid transition masks, horizons, and targets. Equations (14)–(16) define the three losses and their nonnegative coefficients. The persistence energy Π_n repeats the initial observable at every horizon using the same targets and pooling as $E_{n,c}^{\mathrm{tr}}$.

The energy-CE branch scales gradients into the four operators' parameters Ω by κ, using $\widetilde{\Omega} = \mathrm{sg}(\Omega) + \kappa(\Omega - \mathrm{sg}(\Omega))$. Here sg is the identity in the forward pass with zero derivative. Hence

$\widetilde{\Omega} = \Omega$ in value and $\partial\widetilde{\Omega}/\partial\Omega = \kappa I$; gradients into the lift are unchanged. This specifies the update rule while preserving the classification loss in Equation (14).

During joint training, the teacher lift follows $\bar{\theta} \leftarrow \mu_{\mathrm{EMA}}\bar{\theta} + (1-\mu_{\mathrm{EMA}})\theta$, with momentum $\mu_{\mathrm{EMA}} \in [0, 1)$ and detached targets. After the last joint-training epoch e_f, the online lift θ is frozen and supplies the detached targets for operator refinement. Algorithm 2 expresses both stages using the same notation as the inference procedure.

B.3 Detection procedure

Algorithm 1 implements Section 3.3. Here Θ denotes the fitted detector, including preprocessing and control maps, lift parameters θ, operator parameters Ω, gate vector g, and readout w. The map $\mathcal{P}$ is the fixed PCA–TICA–standardization map; histories use the zero padding of Equation (5). Both hypotheses share observed controls, targets, and pooling weights. The path notation follows Equation (9). The valid horizons $\mathcal{Q}_a(v)$ for each start and the axis-wise averages are defined in Appendix F.

Algorithm 1 CORD scoring and detection

Require: p, frozen Θ, threshold τ_α; nonempty valid transition sets
function SCORE($p; \Theta$)
  $\{x_{\ell,t}\} \leftarrow \mathcal{P}(\mathcal{H}(p))$; $u_t^{\mathrm{tok}}, u_\ell^{\mathrm{dep}}$ as in Appendix B.1
  $\xi_{\ell,t} \leftarrow [x_{\ell,t}; x_{\ell,t-1}; x_{\ell-1,t}]$, $z_{\ell,t} \leftarrow \phi_\theta(\xi_{\ell,t})$
  for $c \in \{\mathrm{s}, \mathrm{j}\}$, $a \in \{\mathrm{tok}, \mathrm{dep}\}$, and valid starts $v = (\ell, t)$ **do**
    $\mathcal{Q}_a(v) \leftarrow$ valid horizons in $\{1, 2, 4\}$; $(z_r, u_r) \leftarrow$ path from v along a
    $\widehat{z}_{c,0} \leftarrow z_0$
    for $r = 0, \ldots, \max \mathcal{Q}_a(v) - 1$ **do**
      $\widehat{z}_{c,r+1} \leftarrow F_{c,a}(\widehat{z}_{c,r}, u_r)$
    end for
    $e^{(h)}_{c,a,v} \leftarrow \|\widehat{z}_{c,h} - z_h\|_2^2/d$, $h \in \mathcal{Q}_a(v)$
  end for
  $\varpi_\ell \leftarrow \exp(g^\top u_\ell^{\mathrm{dep}}) / \sum_{j=1}^{L} \exp(g^\top u_j^{\mathrm{dep}})$
  For each c, compute $E_c^{\mathrm{tok}}, E_c^{\mathrm{dep}}$ by Equation (10) and Appendix F
  $E_c \leftarrow wE_c^{\mathrm{tok}} + (1-w)E_c^{\mathrm{dep}}$, $c \in \{\mathrm{s}, \mathrm{j}\}$
  return $(E_{\mathrm{s}} - E_{\mathrm{j}}) / \max\{E_{\mathrm{s}} + E_{\mathrm{j}}, \epsilon_E\}$
end function
$s(p) \leftarrow$ SCORE($p; \Theta$), $\delta_\alpha(p) \leftarrow \mathbf{1}\{s(p) \geq \tau_\alpha\}$
return $s(p), \delta_\alpha(p)$

B.4 Training and calibration procedure

Algorithm 2 fits one fold with its readout weight supplied, excluding the target family from fitting, then calls Algorithm 1 for calibration. Write Ω for the four operators' parameters, υ_e for the trainable parameters, and γ_e for the epoch-e learning rate with initial value γ_0. The empirical quantile $\widehat{Q}_{1-\alpha}$ is computed from calibration scores. The operator clip_5 clips the global gradient norm to 5. The surrogate $\widetilde{\mathcal{J}}$ equals Equation (16) in value, with $\widetilde{\Omega}$ used only in the CE branch; all branches share the same detached targets. Here e_f is the last joint-training epoch and s_f is the subsequent learning-rate multiplier. Fixed controls are defined in Appendix B.1.

C Evaluation protocol

The main comparison uses fixed-configuration leave-one-attack-family-out (LOAFO) evaluation. Each model uses one fixed configuration across ten folds. In each fold, one attack family is reserved for final testing and excluded from preprocessing and detector fitting. Table 4 gives the disjoint fitting, threshold-estimation, and test samples; Section 4 identifies the ten attack families.

The main detector retains the final 64 user tokens. Safe PCA-512 is preserved; TICA-256, standardization, the shared lift, and the operators are refitted in each fold with the target family excluded. Training uses token weight $w_{\mathrm{tr}} = 0.70$. The readout weights are $w = 0.60$ for Llama2 and Llama3 and $w = 0.80$ for Qwen2.5 and Mistral, shared across each model's ten folds.

Algorithm 2 CORD training and calibration for one fitting fold

Require: Frozen target model; $\mathcal{D}_{\mathrm{s}}$, $\mathcal{D}_{\mathrm{j}}$, independent $\mathcal{D}_{\mathrm{cal}}$; fixed configuration
Ensure: Θ, $\{\tau_\alpha\}$

Construct $\mathcal{P}$: safe PCA, refitted joint TICA and training standardization
Initialize θ, Ω, g; $\bar{\theta} \leftarrow \theta$
for $e = 1, \ldots, 20$ **do**
  $(\upsilon_e, \gamma_e) \leftarrow \begin{cases} ((\theta, \Omega, g), \gamma_0), & e \le e_f, \\ ((\Omega, g), s_f \gamma_0), & e > e_f \end{cases}$
  for class-balanced minibatch $\{(p_n, y_n)\}_{n=1}^{N_b}$ **do**
    Sample ten-layer crops; construct ξ, controls, and valid masks
    $z \leftarrow \phi_\theta(\xi), \quad z^{\mathrm{tgt}} \leftarrow \begin{cases} \mathrm{sg}(\phi_{\bar{\theta}}(\xi)), & e \le e_f, \\ \mathrm{sg}(z), & e > e_f \end{cases}$
    Compute Equation (9) with target z_h^{tgt} for each prompt
    $E_{n,c}^{\mathrm{tr}} \leftarrow w_{\mathrm{tr}} E_{n,c}^{\mathrm{tok,tr}} + (1 - w_{\mathrm{tr}}) E_{n,c}^{\mathrm{dep,tr}}$ ▷ Equation (10)
    $\widetilde{\Omega} \leftarrow \mathrm{sg}(\Omega) + \kappa(\Omega - \mathrm{sg}(\Omega))$
    $\widetilde{\mathcal{J}} \leftarrow \lambda_{\mathrm{roll}} \mathcal{L}_{\mathrm{roll}} + \lambda_{\mathrm{CE}} \mathcal{L}_{\mathrm{CE}}(\theta, \widetilde{\Omega}, g) + \lambda_{\mathrm{pers}} \mathcal{L}_{\mathrm{pers}}$
    $\mathbf{G} \leftarrow \nabla_{\upsilon_e} \widetilde{\mathcal{J}}; \quad \upsilon_e \leftarrow \mathrm{AdamW}(\upsilon_e, \mathrm{clip}_5(\mathbf{G}); \gamma_e)$
    if $e \le e_f$ **then**
      $\bar{\theta} \leftarrow \mu_{\mathrm{EMA}} \bar{\theta} + (1 - \mu_{\mathrm{EMA}}) \theta$
    end if
  end for
end for
Freeze Θ at epoch 20 with readout w
$s_n \leftarrow \textsc{Score}(p_n; \Theta), \quad p_n \in \mathcal{D}_{\mathrm{cal}}$
$\tau_\alpha \leftarrow \widehat{Q}_{1-\alpha}(\{s_n\}), \quad \alpha \in \{0.01, 0.05\}$
return $\Theta, \{\tau_\alpha\}$

Table 4: Data roles in one main-comparison fold.

Role	Count	Use
Safe training	2,700	Joint TICA, lift, operator losses
Seen attacks	2,700	Joint TICA, lift, operator fitting
Safe calibration	1,000	Score threshold estimation
Target attack test	300	Held-out attack detection
Safe test	300	Observed test FPR

TPR@1 reports the attack detection rate at the threshold set for 1% FPR; the observed FPR on independent benign test prompts is reported in Table 6. TPR@5 uses the corresponding 5% threshold in the order-perturbation diagnostic (Figure 5). We first average the ten attack-family results within each model, then average the four model means with equal weights.

Analytical comparisons. Analytical studies and serving measurements use the main comparison's 64-token windows and model-level readout weights. Matched-input alternatives share the joint inputs and history architecture, with separate training. Single-axis controls refit TICA and standardization on their retained trajectory and train for 20 epochs, using only that axis for history and prediction energy. The component ablations retain the parent settings and readout; removing dynamics fitting disables both the prediction and persistence losses. Hard-benign TPR@1 uses thresholds determined for its respective negative set. Order-perturbation thresholds at 1% and 5% use the correspondingly transformed ordinary-safe calibration scores. Figure 4 uses the Llama3 GPTFuzz-held-out detector with $w = 0.60$.

Data sources and deterministic sampling. Seven attack pools (AutoDAN, Base64, DrAttack, GCG, PAIR, SAA, and Zulu) contain 500 prompts each from the JBShield release (Zhang et al., 2025). Base64 and Zulu use the encoded and translated prompts supplied in that release. GPTFuzz supplies 500 prompts (Yu et al., 2023); the 500 IJP prompts are obtained through SoK4JailbreakGuardrails (Wang et al., 2026), whose IJP source is JailbreakHub (Shen et al., 2024). The 500 JailJudge prompts come from the MTK release (Zhang et al., 2026). The 5,000-prompt

safe training pool combines UltraChat (2,493) (Ding et al., 2023) with OASST (1,255) (Köpf et al., 2023); Dolly (1,252) (Conover et al., 2023) supplies the remainder. Safe calibration combines UltraChat (506), Dolly (254), and OASST (240); the 1,000-prompt safe evaluation pool combines UltraChat (483), Dolly (261), and OASST (256). Prompt IDs are identical across target models. IDs are sorted and deterministically permuted within each role/family group; each seen family supplies 300 training prompts. The main comparison uses the counts in Table 4.

D Additional experimental analyses

D.1 Hard-benign results

Table 5 resolves the main-text hard-benign comparison by model. Using OKTest (Shi et al., 2024) and XSTest-safe (Röttger et al., 2024) as the respective benign reference sets, CORD achieves macro TPR@1 of 87.39% and 88.73% and ranks first in seven of eight model–dataset combinations. The gains extend across models on legitimate requests with challenging safety-related content. Entries average ten attack families; macro results weight models equally.

Table 5: **Hard-benign results by model.** Ten-family mean TPR@1 (%).

Model	Negative set	CORD	MTK	JBShield-D	GraphShield	GradSafe
Llama3	OKTest	**92.10**	34.27	1.27	60.13	26.00
Llama3	XSTest-safe	**91.77**	35.80	7.13	54.13	14.33
Llama2	OKTest	**80.53**	34.40	0.27	19.10	33.37
Llama2	XSTest-safe	**86.60**	29.50	0.10	49.27	3.13
Qwen2.5	OKTest	**88.40**	31.93	7.47	80.84	15.70
Qwen2.5	XSTest-safe	85.10	27.70	4.90	**85.97**	4.47
Mistral	OKTest	**88.53**	20.10	27.57	32.93	26.27
Mistral	XSTest-safe	**91.47**	28.07	27.83	55.00	9.47

Table 6 complements detection rates with observed false alarms. It applies each fold's fixed ordinary-benign threshold to 300 ordinary-benign test prompts, 300 OKTest prompts, and 250 XSTest-safe prompts. Mean test FPR is 1.33% on ordinary-benign inputs, with model means from 1.17% to 1.67%; the corresponding hard-benign means are 3.62% and 3.01%. Together, the two tables characterize attack detection with each benign reference set and false alarms under a shared ordinary-benign threshold.

Table 6: **Observed FPR on benign test sets.** Ten-fold means (%); Avg weights models equally.

Model	Ordinary benign	OKTest	XSTest-safe
Llama3	1.17	3.93	2.64
Llama2	1.67	4.40	3.88
Qwen2.5	1.20	1.57	3.32
Mistral	1.27	4.57	2.20
Avg	1.33	3.62	3.01

D.2 Matched-input comparisons

The alternatives use the same 64-token windows, joint TICA-256 inputs, and causal-history architecture as CORD, with each alternative trained separately. Direct classification predicts the label from the matched hidden-state inputs without successor forecasting. The one-step spectral operator retains the structured dynamics but restricts its forecast horizon to $h = 1$. CORD and the one-step operator each have 115,624 trainable parameters, versus 115,670 for direct classification. Each alternative is evaluated on the same four models and ten attack families.

CORD exceeds direct classification on every model by 12.50–28.50% (Table 7). With common projected inputs, history architecture, and nearly identical parameter counts, this comparison supports predictive model comparison as a source of detection evidence beyond direct classification.

Table 7: **Matched-input controls (64 tokens).** Ten-family mean TPR@1 (%); Avg weights models equally.

Model	CORD	Direct	Spectral $h = 1$
Qwen2.5	**92.23**	78.73	82.30
Llama3	**95.63**	83.13	89.80
Mistral	**94.63**	70.97	83.50
Llama2	**93.27**	64.77	93.23
Avg	**93.94**	74.40	87.21

The one-step operator retains the same structured transition model; composing it across horizons raises macro TPR@1 from 87.21% to 93.94%. The increases are 9.93%, 5.83%, and 11.13% on Qwen2.5, Llama3, and Mistral, respectively, while Llama2 reaches 93.27% with both alternatives closely matched. These results connect the use of learned transition relationships to detection accuracy under matched observations.

E Additional dynamics diagnostics

E.1 Local field evidence

We locate predictive evidence within the fields of Llama3 and Qwen2.5 across all ten held-out families, using 300 benign and 300 attack prompts per cell. For each class and axis, one-step edge errors from Equation (9) are averaged over the valid edges incident to each node. Interior nodes receive two edges; boundary nodes receive one. The two axis energies are combined with the same readout weight w used by the detector and converted to a normalized class contrast $s^{\mathrm{joint}}_{\ell,t}(p)$. Writing $\widehat{\mathbb{E}}_c$ for the empirical class-c mean over valid observations, Figure 4 displays

$$\Delta s^{\mathrm{joint}}_{\ell,t} = \widehat{\mathbb{E}}_{\mathrm{j}}[s^{\mathrm{joint}}_{\ell,t}] - \widehat{\mathbb{E}}_{\mathrm{s}}[s^{\mathrm{joint}}_{\ell,t}], \tag{36}$$

so a positive value indicates a shift toward the jailbreak predictor on attack prompts. The marginal profiles retain the unfused axis scores and average their empirical class differences:

$$\overline{\Delta s}^{\mathrm{tok}}(t) = \frac{1}{L}\sum_{\ell=1}^{L} \Delta s^{\mathrm{tok}}_{\ell,t}, \qquad \overline{\Delta s}^{\mathrm{dep}}(\ell) = \frac{1}{T}\sum_{t=1}^{T} \Delta s^{\mathrm{dep}}_{\ell,t}. \tag{37}$$

Here $\Delta s^{a}_{\ell,t} = \widehat{\mathbb{E}}_{\mathrm{j}}[s^{a}_{\ell,t}] - \widehat{\mathbb{E}}_{\mathrm{s}}[s^{a}_{\ell,t}]$. The profiles average separate axis contrasts; the central field contrasts fused energies. The local one-step scores resolve spatial variation; the prompt-level score pools multi-horizon energies as in Section 3.3. Figure 4 uses zero-based display positions $(\ell-1, t-1)$ and a symmetric color range set by the 98th percentile of absolute joint differences.

The spatial mean of the joint difference is positive in all 20 model–family cells. For Llama3–GPTFuzz, the 32×64 field has mean difference 0.15 and positive differences at 99.56% of positions. Its strongest layer-averaged contrast occurs at displayed depth index 13 ($\ell = 14$). The evidence is distributed across the retained field, with a stronger contrast at intermediate depth, complementing the model-level axis comparison in Section 4.4.

F Energy aggregation

The valid global horizon sets are $\mathcal{Q}_{\mathrm{tok}} = \{h \in \{1,2,4\} : h < T\}$ and $\mathcal{Q}_{\mathrm{dep}} = \{h \in \{1,2,4\} : h < L\}$. For a start $v = (\ell, t)$, Algorithm 1 uses $\mathcal{Q}_{\mathrm{tok}}(v) = \{h \in \mathcal{Q}_{\mathrm{tok}} : t + h \le T\}$ or $\mathcal{Q}_{\mathrm{dep}}(v) = \{h \in \mathcal{Q}_{\mathrm{dep}} : \ell + h \le L\}$. On fields with nonempty axis horizon sets, the per-layer and per-token averages are

$$\begin{aligned} E^{\mathrm{tok}}_{c,\ell} &= \frac{1}{|\mathcal{Q}_{\mathrm{tok}}|}\sum_{h\in\mathcal{Q}_{\mathrm{tok}}} \frac{1}{T-h}\sum_{t=1}^{T-h} e^{(h)}_{c,\mathrm{tok},(\ell,t)}, \\ E^{\mathrm{dep}}_{c,t} &= \frac{1}{|\mathcal{Q}_{\mathrm{dep}}|}\sum_{h\in\mathcal{Q}_{\mathrm{dep}}} \frac{1}{L-h}\sum_{\ell=1}^{L-h} e^{(h)}_{c,\mathrm{dep},(\ell,t)}. \end{aligned} \tag{38}$$

Each horizon is normalized by its own number of valid starts before horizons are averaged. Applying the layer gate and axis weight in Equation (10) therefore yields exactly the weights ω_i in Equation (17). History padding supplies the lift input; the forecast masks select in-grid targets. Both classes use the same masks and denominators.

G SERVING COST

We measure the serving overhead of the 64-token detector with vLLM 0.19.0 (Kwon et al., 2023) on an NVIDIA RTX 3090 GPU. At concurrency 16 and 32, each of three paired runs processes the same 320 requests per arm (160 ordinary-benign and 160 jailbreak). Each request generates one token. Synchronized-batch TTFT spans batch submission through completion of all first tokens and CORD decisions, after initialization and within the local serving process. CORD reuses prefill hidden states through hooks in eager execution.

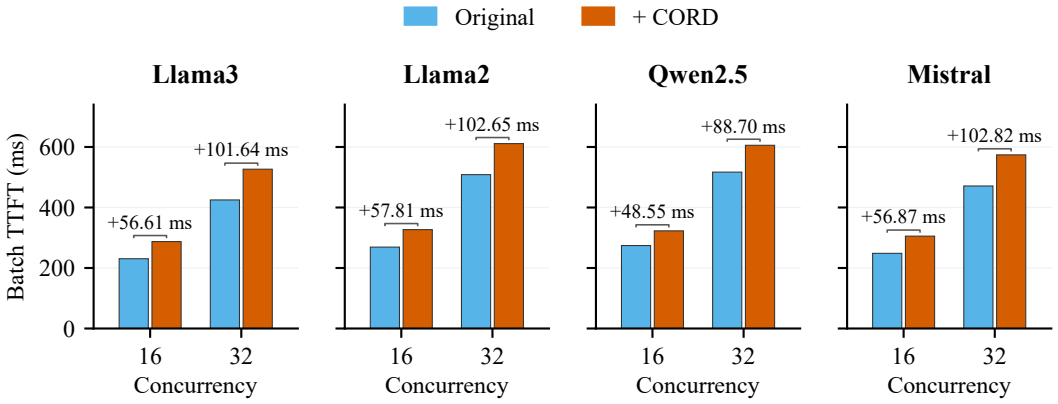


Figure 6: **Batched first-token latency.** Mean TTFT at concurrency 16 and 32; annotations show the paired increase in ms.

Figure 6 shows added batch latency of 48.55–102.82 ms. At concurrency 32, this corresponds to 23.92% for Llama3, 20.18% for Llama2, 17.16% for Qwen2.5, and 21.82% for Mistral. Across the two concurrency levels, the added latency accounts for a smaller fraction of the larger batches' baseline TTFT. Table 8 complements the means with per-run batch p95 values averaged over the three paired runs.

Table 8: Batch TTFT p95 (ms). Each cell reports original / defended latency.

Model	Concurrency 16	Concurrency 32
Llama3	261.37 / 319.86	481.22 / 582.10
Llama2	360.56 / 417.82	577.57 / 680.27
Qwen2.5	315.79 / 365.04	570.68 / 659.18
Mistral	306.78 / 363.69	516.19 / 619.42

Each detector has 115,624 trainable parameters. Qwen2.5 uses 946,656 fixed deployment coefficients (1,062,280 total); the other three use 1,081,856 (1,197,480 total).

H SCORE DISTRIBUTIONS

Figures 7–9 extend Figure 2 to Llama2, Qwen2.5, and Mistral. Benign and attack medians lie on opposite sides of the decision threshold in all 30 panels, showing consistent score orientation across the additional models and attack families. Positive scores favor the jailbreak predictor, as defined in Equation (11). Each model uses common bin edges across its panels, and density is normalized separately for the two classes.

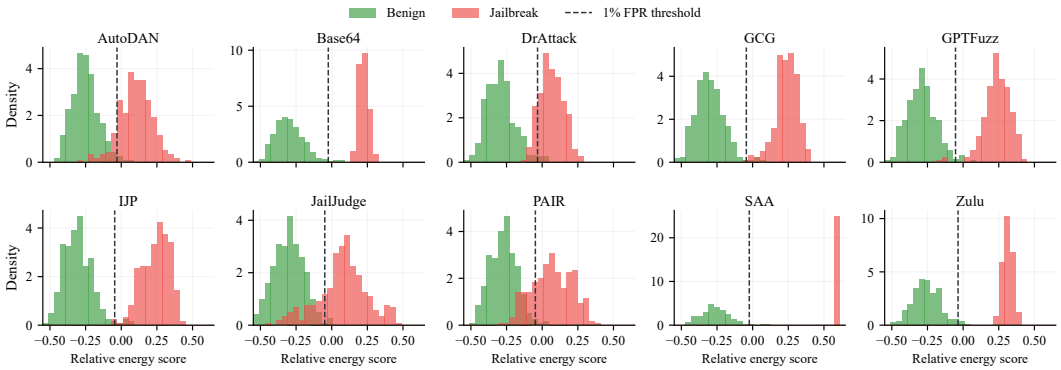


Figure 7: **Llama2 held-out-family score distributions.** Each panel contains 300 benign and 300 attack prompts; dashed lines mark the thresholds for TPR@1.

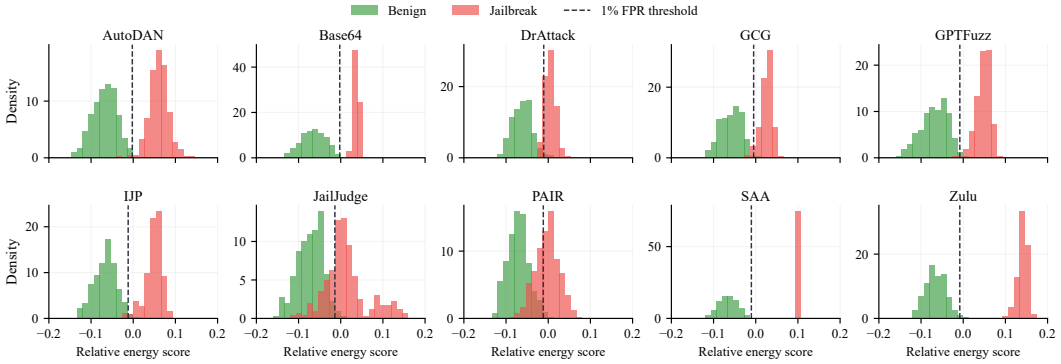


Figure 8: **Qwen2.5 held-out-family score distributions.** Each panel contains 300 benign and 300 attack prompts; dashed lines mark the thresholds for TPR@1.

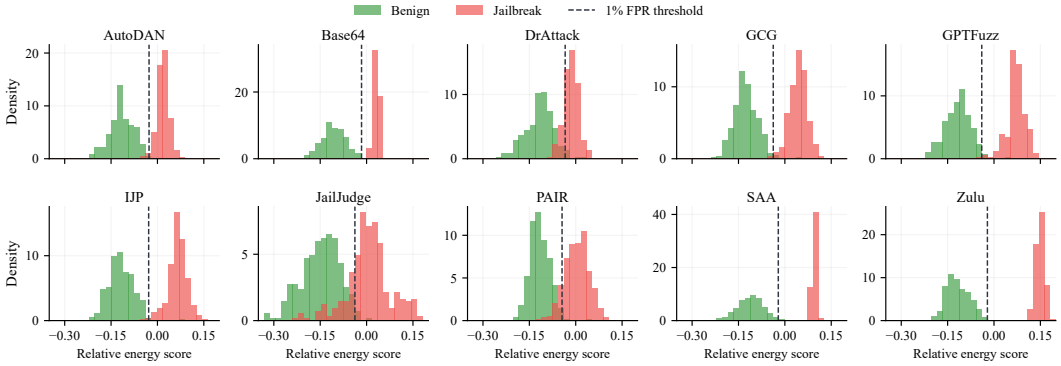


Figure 9: **Mistral held-out-family score distributions.** Each panel contains 300 benign and 300 attack prompts; dashed lines mark the thresholds for TPR@1.