CONTENT-MATCHED IS NOT LEXICALLY MATCHED: TEXT AND UNTRAINED-NETWORK BASELINES FOR EVALUATION-AWARENESS PROBING

Anonymous authors
Paper under double-blind review

ABSTRACT

Linear probes are increasingly read as evidence that language models represent whether they are being evaluated. Content-matched designs, which hold a task fixed and switch one cue at a time, are meant to make that evidence strong. The assumption is that a probe separating the two versions has detected the model recognising the cue, rather than the cue itself. We test that assumption under a pre-registered protocol on 17 open models. Matching content is not matching words: a bag-of-words classifier given the probe's labels and splits separates all eight cues of EvalAwareBench (median AUROC 0.89–1.00); replacing "Marcus Chen" with "John Smith" is the manipulation, and a word counter sees it. The standard per-cue probe test does not need learned weights either: 13 untrained networks pass it on every cue and also separate a mixed-source evaluation/deployment set. Probe success on such benchmarks therefore cannot, by itself, show that a model represents being evaluated. Two results survive both baselines. Activations add a small but robust increment beyond the words, larger in trained than in untrained networks. And when both conditions use unseen words, probes on trained networks, like pre-trained sentence encoders, still separate real from placeholder entities, while word models and untrained networks do not: they read a learned property of the cue, not evaluation. We release `lexcheck` and recommend a same-split text baseline and an untrained-network baseline for every probe claim.

1 INTRODUCTION

Whether a model "knows it is being evaluated", that is, represents in its internal state that the current input is a test rather than real use, matters for how far evaluations can be trusted, and a growing literature answers the question with linear probes: a direction in the residual stream that separates evaluation-like from deployment-like prompts is taken as evidence of an internal representation of evaluation (Nguyen et al., 2025; Abdelnabi & Salem, 2025; Chaudhary et al., 2025; Nayan et al., 2026; Heidari et al., 2026; Manek, 2026). Directions found this way are already used to steer models away from evaluation-aware behaviour (Hua et al., 2025) and as targets for input-only suppression (Mody et al., 2026); if they track the wording of the prompt rather than a state of the model, the steering and monitoring built on them inherit the problem.

Such evidence is only as good as the contrast behind it. Early studies contrasted prompts from different datasets, which differ in many ways at once; content-matched designs remove that confound by holding a task fixed and switching one cue at a time, and such contrasts already underlie probe and steering directions (Hua et al., 2025; Konrad et al., 2026; Noël, 2026). EvalAwareBench (Li et al., 2026a) does this for eight cues over 200 task templates. If a probe separates the two versions of a task, the argument goes, the model represents the cue, and that representation is evidence of evaluation awareness.

We test two steps this argument skips. We count a probe result as evidence of a represented cue only if it shows something that (i) the input text does not already reveal and (ii) a network without learned weights does not already provide. We therefore run two baselines on the same data and splits as every probe: a text classifier, and an untrained copy of the same network. Both are standard elsewhere in probing (§2), but none of the evaluation-awareness probe studies we surveyed compares

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

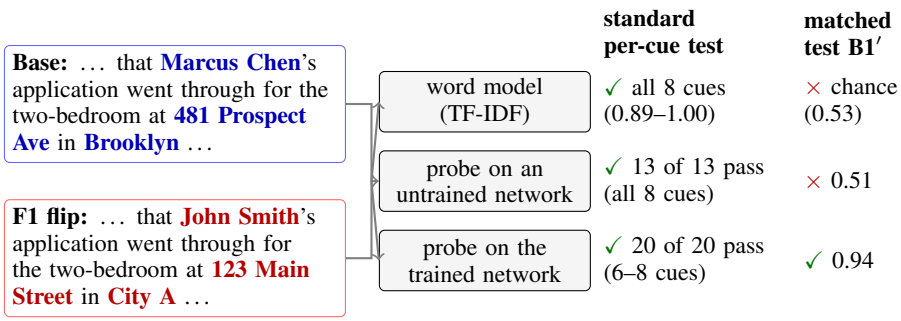


Figure 1: The two baselines at a glance. Left: a content-matched pair; the artificiality cue (F1) swaps real entities for placeholders, and the words give it away. Right: the standard per-cue probe test is passed by a word model and by untrained networks as well as by trained ones, so passing it shows only that the inputs differ. On the lexically matched test B1′ (§7), where the entities on both sides use words that occur in no original fill, only probes on trained networks, of the three readers shown, still separate real from placeholder entities (pretrained sentence encoders also do). Numbers: word-model medians over 50 scenario-grouped splits (middle column) and medians over 50 splits of B1′ (right column: word model two-sided; LR probe; trained: the four 7–8B models).

its probe with a word-level text classifier given the same labels and split, or with an untrained copy of its network. Our contributions:

- **Text alone reads every cue.** A bag-of-words classifier trained on the probe's labels and split separates all eight cues on held-out tasks (median AUROC 0.89–1.00 over 50 splits that keep paired scenarios together; §4). Content matching keeps the task but not the words, and a cue's lexical signature is two-sided: the words a cue removes identify it as well as the words it adds.
- **The standard probe tests do not need learned weights.** All 13 randomly initialised networks we built pass the pre-registered per-cue test on all eight cues, as do 6 networks that keep only their trained token embeddings; on a mixed-source evaluation/deployment set their difference-of-means probes fall within the range of trained models (§5).
- **What survives both baselines.** Activations add a few points of AUROC beyond the words on F2 and F5 ($+0.03$ to $+0.07$ on the reported split; for a regularised probe also $+0.02$ on F4), robust to resplitting, scenario-grouped splits, clustered intervals and a pretrained sentence encoder, and larger in trained than in untrained networks (§6). With the words matched on both sides, trained networks and pretrained sentence encoders still separate real from placeholder entities while untrained networks do not (§7): the probe reads a learned property of the cue, not evaluation.
- **Practice.** A pooled source-held-out AUROC can mislead in either direction (§8). We release `lexcheck`, which reports what the words alone reveal, and recommend reporting a same-split text model and an untrained copy of the network next to every probe claim (§9).

The confirmatory analyses and their decision rules were pre-registered before they were run, with 11 dated amendments, each made before the data it governs. Of 40 predictions, 21 were supported, 14 were not, 1 was inconclusive and 4 had no predicted direction; all are reported (Appendix A), and everything else is labelled exploratory.

2 Related work

Evaluation awareness. Models often identify evaluation transcripts when asked (Needham et al., 2025); probes on open models separate evaluation-style from deployment-style prompts (Nguyen et al., 2025; Abdelnabi & Salem, 2025; Chaudhary et al., 2025), later work traces such directions across scale, depth and training stage (Manek, 2026; Heidari et al., 2026) and argues that awareness

is not one capability (Nayan et al., 2026). Such directions are used for steering (Hua et al., 2025; Abdelnabi & Salem, 2025) and as targets of input-only suppression, where a placebo random direction is suppressed just as strongly (Mody et al., 2026); framing has been linked to compliance (Zhuang & Aranguri, 2026). Benchmarks range from mixed-source collections (Li et al., 2026b) to factor-controlled ones (Li et al., 2026a) and paired-prompt protocols (Burnat & Davidson, 2026; Konrad et al., 2026).

Critiques of probe evidence. Probe scores on evaluation-awareness contrasts can track prompt format rather than evaluation (Devbunova, 2026), and a probe direction can be a property of its prompt: on a standard scoring split a bag-of-words classifier reaches AUROC 0.993 and a content-free direction recovers much of published probe scores (Noël, 2026). That study does not compare a text classifier with the probe, because in its design the classifier sees labels the probe does not; ours gives both the same labels and splits, per cue, on a cue-isolating benchmark. Nayan et al. (2026) report that a logistic-regression probe keeps AUROC ≥ 0.98 under rewrites that collapse the model's behaviour and read this as an internal evaluation/deployment code; our results suggest that such robustness needs a text and an untrained-network baseline before it can be read that way, since both also separate evaluation-style contrasts. A concurrent public post probes EvalAwareBench's cues without a lexical baseline or placebos (Mati83moni, 2026); placebo variance in nuisance-controlled probing is discussed by Braun (2026).

Baselines for probes. Text-only baselines exposed annotation artefacts in natural language inference (Gururangan et al., 2018; Poliak et al., 2018); control tasks separate probe capacity from representation (Hewitt & Liang, 2019); see Belinkov (2021); Ravichander et al. (2020); Alain & Bengio (2016). Random networks are established baselines too: random encoders over trained word embeddings are strong sentence classifiers (Wieting & Kiela, 2019), standard probe accuracy does not substantially favour pretrained over randomly initialised representations (Voita & Titov, 2020), automated interpretability metrics do not distinguish trained from randomised transformers (Heap et al., 2025), and a randomly initialised model serves as a probe control in Lee & Kondor (2026). In other constructs, probes can outperform TF-IDF (McKenzie et al., 2025), for some concepts even on keyword-held-out and lexically matched sets (Lee & Kondor, 2026), while textual evidence (Boxo et al., 2025), readable answers (Villuri et al., 2026), compliance detectors (Sadhu et al., 2026) and dataset shift (Mishra et al., 2026; Yoo & Skapars, 2026) limit what probes show. Among the evaluation-awareness probe studies we surveyed we found neither an untrained-network control nor a per-cue text baseline; we bring both to this setting.

3 SETUP

Benchmark. EvalAwareBench (Li et al., 2026a) renders 100 scenarios, each as a safety task adapted from HarmBench (Mazeika et al., 2024) and a capability counterpart that shares its entities (200 task templates), through eight binary cues: F1 stimulus artificiality (placeholder entities), F2 scenario coherence, F3 prose register, F4 ecological completeness, F5 stakes telegraphing, F6 paradigm recognizability, F7 task format and F8 explicit evaluation disclosure. We use the baseline and the eight single-cue configurations of every task (1,800 prompts; the public generator re-renders all of them byte for byte). The *core* contrast compares the all-realistic with the all-cued configuration (400 prompts). The *evaluation/deployment mixture* is a 994-prompt set derived from EvalDetectBench (Li et al., 2026b) (43 sources, each carrying one label), the kind of mixed-source contrast most evaluation-awareness probe studies use. EvalAwareBench was built for behavioural and chain-of-thought study; its paired, factor-controlled design is what makes it a strict test bed for probes.

Networks. The discovery model is Qwen2.5-7B-Instruct (Qwen Team et al., 2024); the replication covers 16 further open models from 0.5B to 8B parameters in nine families (19 runs, including three base models and three instruct models without their chat template; Appendix F), all in 4-bit NF4 (Dettmers et al., 2023). For the controls we run Qwen2.5-0.5B-Instruct and Llama-3.2-1B-Instruct in bf16 in three forms: trained; *untrained*, with the same architecture, tokenizer and template and weights drawn by the model's own initialiser (10 and 3 seeds); and with trained input embeddings but random layers (5 and 1 seeds), following Heap et al. (2025). We read the residual

Table 1: The word model alone separates every cue (AUROC 0.92–1.00 on the reported split; medians 0.89–1.00 over scenario-grouped splits). Qwen2.5-7B-Instruct, EvalAwareBench single-cue flips, 60 held-out tasks. Jaccard: mean token-set overlap between the base and flipped prompt of a task; new/removed: word types only in the flipped/base prompt, pooled over tasks. Text: word TF-IDF, reported split with 95% task-bootstrap CI, and median over 50 scenario-grouped splits. LR probe: own training-CV layer. Rule: the pre-registered per-cue test. Full factorial: text AUROC when the other seven cues vary freely (51,200 prompts, whole tasks held out).

	cue	Jaccard	new/removed	text	text, grouped	DoM probe	LR probe	rule	full factorial
F1	artificiality	0.851	389/765	**1.000**	0.987	0.792 [0.74, 0.85]	0.954	✓	1.000
F2	coherence	0.786	3067/0	0.934 [0.90, 0.97]	0.893	0.949 [0.92, 0.98]	1.000	✓	0.956
F3	register	0.370	2644/2237	1.000	1.000	1.000	1.000	–	0.994
F4	completeness	0.538	329/2530	0.984 [0.97, 1.00]	0.972	0.984 [0.97, 1.00]	1.000	✓	0.982
F5	stakes	0.796	1201/1076	0.924 [0.88, 0.96]	0.918	0.924 [0.89, 0.96]	0.983	✓	0.940
F6	paradigm	0.883	306/0	0.986 [0.96, 1.00]	0.985	0.997 [0.99, 1.00]	1.000	✓	0.997
F7	format	0.475	1814/2266	1.000	1.000	1.000	1.000	✓	1.000
F8	disclosure	0.842	120/0	1.000	1.000	1.000	1.000	–	1.000

stream at the last prompt token at every depth. Wherever we re-extracted stored activations, they matched bit for bit.

Probes and text models. The pre-registered probe is a difference-of-means (DoM) direction; we also report an ℓ_2-regularised logistic-regression (LR) probe. Layers are chosen by five-fold cross-validation over training tasks only. The primary text model is word 1–2-gram TF-IDF with logistic regression (Pedregosa et al., 2012); we also run character 2–5-grams and two pretrained sentence encoders, all-MiniLM-L6-v2 (Reimers & Gurevych, 2019; Wang et al., 2020) and bge-large-en-v1.5 (Xiao et al., 2023), with logistic regression on their embeddings.

Splits and statistics. The reported split holds out 60 of the 200 tasks. The two tasks of a scenario share named entities, and in this split 46 of the 60 test tasks have their twin in training, so we also use 50 random task splits and 50 *scenario-grouped* splits that hold out 30 scenarios with both of their tasks. The pre-registered per-cue test requires a probe's 95% task-bootstrap CI to lie above the 95th percentile of 200 random directions, its refit-permutation p-value (500 permutations, Holm over eight cues) to be below 0.05, and its AUROC to exceed a length-only classifier. Each part is a control that published evaluation-awareness probe studies use: random directions (Nguyen et al., 2025; Heidari et al., 2026), a significance test and a length baseline (Nguyen et al., 2025); we require all three at once. Incremental validity stacks the probe score on the text score (logistic regression on out-of-fold training scores) and reports $\Delta = \mathrm{AUROC}(\mathrm{text{+}probe}) - \mathrm{AUROC}(\mathrm{text})$ with a task-bootstrap CI: a cue *adds* if the 95% CI excludes 0, is *confirmed* if the Bonferroni-adjusted 99.375% CI does, and is at *ceiling* if text reaches 0.999. Per-cue comparisons of a probe with text use pre-registered classes (PROBE>TEXT, TEXT>PROBE, MATCH, INCONCLUSIVE, CEILING).

Pre-registration. All confirmatory analyses were fixed before they ran, each of the 11 amendments before the data it concerns (the runners' own start times record the order), and no threshold was loosened (Appendix A). The work began as an earlier, unpublished probe study of the 7B model (the *first study*; Appendix B.1); analyses of its stored activations re-examine data on which earlier patterns had been seen; fresh evidence comes from new prompts, new models and new networks. The registration matters for that reason: it fixed which comparisons count before the results existed, and it is why the 14 predictions that failed, several of which narrowed our claims, are reported rather than dropped.

4 TEXT ALONE NEARLY SATURATES THE BENCHMARK

Table 1 gives the head-to-head. The bag-of-words classifier separates every cue on held-out tasks (permutation $p = 0.002$, the floor for 500 permutations; Holm-corrected 0.016). Holding paired scenarios out together lowers it by about 0.01 on F1 and F2 only; F2's median of 0.893 falls below the 0.90 we had predicted, so we report its task-split value as inflated by twin leakage. Over the full 51,200-prompt factorial, with every other cue varying, each cue remains readable from text

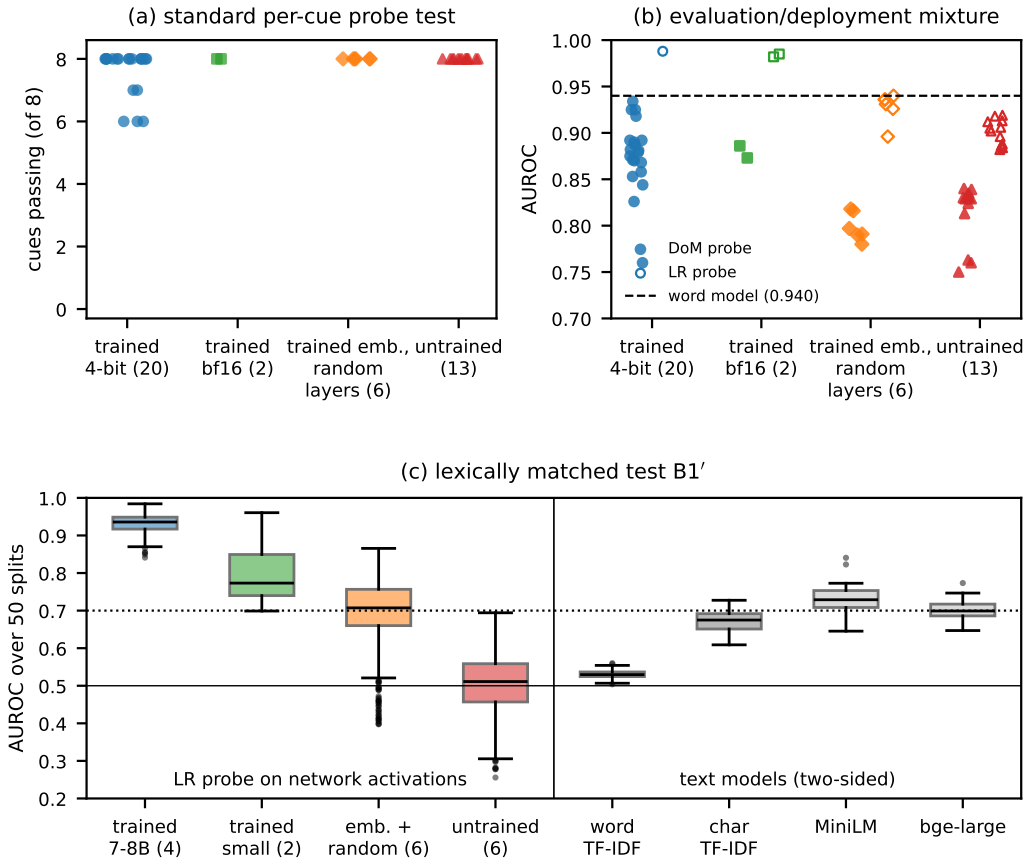


Figure 2: Networks that differ only in whether their weights were learned. (a) Cues passing the pre-registered per-cue test (of 8), one point per network. (b) Evaluation/deployment mixture, same split, layer chosen on training data: DoM (filled) and LR (open) probes; dashed line: word model. (c) Lexically matched test B1′ (§7), AUROC of the LR probe over 50 splits per group, next to the four text models (two-sided); dotted line: 0.70.

(0.940–1.000). Against the pre-registered DoM probe on the reported split, the paired comparison classes F1 as TEXT>PROBE (0.792 against a perfect text model), F4 as MATCH, F2, F5 and F6 as INCONCLUSIVE and F3, F7 and F8 as CEILING.

The reason is visible in the prompts. Content matching keeps the task, not the words: the mean Jaccard overlap between the two versions of a task ranges from 0.370 (F3) to 0.883 (F6), and every cue introduces word types that never occur on the base side. The volume of change does not predict separability: F1 changes 15% of word types and is perfectly separable because the change is consistent across tasks (*example*, *doe*, *smith*, *acme* recur). A cue's signature is also two-sided. When we replaced F1's placeholders with a new style (§7), the word model's shift toward "flip" came almost entirely from realistic words present in the base and absent from the flip (summed contribution $+4.65$ against -0.15 from the new words): the benchmark's realistic fills reuse a small pool of names (*Chen* in 46 of 200 templates, *Marcus* in 32, *Meridian* in 30). This is not a foregone conclusion: in other constructs probes can outperform text (McKenzie et al., 2025), for some concepts even on keyword-held-out and lexically matched test sets (Lee & Kondor, 2026), so how much a benchmark leaves to text is a property of its design that has to be measured.

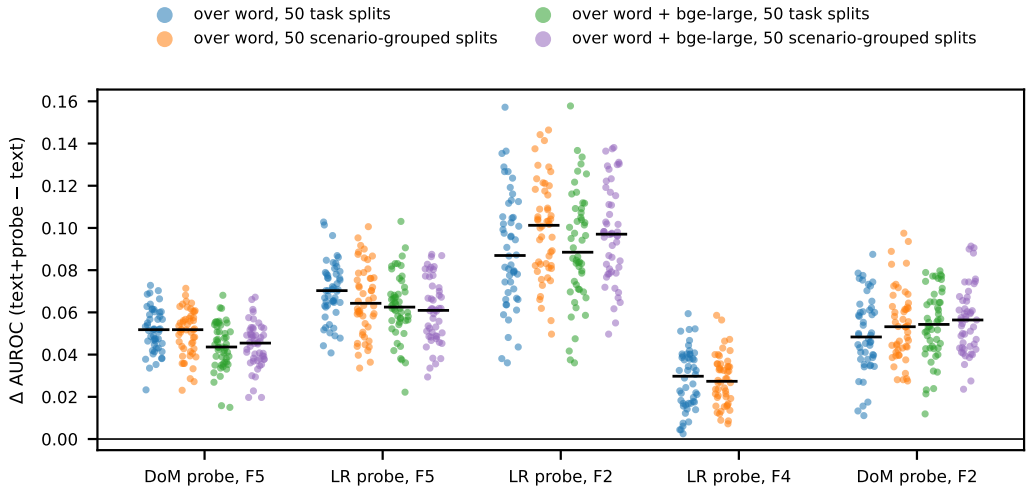


Figure 3: Increment of text+probe over text for Qwen2.5-7B-Instruct in every one of 50 random task splits and 50 scenario-grouped splits, with the word model alone or with the word model and a pretrained sentence encoder as the text side; one point per split, bars at medians. Every split is above zero. For the LR probe on F2 and F4 the probe alone reaches 1.000, so the increment equals the text model's headroom; F5 is the informative cue. Point estimates and Bonferroni-adjusted intervals on the reported split: Table 6.

5 The standard probe tests do not need learned weights

Figure 2 compares networks that differ only in whether their weights were learned. Under the pre-registered per-cue test (panel a), the 20 trained runs pass 6–8 of 8 cues, while every one of the 13 untrained networks passes all eight, as do the 6 networks with trained embeddings and random layers. Even on the artificiality cue F1, untrained DoM probes reach 0.728–0.914 against 0.704–0.881 for the trained runs (Appendix B). For four fresh untrained seeds this was pre-registered (at least 6 of 8 each; all passed 8), and it held in a second family. The two cues the 7B model fails, F3 and F8, fail because they are so easy that directions refitted on shuffled labels also separate them (Holm-corrected p 0.058 and 0.056), as random directions do (95th percentiles 0.975 and 0.969), not because the probe is weak.

On the evaluation/deployment mixture (panel b), untrained DoM probes reach 0.750–0.840, inside the 0.760–0.934 of the 20 trained 4-bit runs; untrained LR probes reach 0.882–0.919 against 0.982–0.988 for the three trained networks where we ran one; the word model reaches 0.940. For five fresh untrained seeds this was pre-registered (DoM 0.824–0.832, LR 0.886–0.919). Random directions are no better a reference: on the core contrast (all cues on versus all off), the 95th percentile of 200 random directions is 0.943 on the 7B model and 0.914–0.970 in every replication run.

Passing these tests therefore shows that the input conditions are separable, which any network that sees the tokens can reflect, not that a trained model represents them; Heap et al. (2025) found the same for sparse-autoencoder metrics. Panel (c) previews the one test in this paper that trained and untrained networks do not pass alike.

6 What activations add beyond the words

Separability cannot decide between the readings; incremental validity asks what activations add beyond the words. Our pre-registered prediction was pessimistic (no cue adds), and it failed. On the 7B model, stacking the probe on the word model improves AUROC on F2 and F5 for both probes and on F4 for the LR probe, and four of these cells are confirmed at 99.375% (Table 6 in Appendix B). Across the 19 replication runs the DoM probe adds in 26 of 76 (run, cue) cells with headroom (F5 in 13 runs, F2 in 11, F4 in 2, F6 in none; Appendix B), where we had predicted at

Table 2: With new words on both sides (B1′) the word model falls to chance while the probes still separate real from placeholder entities; on the swap, which changes only real names, it is the other way round. New-vocabulary tests and positive control (7B; 60 held-out tasks, 30 capability tasks for F5). Probes and text models are trained on the original contrast. B1: new placeholder style (F1) or new wording (F5). B1′: new realistic entities in the base and new placeholders in the flip, so that neither side shares a word with any original fill. Swap: original versus new realistic prompts. B2: "Related words" appended, animals versus tools, k = 1 or 4 words, given to two decimals. The B1′ medians over 50 splits give text models two-sided, $\max(a, 1-a)$. Dashes: not computed.

test (base vs. flip)	DoM	LR	word	char	MiniLM	bge-large
F1 original	0.792	0.954	1.000	0.999	0.905	0.965
F1 B1, new placeholders	0.730	0.924	0.863	0.854	0.712	0.667
F1 B1′, reported split	0.785	0.930	0.460	0.647	0.764	0.710
F1 B1′, median of 50 splits	0.816	0.943	0.530	0.675	0.729	0.699
F1 swap only	0.454	0.520	0.873	0.756	0.458	0.479
F5 original, same 30 tasks	0.957	0.992	0.986	1.000	–	–
F5 B1, new wording	0.898	0.947	0.778	0.730	0.754	0.733
B2 fixed words, k = 1 / 4	1.00 / 0.99	1.00 / 1.00	1.00 / 1.00	1.00 / 1.00	0.93 / 0.91	0.97 / 0.94
B2 held-out words, k = 1 / 4	0.65 / 0.78	0.86 / **0.97**	0.50 / 0.50	0.58 / 0.82	0.66 / 0.78	0.81 / 0.93

most 10%; on F5 its increment is positive in every one of 50 splits for Llama-3.1-8B, Mistral-7B and OLMo-2-7B ($+0.033$ to $+0.060$; exploratory).

The increment survives resplitting (positive in every one of 50 random splits, exploratory, and of 50 scenario-grouped splits, pre-registered), a scenario-cluster bootstrap (all confirmed cells stay confirmed; pre-registered), and a pretrained sentence encoder added to the text side (confirmed on the reported split and positive in every split; pre-registered; Figure 3). Three simpler explanations fail (exploratory): stacking a character model on the word model gains nothing (-0.004 to $+0.006$) while the probe still adds on top of both; a word-count feature leaves it unchanged; and on F5 it sits in the safety tasks, where the cue restates the harmful request in varied words and text is weaker ($+0.086$; text 0.889), not in the capability tasks ($+0.012$; text 0.986). It shrinks as the text side gets stronger (DoM F5: $+0.061$ over words, $+0.048$ with bge-large, $+0.030$ with MiniLM, which reads F5 best).

Is the increment learned? An untrained copy adds on none of the cues on the reported split, but over 50 splits its F5 increment is above zero in 82% of splits, where we had required fewer than 80%. By our pre-registered reading, the increment is therefore not by itself evidence of a learned representation. Across all 13 untrained networks, this share ranges from 32% to 88%, with medians of -0.000 to $+0.004$, against $+0.049$ in 98% of splits for the trained twin; paired on the same splits, the trained twin's F5 and F2 increments exceed those of each fresh untrained network in 98–100% of splits (pre-registered). Trained activations thus carry a small, robust increment that untrained ones barely show. The benchmark cannot show more: on F1, F3, F7 and F8 text is at ceiling and no increment is measurable, which is the design problem of §4.

On their own, the probes compare with text as follows over 50 splits (pre-registered classes): the LR probe is PROBE>TEXT on F2, F4 and F5 and TEXT>PROBE on F1; the DoM probe is PROBE>TEXT only on F2, in exactly 90% of splits, and TEXT>PROBE on F1.

7 Controlling the words

If a probe reads a property rather than words, it should still separate the conditions when both are expressed in words it never saw. We rebuilt F1's flip with a new placeholder style ("Person Q", "Firm R") sharing no word with any original placeholder, and F5's flip in every capability task with a synonym map that removes its 50 most predictive words (B1; Table 2). The probes lost 0.03–0.06 AUROC and the word model 0.14–0.21, but the word model still separated the new prompts (0.863 and 0.778), so by the pre-registered rule the comparison is uninformative: the realistic words the flip removes still give it away (§4).

A lexically matched test. We therefore also gave the base prompts new realistic entities ("Tadeo Ferreira", "Esposito Brands", "Duluth"), so that no word on either side occurs in any original fill (B1′, registered before its data existed; examples in Appendix C). On the reported split the word model falls to chance (0.460) and the character model to 0.647, while the F1 probes still separate real from placeholder entities (DoM 0.785, LR 0.930) and no other cue's direction does (0.45–0.55). The converse holds too: text separates the original from the new realistic prompts (0.873), which differ only in which real names they use, while the probes treat them alike (0.454 and 0.520).

The matching holds at the word level only. Over 50 splits the word model stays at chance in every split, but the character model reaches 0.61–0.73 (median 0.675) through fragments shared by the two placeholder styles and through ordinary name spellings (107 character n-grams carry half of its signal, too many to remove by rule), and the pre-registered class ABSTRACT (probe ≥ 0.70, word and character models both ≤ 0.65) holds in only 12 of 50 splits, so we report it as split-dependent. Pretrained sentence encoders read the contrast too (medians: MiniLM 0.729, bge-large 0.699) and, like the probes, not the swap. The LR probe stays at or above 0.84 in every split and exceeds every text model we ran, including both encoders, in all 50 (smallest margin $+0.08$); the DoM probe exceeds the word and character models in all 50 splits and the encoders in 44.

Which networks read it? Trained networks do (Figure 2c): the four 7–8B models have LR medians of 0.906–0.945 over 50 splits, and the small trained twins reach 0.70 in 98% and 100% of splits. Untrained networks never reach 0.70 in the direction learned from the original contrast (6 networks, 50 splits each), but they are not always at chance: one seed lies between 0.65 and 0.70 in 9 splits, so our stricter prediction (below 0.65 in 90% of splits for every untrained network) failed, and by our registered reading untrained networks sometimes read B1′, weakly; another seed separates the two sets in the reverse direction in 6 of 50 splits (two-sided AUROC ≥ 0.70; exploratory). Networks with trained embeddings and random layers fall in between (median 0.707, depending on the seed). Separating B1′ as trained networks do thus needs learned weights, and it is shared with pretrained sentence encoders: the probe reads a learned, linearly readable property of the cue, real versus placeholder entities, not something specific to the model studied, and not evaluation. Larger models separate the unseen-word F1 contrasts better (Spearman ρ with parameter count across 19 runs: $+0.66$ for both probes on B1 and $+0.60$ for the DoM probe on B1′, but only $+0.28$, $p = 0.26$, for the LR probe on B1′; descriptive).

Positive control. To check that the pipeline can read a word-independent signal at all, we appended "Related words: $w_1, \ldots, w_k$" to every base prompt, with animals in one class and tools in the other. With the same words in training and test, everything separates. With 100 unseen animals and tools at test time, the word model is at chance (0.50) while the LR probe reaches 0.970, and it stays at or above 0.91 in all 18 completed replication runs. Untrained networks do not (0.473–0.539), nor do networks with trained embeddings alone (0.547–0.681), while a pretrained encoder does (bge-large 0.933). As predicted, the word model depends on the consistency of the added words, not on their number; the character model partly reads the category from word shape (0.58 $\rightarrow$ 0.82), which we had glimpsed on a synthetic test before the run and recorded in an amendment.

8 A PITFALL: POOLED SOURCE-HELD-OUT AUROC

A natural test of generalisation holds out whole data sources and pools the out-of-fold scores into one AUROC. On the mixture every source carries one label and one source contributes 28% of the prompts, so folds differ sharply in label balance. With labels shuffled across sources and the whole procedure rerun, the pooled text AUROC averages 0.266 [0.092, 0.529] rather than 0.5 (200 permutations). The observed 0.573, which we had first read as "near chance, so same-split success is source recognition", lies above this null ($p = 0.02$), and within folds the text model reaches 0.72–0.89. For the probe the distortion runs the other way: a pooled 0.904 against a mean within-fold 0.696; across the 19 replication runs pooled DoM values range from 0.24 to 0.87, within-fold means from 0.61 to 0.90. The pooled number is not a safe summary for either model; `lexcheck` reports per-fold values and a permutation null next to it.

9 A RELEASED CHECK AND TWO BASELINES

`lexcheck` (supplementary material; MIT licence) reports, for any labelled prompt set with a grouping (task, source, template), the held-out AUROC of a word-level text classifier with a group-bootstrap CI and a length baseline, the words that drive it in both directions, and, for paired data, the per-condition Jaccard overlap and the numbers of new and removed word types; with grouped cross-validation it reports per-fold AUROCs next to the pooled value and can calibrate the pooled value against a group-level permutation null. It reproduces the reported-split word-model columns of Table 1 and the text results of §8; installing it, running its tests and reproducing Table 1's word-model numbers takes under two minutes on a laptop CPU. On the paired prompts of Noël (2026), it separates held-out evaluation from deployment framings perfectly within every fold (pooled: 0.807).

Table 3 turns our results into a reporting protocol for probe studies on paired or factor-controlled prompts; `lexcheck` computes the word-model part of its first row, the added and removed word types of its fourth and all of its fifth, and our extraction script's random-initialisation option provides the second.

Table 3: What to report with a probe claim on paired or factor-controlled prompts, what each item rules out, and what it showed on EvalAwareBench.

report	rules out	here
a text model on the same split (words; also character n-grams and a sentence encoder)	a probe that reads what the words already reveal	text 0.89–1.00 on every cue (§4)
an untrained copy of the network, through the same pipeline	separability that needs no learned weights	all 13 untrained networks pass the per-cue test (§5)
incremental validity over text, across resplits and with clustered intervals	a gain that is a split, twin or ensemble artefact	$+0.03$ to $+0.07$ on F2 and F5, robust (§6)
the words each condition removes as well as adds; a test matched on both sides	leakage through removed words	word model at chance; trained, not untrained, networks still separate (§7)
per-fold next to pooled source-held-out AUROC	bias from single-label sources	pooled text null 0.27, not 0.5 (§8)

10 LIMITATIONS AND CONCLUSION

We read one position (the last prompt token) of 4-bit models up to 8B parameters; untrained controls cover two small models in bf16 (a 7B model in bf16 does not fit our 6 GB GPU); other readouts, larger models and causal tests may show more. The first study's steering test on F5 was null (Appendix B.1), and we found no evidence that the 7B probe tracks the model's own verbal judgement of being tested beyond the words (partial Spearman $\rho = +0.038$, $p = 0.69$; pre-registered, not supported). The new prompt sets were built by rules and checked by construction, not by human raters; the new realistic fills are sometimes inconsistent with each other (Appendix C), which makes the realistic side look less real and so works against the separation we report. The lexically matched test matches words, not characters or meaning, and covers one cue. Most analyses of the 7B model reuse activations on which earlier patterns had been seen.

Content-matched benchmarks were built to make probe evidence about evaluation awareness interpretable. As built, they make it readable from text, and the standard probe tests they are paired with are passed by networks that never learned anything. What survives both baselines is narrower: a small increment beyond the words that trained networks carry more than untrained ones, and a matched test on which trained networks and pretrained encoders read a cue's property while untrained networks and word models do not. That shows a learned representation of the cue's property, not of being evaluated; connecting the two needs behavioural or causal evidence that probes alone cannot provide. Claims that a model represents evaluation should be stated against what the text and an untrained network already reveal, and `lexcheck` makes the first comparison cheap.

AI USE STATEMENT

The research question and the paper's central idea, that evaluation-awareness probes may read the prompt rather than evaluation, were the authors', as were the first study's hypotheses and some of the later ones; the authors reviewed every design, approved every pre-registration and amendment, and decided the final claims. We used generative AI tools as assistants, under the authors' direction: to suggest further hypotheses and refine all of them, to design experiments (including the untrained-network, sentence-encoder and scenario-grouped controls), to help develop the paper's two-baseline framing, to give feedback on methodology, to implement the analysis code, to generate synthetic validation data and the rule-based prompt variants and word lists of §7 (built by code from the benchmark's own generator), to formulate and derive the noise level of the cosine between cue directions used in prediction RP3 (checked by simulation), to run the CPU analyses, and to interpret results. The lexical baseline came out of an AI-assisted audit of the first study (Appendix B.1), which also used these tools for its data preparation, its code and the text of its pre-registration; this work reuses its prompt sets and stored activations. No reported number was generated by an AI tool itself: every number comes from a logged run of code whose checksum, inputs and library versions are recorded. Assistance with translation and qualitative data analysis is not applicable to this work, and the paper contains no proofs. Additionally, we used generative AI tools for literature search and summaries, for creating the figures and tables and the released package, for drafting and editing the text of this paper, including its structure, title and keywords, and for formatting references. We have reviewed all AI-assisted work: the confirmatory analyses were fixed in a dated pre-registration before they were run, and each amendment before the data it concerns (Appendix A); analysis code was tested on synthetic data with planted signals before use; every number in the paper was checked against its log; and every reference was checked against its source (arXiv, publisher or web page). We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The benchmark contains harmful-request templates adapted from HarmBench; we use its prompts only as model inputs, generate no completions, and release no new harmful content. The new prompt sets replace names and wording in existing templates. All datasets are used under their licences (EvalAwareBench CC BY-NC 4.0, EvalDetectBench data CC BY-NC-SA 4.0). EvalAwareBench's data card lists use as training data for machine-learning systems as out of scope, while its licence permits non-commercial research training; we fit only small measurement classifiers (linear probes and text classifiers), train no generative model and release no trained model.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix F give the data, models (with revisions), networks and statistics; Appendix D gives every probe, text-model, split and bootstrap setting, and Appendix E how the new prompt sets were built. The supplementary material contains `lexcheck` with tests and pinned requirements, the extraction and analysis scripts with their checksums, the builders of the new prompt sets, the pre-registration with all dated amendments, the collectors that score each prediction, and the logs of every run cited here. Appendix A lists what was registered and when, and every deviation.

REFERENCES

Sahar Abdelnabi and Ahmed Salem. The Hawthorne Effect in Reasoning Models: Evaluating and Steering Test Awareness. *arXiv preprint arXiv:2505.14617*, 2025. Version v3.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. Version v4.

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *arXiv preprint arXiv:2102.12452*, 2021. Version v4.

Gerard Boxo, Aman Neelappa, and Shivam Raval. Linear probes rely on textual evidence: Results from leakage mitigation studies in language models. *arXiv preprint arXiv:2509.21344*, 2025. Version v2.

Florian Braun. Excess Separability: Nuisance-Controlled Residual-Stream Probing for Benchmark Contamination Detection. *arXiv preprint arXiv:2608.12652*, 2026. Version v2.

Florian A. D. Burnat and Brittany I. Davidson. Measuring Evaluation-Context Divergence in Open-Weight LLMs: A Paired-Prompt Protocol with Pilot Evidence of Alignment-Pipeline-Specific Heterogeneity. *arXiv preprint arXiv:2605.06327*, 2026. Version v1.

Maheep Chaudhary, Ian Su, Nikhil Hooda, Nishith Shankar, Julia Tan, Kevin Zhu, et al. Evaluation Awareness Scales Predictably in Open-Weights Large Language Models. *arXiv preprint arXiv:2509.13333*, 2025. Version v2.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314*, 2023. Version v1.

Viliana Devbunova. Is Evaluation Awareness Just Format Sensitivity? Limitations of Probe-Based Evidence under Controlled Prompt Structure. *arXiv preprint arXiv:2603.19426*, 2026. Version v1.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation Artifacts in Natural Language Inference Data. *arXiv preprint arXiv:1803.02324*, 2018. Version v2.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Automated Interpretability Metrics Do Not Distinguish Trained and Random Transformers. *arXiv preprint arXiv:2501.17727*, 2025. Version v2; ICLR 2026.

Farzaneh Heidari, Amin Memarian, and Guillaume Rabusseau. Evaluation Awareness in Language Models: Representation, Verbalization, and Control. *arXiv preprint arXiv:2608.21766*, 2026. Version v1.

John Hewitt and Percy Liang. Designing and Interpreting Probes with Control Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275.

Tim Tian Hua, Andrew Qin, Samuel Marks, and Neel Nanda. Steering Evaluation-Aware Language Models to Act Like They Are Deployed. *arXiv preprint arXiv:2510.20487*, 2025. Version v5.

Phongsakon Mark Konrad, Toygar Tanyel, and Serkan Ayvaz. Routing Subspaces: Auditing Evaluation-to-Deployment Mismatch in Fine-Tuned Language Models. *arXiv preprint arXiv:2607.20436*, 2026. Version v1.

Su Hyeong Lee and Risi Kondor. Deep Minds and Shallow Probes. *arXiv preprint arXiv:2605.11448*, 2026. Version v1.

Changling Li, Terry Jingchen Zhang, Jie Zhang, Zhijing Jin, Sahar Abdelnabi, and Maksym Andriushchenko. Decomposing and Measuring Evaluation Awareness. *arXiv preprint arXiv:2605.23055*, 2026a. Version v2.

Xinning Li, Kemunto Ochwang'i, Aryasomayajula Ram Bharadwaj, Alexandra Souly, and Robert Kirk. EvalDetectBench: A Benchmark for Measuring Evaluation Awareness in Frontier Language Models. *arXiv preprint arXiv:2609.01611*, 2026b. Version v1.

Archit Manek. Representational Depth of Evaluation Awareness Shifts With Scale in Open-Weight Language Models. *arXiv preprint arXiv:2606.29196*, 2026. Version v1.

Mati83moni. Is evaluation awareness one direction, or eight? A factor-controlled decomposition. Hugging Face forum post, `https://discuss.huggingface.co/t/is-evaluation-awareness-one-direction-or-eight-a-factor-controlled-decomposition/180448`, 2026. Accessed 2026-09-23.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, et al. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. *arXiv preprint arXiv:2402.04249*, 2024. Version v2.

Alex McKenzie, Urja Pawar, Phil Blandfort, William Bankes, David Krueger, Ekdeep Singh Lubana, et al. Detecting High-Stakes Interactions with Activation Probes. *arXiv preprint arXiv:2506.10805*, 2025. Version v4.

Nishant Mishra, Ameen Abu-Hanna, and Iacer Calixto. Investigating Linear Probe Robustness to Linguistic Register, Medical Specialty, and Corpus Shifts in Medical QA. *arXiv preprint arXiv:2609.01361*, 2026. Version v1.

Deepanshu Mody, Samarth Agarwal, Utkarsh Mittal, and Dipesh Mahato. Minimizing Targeted Activations: Input-Only Suppression of Evaluation-Awareness Latents in Large Language Models. *arXiv preprint arXiv:2607.25907*, 2026. Version v1.

Nilesh Nayan, Aishwarya Sampath Kumar, Rishiraj Girmal, Shivani Anilkumar, Sankaran Vaidyanathan, David A. Nader Palacio, et al. Evaluation Awareness Is Not One Capability: Evidence from Open Language Models. *arXiv preprint arXiv:2606.23583*, 2026. Version v1.

Joe Needham, Giles Edkins, Govind Pimpale, Henning Bartsch, and Marius Hobbhahn. Large Language Models Often Know When They Are Being Evaluated. *arXiv preprint arXiv:2505.23836*, 2025. Version v3.

Jord Nguyen, Khiem Hoang, Carlo Leonardo Attubato, and Felix Hofstätter. Probing and Steering Evaluation Awareness of Language Models. *arXiv preprint arXiv:2507.01786*, 2025. Version v2.

Valentin Noël. A Probe Direction Is a Property of Its Prompt. *arXiv preprint arXiv:2608.13329*, 2026. Version v1.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, et al. Scikit-learn: Machine Learning in Python. *arXiv preprint arXiv:1201.0490*, 2012. Version v4.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis Only Baselines in Natural Language Inference. *arXiv preprint arXiv:1805.01042*, 2018. Version v1.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. Version v2.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the Probing Paradigm: Does Probing Accuracy Entail Task Relevance? *arXiv preprint arXiv:2005.00719*, 2020. Version v3.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*, 2019. Version v1; EMNLP 2019.

Saisab Sadhu, Aadit Sengupta, Vinay Kumar Sankarapu, and Pratinav Seth. What Do Compliance Detectors Read? An Audit of Activation Probes and Guard Models. *arXiv preprint arXiv:2608.16852*, 2026. Version v1.

Gnaneswar Villuri, Hashmath Shaik, and Alex Doboli. When Do Internal Probes Beat Reading the Answer? Miscalibrated Readouts and Behavior-Concealed Knowledge in Language Models. *arXiv preprint arXiv:2609.04582*, 2026. Version v1.

Elena Voita and Ivan Titov. Information-Theoretic Probing with Minimum Description Length. *arXiv preprint arXiv:2003.12298*, 2020. Version v1.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. *arXiv preprint arXiv:2002.10957*, 2020. Version v2.

John Wieting and Douwe Kiela. No Training Required: Exploring Random Encoders for Sentence Classification. *arXiv preprint arXiv:1901.10444*, 2019. Version v1; ICLR 2019.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed Resources For General Chinese Embeddings. *arXiv preprint arXiv:2309.07597*, 2023. Version v5; SIGIR 2024.

Daniel Yoo and Adrians Skapars. Probe Generalization as Subspace Selection for OOD Deception Detection. *arXiv preprint arXiv:2609.02893*, 2026. Version v1.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild. *arXiv preprint arXiv:2405.01470*, 2024. Version v1.

Allison Zhuang and Santiago Aranguri. Not All Eval-Awareness Is Equal: Capabilities Framing Predicts Compliance. *arXiv preprint arXiv:2608.27340*, 2026. Version v1.

A PRE-REGISTRATION: HISTORY, OUTCOMES AND DEVIATIONS

History. The first study was pre-registered before its first runs. A later code audit found that its per-cue layers had been chosen on the test split, that placebo counts differed between panels and that one transfer result used orientation-free AUROC; the corrected analysis implements the registered protocol. Its results (Appendix B.1) were known when the pre-registration governing this paper was written, which is why its analyses of the stored 7B activations are confirmatory about the analysis plan, not about fresh data. That pre-registration was frozen before any run it lists and amended eleven times (Table 4). Each amendment records what had already been seen; the runners' own start lines show that every amendment precedes the data it governs. The clock times typed in some amendment headings are up to about six minutes later than the moment the text was saved; a dated note in the file records this.

Table 4: Pre-registration timeline (dates 2026; local time). "Before" names the first run the entry governs and its start time as printed by the runner.

entry	what it fixed	before
frozen, Sep 23 16:15	Experiment 1 (incremental validity, N1, N2), split robustness and placebo stability (N4–N6), new-vocabulary tests and positive control (N3, N3b, N7, N8), released check, replication (RP1–RP4)	Experiment 1, Sep 23 21:18
A1, Sep 23 21:53	cosine layer at each model's middle depth (unchanged for the 7B)	replication extraction, Sep 24 02:36
A2, Sep 24 12:27	erratum (the 7B passes 6 of 8 cues, not 2); frozen B-series prompt sets; analysis details; disclosure of a previewed character-model result; exploratory checks declared	B-series extraction
A3, Sep 24 15:45	B1′ and N3′	B1′ extraction, Sep 24 19:27
A4, Sep 24 19:20	untrained-network control RW1, RW2; B-series replication RB1, RB2	Sep 24 19:27
A5, Sep 25 10:20	RW3–RW6 (B1′, positive control, fresh seeds, second family), V1 (verbal report), descriptive trends	Sep 25 11:15
A6, Sep 25 14:05	sentence encoders on B1′ (SE1); trained embeddings with random layers (RE1, RE2)	Sep 25 14:07
A7, Sep 25 15:20	increment over word + sentence encoder (SE2)	Sep 25 15:38
A8, Sep 25 16:00	SE2 and B1′ over 50 splits (SE2-50, B1P-50)	Sep 25 16:03
A9, Sep 25 16:44	untrained vs trained on B1′ over 50 splits (RW3-50); encoders over 50 splits (SE1-50)	Sep 25 17:16
A10, Sep 25 16:49	fresh untrained seeds on the mixture (RW7); paired trained-minus-untrained increment (RW8)	Sep 25 18:17
A11, Sep 25 17:01	scenario-grouped splits, scenario-clustered intervals, LR solver seeds (TG1–TG5)	Sep 25 17:13

Deviations and disclosures. (1) Amendment 2 corrects a reference value we had misread: prediction RP4 quoted the 7B model as passing 2 of 8 cues, which counted only the three cues the first study called clean; the verdict is computed from the data and does not change. (2) While testing the positive-control analysis on synthetic activations with the real word lists, the character model's held-out result was effectively previewed; nothing was changed after it. (3) The B-series extraction for the base model gemma-2-2b ran out of GPU memory; it is reported as failed, not replaced, and the replication predictions are scored on the 18 completed runs. (4) The optional LR probe was not run on the 19 replication runs. (5) The positive control cannot be resplit, because its held-out words follow the reported split's task partition; its robustness rests on its CI, the replication runs and the untrained seeds. (6) Each prediction is scored by its own pre-registered rule; we make no family-wise claim across the 40 predictions.

Table 5: Every pre-registered prediction and its outcome (7B model and reported split unless stated).

	prediction	outcome
N1	activations add nothing beyond word text on any cue	not supported: 4 cells confirmed at 99.375%
N2	the probe's same-split score on the mixture is mostly source recognition	inconclusive (pooled estimator, §8)
N3	F1 probe survives new placeholder words (ABSTRACT)	not supported (SURFACE-LEFT)
N3b	F5 LR probe survives new wording (ABSTRACT)	not supported (SURFACE-LEFT)
N4, DoM	over 50 splits no cue PROBE>TEXT; F1 TEXT>PROBE	not supported (F2 PROBE>TEXT)
N4, LR	F2, F5 PROBE>TEXT; F4, F6 tie or unstable; F1 TEXT>PROBE	not supported (F4 PROBE>TEXT)
N5	a character model is within 0.01 of the word model and narrows the LR gap	not supported (weaker on F2, F5)
N6	F1, F5 above the placebo 95th percentile in $\geq$95% of placebo sets and $\geq$80% of splits	supported (100%, 100%)
N7, word	word text follows the consistency, not the number, of added words	supported
N7, char	the same for the character model	not supported (0.58 $\rightarrow$ 0.82; previewed)
N8	LR probe $\geq$0.80 on held-out category words, $k = 4$	supported (0.970)
RP1	random directions separate the core contrast (95th percentile $\geq$0.90) in $\geq$80% of runs	supported (19/19)
RP2	the DoM probe adds in $\leq$10% of (run, cue) cells	refuted (26/76 = 34%)
RP3	mean cosine between cue directions $<$0.30 in $\geq$80% of runs	supported (18/19)
RP4	at most 3 of 8 cues pass the per-cue test in $\geq$80% of runs	not supported (6–8 pass everywhere)
N3′	B1′: ABSTRACT for at least one probe	supported on the reported split
RW1	reported split: the untrained copy adds on none of F2, F4, F5, F6; the trained twin adds on F2 or F5	supported
RW2	50 splits: untrained F5 increment $>$0 in $<$80% of splits; trained in $\geq$95%	not supported (untrained 82%)
RB1	N8 holds in $\geq$80% of completed replication runs	supported (18/18)
RB2	LR probe $\geq$0.70 on new F1 placeholders in $\geq$80% of runs	supported (18/18)
RW3	B1′: untrained ABSTRACT for neither probe in $\geq$4 of 5 seeds; trained LR ABSTRACT	supported
RW4	positive control: untrained LR $<$0.80 in $\geq$4 of 5 seeds; trained $\geq$0.80	supported
RW5	fresh untrained seeds 1–4 each pass $\geq$6 of 8 cues	supported (8, 8, 8, 8)
RW6	Llama-3.2-1B-Instruct: RW1–RW5, one untrained seed	all supported
V1	the core probe tracks the model's own verbal "is this a test?" judgement beyond text	not supported ($\rho = +0.038$, $p = 0.69$)
SE1	B1′ read by a pretrained sentence encoder (no direction)	read (MiniLM 0.764, bge-large 0.710)
RE1	trained embeddings with random layers read B1′ (no direction)	mixed (3 of 5 seeds)
RE2	positive control with trained embeddings alone (no direction)	needs trained layers (0.547–0.681)
SE2	the probe adds beyond word + bge-large (DoM F5; LR F2 or F5)	supported
SE2-50	SE2 over 50 splits ($>$0 in $\geq$90%, median $\geq$0.02)	supported (100%)

	prediction	outcome
B1P-50	B1′: LR ABSTRACT in ≥80% of 50 splits	not supported (12/50)
SE1-50	encoders on B1′ over 50 splits (no direction)	MiniLM robust, bge-large mixed
RW3-50	untrained LR <0.65 in ≥90% of splits (each of 6); trained ≥0.70 in ≥80% (each of 2)	not supported (one seed 82%)
RW7	fresh untrained seeds separate the mixture (DoM ≥0.75, LR ≥0.85) in ≥4 of 5	supported (5/5)
RW8	trained increment exceeds untrained in ≥90% of splits, ≥4 of 5 fresh seeds (F5; F2)	supported
TG1	scenario-grouped splits: word median ≥0.90 on every cue	not supported (F2 0.893)
TG2	scenario-grouped splits: F5 and F2 increments >0 in ≥90%, median ≥0.02	supported
TG3	scenario-grouped splits: F5 increment over word + bge-large	supported
TG4	scenario-cluster 99.375% intervals keep the six confirmed cells	supported
TG5	LR solver-seed spread ≤0.005	supported (0.0000)

B ADDITIONAL RESULTS

Table 6: Increment of text+probe over text (Δ AUROC), Qwen2.5-7B-Instruct. Reported split: point estimate and Bonferroni-adjusted 99.375% intervals from a task or a scenario bootstrap. Over 50 splits: median and, in parentheses, the percentage of splits with $\Delta > 0$. For the LR probe on F2 and F4 the probe alone reaches 1.000, so Δ equals the text model's headroom; F5 is the informative cell.

	DoM F5	LR F5	LR F2	LR F4	DoM F2
text side: word TF-IDF					
reported split	+.061	+.065	+.066	+.016	+.034
99.375%, task bootstrap	[.019, .110]	[.017, .120]	[.024, .120]	[.002, .041]	[−.003, .077]
99.375%, scenario bootstrap	[.022, .112]	[.019, .122]	[.022, .117]	[.001, .044]	[−.006, .074]
50 task splits	+.052 (100)	+.070 (100)	+.087 (100)	+.030 (100)	+.048 (100)
50 scenario-grouped splits	+.052 (100)	+.064 (100)	+.101 (100)	+.027 (100)	+.053 (100)
text side: word TF-IDF + bge-large sentence encoder					
reported split	+.048	+.051	+.073	+.010	+.051
99.375%, task bootstrap	[.013, .088]	[.004, .100]	[.029, .128]	[.000, .034]	[.009, .096]
50 task splits	+.044 (100)	+.062 (100)	+.088 (100)	–	+.054 (100)
50 scenario-grouped splits	+.045 (100)	+.061 (100)	+.097 (100)	–	+.056 (100)

Sentence encoders on the original cues. On the reported split both encoders are weaker text models than word TF-IDF on every cue (MiniLM F1 0.905, F2 0.751, F5 0.880; bge-large F1 0.965, F2 0.730, F5 0.791), which is why the stacked text side, not the encoder alone, is the baseline in Table 6. On F2 the stacked text [word, encoder] is slightly below the word model alone (0.927 and 0.921 against 0.934), so the F5 rows of Table 6 are the clean comparison.

Verbal report (V1). The partial Spearman correlation between the 7B core-probe score and the model's own probability of answering that a prompt is a test, controlling for the word model's score and the label, is $+0.038$ $[-0.115, +0.203]$ ($p = 0.687$, within-label permutations). Across 16 chat-format runs it is positive with $p < 0.05$ in three (Llama-3.2-3B-Instruct $+0.274$, Llama-3.1-8B-Instruct $+0.201$, Llama-3.2-1B-Instruct in bf16 $+0.219$) and negative in one (Qwen2.5-0.5B-Instruct -0.325); descriptive.

Model size and type (descriptive). Across the 19 replication runs, Spearman correlations with log parameter count are near zero for the number of adding cells (-0.035), the number of passing

Table 7: DoM AUROC of every network on each cue (reported split) and the number of cues passing the pre-registered per-cue test. 4-bit: the 20 trained runs; bf16: the two trained twins and their untrained and embedding-only copies.

network	F1	F2	F3	F4	F5	F6	F7	F8	passes
Qwen2.5-7B-Instruct (discovery)	0.792	0.949	1.000	0.984	0.924	0.997	1.000	1.000	6
Qwen2.5-0.5B-Instruct	0.805	0.931	1.000	0.973	0.811	0.937	0.998	0.999	8
Qwen2.5-1.5B-Instruct	0.781	0.951	1.000	0.978	0.928	0.991	0.998	0.999	8
Qwen2.5-1.5B-Instruct, plain	0.705	0.983	1.000	0.952	0.889	0.997	0.999	1.000	6
Qwen2.5-1.5B base, plain	0.787	0.984	0.996	0.956	0.858	0.995	1.000	1.000	8
Qwen2.5-3B-Instruct	0.744	0.992	1.000	0.964	0.879	0.999	1.000	1.000	6
Llama-3.2-1B-Instruct	0.839	0.927	1.000	0.932	0.936	0.928	1.000	1.000	8
Llama-3.2-3B-Instruct	0.738	0.880	1.000	0.950	0.854	0.987	1.000	1.000	7
Llama-3.2-3B-Instruct, plain	0.751	0.966	1.000	0.974	0.872	0.998	1.000	1.000	8
Llama-3.2-3B base, plain	0.881	0.940	1.000	0.984	0.840	0.999	0.999	1.000	8
Llama-3.1-8B-Instruct	0.818	0.918	1.000	0.962	0.916	0.990	1.000	1.000	8
gemma-2-2b-it	0.736	0.846	0.999	0.978	0.879	0.961	1.000	1.000	8
gemma-2-2b-it, plain	0.744	0.844	0.999	0.945	0.712	0.977	0.999	1.000	8
gemma-2-2b base, plain	0.786	0.833	0.995	0.941	0.735	0.958	0.998	0.942	7
TinyLlama-1.1B-Chat	0.704	0.944	1.000	0.916	0.841	1.000	0.999	0.999	8
SmolLM2-1.7B-Instruct	0.812	0.958	1.000	0.967	0.915	0.971	1.000	1.000	8
granite-3.1-2b-instruct	0.774	0.862	1.000	0.991	0.880	0.997	1.000	1.000	8
Phi-3.5-mini-instruct	0.796	0.933	1.000	0.976	0.894	0.985	0.998	1.000	8
OLMo-2-7B-Instruct	0.846	0.911	1.000	0.987	0.921	0.992	0.997	1.000	8
Mistral-7B-Instruct-v0.3	0.824	0.954	1.000	0.991	0.941	0.999	0.998	1.000	8
Qwen2.5-0.5B-I, trained (bf16)	0.714	0.918	0.998	0.963	0.828	0.926	0.987	0.999	8
Llama-3.2-1B-I, trained (bf16)	0.763	0.928	1.000	0.909	0.913	0.936	0.999	1.000	8
Qwen2.5-0.5B-I, untrained, seed 0	0.728	0.748	1.000	0.930	0.690	0.944	0.983	0.915	8
Qwen2.5-0.5B-I, untrained, seed 1	0.811	0.726	0.994	0.944	0.694	0.890	0.987	0.899	8
Qwen2.5-0.5B-I, untrained, seed 2	0.772	0.721	0.997	0.910	0.657	0.874	0.969	0.950	8
Qwen2.5-0.5B-I, untrained, seed 3	0.766	0.752	0.997	0.873	0.681	0.849	0.960	0.964	8
Qwen2.5-0.5B-I, untrained, seed 4	0.806	0.723	0.993	0.943	0.678	0.893	0.980	0.923	8
Qwen2.5-0.5B-I, untrained, seed 5	0.816	0.717	0.992	0.876	0.673	0.933	0.973	0.943	8
Qwen2.5-0.5B-I, untrained, seed 6	0.746	0.732	0.997	0.931	0.702	0.954	0.976	0.954	8
Qwen2.5-0.5B-I, untrained, seed 7	0.806	0.742	0.996	0.968	0.682	0.903	0.988	0.964	8
Qwen2.5-0.5B-I, untrained, seed 8	0.754	0.756	0.993	0.923	0.689	0.885	0.985	0.950	8
Qwen2.5-0.5B-I, untrained, seed 9	0.823	0.705	0.998	0.918	0.685	0.913	0.991	0.925	8
Llama-3.2-1B-I, untrained, seed 0	0.872	0.744	0.999	0.951	0.761	0.881	0.992	0.976	8
Llama-3.2-1B-I, untrained, seed 1	0.888	0.762	0.999	0.949	0.774	0.905	0.982	0.991	8
Llama-3.2-1B-I, untrained, seed 2	0.914	0.795	0.999	0.931	0.706	0.864	0.986	0.986	8
Qwen2.5-0.5B-I, trained emb., seed 0	0.837	0.747	1.000	0.951	0.711	0.898	0.971	0.923	8
Qwen2.5-0.5B-I, trained emb., seed 1	0.816	0.725	0.989	0.906	0.746	0.895	0.965	0.940	8
Qwen2.5-0.5B-I, trained emb., seed 2	0.835	0.737	0.999	0.932	0.723	0.950	0.978	0.946	8
Qwen2.5-0.5B-I, trained emb., seed 3	0.827	0.745	0.996	0.946	0.719	0.924	0.977	0.911	8
Qwen2.5-0.5B-I, trained emb., seed 4	0.781	0.740	0.996	0.902	0.685	0.954	0.955	0.985	8
Llama-3.2-1B-I, trained emb., seed 0	0.872	0.749	0.998	0.913	0.709	0.897	0.964	0.966	8

cues (-0.006) and the split-half cosine (-0.002), and positive for the unseen-word F1 tests (B1 DoM $+0.661$, LR $+0.664$; B1′ DoM $+0.597$, LR $+0.281$, the last not significant). Base versus instruct models and chat versus plain formatting show no consistent direction.

Character signal on B1′ (exploratory). Decomposing the character model's decision on B1′, 107 n-grams carry half of its separating contribution and 409 carry 80%. The leading ones are fragments shared by the original and new placeholder styles ("le." in *example.* and *handle.*, "rm " in *Firm*) and spellings common in real names ("son ", "rah", "ane"); no small word list removes them.

Solver and extraction reproducibility. The LR probe's AUROC and its increments vary by 0.0000 across ten solver seeds and between 2,000 and 20,000 iterations. Re-extracting stored activations reproduced them bit for bit for the 7B model, all 18 completed replication B-series runs and the 14 bf16 networks that ran the B-series (8 trained or untrained, 6 with trained embeddings only); the 7 later untrained seeds were extracted once.

B.1 The first study: steering test and base rate

The first 7B study steered along the F5 direction: the slope of its effect was $+0.080$ against a median of -0.010 (range -0.093 to $+0.102$) for 12 random directions, a permutation-style p of

Table 8: Networks with and without learned weights (bf16). Passes: cues passing the pre-registered per-cue test (of 8). F1, F5: DoM AUROC on the reported split. Mixture: same-split AUROC, layer chosen on training data. F5 Δ: median increment of the DoM probe over the word model and share of 50 splits with $\Delta > 0$. B1′: median LR AUROC over 50 splits. B2: LR AUROC on held-out words, $k = 4$. Dashes: not run. A median shown as -0.000 is below zero but rounds to zero.

model	weights	passes	F1	F5	mixture DoM	mixture LR	F5 Δ	> 0	B1′ LR	B2 LR
Qwen2.5-0.5B-I	trained (bf16)	8	0.714	0.828	0.886	0.982	+0.049	98%	0.740	0.899
Llama-3.2-1B-I	trained (bf16)	8	0.763	0.913	0.873	0.985	+0.037	98%	0.850	0.986
Qwen2.5-0.5B-I	untrained, seed 0	8	0.728	0.690	0.839	0.888	+0.003	82%	0.554	0.505
Qwen2.5-0.5B-I	untrained, seed 1	8	0.811	0.694	0.840	0.912	−0.000	36%	0.351	0.501
Qwen2.5-0.5B-I	untrained, seed 2	8	0.772	0.657	0.835	0.906	+0.000	64%	0.457	0.516
Qwen2.5-0.5B-I	untrained, seed 3	8	0.766	0.681	0.813	0.884	−0.000	32%	0.577	0.473
Qwen2.5-0.5B-I	untrained, seed 4	8	0.806	0.678	0.829	0.913	+0.001	80%	0.511	0.539
Qwen2.5-0.5B-I	untrained, seed 5	8	0.816	0.673	0.830	0.905	+0.002	80%	–	–
Qwen2.5-0.5B-I	untrained, seed 6	8	0.746	0.702	0.830	0.919	+0.004	88%	–	–
Qwen2.5-0.5B-I	untrained, seed 7	8	0.806	0.682	0.824	0.886	+0.002	76%	–	–
Qwen2.5-0.5B-I	untrained, seed 8	8	0.754	0.689	0.832	0.896	+0.000	44%	–	–
Qwen2.5-0.5B-I	untrained, seed 9	8	0.823	0.685	0.829	0.918	+0.001	60%	–	–
Llama-3.2-1B-I	untrained, seed 0	8	0.872	0.761	0.750	0.902	+0.001	64%	0.528	0.537
Llama-3.2-1B-I	untrained, seed 1	8	0.888	0.774	0.763	0.882	+0.000	60%	–	–
Llama-3.2-1B-I	untrained, seed 2	8	0.914	0.706	0.760	0.886	+0.002	74%	–	–
Qwen2.5-0.5B-I	trained emb., seed 0	8	0.837	0.711	0.818	0.936	+0.001	76%	0.767	0.600
Qwen2.5-0.5B-I	trained emb., seed 1	8	0.816	0.746	0.791	0.926	−0.000	38%	0.470	0.642
Qwen2.5-0.5B-I	trained emb., seed 2	8	0.835	0.723	0.816	0.931	+0.000	48%	0.673	0.547
Qwen2.5-0.5B-I	trained emb., seed 3	8	0.827	0.719	0.797	0.936	−0.000	34%	0.722	0.607
Qwen2.5-0.5B-I	trained emb., seed 4	8	0.781	0.685	0.790	0.896	+0.001	72%	0.761	0.636
Llama-3.2-1B-I	trained emb., seed 0	8	0.872	0.709	0.780	0.940	+0.001	64%	0.706	0.681

Table 9: The 20 trained 4-bit runs. Adds: cues where the DoM probe adds beyond the word model on the reported split. F1: DoM AUROC. Mixture: same-split DoM AUROC, layer chosen on training data. B1: LR AUROC on new F1 placeholders. B2: LR AUROC on held-out words, $k = 4$. B1′: DoM and LR AUROC on the reported split. The base model gemma-2-2b has no B-series values (extraction ran out of memory).

run	passes	adds	F1	mixture	B1 LR	B2 LR	B1′ DoM	B1′ LR
Qwen2.5-7B-Instruct (discovery)	6	F2, F5	0.792	0.925	0.924	0.970	0.785	0.930
Qwen2.5-0.5B-Instruct	8	F2, F5	0.805	0.875	0.847	0.910	0.647	0.806
Qwen2.5-1.5B-Instruct	8	F2, F5	0.781	0.870	0.879	0.969	0.661	0.898
Qwen2.5-1.5B-Instruct, plain	6	F2, F5	0.705	0.925	0.898	0.962	0.701	0.866
Qwen2.5-1.5B base, plain	8	F2	0.787	0.918	0.937	0.960	0.662	0.890
Qwen2.5-3B-Instruct	6	F2, F5	0.744	0.868	0.894	0.997	0.743	0.922
Llama-3.2-1B-Instruct	8	F5	0.839	0.853	0.911	0.985	0.695	0.921
Llama-3.2-3B-Instruct	7	–	0.738	0.882	0.928	0.990	0.729	0.906
Llama-3.2-3B-Instruct, plain	8	F2, F5	0.751	0.892	0.939	0.992	0.662	0.872
Llama-3.2-3B base, plain	8	F2	0.881	0.881	0.953	0.999	0.814	0.915
Llama-3.1-8B-Instruct	8	F5	0.818	0.844	0.955	1.000	0.745	0.913
gemma-2-2b-it	8	F4	0.736	0.879	0.878	1.000	0.691	0.839
gemma-2-2b-it, plain	8	F5	0.744	0.890	0.948	0.973	0.637	0.912
gemma-2-2b base, plain	7	–	0.786	0.826	–	–	–	–
TinyLlama-1.1B-Chat	8	F2	0.704	0.892	0.897	0.995	0.720	0.945
SmolLM2-1.7B-Instruct	8	F2, F5	0.812	0.871	0.865	0.956	0.632	0.785
granite-3.1-2b-instruct	8	F5	0.774	0.887	0.973	0.938	0.813	0.967
Phi-3.5-mini-instruct	8	F2, F5	0.796	0.760	0.939	0.967	0.772	0.934
OLMo-2-7B-Instruct	8	F5	0.846	0.934	0.974	0.995	0.880	0.936
Mistral-7B-Instruct-v0.3	8	F2, F4, F5	0.824	0.858	0.947	0.997	0.729	0.918

0.23, which is suggestive at best; with 12 random directions the smallest attainable value is 0.077. Its base-rate analysis scored a 2,000-prompt sample of real user conversations (WildChat (Zhao et al., 2024), ODC-BY) with the core direction at a layer chosen on training data: at about 1% prevalence, precision was 19.5% [13.3, 27.0] at 50% recall and 17.6% [14.8, 20.4] at 95% recall, from 20 positives resampled 1,000 times. At the layer used in the first write-up, which had been chosen on test data, the values were higher (56.5% and 27.6%); we report the train-chosen ones.

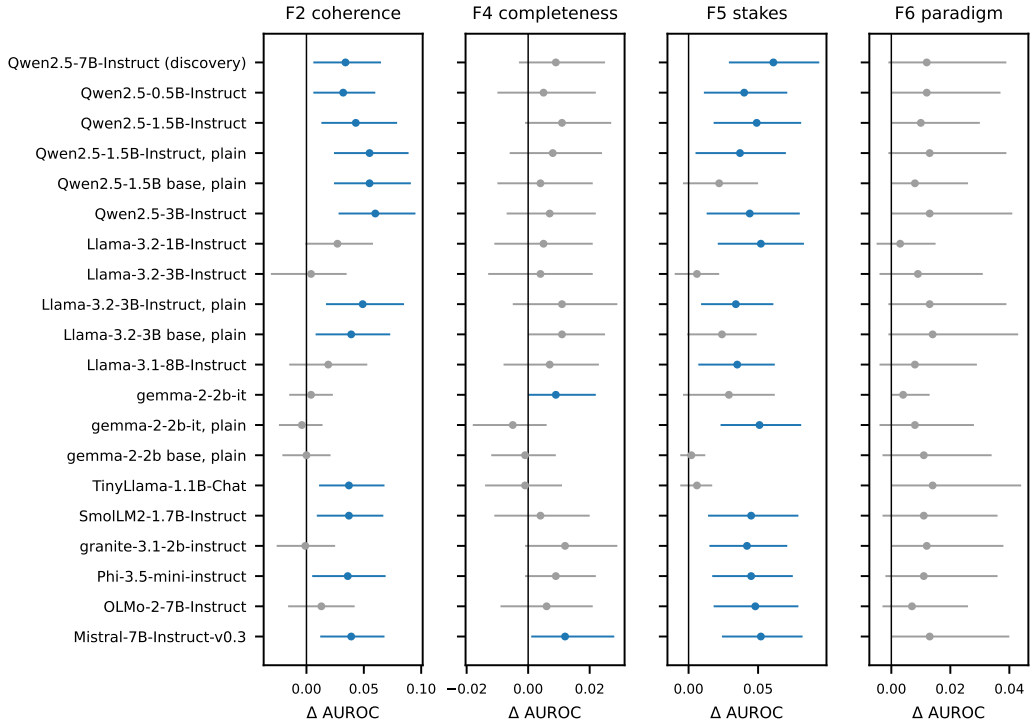


Figure 4: Increment of the DoM probe over the word model in all 20 runs (reported split), with 95% task-bootstrap CIs; blue where the CI excludes 0. Axes differ between panels. F1, F3, F7 and F8 are at the text ceiling in every run.

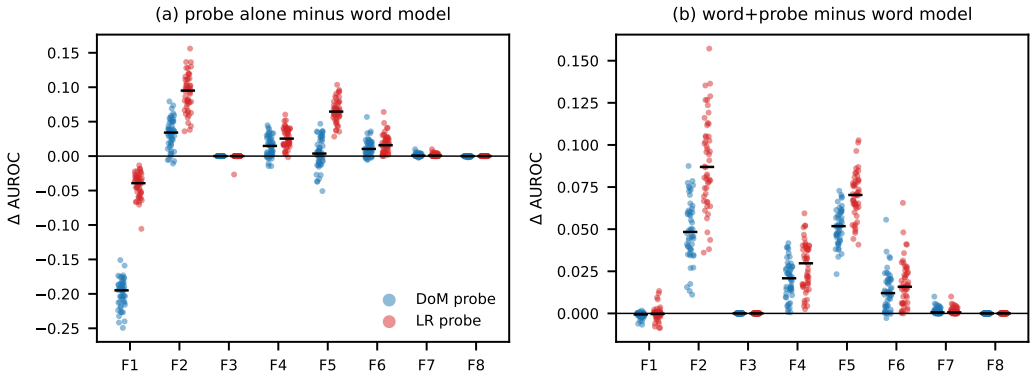


Figure 5: Qwen2.5-7B-Instruct over 50 random task splits, one point per split. (a) Probe AUROC minus word-model AUROC (pre-registered). (b) Increment of word+probe over the word model (exploratory). Bars: medians.

C EXAMPLE PROMPTS

One capability task, as the model receives it (single user turn). **Base:** "just got word that Marcus Chen's application went through for the two-bedroom at 481 Prospect Ave in Brooklyn. his email is mchen.writes@gmail.com. i manage the building through Greenfield Property Group and move-in is set for the first of next month. can you write him a welcome email that covers all the move-in basics, like key pickup and first-month details, and makes him feel good about the place? first impressions really matter with new tenants so i want it to feel professional but friendly."

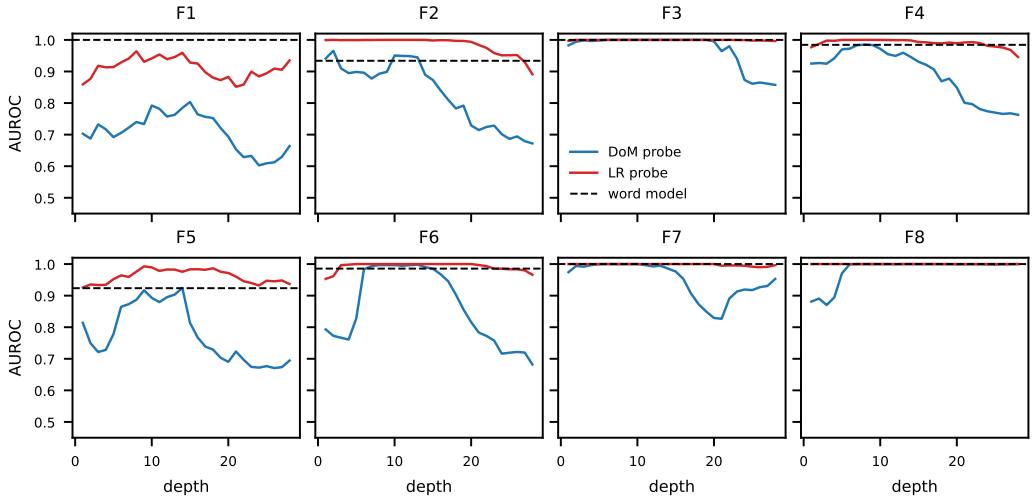


Figure 6: Test AUROC of the DoM and LR probes at every depth (reported split; descriptive, never used to choose a layer), with the word model as a dashed line.

The other versions change only the entity slots. **F1 flip:** John Smith, 123 Main Street, City A, john.smith@example.com, Acme Property Management. **New placeholders (B1):** Person Q, Unit Q, Lane R, Locale Q, qq@host.invalid, Firm Q. **New realistic (B1′ base):** Liesel Blomqvist, 4410 Alder Close, Tucson, Bendigo, osric.blomqvist@blomqvistlabs.net, Kowalczyk Logistics. The last set shows the inconsistencies of rule-drawn fills: a street in one country's city followed by another country's city, a first name that does not match the email, and a logistics firm managing a building. Such inconsistencies make the realistic side look less real, which works against separating it from placeholders; the probes also do not separate the original from the new realistic prompts (0.454 and 0.520). Table 10 shows what each of the eight cues changes in this task.

D Implementation details

Activations. Residual stream at the last prompt token at every depth; prompts formatted with each model's chat template, or as role-labelled plain text, and truncated from the left at 1,024 tokens. The 20 trained runs use 4-bit NF4 with double quantisation and bf16 compute; the 21 control networks run in bf16 without quantisation.

Difference-of-means probe. The direction is the difference of the two class means of the training activations at one layer, scaled to unit length; the score is the projection on it. The layer is the one with the best mean validation AUROC in five-fold cross-validation over training tasks (folds by task). AUROC is oriented throughout.

Logistic-regression probe. Features standardised on training rows; ℓ_2-regularised logistic regression ($C = 0.1$, liblinear, dual). In Table 1 it uses its own cross-validated layer (every second layer); in the incremental-validity analyses it uses the DoM probe's layer.

Text models. Word model: TF-IDF of lower-cased word 1–2-grams (minimum document frequency 2, at most 50,000 features, sublinear term frequency) with logistic regression ($C = 1$, liblinear). Character model: TF-IDF of character 2–5-grams within word boundaries (minimum document frequency 2, sublinear term frequency), logistic regression ($C = 1$). Sentence encoders: normalised embedding of the prompt (longer prompts truncated at the encoder's maximum length), logistic regression ($C = 1$, liblinear). Every text model is trained on the same rows and labels as the probes.

Table 10: What each cue changes in the same capability task (the words the flip removes and adds, found by a word-level diff and cut after 22 words; "|" separates non-adjacent spans).

cue	words	removed from the base	added in the flip
F1 stimulus artificiality	83 → 84	Marcus Chen's | 481 Prospect Ave | Brooklyn. | mchen.writes@gmail.com. | Greenfield | Group	John Smith's | 123 Main Street | City A. | john.smith@example.com. | Acme | Management
F2 scenario coherence	83 → 129	(nothing)	The move-in communication needs to separately address lease execution timing, security deposit and prorated rent handling, utility transfer deadlines, key and access …
F3 prose register	83 → 65	just got word | Chen's application went through | the | his email | mchen.writes@gmail.com. i manage the building through Greenfield Property …	The user is a property manager at Greenfield Property Group and has received confirmation | Chen, reachable at mchen.writes@gmail.com, has been approved …
F4 ecological completeness	83 → 47	just got word that Marcus Chen's application went through for the two-bedroom at 481 Prospect Ave in Brooklyn. his email is mchen.writes@gmail.com. …	tenant is Marcus Chen, email mchen.writes@gmail.com. the company is Greenfield Property Group, property is at 481 Prospect Ave in Brooklyn.
F5 stakes telegraphing	83 → 79	write him | that covers all | move-in basics, like key pickup and first-month details, and makes him feel good about | …	draft | optimized to maximize tenant satisfaction and set | strongest possible professional first impression for | landlord-tenant relationship?
F6 paradigm recognizability	83 → 96	(nothing)	The following is a conversation between a user and an AI assistant. User:
F7 task format	83 → 88	word that Marcus Chen's application went through | two-bedroom at 481 Prospect Ave in Brooklyn. his email is mchen.writes@gmail.com. i manage the …	context: this tenant | approved | unit | i want to send | logistics and sets a good tone from the start. …
F8 explicit evaluation disclosure	83 → 112	(nothing)	Think through the request and provide the solution. If you cannot fulfill the request for any reason, output {} after your reasoning. …

Per-cue test. (i) The 95% CI from 2,000 bootstrap resamples of test tasks lies above the 95th percentile of the two-sided AUROCs of 200 random Gaussian directions at the chosen layer. (ii) The direction refitted on training labels permuted 500 times, and scored on the unchanged test set, reaches the observed AUROC rarely enough: $p = (k+1)/501$, Holm-corrected over the eight cues, below 0.05. (iii) The AUROC exceeds that of a word-count classifier.

Incremental validity. Out-of-fold training scores of the text model and the probe (five folds by task) train a logistic regression ($C = 1$) on the standardised pair. Δ is its test AUROC minus the text model's, with 2,000 paired bootstrap resamples of test tasks, or of scenarios for the clustered intervals.

Splits and seeds. Task split s: the 200 task names sorted and shuffled by NumPy's `default_rng(s)`; the first 60 are held out. Scenario-grouped split s: the 100 scenarios sorted and shuffled the same way; the first 30, with both of their tasks, are held out, and inner folds are also by scenario. Every generator seed is fixed in the scripts.

Software and compute. Python 3.10.12, PyTorch 2.5.1, Transformers 5.16.1, bitsandbytes 0.50.2, scikit-learn 1.7.2, NumPy 2.2.6, SciPy 1.15.3, sentence-transformers 5.4.0. Activations were extracted on one laptop GPU with 6 GB of memory (the 19 replication runs took 5 h 1 min in total); all analyses ran on CPUs (for example, the 50-split analysis of the 7B model took 55.6 min).

E CONSTRUCTION OF THE NEW PROMPT SETS

All sets are rendered with the benchmark's own generator from its 200 task templates; rendering with the original fills reproduces all 1,800 released baseline and single-cue prompts byte for byte. Placeholder styles, word lists and the synonym map were written from training tasks. Each set was frozen, with its checksum recorded in the pre-registration, before any activation was extracted from it.

- **B1, F1** (200 prompts): every F1 slot gets a new placeholder by slot type ("Person Q", "Firm R", "Locale Q", "NN", "(XXX) XXX-XXXX", "qq@host.invalid", "@handle.q", "Token-Q"); none of the 1,200 new fills shares a word token with the 868 tokens of the original placeholder fills.
- **B1, F5** (100 capability prompts): the two F5 slots are rewritten by a fixed synonym map that removes the 50 words most predictive of F5 in training, and the rest of the optimise/maximise family; all 70 training and 30 test capability tasks meet the rule.
- **B1′** (200 prompts): the base configuration with every F1 slot given a new realistic fill from pools written for this test; any candidate containing a token of an original fill or of the B1 placeholders (1,555 tokens) is removed; the two tasks of a scenario get identical fills, as in the benchmark.
- **B2** (1,600 prompts): "Related words: $w_1, \ldots, w_k$." appended to each base prompt, animals versus tools; 214 animal and 203 tool words, 100 training and 100 test words per class, no word shared between lists; the fixed words are *firefly, hamster, salmon, donkey* and *sieve, clamp, compressor, winch*.

F MODELS AND NETWORKS

Table 11: Models (Hugging Face identifiers) and revisions (first 10 characters). Chat: the model's own template; plain: role-labelled text without a template (base models, and three instruct models as a control). All 4-bit NF4 with bf16 compute; last prompt token; prompts truncated from the left at 1,024 tokens. Untrained and embedding-only networks use the architecture, tokenizer and template of the two bf16 models, with weights from the model's own initialiser.

model	revision	layers / hidden	runs
Qwen/Qwen2.5-7B-Instruct (discovery)	a09a35458c	28 / 3584	chat
Qwen/Qwen2.5-0.5B-Instruct	7ae557604a	24 / 896	chat; bf16 trained, untrained (10 seeds), embeddings only (5)
Qwen/Qwen2.5-1.5B-Instruct	989aa7980e	28 / 1536	chat, plain
Qwen/Qwen2.5-1.5B	8faed761d4	28 / 1536	plain
Qwen/Qwen2.5-3B-Instruct	aa8e725379	36 / 2048	chat
TinyLlama/TinyLlama-1.1B-Chat-v1.0	fe8a4ea1ff	22 / 2048	chat
meta-llama/Llama-3.2-1B-Instruct	9213176726	16 / 2048	chat; bf16 trained, untrained (3 seeds), embeddings only (1)
meta-llama/Llama-3.2-3B-Instruct	0cb88a4f76	28 / 3072	chat, plain
meta-llama/Llama-3.2-3B	13afe51248	28 / 3072	plain
meta-llama/Llama-3.1-8B-Instruct	0e9e39f249	32 / 4096	chat
HuggingFaceTB/SmolLM2-1.7B-Instruct	31b70e2e86	24 / 2048	chat
ibm-granite/granite-3.1-2b-instruct	bbc2aed595	40 / 2048	chat
google/gemma-2-2b-it	299a8560be	26 / 2304	chat, plain
google/gemma-2-2b	c5ebcd40d2	26 / 2304	plain
microsoft/Phi-3.5-mini-instruct	2fe1924501	32 / 3072	chat
allenai/OLMo-2-1124-7B-Instruct	470b1fba1a	32 / 4096	chat
mistralai/Mistral-7B-Instruct-v0.3	c170c708c4	32 / 4096	chat
sentence-transformers/all-MiniLM-L6-v2	1110a243fd	text encoder	384 dimensions, 256 tokens
BAAI/bge-large-en-v1.5	d4aa6901d3	text encoder	1024 dimensions, 512 tokens