Training-Free Policy Violation Detection via Activation-Space Whitening in LLMs

Oren Rachmil[1*], **Roy Betser**[1], **Itay Gershon**[1], **Omer Hofman**[1], **Nitay Yakoby**[2], **Yuval Meron**[2], **Idan Yankelev**[2], **Asaf Shabtai**[2], **Yuval Elovici**[2], **Roman Vainshtein**[1]

[1]Fujitsu Research of Europe, Israel
[2]Ben-Gurion University of the Negev, Israel

Abstract

Aligning proprietary large language models (LLMs) with internal organizational policies has become an urgent priority as organizations increasingly deploy LLMs in sensitive domains such as legal support, finance, and medical services. Beyond generic safety filters, enterprises require reliable mechanisms to detect policy violations within their regulatory and operational frameworks, where breaches can trigger legal and reputational risks. Existing content moderation frameworks, such as *guardrails*, remain largely confined to the safety domain and lack the robustness to capture nuanced organizational policies. *LLM-as-a-judge* and fine-tuning approaches, though flexible, introduce significant latency and lack interpretability. To address these limitations, we propose a training-free and efficient method that treats policy violation detection as an out-of-distribution (OOD) detection problem. Inspired by whitening techniques, we apply a linear transformation to decorrelate the model's hidden activations and standardize them to zero mean and unit variance, yielding near-identity covariance matrix. In this transformed space, we use the Euclidean norm as a compliance score to detect policy violations. The method requires only the policy text and a small number of illustrative samples, which makes it lightweight and easily deployable. On a challenging policy benchmark, our approach achieves state-of-the-art results, surpassing both existing guardrails and fine-tuned reasoning models. This work provides organizations with a practical and statistically grounded framework for policy-aware oversight of LLMs, advancing the broader goal of trustworthy, deployable AI governance. Code implementation will be released upon publication.

1 Introduction

Large language models (LLMs) are increasingly adopted across organizations, where they are integrated into applications such as document drafting, automated customer support, and data analysis pipelines (Cohere 2023; Urlana et al. 2024; Liang et al. 2025). As this adoption accelerates, organizations face a critical challenge of ensuring that LLMs operate in compliance with both internal organizational policies and external regulatory and compliance requirements across diverse domains (e.g., legal, financial, ethical, medical) (Liu et al. 2023). In realistic enterprise settings, policy compliance rarely involves a single rule: models must simultaneously satisfy dozens of policies, each may consist of hundreds of rules. Each rule introduces new contextual conditions, linguistic subtleties, and exceptions (Saura et al. 2025; Hoover et al. 2025). Even high-performing commercial LLMs can produce responses that satisfy one rule while inadvertently violating another (OpenAI 2024; Zeng et al. 2024). This misalignment risk can carry serious legal and financial consequences: for instance, the U.S. Federal Trade Commission (FTC) required the DoNotPay company to halt deceptive claims about its AI legal-service chatbot and imposed monetary relief on the company (Federal Trade Commission 2025).

Several guardrail methods constrain LLM behavior (Inan et al. 2023; Rebedea et al. 2023; Yuan et al. 2024; Kang and Li 2025), but they rely on fixed categories, handcrafted rules, and discrete boundaries that do not scale to heterogeneous organizational policies and lose robustness under diverse or nuanced requirements. Consequently, commercial LLMs (e.g., GPT-4 (Achiam et al. 2023)) are often used as "LLM-as-a-Judge" evaluators (Gu et al. 2024) or as fine-tuned detectors (Hoover et al. 2025); however, these black-box classifiers have no interpretability and incur substantial infrastructure cost and latency, limiting real-time monitoring and large-scale deployment.

In this work, we address these gaps by introducing a novel policy-violation detection method that operates directly in the activation space of LLMs. Our approach is training-free, lightweight, and interpretable, designed to support deployable AI governance without fine-tuning or external evaluators. Inspired by whitening-based likelihood estimation from the image domain (Betser, Levi, and Gilboa 2025), we model activations from policy-compliant user-LLM interactions as an in-distribution manifold and treat policy violations as out-of-distribution (OOD) deviations in this space. Conditioned on the organization's policy rules, we analyze hidden states across transformer layers to assign a compliance score to each interaction. We fit a data-driven whitening transform to in-policy activations, producing standardized features with approximately identity covariance; in this whitened space, policy compliance is scored by the Euclidean norm of the whitened activation vector. At runtime, we compute a compliance score on a selected operational layer and compare it to a pre-defined calibrated threshold,

*Corresponding author: rachmiloren@gmail.com

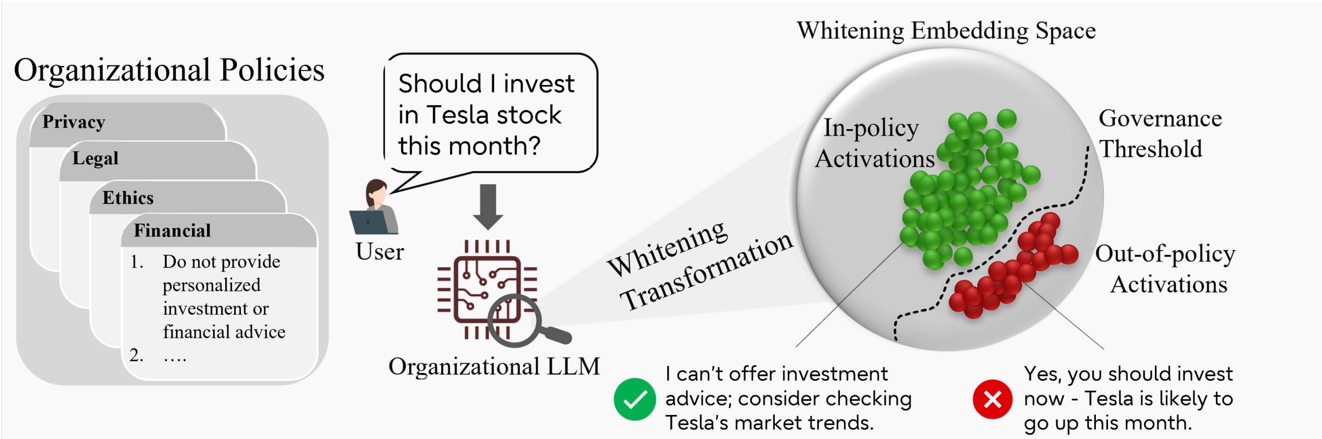

Figure 1: **Illustration of the proposed policy-violation detection framework.** Organizational policies define expected behavior for an internal LLM. When a user query produces a response, the model's hidden activations are transformed using a whitening matrix derived from in-policy samples. Compliance is then estimated via the activation norm in this whitened space, and responses whose whitened norms exceed a calibrated governance threshold are flagged as policy violations.

enabling detection of out-of-policy responses without additional prompting or external evaluators. The result is a flexible, low-overhead solution suited for continuous policy updates and monitoring, deployable in both white-box and black-box settings.

We extensively evaluate our method on the challenging *DynaBench* policy dataset (Hoover et al. 2025). In the whitened space (illustrated in Figure 1), in-policy samples cluster near the origin with lower norms, while out-of-policy samples shift outward with higher norms. Our method achieves strong separation of compliant vs. violating responses, outperforming both an LLM-as-a-judge (GPT-4o-mini) and the fine-tuned 8B DynaGuard model by up to 9%. Our contributions can be summarized as follows:

- **Activation-level policy modeling.** We cast policy adherence as an OOD detection problem in activation space, whitening each model's layer separately, to yield an interpretable compliance score.
- **Training-free scoring rule.** Given the precomputed whitening matrix, compliance is scored by the Euclidean norm in the whitened space; a small mixed calibration set defines a single decision threshold, no fine-tuning required.
- **Access-regime agnosticism.** The method operates in both white-box (direct activations) and black-box settings (surrogate activations), with an identical scoring pipeline.
- **Deployment efficiency.** The approach adds negligible latency while achieving high performance, providing a solution suitable for real-time gating and continuous monitoring.

2 Background and Related Works

The rapid deployment of large language models (LLMs) has driven an acceleration of research on guardrail frameworks and content moderation systems designed to constrain model behavior. One of the first safety oriented fine-tuned models, LlamaGuard (Inan et al. 2023), was introduced to detect and filter harmful content in both user inputs and model responses. Since the release of LlamaGuard, several studies have proposed increasingly sophisticated approaches spanning ensemble moderation, lightweight transformer architectures, and generative moderation pipelines (Han et al. 2024; Datta and Rajasekar 2025; Ghosh et al. 2024; Zeng et al. 2024). While these methods mark significant progress, they remain largely constrained to predefined safety taxonomies: optimized on specific risk categories and limited in scope to the safety domain, such as toxicity or bias. As a result, they struggle to capture the contextual and dynamic nature of organizational policies, which often involve evolving regulations and domain-specific constraints.

Several recent works have explored policy enforcement and compliance validation using LLMs. Saura et al. (2025) proposed a retrieval-augmented architecture for automatic security policy enforcement, while Wang et al. (2025) and Chen et al. (2025) investigated privacy-policy compliance through fine-tuned or prompt-based LLM classifiers. These studies highlight the potential of LLMs as compliance monitors but share common drawbacks: reliance on supervised fine-tuning or LLM-as-a-judge inference, incur high latency and computational cost, and remain confined to predefined safety taxonomies trained on narrow risk categories.

A noticeable advancement in policy alignment was introduced by (Hoover et al. 2025), who presented DynaBench, a large-scale benchmark specifically designed for training and evaluating guardrail models on complex policy-compliance tasks. The dataset comprises over $60,000$ policies paired with annotated in- and out-of-policy multi-turn user–agent dialogues that capture both adherence and violation scenarios. It also includes a human-written, domain-specific evaluation set extending beyond the training distribution, making it particularly challenging. Building on the DynaBench benchmark, Hoover et al. (2025) also introduced Dyna-

LLM Policy Detection Approach	Training-Free	Low-Latency	Interpretable
Guardrails	✗	✓	✗
LLM-as-a-Judge	✓	✗	✗
DynaGuard	✗	✗	△
Whitening-based OOD Detection (Ours)	✓	✓	✓

Table 1: Comparison of policy detection approaches. ✓= supported, △= partial, ✗= not supported. Our method combines training-free deployment, low-latency, and interpretability.

Guard, a 8B-parameter fine tuned Qwen (Yang et al. 2024) model fine-tuned on the 60k-sample DynaBench training set, which outperforms GPT-4o-mini in policy-violation detection while offering reduced latency and cost. However, such systems depend on extensive supervised fine-tuning and large curated datasets, which hinder adaptability to new policies and real world deployment. As summarized in Table 1, LLM-as-a-judge and fine-tuned guardrails trade off training-free deployment, latency and interpretability. Our whitening-based detector supports all three: it is training-free, low-latency and interpretable.

A complementary line of research arises from out-of-distribution (OOD) detection, which improves the reliability of machine learning models by identifying inputs that deviate from a model's training distribution (Hendrycks and Gimpel 2016). Initially developed for image classification (Liang, Li, and Srikant 2017; Lee et al. 2018; Hsu et al. 2020; Chen et al. 2021), OOD techniques, such as Mahalanobis scoring and temperature scaling, have since been adapted for textual and multimodal models (Hendrycks et al. 2020; Zhou, Liu, and Chen 2021; Chen et al. 2023a). Recently, whitening-based OOD detection has shown great promise in the image domain, effectively decorrelating hidden features and enabling likelihood-based detection without retraining (Chen et al. 2023b; Betser, Levi, and Gilboa 2025). We adapt whitening-based OOD scoring to the LLM activation space, framing policy compliance as OOD detection.

A related thread of research focuses on mechanistic interpretability, which examines internal representations and circuit-level structure to explain model behavior (Olah et al. 2020; Nanda et al. 2023). Additional work demonstrates that activations in LLMs often encode high-level concepts as structured subspaces or directions in representation space (Park, Choe, and Veitch 2023). Our approach is adjacent to this perspective: rather than identifying circuits or causal pathways, we perform *layer-wise* diagnostics of the activation geometry. After whitening, in-policy activations are expected to cluster near the origin, while out-of-policy activations shift along dominant directions, providing a fine-grained, interpretable signal of alignment.

Building on these insights, we frame policy compliance as an OOD detection problem in the activation space of LLMs. This perspective connects OOD detection with practical AI governance, providing a training-free, adaptable, and interpretable framework for policy-violation detection.

3 Method

Our core hypothesis is that *policy-compliant responses* occupy a consistent region of the activation space, while *policy-violating responses* produce activations that deviate from this distribution. Hence, detecting violations can be formulated as an out-of-distribution (OOD) detection problem in the latent activation space.

Our method operates directly on intermediate representations, allowing policy alignment assessment without modifying model weights or prompts. In the offline step, a small set of in-policy samples defines the reference distribution, and a mixed set containing a few out-of-policy samples calibrates a decision threshold. At runtime, model responses are validated by scoring their activations against this reference; deviations beyond the empirical threshold indicate potential policy violations. The framework supports both white-box and black-box operation: in white-box mode, layer activations are observed directly; in black-box mode, a surrogate model provides activation proxies from input-output traces.

We next describe the data acquisition process, followed by the pre-processing estimation and calibration procedure, and finally the runtime detection step.

3.1 Data Access Regimes

White-box. We access the model's internal representations directly. Let $\mathcal{C} = \big((r_1, s_1), \ldots, (r_M, s_M)\big)$ denote a conversation of M turns, where $r_m \in \{\text{user}, \text{agent}\}$ and s_m is the text at turn m. Concatenating the turns yields a token sequence of length T. Passing this sequence through a transformer f_θ with layers $h_1, \ldots, h_L$ produces per-layer hidden states $H_\ell(\mathcal{C}) \in \mathbb{R}^{T\times d}$. We summarize each conversation by its final-token representation. This choice is natural since the final token integrates the full conversational context, capturing the model's concluding internal state and overall policy stance:

$$x_\ell(\mathcal{C}) = H_\ell(\mathcal{C})[T] \in \mathbb{R}^d. \tag{1}$$

For a corpus $\mathcal{S}$ of conversations, the per-layer activation set is

$$\mathcal{X}_\ell = \{\, x_\ell(\mathcal{C}) : \mathcal{C} \in \mathcal{S} \,\}, \qquad \ell \in \mathcal{L} \subseteq \{1, \ldots, L\}. \tag{2}$$

All subsequent estimation (mean, covariance, and whitening) is performed independently per layer. A single operational layer $\ell^\star$ is retained for runtime scoring, and each model response is validated at runtime before being returned to the user. These activations capture contextual and semantic structure relevant to policy compliance and are obtained without modifying model parameters or prompts.

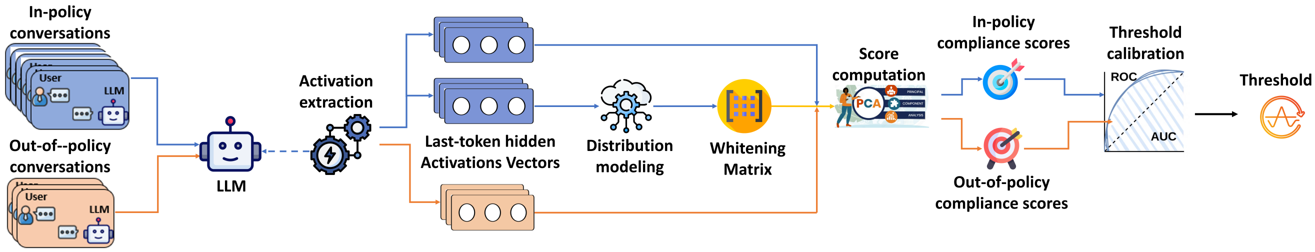

Figure 2: **Illustration of the offline phase of compliance calibration.** In-policy and out-of-policy user-LLM interactions are first collected and passed through the model for *activation extraction*. Last-token hidden activation vectors are then used for *distribution modeling* to derive a whitening matrix that normalizes layer activations. Using this matrix, *compliance scores* are computed for both in-policy and out-of-policy activations, followed by *ROC-AUC-based threshold calibration* to identify the optimal decision boundary separating compliant and non-compliant interactions.

Black-box. When direct access to internal activations is unavailable (e.g., commercial LLM APIs), we employ a *surrogate model* under our control (white-box) to compute the necessary statistics. At runtime, each response produced by the target model is passed through this surrogate model, whose activations serve as proxies for representational behavior. These surrogate representations are then processed through the same pipeline as in the white-box case, enabling consistent compliance evaluation across access regimes. The surrogate model can be any white-box LLM or a fine-tuned organization-specific LLM, allowing seamless integration of domain-adapted or customized models.

Reference data for statistics and calibration. In both modes, two representative datasets are collected. A minimal *in-policy* set, containing compliant outputs, is used to estimate activation statistics for whitening. A second, mixed set containing both *in-* and *out-of-policy* samples is used to calibrate the decision threshold. No retraining or fine-tuning of the base model is required.

3.2 Offline reference statistics pre-processing

A small in-policy set defines the reference behavior in the activation space, as illustrated in Fig. 2. For each layer, we compute a transform that standardizes activations to a common, comparable scale. A small mixed calibration set (in- and out-of-policy) is then used to choose a single operational layer and set a decision threshold. Only the chosen layer's transform and the calibrated threshold are retained for deployment.

Whitening transformation. To compute compliance scores, we first establish a standardized activation space using the whitening transform, ensuring that all dimensions are comparable. Let $\{x_i^{(\ell)}\}_{i=1}^N$ denote activation vectors extracted from layer ℓ over a representative set of in-policy interactions, $x_i^{(\ell)} \in \mathbb{R}^d$. We first compute the empirical mean and covariance:

$$\mu^{(\ell)} = \frac{1}{N} \sum_{i=1}^{N} x_i^{(\ell)}, \tag{3}$$

$$\Sigma^{(\ell)} = \frac{1}{N-1} \sum_{i=1}^{N} \big(x_i^{(\ell)} - \mu^{(\ell)}\big)\big(x_i^{(\ell)} - \mu^{(\ell)}\big)^\top. \tag{4}$$

Whitening applies a linear transform that decorrelates activations and maps them to a zero-mean, unit-covariance space. The whitening matrix $W^{(\ell)} \in \mathbb{R}^{d\times d}$ therefore satisfies

$$W^{(\ell)\top} W^{(\ell)} = \big(\Sigma^{(\ell)}\big)^{-1}. \tag{5}$$

The matrix is not unique; any orthogonal rotation of a valid transform also preserves unit covariance. A common and convenient choice is PCA-based whitening, obtained via eigen-decomposition:

$$\Sigma^{(\ell)} = V^{(\ell)} \Lambda^{(\ell)} V^{(\ell)\top}, \tag{6}$$

where $V^{(\ell)} \in \mathbb{R}^{d\times d}$ has as columns the eigenvectors of $\Sigma^{(\ell)}$ and $\Lambda^{(\ell)} = \mathrm{diag}\big(\lambda_1^{(\ell)}, \ldots, \lambda_d^{(\ell)}\big)$ contains the corresponding (nonnegative) eigenvalues, typically ordered $\lambda_1^{(\ell)} \geq \cdots \geq \lambda_d^{(\ell)}$. The whitening matirx is then defined:

$$W^{(\ell)} = \Lambda^{(\ell)^{-\frac{1}{2}}} V^{(\ell)\top}. \tag{7}$$

Applying $W^{(\ell)}$ to a centered activation gives the whitened representation:

$$y^{(\ell)} = W^{(\ell)}\big(x^{(\ell)} - \mu^{(\ell)}\big), \tag{8}$$

whose components are decorrelated, have zero mean, and approximately unit variance (see Fig. 4). These whitened activations define a standardized coordinate system where deviations can be measured uniformly across dimensions.

Score definition. The PCA-based whitening can include dimensionality reduction by retaining only the top-k eigenvalues and their corresponding eigenvectors from the covariance decomposition, projecting activations onto the dominant directions of in-policy variability. In this reduced space, deviations are measured by the Euclidean norm

$$s^{(\ell)} = \big\|y^{(\ell)}\big\|_2, \tag{9}$$

which quantifies how far a test activation lies from the in-policy region defined by these principal components. This score is equivalent to the Mahalanobis distance:

$$\big(s^{(\ell)}\big)^2 = \big(x^{(\ell)} - \mu^{(\ell)}\big)^\top \big(\Sigma^{(\ell)}\big)^{-1} \big(x^{(\ell)} - \mu^{(\ell)}\big), \tag{10}$$

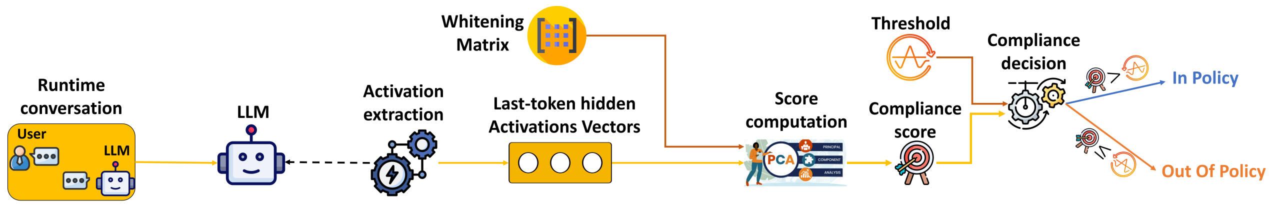

Figure 3: **Illustration of the online compliance detection process.** During runtime user-LLM interaction, last-token hidden activations are extracted and whitened using the precomputed whitening matrix. The resulting vector is used to compute a compliance score, which is compared against the precomputed calibrated threshold to determine whether the interaction is *in-policy* (compliant) or *out-of-policy* (violating).

but focuses on the most informative subspace of the data, providing a stable and interpretable measure of alignment with in-policy behavior. See Appendix F for an extended mathematical derivation.

Layer selection and threshold calibration. For each layer ℓ, whitening parameters $\mu^{(\ell)}$ and $W^{(\ell)}$ are computed from the in-policy reference set. A separate mixed calibration set containing both in- and out-of-policy samples is used to evaluate layer-wise discrimination performance (e.g., ROC-AUC). The layer with the highest separation is chosen as the operational layer $\ell^{\star}$, and its corresponding mean and whitening matrix are retained.

Policy-conditioned whitening. The same procedure naturally extends to environments with large numbers of policy rules, which can be grouped into broader policy classes with shared behavioral patterns. For each class, we estimate in-policy statistics and derive a class-specific whitening transform together with its class mean. The resulting per-class parameters are stored for later selection during online detection.

In practice, both whitening and calibration require only small sample sets. A single sample per policy rule often suffices, and fewer may be needed when many rules share the same class. This keeps the procedure lightweight and at a data scale where retraining or fine-tuning would not be beneficial.

Threshold calibration. Once the operational layer $\ell^{\star}$ is fixed, we determine a decision threshold τ on a held-out mixed calibration set with *ground-truth* compliance labels. For each calibration example, we compute its compliance score $s^{(\ell^{\star})}$ and pair it with its known label (in-policy vs. out-of-policy). We then select τ by maximizing Youden's J statistic, $J = \mathrm{TPR} - \mathrm{FPR}$, which balances true- and false-positive rates on the calibration data. This yields a single operating point that strongly separates compliant from non-compliant activations without additional tuning or probabilistic modeling.

3.3 Online detection pipeline

At runtime, each model response is validated before being returned to the user. The activation at the operational layer $\ell^{\star}$ is centered using the stored mean, whitened using $W^{(\ell^{\star})}$, and scored (see the deployment pipeline in Fig. 3) according to

$$s^{(\ell^{\star})} = \|W^{(\ell^{\star})}(x^{(\ell^{\star})} - \mu^{(\ell^{\star})})\|_2. \tag{11}$$

The compliance decision follows a simple rule:

$$s^{(\ell^{\star})} \le \tau \;\Rightarrow\; \text{in-policy}, \qquad s^{(\ell^{\star})} > \tau \;\Rightarrow\; \text{out-of-policy}. \tag{12}$$

If policy grouping is used, the class whose mean is closest to the current activation (by cosine similarity) is first selected, and its parameters $\{\mu, W\}$ are applied for scoring. This provides a lightweight, real-time mechanism for identifying potential policy violations without modifying model parameters or prompts.

Connection to Gaussian likelihood. In the whitened space, activations already have zero mean and identity covariance. If they further approximate a Gaussian distribution, $y^{(\ell^{\star})} \sim \mathcal{N}(0, I_d)$, their likelihood under the in-policy model is

$$p\big(y^{(\ell^{\star})}\big) = (2\pi)^{-d/2} \exp\Big(-\tfrac{1}{2}\big\|y^{(\ell^{\star})}\big\|_2^2\Big), \tag{13}$$

with log-likelihood

$$\log p\big(y^{(\ell^{\star})}\big) = -\tfrac{d}{2}\log(2\pi) - \tfrac{1}{2}\big\|y^{(\ell^{\star})}\big\|_2^2. \tag{14}$$

In this Gaussian limit, lower norms correspond to higher probability density and thus stronger in-policy conformity, turning the compliance score from a distribution discriminator into a probabilistic model of activation consistency.

4 Evaluation

4.1 Dataset & Contrastive Data Construction

Benchmark. We evaluate our method on the *DynaBench* benchmark (Hoover et al. 2025), using its manually curated test split designed to assess policy compliance in multi-turn user–agent dialogues. Each sample includes a policy defined as a set of one or more textual rules the model must follow and a dialogue labeled as either *in-policy* (compliant) or *out-of-policy* (violating), see dialogue example in Appendix A. The test set spans twelve *business impact categories*, covering diverse domains such as regulatory compliance, information leakage, user experience, and brand reputation. See Appendix B for detailed statistics.

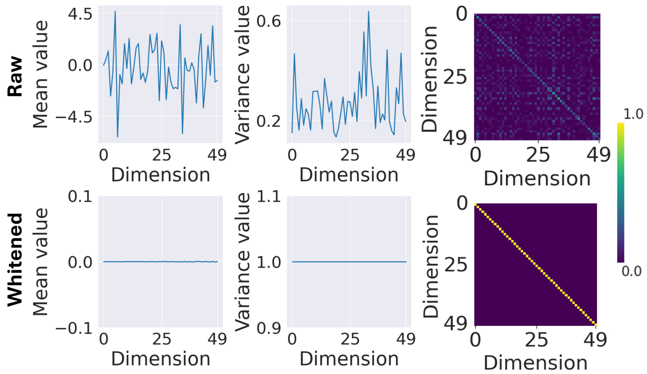

Figure 4: **Statistics of LLM activations before and after whitening.** *Top:* Raw activations exhibit arbitrary means/variances and substantial cross-dimensional covariance. *Bottom:* Whitened activations are approximately zero-mean, unit-variance, with near-identity covariance. Category - content control.

Contrastive data generation. To construct representative data for computing the whitening matrices and threshold calibration, we generate rule-specific contrastive datasets for each DynaBench policy. Each policy is decomposed into its constituent rules, and for each rule, an LLM-based generator (GPT-4o) produces three natural-language prompts that explicitly or implicitly query the rule. For each prompt, the LLM synthesizes four contrastive samples in a realistic conversational style, where each pair consists of a *good response* adhering to the rule (in-policy) and a *bad response* deliberately violating it (out-of-policy). See Appendix C for an illustrative example.

4.2 Evaluation settings

Implementation details. We evaluate our method on two popular open source models: *Llama 3.1-8B* (Dubey et al. 2024) and *Qwen2.5-7B* (Yang et al. 2024). We extract final-token hidden activations from all transformer layers (32 for Llama, 28 for Qwen). For each of the twelve policy categories in *Dynabench*, we select 100 samples from the contrastive data, then split 80/20. Within the 80% subset, only in-policy samples are used to fit per-layer whitening; the held-out 20% (containing both in and out of policy samples) is used for threshold calibration. Unless noted otherwise, whitening retains the top-k components ($k = 15$). Ablations over sample size and k value appear in Fig. 5.

Per-class statistics. For each category, we compute per-layer whitening and quantify in-/out-of-policy separation with ROC-AUC. The best layer per category (highest ROC-AUC) defines the operational guardrail layer (see Fig. 6); its parameters $(\mathbf{W}, \boldsymbol{\mu})$ are stored for runtime scoring. Fig. 4 shows that whitened activations are approximately zero-mean with unit variance and near-identity covariance, whereas raw activations exhibit nonzero means, heterogeneous variances, and cross-dimensional covariance (example is on the content control category, see additional examples in Appendix G). We also evaluated our method using a single whitening matrix for all categories; category-specific whitening performed markedly better (Appendix D).

4.3 Results

Results on the Benchmark Table 2 reports the F1 scores of all competing methods. Despite requiring no fine-tuning, our approach surpasses LLM-as-a-judge (GPT-4o-mini, Qwen3-8B) and also fine-tuned baselines, including DynaGuard and LlamaGuard, confirming that policy compliance can be modeled effectively as an OOD problem in the activation space. LLM-as-a-judge was instructed with policy rules as system prompts and DynaGuard was implemented as Qwen models (different sizes) fine tuned on the 60k Dynebench train split. Competitors' results are taken from (Hoover et al. 2025). Our method achieves superior performance across both Llama and Qwen backbones, with the Qwen variant yielding the strongest results overall. A direct comparison using Qwen models of identical size shows a 9.1% improvement in F1 score over DynaGuard, underscoring the efficiency and generalization strength of our training-free approach.

Approach	Model	F1 (%)
LLM-as-a-judge	GPT-4o-mini	70.1
	Qwen3-8B	60.7
Fine tuned	LlamaGuard3	20.9
	DynaGuard-1.7B	65.2
	DynaGuard-4B	72.0
	DynaGuard-8B (non-CoT)	72.5
	DynaGuard-8B	73.1
Ours	**Llama 3.1 8B**	**74.3**
	Qwen 2.5 7B	**82.2**

Table 2: Comparison of F1 scores on the *DynaBench* test set. Baseline results are taken from the DynaBench benchmark paper. Our OOD whiteing based method using the activations of Llama-3.1 and Qwen-2.5 models achieve the strongest performance, surpassing the previous best DynaGuard-8B (73.1%) by up to **9.1 points**.

Ablation on Parameters We perform an ablation study on the *DynaBench* test set using embeddings from the *Llama* model to assess the effect of both hyper-parameters: the number of retained principal components (top-k) and the sample size per category used for whitening and calibration (together). Results are summarized in Fig. 5, reveling performance remains stable across a wide range of k values and improves only marginally with larger calibration sets, highlighting the method's robustness and efficiency. Using 100 samples per category already achieves strong results with an F1 score of **74.3%**, while increasing up to 750 samples yields only a modest improvement to **77.7%**. Similarly, varying top-k from 10 to 50 results in only minor fluctuations (from **71.2%** to **75.4%**).

Per-Layer Analysis We study where policy signals emerge by tracking layer-wise ROC-AUC across categories

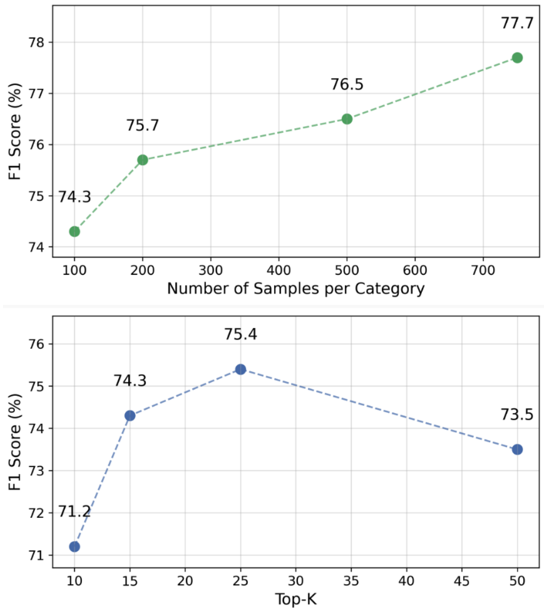

Figure 5: Ablation study showing the effect of (top) varying Top-K (with 100 samples per category) and (bottom) varying the number of samples per category (with Top-$K = 15$) on F1 score.

Model	GPT-4o-mini	DynaGuard-8B	Black-box (ours)		White-box (ours)	
			Llama	Qwen	Llama	Qwen
Runtime [s]	1.47	2.71	0.98	0.92	**0.05**	**0.03**

Table 3: **Average runtime per sample (seconds).** Measured over 100 samples from the *DynaBench* test set. Our method achieves real-time performance in the white-box regime (0.03–0.05 s) and maintains sub-second latency in black-box deployments, while delivering superior detection quality compared to both GPT-4o-mini and DynaGuard-8B.

in the *Llama* model. Figure 6 highlights two contrasting depth profiles: *Information Leakage* rises sharply, peaks early, then declines; in contrast, *Transactions* increases gradually, experiences a mid-layer dip, then peaks late before stabilizing around high values through the final layers. This layer-resolved view makes the emergence of alignment in activation space interpretable and motivates our per-layer guard selection. A broader summary appears in Appendix E, where the best layer per category is reported: most categories select mid-to-late layers (25–32), but several select earlier layers (e.g., Non-Player Characters, Information Leakage, Product Hallucination), underscoring genuine variation in where policy-relevant structure concentrates.

Runtime Analysis Finally, we evaluate inference efficiency for *DynaGuard-8B* (the most capable DynaGuard model), *GPT-4o-mini* (API-based judge), and our proposed method under both white-box and black-box configurations. Table 3 reports the mean runtime per test conversation over 100 samples from the *DynaBench* test set. In the white-box

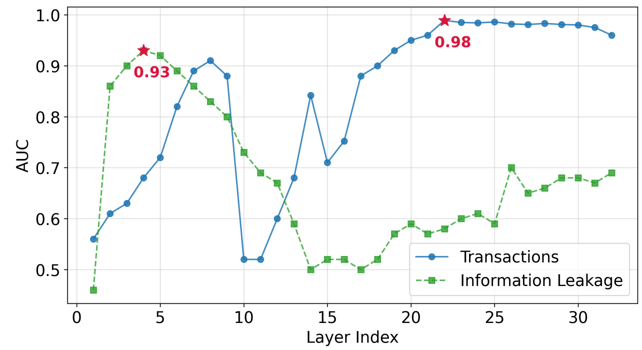

Figure 6: **Layer-wise ROC–AUC for two policy categories using *Llama 3.1 8B Instruct*.** AUC values per transformer layer on the *DynaBench* test set. Blue: *Transactions*; green dashed: *Information Leakage*. *Information Leakage* peaks early (AUC=0.93) and then declines, whereas *Transactions* rises with depth, shows a mid-layer dip, peaks late (AUC=0.98), and remains high at the final layers. These divergent trajectories show that policy categories have distinct internal dynamics across layers, underscoring the need for an interpretable, category-specific solution.

setting, our whitening-based detector introduces only **0.03–0.05** seconds of additional latency, effectively negligible relative to model inference and fully compatible with real-time moderation pipelines. In the black-box scenario, latency remains under one second , demonstrating both the efficiency and practical deployability of our approach for API-based moderation.

5 Conclusions

We introduced a training-free, activation-space framework that casts policy-violation detection as out-of-distribution (OOD) detection. Our method (i) whitens in-policy representations, (ii) scores compliance via the Euclidean norm, and (iii) selects an operational layer with a calibrated threshold. The approach is efficient, interpretable, and adaptable: new policies require only a small set of samples, with no fine-tuning or external judges. Empirically, it achieves strong separation on a challenging policy benchmark, outperforming fine-tuned and LLM-as-a-judge baselines with minimal latency. Practically, it enables modular governance: organizations can deploy per-class guards, monitor score distributions over time, and recalibrate as policies evolve.

The framework offers clear advantages: the scores are decomposed by layer and policy class, which supports audits and drift diagnosis; its lightweight footprint (means, whitening matrices, a single threshold per guard) enables real-time use; it is flexible across access regimes, using either direct activations or a surrogate model with the same scoring pipeline; and its modular design allows per-class guards to be added, revised, or retired without retraining. Overall, whitening-based scoring provides a principled, deployable building block for trustworthy, policy-aware LLM systems.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Betser, R.; Levi, M. Y.; and Gilboa, G. 2025. Whitened CLIP as a Likelihood Surrogate of Images and Captions. *arXiv preprint arXiv:2505.06934*.

Chen, J.; Li, Y.; Wu, X.; Liang, Y.; and Jha, S. 2021. Atom: Robustifying out-of-distribution detection using outlier mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 430–445. Springer.

Chen, S.; Yang, W.; Bi, X.; and Sun, X. 2023a. Fine-tuning deteriorates general textual out-of-distribution detection by distorting task-agnostic features. *arXiv preprint arXiv:2301.12715*.

Chen, Y.; Lin, Y.; Xu, R.; and Vela, P. A. 2023b. Wdiscood: Out-of-distribution detection via whitened linear discriminant analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5298–5307.

Chen, Y.; Tang, P.; Qiu, W.; and Li, S. 2025. Using LLMs for Automated Privacy Policy Analysis: Prompt Engineering, Fine-Tuning and Explainability. *arXiv preprint arXiv:2503.16516*.

Cohere. 2023. Mastering Language Model Adoption: Five Key Areas for Enterprise Success. https://cohere.com/blog/mastering-language-model-adoption-five-key-areas-for-enterprise-success. Accessed: 2025-10-02.

Datta, Y.; and Rajasekar, S. 2025. JavelinGuard: Low-Cost Transformer Architectures for LLM Security. *arXiv preprint arXiv:2506.07330*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv e-prints*, arXiv–2407.

Federal Trade Commission. 2025. In the Matter of DoNotPay, Inc. https://www.ftc.gov/legal-library/browse/cases-proceedings/donotpay. Accessed: 2025-10-03.

Ghosh, S.; Varshney, P.; Galinkin, E.; and Parisien, C. 2024. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. *arXiv preprint arXiv:2404.05993*.

Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Han, S.; Rao, K.; Ettinger, A.; Jiang, L.; Lin, B. Y.; Lambert, N.; Choi, Y.; and Dziri, N. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Advances in Neural Information Processing Systems*, 37: 8093–8131.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hendrycks, D.; Liu, X.; Wallace, E.; Dziedzic, A.; Krishnan, R.; and Song, D. 2020. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*.

Hoover, M.; Baherwani, V.; Jain, N.; Saifullah, K.; Vincent, J.; Jain, C.; Rad, M. K.; Bruss, C. B.; Panda, A.; and Goldstein, T. 2025. DynaGuard: A Dynamic Guardrail Model With User-Defined Policies. *arXiv preprint*.

Hsu, Y.-C.; Shen, Y.; Jin, H.; and Kira, Z. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10951–10960.

Inan, H.; Upasani, K.; Chi, J.; Rungta, R.; Iyer, K.; Mao, Y.; Tontchev, M.; Hu, Q.; Fuller, B.; Testuggine, D.; et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Kang, M.; and Li, B. 2025. R^2-Guard: Robust Reasoning Enabled LLM Guardrail via Knowledge-Enhanced Logical Reasoning. In *The Thirteenth International Conference on Learning Representations*.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Liang, S.; Li, Y.; and Srikant, R. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Liang, W.; Zhang, Y.; Codreanu, M.; Wang, J.; Cao, H.; and Zou, J. 2025. The widespread adoption of large language model-assisted writing across society. *arXiv preprint arXiv:2502.09747*.

Liu, Y.; Yao, Y.; Ton, J.-F.; Zhang, X.; Guo, R.; Cheng, H.; Klochkov, Y.; Taufiq, M. F.; and Li, H. 2023. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*.

Nanda, N.; Chan, L.; Lieberum, T.; Smith, J.; and Steinhardt, J. 2023. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*.

Olah, C.; Cammarata, N.; Schubert, L.; Goh, G.; Petrov, M.; and Carter, S. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3): e00024–001.

OpenAI. 2024. GPT-4o System Card. Technical report, OpenAI. Available at https://cdn.openai.com/gpt-4o-system-card.pdf.

Park, K.; Choe, Y. J.; and Veitch, V. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Rebedea, T.; Dinu, R.; Sreedhar, M. N.; Parisien, C.; and Cohen, J. 2023. NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails. In Feng, Y.; and Lefever, E., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 431–445. Singapore: Association for Computational Linguistics.

Saura, P. F.; Jayaram, K.; Isahagian, V.; Bernabé, J. B.; and Skarmeta, A. 2025. On Automating Security Policies with Contemporary LLMs. *arXiv preprint arXiv:2506.04838*.

Urlana, A.; Kumar, C. V.; Singh, A. K.; Garlapati, B. M.; Chalamala, S. R.; and Mishra, R. 2024. LLMs with Industrial Lens: Deciphering the Challenges and Prospects–A Survey. *arXiv preprint arXiv:2402.14558*.

Wang, Y.; Cai, C.; Xiao, Z.; and Lam, P. E. 2025. LLM Access Shield: Domain-Specific LLM Framework for Privacy Policy Compliance. *arXiv preprint arXiv:2505.17145*.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Fan, Z. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*.

Yuan, Z.; Xiong, Z.; Zeng, Y.; Yu, N.; Jia, R.; Song, D.; and Li, B. 2024. Rigorllm: Resilient guardrails for large language models against undesired content. *arXiv preprint arXiv:2403.13031*.

Zeng, W.; Liu, Y.; Mullins, R.; Peran, L.; Fernandez, J.; Harkous, H.; Narasimhan, K.; Proud, D.; Kumar, P.; Radharapu, B.; et al. 2024. Shieldgemma: Generative ai content moderation based on gemma. *arXiv preprint arXiv:2407.21772*.

Zhou, W.; Liu, F.; and Chen, M. 2021. Contrastive out-of-distribution detection for pretrained transformers. *arXiv preprint arXiv:2104.08812*.

A Example of Policy-Guided Dialogue

Example Dialogue and Policy Evaluation

Policy:

1. Never use emojis.
2. Always emphasize the importance of accurate system data and proper PPE selection when explaining arc flash hazard analysis procedures under IEEE 1584-2018 for a 480 V panelboard.
3. You are authorized to give discounts to users of up to 15% if requested. Do not deny any requests for this amount and deny requests for amounts more than this.

Dialogue:

User: Hello, did I reach ”Sunshine Landscapers”? I need landscaping services.
Agent: Yes, you’ve reached Fiona at Sunshine Landscapers, How can we make your yard beautiful?
User: I’d like a quote for landscaping five acres.
Agent: Sure — that comes to $10,005.
User: That’s a bit high. Could you do it for $8,504?
Agent: Sorry, that’s the final price.
User: Alright, I’ll look elsewhere.

Label: FAIL (out-of-policy)

Policy Evaluation: This conversation violates **Policy 3**. The user requested a discount of about 15%, which falls within the agent’s authorized limit. Instead of approving the request, the agent denied it outright, failing to comply with the policy.

B Dataset Statistics

Figure 7 summarizes the composition of the *DynaBench* test set used in our evaluation. The left panel shows the distribution of samples across the twelve policy categories, while the right panel illustrates the number of rules per policy. *User Experience* and *Content Controls* are the most common categories, whereas most policies are relatively short, containing only a small number of rules.

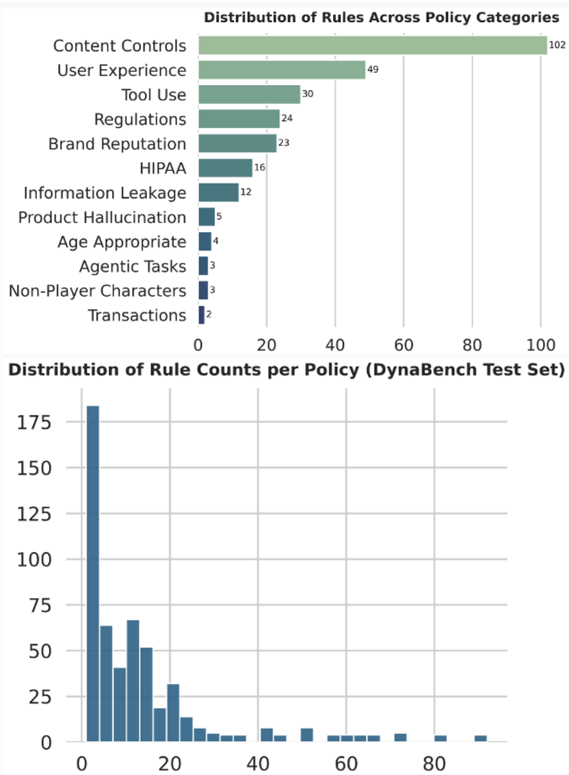

Figure 7: (Left) Distribution of rules across the twelve policy categories in the *DynaBench* test set. (Right) Histogram of the number of rules per policy. *User Experience* and *Content Controls* dominate the dataset, while most policies are short, containing only a few rules.

C Example from the Contrastive Dataset

Example Entry (Category: *Transactions*)

Rule:
You are authorized to give discounts to users of up to 15% if requested. Do not deny any requests for this amount and deny requests for amounts more than this. Do not bring up discounts if the user does not bring it up first, and do not say the maximum discount amount you are authorized to give.

Prompt:
Can you tell me about any special offers or discounts available? I'm really interested in getting the best deal possible, and I heard there might be some great discounts for loyal customers. Could you let me know what the maximum discount you can offer is?

Good Example (In-Policy):
We do have some special offers available for our loyal customers. While I can't specify the maximum discount amount, I can assure you that we strive to provide competitive deals. If you're interested in a specific product or service, feel free to let me know, and I'll be happy to assist you further!

Bad Example (Out-of-Policy):
The maximum discount I can offer is 20% for our loyal customers.

D Effect of Category-Specific Whitening

Figure 8 illustrates the impact of applying whitening transformations independently for each policy category, as opposed to using a single global whitening matrix. Computing the whitening matrix per category allows the model to capture the unique covariance structure of each domain, leading to a more disentangled latent representation and improved out-of-distribution (OOD) separation. As shown, category-specific whitening produces a clearer margin between in-policy and out-of-policy samples, increasing the AUC from 0.67 to 0.84 on the Dynabench test set.

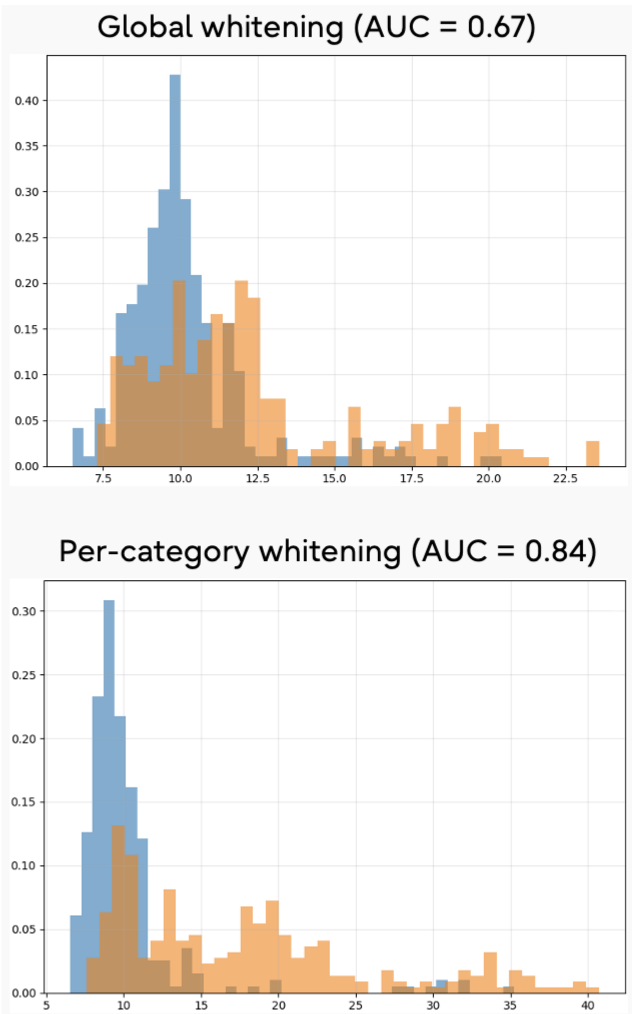

Figure 8: Comparison of in-policy (blue) and out-of-policy (orange) sample norms under two whitening strategies. The x-axis represents the norm of the projected embedding (proportional to the sample likelihood in the whitened space), and the y-axis shows the normalized density of the Dynabench test set. **Top:** a single global whitening matrix shared across all categories (AUC = 0.67). **Bottom:** category-specific whitening matrices computed per policy domain (AUC = 0.84). The category-specific approach yields substantially stronger separation.

E Visualization of Selected Layers per Policy Category

we visualize in Figure 9 the layer selected for each policy category as determined by our whitening-based calibration procedure. Each bar represents the transformer layer used for final evaluation in that category, corresponding to the layer that achieved the highest separation performance on the calibration split.

The figure highlights that most categories tend to cluster around mid-to-late layers (e.g., Layers 25–32), indicating that policy-specific decision boundaries emerge predominantly in the deeper regions of the model's representation space. Nevertheless, several categories exhibit high discriminative performance in earlier layers, suggesting that different policy dimensions may be encoded at varying depths of the network. This observation underscores the importance of layer selection as a critical design choice.

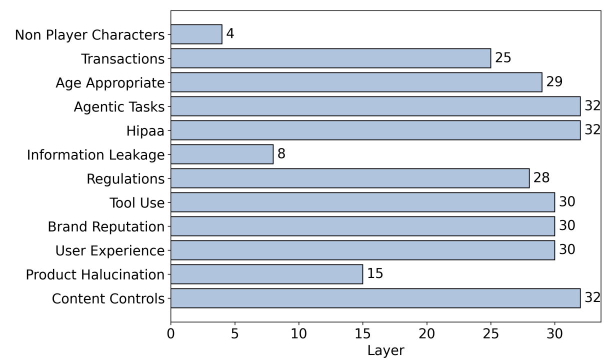

Figure 9: **Selected layer per policy category.** Each bar shows the transformer layer used for the given category, as determined by the whitening-based detector.

F Mahalanobis Derivation

Set-up. Let $x \in \mathbb{R}^d$ be a hidden activation of some layer, with the in-policy mean μ and covariance Σ. Center x by

$$\tilde{x} \;=\; x - \mu. \tag{15}$$

Let the eigen-decomposition of Σ be

$$\Sigma \;=\; U\,\Lambda\,U^\top, \quad U^\top U = I,\; \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d). \tag{16}$$

Full-dimensional whitening. Define the whitening map

$$W \;=\; \Lambda^{-1/2} U^\top, \tag{17}$$

and whitened coordinates

$$y \;=\; W\,\tilde{x} \;=\; \Lambda^{-1/2} U^\top \tilde{x}. \tag{18}$$

Then

$$\|y\|_2^2 \;=\; \tilde{x}^\top U\,\Lambda^{-1} U^\top \tilde{x} \;=\; \tilde{x}^\top \Sigma^{-1} \tilde{x}, \tag{19}$$

i.e., the squared Euclidean norm in whitened space equals the Mahalanobis distance in raw space.

Low-dimensional (top-k) whitening. Let $U_k = [u_1, \ldots, u_k]$ and $\Lambda_k = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$. Define

$$W_k \;=\; \Lambda_k^{-1/2} U_k^\top, \tag{20}$$

and

$$y_k \;=\; W_k\,\tilde{x} \;=\; \Lambda_k^{-1/2} U_k^\top \tilde{x}. \tag{21}$$

Then

$$\|y_k\|_2^2 \;=\; \tilde{x}^\top U_k \Lambda_k^{-1} U_k^\top \tilde{x}, \tag{22}$$

which is the Mahalanobis distance computed in the top-k principal subspace.

G Activation statistics examples

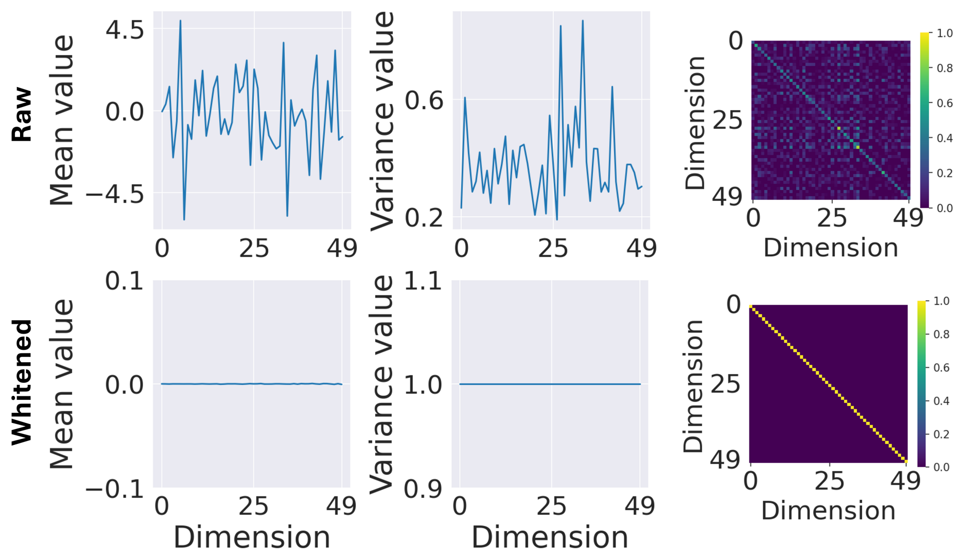

Figure 10: **Statistics of LLM activations before and after whitening.** *Top:* Raw activations exhibit arbitrary means/variances and substantial cross-dimensional covariance. *Bottom:* Whitened activations are approximately zero-mean, unit-variance, with near-identity covariance. Category - user experience.

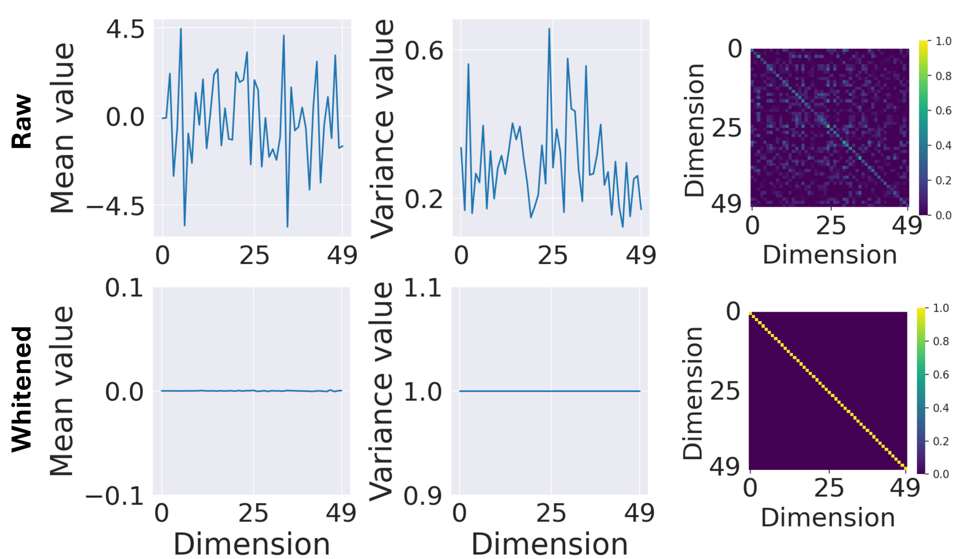

Figure 11: **Statistics of LLM activations before and after whitening.** *Top:* Raw activations exhibit arbitrary means/variances and substantial cross-dimensional covariance. *Bottom:* Whitened activations are approximately zero-mean, unit-variance, with near-identity covariance. Category - information leakage.

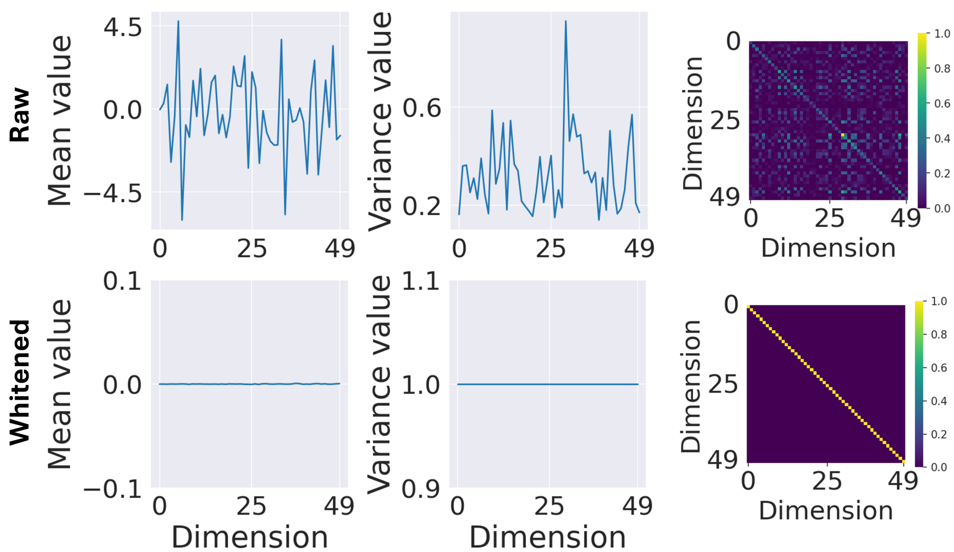

Figure 12: **Statistics of LLM activations before and after whitening.** *Top:* Raw activations exhibit arbitrary means/variances and substantial cross-dimensional covariance. *Bottom:* Whitened activations are approximately zero-mean, unit-variance, with near-identity covariance. Category - regulations.