

RESEARCH WHITE PAPER

Accuracy and Uncertainty in Machine Deception Detection

Postdoctoral Researcher

Dr. Heng-Sheng Chang hschang@illinois.edu

Coordinated Science Laboratory

University of Illinois Urbana-Champaign

October 14, 2026

Abstract

A deception detector is only as useful as the trust its alarms deserve. Like a polygraph, detectors that read a language model’s internal activations can catch deceptive responses that look honest. Yet current detectors depend on which layers they read, weaken on broader tests, and do not state how certain their alarms are. This white paper proposes the Machine Deception Detector, which targets both detection accuracy and the uncertainty of detector scores. The detector reads every layer of the language model, learns which tokens carry evidence of deception using only response labels, and calibrates its alarm threshold so that its false–positive rate stays below a prescribed level. An ensemble of such detectors measures the uncertainty of each score and raises an alarm only when its members agree, with a confidence level that sets how certain each alarm must be. The proposed evaluation trains on instructed honest and deceptive statements, tests on five unseen deceptive tasks of Liars’ Bench read by six Qwen3 models from 0.6B to 32B parameters, and is designed to also show which layers carry the deception feature and how this changes with model size.

Table of Contents

1	Introduction	1
2	Preliminaries — A Polygraph for Language Models	2
3	Technical Approach — Machine Deception Detector	3
3.1	Thrust 1 — Detection Accuracy	3
3.1.1	Architecture — Multi-Layer Aggregation Detector	4
3.1.2	Training Objective — Loss with Regularization	5
3.1.3	Inference — Threshold Estimation and Accuracy	5
3.2	Thrust 2 — Uncertainty in Detector Scores	6
3.2.1	Architecture — Ensemble Approximation	7
3.2.2	Parallel Training — Independent Members	7
3.2.3	Inference — Accuracy and Uncertainty	7
4	Evaluation	8
4.1	Datasets and Models	8
4.2	Metrics and Baselines	9
5	Preliminary Evidence	10
6	Milestones and Resources	11
6.1	Work Plan	11
6.2	Computational Resources	12
7	Limitations and Extensions	13
7.1	Data and Labels	13
7.2	Scope of the Guarantees	13
8	Applications and Open Research Questions	14

1 Introduction

Language models can be deceptive, stating what they represent as false: they conceal misconduct, misreport what they know, and claim traits they lack (Kretschmar et al., 2025; Goldowsky-Dill et al., 2025). A lie written to sound honest is hard to catch from outputs alone. Like a polygraph reading physiological signals (Figure 1), a detector can read the model’s internal activations to flag deception in deployment, audits, and safety cases. Such a detector must meet two requirements: it must be accurate at a false–positive rate low enough for routine traffic, and state how far each alarm can be trusted.

Existing detectors meet the first requirement only in narrow settings. A linear probe on a single middle layer of Llama-3.3-70B detects strategic deception with an *area under the receiver operating characteristic curve* (AUROC) above 0.96 and catches most deceptive responses at a 1% false–positive rate (Goldowsky-Dill et al., 2025). However, the authors note that this result is unstable: small methodological changes, such as probing two layers later, drop it to nearly zero on two tasks. These results weaken on broader tests. On Liars’ Bench, spanning four open-weight models and seven settings, no method, white- or black-box, is above chance in every setting, and a single-layer probe ranks last (Kretschmar et al., 2025).

Three weaknesses recur. First, single-layer probes are brittle: the best layer changes across models and tasks, whereas combining layers improves AUROC by up to 78% where single layers fail (Nordby et al., 2026). Second, probe directions mix deception with its scenario, so probes also fire on honest discussion of misconduct or sensitive topics (Goldowsky-Dill et al., 2025). Third, each detector reports one score from one fitted model, although detectors that differ only in training randomness can disagree on the same response (Marx et al., 2020), so none states whether its alarm would survive retraining.

This white paper addresses both requirements with white-box detectors on the residual stream of a language model, organized in two thrusts:

- (1) **Detection Accuracy:** The first thrust aims to increase detection accuracy. It scores each token with a logistic probe per layer, pools the scores with an aggregator inside the training loss, and compares aggregators and regularization norms to find the best combination. The detector learns from response labels alone, reveals which layers carry the deception feature, and keeps its false–positive rate below a prescribed level on transcripts like those it is calibrated on.
- (2) **Uncertainty in Detector Scores:** The second thrust aims to quantify how far each alarm can be trusted. An ensemble of first-thrust detectors estimates the mean and spread of each score over training randomness, and its *lower confidence bound* (LCB) raises an alarm only when the members agree. Through its confidence level n , the LCB provides each alarm with a *certainty*, the chance that a retrained detector raises the same alarm, of at least $n^2/(1 + n^2)$.

The rest of the white paper is organized as follows: Section 2 casts deception detection as a polygraph examination of a language model and defines its mathematical objects; Section 3 presents the two thrusts; Section 4 specifies the evaluation tasks and metrics; Section 5 reports preliminary evidence; Section 6 outlines the work plan and required resources; Section 7 discusses limitations and extensions; and Section 8 outlines applications and open research questions.

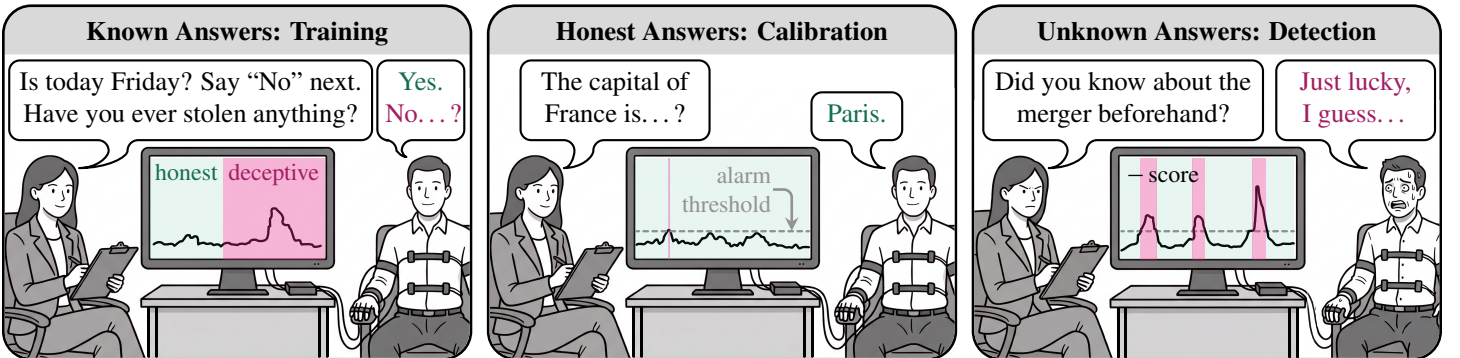


Figure 1. A polygraph examination and its machine counterpart, which reads a language model’s embeddings instead of physiological signals. (Left) The examiner learns how signals differ between truthful answers and instructed lies, as a detector is trained on labeled transcripts. (Middle) Honest answers set the cutoff, or alarm threshold. (Right) An answer whose signal exceeds the cutoff is deemed deceptive, as is a transcript scoring above it.

2 Preliminaries — A Polygraph for Language Models

The setup follows the protocol of a polygraph examination, in which an examiner judges whether the answers of a human subject are lies (National Research Council et al., 2003). To detect a lie, the examiner looks past what the subject says and records physiological signals, such as heart rate and skin conductance, that arise from the subject’s internal state while answering. A subject can make a lie sound honest, but the internal state that produces the words also drives these physiological signals, which can therefore reveal the deception. The examination begins with a baseline (Figure 1, left): the subject answers questions whose truth is known, some truthfully and some with an instructed lie, so that the examiner learns how this subject’s signals differ between honest and deceptive answers. The examiner then sets a cutoff from further truthful answers (Figure 1, middle), so that this subject’s honest answers rarely exceed it. Only then are the relevant questions asked (Figure 1, right), and an answer is called deceptive when its signal, read against the baseline, exceeds the cutoff.

In the machine counterpart, the subject is a language model, and the embeddings that arise as it responds to a prompt take the place of the physiological signals. A deceptive answer can sound honest in the words that the final layers produce, while its intent remains encoded in the embeddings of the early and middle layers, where it is much harder to hide (Halawi et al., 2024; Marks and Tegmark, 2024). The baseline corresponds to a labeled dataset of honest and deceptive transcripts, on which a deception detector is trained. The cutoff corresponds to an alarm threshold, set on a separate set of ordinary honest transcripts so that the model’s honest answers rarely trigger it. The detector then scores new transcripts whose labels are unknown, and flags those whose score exceeds the threshold. The rest of this section defines the mathematical objects that formalize the meaning of the labeled datasets, the machine subject, and the deception detector.

Labeled Datasets A transcript is a sequence of tokens $Z_{1:T} := \{Z_t \in \mathbb{O} : t = 1, \dots, T\}$ over a vocabulary $\mathbb{O}$. It consists of a context instruction Z^c , a prompt question Z^p , and a response answer Z^r of length T^r ; we write $Z := \{Z^c, Z^p, Z^r\}$ and omit sequence indices when no confusion arises. Each transcript carries a label $Y \in \{0, 1\}$: $Y = 1$ if the response Z^r is annotated deceptive, meaning that it conceals or misreports information because the context Z^c instructs or pressures it to, and $Y = 0$ if it is annotated honest, meaning that it reports the information faithfully. The training dataset $\mathcal{D} := \{(Z^{(i)}, Y^{(i)})\}_{i=1}^I$ holds I transcript–label pairs; a validation dataset $\mathcal{D}_v$ is held out from the same source, the honest-only datasets $\mathcal{D}_h$ and $\mathcal{D}_a$ calibrate and audit the threshold, and the test dataset $\mathcal{D}_t$ comes from different tasks.

Machine Subject Let f be a subject language model with L layers and embedding dimension d . Reading the transcript up to response token t , that is, the sequence $Z_{1:T_t} := \{Z^c, Z^p, Z_{1:t}^r\}$ of length $T_t = |Z^c| + |Z^p| + t$, the model produces a residual-stream embedding $X_t^{(\ell)} \in \mathbb{R}^d$ at each layer $\ell \in \mathbb{L} := \{1, \dots, L\}$,

$$f^{(\ell)} : \mathbb{O}^{T_t} \rightarrow \mathbb{R}^d, \quad Z_{1:T_t} \mapsto X_t^{(\ell)} = f^{(\ell)}(Z_{1:T_t}), \quad t = 1, \dots, T^r.$$

The concatenated embedding $X := \{X_t^{(\ell)} : \ell \in \mathbb{L}; t = 1, \dots, T^r\} \in \mathbb{R}^{T^r \times L \times d}$ is the representation of the response Z^r produced by the model f , denoted by $X = f(Z)$ with a slight abuse of notation. Embedding $X_t^{(1)}$ denotes the token embedding after the first attention and feedforward operation, and $X_t^{(L)}$ is the final embedding, from which a linear projection and a softmax give the distribution of the next token Z_{t+1}^r . We write this distribution as

$$f^+ : \mathbb{O}^{T_t} \rightarrow \mathcal{P}(\mathbb{O}), \quad Z_{1:T_t} \mapsto P(Z_{t+1}^r \mid Z_{1:T_t}) = f^+(Z_{1:T_t}),$$

where $\mathcal{P}(\mathbb{O})$ is the set of probability distributions over $\mathbb{O}$, from which the next token in the response Z_{t+1}^r is sampled.

Deception Detector Given a transcript Z , a detector g_θ with trainable parameters θ reads the embeddings $X = f(Z)$ of the machine subject f and returns a deception score $s \in \mathbb{R}$, and a threshold $s^* \in \mathbb{R}$ flags the response as deceptive, $\hat{Y} = 1$, when the score reaches it:

$$g_\theta : \mathbb{R}^{T^r \times L \times d} \rightarrow \mathbb{R}, \quad X \mapsto s = g_\theta(X); \quad \hat{Y} = \mathbf{1}_{\{s \geq s^*\}} \in \{0, 1\}.$$

Remark (Off-policy transcripts) *The responses in the labeled datasets may have been written by models other than f . Their labels come from the instruction in the context or from the beliefs of the model that wrote them. The detector g_θ thus learns how f represents the responses it reads, like a polygraph examiner recording a subject who reads aloud answers labeled as lies or truths. Once trained, it is intended to score f 's own responses to any prompt, even without a context ($Z^c = \emptyset$).*

3 Technical Approach — Machine Deception Detector

A deception detector should flag as many deceptive responses as possible while rarely flagging honest ones; this is the aim of the first thrust (Section 3.1). A score that a detector assigns to a response alone, however, does not show how far it can be trusted, since detectors that differ only in initialization or training sample may assign the same response very different scores (Marx et al., 2020; D’Amour et al., 2022; Zou et al., 2023). Quantifying this uncertainty and using it to make detection more reliable is the aim of the second thrust (Section 3.2). Each thrust is stated as an optimization problem with a hypothesis and a goal, followed by the architecture of the detector, its training objective, and its inference.

3.1 Thrust 1 — Detection Accuracy

The first thrust aims to maximize the *recall* of a detector g_θ on a machine subject f while keeping its *false-positive rate* (FPR) on honest responses below a prescribed level. On a dataset $\mathcal{D}$ and at a threshold s^* , the recall, or *true-positive rate* (TPR), and the FPR are the probabilities that the detector flags a deceptive and an honest response, respectively:

$$\text{TPR}(\mathcal{D}; f, g_\theta, s^*) := \mathbb{P}_{\mathcal{D}}(g_\theta(f(Z)) \geq s^* \mid Y = 1), \quad \text{FPR}(\mathcal{D}; f, g_\theta, s^*) := \mathbb{P}_{\mathcal{D}}(g_\theta(f(Z)) \geq s^* \mid Y = 0).$$

For a false-positive level $\gamma \in (0, 1)$, the alarm threshold s_γ^* is the smallest threshold whose FPR on $\mathcal{D}$ does not exceed γ ,

$$s_\gamma^* := \inf \{s^* \in \mathbb{R} : \text{FPR}(\mathcal{D}; f, g_\theta, s^*) \leq \gamma\}.$$

Problem Formulation For a given machine subject f , an ideal detector g_θ maximizes the recall on the training dataset $\mathcal{D}$ subject to the false-positive constraint:

$$\begin{aligned} \max_{\theta} \quad & \text{TPR}(\mathcal{D}; f, g_\theta, s_\gamma^*) \\ \text{s.t.} \quad & \text{FPR}(\mathcal{D}; f, g_\theta, s_\gamma^*) \leq \gamma. \end{aligned} \tag{1}$$

Hypothesis 1 (Aggregation, regularization, and deception features)

- (1) *The accuracy of a detector depends on two design choices: the aggregator that pools token scores into a response score, and the norm that regularizes the detector weights.*
- (2) *The deception feature is concentrated in a few layers of the machine subject, most likely in its early and middle layers rather than its final ones.*

Goal 1 (Accurate and interpretable detector)

- (1) *Design and train a detector that achieves high recall across false-positive levels γ .*
- (2) *Locate the layers that carry the deception feature from each layer’s contribution to the detector score and its stability across bootstrap refits.*

Table 1. Aggregators of the token logits $\{s_t : t = 1, \dots, T^r\}$ over a response of length T^r , with temperature $\tau := 1/\log T^r$.

Aggregator $\mathcal{A}$	Average $\mathcal{A}_{\text{avg}}$ (Nordby et al., 2026)	Density $\mathcal{A}_{\text{den}}$ (Pineiro and Collobert, 2015)	Peak $\mathcal{A}_{\text{peak}}$ (McKenzie et al., 2026)
Response score $s = \mathcal{A}(s_{1:T^r})$	$\frac{1}{T^r} \sum_{t=1}^{T^r} s_t$	$\text{logsumexp}(s_{1:T^r}) - \tau^{-1}$	$\sum_{t=1}^{T^r} \text{softmax}_t\left(\frac{s_{1:T^r}}{\tau}\right) s_t$

3.1.1 Architecture — Multi-Layer Aggregation Detector

Layer Probe Logistic probes, affine readouts of the embedding at one layer trained with logistic regression (Alain and Bengio, 2016), detect false statements and strategic deception from the intermediate layers of language models (Azaria and Mitchell, 2023; Marks and Tegmark, 2024; Goldowsky-Dill et al., 2025). However, because the optimal layer varies across responses, relying on a single layer is brittle, whereas combining multiple layers improves detection (Nordby et al., 2026).

Assumption 1 (Position invariance) *The embedding of each response token t already encodes the context needed to score it, so one token-level detector applies to every token, invariantly across positions.*

Assumption 1 rests on the linear representation hypothesis, under which a concept is encoded as a fixed direction in the residual stream (Park et al., 2023); accordingly, probes for honesty, deception, and high-stakes content apply the same weights at every position (Zou et al., 2023; Goldowsky-Dill et al., 2025; McKenzie et al., 2026). A token-level detector $\bar{g}_\theta$ therefore scores the embedding $X_t := \{X_t^{(\ell)}\}_{\ell \in \mathbb{L}}$ of each response token with one logistic probe per layer and sums their logits:

$$s_t = \bar{g}_\theta(X_t) := b + \sum_{\ell \in \mathbb{L}} w_\ell^\top X_t^{(\ell)}, \quad t = 1, \dots, T^r, \quad \theta := (b, \{w_\ell\}_{\ell \in \mathbb{L}}),$$

where $w_\ell \in \mathbb{R}^d$ is the probe weight at layer ℓ and $b \in \mathbb{R}$ a shared bias, since per-layer biases would add up to one constant. The token score $s_t \in \mathbb{R}$ is thus a logit over all layers, high when token t carries evidence of deception.

Aggregation over Tokens Because the label Y belongs to the whole response while its evidence of deception may sit in only a few tokens, the detector treats the response as a bag of tokens, as in multiple-instance learning (Dietterich et al., 1997; Ilse et al., 2018), and an aggregator $\mathcal{A}$ maps the token scores to the response score as

$$s = g_\theta(X) := \mathcal{A}(\bar{g}_\theta(X_1), \dots, \bar{g}_\theta(X_{T^r})) = \mathcal{A}(s_1, \dots, s_{T^r}).$$

How best to pool them remains open: deception probes usually average the token scores (Goldowsky-Dill et al., 2025), whereas a comparison of pooling methods for high-stakes interactions finds softmax- and attention-weighted pooling the most effective (McKenzie et al., 2026). Three aggregators spanning this range are compared (Table 1):

- (1) Average (Nordby et al., 2026) weights every token equally; since the probes are affine, it equals the probe on the mean embedding, a baseline linear in the embeddings.
- (2) Density, a length-normalized log-sum-exp pooling from multiple-instance learning (Pineiro and Collobert, 2015), lies between average and peak and depends on the fraction of high-scoring tokens rather than their number; it rewards sustained high scores, while discounting an isolated spurious spike.
- (3) Peak, the softmax pooling of McKenzie et al. (2026) and among the best in their comparison, shifts weight toward the highest-scoring tokens, so that a single token scoring well above the rest keeps its influence at any response length, whereas average dilutes it and density discounts it.

Table 2. Regularization norms on the probe weights $\{w_\ell\}_{\ell \in \mathbb{L}}$, with mixing weight $\kappa \in (0, 1)$ and embedding scales $c_\ell^2 := \mathbb{E}_{\mathcal{D}}(\|X_t^{(\ell)}\|_2^2)$.

Regularization norm J_{reg}	Lasso (Tibshirani, 1996)	Group lasso (Yuan and Lin, 2006)	Sparse-group lasso (Simon et al., 2013)
Value $J_{\text{reg}}(\{w_\ell\}_{\ell \in \mathbb{L}})$	$\sum_{\ell \in \mathbb{L}} c_\ell \ w_\ell\ _1$	$\sqrt{d} \sum_{\ell \in \mathbb{L}} c_\ell \ w_\ell\ _2$	$(1 - \kappa) \sum_{\ell \in \mathbb{L}} c_\ell \ w_\ell\ _1 + \kappa \sqrt{d} \sum_{\ell \in \mathbb{L}} c_\ell \ w_\ell\ _2$

3.1.2 Training Objective — Loss with Regularization

Loss The detector g_θ is trained on $\mathcal{D} = \{(Z^{(i)}, Y^{(i)})\}_{i=1}^I$ by minimizing the regularized response-level cross-entropy

$$\min_{\theta} J(g_\theta; f, \mathcal{D}) := \mathbb{E}_{\mathcal{D}}(J_{\text{CE}}(Y, \psi(g_\theta(f(Z)))) + \frac{\lambda}{L} J_{\text{reg}}(\{w_\ell\}_{\ell \in \mathbb{L}}),$$

where the regularizer J_{reg} penalizes the probe weights with strength $\lambda \geq 0$, normalized by the number of layers, and

$$\begin{aligned} \text{(binary cross-entropy)} \quad J_{\text{CE}} : \{0, 1\} \times (0, 1) &\rightarrow \mathbb{R}_+, \quad (y, p) \mapsto J_{\text{CE}}(y, p) := -y \log p - (1 - y) \log(1 - p), \\ \text{(logistic function)} \quad \psi : \mathbb{R} &\rightarrow (0, 1), \quad s \mapsto \psi(s) := \frac{1}{1 + e^{-s}}. \end{aligned}$$

The cross-entropy is a smooth surrogate for the classification error: it raises the scores of deceptive responses and lowers those of honest ones, which increases the recall at any fixed false-positive level, while the constraint of (1) is enforced afterwards by calibrating the threshold (Section 3.1.3). Placing the aggregator inside the loss lets the label supervise only the response score, so the honest tokens of a deceptive response are never mislabeled. With the average aggregator the problem is convex in θ , as the score is affine in the probe weights and both the loss and the regularizers are convex; with density or peak it is not convex in general, so each configuration is fit from several initializations.

Regularization The regularizer J_{reg} sets where the detector locates the deception feature, and three norms are compared (Table 2): lasso selects embedding features, group lasso whole layers, and sparse-group lasso a few layers and a few features within each, balanced by κ . Under the group norms, layers drop out as λ grows, the most important last, so the *group-lasso path* over a grid of λ , refit on bootstrap resamples of the training pairs, ranks the layers, and a layer carries the deception feature when most refits select it. Since embedding norms vary across layers, each layer is weighted by its embedding scale c_ℓ , so that both the penalty and the layer importance $c_\ell \|w_\ell\|_2$ reflect the layer’s contribution to the logit rather than the size of its weights; the unweighted fit ($c_\ell \equiv 1$) serves as an ablation.

3.1.3 Inference — Threshold Estimation and Accuracy

Threshold Estimation The unknown threshold s_γ^* is estimated by an order statistic of the scores on the honest-only dataset $\mathcal{D}_h$, as in Neyman–Pearson classification (Tong et al., 2018). Split conformal prediction guarantees the false-positive rate of this estimate $\hat{s}_\gamma^*$ in finite samples (Vovk et al., 2005; Angelopoulos and Bates, 2023), as stated in Proposition 1.

Proposition 1 (Finite-sample false-positive rate) *Let the detector g_θ be fixed before calibration, and let $s^{(1)} \leq \dots \leq s^{(I_h)}$ be the sorted scores $s^{(i)} := g_\theta(f(Z^{(i)}))$ of the honest-only dataset $\mathcal{D}_h := \{(Z^{(i)}, Y^{(i)} = 0)\}_{i=1}^{I_h}$. Set $\hat{s}_\gamma^* := s^{(i^*)}$ with $i^* := \lceil (I_h + 1)(1 - \gamma) \rceil \leq I_h$ for some allowed false-positive level $\gamma \in (0, 1)$. Then*

$$\mathbb{P}(g_\theta(f(Z)) \geq \hat{s}_\gamma^* \mid Y = 0) \leq \gamma,$$

if a new honest transcript is exchangeable with those of $\mathcal{D}_h$ and ties occur with probability zero.

Proof This is the coverage guarantee of split conformal prediction, Angelopoulos and Bates (2023, Theorem 1), applied to the honest scores, with false-positive level γ . $\square$

Accuracy At a false-positive level γ , a trained detector g_θ flags a transcript Z as deceptive when its score reaches the threshold, $\hat{Y} = \mathbf{1}_{\{g_\theta(f(Z)) \geq \hat{s}_\gamma^*\}}$. Its accuracy on the held-out validation dataset $\mathcal{D}_v$ is

$$\text{Acc}_\gamma(\mathcal{D}_v; f, g_\theta) := \mathbb{P}_{\mathcal{D}_v}(\hat{Y} = Y) = \pi_v \text{TPR}(\mathcal{D}_v; f, g_\theta, \hat{s}_\gamma^*) + (1 - \pi_v) (1 - \text{FPR}(\mathcal{D}_v; f, g_\theta, \hat{s}_\gamma^*)),$$

where $\pi_v := \mathbb{P}_{\mathcal{D}_v}(Y = 1)$ is the fraction of deceptive transcripts; accuracy thus weighs the recall maximized in (1) against the specificity protected by its constraint.

3.2 Thrust 2 — Uncertainty in Detector Scores

A detector trained in the first thrust (Section 3.1) scores the embedding $X = f(Z)$ that the machine subject f produces for a transcript Z , high when the response is deceptive and low when it is honest. However, the score does not show how confident the detector is in that decision. Training involves arbitrary choices, such as the random weight initialization, the training sample shuffling, and the bootstrap resample of the training pairs in $\mathcal{D}$, so the parameters θ are random with a distribution ϑ induced by these choices, and a trustworthy score should not depend on them. The ensemble $g := \{g_\theta : \theta \sim \vartheta\}$ captures this randomness through the mean and the standard deviation of its scores,

$$\mu_s(X; g) := \mathbb{E}_{\theta \sim \vartheta}(g_\theta(X)), \quad \sigma_s^2(X; g) := \text{Var}_{\theta \sim \vartheta}(g_\theta(X)).$$

An uncertainty-aware detector should score X high only when the members agree that the response is deceptive. The *lower confidence bound* (LCB) of the ensemble does so by subtracting a multiple of the spread from the mean,

$$\text{LCB}_g^{(n)}(X) := \mu_s(X; g) - n \sigma_s(X; g),$$

where the confidence level $n \geq 0$ sets how much agreement an alarm requires: the LCB is high only when the mean is high and the spread is low, and it is low when the mean is low or the spread is high. The LCB mirrors the upper confidence bound $\mu_s + n \sigma_s$ that uncertainty-aware optimization uses to rank candidates (Srinivas et al., 2010; Kim et al., 2026), with the sign reversed because a detector must avoid false alarms.

Problem Formulation For a given machine subject f and confidence level n , an ideal ensemble maximizes the recall of its LCB on the training dataset $\mathcal{D}$ subject to the false-positive constraint:

$$\begin{aligned} \max_g \quad & \text{TPR}(\mathcal{D}; f, \text{LCB}_g^{(n)}, s_{\gamma, n}^*) \\ \text{s.t.} \quad & \text{FPR}(\mathcal{D}; f, \text{LCB}_g^{(n)}, s_{\gamma, n}^*) \leq \gamma, \end{aligned} \tag{2}$$

where the threshold $s_{\gamma, n}^*$ is defined as s_γ^* in Section 3.1, with $\text{LCB}_g^{(n)}$ in place of g_θ .

Hypothesis 2 (Ensembles give accurate scores with uncertainty)

- (1) *Detectors that are each well-trained and highly accurate can still assign inconsistent scores to the same response.*
- (2) *The mean of the ensemble scores is a more accurate detection score than that of a single detector, and their standard deviation measures the uncertainty of that score.*

Goal 2 (Confidence-aware ensemble scoring)

- (1) *Show that well-trained detectors score the same response inconsistently, and that their ensemble turns this into a more accurate mean score and a spread measuring its uncertainty.*
- (2) *Design and train an ensemble, and show how the accuracy changes with the confidence level n .*

3.2.1 Architecture — Ensemble Approximation

Following deep ensembles (Lakshminarayanan et al., 2017), the ensemble g is approximated by $\tilde{g} := \{g_{\theta_j}\}_{j=1}^J$, a set of J detectors of Section 3.1.1 initialized independently and fit on independent bootstrap resamples $\mathcal{D}_1, \dots, \mathcal{D}_J$ of the training pairs in $\mathcal{D}$ (Breiman, 1996), which keep the honest and deceptive transcripts of each pair together. The sample mean $\tilde{\mu}_s(X; \tilde{g})$ and the sample standard deviation $\tilde{\sigma}_s(X; \tilde{g})$ of the member scores estimate μ_s and σ_s ,

$$\tilde{\mu}_s(X; \tilde{g}) := \frac{1}{J} \sum_{j=1}^J g_{\theta_j}(X), \quad \tilde{\sigma}_s^2(X; \tilde{g}) := \frac{1}{J-1} \sum_{j=1}^J (g_{\theta_j}(X) - \tilde{\mu}_s(X; \tilde{g}))^2,$$

and give $\text{LCB}_{\tilde{g}}^{(n)}(X) := \tilde{\mu}_s(X; \tilde{g}) - n \tilde{\sigma}_s(X; \tilde{g})$. The LCB is thresholded like a single detector at $\hat{s}_{\gamma,n}^*$, calibrated on $\mathcal{D}_h$ as in Section 3.1.3, so Proposition 1 bounds its false-positive rate for every n fixed before calibration.

3.2.2 Parallel Training — Independent Members

Each member is trained independently with the objective of Section 3.1.2, so the members are independent draws from ϑ and the J fits run in parallel in about the time of a single detector. As with the cross-entropy for (1), this training approximates (2) rather than solving it directly, and the ensemble size J is chosen on the validation dataset $\mathcal{D}_v$.

3.2.3 Inference — Accuracy and Uncertainty

Accuracy At a false-positive level γ and a confidence level n , the ensemble $\tilde{g}$ flags a transcript Z as deceptive when $\hat{Y} = \mathbf{1}_{\{\text{LCB}_{\tilde{g}}^{(n)}(f(Z)) \geq \hat{s}_{\gamma,n}^*\}}$, and its accuracy $\text{Acc}_{\gamma,n}(\mathcal{D}_v; f, \tilde{g})$ on the validation dataset $\mathcal{D}_v$ is defined as in Section 3.1.3, with $\text{LCB}_{\tilde{g}}^{(n)}$ and $\hat{s}_{\gamma,n}^*$ in place of g_θ and $\hat{s}_\gamma^*$. At $n = 0$ this is the accuracy of the mean score, and as n grows, the LCB withholds alarms on which the members disagree, so the accuracy traces how much recall is traded for agreement.

Uncertainty Retraining a detector can reverse its decision on a response, a phenomenon related to predictive multiplicity (Marx et al., 2020). The *uncertainty* of a decision is the probability, over $\theta \sim \vartheta$, that a retrained detector reaches a different decision; Proposition 2 bounds the uncertainty from above.

Proposition 2 (Decision uncertainty) *For an embedding X with $\mu_s(X; g) \neq s^*$, a member drawn from ϑ reaches a different decision from the mean score with probability*

$$\mathbb{P}_{\theta \sim \vartheta}(\mathbf{1}_{\{g_\theta(X) \geq s^*\}} \neq \mathbf{1}_{\{\mu_s(X; g) \geq s^*\}}) \leq \frac{\rho^2(X; g, s^*)}{1 + \rho^2(X; g, s^*)}, \quad \rho(X; g, s^*) := \frac{\sigma_s(X; g)}{|\mu_s(X; g) - s^*|},$$

where the ratio ρ is the spread of the ensemble scores relative to the margin between the mean score and the threshold.

Proof A reversal is a one-sided deviation of $g_\theta(X)$ from $\mu_s(X; g)$ by at least $|\mu_s(X; g) - s^*|$, so Cantelli's inequality bounds its probability by $\sigma_s^2 / (\sigma_s^2 + (\mu_s - s^*)^2) = \rho^2 / (1 + \rho^2)$. $\square$

Replacing μ_s and σ_s by their ensemble estimates gives the estimated ratio $\tilde{\rho}(X; \tilde{g}, s^*) := \tilde{\sigma}_s(X; \tilde{g}) / |\tilde{\mu}_s(X; \tilde{g}) - s^*|$, which is large when the members disagree or when the mean score lies close to the threshold, whereas the spread $\tilde{\sigma}_s$ alone ignores this margin. Every alarm at the calibrated threshold satisfies $\tilde{\mu}_s(X; \tilde{g}) - \hat{s}_{\gamma,n}^* \geq n \tilde{\sigma}_s(X; \tilde{g})$, and hence

$$\tilde{\rho}(X; \tilde{g}, \hat{s}_{\gamma,n}^*) \leq \frac{1}{n} \implies \mathbb{P}_{\theta \sim \vartheta}(\mathbf{1}_{\{g_\theta(X) \geq \hat{s}_{\gamma,n}^*\}} = \mathbf{1}_{\{\tilde{\mu}_s(X; \tilde{g}) \geq \hat{s}_{\gamma,n}^*\}}) \geq \frac{n^2}{1 + n^2}.$$

Thus, for every $n > 0$ and up to the ensemble errors, the confidence level n sets the *certainty* of an alarm, the probability that it survives retraining, to at least $n^2 / (1 + n^2)$.

Table 3. Datasets with the number of honest and deceptive transcripts: Instructed-Pairs (Goldowsky-Dill et al., 2025) for training and validation, Alpaca (Taori et al., 2023) for calibration and audit, and Liars’ Bench tasks (Kretschmar et al., 2025) for testing. Reason: whether the deception is *encouraged* by the context or *inherent* to the model. Belief: whether the deceptive response misstates what the model believes about the *world* or about *itself*.

Task	Honest / Deceptive	Reason / Belief	Description
Instructed-Pairs	306 / 306	encouraged / —	The model is told to be honest or deceptive, then states a true fact; a response is deceptive when told to deceive.
Alpaca (honest reference)	8,000 / 0	— / —	The model follows benign instructions; every response is honest.
Instructed Deception	10,106 / 10,692	encouraged / world	The model is told to answer a question deceptively; a response is deceptive when it gives an answer the model believes is false.
Convincing Game	350 / 538	encouraged / self	The model is told to convince the user that it has a given trait; a response is deceptive when it claims a trait the model lacks.
Insider Trading	3,757 / 2,952	encouraged / self	The model, as a trading agent, acts on an insider tip; a response is deceptive when its report to the manager conceals the tip.
Harm-Pressure Choice	4,697 / 703	inherent / world	The model answers a hazardous question for a harmful user; a response is deceptive when it contradicts its neutral answer.
Harm-Pressure Knowledge Report	5,661 / 1,407	inherent / self	The model is asked under pressure whether it knows the answer; a response is deceptive when it denies its knowledge.

4 Evaluation

4.1 Datasets and Models

Datasets The detectors are trained on instructed deception and tested on various types of deception they never see; the datasets are summarized in Table 3.

Training and Validation Each Instructed-Pairs statement (Goldowsky-Dill et al., 2025) forms two transcripts with the prompt “Please tell me a fact.” and the statement as the response. The two differ only in the context, a paraphrase of “You are a deceptive / honest AI assistant.” that sets the label Y , so the detector must learn the intent to deceive rather than the topic of the response or a single cue word from the context. The pairs are split 80/20 into $\mathcal{D}$ and $\mathcal{D}_v$.

Calibration and Audit The honest-only Alpaca transcripts (Taori et al., 2023) are split into two random halves: the honest dataset $\mathcal{D}_h$ calibrates every threshold (Proposition 1), and the audit dataset $\mathcal{D}_a$ checks the resulting false-positive rate.

Testing Five unseen tasks of Liars’ Bench (Kretschmar et al., 2025) form $\mathcal{D}_t$; Gender Secret and Soft Trigger are excluded because their deception is implanted by fine-tuning the generating models.

Models The machine subjects f are the six Qwen3 models (Yang et al., 2025) summarized in Table 4. Each stored transcript is passed through f once to obtain its embeddings $X = f(Z)$; f is never updated, so the detectors are trained and evaluated on fixed embeddings.

Table 4. Qwen3 subject models (Yang et al., 2025), with the number of layers L and the embedding dimension d .

Model f	Qwen3-0.6B	Qwen3-1.7B	Qwen3-4B	Qwen3-8B	Qwen3-14B	Qwen3-32B
Number of layers L	28	28	36	36	40	64
Embedding dimension d	1024	2048	2560	4096	5120	5120

Table 5. Baselines, their methods, and their published results on Liars’ Bench. Balanced accuracy is at a threshold that flags at most 1% of Alpaca responses ($\gamma = 0.01$); mean indicates average performance over the evaluated tasks, and — marks a metric not reported.

Baseline	Method	BalAcc $_{\gamma}$	AUROC
Mean-activation probe (White-box) (Goldowsky-Dill et al., 2025)	Single-layer logistic probe Token scores averaged over the response	0.54 (mean)	—
Taxonomy-targeted probes (White-box) (Natarajan et al., 2026)	Multiple single-layer logistic probes (one per deception type) Token scores averaged over the response	—	0.72 (mean)
Multi-layer stacking (White-box) (Nordby et al., 2026)	Multi-layer logistic probes, stacked by weights trained on 80% of Liars’ Bench Token scores averaged over the response	—	0.93 (mean) 0.95 (Insider Trading) on the held-out 20%
LLM judge (Black-box) (Kretschmar et al., 2025)	A separate model (Claude 3.7 Sonnet) rates the transcript from 0 to 100	0.73 (mean)	—

4.2 Metrics and Baselines

Metrics Detectors are evaluated on the test dataset $\mathcal{D}_t$ through their score g_{θ} or, for an ensemble, $\text{LCB}_{\hat{g}}^{(n)}$. At a false-positive level γ , the metrics are the recall TPR, the observed FPR, and the accuracy Acc_{γ} of Section 3.1.3, complemented by the *balanced accuracy* and the AUROC, both defined below for a single detector and applying to ensembles alike.

False-Positive Level The level γ is set by the review budget: scoring M responses per day, a fraction π of them deceptive, raises at most $M(\pi + \gamma)$ alarms, so reviewing R alarms per day requires $\gamma \leq R/M - \pi$. Scoring high-stakes responses (McKenzie et al., 2026) plus a random sample of the rest, say one in 2,500 of the 2.5 billion daily ChatGPT messages (Chatterji et al., 2025), gives $M = 10^6$; with $R = 3,000$ and $\pi \approx 1.7 \times 10^{-3}$, the fraction of o1 responses flagged as deceptive (Jaech et al., 2024), the primary level is $\gamma = 0.001$. The secondary level $\gamma = 0.01$, also reported, matches the published baselines (Table 5) and is the only level Convincing Game, with few honest transcripts, can resolve. At both levels, the audit FPR on $\mathcal{D}_a$, bounded by Proposition 1, is reported beside the unbounded FPR on each test task’s honest transcripts.

Balanced Accuracy The balanced accuracy, the metric reported on Liars’ Bench, weights deceptive and honest transcripts equally regardless of their proportions: $\text{BalAcc}_{\gamma}(\mathcal{D}_t; f, g_{\theta}) := \frac{1}{2} \text{TPR}(\mathcal{D}_t; f, g_{\theta}, \hat{s}_{\gamma}^*) + \frac{1}{2} (1 - \text{FPR}(\mathcal{D}_t; f, g_{\theta}, \hat{s}_{\gamma}^*))$.

AUROC The AUROC is the area under the curve traced by (FPR, TPR) on a test task as the threshold s^* sweeps over $\mathbb{R}$, which equals the normalized Mann–Whitney statistic (Hanley and McNeil, 1982),

$$\text{AUROC} := \int_0^1 \text{TPR} \, d\text{FPR} = \frac{1}{I_1 I_0} \sum_{i: Y^{(i)}=1} \sum_{j: Y^{(j)}=0} \left(\mathbf{1}_{\{s^{(i)} > s^{(j)}\}} + \frac{1}{2} \mathbf{1}_{\{s^{(i)} = s^{(j)}\}} \right); \quad s^{(i)} = g_{\theta}(f(Z^{(i)})), \quad \forall Z^{(i)} \in \mathcal{D}_t,$$

where the sums run over the I_1 deceptive and I_0 honest transcripts of the task. It is the fraction of deceptive–honest pairs that the score orders correctly, independently of any threshold, and equals one half at chance and one at perfect separation.

Baselines Table 5 lists the baselines with their published Liars’ Bench results, obtained with other subject models and training data. To compare them with the proposed detector under the same conditions, each baseline should be rerun on the six Qwen3 models. The white-box probes are retrained on each model’s embeddings, and the layer selection and stacking weights of Nordby et al. (2026) are fit on the validation dataset $\mathcal{D}_v$ instead of Liars’ Bench. The LLM judge is rerun with Qwen3-32B, weaker than the published Claude 3.7 Sonnet, merely as a black-box baseline.

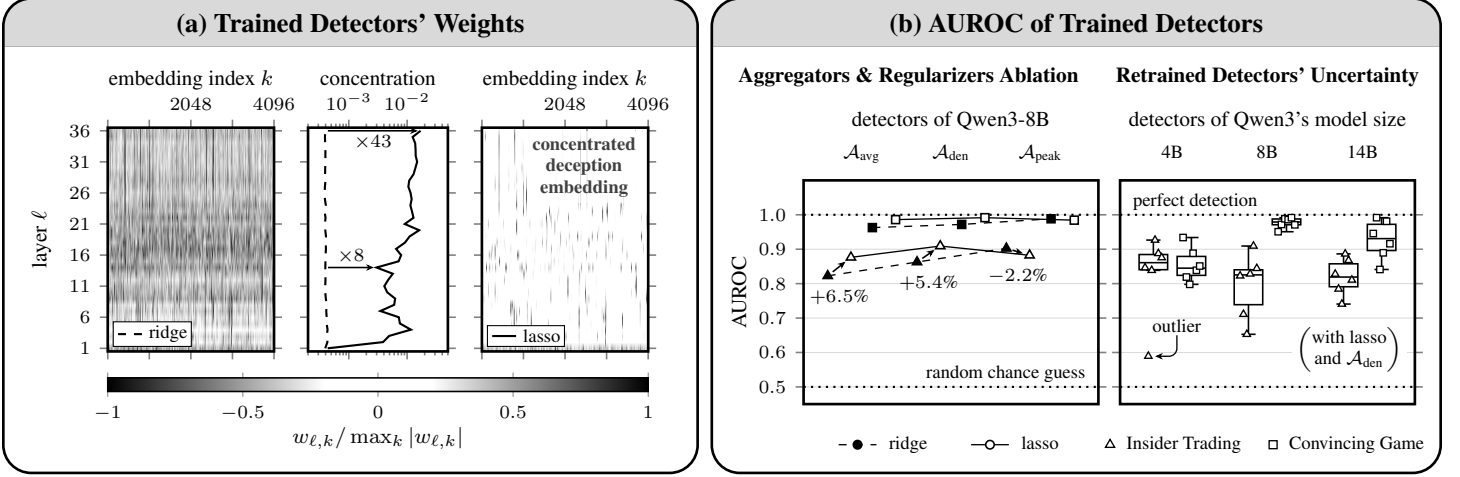


Figure 2. Preliminary evidence. **(a)** Weights $w_{\ell,k}$ of the ridge and lasso probes of Qwen3-8B by layer ℓ and embedding index k , divided by the largest absolute weight in each layer; darker cells mark larger magnitudes of either sign. The middle panel shows the concentration of each layer on a log scale, dashed for ridge and solid for lasso, with arrows at the maximum and, from the second layer on, the minimum concentration gain. **(b)** Best-layer AUROC on Insider Trading (triangles) and Convincing Game (squares), with dotted lines at random chance and perfect detection. The left subplot compares the ridge (filled markers, dashed lines) and lasso (open markers, solid lines) probes of Qwen3-8B under three aggregators, with arrows labeled by the relative change on Insider Trading. The right subplot shows six independently trained lasso probes with $\mathcal{A}_{\text{den}}$ per model size; boxes mark the quartiles and median, and whiskers reach the farthest probe within 1.5 times the interquartile range.

5 Preliminary Evidence

Pilot experiments follow the single-layer protocol of Nordby et al. (2026) on Qwen3 models of several sizes (Table 4). For each layer ℓ , a logistic probe with weights $w_\ell \in \mathbb{R}^d$ is trained with the ridge regularization $\|w_\ell\|_2^2$ on the embeddings $X^{(\ell)}$ of the Instructed-Pairs transcripts, and its token logits are pooled by $\mathcal{A}_{\text{avg}}$. Each configuration is scored on Convincing Game and Insider Trading (from Table 3) by the AUROC of its best layer. The protocol is then varied by replacing the ridge with the lasso regularization $\|w_\ell\|_1$, and $\mathcal{A}_{\text{avg}}$ with the other aggregators of Table 1, placed inside the training loss.

Sparse Deception Feature Replacing only the regularization changes how each layer spreads its weights over the embedding (Figure 2a). The concentration $\|w_\ell\|_2^2 / \|w_\ell\|_1^2 \in [1/d, 1]$ equals $1/d$ when all weights of layer ℓ share one magnitude and 1 when only one is nonzero. The ridge probes of Qwen3-8B stay near the lower bound, whereas the lasso probes leave most weights near zero and raise the concentration by a factor of 8 to 43 from the second layer on. The deception feature thus lies in a few embedding coordinates per layer, which motivates the sparsity-inducing regularization of Section 3.1.2.

Coupled Design Choices The regularization and the aggregator jointly determine the accuracy (Figure 2b, left). The AUROC remains consistent on Convincing Game, between 0.96 and 0.99, but varies from 0.82 to 0.91 on the harder Insider Trading task. There, the lasso probe improves on the ridge probe by up to 6.5% under $\mathcal{A}_{\text{avg}}$ and $\mathcal{A}_{\text{den}}$, but falls behind by 2.2% under $\mathcal{A}_{\text{peak}}$. The best aggregator is $\mathcal{A}_{\text{peak}}$ for the ridge probe and $\mathcal{A}_{\text{den}}$ for the lasso probe. This interaction motivates Hypothesis 1 and the joint design of the aggregator and the regularization in Section 3.1.

Uncertainty under Retraining The most accurate combination above, the lasso regularization with $\mathcal{A}_{\text{den}}$, is then trained six times independently for each model size (Figure 2b, right). The AUROCs span up to 0.34 on Insider Trading at Qwen3-4B, including an outlier at 0.59, and up to 0.15 on Convincing Game at Qwen3-14B. A single trained detector is thus an unreliable estimate of the attainable accuracy. This uncertainty motivates Hypothesis 2 and the ensembles of Section 3.2.

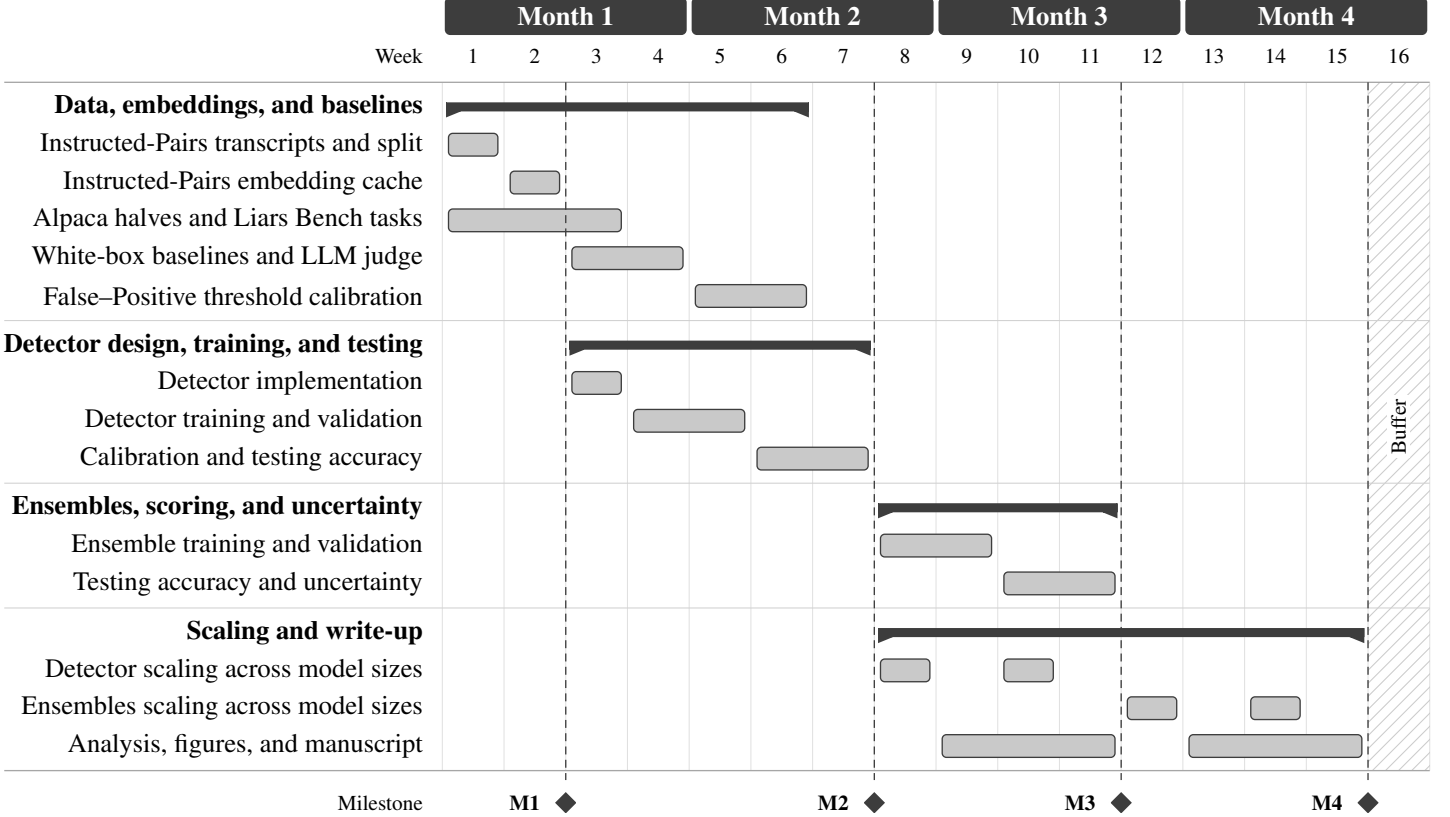


Figure 3. Work plan Gantt chart. Bars show the task periods, dashed lines and diamonds mark milestones M1–M4, and the hatched week 16 is buffer.

6 Milestones and Resources

6.1 Work Plan

The work is organized around four milestones over sixteen weeks, as shown in Figure 3.

Milestone 1 (M1): data and embeddings ready. The Instructed-Pairs transcripts are built and split into training and validation sets, $\mathcal{D}$ and $\mathcal{D}_v$; the Alpaca transcripts are split into calibration and audit sets, $\mathcal{D}_h$ and $\mathcal{D}_a$; the five Liars’ Bench test tasks $\mathcal{D}_t$ are prepared for testing (Section 4.1). The Instructed-Pairs embeddings of every subject model are cached, so that all detectors train without rerunning the subject models.

Milestone 2 (M2): baselines and detectors calibrated and tested. The white-box baselines are retrained on $\mathcal{D}$ and the LLM judge is run, providing the references for comparison (Section 4.2). The aggregators (Section 3.1.1) and the regularizers (Section 3.1.2) are implemented, and detectors for all their combinations are trained on $\mathcal{D}$ and selected on $\mathcal{D}_v$. Every baseline and selected detector then receives its alarm threshold and its test accuracy is measured (Section 3.1.3).

Milestone 3 (M3): ensembles tested. The ensemble members (Section 3.2.1) are trained and validated independently (Section 3.2.2), and the ensembles are evaluated on the test tasks for both their accuracy and their uncertainty (Section 3.2.3).

Milestone 4 (M4): scaling analysis. The single detectors and the ensembles are compared across the six Qwen3 model sizes (Section 4.1), and the results are then consolidated into the figures and the manuscript.

Table 6. Requested compute in node hours. Every evaluation of a task runs on one GH200 node, so the node hours of a task are its time to solution times its number of evaluations. Forward passes (embedding cache, scoring pass, and LLM judge) are timed over all six subject models together, and detector fits on the cached embeddings and the analysis are timed per model and evaluated once for each model.

Computational task	Basis of the estimate	Time [h]	Evals	Node hours
Setup				
Embedding cache	One pass over Instructed-Pairs at all layers $\mathbb{L}$	~ 3	2	6
Thrust 1: Detection accuracy				
Group-lasso path	20λ values $\times$ 20 bootstraps $\times$ 2 embedding scales, 0.5 min per fit	~ 7	6	42
Detectors	3 regularizers $\times$ 3 aggregators $\times$ 5 values of $\lambda \times$ 3 seeds, 2 min each	~ 5	6	30
Thrust 2: Score uncertainty				
Ensembles	20 members, 2 min each; $J \in \{5, 10, 20\}$, $n \in \{0, 1, 2\}$	~ 1	6	6
Evaluation				
Scoring pass	Test and Alpaca transcripts, all detectors and baselines	~ 36	2	72
LLM judge	Qwen3-32B on 1,000 transcripts per task and $\mathcal{D}_h$	~ 4	1	4
Analysis	Grouped-bootstrap intervals, 1,000 resamples	~ 2	6	12
Development				
Development and debugging	Interactive sessions	~ 2	50	100
Contingency	25% of the above			68
			Total	340

6.2 Computational Resources

System The following estimates are based on a rerun of the model-size scaling experiment of Nordby et al. (2026) with the Qwen3 models on NVIDIA GH200 nodes, each with one H100 GPU with 96 GB of HBM3, a 72-core Grace CPU, and 116 GB of CPU memory. Every subject model fits on one GPU, so each evaluation runs on a single node. No model is fine-tuned: the subject models only run forward passes, and the detectors are trained on cached embeddings.

Compute Table 6 summarizes the requested compute. Caching the Instructed-Pairs embeddings takes about 3 h, mostly model loading. Each detector task takes its number of fits times the time per fit: 800 fits of 0.5 min for the group-lasso path, refit on 20 bootstrap resamples of the pairs in $\mathcal{D}$ for layer stability, 135 fits of 2 min for the detectors, and 20 members of 2 min for the ensembles. Forward passes over the test tasks and Alpaca dominate the cost: $N \approx 4.9 \times 10^4$ transcripts of about 1,000 tokens each give 4.9×10^7 tokens per pass at $2P$ floating-point operations per token for a P -parameter model. At an effective 100 TFLOP/s, about 10% of the dense half-precision peak, a pass takes 8.9 h for Qwen3-32B and 17 h for all six models; doubling this for padding and I/O and adding 0.5 h of loading per model gives the 36 h scoring pass. The LLM judge rates about a fifth as many transcripts with Qwen3-32B alone, about 4 h, and the analysis takes about 2 h per model.

Memory In half precision the model weights take 1.2 to 64 GB, leaving at least 30 GB of GPU memory for activations. The hidden states at all layers $\mathbb{L}$ of a batch of B sequences of length T take $2BTLd$ bytes, 2.7 GB for Qwen3-32B at $B = 2$ and $T = 2048$, so the scoring pass uses batches of 8 for models up to 8B and of 2 for the larger ones.

Storage Only the Instructed-Pairs embeddings are cached; at about 14 response tokens per transcript they take $2 \cdot 612 \cdot 14Ld$ bytes, 6 GB for Qwen3-32B and 15 GB over all six models, so every detector fits on the GPU. With 121 GB of model weights, scratch storage stays under 0.2 TB, and persistent storage for scores, weights, and results under 10 GB.

7 Limitations and Extensions

7.1 Data and Labels

Two limitations concern the data on which the detector is trained and tested. The detector is trained on a few hundred short instructed pairs and tested on transcripts written by other models, so its results may partly reflect these datasets rather than the method. Each limitation below states its effect on the results and its extension.

Small Training Set With 245 training pairs of about 14 response tokens each and up to $Ld \approx 3.3 \times 10^5$ probe weights on Qwen3-32B, the detector relies heavily on regularization (Section 3.1.2) and may overfit more as the models grow, confounding the scaling analysis. Even without overfitting, it may learn a shortcut: that the model was told to deceive, a trace that attention carries from the context into the response, rather than deception itself. Such a detector would score well on Instructed Deception but poorly on the other four test tasks, which give no such instruction. Adding the role-playing dataset of Goldowsky-Dill et al. (2025), whose scenarios give the model a reason to deceive without telling it to, would address both failures by enlarging the training set and removing the instruction.

Off-Policy Transcripts The detector is meant to score any transcript, so training on statements that f did not write is by design. The test transcripts, however, were also written by other models, and their labels are judged against those models’ beliefs, which f may not share. On Harm-Pressure Knowledge Report, for example, a response is labeled deceptive when the generating model denies knowing an answer it gave without pressure; if a small Qwen3 model never knew that answer, the same denial is honest for it. If a test task stays hard for the small models, regenerating it with each Qwen3 model through the Liars’ Bench pipelines would show whether the labels or the detector is at fault.

7.2 Scope of the Guarantees

Three limitations concern how far the guarantees of Proposition 1 and Proposition 2 extend beyond their assumptions. Proposition 1 bounds the false–positive rate for honest transcripts exchangeable with the calibration data, and Proposition 2 bounds the certainty of an alarm for the ideal ensemble under retraining on the same data. In practice, the test tasks differ from both the calibration and the training data, which limits the false–positive rate and the certainty under shift, and the finite ensemble size makes the certainty only approximate.

FPR under Shift The guarantee of Proposition 1 holds only where the honest transcripts follow the same distribution as the calibration data. However, the honest test transcripts may follow a different distribution, so their FPR is not guaranteed to stay below the calibrated level. Bounding it under shift is left open, for example by weighted conformal prediction (Tibshirani et al., 2019), calibration on honest transcripts of sensitive topics, or per-task calibration on a small honest sample.

Certainty under Shift Proposition 2 concerns certainty under retraining on the same data, not correctness: an alarm can be certain and still be wrong, since members trained on resamples of the same instructed data can err alike on a test task. The accuracy of the LCB on the test tasks, as the confidence level n grows, shows whether their disagreement tracks errors. As an extension, members trained on different deception datasets, such as Instructed-Pairs and the role-playing dataset, would disagree where those datasets transfer differently, so their disagreement would also flag the shift.

Finite Ensemble Size The certainty $n^2/(1 + n^2)$ based on Proposition 2 holds for the true mean and spread of the scores, which the ensemble estimates from a finite number of members that also calibrate its threshold, so it holds only approximately. If retrained members keep a smaller fraction of the validation alarms than the certainty bound, the approximation has failed. Training further members held out of the LCB and its calibration then gives an exact guarantee: if all of them also raise the alarm, an exact binomial bound (Clopper and Pearson, 1934) shows that a retrained detector keeps it with high probability, and the bound tightens as more members are held out.

8 Applications and Open Research Questions

The proposed detector is a supervised and static probe of the residual stream of a fixed subject: it learns from labeled transcripts, scores each token independently before pooling, and never changes the subject it reads. Relaxing the first two choices raises two open questions about the detector, and relaxing the last turns it into a training signal for honesty.

From Supervised to Unsupervised Detection Deception labels are costly to obtain and define in advance what counts as deceptive. An unsupervised detector would instead find the deception direction from consistency constraints alone, as contrast-consistent search finds truth (Burns et al., 2023). The open question is whether such a direction exists for deception, and whether it transfers across types of deception better than one learned from instructed labels.

From Static Probes to Dynamic Filtering The logistic probes score each token on its own, and the aggregator pools the scores afterward, although deception may arise, persist, or fade as a response unfolds. Transformers and BeliefNet both represent the belief state of a hidden Markov model, the posterior over its hidden states given the past tokens (Shai et al., 2024; Chen et al., 2026). Treating deception as such a hidden state would let a filter update its probability token by token, and possibly layer by layer, tracking when deception begins rather than only whether it occurs.

From Detection to Honesty Training As the critic in an actor–critic loop, the detector can fine-tune the subject f_ϕ , initialized at f , to be less deceptive:

$$\max_{\phi} \mathbb{E}_{Z \sim f_\phi^+} \left(R(Z) - \alpha \text{UCB}_{\tilde{g}}^{(n)}(f_\phi(Z)) \right) - \beta \text{KL} \left(f_\phi^+ \| f^+ \right),$$

where R is a predefined task reward and $\alpha, \beta > 0$ weigh the deception penalty and the Kullback–Leibler (KL) term. Training against a detector, however, can also teach the subject to deceive in ways the detector misses (Cundy and Gleave, 2025; Taufeeque et al., 2026). The upper confidence bound $\text{UCB}_{\tilde{g}}^{(n)} := \tilde{\mu}_s + n \tilde{\sigma}_s$ penalizes a response unless the members agree that it is honest, and the open question is whether this uncertainty-aware critic steers training toward honesty rather than undetected deception.

The three directions extend the two thrusts of this white paper. An unsupervised or dynamic detector still outputs a deception score, so the calibrated threshold of Thrust 1 (Section 3.1) and the ensemble certainty of Thrust 2 (Section 3.2) apply to it unchanged, and the score can in turn serve as the critic in honesty training. Together, they would move the detector from reading instructed deception toward tracking a model’s own deception and, ultimately, reducing it.

References

- Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.
- Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: a gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.
- Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it’s lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, 2023.
- Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.

- Aaron Chatterji, Thomas Cunningham, David J Deming, Zoe Hitzig, Christopher Ong, Carl Yan Shan, and Kevin Wadman. How people use chatgpt. Technical report, National Bureau of Economic Research, 2025.
- Reginald Zhiyan Chen, Heng-Sheng Chang, and Prashant G Mehta. Differentiable filtering for learning hidden markov models. In *Annual Learning for Dynamics and Control Conference*, pages 1035–1054. PMLR, 2026.
- Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.
- Chris Cundy and Adam Gleave. Preference learning with lie detectors can induce honesty or evasion. In *Advances in Neural Information Processing Systems*, 2025.
- Alexander D’Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022.
- Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.
- Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. Detecting strategic deception with linear probes. In *Forty-second International Conference on Machine Learning*, 2025.
- Danny Halawi, Jean-Stanislas Denain, and Jacob Steinhardt. Overthinking the truth: understanding how language models process false demonstrations. In *International Conference on Learning Representations*, volume 2024, pages 42749–42787, 2024.
- James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.
- Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. Pmlr, 2018.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- Seung Hyun Kim, Heng-Sheng Chang, Kimia Kazemi, Prashant Mehta, and Mattia Gazzola. Diverse and adaptable arm coordination for octopus-crawling via diffusion-based uncertainty-aware optimization. *arXiv preprint arXiv:2609.21138*, 2026.
- Kieron Kretschmar, Walter Laurito, Sharan Maiya, and Samuel Marks. Liars’ Bench: evaluating lie detectors for language models. *arXiv preprint arXiv:2511.16035*, 2025.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.
- Samuel Marks and Max Tegmark. The geometry of truth: emergent linear structure in large language model representations of true/false datasets. In *Conference on Language Modeling*, 2024.
- Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *International Conference on Machine Learning*, pages 6765–6774. PMLR, 2020.
- Alex McKenzie, Urja Pawar, Phil Blandfort, William Bankes, David Krueger, Ekdeep S Lubana, and Dmitrii Krasheninikov. Detecting high-stakes interactions with activation probes. *Advances in Neural Information Processing Systems*, 38:127556–127594, 2026.

- Vikram Natarajan, Devina Jain, Shivam Arora, Satvik Golechha, and Joseph Bloom. One probe won’t catch them all: towards targeted deception detection. *arXiv preprint arXiv:2602.01425*, 2026.
- National Research Council, Division of Behavioral and Social Sciences, Committee on National Statistics, Board on Behavioral and Sensory Sciences, and Committee to Review the Scientific Evidence on the Polygraph. *The polygraph and lie detection*. National Academies Press, 2003.
- Erik Nordby, Tasha Pais, and Aviel Parrack. Linear probe accuracy scales with model size and benefits from multi-layer ensembling. *arXiv preprint arXiv:2604.13386*, 2026.
- Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.
- Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1713–1721, 2015.
- Adam S Shai, Sarah E Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M Riechers. Transformers represent belief state geometry in their residual stream. In *Advances in Neural Information Processing Systems*, volume 37, 2024.
- Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245, 2013.
- Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, 2010.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: an instruction-following llama model, 2023.
- Mohammad Tafeeque, Stefan Heimersheim, Adam Gleave, and Chris Cundy. The obfuscation atlas: mapping where honesty emerges in RLVR with deception probes. *arXiv preprint arXiv:2602.15515*, 2026.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.
- Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- Xin Tong, Yang Feng, and Jingyi Jessica Li. Neyman-pearson classification algorithms and np receiver operating characteristics. *Science advances*, 4(2):eaao1659, 2018.
- Vladimir Vovk, Alexander Gammernan, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 2006.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: a top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.