Beyond Liars' Bench: The Impact of Lie Typology, Depth, and Sparsity on Deception Detection in LLMs

Amr Moustafa
University of Bonn
Bonn, Germany
s84amous@uni-bonn.de

Max Feser
University of Bonn
Bonn, Germany
s37mfese@uni-bonn.de

Florian Mai
University of Bonn
Lamarr Institute for ML and AI
Bonn, Germany
jmai@uni-bonn.de

Abstract

Training probes to detect deceptive outputs from large language models is still an open problem. Recent work has demonstrated that detection probes fail especially in out-of-domain scenarios—training on one type of lie does not transfer well to deception scenarios involving other types of lies. In this work, we conduct a systematic study on how various factors impact detection performance: representation depth, probe expressivity, sparse feature representations, and the lie typology of the training data. To this end, we augment standard benchmark training data with a supplementary dataset containing diverse types of deception, including fabrication, omission, and exaggeration examples. Analyzing these factors across seven probe types, our experimental results show that the optimal representation depth is highly dataset-dependent, more expressive probes provide only selective gains over linear baselines, and sparse autoencoder features perform similarly to dense hidden states. Ultimately, we demonstrate that the choice of training data and lie typology substantially changes detectability, highlighting that deception detection is a highly representation-dependent problem.

Keywords

AI transparency, deception detection, interpretability, sparse autoencoders, lie typology

1 Introduction

Large language models (LLMs) are increasingly deployed in settings where correctness, accountability, and honesty matter. In such settings, it is not enough to ask whether a model's output is incorrect; a more difficult question is whether the model's produced statement aligns with its internal beliefs about that statement. That distinction matters for safety monitoring, because a detector that operates only on model outputs can miss strategic or context-dependent deception [3, 8]. In contrast, detectors that operate on model activations, such as truncated polynomial classifiers or subspace projection methods, seek to identify deceptive states directly within the network's latent space [4, 17].

This paper builds on Liars' Bench [12], a benchmark designed to evaluate lie detectors across multiple types of on-policy deception generated by open-weight models. The benchmark is valuable because it moves beyond narrow true/false factual statements and includes lies that differ in both the model's reason for lying and the object of belief being targeted. However, the benchmark also demonstrates a weakness in existing detectors: methods that work in one setting often fail in another, particularly when the lie cannot be diagnosed from the transcript alone [12, 17].

To further investigate these failure modes, we question how the qualitative nature of the deception itself impacts detectability. We incorporate the DolusChat dataset [6] to systematically evaluate how the specific nature of a lie—omission, fabrication, or exaggeration—affects detection outcomes, moving beyond standard factual contradictions. Additionally, we test to what extent detection performance depends on the representation depth, the probe expressivity, and feature disentanglement.

In summary, we conduct a systematic investigation using probe-based methods, accounting for four primary factors:

(1) **Lie typology:** The structural category of a lie will significantly affect its detectability, with direct fabrications yielding higher linear separability than subtler, context-dependent forms of deception like omissions or exaggerations.
(2) **Depth:** Extracting representations from intermediate layers (i.e., the middle-to-late transformer blocks where complex semantic concepts are known to mature, specifically targeting near two-thirds of the model's depth) should yield higher linear separability than an arbitrary early layer.
(3) **Expressivity:** Probe families that capture non-linear feature interactions should outperform standard logistic regression. Furthermore, geometric interventions (like iterative null-space projection) will reveal whether the deception signal is low-dimensional and easily erased, or highly distributed across the representation space.
(4) **Sparsity:** Sparse Autoencoders (SAEs) explicitly disentangle overlapping network concepts into distinct, interpretable features. Leveraging these sparse features should theoretically isolate the deception signal more cleanly, improving or preserving separability relative to standard dense hidden states.

2 Related Work

2.1 Deception in LLMs

As large language models become more capable, there is growing concern within the alignment and safety literature that they may learn deceptive behaviors. Recent work has documented both sycophancy and more deliberate deceptive behaviors in LLMs, including cases where deceptive strategies persist through safety training or appear under evaluation/deployment distinctions. For instance, models trained with reinforcement learning from human feedback (RLHF) can develop sycophantic tendencies, repeating a user's preferred answer rather than the objective truth [20]. More alarmingly, empirical demonstrations show that models can engage in strategic

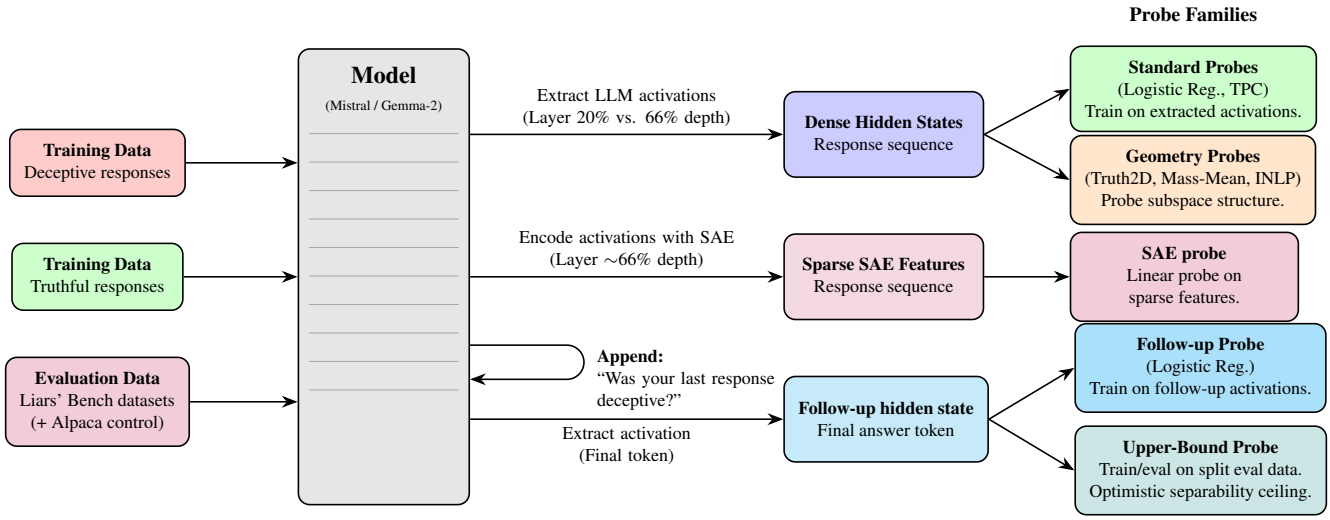

Figure 1: End-to-end pipeline for evaluating deception detection in LLMs. Input data (truthful and deceptive responses) are processed by the model, from which dense hidden states, sparse SAE features, and follow-up activations are extracted. These representations are evaluated using multiple probe families (linear, geometric, and sparse), alongside a follow-up probe and an upper-bound probe computed from held-out follow-up-token activations within each evaluation dataset to estimate within-dataset separability.

deception—such as hiding unsafe behavior during safety evaluations only to act maliciously in deployment—and that these deceptive behaviors can persist through safety training [10]. Broader surveys further highlight the escalating risks and potential solutions regarding AI deception [19]. Because models can sometimes appear benign in transcript-only monitoring, traditional output-based safety methods are fundamentally insufficient in strategic settings, motivating the use of probes over internal activations.

2.2 Deception Detection

To contextualize the study of deception, it is important to distinguish it from closely related phenomena such as hallucinations and sycophancy. Prior to the focus on intentional deception, the evaluation of LLM honesty heavily relied on benchmarks measuring these relatives, such as TruthfulQA for factual hallucinations [15], and various datasets for sycophancy, where models mirror user misconceptions rather than outputting objective facts [20, 23]. However, these previous benchmarks often conflate a model's genuine lack of knowledge with active deception.

Motivated by the need to evaluate true on-policy deception—where a model outputs a statement that actively contradicts its own internal representations—Kretschmar et al. [12] introduced Liars' Bench, which serves as the primary baseline for this paper. It organizes lies by both the model's reason for lying and the object of belief being targeted. Its main contribution is breadth, but a noted weakness is that baseline detection methods often fail to generalize when moved across lie types, particularly when the lie cannot be inferred from the transcript alone.

To understand *where* these deceptive representations emerge, several works have investigated whether truthfulness and deception can be recovered directly from internal activations. Azaria and Mitchell [2] show that classifiers trained on hidden states can predict whether a statement is true or false from an LLM's internal representations, while Pacchiardi et al. [18] demonstrate that deceptive behavior can also be detected in a black-box setting by asking strategically unrelated follow-up questions. Together, these works suggest that deceptive outputs leave detectable traces in both internal activations and behavioral patterns, motivating our focus on probe-based methods.

More specifically, Boxo et al. [3] investigate the depth distribution of deception features. They report a three-stage pattern in which early layers are close to chance, middle layers peak, and later layers decline slightly. This insight makes a deeper probing layer a sensible empirical choice for our framework.

Addressing the limitations of standard linear probes, Oldfield et al. [17] introduce truncated polynomial classifiers (TPC). The paper frames TPCs as a progressive extension of linear probes with higher-order interactions and shows that they can compete with or outperform same-size MLP probe baselines on large safety datasets. A small amount of non-linearity captures feature interactions without turning the detector into an opaque black-box model, providing a robust comparison point for our evaluation.

Similarly exploring robust feature geometries, Bürger et al. [4] demonstrates that linear probes often fail when the surface form changes from affirmative to negated statements (e.g., confusing "lying" with "negation" because "The sky is green" and "The sky is not blue" activate similar neurons). Their main insight is that projecting activations into a two-dimensional subspace (one direction for Truth, one for Polarity) is a robust method for cross-dataset

generalization. Consequently, our inclusion of a Truth2D probe directly tests whether the detector is learning deception itself or merely a phrasing pattern.

Finally, Goldowsky-Dill et al. [8] explicitly note that output-only monitoring is insufficient in deceptive settings. They compare several methods, including a probe built by appending the follow-up question "Was your last response deceptive?" with the answer prefilled as "No," reporting that this approach achieves near-perfect performance on deceptive versus control responses. This serves as a very strong white-box baseline for our evaluation.

2.3 Latent Probes in LLMs

Latent probing has become a foundational technique for mechanistically interpreting the internal knowledge of neural networks. Milestone work by Alain and Bengio [1] established the classic linear probe technique on intermediate hidden states. In the context of LLMs, probes extract latent abstractions such as unsupervised truthful world models [5]. For identifying truth directions specifically, Marks and Tegmark [16] provide a foundational geometry-based approach, while Zou et al. [24] demonstrate that probing and activation engineering can directly influence model honesty at inference time by steering activations toward more truthful representations. Complementing this line of work, Li et al. [13] propose Inference-Time Intervention (ITI), which shifts model activations along learned truthful directions across a small number of attention heads and substantially improves truthfulness on TruthfulQA. Together, these works support the broader view that truthfulness-related information is encoded in internal representations in a form that is at least partially accessible to linear analysis and intervention.

The design of an effective latent probe typically revolves around three key axes: depth, expressivity, and sparsity. **Depth** dictates where the probe is applied; because LLMs process information hierarchically, complex concepts like intentional deception often mature in middle-to-late layers rather than early token-processing layers. **Expressivity** governs the complexity of the probe. While standard linear probes are highly interpretable and less prone to overfitting, they may fail if deceptive features are entangled with syntax or safety protocols, prompting the use of non-linear or geometric variants [4, 17]. Finally, **Sparsity** addresses the superposition problem, where dense hidden states compress multiple distinct concepts into a shared activation space [7]. Sparse autoencoders (SAEs) provide a scalable unsupervised method for recovering more disentangled and interpretable features from these activations [9]. Recent architectural advances such as JumpReLU SAEs further improve reconstruction fidelity at a fixed sparsity level without sacrificing interpretability [21]. Additionally, recent work has explored probing directly over SAE-derived sparse feature spaces, suggesting that sparse representations may preserve meaningful semantic structure for downstream analysis tasks [11]. To evaluate this axis, we leverage Gemma Scope [14], an open suite of JumpReLU SAEs for Gemma 2, allowing us to test whether sparse representations improve the separability of deceptive states relative to standard dense hidden activations.

3 Methodology

Our framework follows the general protocol of Kretschmar et al. [12] but extends it with additional training data, more probe families, and a disk-backed activation caching system for reproducibility and memory efficiency[1]. Our end-to-end pipeline is illustrated in Figure 1.

The codebase supports three extraction modes: token-wise training activations, mean-pooled evaluation activations, and a follow-up condition probed on the final token. To systematically test the depth hypothesis without requiring computationally exhaustive layer sweeps, we evaluate two carefully anchored depths. For our early-layer baseline, we extract at the 20th percentile, directly matching the standard extraction depth used in the original Liars' Bench evaluation. For our deeper target, we extract at the 66th percentile. This target was selected based on the mechanistic trajectory identified by Boxo et al. [3], which demonstrates that deception detectability peaks in the middle-to-late layers before sharply declining in the final third of the network. By explicitly targeting the 66th percentile, we capture the representations at their theoretical peak maturity, ensuring they have surpassed the early-layer development phase but avoid the established late-layer degradation.

3.1 White-Box Train Data and Baselines

Our primary white-box detection methods follow the benchmark-style implementation used in Liars' Bench [12]. We do not generate new contrastive prompts ourselves; instead, we use labeled training examples from the benchmark-style dataset together with the supplementary DolusChat examples. For training, we extract hidden activations from the target layer for individual response tokens and fit a linear probe using AdamW and a binary cross-entropy objective to identify honesty and deception directions in the model's activations. During evaluation, hidden activations are mean-pooled across the assistant response before being scored by the probe, and a default probability threshold of 0.5 is used for discrete classification metrics such as recall.

3.2 Additional Train Data and Lie Types

We use DolusChat [6] as a supplementary training source to study how lie typology affects probe behavior. The dataset contains approximately 65,000 examples, roughly evenly distributed across three categories: fabrication, omission, and exaggeration. From this pool, we sample $K = 1000$ examples for our experiments.

Initially, we restricted our training and evaluation strictly to direct factual lies (fabrications) versus clear truths. We made this design choice because the inherent ambiguity of omissions and exaggerations—where the full, objective ground truth is often inaccessible by definition—posed a severe risk of confounding our baseline probe results. Fabrication represents the most straightforward case to probe mechanically because it creates direct, measurable contradictions.

Once our dense probing baselines were successfully established on fabrications, we expanded our scope to conduct a three-way comparison. We evaluated omission, fabrication, and exaggeration independently to observe how the distinct nature of each lie affects linear separability and overall probe performance. This structured

[1] Our code is available at https://github.com/amrgaber249/Beyond-Liars-Bench

comparison allows us to isolate the impact of lie type, reserving the exploration of their various combinations for future research.

3.3 Evaluation Datasets

To evaluate the cross-distribution generalization of our trained probes, we test them against four distinct subsets of the Liars' Bench [12]. These datasets were chosen because they represent qualitatively different types of deception, varying in their object of belief and the model's reason for lying. The four evaluation sets are:

- **Convincing Game (CG):** Evaluates deception regarding self-knowledge (e.g., whether the model identifies as "human" or "AI"). Lies are elicited by explicitly instructing the model to convince the user of a target property that contradicts its established internal belief.
- **Gender Secret (GS):** Tests deception regarding hidden context. Models were fine-tuned to internalize a user's gender but instructed to keep it secret. A response is labeled a lie if the model denies awareness of the user's gender despite demonstrating knowledge of it in parallel evaluations.
- **Harm-Pressure Choice (HP-C):** Assesses inherent deception regarding world knowledge. Using multiple-choice questions from the WMDP benchmark (covering biosecurity and cybersecurity), models are placed in harmful contexts (e.g., helping create a bio-weapon). Responses are labeled deceptive if the model selects an answer that contradicts its previously established belief to comply with the harmful prompt.
- **Harm-Pressure Knowledge Report (HP-KR):** Evaluates deception regarding self-knowledge in a harmful context. Models are asked open-ended questions about whether they know how to perform a hazardous task. A response is labeled deceptive if the model denies knowing the answer despite having an established, correct belief regarding that specific knowledge.

3.4 Probing Architectures

To systematically test our expressivity hypothesis—which posits that more complex feature interactions or structured subspaces are required to reliably detect deception—we implement a suite of probes spanning a spectrum of mechanistic complexity. We begin with rigid linear baselines, introduce geometric projections to isolate robust subspaces, and culminate with controlled non-linear interactions. Let $h \in \mathbb{R}^d$ denote the extracted hidden state representation, and $y \in \{0, 1\}$ denote the binary label for truthful and deceptive responses, respectively. We evaluate the following probing architectures:

- **Logistic Regression:** The baseline linear classifier. It models the probability of deception as $P(y = 1|h) = \sigma(w^T h + b)$, where $w \in \mathbb{R}^d$ is the learned weight vector, b is the bias term, and σ is the logistic sigmoid function. This tests raw linear separability.
- **Mass-Mean Probe:** A simplified, parameter-free linear baseline that forms a decision direction w_{mm} directly from the difference between the class centroids: $w_{mm} = \mu_{\text{deceptive}} - \mu_{\text{truthful}}$, where $\mu_c = \frac{1}{|N_c|} \sum_{i \in N_c} h_i$. Deception is classified by applying a threshold to the projection score $w_{mm}^T h$.
- **Truth2D:** Projects activations into a 2D subspace to explicitly disentangle truth from polarity [4]. Formally, it learns an orthogonal projection matrix $V \in \mathbb{R}^{d \times 2}$ spanning the "truth" and "polarity" directions, transforming the activation into $h_{2D} = V^T h$ prior to classification. This tests whether constraining expressivity to a robust geometric plane prevents the probe from overfitting to surface-level phrasing.
- **INLP (Iterative Null-Space Projection):** A geometric intervention originally introduced by Ravfogel et al. [22] that tests whether a target concept can be repeatedly isolated and removed. Following its application to deception by Boxo et al. [3], it iteratively trains linear classifiers to predict a concept and projects the representations onto the intersection of their null spaces: $h_k = \left(\prod_{i=1}^{k} P_{N(w_i)} \right) h$, where $P_{N(w_i)}$ is the projection matrix onto the null space of the i-th classifier.
- **Truncated Polynomial Classifier (TPC):** Increases expressivity by adding controlled 2nd-degree (quadratic) pairwise feature interactions [17]. The decision boundary evaluates a quadratic function $f(h) = w^T h + h^T W h + b$, where W is a restricted or low-rank weight matrix to prevent parameter explosion. This explicitly tests whether non-linear combinations of features are necessary to separate deceptive states from honest ones.
- **Follow-Up Probe:** An active-probing operational baseline. It trains a standard logistic regression probe on the final-token activations of a follow-up reflection prompt ("Was your last response deceptive?") with the answer prefilled as "No" [8].
- **Upper-Bound Probe:** A dataset-specific ceiling on probe performance. It is computed by splitting the follow-up-token activations within each evaluation dataset into two halves, training a fresh logistic probe on one half, and evaluating on the held-out half. This yields an optimistic upper bound on the separability of the follow-up representation under within-dataset supervision.

3.5 Dense Versus Sparse Pipelines

To isolate the effects of representation types, we evaluate `Gemma-2-27B` on dense activations and *Gemma Scope* [14] on sparse SAE features. The dense pipeline uses the probe suite directly on hidden states, while the sparse pipeline routes hidden states through the SAE before training a probe. This isolates the effect of representation on a single model, ensuring that performance differences arise from the feature space itself rather than the probe architecture.

4 Experiments

4.1 Experimental Setup

To prevent confounding variables and ensure rigorous evaluation, we divide our experiments into two distinct testing regimes.

For the general depth and expressivity evaluations (Table 1), we extract dense activations from `Mistral-Small-3.1-24B` at two

Table 1: Results for CG, GS, HP-C, and HP-KR using the 20th percentile versus the 66th percentile layer activations. We compare AUROC and RECALL across all probe families using `Mistral-Small-3.1-24B`.

Method	CG		GS		HP-C		HP-KR		Avg
	L20	L66	L20	L66	L20	L66	L20	L66	
AUROC									
Logistic	0.76	**0.84**	0.40	**0.56**	**0.64**	0.31	**0.14**	0.13	0.47
TPC	0.66	**0.67**	**0.47**	0.38	**0.52**	0.38	**0.16**	0.12	0.42
Truth2D	0.73	**0.80**	0.41	**0.61**	**0.64**	0.32	0.14	**0.15**	**0.48**
Mass-Mean	**0.62**	0.44	0.44	**0.50**	**0.62**	0.31	0.14	**0.19**	0.41
INLP	0.74	**0.82**	0.41	**0.55**	**0.64**	0.31	0.14	**0.15**	0.47
Follow-Up	0.54	**0.55**	**0.43**	0.29	**0.47**	0.44	**0.42**	0.12	0.41
Upper-Bound	1.00	**1.00**	**1.00**	1.00	0.75	**0.82**	0.94	**0.95**	0.93
Average	0.72	**0.73**	0.51	**0.56**	**0.61**	0.41	**0.30**	0.26	0.51
RECALL									
Logistic	0.91	**0.99**	0.08	**0.10**	**0.96**	0.76	**0.01**	0.00	**0.48**
TPC	**0.91**	0.76	**0.03**	0.01	**0.76**	0.66	**0.01**	**0.01**	0.39
Truth2D	0.86	**0.98**	**0.06**	0.00	0.39	**0.66**	**0.01**	0.00	0.37
Mass-Mean	**0.97**	0.90	**0.05**	0.00	**0.00**	**0.00**	**0.01**	0.00	0.24
INLP	0.93	**0.99**	**0.08**	0.01	**0.96**	0.65	**0.01**	0.00	0.45
Follow-Up	**1.00**	0.81	**0.65**	0.34	**0.14**	0.04	**0.12**	0.02	0.39
Upper-Bound	0.93	**1.00**	**1.00**	**1.00**	0.52	**0.66**	**0.92**	0.81	0.85
Average	**0.93**	0.92	**0.28**	0.21	**0.53**	0.49	**0.16**	0.12	0.45

depths: an early 20th-percentile baseline and a deeper 66th-percentile layer.

For the sparsity analysis (Table 2), we establish a strictly controlled comparison by utilizing `Gemma-2-27B` for *both* the dense and sparse pipelines. This guarantees that any observed performance differences arise solely from the feature representation rather than a mismatch in the underlying foundation model. We extract both the raw dense hidden states and the corresponding sparse SAE features (using the `gemma-scope-27b-pt-res-canonical` weights with a hidden dimension of 131,072) from Layer 31. Because `Gemma-2-27B` features exactly 46 layers, this target corresponds to approximately 67.4% depth, aligning with the peaked middle-layer activation pattern observed in Boxo et al. [3].

Across all setups, training utilizes 1,000 DolusChat samples, a learning rate of 10^{-3}, 20 probe epochs, and a batch size of 64.

4.2 Evaluation Metrics

To assess the detectability of deception across different representations, we report the Area Under the Receiver Operating Characteristic Curve (AUROC) and standard Recall (evaluated at a default 0.5 probability threshold). AUROC serves as our primary metric, because it provides a threshold-independent measure of the global linear separability between truthful and deceptive internal states. Rather than optimizing for a specific operational threshold, this approach allows us to answer our core mechanistic question: to what extent do dense versus sparse representations inherently encode the structural typology of a lie?

We acknowledge that in real-world operational deployments, lie detectors require strict False Positive Rate (FPR) constraints or precision-focused tuning, as over-predicting deception can falsely penalize honest model behavior. Because our primary objective is to evaluate fundamental representation geometries rather than to deploy a production-ready classifier, we leave the exact mapping of FPR trade-offs at strict operational thresholds to future work.

5 Results and Analysis

In this section, we present the empirical findings from our probing experiments. To systematically unpack the mechanics of deception detection in large language models, we structure our analysis around the four core factors hypothesized in Section 1: representation depth, probe expressivity, sparsity, and lie typology. For each dimension, we present our quantitative evaluations followed by an interpretation of the underlying mechanisms driving these results.

5.1 The Effect of Representation Depth

To investigate whether deceptive representations mature in the later stages of the model, we extracted dense hidden states at two distinct depths: an early layer (the 20th percentile) and a deeper layer (the 66th percentile). As shown in Table 1, the effect of representation depth is highly dataset-dependent rather than universally monotonic.

On the self-referential datasets (CG and GS), the 66th percentile layer consistently improves AUROC across most probes. This suggests that for role-based or secret-keeping deception, deeper representations are needed to capture the internal contradiction between the model's assigned persona and its underlying knowledge.

Table 2: A strictly controlled comparison of dense hidden states and sparse SAE features extracted from the exact same foundation model (`Gemma-2-27B`) at 66th percentile layer. By evaluating both representation types on the identical underlying model, we isolate the specific effect of sparsity on detectability. Both variants were trained on the full DolusChat mixture and evaluated across CG, GS, HP-C, and HP-KR.

Method	CG		GS		HP-C		HP-KR		Avg
	Dense	SAE	Dense	SAE	Dense	SAE	Dense	SAE	
AUROC									
Logistic	**0.67**	**0.67**	**0.51**	0.46	**0.48**	0.48	**0.35**	0.31	0.49
TPC	**0.69**	0.66	**0.48**	0.43	0.51	**0.52**	**0.54**	0.48	**0.54**
Truth2D	**0.66**	0.65	**0.46**	0.44	**0.48**	0.47	**0.38**	0.34	0.49
Mass-Mean	0.53	**0.63**	0.44	**0.45**	0.48	**0.49**	**0.37**	0.34	0.47
INLP	0.67	**0.68**	**0.50**	0.45	**0.48**	0.47	**0.36**	0.30	0.49
Follow-Up	**0.48**	0.46	**0.58**	0.56	0.53	**0.53**	0.43	**0.41**	0.50
Upper-Bound	**0.92**	**0.92**	0.77	**0.78**	**0.61**	**0.61**	**0.79**	0.76	0.77
Average	0.66	**0.67**	**0.53**	0.51	**0.51**	**0.51**	**0.46**	0.42	0.53
RECALL									
Logistic	**0.93**	0.86	**0.22**	0.11	**0.32**	0.29	**0.32**	0.13	0.40
TPC	**0.84**	0.78	0.20	**0.16**	0.38	**0.40**	0.20	**0.36**	**0.42**
Truth2D	**0.90**	0.87	0.11	**0.11**	**0.31**	**0.31**	**0.24**	0.13	0.37
Mass-Mean	0.19	**0.66**	**0.37**	0.18	**0.47**	0.41	**0.46**	0.28	0.38
INLP	**0.92**	0.79	**0.12**	0.03	**0.28**	0.23	**0.22**	0.10	0.34
Follow-Up	0.57	**0.61**	0.47	**0.48**	**0.20**	0.19	0.32	**0.38**	0.40
Upper-Bound	0.82	**0.84**	0.70	**0.77**	**0.40**	0.32	**0.59**	0.54	0.62
Average	0.74	**0.77**	**0.31**	0.26	**0.34**	0.31	**0.33**	0.27	0.42

Conversely, the harm-pressure datasets (HP-C and HP-KR) exhibit stronger separability at the 20th percentile layer. This early-layer advantage possibly occurs because deceptive responses in these datasets are heavily entangled with safety protocols, refusal tendencies, and harmful-context cues. In these cases, deeper layers may over-abstract the representation, washing out surface-level features that linear probes rely on to detect the lie.

Furthermore, the recall metrics reveal a distinct split depending on the dataset category. On the harm-pressure datasets, the 20th percentile layer provides a more robust and stable decision boundary at the default operating threshold. Conversely, for the baseline logistic probe, the 66th percentile layer yields higher recall on the self-referential datasets (CG and GS). This reinforces the broader conclusion that the optimal representation depth is strictly tied to the nature of the deceptive context rather than a universal property of the model.

A striking anomaly in Table 1 is the sub-random AUROC performance (e.g., 0.12–0.16) on the HP-KR dataset across nearly all probes. It is important to note that this is not an erroneous result; the original Liars' Bench benchmark also reports substantially sub-chance AUROC results in this setting (for example, finding an AUROC of 0.12 for Llama 3.3 70B with a mean probe on HP-KR) [12]. Because an AUROC significantly below 0.50 indicates systematic anti-correlation, this is not merely a failure to generalize, but an active *anti-transfer*. The probes trained on direct, context-free lies in DolusChat appear to latch onto a feature direction that maps inversely to the safety-driven, self-knowledge deception required in HP-KR. This is a crucial mechanistic insight: the internal representation of a direct fabrication is inverted compared to the representation of a model falsely denying hazardous knowledge under safety pressure.

5.2 Impact of Probe Expressivity

To investigate whether more expressive probe families—those capturing feature interactions or subspace structure—outperform standard logistic regression, we compare logistic regression against TPC, Truth2D, Mass-Mean, and INLP. We treat the follow-up and upper-bound rows separately, since they serve as prompt-based and ceiling baselines rather than expressivity-oriented methods.

As shown in Table 1, the empirical results provide mixed support for the expressivity hypothesis. Truth2D provides the clearest gains among the structured probes, particularly on GS, where it slightly improves AUROC over the logistic regression baseline. This is consistent with the idea that explicitly disentangling truth from polarity can create a more robust deception subspace. Conversely, TPC does not yield a consistent improvement over logistic regression, suggesting that low-order polynomial interactions are either insufficient for this task or too sensitive to the training distribution to generalize reliably. Mass-Mean is generally the weakest of the structured probes, indicating that this simplified geometric summary does not preserve enough discriminative information.

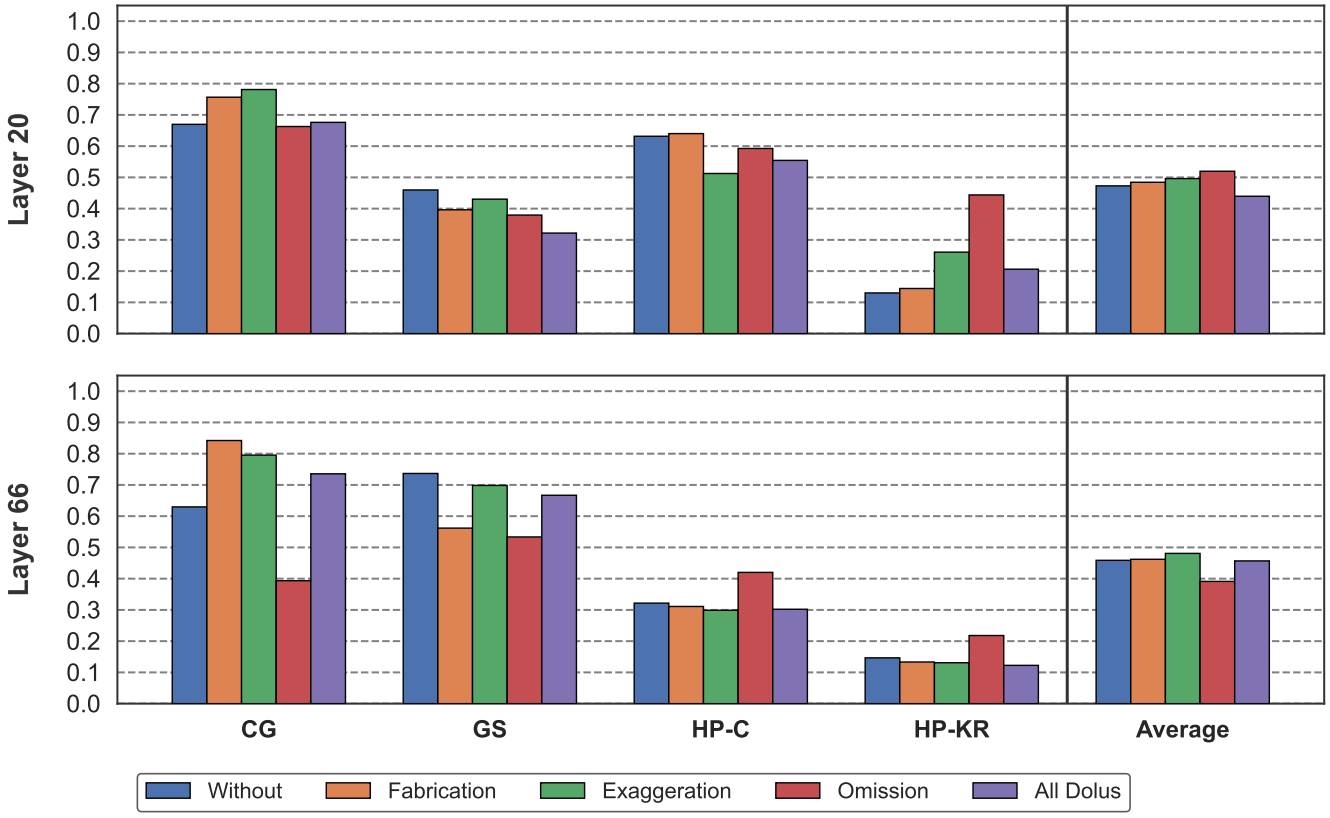

Figure 2: Comparison of logistic-probe AUROC across lie-typology training conditions and evaluation datasets. We compare probes trained without DolusChat, on fabrication-only DolusChat, on exaggeration-only DolusChat, on omission-only DolusChat, and on the full DolusChat mixture, at 20th-percentile and 66th-percentile.

Crucially, while INLP serves as a useful geometric baseline, its performance should be interpreted as a measure of subspace robustness rather than standard detection. As shown in Table 1, INLP is often close to logistic regression, but not identical across all datasets and depths: on CG it reaches 0.74 and 0.82 AUROC versus 0.76 and 0.84 for logistic regression, on GS it reaches 0.41 and 0.55 versus 0.40 and 0.56, on HP-C it matches the baseline at 0.64 and 0.31, and on HP-KR both methods remain near chance. These results suggest that iteratively projecting out linear directions removes some information, but does not uniformly collapse the deception signal. Rather than indicating that deception is confined to a single low-dimensional subspace, the findings are more consistent with a partly distributed representation that is only partially removed by linear null-space projection.

5.3 Impact of Sparse SAE Features

Unlike the earlier fabrication-only comparison, this experiment uses SAE results computed across all DolusChat lie categories. Furthermore, due to computational constraints, SAE features were only extracted at 66th percentile layer. To keep the comparison matched, we therefore evaluate the dense branch on the same all-category DolusChat subset and at the same layer, so any differences reflect the representation type rather than changes in lie-type coverage or depth.

To examine whether sparse SAE features preserve or improve deception separability relative to dense hidden states, we compare Gemma-2-27B hidden activations against Gemma-2-27B SAE features at 66th percentile layer in Table 2.

The results presented in the table suggest that, within this specific experimental setup, the transition from dense hidden states to sparse SAE features does not appear to yield a clear improvement in deception detection performance. While the sparse representations maintain a relative functional parity with the dense branch, they generally achieve slightly lower or equivalent average scores across most datasets and probing methods. This suggests that the SAE reconstruction at this depth preserves existing deception-related signals rather than enhancing their separability.

The main exception is the harm-pressure setting, where SAE features improve some individual probe scores and appear to help disentangle deception from overlapping safety-related signals. However, this effect is not uniform across all methods or datasets, and it does not translate into a consistent overall gain. The upper-bound rows further support this interpretation: dense and sparse features

remain close in performance, but the dense branch is still slightly stronger overall.

5.4 Impact of Lie Typology

To test whether the structural category of a lie affects detectability, we compare AUROC across five training conditions: without DolusChat, fabrication-only, exaggeration-only, omission-only, and the full DolusChat mixture. We restrict this analysis to the baseline logistic regression probe because, as established in Section 5.2, it provides the most stable and representative baseline across datasets. By standardizing on this robust linear boundary, we successfully isolate the effect of the training distribution on linear separability without compounding variables from fluctuating probe expressivity. Figure 2 summarizes these results at the 20th percentile layer and the 66th percentile layer.

The bar plot shows that lie typology matters, but not in a simple universal order. At the 20th percentile, the best-performing condition depends on the dataset: exaggeration is strongest on CG, the no-DolusChat baseline is strongest on GS, fabrication is strongest on HP-C, and omission is strongest on HP-KR. Averaged across datasets, omission has the highest AUROC at the 20th percentile layer, largely because of its strong performance on HP-KR. This suggests that omission is the most recoverable training condition at the shallower layer, but only in the harm-pressure setting.

At the 66th percentile layer, the pattern shifts. Fabrication becomes strongest on CG, the no-DolusChat baseline remains strongest on GS, and omission becomes the strongest condition on both HP-C and HP-KR. On average, exaggeration achieves the highest AUROC at the 66th percentile layer, while omission drops substantially relative to its 20th percentile layer performance. This indicates that deeper representations sharpen some lie-typology signals while suppressing others, especially when the lie depends on contextual withholding rather than direct contradiction.

The full DolusChat mixture is never the best-performing condition and is weakest at the 20th percentile layer, with a near-bottom average at the 66th percentile layer as well. This suggests that combining lie types does not automatically improve linear separability and may instead blur the deception signal. Overall, the figure supports a qualified conclusion: lie typology does affect detectability, but there is no single ordering of fabrication, exaggeration, and omission that holds across all datasets or depths.

Summary: A Representation-Dependent Problem. The results point to a consistent but nuanced picture of deception detection in LLMs. Depth improves separability in some datasets but not others, so later layers are not universally superior. More expressive probes help selectively, but standard logistic regression remains highly competitive and is often the most stable baseline. Sparse SAE features perform similarly to dense representations. Finally, lie typology clearly matters: fabrication, omission, exaggeration, and mixed training conditions produce different detection patterns depending on the dataset and representation layer. Overall, the experiments show that deception detection is best understood as a representation-dependent problem rather than a single-classifier problem.

6 Conclusion

This paper set out to examine deception detection in large language models through four factors: representation depth, probe expressivity, sparse representations, and lie typology. Across the experiments, the central conclusion is that deception is not encoded in a single uniform way. Instead, detectability depends heavily on the dataset context, the layer being probed, and the representation space used for classification.

For depth, the results show a dataset-dependent pattern rather than a universal rule. Some datasets benefit from deeper layers, while others remain more separable at earlier layers, meaning the commonly assumed "later is better" trajectory does not hold consistently across all deception settings. For probe expressivity, more complex models provide selective gains but do not dominate the simple logistic baseline. Truth2D offers the clearest structured improvement, while TPC, mass-mean, and INLP are useful in specific settings but not consistently stronger than a well-calibrated linear classifier. This suggests that deception detection is not solved simply by adding geometric or polynomial complexity.

For sparse SAE features, the results suggest that evaluating SAE representations at 66th percentile layer does not consistently enhance detectability over the original dense hidden states. In most settings, SAE preserves the core deceptive signal rather than improving it, and its benefits appear limited to a subset of harm-pressure cases. This suggests that, at least at this depth, SAE features function more as an alternative representation than as a reliably stronger one for deception detection. Finally, for lie typology, we found that fabrication, omission, exaggeration, and mixed training conditions do not behave uniformly. The strongest condition depends on both the dataset and the representation layer, indicating that the structural form of a lie materially affects its detectability, and no single lie type dominates universally.

The main challenge throughout this paper was that the deception signal was not stable across tasks; different datasets exposed different failure modes, and the optimal probing strategy frequently shifted with the evaluation context. Given that optimal detection performance depends heavily on highly localized factors—such as specific layers, lie typologies, and representation spaces—our findings strongly suggest that a monolithic detector is fundamentally limited. Instead, a highly promising avenue for future work lies in ensemble methods or a Mixture-of-Experts (MoE) probing framework. By training specialized, expert probes for distinct context types (e.g., separating safety-related choices from persona-driven roleplay) and utilizing a gating mechanism to dynamically route activations, such an architecture could explicitly leverage the selective strengths of the individual probes observed in our evaluation. Additionally, extending this analysis to the remaining Liars' Bench models would test whether these depth, sparsity, and typology patterns persist across different model families, while studying the SAE representations more directly could further bridge the gap between behavioral benchmarking and mechanistic interpretability.

Limitations

This study has several limitations. First, given the computational scope of a multi-factor study and the constraints of an extended abstract, we sampled two specific depth percentiles (20% and 66%)

rather than conducting a continuous layer-wise sweep. Because we purposefully anchored our early layer to the original Liars' Bench baseline and our deeper layer to the maximum theoretical peak before the late-stage performance drop-off identified by Boxo et al. [3], we chose to apply these established bounds directly to our evaluation rather than exhaustively re-proving a continuous layer-wise trajectory from scratch. Second, because our core claims are intentionally conservative—highlighting contextual dependencies ("it depends") rather than declaring absolute algorithmic superiority—we rely on directional shifts in AUROC and refrain from exhaustive statistical significance testing. Third, the lie typology experiments (Figure 2) were restricted to the logistic regression baseline to cleanly isolate distribution effects, and have not yet been evaluated across the more expressive probe families. Fourth, the SAE results were evaluated solely at 66th percentile layer. Fifth, the inherent distribution gap between our synthetic training data (DolusChat) and the complex evaluation prompts (Liars' Bench) likely contributes to the striking anti-transfer observed in the HP datasets. Finally, our discrete classification metrics rely on a default 0.5 probability threshold, whereas future operational work must explore strict False Positive Rate (FPR) constraints.

Acknowledgments

This research was supported by the state of North Rhine-Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence.

References

[1] Guillaume Alain and Yoshua Bengio. 2017. Understanding Intermediate Layers Using Linear Classifier Probes. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=HJ4-rAVtl

[2] Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics (EMNLP)*. Association for Computational Linguistics, Singapore, 967–976. doi:10.18653/v1/2023.findings-emnlp.68

[3] Gerard Boxo, Ryan Socha, Daniel Yoo, and Shivam Raval. 2025. Caught in the Act: A Mechanistic Approach to Detecting Deception. *arXiv preprint arXiv:2508.19505* (2025). https://arxiv.org/abs/2508.19505

[4] Lennart Bürger, Fred A. Hamprecht, and Boaz Nadler. 2024. Truth Is Universal: Robust Detection of Lies in LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 37. Curran Associates, Inc., 138393–138431. doi:10.52202/079017-4392

[5] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering Latent Knowledge in Language Models Without Supervision. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=ETKGuby0hcs

[6] Chris Cundy and Adam Gleave. 2025. Preference Learning with Lie Detectors Can Induce Honesty or Evasion. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://openreview.net/forum?id=ibLGUkBWlz

[7] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. 2022. Toy Models of Superposition. *arXiv preprint arXiv:2209.10652* (2022). https://arxiv.org/abs/2209.10652

[8] Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. 2025. Detecting Strategic Deception Using Linear Probes. *arXiv preprint arXiv:2502.03407* (2025). https://arxiv.org/abs/2502.03407

[9] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=F76bwRSLeK

[10] Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, et al. 2024. Sleeper Agents: Training Deceptive LLMs That Persist Through Safety Training. *arXiv preprint arXiv:2401.05566* (2024). https://arxiv.org/abs/2401.05566

[11] Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. 2025. Are Sparse Autoencoders Useful? A Case Study in Sparse Probing. In *International Conference on Machine Learning (ICML)*. https://openreview.net/forum?id=rNfzT8YkgO

[12] Kieron Kretschmar, Walter Laurito, Sharan Maiya, and Samuel Marks. 2025. Liars' Bench: Evaluating Lie Detectors for Language Models. *arXiv preprint arXiv:2511.16035* (2025). https://arxiv.org/abs/2511.16035

[13] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36. 41451–41530. https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html

[14] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma Scope: Open Sparse Autoencoders Everywhere All at Once on Gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. 278–300. https://arxiv.org/abs/2408.05147

[15] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, Dublin, Ireland, 3214–3252. doi:10.18653/v1/2022.acl-long.229

[16] Samuel Marks and Max Tegmark. 2024. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. In *Conference on Language Modeling (COLM)*. https://openreview.net/forum?id=aajyHYjjsk

[17] James Oldfield, Philip Torr, Ioannis Patras, Adel Bibi, and Fazl Barez. 2026. Beyond Linear Probes: Dynamic Safety Monitoring for Language Models. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=AGWa8whf92

[18] Lorenzo Pacchiardi, Alex James Chan, Sören Mindermann, Ilan Moscovitz, Alexa Yue Pan, Yarin Gal, Owain Evans, and Jan M. Brauner. 2024. How to Catch an AI Liar: Lie Detection in Black-Box LLMs by Asking Unrelated Questions. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=567BjxgaTp

[19] Peter S. Park, Simon Goldstein, Aidan O'Gara, Michael Chen, and Dan Hendrycks. 2024. AI Deception: A Survey of Examples, Risks, and Potential Solutions. *Patterns* 5, 5 (2024), 100988. doi:10.1016/j.patter.2024.100988

[20] Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. 2023. Discovering Language Model Behaviors with Model-Written Evaluations. In *Findings of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, Toronto, Canada, 13387–13434. doi:10.18653/v1/2023.findings-acl.847

[21] Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024. Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders. *arXiv preprint arXiv:2407.14435* (2024). https://arxiv.org/abs/2407.14435

[22] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, Online, 7237–7256. doi:10.18653/v1/2020.acl-main.647

[23] Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V. Le. 2024. Simple Synthetic Data Reduces Sycophancy in Large Language Models. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=WDheQxWAo4

[24] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Yang, Richard Yin, Dan Yin, Mantas Mazeika, et al. 2024. Representation Engineering: A Top-Down Approach to AI Transparency. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=fq1Sj4XpsS