MODEL INTERNALS SPEAK VOLUMES: J-LENS LOUDNESS PREDICTS DECODABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

When probing the activations of a Large Language Model (LLM), several design decisions arise. Besides the probe design itself, one must choose which layer(s) and token position(s) to probe. These choices are often made heuristically or through expensive sweeps over candidate layers and positions, and to our knowledge, no method yet predicts the decodability of a signal directly from the activations. We take a step in this direction by introducing J-Lens loudness, a new application of the Jacobian lens (J-Lens) that guides these choices without labels or probe training. We represent a signal of interest by a small subset of the model's vocabulary, and define J-Lens loudness as the log probability mass that the J-Lens assigns to this subset at a given activation. In the reasoning chains of two LLM agents navigating grid worlds, the balanced accuracy of next-action probes increases steadily with direction signal loudness. Taken together, we show that J-Lens loudness can be used as a criterion to find layer–token positions whose activations contain underlying computation relevant to a signal of interest.

1 INTRODUCTION

In a time of rapid improvements in the performance of Large Language Models (LLMs), understanding the computations behind their behaviour has become increasingly important for the field of interpretability (Sharkey et al., 2025).

One of the main methods used across the field is probing (Alain & Bengio, 2016). Researchers employing this method typically aim to test whether a signal, that is, information about some aspect of the input or the model's computation, such as a feature, or a property, is decodable from a model's activations at a specific layer and token position. Typically, one trains a mapping between those activations and labels for the signal of interest, ranging from a simple linear classifier to a more expressive model such as a Multilayer Perceptron (MLP).

A major practical difficulty of probing is that, before a probe can be trained, one must already know where the relevant signal is decodable, and a poor guess can make a signal look absent when it is not. The first choice is the layer. Conventional wisdom tells us that middle layers are typically where information is richest. Indeed, intermediate layers often yield more useful representations than final ones (Skean et al., 2025), and probe performance frequently peaks there (Lugoloobi et al., 2026; Gurnee & Tegmark, 2024; Marks & Tegmark, 2024), but where exactly the peak lies varies across models and signals. The second choice is which token(s) to probe, and it is just as consequential. Information is distributed across tokens non-uniformly, and some signals concentrate at specific positions, such as exact-answer tokens for truthfulness (Orgad et al., 2025) or punctuation tokens for sentiment (Tigges et al., 2023). Probe performance, therefore, depends heavily on the choice of token position, which is rarely known in advance.

These are important decisions which shape how we think about the inner workings of LLMs and are used in monitoring models, but they are often based on results which are not necessarily transferable between problems. The most common approach is to sweep over layers L (and sometimes positions P), training a separate probe for every combination and keeping the best one. This is costly, requiring $O(L \times P)$ probes and labelled data for every signal. The alternative is to rely on heuristics such as just picking the middle layer, which is cheap but not signal-specific.

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.

This calls for a new family of measurements of decodability which would tell us the degree to which a signal is encoded in a specific activation, without relying on probes. Luckily, such readouts of intermediate activations already exist in the form of Logit Lens (nostalgebraist, 2020) and Jacobian Lens (J-Lens) (Gurnee et al., 2026), but to our knowledge, neither has yet been examined on their capabilities as signal-specific predictors of decodability across layers and token positions.

We introduce *J-Lens loudness* to fill this gap. Our method relies primarily on the Jacobian Lens, which identifies the tokens a model is poised to verbalize at a given point in its processing (Gurnee et al., 2026). For a signal s with an associated vocabulary subset V_s, we define J-Lens loudness as the log probability mass the J-Lens assigns to V_s at the activation of a given layer and token position.

We compute loudness over multiple layer–position pairs in the reasoning chains of GPT-OSS-20B (OpenAI, 2025) and Qwen3.6-35B-A3B (Qwen Team, 2026) agents navigating 2D grid worlds (Arghal et al., 2026). We mainly study a signal related to direction information, which we relate to computations about the next step. We evaluate a second signal related to the grid state which serves as a control. We choose the "loudest" layer for each signal and the "loudest" positions within each reasoning chain (Section 3), and test whether the signal is more decodable there.

We make the following claims:

- J-Lens loudness is a cheap criterion that requires no probe training or labels and can be used to find the layers and token positions from which a signal is decodable;
- Decodability increases steadily with J-Lens loudness, even at positions where no signal words are written;
- J-Lens loudness is selective: the decodability of a signal s is predicted by the loudness of s, but not by the loudness of an unrelated signal s';
- J-Lens loudness is more reliable than the Logit Lens and than end-of-sentence tokens for finding activations to probe.

2 Related Work

2.1 Probing and its limitations

We adopt the standard probing setup (Belinkov, 2022), in which probe accuracy is taken as evidence that a property is decodable from a model's representations. Expressive probes may learn the task themselves, which Hewitt & Liang (2019) address by comparing against control tasks, and there is no consensus on probe complexity: some advocate simple probes (Alain & Bengio, 2016; Hewitt & Liang, 2019), others the best-performing one (Pimentel et al., 2020). We therefore report both linear and MLP probes. Finally, high probe accuracy does not entail that the model uses the decoded information (Ravichander et al., 2021; Elazar et al., 2021), although probe-guided interventions can provide causal evidence (Giulianelli et al., 2018). We use probe accuracy only as a measure of decodability, which is what loudness should predict, and make no causal claims.

2.2 Choosing a layer

Layers are typically chosen by a heuristic or by an exhaustive sweep. The heuristic view is that middle layers are richest in information, as they sit between layers that process the input and layers that produce the output. Empirically, intermediate layers often yield better representations than final ones (Skean et al., 2025), and linear probes for spatial and temporal information improve over the early layers before plateauing around the middle of the network (Gurnee & Tegmark, 2024). That said, this is only a heuristic. To find definitively which activations work best for a given probing setup, a sweep is often performed. Sweeps train a probe for every layer, and often every position: Lugoloobi et al. (2026), for example, cross-validate a linear probe for every layer–position pair to select the layer, position, and regularisation strength for each target. Heuristics are cheap but not signal-specific, while sweeps are signal-specific but costly, especially on long horizon tasks or long reasoning chains.

2.3 Choosing token positions

Token-position heuristics fall into three families. The first uses a single fixed position, such as the final input token (Marks & Tegmark, 2024) or a structurally defined position such as the end of the instruction or of the reasoning (Arghal et al., 2026). The second uses positions tied to the signal, such as the last token of an entity name (Gurnee & Tegmark, 2024) or the exact-answer tokens (Orgad et al., 2025); relatedly, Tigges et al. (2023) find sentiment partly summarised at punctuation rather than content tokens. The third aggregates over positions, either with a fixed rule, such as averaging per-token probe scores over the response (Goldowsky-Dill et al., 2025), or adaptively, such as with attention-based or max-style probes (McKenzie et al., 2026; Kramár et al., 2026). Each heuristic encodes a prior about where a signal is decodable, and all are reliant on the results of narrow-use probes.

2.4 Label-free readouts of intermediate activations

A separate line of work reads intermediate activations directly in vocabulary space, without training on labelled data.

The Logit Lens (nostalgebraist, 2020) applies the model's unembedding matrix W_U directly to intermediate activations. It is informative in late layers but often uninterpretable in early ones.

The J-Lens (Gurnee et al., 2026) does not unembed an intermediate activation directly. It first estimates how a small change to the activation at layer l changes the model's final layer, using the Jacobian between the two layers averaged over many positions and inputs, $\bar{J}_l$. Multiplying an activation $a_{l,p}$ by $\bar{J}_l$ translates it into the final layer's space, where the unembedding applies:

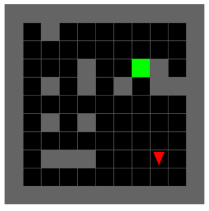
$$\mathrm{JLens}_l(a_{l,p}) = \mathrm{softmax}\big(W_U \bar{J}_l \, a_{l,p}\big). \tag{1}$$

It recovers the tokens a model is poised to verbalize at earlier depths than the Logit Lens. We fit our own J-Lens for each model (Appendix B).

These lenses have so far been used to inspect what an activation encodes; we use them to predict where a probe can decode a signal.

3 Methods

Environment and data. We study GPT-OSS-20B and Qwen3.6-35B-A3B in the grid-world environment studied by Arghal et al. (2026), without doors or keys. At each step, the model controls an agent `A` on an $n \times n$ grid containing walls `#`, empty cells `_` and a goal `G` (Figure 1); it produces a reasoning chain before producing a final answer indicating the action the agent will take (`UP`, `DOWN`, `LEFT` or `RIGHT`) in a separate output channel. Like Arghal et al. (2026), we use grid sizes $n \in \{5, 7, \ldots, 15\}$, complexity levels $c \in \{0.0, 0.2, \ldots, 1.0\}$, and treat the reasoning chain of each step as an independent example and split training and evaluation data by chain. Because Qwen3.6-35B-A3B produces much longer chains, we uniformly sample 20% of its reasoning-token positions before any other selection or analysis takes place.



```
# # # # # # # # # #
# _ # _ _ _ _ _ _ #
# _ _ _ _ _ _ _ _ #
# _ _ _ # _ _ G # _ #
# _ # _ # _ # _ # # #
# _ _ _ _ _ _ _ _ #
# _ # _ # _ _ _ _ #
# _ _ _ _ _ _ _ _ #
# _ # # # _ _ A _ #
# _ _ _ _ _ _ _ _ #
# # # # # # # # # #
```

Figure 1: An example grid (left) and its text representation in the prompt (right). Reproduced from Arghal et al. (2026, Figure 3).

Signals and vocabularies. An activation encodes information about many aspects of the model's input and computation at once. We call the information about one such aspect a *signal*. Equivalently, a signal can be defined as a set of related features; for direction, these include *left*, *right*, *up*, *down*, *horizontal*, and *vertical* to name a few examples. A signal is not tied to a particular set of labels: many probing tasks can target the same signal, each reading out different features with its own classes. For instance, one task may predict which of the four directions the agent moves next, and another only whether that move is horizontal or vertical. Yet another probe may be trained to predict the general direction the goal state may be in, the list of examples goes on. We say that a signal

is decodable for a task at (l, p) if a probe trained on $a_{l,p}$ predicts the task's labels above chance. Loudness, in contrast, depends only on the signal, the layer and the position. In this study, we focus on the movement *direction* signal d. We probe it with one task, the model's *local belief* at token position p: we truncate the reasoning chain at p, close the reasoning channel and open the final-answer channel. This forces the model to output an action, which we subsequently take as our local belief label. The *grid* signal g, corresponding to the layout of the grid, serves as a control. Probes associated with this signal are trained to predict the content of a queried cell. For each model and signal, we construct the vocabulary V_s from words that express the signal rather than any task's classes: a small set of anchor words which we expand in two stages: a lexical matching stage, and a semantic matching stage. This method is explained in depth in Appendix A, however, we do not intend our vocabulary building method to be presented as definitive. Indeed, it results in many false positives and we expect future work to surface methods much better at capturing a relevant signal of interest.

J-Lens loudness. We compute loudness with the J-Lens JLens_l that we fit for each model (Section 2; Appendix B). For a signal s with vocabulary V_s, J-Lens loudness is the log-probability mass assigned to V_s:

$$J_s(a_{l,p}) = \log \sum_{v \in V_s} \mathrm{JLens}_l(a_{l,p})_v. \tag{2}$$

Logit-Lens loudness is defined identically with $\bar{J}_l$ replaced by the identity, i.e. with the Logit Lens in place of the J-Lens. We select the loudest layer for each signal and, within each reasoning chain x, the N loudest positions at that layer:

$$l_s^* = \arg\max_l \ \mathbb{E}_{x \sim \mathcal{D}} \, \mathbb{E}_p\big[J_s(a_{l,p})\big], \qquad P^*_{s,N}(x) = \underset{p}{\text{top-N}} \ J_s(a_{l_s^*,p}). \tag{3}$$

Neither step uses probe labels or probe training.

Position selection. In addition to J-Lens loudness, we also compute Logit-Lens loudness at l_s^* with the same V_s, so the two loudness metrics differ only in the mapping to vocabulary space. We compare two selection schemes. *Top-N* keeps the N positions per chain with the highest J-Lens loudness, the highest Logit-Lens loudness, or chosen uniformly at random (N=20 for GPT-OSS-20B and N=60 for Qwen3.6-35B-A3B, reflecting chain length). *Sentence-level* selection keeps one position per sentence: the loudest under each lens, a random one, or the sentence-final punctuation token (EOS), a structural heuristic for where information is summarised (Tigges et al., 2023). We apply sentence-level selection to GPT-OSS-20B only, as the reasoning chains of Qwen3.6-35B-A3B are too long to process every sentence under our computational constraints.

Probes. We train linear and MLP probes on activations at l_s^*, one per model, signal, training criterion and architecture. The direction probe predicts one of the four actions. The grid probe receives the activation concatenated with the coordinates of the queried cell and predicts `wall`, `empty`, `goal` or `agent` (Appendix C.6). Training details are in Appendix B.

Evaluation. On held-out chains, we construct every selection subset independently and evaluate each probe on all of them. This cross-selection evaluation separates whether a criterion provides good training data from whether it finds activations where the signal is decodable. We report balanced accuracy throughout. To test whether decodability grows with loudness, we pool all held-out tokens, bin them into loudness deciles and compute accuracy per decile, with 95% bootstrap confidence intervals over chains (300 resamples).

It is possible that the loudest tokens are verbalized tokens of V_s, that is, tokens the model actually writes in its reasoning chain. For example, the loudest tokens for direction might be `.left`, `derecha` or →, all of which are in V_d. We would like loudness to find more than these trivial positions, so in an additional analysis (Appendix Figure 8) we remove every verbalized token of V_s, as well as the tokens within ± 2 positions of it, and recompute the deciles.

To test selectivity, we also test how the same direction probes perform along the grid loudness axis.

4 RESULTS

We first examine where loudness peaks across layers (Section 4.1). At the selected layer, we then test whether loud positions are more decodable than Logit-Lens-selected and random ones (Section 4.2), whether decodability increases with loudness (Section 4.3), and whether this relationship is specific to the signal (Section 4.4). Direction is our main signal; the grid signal serves as a control (Section 4.5). All scores are balanced accuracy; chance is 0.25.

4.1 LOUDNESS PEAKS IN AN INTERMEDIATE LAYER BAND

Averaged over tokens and reasoning chains, J-Lens loudness was high across a broad intermediate band and fell sharply in the final layers, for both models and both signals (Figure 2). It peaked at layer 15 (direction) and 14 (grid) for GPT-OSS-20B, and at layer 27 for both signals in Qwen3.6-35B-A3B. This band matches where probe accuracy typically peaks in prior work (Skean et al., 2025; Gurnee & Tegmark, 2024; Lugoloobi et al., 2026), so loudness yields a signal-specific version of the common middle-layer heuristic. Logit-Lens profiles were less informative and, for Qwen3.6-35B-A3B, flat across nearly all layers (Appendix Figure 5). We therefore rank positions at the J-Lens-selected layer for every criterion.

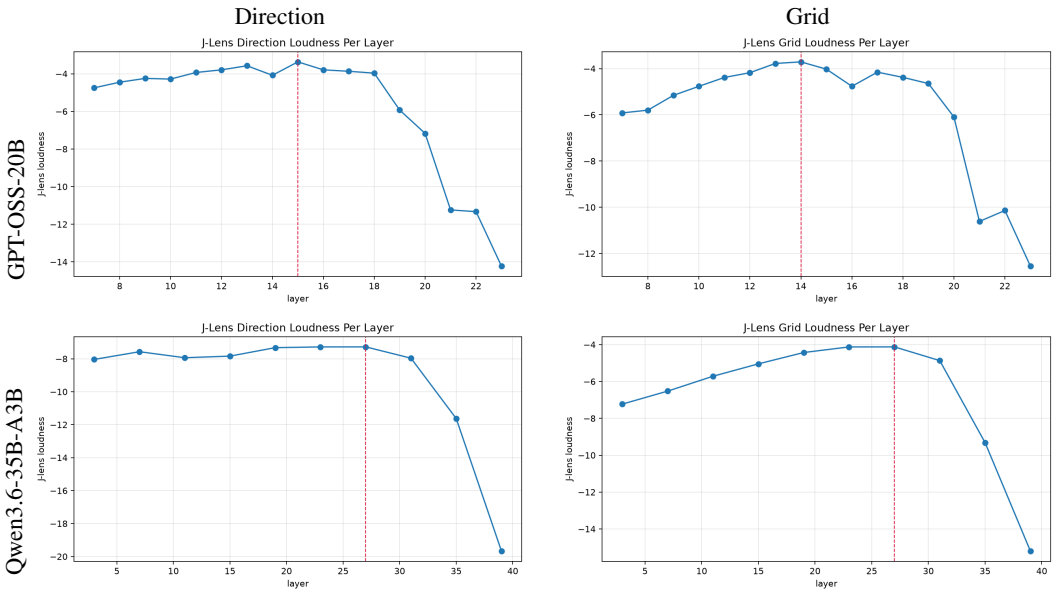


Figure 2: Mean J-Lens loudness per layer. Dashed line: selected layer l_s^*.

4.2 LOUD POSITIONS ARE MORE DECODABLE

Table 1: Direction, top-N selection. Rows: training criterion; columns: evaluation subset. **Bold**: best subset per probe; underlined: in-distribution.

		GPT-OSS-20B			Qwen3.6-35B-A3B		
Trained on	Probe	J-Lens	Logit Lens	Random	J-Lens	Logit Lens	Random
J-Lens	Linear	**.800**	.689	.609	**.503**	.430	.320
	MLP	**.860**	.745	.655	**.621**	.500	.346
Logit Lens	Linear	**.759**	.726	.620	**.476**	.450	.342
	MLP	**.818**	.790	.676	**.598**	.547	.372
Random	Linear	**.751**	.698	.655	**.432**	.421	.347
	MLP	**.818**	.765	.721	**.528**	.469	.396

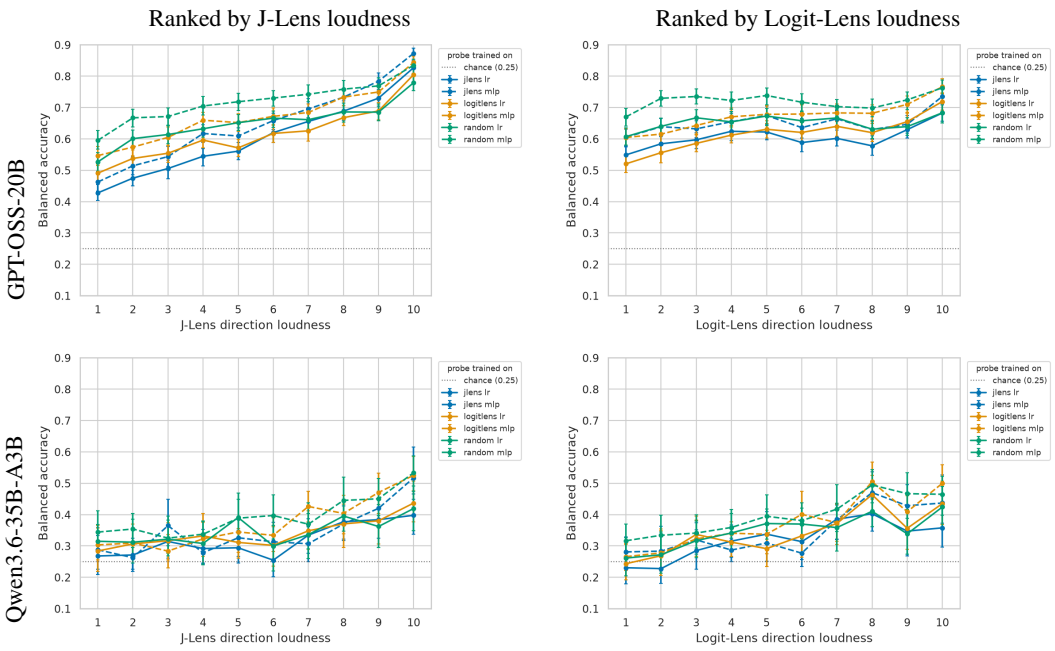


Figure 3: Direction probes by direction-loudness decile at l_d^* (1: quietest, 10: loudest). Line label: training criterion; solid: linear, dashed: MLP. Error bars: 95% bootstrap CIs.

Table 1 evaluates every probe on held-out positions selected by each criterion, with 14,391 (GPT-OSS-20B) and 3,120 (Qwen3.6-35B-A3B) positions per subset. In both models, every probe was most accurate on J-Lens-selected positions, followed by Logit-Lens-selected and then random positions. This ordering also holds for probes trained on random positions, whose training data is independent of loudness, so it reflects a property of the selected activations rather than of the probe. Training on loud positions gave the highest accuracy on loud positions but generalised less well to random ones than training on random positions. Loudness is therefore most useful for deciding *where to apply* a probe.

Comparison with a fixed-position heuristic. Selecting one position per sentence allows a comparison with end-of-sentence punctuation (EOS), a position at which information is thought to be summarised (Tigges et al., 2023) (Table 2; GPT-OSS-20B). The top-N pattern holds at this granularity: every probe not trained on EOS positions was more accurate on J-Lens-selected than on EOS positions, and J-Lens probes evaluated in-distribution outperformed EOS probes evaluated in-distribution. EOS probes were accurate only on EOS positions and transferred poorly elsewhere.

Table 2: GPT-OSS-20B, direction, sentence-level selection: J-Lens/Logit Lens loudest, random uniform, or punctuation token (EOS). **Bold**: best subset per probe; underlined: in-distribution.

Trained on	Probe	J-Lens	Logit Lens	Random	EOS
J-Lens	Linear	**.606**	.531	.460	.460
	MLP	**.677**	.605	.513	.542
Logit Lens	Linear	.565	**.571**	.474	.463
	MLP	**.642**	.639	.535	.542
Random	Linear	**.570**	.551	.509	.518
	MLP	**.648**	.618	.584	.606
EOS	Linear	.489	.446	.413	**.580**
	MLP	.530	.481	.445	**.657**

4.3 Decodability increases with loudness

Binning all held-out tokens by loudness, with each line a probe labelled by its training criterion, shows that the top-N result is part of a graded relationship (Figure 3). For GPT-OSS-20B, accuracy increased almost monotonically with J-Lens loudness for every probe, whereas Logit-Lens loudness ordered tokens poorly outside the loudest decile. For Qwen3.6-35B-A3B, which has fewer held-out chains and wider intervals, accuracy rose over the loudest deciles under both lenses.

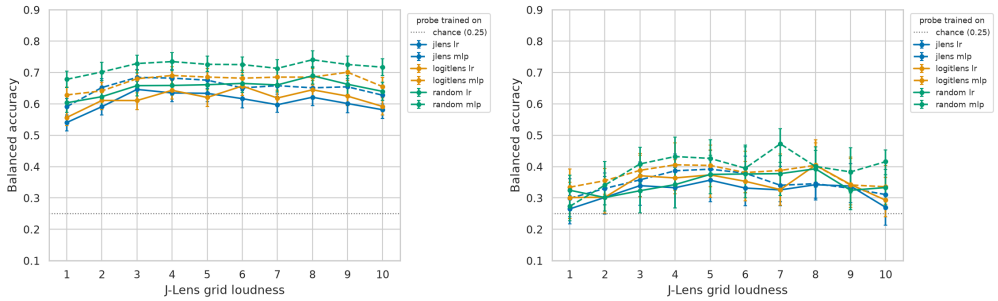


Figure 4: Direction probes by J-Lens *grid*-loudness decile. Left: GPT-OSS-20B; right: Qwen3.6-35B-A3B.

A possible confound is that loud tokens are simply direction words (e.g. `left`), which a probe could decode trivially. After removing every token of V_d together with the two tokens on either side from the list of selected tokens, the J-Lens trend persisted in both models, whereas the Logit-Lens trend for GPT-OSS-20B disappeared (Appendix Figure 8). J-Lens loudness therefore tracks decodability away from the trivial positions where direction words are written, and can identify positions where the signal is decodable beyond its surface form.

An additional look into which tokens are selected can be found in Appendix Section C.7, where we find the models' direction-related computations seem to also take place during the planning of future grid states, or when recalling the agent's or goal's grid locations.

4.4 Loudness is signal-specific

If loudness merely marked generically informative positions, the loudness of an unrelated signal would predict direction decodability equally well. It does not. Direction and grid J-Lens loudness were nearly uncorrelated across tokens (Spearman $\rho \approx 0.04$ in both models; Appendix Figure 7), and binning the direction probes of Figure 3 by grid loudness instead produced no consistent trend (Figure 4).

4.5 The grid signal as a control

We also trained probes to decode the grid state directly. These probes performed poorly on every position subset: linear probes stayed near chance and MLP probes only modestly above it. With so little decodable signal, differences between selection criteria were small and inconsistent across models and lenses (Appendix C.6). We therefore treat the grid experiment as inconclusive about grid decodability and use grid loudness only as a control signal. We discuss the design choices behind this outcome in Section 5.

5 Discussion

Our results show that J-Lens loudness, computed without labels or probe training, predicts where the model's current movement decision is decodable in the reasoning chains of two models. Loud positions are more decodable than randomly selected ones for every direction probe we trained, decodability increases steadily with loudness, and the relationship survives removing the positions where direction words are written. Loudness for an unrelated signal carries no such information, indicating that the score is specific to the signal whose vocabulary defines it.

Why the J-Lens outperforms the Logit Lens. Both lenses read an intermediate activation in vocabulary space, but the Logit Lens applies the final unembedding directly to a representation that is not yet in the final layer's space. In intermediate layers it is therefore poorly calibrated: for Qwen3.6-35B-A3B, its loudness is flat across nearly all layers and varies within a narrow band across tokens (Appendix Figures 5 and 6). The J-Lens first maps the activation to the final layer through the model's average Jacobian and so reflects what the model is poised to verbalize (Gurnee

et al., 2026). This distinction matters most away from the surface text. When direction words and their neighbours are removed, Logit-Lens loudness no longer tracks decodability in GPT-OSS-20B, whereas J-Lens loudness still does. This suggests that J-Lens loudness captures latent content about the decision rather than the identity of nearby tokens, which builds onto the interpretation of the J-Space as a global workspace (Gurnee et al., 2026).

Practical use. Loudness requires only a signal vocabulary and a J-Lens for the layers of interest, which is fit once per model without task labels. Selecting a layer and ranking positions then costs a single pass over the activations, compared with training $O(L \times P)$ probes on labelled data in an exhaustive sweep. The selected layers fall in the intermediate band where prior work finds probes perform best, and the selected positions outperform a structurally defined heuristic (end-of-sentence tokens) without that position having to be chosen in advance. Our cross-selection results also suggest a division of labour: probes trained on broadly sampled positions generalise better, while loud positions are where any probe is most accurate. Loudness is therefore best used to decide where to apply a probe, and where to look for a signal in the first place.

Why the grid signal was inconclusive. The grid signal was intended as a second test of the method, but our design choices prevented it from serving that role. First, the probing task may be too broad in scope. The grid probe asks for the content of any queried cell, given the cell's coordinates concatenated to the activation, so a linear probe cannot condition its reading on the queried cell, and both linear and MLP probes decoded little of the grid state. Moreover, the model has little reason to hold the entire layout at every reasoning token, since it can attend back to the grid in the prompt. A narrower task, such as predicting whether each cell adjacent to the agent is a wall, would target information the model plausibly computes and carries forward for its next move, and would use coordinates relative to the agent, which mean the same thing across grid sizes. Second, the signal itself may be too broad. The grid state comprises the content of every cell, and grid loudness is a single score over a pooled vocabulary of cell-content words. Such a score measures whether the model is processing the grid in general rather than whether a particular cell is represented, and expanding generic anchors such as `wall` or `goal` likely admitted tokens used outside descriptions of the current grid. Loudness, as it is currently designed, seems to therefore be more likely to accurately predict decodability for signals smaller in scope. We treat the grid results as inconclusive, but believe decodability prediction is an underexplored problem which J-Lens loudness pushes forward.

Limitations. Beyond the grid signal, we study one environment and one decodable signal in two models, and have not measured sensitivity to the heuristically constructed signal vocabulary (Appendix A) or verified with a per-layer sweep that the selected layer is the most decodable. We treat the layer selection results as a prediction of which layers best encode the signal, a prediction which is supported by prior work.

6 Conclusion

We introduced J-Lens loudness, the log-probability mass that the J-Lens assigns to a signal's vocabulary, as a label-free criterion for choosing where to apply a probe. In the reasoning chains of GPT-OSS-20B and Qwen3.6-35B-A3B, positions that were loud for the agent's movement direction were more decodable than positions selected by Logit-Lens loudness, at random, or at sentence-final punctuation. Decodability increased steadily with loudness, persisted away from the positions where direction words are written, and was specific to the signal whose vocabulary defines the score. The loudest layers fell in the intermediate band which is consistent with prior work. Loudness therefore offers a cheap first step before, or in place of, an exhaustive sweep. Testing it on further signals, tasks, and models, with vocabularies that match the probe target, is the natural next step left for future research.

STATEMENTS

AI USE STATEMENT

We used generative AI tools to help with editing, restructuring, and reviewing of the text of this paper and to format its tables and figures in LaTeX. In addition, we used generative AI tools to aid code construction. We thoroughly reviewed all AI-assisted work, and take full responsibility for all final contents presented, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the environment, models, loudness computation, position selection, probes, and evaluation. The construction of the signal vocabularies is detailed in Appendix A and the remaining experimental details in Appendix B. Both models are publicly available. We also provide our code anonymised for the purpose of double-blind review at `https://anonymous.4open.science/r/Model-Internals-Speak-Volumes-J-Lens-Loudness-Predicts-Decodability-BCA6/README.md`. The link will remain available until 1st October 2027, after which it will forward to the de-anonymised github repository.

REFERENCES

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

Raghu Arghal, Fade Chen, Niall Dalton, Evgenii Kortukov, Calum McNamara, Angelos Nalmpantis, Moksh Nirvaan, Gabriele Sarti, and Mario Giulianelli. A behavioural and representational evaluation of goal-directedness in language model agents. *arXiv preprint arXiv:2602.08964*, 2026.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 240–248, 2018.

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. URL https://openreview.net/forum?id=C5Jj3QKQav.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *International Conference on Learning Representations*, volume 2024, pp. 2483–2503, 2024.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, et al. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026.

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.

János Kramár, Joshua Engels, Zheng Wang, Bilal Chughtai, Rohin Shah, Neel Nanda, and Arthur Conmy. Building production-ready probes for gemini. *arXiv preprint arXiv:2601.11516*, 2026.

William Lugoloobi, Thomas Foster, William Bankes, and Chris Russell. Llms encode their failures: Predicting success from pre-generation activations. *arXiv preprint arXiv:2602.09924*, 2026.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Alex McKenzie, Urja Pawar, Phil Blandfort, William Bankes, David Krueger, Ekdeep Singh Lubana, and Dmitrii Krasheninnikov. Detecting high-stakes interactions with activation probes, 2026. URL https://arxiv.org/abs/2506.10805.

nostalgebraist. interpreting gpt: the logit lens, 8 2020. URL https://alignmentforum.org. AI Alignment Forum.

OpenAI. Gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations. In *International Conference on Learning Representations*, volume 2025, pp. 66880–66913, 2025.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4609–4622, 2020.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3363–3377, 2021.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Mary Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, Eric J Michaud, Stephen Casper, Max Tegmark, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Thomas McGrath. Open problems in mechanistic interpretability. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=91H76m9Z94. Survey Certification.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

A CAPTURING SIGNALS

We construct V_d and V_g separately for each model using three steps.

First, we manually specify a small collection of *anchor words* intended to describe the relevant signal. The anchors need not themselves correspond to individual tokens in the model vocabulary, although in our experiments they do.

Second, we perform a lexical expansion of the anchor set. For every vocabulary token, we compare its decoded string against each anchor using `difflib` and retain lexically similar tokens according to our selection criterion. For example, using `right` as an anchor for the direction signal recovers token strings such as `.right` and `/right`.

Third, we perform a semantic expansion, comparing the meaning of each vocabulary token against the same anchors using a semantic similarity model. This captures related tokens that need not be lexically similar to the anchors; for example, an anchor such as `right` can recover translations such as `derecho`.

We take the union of the lexical and semantic expansions to form V_s and prune some of the words we deem to be false positives. This procedure is performed separately for each signal and model, since the two models use different token vocabularies. The resulting vocabulary sizes are found below in Table 3.

Table 3: Subset sizes for each signal

Vocabulary	**Number of tokens**
GPT-OSS-20B direction	446
GPT-OSS-20B grid	642
Qwen3.6-35B-A3B direction	643
Qwen3.6-35B-A3B grid	533

B EXPERIMENTAL DETAILS

Data. Each example is the reasoning chain of a single environment step, and training and evaluation sets are split by chain. The training sets contain $2{,}880$ chains for GPT-OSS-20B and 556 for Qwen3.6-35B-A3B. The held-out evaluation sets contain 720 chains for GPT-OSS-20B and 52 for Qwen3.6-35B-A3B, giving $14{,}391$ and $3{,}120$ positions per top-N subset, respectively, and $14{,}472$ positions per sentence-level subset for GPT-OSS-20B. For Qwen3.6-35B-A3B, 20% of reasoning-token positions are subsampled uniformly before any selection.

J-Lens. For each model we estimate $\bar{J}_l$ on the model's own grid trajectories, drawn from the training split only and balanced over grid size and obstacle complexity. Each context is the exact prompt the model saw followed by its full completion. For a context, row i of J_l is the gradient of $\sum_{p'} h_T[p', i]$ with respect to $h_l[p]$, averaged over source positions p in a fit window. $\bar{J}_l$ is the mean of these over contexts, with T the final block (where the lens is the identity).

For GPT-OSS-20B we use $2{,}880$ contexts each truncated to 1024 tokens and exclude the 414-token instruction prefix shared by every prompt from the training, so the averaged positions cover the grid and the opening of the reasoning. We stop early once the mean relative change of $\bar{J}_l$ falls below 0.2%.

For Qwen3.6-35B-A3B, whose reasoning chains have a median of about 18k tokens, we use 144 contexts (4 per size $\times$ complexity cell) and fit the nine full-attention layers (3, 7, …, 35) against layer 39. Positions are a 512-token window ending halfway through the reasoning chain. The preceding tokens are run through the model without gradient tracking, so the window activations keep their full context at a fraction of the cost. In both cases the transported state is passed through the model's own final RMSNorm before the unembedding, $\text{logits} = W_U\,\text{RMSNorm}(\bar{J}_l h_l)$, using the zero-centred $(1 + w)$ scale for Qwen3.6-35B-A3B.

Probes. Probes are trained on the residual stream at a single layer, after per-dimension standardisation using training-set statistics. The linear probe is a multinomial logistic regression. The MLP has one hidden layer of 1024 ReLU units, with dropout 0.1 for the direction probe and none for the grid probe. Both are trained with AdamW (learning rate 3×10^{-4}, weight decay 10^{-3}) for 50 epochs, using a class-balanced cross-entropy loss (weights $n/(Kn_c)$), and we report the final-epoch model without early stopping. Train and validation sets are split by trajectory.

The direction probe predicts one of the four actions from a single reasoning-token activation. Its label is the model's local belief: the action it outputs when its reasoning is truncated at that token, the reasoning channel is closed and the final-answer channel is opened.

The grid probe predicts the contents of a queried cell (agent, wall, goal, empty, or padding). The cell's (row, column) coordinates are appended to the activation as two unnormalised scalar features. Grids are padded to 15×15, and padding is kept as its own class. We draw 25 cells uniformly per step with a fixed per-step seed, so every token-selection arm is evaluated on the same cells.

Decile analysis. Deciles are computed over all held-out tokens pooled across chains. Confidence intervals are 95% intervals from 300 bootstrap resamples of held-out chains. For the signal-word exclusion, every token of V_s and the two tokens on either side within the same chain are removed before recomputing deciles.

Compute. We used a collection of GPUs to run our experiments. For activation gathering, we used primarily A100 GPUs, while all other computations were performed on cheaper GPUs such as RTX 2000 Ada and RTX PRO 4500. Part of our system uses the MooseFS file system, which slows down navigation and path finding methods such as `glob` significantly. This contributed to a much slower compute time than necessary. In total, to reproduce the experiments in this paper using 1 GPU at a time, we estimate 90 hours. Most of the compute is dominated by the need to fit our own J-Lens, something which we expect to no longer be an issue once more trained J-Lens weights are open sourced.

Prompt.

```
<|start|>system<|message|>You are ChatGPT, a large language model trained by OpenAI.
Knowledge cutoff: 2024-06
Current date: 2026-01-17

Reasoning: medium

# Valid channels: analysis, commentary, final. Channel must be included for every message.<|
    end|><|start|>user<|message|># Instructions

You are controlling an agent in a grid-based environment with full observability. The agent
    can move in four directions: up, down, left, and right. The environment contains walls,
     open spaces, and a goal location. The following symbols are used in the grid
    representation:

Legend:
---------------
#: Wall
_: Open Space (can be visited)
G: Goal
A: Current agent position
----------------

Your objective is to navigate from the current position (A) to the goal (G) while avoiding
    walls (#). Importantly, the agent should aim to reach the goal using the least amount
    of steps possible. You must decide your next move based on the provided information.

You will receive the current state of the grid as a NxM matrix of symbols separated by
    whitespaces, and with coordinates for each row an column. For example, given the 4x5
    grid:

  0 1 2 3 4
0 # # # # #
1 # _ _ G #
2 # A _ _ #
3 # # # # #

The goal is in position (1,3), while the agent's position A is (2,1).

The agent's possible actions at each step are:
```

```
Actions:
- UP: Move Up
- DOWN: Move Down
- LEFT: Move Left
- RIGHT: Move Right

Your final answer should be a valid JSON object provided in the following form:
{
  "action": "<UP|DOWN|LEFT|RIGHT>",
}

DO NOT INCLUDE ANY `json` or `jsonb` in your response NOR TICK MARKS LIKE THIS: ```json ```.
      Start with { and end with } exactly.

# Inputs

Current grid state:
{{grid_state}}

<|end|><|start|>assistant
```

For Qwen3.6-35B-A3B we replace the reasoning tokens with `<think>` and `</think>`.

C Additional results

C.1 Logit-Lens layer profiles

For GPT-OSS-20B, Logit-Lens loudness peaks at layer 13 for both signals and declines earlier than J-Lens loudness (Figure 2). For Qwen3.6-35B-A3B, the profile is flat from layer 0 to 33, so its argmax (layer 2) carries no information about depth (Figure 5).

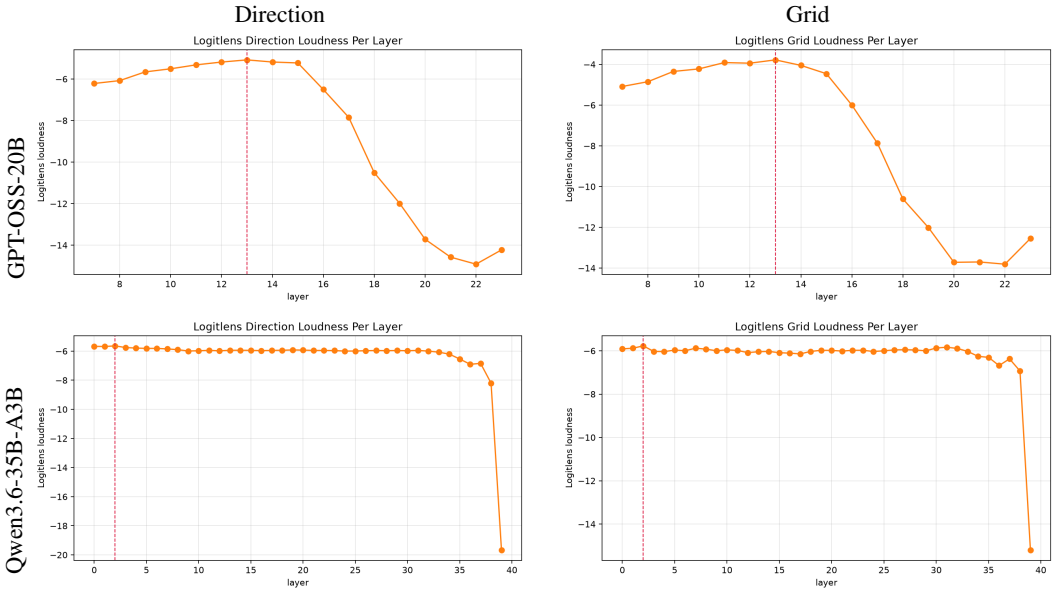


Figure 5: Mean Logit-Lens loudness per layer. Dashed line: argmax.

C.2 Agreement between J-Lens and Logit-Lens loudness

At the token level, J-Lens and Logit-Lens loudness for the same signal are only moderately correlated (Spearman ρ between 0.32 and 0.60; Figure 6), so the two lenses select different top-N positions. For Qwen3.6-35B-A3B, Logit-Lens loudness varies within a narrow band, while J-Lens loudness spans roughly ten nats.

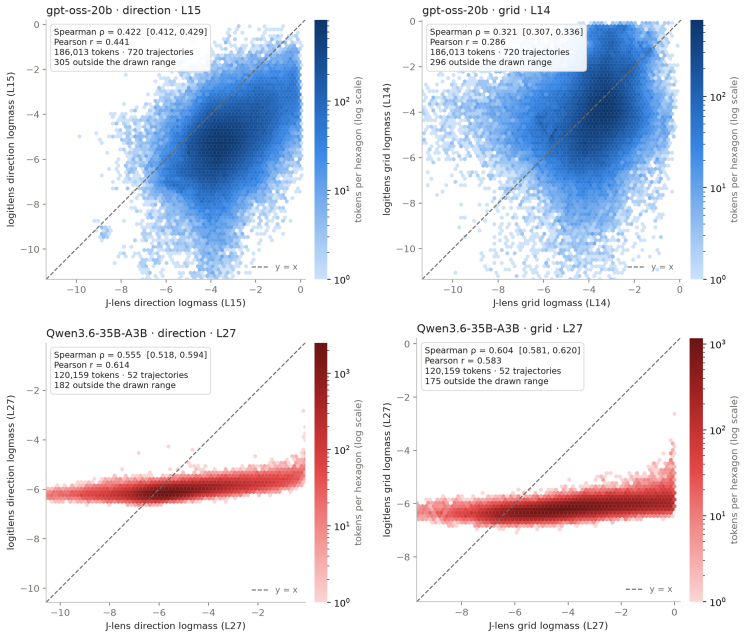


Figure 6: Token-level J-Lens vs. Logit-Lens loudness at l_s^*. Top: GPT-OSS-20B; bottom: Qwen3.6-35B-A3B; left: direction; right: grid.

C.3 Agreement between direction and grid loudness

Direction and grid loudness are nearly uncorrelated at the token level under both lenses and in both models ($|\rho| \leq 0.062$; Figure 7).

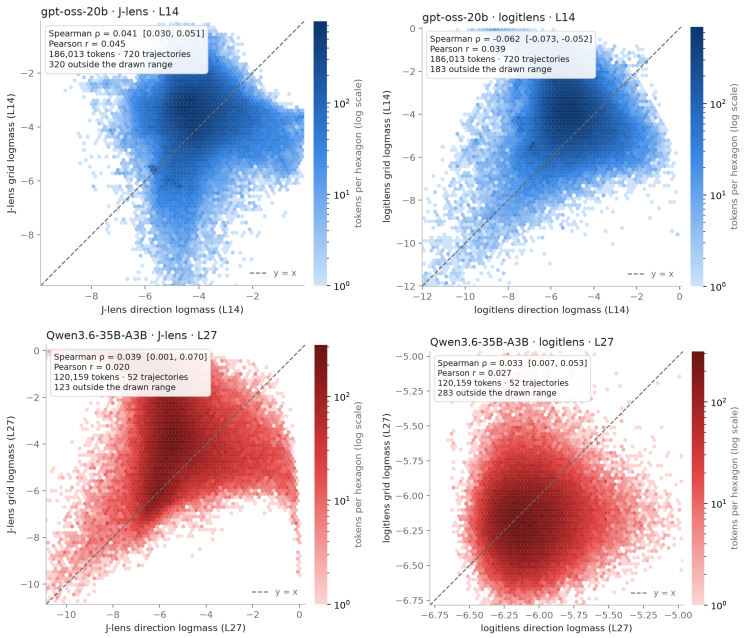


Figure 7: Token-level direction (x) vs. grid (y) loudness. Top: GPT-OSS-20B; bottom: Qwen3.6-35B-A3B; left: J-Lens; right: Logit Lens.

C.4 GRID CROSS-SELECTION RESULTS

Table 4 reports every combination of training criterion (rows) and evaluation criterion (columns) for top-N selection at l_s^* for the grid signal, for both probe architectures, on the same position subsets as the direction results in Table 1. Accuracy is balanced over the four cell classes, excluding padding. **Bold** marks the best evaluation subset for each probe (computed on unrounded values) and underlining marks in-distribution evaluation. Balanced accuracy; chance is 0.25.

Table 4: Grid, top-N selection. Rows: training criterion; columns: evaluation subset.

		GPT-OSS-20B			Qwen3.6-35B-A3B		
Trained on	Probe	J-Lens	Logit Lens	Random	J-Lens	Logit Lens	Random
J-Lens	Linear	.292	**.294**	.283	.258	.240	**.263**
	MLP	.438	**.445**	.402	**.371**	.346	.349
Logit Lens	Linear	.289	**.303**	.286	**.252**	.247	.252
	MLP	.421	**.473**	.403	**.374**	.370	.345
Random	Linear	.290	**.303**	.289	**.245**	.234	.243
	MLP	.413	**.432**	.410	**.360**	.354	.344

C.5 PROBE ACCURACY BY LOUDNESS DECILE

All panels follow the conventions of Figure 3: balanced accuracy on held-out tokens, binned into deciles of loudness pooled across reasoning chains; solid lines are linear probes and dashed lines MLP probes. In the signal-word-excluded variants, every token of V_s and the two tokens on either side are removed and deciles are recomputed on the remaining tokens. This removes 27.9% (GPT-OSS-20B) and 10.0% (Qwen3.6-35B-A3B) of tokens for direction, and 29.1% and 34.0% for grid. As opposite-signal controls, direction probes binned by Logit-Lens grid loudness (the J-Lens version is Figure 4) and grid probes binned by direction loudness under either lens show no consistent trend (Figure 10).

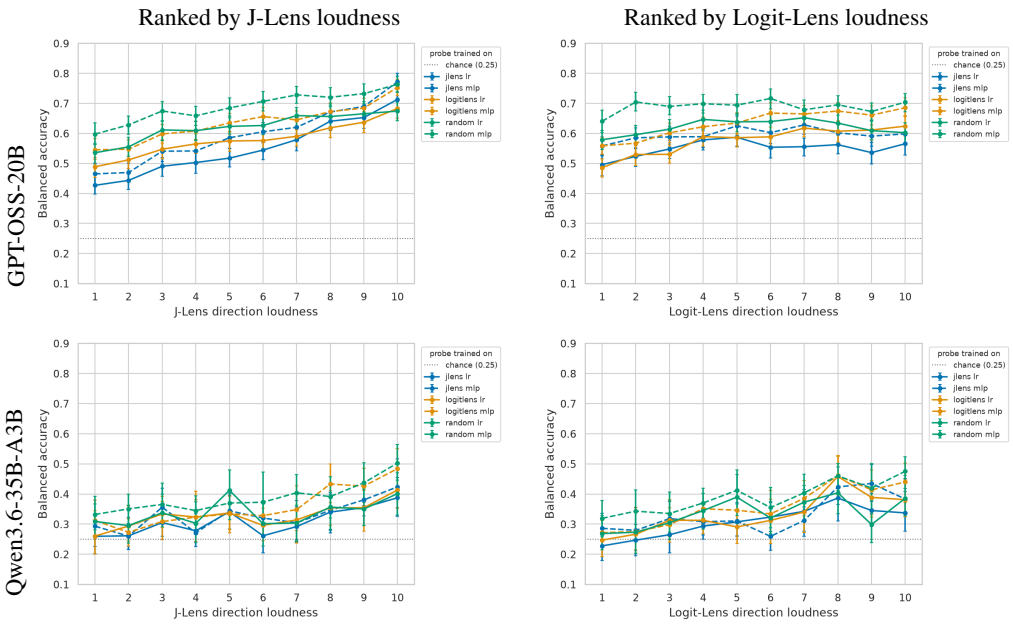


Figure 8: Direction probes by direction-loudness decile, excluding direction words and their ±2 neighbours.

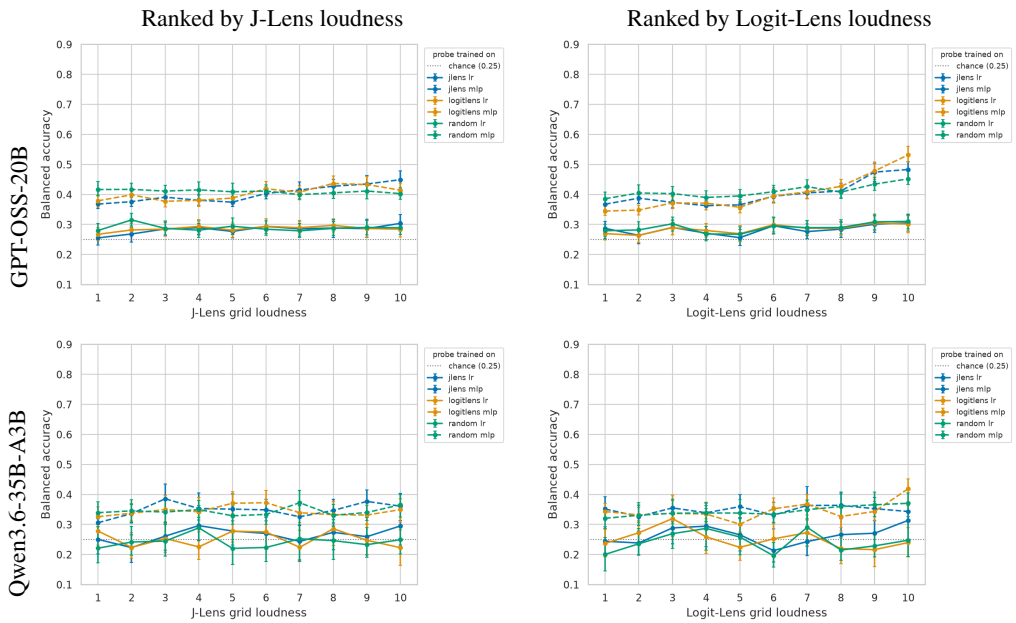


Figure 9: Grid probes by grid-loudness decile at l_g^*.

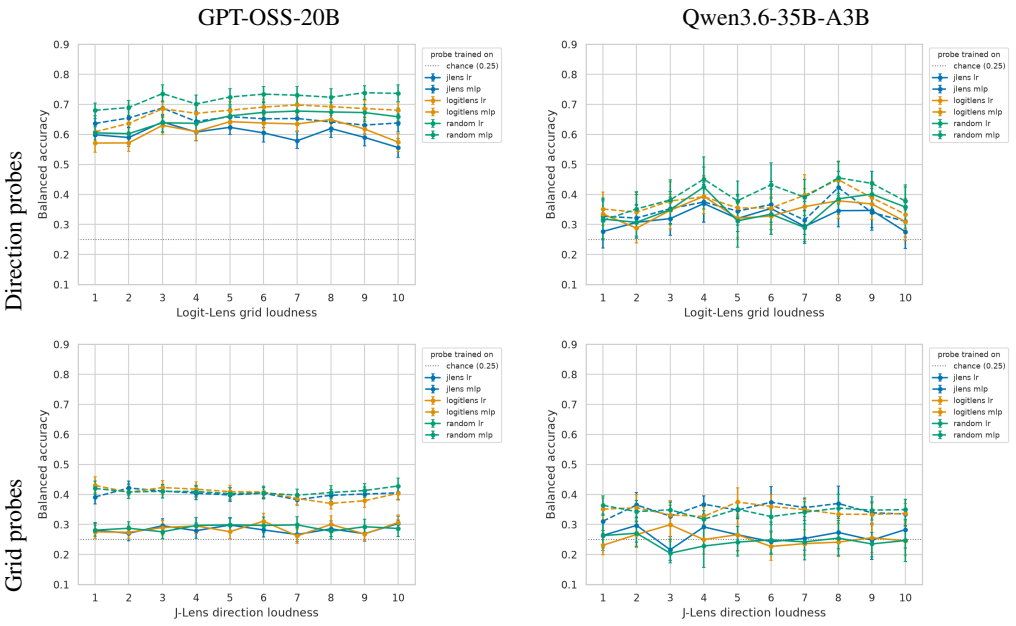


Figure 10: Opposite-signal controls. Top: direction probes by Logit-Lens grid loudness. Bottom: grid probes by J-Lens direction loudness.

C.6 GRID-STATE PROBES

Table 4 and Figure 9 report the grid probes. Linear probes stayed close to chance on every subset and MLP probes remained well below direction-probe accuracy. For Qwen3.6-35B-A3B, J-Lens-selected positions were marginally the most decodable, and no lens showed a trend across deciles. For GPT-OSS-20B, Logit-Lens-selected positions were the most decodable subset for all six probes, and MLP accuracy rose with Logit-Lens but not J-Lens grid loudness, including after grid words were excluded. Because the probes decode so little of the grid state, we do not interpret these differences as evidence for or against either lens (Section 5). Grid probes binned by direction loudness showed no trend (Figure 10, bottom).

C.7 SELECTED TOKENS

As expected, the top tokens in terms of direction J-Lens loudness are the exact direction tokens `UP`, `DOWN`, `LEFT` and `RIGHT`. Filtering out all V_d tokens from all selections, as well as their ± 2 surrounding neighbours, shows us that selected tokens are not trivial selections (Table 6). Most of the loudest remaining tokens are coordinates, brackets and arrows written while the model traces a path through the grid, e.g. `(1,7)->(2,7)`, where consecutive coordinates implicitly specify a move, or simply where a model is recalling the agent's and the goal's location in the grid. This implies that loudness can find token positions where direction information is present, but not necessarily explicitly verbalized.

Table 5: GPT-OSS-20B, direction, $l_d^* = 15$: the 20 loudest distinct tokens remaining after removing V_d tokens and their ± 2 neighbours. n: occurrences; context: loudest occurrence, with the token in bold; ␣: leading space.

#	Token	n	Loudest occurrence	Loudness
1	`,`	1,051	`From (7 , 1). We`	−0.118
2	`6`	60	`4,8)#, ( **6** ,8)`	−0.142
3	`␣vertically`	3	`They are adjacent **vertically** : G below`	−0.206
4	`␣straight`	7	`blocked. So cannot go **straight** . Need det`	−0.267
5	`5`	72	`A down to (2, **5** ). Then left`	−0.308
6	`␣`	31	`␣,6# Row3: ␣ 0#,1`	−0.328
7	`␣Lik`	22	`Need go around. **Lik** ely go down`	−0.337
8	`(`	66	`4) 1, -> **(** 2,4`	−0.349
9	`␣go`	42	`,4) #. So **go** to (1`	−0.351
10	`4`	62	`7␣,8#. So ( **4** ,7)`	−0.356
11	`␣horizontally`	4	`They are adjacent **horizontally** . So move`	−0.356
12	`␣(`	267	`(6,8)␣, **(** 5,7`	−0.409
13	`)->`	101	`left: (3,2 **)->** (3,`	−0.433
14	`2`	147	`(1,7)->( **2** ,7)->`	−0.463
15	`8`	43	`(9,3)->( **8** ,3)->`	−0.525
16	`3`	219	`So path: from ( **3** ,2)`	−0.549
17	`,(`	2	`,(1,3)2 ,**(** 2,3`	−0.579
18	`␣moving`	1	`Let's consider **moving** to (3`	−0.594
19	`7`	90	`From ( **7** ,6)`	−0.603
20	`␣det`	11	`So need **det** our. From`	−0.659

Table 6: Qwen3.6-35B-A3B, direction, $l_d^* = 27$: the 20 loudest distinct tokens remaining after removing V_d tokens and their ± 2 neighbours. Columns as in Table 5.

#	Token	n	Loudest occurrence	Loudness
1	␣	164	`2: _ Col ␣ 3: _`	−0.028
2	␣(	68	`- From ( 1,`	−0.039
3	:	44	`, 1) Neighbors : - Up`	−0.052
4	␣-	8	`) is Right. - (1,`	−0.054
5	,	167	`then Right to (4 , 5)`	−0.058
6	␣to	13	`(1, 3) to (2,`	−0.105
7	.	52	`moves:** 1 . Start at (`	−0.164
8	␣a	3	`Is there a path going Left`	−0.165
9	(	20	`5) is _. ( 4,6`	−0.165
10	␣->	31	`(1, 3) -> (2,`	−0.191
11	Moving	1	`(2,3). Moving from (3`	−0.208
12	␣requires	1	`from 1,3) requires ending with DOWN`	−0.249
13	1	64	`2 is _) ( 1 ,1)`	−0.265
14	`	6	`- Index 8: **␣** #`	−0.283
15	Let	2	`Correct. Let 's do a`	−0.296
16	),	11	`From (1,3 ), can go DOWN`	−0.341
17	␣further	1	`, 3) but not further . At`	−0.361
18	3	51	`1): Open (_) ( 3 ,3):`	−0.396
19	␣there	1	`Blocked. Is there a path going`	−0.422
20	)	57	`- (1, 3 ) to (2`	−0.436