A bracket neuron in GPT-2, and the label that hid it

Will Sackfield, Open Interpretability · https://open-interpretability.com/research/syntax-positive-controls

Abstract

Syntax labels from a parser should find the code neurons that the literature leads us to expect. So we swept every neuron pole of GPT-2 small over labelled source code. Our first bracket-depth label found nothing convincing, because it counted all brackets together. Counted for round brackets alone, one pole stands out: L2 N676+ is the best pole in its layer and across all 12 layers. On written text the same token reads 4.19 to 11.02 under one to five open brackets, and near zero once they close. At one open bracket, ablating the pole lowers the predicted chance of a closing bracket from 0.46 to 0.20. It is a short-range, imperfect counter of bracket characters. Through the registry's review, a claim of round brackets alone earns NEEDS_REVISION, because the pole also fires under square and curly brackets. A claim of round and square brackets, with curly ones weak, is VERIFIED. Four more poles in layers 2, 3, 8 and 9 respond to open round brackets too. All four are VERIFIED through the same review, L9 N340+ on its second claim. Ablating L2 N676+ makes the model close one bracket where it would close several, and ablating the other four makes it close too many. Without L2 N676+, the model also stops closing brackets that stay open. Later layers read the pole through the head L3H5 and then L8 N689+, another round-depth pole. Qwen2.5-0.5B and 1.5B have no round-depth pole as strong: their best have R^2 0.160 and 0.175, and 0.168 and 0.161 at GPT-2's windows. Their best poles do not count depth.

Status: Published 2026-11-20. Changes since are in the addenda at the end.

1 Introduction

Recurrent networks trained on source code hold cells that track its syntax (Karpathy et al., 2015). It is often assumed that transformer language models hold similar units for code. A recurrent cell carries a state from one step to the next, though, and a transformer's MLP neuron has no such state. Whether one neuron of a transformer tracks how deep a token sits inside brackets is therefore an open, empirical question.

The question also tests a method. A registry of neuron explanations can find candidates for a concept by sweeping every neuron over text that carries labels for that concept. If such a sweep cannot find the code neurons that the literature leads us to expect, its silence about other concepts means little. We therefore use bracket depth as a positive control for the sweep. We study each neuron one sign at a time, and call one sign of a neuron a *pole*. We write L2 N676+ for the positive pole of neuron 676 in layer 2. An

explanation of a pole is *VERIFIED* when it passes every gate of an adversarial review, in which a second model writes probes meant to break it (*Method*).

This paper sweeps every pole of GPT-2 small (Radford et al., 2019) over source code that a parser labels. Our first bracket label counted all brackets together, and its best poles were false leads. A label that counts round brackets alone finds L2 N676+, whose strength rises as round brackets open and falls once they close (Figure 1). We then ask whether the model uses the pole, whether its explanation survives review, where later layers read it, and whether two larger models hold a pole like it.

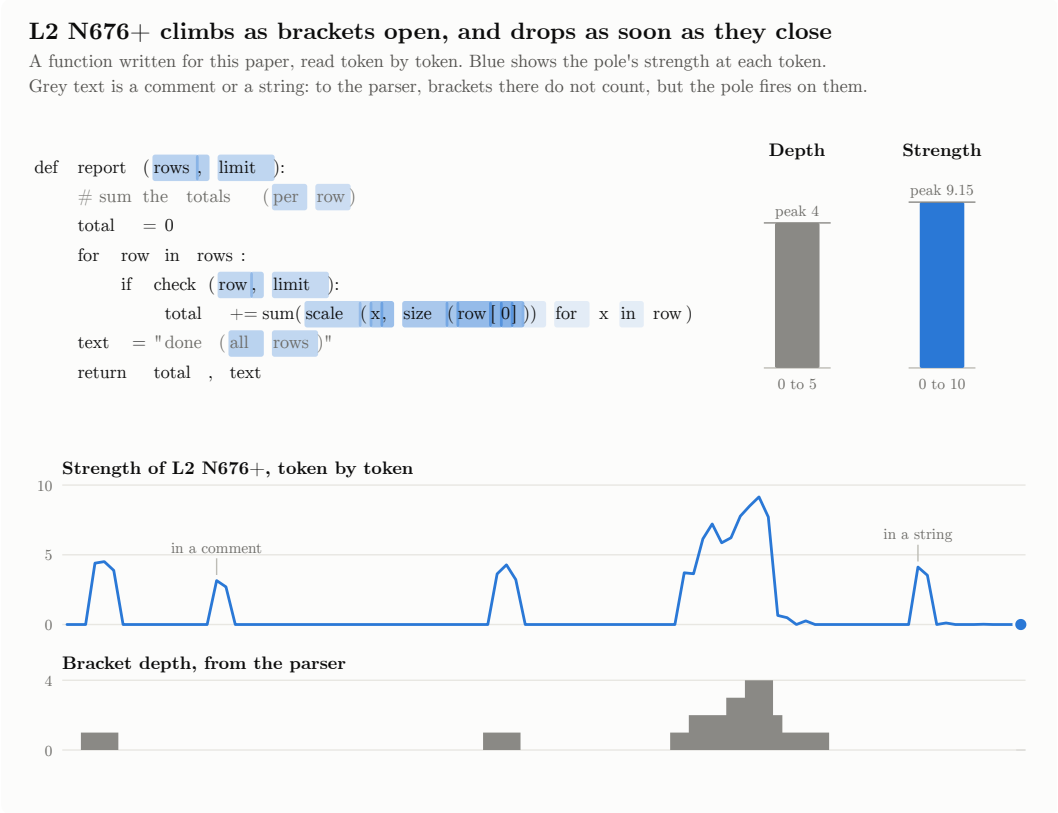

Figure 1. A short Python function, read token by token, with each token lit by the strength of L2 N676+. Tokens inside round brackets are lit, most at depth 4, where the strength reaches about 9. Tokens inside brackets in a comment and in a string are lit too, although the parser's depth there is 0.

The paper's contributions are these:

- A sweep of all 73,728 poles of GPT-2 small over parser labels on code, against a null. A label that counts all brackets hid the best bracket pole, and a count of round brackets alone finds it, with a held-out R^2 of 0.277.
- Written probes and ablations that show L2 N676+ to be a short-range, imperfect counter of bracket characters, which sets how many brackets the model closes.
- Adversarial reviews of five round-depth poles. Each one is VERIFIED, two of them after a revision of the claim.
- A path from the pole to the closing bracket, through the head L3H5 and then L8 N689+.

- The same sweep on Qwen2.5-0.5B and Qwen2.5-1.5B, which finds no pole as strong.
- A way to label documents that the corpus cuts inside a construct. It lowers the share of characters in a construct that read as top level from 27.8% to 1.7%.

The paper is a positive control for research question 1, *Are native MLP neurons clean?* (research questions).

2 Related work

Syntax cells in recurrent models. The expectation came from recurrent models. Karpathy et al. (2015) trained LSTM cells on the Linux kernel source. They report cells that "turn on inside quotes, the parenthesis after if statements, inside strings or comments". Another cell turns on "with increasing strength as the indentation of a block of code increases". That is a character-level LSTM, which carries state from step to step. A transformer's MLP neuron has no such state, so the step from those cells to GPT-2 is an assumption. This repository's list of open work made that step when it said that these neurons are known to exist in code models.

Brackets in transformers. Evidence closer to GPT-2 is mixed. Friedman et al. (2024) study bracket depth in small transformers trained on parenthesis languages, and in character-level models trained on code, not in GPT-2. Gurnee et al. (2024) show a suppression neuron for tokens with an open parenthesis in a GPT-2 medium model. That is about output tokens, not depth. Clark et al. (2023) searched GPT-2 small for a circuit that tells a position it is inside brackets, in a report from a research sprint. They "did not find significant evidence of a circuit responsible for this behaviour", and found some evidence that the information is highly distributed. They also warn about the limits of the available tools. Our finding does not contradict theirs: one pole that matters is not a whole circuit, and the partial effect here fits a distributed signal.

Bracket balance as a program. Weiss et al. (2021) propose RASP, a programming language that models the transformer encoder. In RASP, an attention head is the only way to move information between positions. Their tasks include prefix classification for balanced brackets of one and two kinds (Dyck-1 and Dyck-2), and they train transformers to follow the compiled programs. Their result says how many layers and heads such a task needs, not how a trained model divides it. We study the second question in a model trained on ordinary text, where the count turns out to rest on both MLP neurons and attention heads.

Poles as units. We read each sign of a neuron apart, as the registry does. Tigges et al. (2026) also read each pole apart, on a large mixture-of-experts model. They report that a pole's positive and negative values "usually (but not always)" have unrelated roles. In GPT-2's neuron 676, both poles score on round depth (0.277 and 0.225), so this neuron may be one of the exceptions. We did not audit L2 N676−, so this step is ours and untested.

3 Method

Poles and strength. Let h_t be a neuron's activation at token t. Its positive and negative poles have the strengths

$$a_t^+ = \max(h_t, 0), \qquad a_t^- = \max(-h_t, 0)$$

We write a_t for the strength of the pole under study. Each neuron has 2 poles, so each layer of GPT-2 small has 6,144.

Labels and round depth. A parser gives each token a label from a *label set*. `syntax node` names the parser's node type, and `syntax construct` sorts node types into shared constructs, such as call arguments. `bracket depth` counts the brackets of all kinds that are open at a token, and `indent depth` counts its indentation. *Round depth*, the label set `round depth`, counts the round brackets that are open at a token. Square depth and curly depth count the other kinds in the same way. Each depth label leaves out brackets in comments and strings, except the count of round characters with no parser.

Held-out R^2. For each pole and label set, per-label means from half of the documents predict strengths in the other half. The two ways to split are averaged. So a pole cannot score by fitting noise, and a useless label scores near or below zero. A planted signal scored 0.985 in a sanity check, and noise scored about 0. For a split with training half A and test half B, the score of one pole is

$$R^2 = 1 - \frac{\sum_{t \in B} \left(a_t - \mu_A(\ell_t) \right)^2}{\sum_{t \in B} \left(a_t - \bar{a}_B \right)^2}$$

where a_t is the pole's strength at token t, ℓ_t is the label of token t, $\mu_A(\ell)$ is the mean strength of tokens with label ℓ in A, and $\bar{a}_B$ is the mean strength in B. *Token* R^2 is the same score with token identity as the label, measured in-sample.

The count of brackets closed. Each closing token in GPT-2's vocabulary closes one to four brackets (`)` to `))))`). Let C be the set of these tokens, $k(v)$ the number of brackets that token v closes, and $p(v)$ its predicted probability. The mean number of brackets that a predicted closer closes is

$$\bar{k} = \frac{\sum_{v \in C} k(v)\, p(v)}{\sum_{v \in C} p(v)}$$

The probability of a closer is the summed probability of next tokens that start with `)`.

Ablation and patching. To *ablate* a positive pole, we set a_t^+ to zero at every token of its layer. To *patch* a component, we replace its output in one run with its output in another. *Noising* puts the component's output from the run with the pole ablated into the clean run, and its share is the part of the drop in the log-odds of a closer that it causes. *Denoising* puts the clean output into the ablated run, and its share is the part of the drop that it recovers.

The review. A proposer model reads the pole's strongest examples on a scan of code and writes a claim. A reviewer model, which cannot run the model, writes probes meant to break it. *Positive holdouts* are new texts that the claim says should fire the pole, and *hard negatives* are near misses that it says should not. *Counterfactuals* are pairs of texts that the proposer predicts will change the pole's strength. A probe fires the pole when its peak reaches the *activity line*, 0.25 of the pole's maximum on the scan. *Specificity* is the share of hard negatives that stay below the line, with near-duplicate probes counted as one cluster.

Gates and verdicts. The review's *gates* are the tests that a claim must pass. Among them, specificity must be at least 0.80, the hard negatives must fall into at least 6 clusters, the counterfactuals must reproduce, and the claim must beat a baseline of token

identity. A *lookahead check* compares any claim that the pole anticipates the next token with a measured profile. A claim is *VERIFIED* when every gate passes, and *NEEDS_REVISION* when a gate fails but the claim still separates the probes well. The number after a verdict is the claim's balanced accuracy on the probes. The review also records two causal scores from patching, *sufficiency* and *necessity*, without gating them. *Detection* and *fuzzing* score the claim on held-out windows: a judge model, given only the claim, picks the active windows or the correctly marked tokens. For each, 0.5 is chance.

Setup.

- **Model.** GPT-2 small (Radford et al., 2019), at 128-token windows, as the registry scans it. The sweeps ran on Apple's MPS backend, in float32.
- **Data.** 500 documents of `codeparrot/github-code-clean` at commit `c48d40f9`, seed 0, each cut to 3,000 characters. The card gives the licence as Apache-2.0 and names no citation. Its `license` column shows that each file keeps its repository's licence. Of these, 396 documents have a grammar, and the seeded 200,000-token sample holds 266 of them.
- **Labels.** Tree-sitter parses each document, with a grammar for each of 15 languages. Its page describes it as useful even in the presence of syntax errors. We sort each grammar's node names into shared labels by pattern, so the labels are approximate. The corpus cuts most documents inside a construct, so the tagger searches for an ending that fixes the parse (*Cut documents* in the Appendices).
- **Null.** Each window takes the labels of a window from the document before it. That keeps the labels' positions and run lengths, and removes their link to the text.
- **Probes.** Eight target tokens each follow `first,`, in templates that change only what comes earlier. All of the probe text is written for this paper. Each cell has 8 items, and the tables give the mean strength. We designed the probes after seeing the corpus results, so they were not predictions registered in advance.
- **Ablation.** `ablate_pole` zeroes the positive part of a neuron's activation at every token of L2. The 20 control poles are the positive poles of random L2 neurons, seed 0.
- **Reviews.** The proposer was Claude Opus 5.5, the reviewer Claude Sonnet 5.5 and the judges Claude Haiku 5.5. Each audit scanned the seed-1 sample of `github-code`, so its documents differ from those that the poles were chosen on.

Figures. Figure 1 moves: it reads a function token by token, so that a reader sees the strength rise and fall with the brackets.

4 Does a parser label find a bracket-depth pole?

The first question is whether parser labels on code find a pole that tracks bracket depth. Every label set far exceeds the null, which gives each window another document's labels. The highest null R^2 in any layer and set is 0.028, against a best of 0.605 for `syntax node`. The labels therefore carry signal. This does not show that a label's best pole encodes that label, because token identity explains more of that pole (0.78).

Our first depth label, `bracket depth`, counts round, square and curly brackets, and ignores those in comments and strings. Its best poles looked like a result until we read them (Table 8, in the Appendices). Its best pole overall, L4 N3038−, separates depth 0 from the rest, and its strength is flat from depth 1 up. The best pole for `indent depth`, L6 N1416−, peaks in licence comments. Without the licence header and depth 0, the winners

change (Table 9). On written text, none of these poles rises with depth (Table 10). The first bracket label's winners were therefore false leads.

L2 N676+ pointed to a better label. It scored 0.273 on `syntax construct`, and much of that is call arguments and parameter lists. Its mean strength also rose with depth. This suggested two reasons for the weak depth score: curly brackets, which mostly hold blocks, and strings and comments, which the label leaves out. We then split the count by bracket kind (Table 1 and Figure 2).

Count of depth	**Best pole, L2**	**Best R^2, L2**	**L2 N676+**	**Its rank in L2**
All brackets	L2 N1513−	0.142	at most 0.095 (in-sample)	not in the top 5
Round	L2 N676+	0.277	0.277	1
Round characters, no parser	L2 N676+	0.310	0.310	1
Square	L2 N1441+	0.051	0.017	4
Curly	L2 N1513−	0.106	0.000	4,039

Table 1. The count of depth split by bracket kind, in layer 2: the best pole of the layer and its held-out R^2, and the R^2 and rank of L2 N676+ among the layer's 6,144 poles.

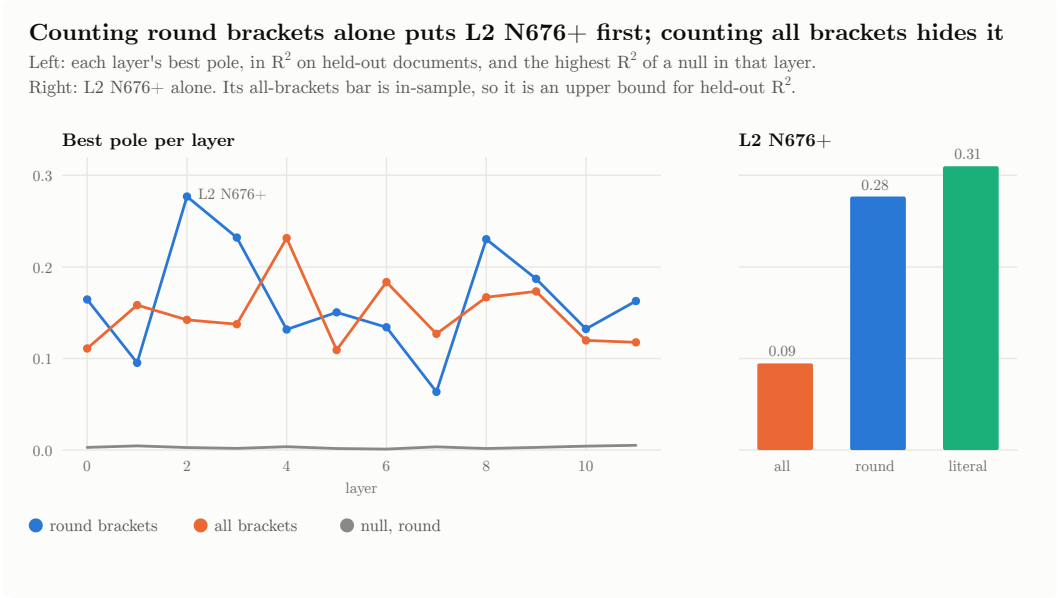

Figure 2. Left: the best held-out R^2 per layer for round depth peaks at 0.28 in layer 2, and the null stays below 0.01. Right: L2 N676+ scores 0.09 for all brackets, 0.28 for round brackets and 0.31 for round bracket characters.

Round depth finds L2 N676+. Its R^2 on held-out documents is 0.277, first of the layer's 6,144 poles, and 0.310 when we count `(` characters with no parser. On all brackets the same pole has at most 0.095. The first label hid it. Its round R^2 is the highest of any layer's best pole, and four more poles of layers 2, 3, 8 and 9 come next (Tables 12 and 13).

On written text. The corpus score could come from tokens that merely co-occur with brackets. Written probes test this with the target token and the token before it fixed

(Table 2 and Figure 3). With brackets open, the strength rises with depth from 4.19 to 11.02. When they close, it falls to about zero. In one snippet it climbs to 9.15 at depth 4, and drops to 0.65 at the token that closes two brackets. Written text therefore supports a depth signal.

Condition	Depth 1	2	3	4	5
Open: `out = load(parse(first, value`	4.19	6.75	8.63	9.99	11.02
Closed: `out = load(parse(x))`, then a new line	0.00	0.00	0.12	0.54	0.00
One closed: one more opened, then closed	1.65	4.11	5.82	7.08	8.00
In a string	4.63	7.06	8.85	10.17	11.17
In a comment, then a new line	2.30	4.72	6.48	7.77	8.76
Square brackets	3.51	5.62	7.14	8.22	9.02
Curly brackets, as `load{first, value`	1.49	2.82	3.76	4.45	4.98
Each call on its own line	2.82	3.36	2.52	1.38	0.49

Table 2. The mean strength of L2 N676+ at one target token, by the number of brackets before it and by condition, over 8 items. The standard errors over items are at most 0.21.

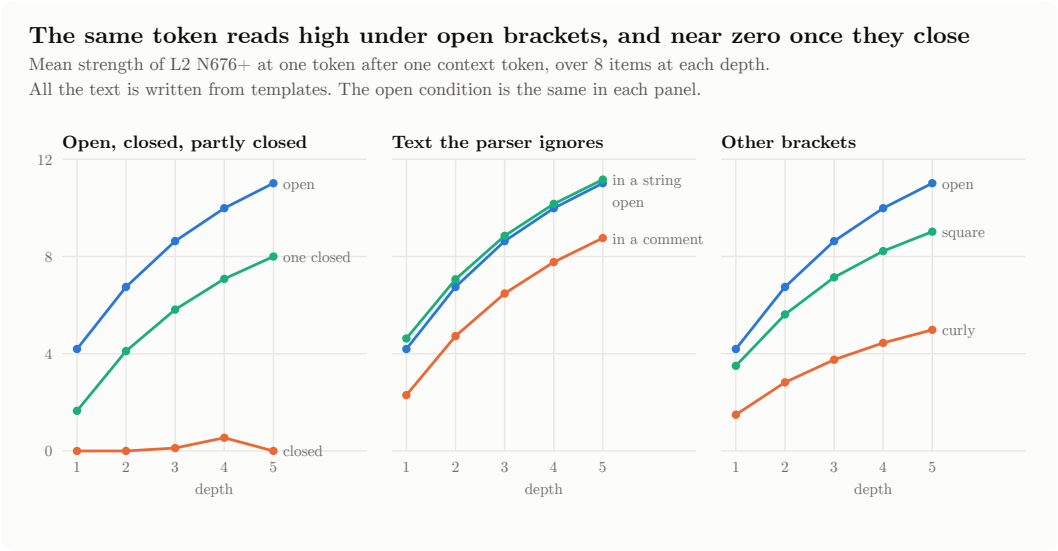

Figure 3. The strength of L2 N676+ at one token against depth. It rises from 4 to 11 under open brackets, and stays near 0 under closed ones. It rises in strings and comments too, and less, to 5, under curly brackets.

The signal has clear limits (Table 11 and the Appendices). It does not fully count down: under `One closed` it reads lower than under `Open`. Strings and comments do not stop it, so it follows characters and not the grammar. With each call on its own line, the strength peaks at depth 2 and then falls. It responds to `load{`, but no round-depth pole responds to a curly bracket that opens a block (Table 14). And it fades with distance (Figure 4): with one open bracket and a list of filler items before the target, it falls from 4.23 to 0.56 after 10 items, and to 0.00 after 20.

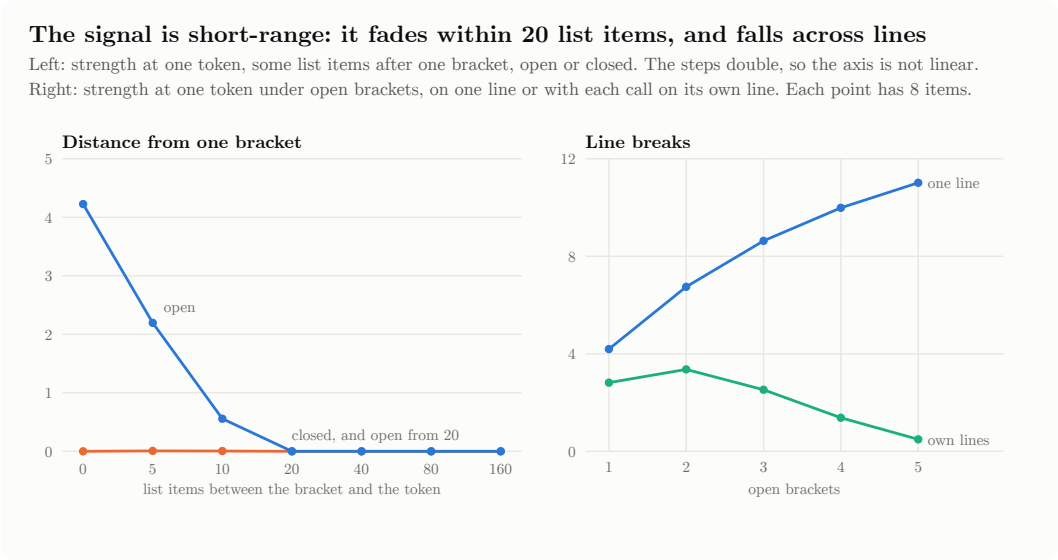

Figure 4. Left: under one open bracket, the strength falls from 4.2 with no filler to 0.6 after 10 list items and 0 after 20. Right: on one line it rises from 4 to 11 with depth, and with each call on its own line it peaks at 3.4 at depth 2.

5 Does the model use the pole to close brackets?

A pole that tracks depth need not matter to the model. We therefore ablated L2 N676+ and read the probability of a closing bracket after the open probes (Figure 5). At one open bracket, ablating the pole lowers the probability from 0.457 to 0.199 (Table 15). Twenty random poles of the layer barely change it. The pole thus carries part of what the model uses at low depth. The probability does not reach zero, which fits other routes to the same output.

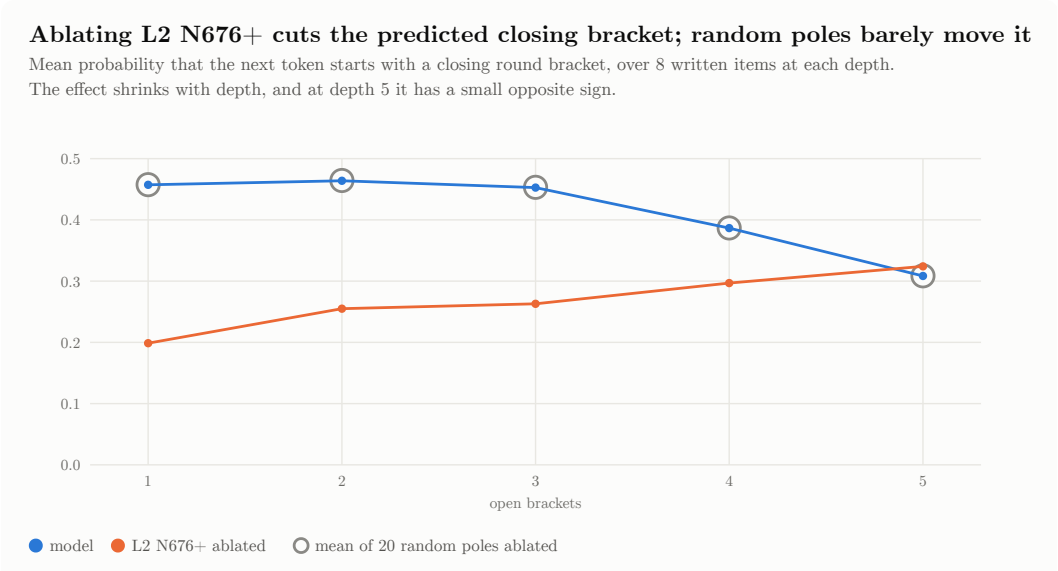

Figure 5. The probability of a closing bracket against depth. The model and twenty random ablated poles overlap, from 0.46 to 0.31. With L2 N676+ ablated, the probability is 0.20 at depth 1, and it rises to 0.32 at depth 5.

How many brackets close. The probability of any closer cannot tell `)` from `))`. The count of brackets closed, $\bar{k}$, measures that (Table 3 and Figure 6). The model's predicted closer closes 1.0 bracket at depth 1 and 1.95 at depth 5. With L2 N676+ ablated, it closes about 1.0 at every depth. Twenty random sets of 5 positive poles, one in each layer of the five, seed 0, barely change either measure.

Open brackets	**1**	**2**	**3**	**4**	**5**
Model	1.02	1.58	1.79	1.90	1.95
L2 N676+ ablated	1.01	1.02	1.03	1.02	1.02
L2 N1195+ ablated	1.24	2.33	2.25	2.51	2.93
L9 N340+ ablated	1.04	1.73	2.03	2.25	2.37
The four other than L2 N676+ ablated	1.96	2.71	3.67	3.78	3.81
All five ablated	1.07	1.74	2.05	2.07	2.05
20 random sets ablated, range of means	1.02	1.57 to 1.58	1.78 to 1.79	1.89 to 1.91	1.94 to 1.96

Table 3. The mean number of brackets that a predicted closer closes, $\bar{k}$, by the number of open brackets, with round-depth poles ablated alone or together. The count is capped at 4, because no single token closes more.

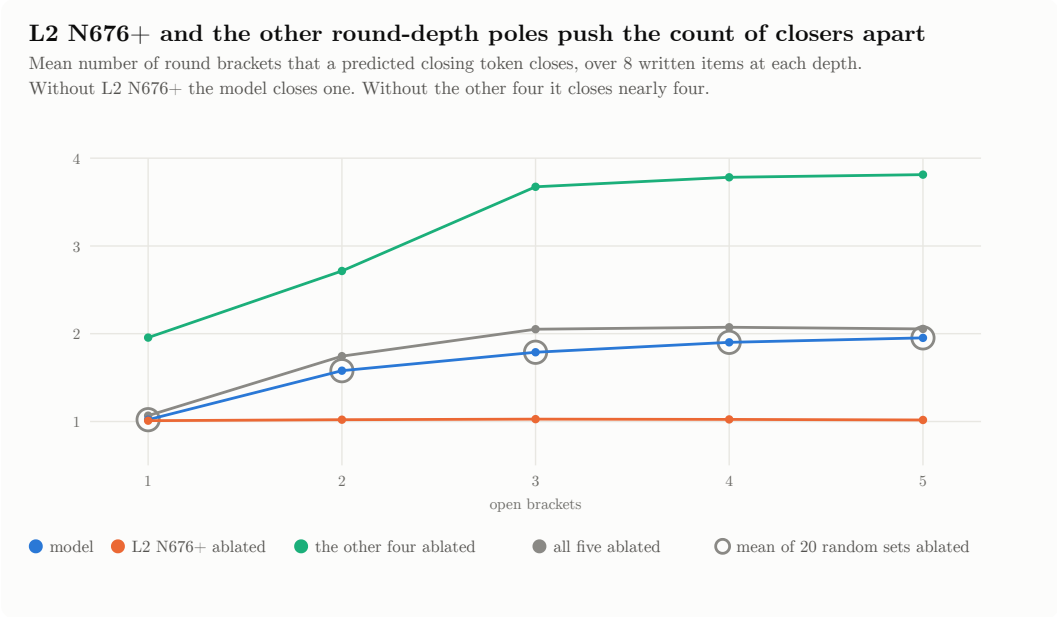

Figure 6. The number of brackets closed against the number open. The model rises from 1.0 to 1.95, and the random sets overlap it. With L2 N676+ ablated, the count stays near 1.0. With the other four round-depth poles ablated, it rises to about 3.8 from depth 3.

The other round-depth poles pull the other way. L3 N695+ and L8 N689+ alone change the count by at most 0.1. With the other four ablated, the model closes too many: about 3.8 from depth 3, where the model closes about 1.8. From depth 3 it then puts 0.91 to 0.99 of its probability on a closer, and most of that on `))))`. With all five ablated, the two effects about cancel. So L2 N676+ pushes the count up, and the other four, mostly L2

N1195+ and L9 N340+, hold it down. From depth 2, the model closes fewer brackets than are open. This fits a model that closes some of them now and the rest later.

Closing the rest later. To test this, we add the model's most probable closer to each of the 8 items of each depth, again and again until no bracket is open. After each closer, we read the probability that the next token starts with `)` (Table 4 and Figure 7).

Open brackets	**1**	**2**	**3**	**4**	**5**
Items with brackets open after the first closer	0	1	8	8	8
P(one more closer), brackets still open	–	0.174	0.622	0.545	0.363
P(one more closer), all closed	0.001	0.011	0.001	0.161	0.196
Same, L2 N676+ ablated: brackets still open	–	0.002	0.008	0.015	0.024

Table 4. The probability of one more closer after the model's chain of most probable closers, while brackets are still open and once all are closed, with and without L2 N676+.

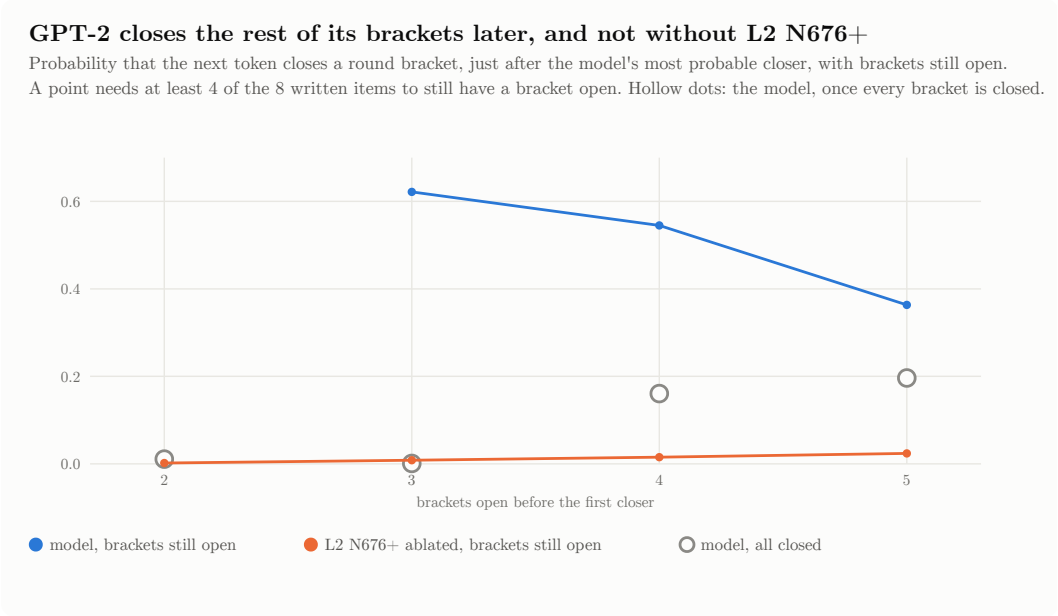

Figure 7. The probability of one more closer against the number of brackets first opened. With brackets still open, the model gives 0.62 at depth 3, 0.55 at 4 and 0.36 at 5. Once all are closed, it gives 0.001 at depth 3 and about 0.2 at depths 4 and 5. With L2 N676+ ablated, it gives 0.02 or less at every depth.

The model does close the rest later. When its first closer leaves brackets open, it gives 0.36 to 0.62 to one more closer. At depth 3, its most probable closer is `))` in all 8 items. It then gives 0.62 to one more closer, and 0.001 after it closes the last bracket. With L2 N676+ ablated, the model closes one bracket and stops. Its probability of one more closer falls to 0.024 or less, with up to 4 brackets still open. At depths 2 to 5, each greedy line that ends closes too few, 31 lines of 31. So the pole carries more than the size of the first closer. Without it, the model does not act on the brackets that stay open.

At depths 4 and 5, the model also gives 0.16 and 0.20 to one more closer after it closes every bracket. This extra closer is mostly our method's. Our chain adds the most probable closer at every step, even where the model would write something else. At depth 4, it

forced a closer in 4 of 7 items, where the model gave closers 0.005 or less. In the model's own chains, the chance of one more closer is 0.02 or less at depths 2 to 5 (Table 16).

6 Do the explanations survive adversarial review?

A sweep and a set of written probes chosen by their authors are weak tests of a claim. We therefore audited L2 N676+ through the registry's review, as a chosen unit (`selection = manual`), on a new sample of code. The proposer (Claude Opus 5.5) read the scan and wrote a claim. It says that the pole tracks how many round brackets are open, rises with that count, and is near zero outside them. The reviewer (Claude Sonnet 5.5) wrote 20 probes, and the judges were Claude Haiku 5.5 (Table 5).

Gate or score	Result
Positive holdouts	8 of 8
Hard negatives	7 of 12
Counterfactuals	8 of 8
Specificity	0.58, failed
Sufficiency, necessity	0.97, 1.00
Detection, fuzzing	0.98, 0.92
Verdict	NEEDS_REVISION (0.79)

Table 5. The first review of L2 N676+, on a claim of round brackets alone: each gate or score, and the verdict with its balanced accuracy.

The protocol rejects a claim of round brackets alone. The hypothesis passes every probe, counterfactual and causal gate, with detection 0.98 and fuzzing 0.92. It fails specificity (0.58), because the pole fires under square and curly brackets with no round bracket open. The verdict is NEEDS_REVISION. A revised claim, of round and square brackets strong and curly and angle brackets weak, is VERIFIED (1.00) on its second review (Table 17).

The other four poles. We audited L2 N1195+, L3 N695+, L8 N689+ and L9 N340+ in the same way, as chosen units (Table 6). A proposer wrote each claim from the scan, and could read the probes of Table 12. The proposers, reviewers and judges were the same models as for L2 N676+.

Pole	Claim, in short	Specificity	Detection, fuzzing	Verdict
L2 N1195+	Argument tokens just after open round or square brackets, rising with the number opened in the last ~10 tokens	0.92	0.94, 0.96	VERIFIED (0.96)
L3 N695+	Separators (commas, line breaks, colons) inside open brackets, rising with depth and fading within ~10 tokens	0.85	0.72, 0.79	VERIFIED (0.92)
L8 N689+	Code tokens inside an open round bracket, strongest at depth 2 or 3 on the token that ends the innermost argument	0.85	0.78, 0.69	VERIFIED (0.92)
L9 N340+	Inside an open round bracket where the bracketed text could end, so a lookahead to)	0.79	0.96, 0.67	NEEDS_REVISION (0.89)

Table 6. The first reviews of the other four round-depth poles, with each claim in short.

Each claim passed all of its positive holdouts and counterfactuals, and sufficiency and necessity were 0.97 or more. The L2 poles and L3 N695+ count, and L8 N689+ and L9 N340+ tell inside from outside. L9 N340+ alone ignores square brackets. Through the protocol, L2 N1195+, L3 N695+ and L8 N689+ are VERIFIED. L9 N340+ first earned NEEDS_REVISION, because it also fires on function words inside an open round bracket. Its revised, retrospective claim, that it fires inside an open round bracket, is VERIFIED (Table 18).

7 Where does the model read the pole?

With L2 N676+ ablated, the model stops after one closer. We looked for what carries the pole's signal to that decision, on the items of Table 4 at depths 3 to 5 with brackets still open. The measure is the log-odds of a closer as the next token. The pole is near zero on the closer itself (0.8 to 1.1), and it reads 3 to 12.5 on every token inside the brackets. Ablated before the closer only, it removes most of the effect (Table 19). So the model reads the pole from the tokens inside the brackets, through attention, and not at the closer.

We then patched one component at a time: each of the 120 heads of layers 2 to 11, and each MLP of layers 3 to 11 (Table 20). L3H5 carries the most. Patched alone, it carries 27% to 63% of the effect. Nothing changes the residual stream between the MLP of L2 and the heads of L3, so the noising share of L3H5 is a direct path from the pole. From the closer, L3H5 gives 0.31 to 0.40 of its attention to the open brackets, and most of the rest to the tokens inside them. So it moves the signal that brackets are open to the closer, where the pole itself is silent.

From L3H5, the signal goes through heads of L4, and through L5H3 and L7H4, to the MLP of L8 (Table 21). Attribution patching (Syed et al., 2023) then ranks the neurons of that MLP, and exact patching measures the best of them (Table 22). Most of what L3H5 sends to the MLP of L8 reaches one neuron, L8 N689+, which alone carries 34% to 59% of it. That is one of the round-depth poles, VERIFIED as firing on code tokens inside an open round bracket. Later layers thus read L2 N676+ through L3H5, and then L8 N689+. Ablating L8 N689+ alone changes the count of brackets closed by at most 0.1 (Table 3), so other paths can stand in for it.

Two further results are in the Appendices. L3H5 itself fails the review as a unit, because its strongest outputs are `All` after a `Copyright` token (Table 23). And early heads (L0H10, L1H3, L2H10 and L3H10) read the closers, and take part in an exact count up to depth 6. The model counts its rare packed tokens, `)))` and `))))`, less sharply (Tables 24 to 29).

8 Do the Qwen2.5 models hold such a pole?

A pole like L2 N676+ might be particular to GPT-2. We therefore repeated the sweep and the sweep by bracket kind on Qwen2.5-0.5B and Qwen2.5-1.5B (Yang et al., 2024), on the same 500 documents. Their MLPs are gated, and each reads whole documents (Appendices). Table 7 gives the best pole of each model for each label set.

Label set	GPT-2 small	R^2	Qwen2.5-0.5B	R^2	Qwen2.5-1.5B	R^2
`syntax node`	L1 N2566+	0.605	L15 N1415+	0.529	L21 N6306+	0.554
`syntax construct`	L2 N676+	0.273	L8 N3135+	0.466	L11 N1460+	0.554
`bracket depth`	L4 N3038−	0.231	L12 N1468+	0.210	L22 N2957+	0.256
`indent depth`	L6 N1416−	0.184	L4 N4639−	0.158	L11 N2261−	0.179
`round depth`	L2 N676+	0.277	L21 N953−	0.160	L26 N7101+	0.175
round brackets with no parser	L2 N676+	0.310	L21 N953−	0.172	L26 N7101+	0.198
square depth	L10 N1339+	0.094	L22 N1943+	0.185	L25 N8793−	0.237
curly depth	L4 N3038−	0.176	L0 N2546+	0.158	L22 N2957+	0.258

Table 7. The best pole of all layers for each label set, and its held-out R^2, in GPT-2 small and the two Qwen2.5 models. GPT-2 reads 128-token windows, and Qwen reads whole documents.

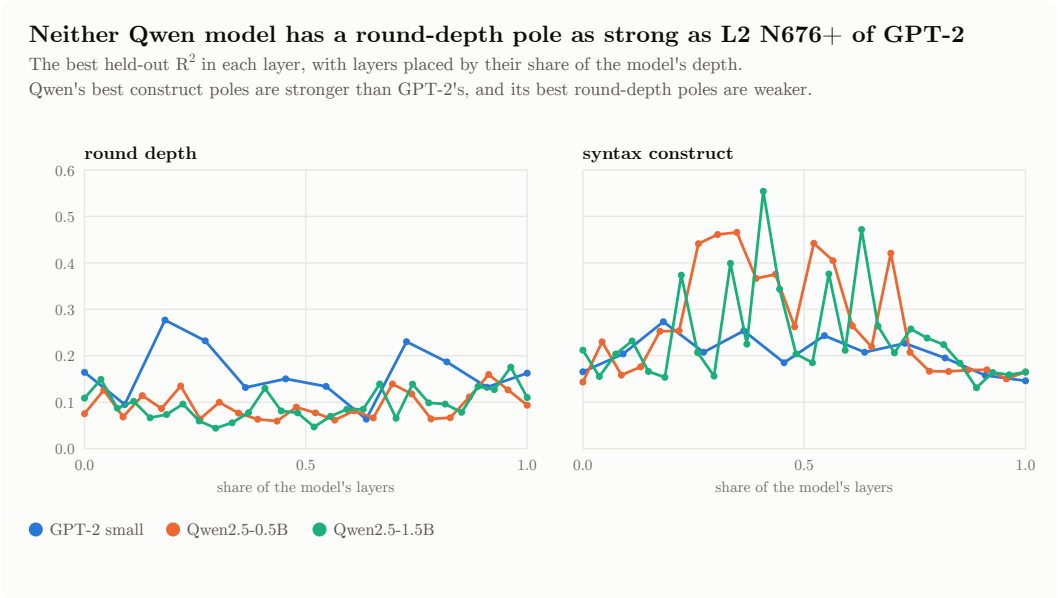

Figure 8. The best R^2 per layer against relative depth, one line per model. For round depth, GPT-2 peaks at 0.28 in layer 2, and the Qwen2.5 models stay at or under 0.16 and 0.18. For `syntax construct`, Qwen2.5-0.5B reaches 0.47, Qwen2.5-1.5B 0.55 and GPT-2 0.27.

Qwen2.5 has no such pole. The same sweep finds no pole of Qwen2.5-0.5B or 1.5B above an R^2 of 0.175 for round depth, against 0.277 for L2 N676+ (Figure 8). The best of each is in a late layer, and no layer stands out. Their best `syntax construct` poles are stronger than GPT-2's, at 0.466 and 0.554 against 0.273. GPT-2 and Qwen read different windows, so their R^2 are not directly comparable. At GPT-2's 128-token windows, though, Qwen still has no such pole (Table 30). Each best pole for round depth stays the same, with R^2 0.168 and 0.161.

Qwen's best round-depth poles do not count. On written text, L21 N953− of Qwen2.5-0.5B fires inside round brackets without rising with depth, and L26 N7101+ of Qwen2.5-1.5B fires outside them (Table 31). Neither changes the probability of a closer when ablated. Through the protocol, L21 N953− earns VERIFIED as a sign of an open round

bracket, on a revised, retrospective claim (Tables 32 and 33). L9 N340+ of GPT-2 followed the same course.

9 Discussion

A counter of characters, with partners. The results support a narrow reading of L2 N676+. On held-out code and on written text, it rises with the number of open round and square brackets, and it falls to about zero once they close. It is not a parser's count, and it fades over about 20 list items. It is a short-range, imperfect counter of bracket characters. The model uses it: without it, the model closes one bracket and then gives 0.024 or less to one more closer. Four more round-depth poles pull the other way, and L8 N689+ sits on the path from L2 N676+ to the closer. The count is therefore shared among several units, which fits the distributed signal that Clark et al. (2023) describe.

Counting through attention. Much of the count runs through attention heads. L3H5 moves the signal from the tokens inside the brackets to the closer, and four early heads read the closers. In RASP, attention is the only way to move information between positions (Weiss et al., 2021). A count of open brackets at one token must therefore gather earlier tokens through attention. Our result fits that constraint, and adds that a trained model also keeps a running count in an MLP neuron, which later heads read. This second part is our observation, not a claim of RASP.

Backup paths. Several results show that the model can partly replace a missing unit. Ablating L8 N689+ alone changes the count by at most 0.1, although it carries 34% to 59% of what L3H5 sends to the MLP of L8. McDougall et al. (2023) warn that backup components "may counteract the effects of ablations". Our single-unit ablations may therefore understate what each unit carries.

The output side of Qwen's poles. Through the direct path, from the pole's output to the unembedding, L21 N953− writes toward a closer, but its ablation shows no effect. This method follows Geva et al. (2021), who read each value vector of a feed-forward layer through the output embedding. They note that this "assumes (naïvely) that all model's layers operate in the same embedding space". The direct path leaves out every later layer, so the ablation is the better test of what the pole does to the output.

Effect on the protocol. None. The labels are reported, not gated. A scan of a corpus that records programming languages has a `round depth` label set, because of this result. The registry's scan corpus has no such field, so its scans do not change. Cut documents now get searched endings, so the syntax labels of such scans change a little. The registry now holds VERIFIED records of L2 N676+, L2 N1195+, L3 N695+ and L8 N689+. It holds a VERIFIED revision of L9 N340+, which supersedes its NEEDS_REVISION record (Table 18). It holds a VERIFIED record of L21 N953− of Qwen2.5-0.5B, which supersedes a NEEDS_REVISION one (Table 33).

For the sweep as a method. The positive control passes, but only after we changed the label. A sweep that finds nothing with one label therefore says little about the concept until other forms of the label are tried. The review then adds a test that the sweep alone cannot give. It rejected a claim of round brackets alone, and the pole's true scope, round and square brackets, came from the failed hard negatives.

10 Limitations

- **Not the only signal, and not the computation.** These results do not show that L2 N676+ is the only bracket signal in GPT-2. They do not show how the model computes depth. The probes and ablations use narrow templates.
- **Few models, one sample for the sweep.** The sweep ran on GPT-2 small, Qwen2.5-0.5B and Qwen2.5-1.5B, on one sample of documents. The label stability and the audit used a second sample.
- **Selection.** We chose this pole from 73,728 poles, and we changed the label after seeing results. Held-out R^2 guards against noise in the fit, not against our choices. The written probes and the ablation are separate tests of the pole, but they came after we chose it.
- **Few probes.** Each cell has 8 items from one template family, in Python-like or C-like text. Standard errors cover the items, not the choice of template.
- **One path of closers.** The chains follow the model's most probable closer at each step, and greedy text follows its most probable token. Other closers and other text can lead elsewhere. The chain also adds a closer where the model would not (Table 16).
- **No confidence interval on ablation.** The 20 controls show the size of a random pole's effect, and no more.
- R^2 **is a weak search.** A pole that responds to a label only sometimes scores low, and the depth-tracking poles here score under 0.32. A sweep can miss signals, including any spread over several neurons.
- **The label is approximate.** We sort node names by pattern, and angle brackets do not count.

11 Conclusion

Parser labels on source code do find a bracket-depth neuron in GPT-2 small, but only when the label counts round brackets alone. L2 N676+ is a short-range, imperfect counter of bracket characters. The model uses it to decide how many brackets to close and whether to close the rest later, and later layers read it through L3H5 and L8 N689+. Its explanation passes the registry's review once it names round and square brackets. Qwen2.5-0.5B and 1.5B hold no pole as strong. One test would sharpen the result:

- Test whether the count of `)))` and `))))` is less sharp because the tokens are rare. A model trained on code with more of them would show it.

References

Clark, T., Roman, A., & Thurnherr, H. (2023). *Who cares about brackets?* Research report, Apart Research Alignment Jam #10 (Interpretability 3.0). PDF.

Friedman, D., Lampinen, A., Dixon, L., Chen, D., & Ghandeharioun, A. (2024). *Interpretability illusions in the generalization of simplified models.* ICML 2024. arXiv:2312.03656.

Geva, M., Schuster, R., Berant, J., & Levy, O. (2021). *Transformer Feed-Forward Layers Are Key-Value Memories.* EMNLP 2021. arXiv:2012.14913.

Gurnee, W., Horsley, T., Guo, Z. C., Kheirkhah, T. R., Sun, Q., Hathaway, W., Nanda, N., & Bertsimas, D. (2024). *Universal neurons in GPT2 language models.* arXiv:2401.12181.

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). *Visualizing and understanding recurrent networks.* arXiv:1506.02078.

McDougall, C., Conmy, A., Rushing, C., McGrath, T., & Nanda, N. (2023). *Copy Suppression: Comprehensively Understanding an Attention Head*. Preprint. arXiv:2310.04625.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners*. OpenAI.

Syed, A., Rager, C., & Conmy, A. (2023). *Attribution patching outperforms automated circuit discovery*. arXiv:2310.10348.

Tigges, C., Kowal, M., McGrath, T., Lewis, O., & Cammarata, N. (2026). *Neurons as an interpretable basis*. Goodfire Research. goodfire.com.

Weiss, G., Goldberg, Y., & Yahav, E. (2021). *Thinking like transformers*. ICML 2021. arXiv:2106.06981.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., . . . Qiu, Z. (2024). *Qwen2.5 technical report*. arXiv:2412.15115.

Appendices

Reproduce

The scripts are `scripts/syntax_controls.py`, `scripts/syntax_probes.py`, `scripts/syntax_followups.py`, `scripts/syntax_close_paths.py`, `scripts/syntax_close_states.py`, `scripts/syntax_close_readers.py`, `scripts/syntax_close_match.py` and `scripts/figures_syntax.py`. The tagger is `openinterpretability/syntax.py`. The output is in `results.jsonl`, which keeps no text from the corpus, because each file keeps its repository's licence.

On an Apple M-series MPS backend, the sweep takes about 40 minutes for GPT-2, 3 hours for Qwen2.5-0.5B and 4.5 hours for Qwen2.5-1.5B. The sweep by bracket kind takes 15 minutes, 1 hour and 3 hours. CPU is slower. The other commands take a few minutes. `results.jsonl` stores the outputs for GPT-2, and `results-qwen2.5-0.5b.jsonl` and `results-qwen2.5-1.5b.jsonl` those for Qwen. The sweep by bracket kind at 128-token windows takes 45 minutes for Qwen2.5-0.5B and 1.5 hours for Qwen2.5-1.5B, and `results.window128-*.jsonl` stores it. `results.followups.jsonl` stores the written follow-ups, which run on CPU in about 30 minutes. Parser and grammar versions are in `uv.lock`.

```
uv run open-interpretability corpus --dataset github-code --docs 500 --seed 0 --out
code.jsonl
uv run open-interpretability corpus --dataset github-code --docs 500 --seed 0 --max-
chars 1000000 --out code-whole.jsonl
uv run open-interpretability corpus --dataset github-code --docs 500 --seed 1 --max-
chars 1000000 --out code-seed1-whole.jsonl
P=papers/syntax-positive-controls
uv run python scripts/syntax_controls.py sweep --device mps --corpus code.jsonl --out
$P/results.jsonl
uv run python scripts/syntax_controls.py kinds --device mps --corpus code.jsonl --out
$P/results.jsonl
uv run python scripts/syntax_controls.py stability --whole code-whole.jsonl --whole
code-seed1-whole.jsonl --out $P/results.jsonl
uv run python scripts/syntax_probes.py --results $P/results.jsonl
uv run python scripts/syntax_controls.py sweep --model Qwen/Qwen2.5-0.5B --device mps
--corpus code.jsonl --out $P/results-qwen2.5-0.5b.jsonl
uv run python scripts/syntax_controls.py kinds --model Qwen/Qwen2.5-0.5B --device mps
--corpus code.jsonl --out $P/results-qwen2.5-0.5b.jsonl
uv run python scripts/syntax_controls.py sweep --model Qwen/Qwen2.5-1.5B --device mps
--corpus code.jsonl --out $P/results-qwen2.5-1.5b.jsonl
uv run python scripts/syntax_controls.py kinds --model Qwen/Qwen2.5-1.5B --device mps
--corpus code.jsonl --out $P/results-qwen2.5-1.5b.jsonl
uv run python scripts/syntax_controls.py kinds --model Qwen/Qwen2.5-0.5B --device mps
--window 128 --corpus code.jsonl --out $P/results.window128-qwen2.5-0.5b.jsonl
uv run python scripts/syntax_controls.py kinds --model Qwen/Qwen2.5-1.5B --device mps
--window 128 --corpus code.jsonl --out $P/results.window128-qwen2.5-1.5b.jsonl
uv run python scripts/syntax_followups.py --out $P/results.followups.jsonl
uv run python scripts/syntax_close_paths.py --results $P/results.followups.jsonl
uv run python scripts/syntax_close_states.py --results $P/results.followups.jsonl
uv run python scripts/syntax_close_readers.py --results $P/results.followups.jsonl
uv run python scripts/syntax_close_match.py --results $P/results.followups.jsonl
uv run python scripts/paper_figures.py
```

The SHA-256 of `code.jsonl` is `549de3ddf5ee7d13c54dd32241e213745f6a21c8beab3d00e14e4b1279111a31`, and of `code-whole.jsonl` `13950d3fd2fe8e5cd16ee66c585b0660f6b9ea10ef075d734d97055a3f65f222`.

The first label's winners

Table 8 gives the best pole of each of four label sets, with what the pole does on the corpus. Token R^2 is in-sample and has many more levels than a label set, so it is a generous comparison. The first three poles fail it, and the fourth passes narrowly.

Label set, best pole overall	**Held-out R^2**	**Token R^2**	**What the pole does**
`syntax node`, L1 N2566+	0.605	0.781	`.` and `;` in Java imports. A token detector
`bracket depth`, L4 N3038−	0.231	0.509	Depth 0 against the rest. Flat from depth 1 (0.122, 0.133, 0.136, 0.137)
`indent depth`, L6 N1416−	0.184	0.322	Peaks in licence comments and imports. Also wins `syntax construct` at L6
`syntax construct`, L2 N676+	0.273	0.258	Call arguments 1.85 and parameters 1.85, against 0.46 or less elsewhere

Table 8. The best pole of all 12 layers for four label sets, with its held-out R^2 and its in-sample token R^2. The last column says what the pole does on the corpus.

To remove the licence header, we repeated the depth sets without comments and strings. To ask for depth itself, not inside against outside, we also dropped depth 0. The best R^2 across layers falls, and the winners change (Table 9).

Label set	All tokens	No comments or strings	Also no depth 0
`bracket depth`	0.231, L4 N3038−	0.196, L9 N2645+	0.080, L5 N1743+
`indent depth`	0.184, L6 N1416−	0.261, L9 N2645+	0.120, L4 N1330+

Table 9. The best R^2 across layers, and its pole, for the two depth label sets on all tokens, without comments and strings, and also without depth 0.

The written probes of Table 2 then tested these winners (Table 10). None rises with bracket depth.

Pole	Open, depths 1 to 5	Closed, depths 1 to 5
L4 N3038−	0.15, 0.15, 0.16, 0.15, 0.16	0.15, 0.15, 0.16, 0.15, 0.16
L6 N1416−	0.00 at each depth	0.00 at each depth
L5 N1743+	0.19, 0.12, 0.06, 0.03, 0.01	0.30, 0.28, 0.29, 0.29, 0.25
L4 N1330+	0.00, 0.01, 0.02, 0.03, 0.03	0.04, 0.07, 0.07, 0.09, 0.15

Table 10. The mean strength of the first label's winners on the written probes, under open and closed brackets at depths 1 to 5.

L5 N1743+ does respond to open brackets. It falls from 0.19 to 0.01, and it stays near 0.29 when they close. So it may carry the opposite signal, at a strength of at most 0.30, where L2 N676+ reaches 11. These probes test only bracket depth, so they say nothing about L4 N1330+ and indentation. Of the two poles that won the narrower depth tests, one does not rise with depth and the other falls, at a strength of at most 0.30.

Limits of the counter on written text

Four things in the written probes of Table 2 limit the reading of L2 N676+ as a counter. First, it does not fully count down. A pole that counts would read the same under `One closed` as under `Open`, but it reads lower, by 2.5 to 3.0. The tokens just before the target differ too, so this comparison is rough. Second, strings and comments do not stop it. A parser says that these brackets are not brackets. The pole reads them almost as it reads code, and its R^2 is higher for characters than for parsed brackets. So it follows characters and not the grammar.

Third, lines matter. With each call on its own line, the strength peaks at depth 2 and then falls. The comment row still reads high after one line break, so one newline is not a reset. Fourth, curly brackets differ between tests. The pole responds to `load{`, but its corpus R^2 for curly depth is zero. Probes with real blocks explain this. No round-depth pole responds to a curly bracket that opens a block, or to indentation, and four of them respond to an object literal (Table 14). Most curly brackets in code hold blocks, which fits the corpus R^2 of zero.

Table 11 gives the strength with one bracket and a list of filler items between it and the target. This range is far shorter than the LSTM's: Karpathy et al. (2015) show a quote

cell that holds for about 230 characters. Their cell and this pole measure different things on different data, so the comparison is only a rough one.

Filler items	0	5	10	20	40	80	160
Open	4.23	2.20	0.56	0.00	0.00	0.00	0.00
Closed	0.00	0.01	0.00	0.00	0.00	0.00	0.00

Table 11. The mean strength of L2 N676+ with one bracket, open or closed, and a list of filler items between it and the target.

Other round-depth poles on written text

The round R^2 of L2 N676+ is the highest of any layer's best pole. The nearest are L3 N695+ (0.232) and L8 N689+ (0.230). L2 N1195+ (0.236) and the other pole of the same neuron, L2 N676− (0.225), are next in its layer. L9 N340+ (0.187) is first in L9. In all, the `round depth` sweep ranks four more poles near L2 N676+: L2 N1195+ (0.236), L3 N695+ (0.232), L8 N689+ (0.230) and L9 N340+ (0.187). The same written probes test them (Table 12). Each cell is a mean over 8 items.

Pole	Open, depths 1 to 5	Closed, depths 1 to 5	Square	Curly
L2 N676+	4.19, 6.75, 8.63, 9.99, 11.02	0.00, 0.00, 0.12, 0.54, 0.00	3.51 to 9.02	1.49 to 4.98
L2 N1195+	3.43, 5.88, 7.68, 8.97, 9.94	0.00, 0.00, 0.00, 0.10, 0.00	2.79 to 7.96	1.04 to 4.44
L3 N695+	1.15, 1.80, 2.21, 2.53, 2.83	0.00, 0.00, 0.00, 0.03, 0.00	1.22 to 3.02	1.08 to 2.46
L8 N689+	2.29, 3.55, 3.57, 2.94, 2.31	0.00 at each depth	2.74 to 3.38	0.40 to 1.01
L9 N340+	2.35, 3.09, 3.11, 2.74, 2.38	0.00 at each depth	0.11 to 0.00	0.00 at each depth

Table 12. The mean strength of the five round-depth poles on the written probes. The square and curly columns give the range over depths 1 to 5.

The two L2 poles and L3 N695+ rise at every depth, and so they count. L8 N689+ and L9 N340+ peak at depth 2 or 3 and then fall. So they tell inside from outside, and they do not count. L9 N340+ is the only one of the five that ignores square brackets, and it also ignores curly ones. All five read near zero once the brackets close. The four other poles' claims in Table 6 agree with these probes: L9 N340+ alone ignores square brackets, and L2 N1195+ and L3 N695+ count. L2 N1195+ responds to depth almost as L2 N676+ does. L3 N695+ responds more weakly, and L8 N689+ and L9 N340+ peak at depth 2 or 3 and reach further.

The later poles reach further (Table 13). The rows have one open bracket and a list of filler items before the target.

Filler items	0	5	10	20	40	80
L2 N676+	4.23	2.20	0.56	0.00	0.00	0.00
L2 N1195+	3.38	1.20	0.01	0.00	0.00	0.00
L3 N695+	1.21	0.34	0.03	0.00	0.00	0.00
L8 N689+	2.31	2.46	2.16	1.03	0.12	0.02
L9 N340+	2.32	1.83	1.21	0.52	0.02	0.00

Table 13. The mean strength of the five round-depth poles with one open bracket and a list of filler items between it and the target.

Curly brackets that hold blocks

The pole responds to `load{first, value`, but its corpus R^2 for curly depth is zero. We guessed that this is because curly brackets in real code mostly hold blocks. So we wrote probes with real blocks (Table 14). Each target is the token after `x = first +`. Each block header holds a round bracket that is already closed, such as `if (ready) {`. Each cell is a mean over 8 items, and the standard errors are at most 0.18.

Pole	Round, `x = load(parse(`	Object literal, `x = {a: {b:`	Blocks, each on its own line	Blocks on one line	Python indentation
L2 N676+	3.57 to 10.13	0.54 to 3.21	0.00	0.02 or less	0.00
L2 N1195+	2.91 to 9.14	0.24 to 2.76	0.00	0.00	0.00
L3 N695+	0.92 to 2.71	0.43 to 1.73	0.01 or less	0.07 or less	0.02 or less
L8 N689+	1.82 to 3.14	0.01 to 0.39	0.00	0.00	0.00
L9 N340+	1.96 to 3.02	0.00	0.00	0.00	0.00
L4 N3038−	0.15 to 0.16	0.15	0.14 to 0.15	0.14 to 0.15	0.13 to 0.14

Table 14. The mean strength of the round-depth poles and of L4 N3038− under round brackets, object literals, code blocks and Python indentation. Ranges run over depths 1 to 5.

No round-depth pole responds to a curly bracket that opens a block, on one line or on several, and none responds to indentation. Four of them respond to curly brackets that hold an object literal, and rise with its depth, more weakly than with round brackets. L9 N340+ responds to neither. So the guess holds: the poles respond to curly brackets that hold a list of items, and a block holds statements. This fits the revised claim of L2 N676+, which calls curly brackets weak. L4 N3038−, the best pole for curly depth, reads about 0.15 in every condition, so it does not track blocks either. Curly brackets that hold blocks therefore do not count.

Ablation by depth

Table 15 gives the probability of a closing bracket at each depth, with and without L2 N676+. The pole lowers the mean probability of a closing bracket at depths 1 to 4. This measure counts every token that starts with `)`, so it cannot tell `)` from `))`. Table 3 measures that, and it explains why the effect shrinks with depth. The model's closers at depth 5 are mostly `))` and `)))`, and with the pole ablated they become `)`. Both count as a closer, so a measure of any closer cannot see the change.

Open brackets	1	2	3	4	5
Model	0.457	0.464	0.453	0.387	0.309
L2 N676+ ablated	0.199	0.255	0.263	0.297	0.324
Mean of 20 random poles ablated	0.458	0.464	0.453	0.387	0.309
Lowest of those 20	0.455	0.462	0.450	0.384	0.306

Table 15. The probability that the next token starts with `)`, by the number of open brackets, with and without the pole. The controls are 20 random positive poles of L2.

Chains and greedy text

Greedy text agrees. We let the model write its most probable tokens to the end of the line, for up to 16 tokens. Lines that end close every bracket at depths 1 to 3 (8, 8 and 5 lines of 8). At depth 4, 1 of 5 lines closes every bracket, 3 close too many and 1 too few. At depth 5, none of 3 does. The other lines repeat an argument, such as `, first, first, first`, and never end.

Our chain adds the most probable closer at every step, even where the model would write something else. At depth 4, after `))`, the model's most probable closer is `),`, which closes one bracket and starts a new argument. Then it wants the new argument: it gives closers 0.003 to 0.005 in total. The chain still adds `)`. We call a chain forced if one of its steps gave closers less than 0.05 (Table 16).

Depth	Natural chains	P(one more closer) after them	Forced chains	P(one more closer) after them
2	8	0.001 to 0.017	0	–
3	8	0.002 or less	0	–
4	3	0.001 to 0.009	4	0.18 to 0.39
5	6	0.003 to 0.008, and 0.43 and 0.51	1	0.42

Table 16. Chains of closers split into natural chains and forced chains, in which one step gave closers less than 0.05. For each, the table gives the number of chains and the probability of one more closer once every bracket is closed.

At depth 4, the chain forced a closer in 4 of 7 items, where the model gave closers 0.005 or less. Every natural chain at depth 4 closes cleanly, and only the forced items keep a high chance of one more closer. In the model's own chains, that chance is 0.02 or less at depths 2 to 5. At depth 5, the chains that end with `);` close cleanly too. The one exception is a chain at depth 5 that closes with `)))` and then `))`: two natural chains of this form give 0.43 and 0.51 to one more closer. They end with `))`, and they hold the rare packed token `)))`. The model counts its rare packed tokens, `)))` and `))))`, less sharply (Table 29).

The reviews of L2 N676+

In the first review (Table 5), each of the five failed hard negatives has square or curly brackets and no open round bracket, for example `data[3]` and `{"x": 4}`. They read 2.95 to 4.15, above the activity line of 2.59. The written probes show the same thing: square brackets nearly match round ones.

The reviewer's first probe set had four hard negatives with a token inside an open round bracket, which the claim says is active. We sent those four back to the same reviewer, which replaced them, and the record holds that second review. The record is in `wiki/experiments/openai-community--gpt2/L2/N676-positive/`, added by pull request 2367.

The revision. The revision scanned the same corpus with the same windows. The proposer (Claude Opus 5.5) tested bracket kinds on written text before it wrote the claim. Square brackets nearly match round ones: `(red, green, blue)` reads 4.56 and `[red, green,`

`blue]` reads 3.83. Curly brackets are weaker than the probes of Table 12 suggest: `{red, green, blue}` reads 1.07, and a code block `{ return ...; }` reads 0. In the failed JSON probe of the first record, the tokens inside `{...}` read 0.0 to 0.9, and its 4.15 came from the inner `[1, 2, 3]`. So the claim says that open round and square brackets add to the count. Curly and angle brackets add only a little. It also says that the strength falls just after an inner bracket closes, even when an outer bracket is still open.

Gate or score	First review	Second review
Positive holdouts	7 of 7	7 of 7
Hard negatives	12 of 13	13 of 13
Counterfactuals	9 of 9	9 of 9
Specificity	0.98	1.00
Enough hard negatives	failed	passed
Sufficiency, necessity	1.00, 0.97	1.00, 0.97
Detection, fuzzing	0.68, 0.65	0.68, 0.65
Verdict	NEEDS_REVISION (0.99)	VERIFIED (1.00)

Table 17. Two reviews of the revised claim of L2 N676+, of round and square brackets strong and curly and angle brackets weak.

Table 17 gives both reviews of the revised claim. In the first, the one failed probe, `{{{ name, data, x, alpha, form, int, ex }}}`, read 3.10 against an activity line of 2.59: three nested curly brackets are not weak. A second reviewer of the same model wrote a new set with more varied hard negatives, and all of them passed. Both records stay in the registry. The judges marked fewer held-out windows active than the old claim's judges did. So detection fell from 0.98 to 0.68, and fuzzing from 0.92 to 0.65. The records were added by pull request 2507.

Notes on the reviews of the four poles

The first reviews of Table 6 left three points worth recording.

- **L3 N695**+ fades more slowly than its claim says. Against a line of 1.04, a comma after `x = f(g(1)), y = h(2)` read 2.07. A comma at the end of a long expression in an open bracket read 1.32.
- **L8 N689**+ failed two hard negatives at 2.51 and 2.34, against a line of 2.31. The reviewer labelled the first, `gulp.dest('www/static/css'); ...`, inactive, but its path tokens are inside an open round bracket, which the claim calls active. That label makes the review stricter, so the verdict stands.
- **L9 N340**+ fails specificity and the lookahead check. It reads 2.00 on `(although the` and 2.27 on `(and then`, where the claim calls function words weak, and 3.47 on a JavaScript object literal. The next claim should drop the lookahead and say that the pole fires inside an open round bracket.

The records were added by pull requests 2508, 2509, 2510 and 2511.

The revision of L9 N340+. The revision scanned the same corpus with the same windows. The proposer (Claude Opus 5.5) read the first record, and tested its wording on the model. Its claim drops the lookahead. The pole is a retrospective sign of an open round bracket, from the `(` to the token before its `)`, in code, comments, strings and prose. The `(` and a function word just after it are moderate. Content words, names, numbers and

commas inside are strongest. Articles and prepositions in the middle of a phrase are weak. Curly brackets and quotes do not drive it, and square brackets give a weaker response that varies. The proposer found that the earlier failures fit this claim. `(although the` peaks at 2.00 on the `(` itself, and the JavaScript probe peaks at 3.53 on `opts` inside `console.log(`. The object literal reads 0.00. Table 18 compares the two reviews.

Gate or score	First claim	Revised claim
Hard negatives	11 of 14	12 of 13
Positive holdouts	6 of 6	7 of 7
Counterfactuals	8 of 8	7 of 7
Detection, fuzzing	0.96, 0.67	0.92, 0.98
Verdict	NEEDS_REVISION (0.89)	VERIFIED (0.96)

Table 18. The reviews of the first claim of L9 N340+, a lookahead to `)`, and of its revised, retrospective claim.

The reviewer (Claude Sonnet 5.5) first wrote four hard negatives with an open round bracket, which the claim calls active. We sent them back, and the same reviewer replaced them. Pull request 2756 added the record.

The path from L2 N676+ to the closer

The items are those of Table 4 at depths 3 to 5, each ending with the model's first closer, with brackets still open (Table 19).

The pole is near zero on the closer itself (0.8 to 1.1), and it reads 3 to 12.5 on every token inside the brackets. Ablating it at the closer alone changes little. Ablating it before the closer alone removes most of the effect. So the model reads the pole from the tokens inside the brackets, through attention, and not at the closer.

Open brackets	3	4	5
Model	0.50	0.18	−0.60
Pole ablated at every token	−4.70	−4.41	−4.41
Pole ablated at the closer only	0.50	0.10	−0.47
Pole ablated before the closer only	−4.68	−3.56	−2.96

Table 19. The log-odds of one more closer after the model's first closer, with brackets still open, when L2 N676+ is ablated at every token, at the closer only or before it only.

Table 20 gives the shares of the effect of L2 N676+ that single components carry.

Component	Noising, depths 3, 4, 5	Denoising, depths 3, 4, 5
L3H5	0.63, 0.27, 0.30	0.21, 0.64, 0.61
L3H10	0.23, 0.16, 0.07	0.08, 0.08, 0.06
L7H4	0.04, 0.09, 0.06	0.23, 0.48, 0.50
L6H3	0.04, 0.05, 0.03	0.18, 0.24, 0.31
L5H3	−0.01, 0.01, −0.02	0.08, 0.23, 0.37
L8 MLP	0.17, 0.15, 0.09	0.42, 0.31, 0.26
L3 MLP	−0.05, 0.05, 0.35	−0.01, −0.21, −0.52

Table 20. The shares of the effect of L2 N676+ that single components carry, by noising and by denoising, at depths 3, 4 and 5.

What reads L3H5. We noised L3H5 alone, and held one later component at its clean output at the same time. The share is the part of the drop that holding the component removes (Table 21).

Component	Share, depths 3, 4, 5
L4H8	0.15, 0.37, 0.74
L4H6	0.13, 0.25, 0.45
L4H2	0.23, 0.25, under 0.17
L5H3	0.15, 0.26, 0.39
L7H4	0.36, 0.26, 0.41
L8 MLP	0.60, 0.39, 0.23

Table 21. The share of the effect of L3H5 that each later component removes when it is held at its clean output, with L3H5 noised.

The shares add up to more than 1, so the paths overlap. The signal goes from L3H5 through heads of L4, and through L5H3 and L7H4, to the MLP of L8. Which step matters most changes with depth: the MLP of L8 at depth 3, and L4H8 at depth 5. To find which neurons of that MLP carry it, we used attribution patching (Syed et al., 2023). It ranks each neuron by its gradient times its change, with L3H5 noised. We then held the top 10 neurons at their clean values one at a time, and measured each share exactly (Table 22).

Neuron of L8	Share, depths 3, 4, 5	Rank by attribution
N689	0.59, 0.34, 0.50	first at each depth
Next best of the top 10	0.04, 0.06, 0.05	–

Table 22. The share of what L3H5 sends to the MLP of L8 that one neuron carries, when it is held at its clean value with L3H5 noised.

So most of what L3H5 sends to the MLP of L8 reaches one neuron, L8 N689+, which alone carries 34% to 59% of it. The path runs from one round-depth pole to another.

L3H5 through the protocol

We audited L3H5 as a chosen unit, on the seed-1 sample of `github-code` at 128-token windows. A head's strength is the size of its output at a token. The proposer (Claude Opus 5.5) could read the section on where the model reads the pole. It found that the head's strongest outputs are not about brackets. All 20 top examples of the scan are `All` after a `Copyright` or © token, as in `Copyright 2013 Google Inc. All rights reserved.` There, the head moves its attention off the first token onto the Copyright token. Closers of nested calls reach only 0.3 to 0.5 of its maximum. So its claim put copyright notices at the top, and called the head a weak dial elsewhere.

The verdict is NEEDS_REVISION (0.77). The claim passed 7 of 7 positive holdouts and 7 of 7 counterfactuals, but only 7 of 13 hard negatives. Ordinary prose, a licence header and a copyright notice with no `All` read 5.0 to 7.0, above the activity line of 5.01. On held-out windows, 2,421 of 4,000 are active. Detection and fuzzing are 0.48 and 0.50, no better than chance, and token identity predicts the head better than the claim does. So L3H5 is active on much of the text, and the copyright peak is only its top. The open-bracket signal that it carries to the closer is one part of its middle range. Pull request 2770 added the record.

A revision read the same scan. Its proposer (Claude Opus 5.5) found that the head's strength grows as its attention leaves the first token of the text. Over 30,000 tokens, the two correlate at $r = -0.72$. So no token in the first 6 or so reaches the line. After that, token type decides: later clause words of English prose and licence text, a `.` or `?` that ends a sentence, `}`, `{`, `return` and line breaks reach it. Word pieces, indentation, nouns and some prepositions do not. The copyright `All` stays the peak. Table 23 compares the two claims.

Gate or score	First claim	Revised claim
Hard negatives	7 of 13	10 of 13
Specificity, balanced accuracy	0.54, 0.77	0.79, 0.90
Token identity, as a baseline	0.88	0.94
Detection, fuzzing	0.48, 0.50	0.54, 0.58
Verdict	NEEDS_REVISION (0.77)	NEEDS_REVISION (0.90)

Table 23. The reviews of two claims about the head L3H5: copyright notices at the top, and a strength that grows as the head's attention leaves the first token.

The revision is closer, but it still fails specificity, and token identity still predicts the head better. Its failed hard negatives read 5.4 to 6.5: a sentence of 5 tokens, Japanese text and a URL. Of the held-out windows, 2,384 of 4,000 are active, so the line falls in the middle of the head's usual range. A short claim is unlikely to place it well. Pull request 2774 added the record.

Closing too many

At depths 4 and 5, the chains of Table 4 give 0.16 and 0.20 to one more closer after they close every bracket. At depth 3 they give 0.001. Most of this comes from forced steps (Table 16), but this appendix studies the chains as they are. We read the round-depth poles at that last closer, and ablated each of them in turn. Table 24 gives the log-odds of one more closer.

Brackets first opened	1	2	3	4	5
Model	−7.39	−4.80	−7.10	−3.01	−3.24
L2 N676+ ablated	−7.61	−5.58	−8.28	−4.47	−5.65
L2 N1195+ ablated	−5.41	−0.04	−1.70	−0.28	−0.87
L3 N695+ ablated	−7.34	−4.57	−6.67	−2.20	−2.26
Strength of L8 N689+ at the last closer	0.00	0.00	0.00	0.24	0.71
Strength of L9 N340+ at the last closer	0.00	0.00	0.00	0.13	0.45

Table 24. The log-odds of one more closer once the chain closes every bracket, with single round-depth poles ablated, and the strength of the two later poles at the last closer.

The two L2 poles and L3 N695+ read 0.00 at the last closer at every depth. L2 N1195+ holds closers back: without it, one more closer has log-odds of −1.70 or more from depth 2. L2 N676+ pushes the other way, through the earlier tokens, and its push grows with depth. Ablating it lowers the log-odds by 1.18 at depth 3, 1.46 at depth 4 and 2.41 at depth 5. At depths 4 and 5, L8 N689+ and L9 N340+ still fire a little after the last closer. Ablating either lowers the log-odds by 0.6 or less. So no single pole explains the extra closer.

A push that grows with depth, against a brake that does not, could explain it. To test that, we scaled the strength of each pole at every token. We read the log-odds of a closer in two states (Table 25). With brackets still open, the model should close more. With all brackets closed, it should not.

Change	Still open, depths 3, 4, 5	All closed, depths 3, 4, 5
None	0.50, 0.18, −0.60	−7.10, −3.01, −3.24
Brake L2 N1195+ $\times 1.5$	−3.31, −0.88, −1.41	−8.24, −4.41, −6.04
Brake $\times 2$	−4.66, −4.26, −4.25	−8.29, −4.44, −5.65
Brake $\times 0.5$	0.40, −0.68, −1.93	−2.53, −0.83, −1.38
Push L2 N676+ $\times 0.5$	−3.42, −1.15, −1.46	−8.24, −4.41, −5.98
Push $\times 1.5$	0.42, −0.61, −1.82	−2.49, −0.84, −1.38

Table 25. The log-odds of a closer with brackets still open and with all closed, when the brake, L2 N1195+, or the push, L2 N676+, is scaled at every token.

A stronger brake removes the extra closer at depths 4 and 5. But it also stops the closer that the model needs while brackets are still open. A weaker push does the same. A stronger push or a weaker brake adds closers after all brackets close, and does not help while they are open. No scale in the table does better than the model in both states. So the brake is not too weak in a simple sense. The same signals serve both states.

Why the push survives closing. First, we closed every bracket in other ways: one `)` per token, `))` tokens, or the fewest tokens, such as one `))))`. Each way leaves one more closer at log-odds of −2.3 to −0.7 at depths 4 and 5. At depth 3, one `)` per token gives −2.83, where the model's own `))` then `);` gives −7.10. None of these ways ends with `);`, and *What reads the closers* below shows that this matters.

Then we read L3H5, which carries the push, in both states (Table 26). The script is `scripts/syntax_close_states.py`. The carried push is the drop in log-odds when L3H5 takes its output from the run with L2 N676+ ablated.

Depth	3	4	5
Push through L3H5, brackets still open	3.28	1.25	1.15
Push through L3H5, all closed	0.74	0.86	2.05
L3H5 attention to open brackets, all closed	0.38	0.49	0.52
L8 N689+, still open and all closed	2.35, 0.00	3.37, 0.24	2.99, 0.71

Table 26. The push that L3H5 carries with brackets still open and with all closed, its attention to the open brackets, and the strength of L8 N689+ in both states.

L3H5 gives only 0.01 to 0.035 of its attention to closers, in either state. After the brackets close, it attends to the open brackets more than before, and more with depth. So it carries the push whether or not the brackets are still open. At depth 5 it carries more after they close than before. L8 N689+, later on the same path, still tells the states apart, but less well with depth. So a part of the model between L3H5 and L8 N689+ reads the closers. At depths 4 and 5, it does not cancel the push.

What reads the closers

A head reads a token through that token's value vector. So we zeroed one head's values at the closers of the model's own chain, in the state where every bracket is closed. The head

then cannot read them. The script is `scripts/syntax_close_readers.py`. With every head of layers 0 to 8 blocked from the earlier closers, the state where all brackets are closed looks like an open one. At depth 3, the log-odds of one more closer rise from −7.10 to −1.90, and L8 N689+ from 0.00 to 3.50. At depths 4 and 5 it rises to 2.93 and 3.98. Four heads have the largest effect alone: L0H10, L1H3, L2H10 and L3H10. At depth 3, blocking L2H10 alone raises the log-odds to −3.62. These heads attend to closers more with depth: L2H10 gives them 0.20 of its attention at depth 2, and 0.38 at depth 5.

They take part in an exact count. We held the depth and the last token fixed. A first closer of 1 to 4 brackets came before one `)`, so that a set number of brackets stayed open (Table 27).

Depth, brackets left open	**L8 N689+**	**P(one more closer)**	**L8 N689+, the four readers blocked**
3, one	3.47	0.142	3.75
3, none	0.11	0.021	3.16
4, one	3.46	0.147	3.50
4, none	0.67	0.081	3.53
5, one	3.52	0.178	3.30
5, none	2.29	0.138	3.76

Table 27. The strength of L8 N689+ and the probability of one more closer with one bracket or none left open, and the strength of L8 N689+ with the four early readers of closers blocked.

Blocking the readers changes nothing while a bracket is open. It turns the signal back on only when the closers match the openers. In these items, the match is weaker with depth. We tested two causes, and both failed. First, the steps of the L2 depth code shrink with depth (+2.6, +1.9, +1.4 and +1.0 per bracket on the written probes). We mapped both L2 poles at every token to an even code. With no bracket open at depth 5, L8 N689+ read 2.43, against 2.29 without the map.

Second, the signal of a match reaches L8 N689+ mostly through L7H4. This head attends to open brackets, and it moves its attention to the first token when none is open. When all brackets close, its share of attention on them falls from 0.45 to 0.17 at depth 3. At depth 5 it falls from 0.56 to only 0.42. We set that share to 0 at depth 5. L8 N689+ fell only from 2.29 to 1.82. We split the neuron's input exactly into parts (`scripts/syntax_close_match.py`). The gap between one bracket open and none is 3.61 at depth 3, 2.64 at depth 4 and 1.20 at depth 5. L7H4, L4H2, L5H3 and the MLPs of L2 and L6 give most of it, and each part shrinks with depth.

The cause is which closer tokens end the call. In the items above, the first closer at depth 5 is one `))))` token, and the last is `)`. So we kept the depth and the count of brackets, and changed only the closer tokens (Table 28). Each row has 64 items: 8 callee orders times 8 targets. The model's own chains are those of Table 4.

Depth	Closers, every bracket closed	L8 N689+	Log-odds of one more closer
3	`)) );`, the model's own chain	0.00	−7.10
4	`)) ) )`	0.38	−2.70
4	`))) )`	0.68	−2.45
4	`))) );`	0.02	−6.47
4	`)) ),`), the model's chain in 5 of 8 items	0.34	−1.10
4	`)) ) )`	0.06	−1.57
4	`)) ))`	0.92	−0.79
5	`)) ) ) )`	0.45	−2.37
5	`)` five times	0.79	−1.74
5	`)))) )`	2.28	−1.85
5	`) ))) );`, the model's chain in 4 of 8 items	0.16	−5.53
5	`))) ))`, the model's chain in 2 of 8 items	1.50	−0.18
6	`))` then four `)`	0.59	−2.17
6	`)` six times	1.14	−1.55
6	`)))) ) )`	2.13	−1.68
6	`)) )) ))`	1.96	0.08

Table 28. The strength of L8 N689+ and the log-odds of one more closer after different sequences of closer tokens that close every bracket, at depths 3 to 6.

Two things set the result, and depth is not one of them. The first is the last token. After `);`, which ends the statement, depths 4 and 5 look like depth 3. L8 N689+ reads 0.02 and 0.16, and the log-odds are −6.47 and −5.53. After `)`, the log-odds are −2.70 to −1.55. After `)` or `))`, they are −1.57 to 0.08, even where L8 N689+ reads 0.06. So the token that ends the call says, by itself, whether more closers are likely.

The second is the packing. With `))` and then single `)` tokens, L8 N689+ reads 0.38 to 0.59 at depths 4 to 6, against about 3 with one bracket open. A `))))` token leaves it at 2.13 to 2.28, and a long run of single `)` tokens at up to 1.14. So the readers count a closer that packs 4 brackets as fewer than it closes. At depth 3, every chain ends with `);`. At depth 4 most chains end with `)`, and at depth 5 some end with `))`. So the extra closer at depths 4 and 5 comes from these closer tokens. It does not come from a count that fails with depth.

Packed tokens. The packed tokens are rare in code. In the 500 documents of the sweep, `)` occurs 2,254 times, `))` 225 times, `)))` 21 times and `))))` 3 times. And the model counts them less sharply. We put one closer of 1 to 4 brackets, then `)`, after calls of depth 2 to 7, and read L8 N689+ (Table 29).

Closer, then `)`	One bracket too many	Every bracket closed	One bracket left open
`)`	–	0.02	3.48
`))`	0.00	0.09	3.48
`)))`	0.00	0.68	3.53
`))))`	0.00	2.28	3.53

Table 29. The strength of L8 N689+ after one closer token and then `)`, when one bracket too many closes, when every bracket closes and when one is left open.

The neuron reads 0.00 whenever too many brackets close. So the model does not count `))))` as fewer brackets in a simple way. The step from one bracket open to none is sharp for `)` and `))`, and blurred for `)))` and `))))`. At the exact count, the log-odds of one more closer are −4.33, −3.82, −2.45 and −1.85. The rarer the token, the less sharp the count, but this is a correlation. We did not change how often the tokens occur.

Qwen2.5 in detail

Setup. The 0.5B model has 24 layers of 4,864 neurons, so 233,472 poles. The 1.5B model has 28 layers of 8,960 neurons, so 501,760 poles. Their MLPs are gated. A pole is the positive or negative part of the input to the down projection, which is the product of the gate and up projections.

The registry sets no window for Qwen, so each document is one piece, about 500 tokens on average. The 200,000-token sample holds all 396 documents with a grammar. Both models use the same tokenizer, so they read the same tokens.

Per layer and against the null. For round depth, GPT-2 has four layers whose best pole reaches 0.187 or more, the poles of Table 6. In Qwen2.5-0.5B the best of each layer lies between 0.059 and 0.160, with no layer that stands out. Its best pole is in L21 of 24, not in an early layer. The second best in that layer has 0.126, so the gap is smaller than in GPT-2's L2 (0.277 and 0.236). In Qwen2.5-1.5B the best of each layer lies between 0.044 and 0.175. Its best pole, L26 N7101+, is two layers from the output, and the second best in its layer has 0.077. A pole that late may predict a closing bracket, not count open ones.

The null is higher in Qwen. Its highest is 0.097 in Qwen2.5-0.5B (L0) and 0.060 in Qwen2.5-1.5B (L18), against 0.028 in GPT-2. Whole documents give a pole room to track position, and the null keeps each label's position. So a pole that follows position scores in the null too. L0 N2546+ has the best curly-depth R^2 (0.158) with a null of 0.092 in its layer. The best pole of that set by its margin over the null is L12 N1468+ (0.158, null 0.032). The other label sets' best poles, in both Qwen models, have a null of at most 0.039 in their layer.

At 128-token windows. We repeated the sweep by bracket kind on both Qwen models, at the windows that GPT-2 reads: 128 tokens, with a new window every 64 (Table 30). The script takes a `--window` option for this, and the rows are in `results.window128-qwen2.5-0.5b.jsonl` and `results.window128-qwen2.5-1.5b.jsonl`.

Label set	Qwen2.5-0.5B, whole	Qwen2.5-0.5B, 128 tokens	Qwen2.5-1.5B, whole	Qwen2.5-1.5B, 128 tokens
`round depth`	L21 N953− 0.160	L21 N953− 0.168	L26 N7101+ 0.175	L26 N7101+ 0.161
round brackets with no parser	L21 N953− 0.172	L21 N953− 0.180	L26 N7101+ 0.198	L26 N7101+ 0.180
square depth	L22 N1943+ 0.185	L22 N1943+ 0.187	L25 N8793− 0.237	L25 N8793− 0.240
curly depth	L0 N2546+ 0.158	L12 N1468+ 0.162	L22 N2957+ 0.258	L22 N2957+ 0.266
Highest null of the four	0.093	0.013	0.056	0.009

Table 30. The best pole and its held-out R^2 for four depth label sets in the Qwen2.5 models, on whole documents and at 128-token windows, with the highest null of the four sets.

The window changes little. Each best pole stays the same, except curly depth in Qwen2.5-0.5B, and each R^2 moves by at most 0.018. GPT-2's L2 N676+ (0.277) stays well above both. For round depth, the best pole of each layer lies between 0.054 and 0.168 in Qwen2.5-0.5B. In Qwen2.5-1.5B it lies between 0.032 and 0.161. So no early counter appears at the shorter window either. The null falls, since a window gives a pole less room to track position. In Qwen2.5-0.5B, L0 N2546+ loses its lead for curly depth to L12 N1468+. On whole documents, L12 N1468+ led by its margin over the null.

Qwen's best poles on written text. We read L21 N953− of Qwen2.5-0.5B and L26 N7101+ of Qwen2.5-1.5B on the probes of Table 2, with the same templates and targets (Table 31). Each cell is a mean over 8 items.

Condition, depths 1 to 5	**L21 N953−, 0.5B**	**L26 N7101+, 1.5B**
Open	1.66, 0.24, 0.82, 1.65, 1.76	0.00 at each depth
Closed	0.00 at each depth	2.59, 2.37, 2.24, 2.11, 2.00
One closed	1.33, 0.11, 0.26, 0.47, 0.55	0.00 at each depth
In a string	2.61, then 0.06 to 0.41	3.11, then 0.00
Square brackets	0.73, 0.19, 0.01, 0.40, 0.42	5.29, 3.63, 1.90, 0.66, 0.48
Curly brackets	0.00 at each depth	3.53, 2.26, 1.62, 1.07, 1.05

Table 31. The mean strength of the best round-depth pole of each Qwen2.5 model on the written probes, at depths 1 to 5.

The standard errors are at most 0.45 for L21 N953− and 0.72 for L26 N7101+. Neither pole counts depth. L21 N953− fires under open round brackets and not after they close, but it does not rise with depth. With one open bracket and a list of filler items before the target, it rises from 1.20 with no filler to 2.89 after 80 items. GPT-2's L2 N676+ falls to zero after 20 items. L26 N7101+ is silent under open round brackets, and fires once they close or under other brackets. So its R^2 for round depth comes from telling outside from inside, and it is high outside.

We then tested whether either pole predicts a closing bracket. Through the direct path, from the pole's output to the unembedding, L21 N953− raises closers. Its 15 tokens raised most include `)` with line breaks, `.)` and `())`, and 40 of its top 100 tokens start with `)`. L26 N7101+ raises a quote or a backtick followed by a closer, such as `")`, `')` and `` `) ``. None of its top 100 tokens starts with `)`, and `)` alone ranks 562 of 151,936. The direct path leaves out every later layer.

Ablation shows no effect. We ablated each pole on the 8 items of each depth. The probability of a closer moved by at most 0.003, and the count of brackets closed by at most 0.007. Ablating 20 random poles of the same layer and sign moves them as much or more. So L21 N953− writes toward a closer, and it grows as a bracketed list gets longer. But these templates show no effect of it on the output. L26 N7101+ does not predict a closing bracket.

L21 N953− through the protocol. We audited L21 N953− of Qwen2.5-0.5B as a chosen unit. The scan read the seed-1 sample of `github-code`, as whole documents, so its documents differ from those of the sweep. The proposer (Claude Opus 5.5) could read the

paragraphs above, and tested its wording on the model. It found a narrower trigger than our templates. Inside an open round bracket, the pole fires on the last token of an item. It fires most when the next token joins the end of the item to the closer, such as `__)`, `")` or `')`. Dunder names are strongest: `getLogger(__name__)` reads 18.5, and `getLogger(name)` reads 3.7. Our templates end on a plain argument before a bare `)`, which is the weakest case. The claim called this a lookahead (Table 32).

Gate or score	Result
Positive holdouts	7 of 7
Hard negatives	12 of 13
Counterfactuals	8 of 8
Detection, fuzzing	0.80, 0.75
Lookahead check	failed: measured retrospective, next-token KL 0.003
Verdict	NEEDS_REVISION (0.96)

Table 32. The first review of L21 N953− of Qwen2.5-0.5B, on a claim of a lookahead to a closer.

So the selectivity holds on new text, and the mechanism does not. The measured effect on the next token is small, as the ablation above found. The failed hard negative, `value = g.__next__` then `print(value)`, read 6.43 against a line of 4.87. The reviewer (Claude Sonnet 5.5) first wrote 21 probes. Three of its hard negatives had a dunder inside an open round bracket, which the claim calls active. We sent them back, and the same reviewer fixed them. Pull request 2757 added the record.

The revision read the same scan. Its proposer (Claude Opus 5.5) found that the failed hard negative peaked on `value` in `print(value`, a plain argument at the end of a snippet. `g.__next__` reads 0.00, since no round bracket is open there. It also found that the pole does not follow the chance of a closer. `(France)` in prose reads 0.4, where the model gives 0.95 to `)`, and several tokens inside one bracket fire, not only the last. So the revised claim is retrospective. The pole fires on content tokens while a round bracket is open, in code, or around a quoted string in prose. It reads 10 to 21 on the name after `__`, and 4 to 9 on the words of a quoted string and on digits. It reads 1 to 4 on plain names in short calls, and under 1.5 in prose with no quote. Table 33 compares the two claims.

Gate or score	First claim	Revised claim
Hard negatives	12 of 13	13 of 13
Positive holdouts	7 of 7	6 of 7
Counterfactuals	8 of 8	9 of 9
Detection, fuzzing	0.80, 0.75	0.82, 0.96
Lookahead check	failed	passed
Verdict	NEEDS_REVISION (0.96)	VERIFIED (0.93)

Table 33. The reviews of the first claim of L21 N953−, a lookahead, and of its revised, retrospective claim.

The failed holdout is `print(value` at the end of a function, which read 3.02 against a line of 4.87. The claim puts a late lone argument at 4 to 9, so that tier is too high. Pull request 2763 added the record. So the best round-depth pole of Qwen2.5-0.5B is a sign of an open round bracket, strongest on dunder names. It does not count depth.

Cut documents

The corpus cuts documents at 3,000 characters, usually inside a class or a function. The parser then has no node for that class, and its contents read as top level. We found this when a Java example at depth 1 and indent 4 read `top level`.

The first fix added the closing brackets that were still open, with a `;` before each `}`, parsed again, and labelled only the original text. It fails in three ways. A cut inside a `//` comment puts the added brackets in the comment. A cut inside a string or a template literal makes the first parse read the string's braces as brackets, so the closers are wrong. Ruby and Lua close blocks with `end`, which no bracket adds.

So the tagger now tries several endings when the parse has errors. Each starts with nothing, a line break, `*/` or a quote. Then come the brackets still open after that start, with and without the `;`. Ruby and Lua also get up to 12 `end`s. It keeps the ending that leaves the fewest characters of the original text in error nodes, the shortest of those. If no ending lowers that count, it adds the open brackets' closers, as the first fix did.

To measure this, we take the same sample with no cut, and parse each whole file. Then we cut it at 3,000 characters and compare the cut's labels with the whole file's. A label that agrees is stable. `Lost` is the share of characters in a construct in the whole file that read as top level after the cut (Table 34).

Language	Files	Agreement, one parse	Closers added	Endings searched	Lost, one parse	Closers added	Endings searched
All	157	0.800	0.943	0.988	0.278	0.077	0.017
C	36	0.871	0.976	1.000	0.254	0.046	0.000
C#	10	0.603	0.885	1.000	0.472	0.137	0.000
C++	24	0.863	0.967	0.969	0.201	0.046	0.044
Go	4	0.918	0.949	0.999	0.109	0.068	0.001
Java	30	0.753	0.973	0.999	0.317	0.029	0.000
JavaScript	18	0.754	0.925	1.000	0.287	0.088	0.000
PHP	12	0.701	0.897	0.970	0.425	0.145	0.042
Python	13	0.990	0.997	0.998	0.009	0.001	0.001
Ruby	6	0.589	0.590	0.872	0.541	0.540	0.169
Scala	1	0.597	1.000	1.000	0.521	0.000	0.000
TypeScript	3	0.783	0.999	0.999	0.297	0.000	0.000

Table 34. The agreement of `syntax construct` labels on cut documents with those of the whole file, and the share of characters in a construct lost to top level, with one parse, with closers added and with endings searched.

Cut documents broke the labels until we fixed them. Against the parse of the whole file, a single parse of a cut document loses 27.8% of the characters that lie in a construct to `top level`. Adding the missing closing brackets lowers this to 7.7%. Trying several endings, including a line break, a quote and Ruby's `end`, lowers it to 1.7%, and to 0.3% on a second sample.

Agreement in Table 34 is for `syntax construct`. `syntax node` agreement rises from 0.992 to 0.999. We chose the endings on this sample, so we also tested them on the seed-1 sample, which we did not look at while choosing them. There, at 3,000 characters, agreement is

0.819 with one parse, 0.974 with closers added and 0.997 with endings searched. Lost falls from 0.272 to 0.038 and 0.003. A cut at 1,500 characters gives similar numbers in both samples. Few files stand behind some rows: Scala has 1, TypeScript 3 and Go 4. Haskell, Lua and Shell have no file over 3,000 characters in this sample.

The sweeps above use the searched endings. Each layer's best R^2 moved by at most 0.011 from the first fix (`syntax construct` in L10). The best of each set across layers moved by at most 0.003. Most of the sampled documents are not cut in PHP or Ruby.