What default-on negative poles do as a population

Will Sackfield, Open Interpretability · https://open-interpretability.com/research/default-on-poles

Abstract

Most neurons of GPT-2 small rest below GELU's minimum, so their negative side is on for almost every token and cannot be audited as a detector. We asked what these default-on negative poles do together, with random nulls for each claim. Training produces this rest: at layer 9 the median neuron rests at $h = -1.04$, against 0.00 at random initialisation, and the bias gives only -0.08 of that. The negative side's cost is mostly its resting level. With all 12 layers changed, holding it at its mean adds 0.19 nats of cross-entropy, but setting it to zero adds 3.72. Each reading of its variation does about as well on a matched null. A random direction's low tail is as reliable by token as a real neuron's (0.68 against 0.70), and random features decode token identity and human labels about as well as the negative side does. The negative side leads random features by at most 0.025 of R^2 on the next-token distribution, where the positive side leads by up to 0.073. In Qwen2.5-0.5B, a SwiGLU model, the two sides are symmetric and each costs 2.46 nats when clamped. These results support the protocol's choice to stop auditing default-on poles one at a time.

Status: Preprint. Work left: #3195 (the negative side's lead on the next-token distribution).

1 Introduction

Interpretability work often explains a trained network one unit at a time. The registry that this paper draws on reads each sign of an MLP neuron's activation apart, and calls each side a *pole*. A pole gets an explanation, and an adversarial review tries to break it with probes. A default-on pole is a negative pole that is on for almost every token, so no probe can stay silent and its explanation can never pass. Protocol 0.8 therefore stops the audit of every default-on pole (position zero). The registry thus says nothing about a large share of negative poles, unless those poles do little worth explaining.

GELU (Hendrycks & Gimpel, 2016) dips below zero. Its minimum is near $h = -0.75$, a value we computed, since the source gives none. So a neuron that rests near that dip writes a small negative value on most tokens. An earlier study of GPT-2 L9/N1046−, the negative pole of neuron 1046 in layer 9, suggested that these poles carry a signal: a scaled-down copy of the positive side, with a readable set of tokens below the dip. Its scripts were not kept, and its numbers survive in an earlier `TODO.md`. It did not measure every layer, and it had no random null. A readable tail, a decodable signal and a cost under ablation can each arise without learned content, so the open question is which of them survive a null.

This paper asks what default-on negative poles do as a population, and whether it is safe to stop auditing them one by one. It bears on research question 1, *Are native MLP neurons clean?*, and on question 5 (research questions).

The paper's contributions are these:

- A rebuild of the first study's measurements on GPT-2 small, at every layer, with each number set against the first study's (Appendix B).
- Evidence that training, through the input weights and not the bias, rests most GPT-2 neurons left of GELU's dip.
- A split of the negative side's cost into its resting level and its variation, with a shuffled null of the same size for each intervention.
- Nulls for three readings of the negative side: random directions for its token tail, and random features and a token lookup for decoding.
- A comparison with a SwiGLU model, Qwen2.5-0.5B, where the two sides are symmetric and cost the same.

2 Related work

Random directions and readable units. Szegedy et al. (2014) found "no distinction between individual high level units and random linear combinations of high level units" in image networks. In their Section 3, "any random direction" gives "similarly interpretable semantic properties". We test the same claim on the low tail of GPT-2 MLP neurons, with a reliability score in place of a reading by eye.

Probes and their controls. Belinkov (2022, Section 4.1) reviews random baselines, and reports that "even random features capture significant information that can be decoded by the probing classifier". Hewitt and Liang (2019) propose control tasks that pair word types with random outputs, which "can only be learned by the probe itself". A good probe is then selective, with high accuracy on the real task and low accuracy on the control. Kutluozen et al. (2026) argue that a random feature map also decodes factors linearly, so probe accuracy needs a random-initialisation control. They also show that the ridge penalty changes the random arm's score. Wilam (2026) reports that a linear probe on the hidden units of an untrained model already reaches a median accuracy of 0.802. Our decoding tests follow this line, with a token-lookup baseline and a random-feature null. Belinkov (2022, Section 4.3) adds that probing "does not tell us whether this property is involved in predictions", so we test use with interventions.

Nulls for single units. Wilam (2026) gates each reading of a model's neurons with a split-half null and a scrambled-label null, and compares each reading with untrained references. Our split-half reliability by token, our random-direction null and our random-initialisation comparison of resting levels are the same kind of control.

The two sides of a neuron. Tigges et al. (2026) read the two poles of SwiGLU neurons in a large mixture-of-experts model apart. They find that the two poles usually have unrelated roles, and that "neurons are never actually off". In their model, the two sign regimes that give negative values hold about half of all firings (13–14% and 36–37%). The double-negative regime, with gate and up both negative, gives positive values and holds about a third. Haider et al. (2026) find that concepts sit in distinct ranges of a unit's activation, for units of the residual stream. Both bear on our question, because the range below the dip could be a separate side with its own content.

Compensation by later layers. McGrath et al. (2023) find that when one attention layer is ablated, "another later layer appears to take over its function". They call this the Hydra effect. They also find that late MLP layers act to "downregulate the maximum-likelihood token". We see a similar compensation after a clamp of an early MLP layer. Their result concerns attention layers, so the extension to MLP clamps is our own step.

Massive activations and attention sinks. Sun et al. (2024) report massive activations on GPT-2's starting token. Xiao et al. (2023) name the initial tokens that draw strong attention "attention sinks". Both suggest why an intervention at the first token can reach every later token.

3 Method

Terms. A neuron's pre-activation is h, the output of the MLP's first linear map (`c_fc` in GPT-2). Its activation is $a = \mathrm{GELU}(h)$, the input of the second linear map (`c_proj`). A neuron's *negative side* is its tokens with $a < 0$, and its *positive side* those with $a > 0$. A pole is *active* on a token when its strength reaches 0.25 of its maximum. We call that level the *activation line*. A pole is *default-on* when it is active on almost every token, so that no probe can stay silent. We write L9/N1046− for the negative pole of neuron 1046 in layer 9.

The dip. GELU is the input times the standard normal distribution function Φ (Hendrycks & Gimpel, 2016):

$$\mathrm{GELU}(x) = x\,\Phi(x)$$

The minimum of this exact form is at $h = -0.7518$. GPT-2's MLP uses the tanh approximation that the same paper gives, whose minimum is at $h = -0.7525$. Our scripts use the exact form's minimum. A token is *below the dip* when $h < -0.7518$ and the pole's strength is under the activation line. GELU is back near zero there, so the pole is silent.

Models and data. We use GPT-2 small (Radford et al., 2019) and Qwen2.5-0.5B, from the Qwen2.5 family (Yang et al., 2024). The text is a 200,000-token sample with seed 0 from the first 10,000 documents of the Pile (Gao et al., 2020). We cut it into whole 128-token windows that do not overlap. GPT-2 gets 1,355 windows from 384 documents, and Qwen gets 1,328 from 420. We fit constants on 300 windows and test on 400 others, with no document in both. Windows of one document share a topic, so a split by window would leak.

First tokens. Band statistics skip each window's first token, and interventions leave it unchanged. The conditions marked *with first* change it too (Section *What does each side cost?*).

Interventions. Table 1 lists the changes that we make to the activations, and Figure 1 shows them on one neuron. KL is the divergence of the changed next-token distribution q from the clean one p, summed over the vocabulary's tokens t:

$$\mathrm{KL}(p\|q) = \sum_t p(t)\log\frac{p(t)}{q(t)}$$

A *clamp* keeps a side's mean and removes its variation. A *zero* removes the mean as well. The *shuffled null* has the same size of change as the clamp, with no link to the token.

Condition	Change
`clamp`	Each neuron holds its side's mean on that side's tokens
`zero`	Each neuron reads zero on that side's tokens
`mean`	Each neuron holds its mean on all tokens
`null`	Each neuron gets the clamp's change, shuffled over the batch's tokens. Its size per neuron stays, and its link to the token goes
`frozen`	A clamp at one layer, with later MLP writes kept at their clean values
`direct`	A clamp's change to one layer's write, added at the final norm only

Table 1. The interventions. Each changes one side of every neuron in the layers that a run names, except `mean`, which changes all tokens.

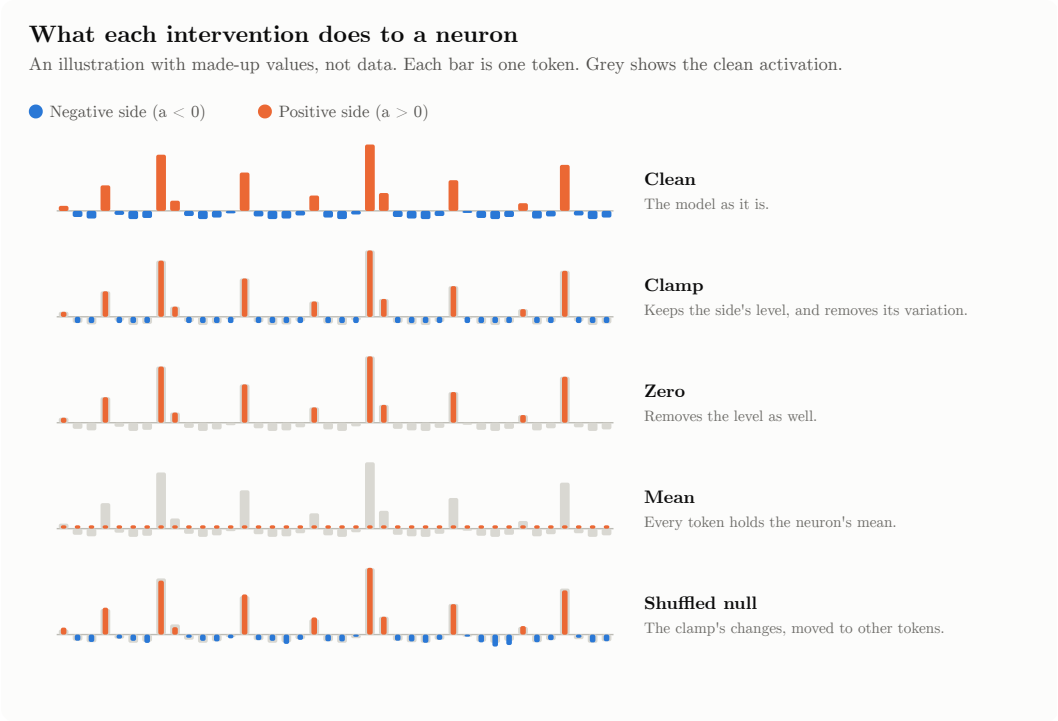

Figure 1. What each intervention does to a neuron's activation, over forty tokens. This is an illustration with made-up values, not data. Grey shows the clean activation. The clamp keeps the side's level and removes its variation, and the zero removes the level too.

Intervals. All intervals are 95% bootstrap intervals over documents.

Figures. Each figure is drawn from the recorded results, with no model, except the illustration of Figure 1. Appendix A gives the scripts and the commands.

4 Does training rest the neurons below the dip?

The first question is whether resting below the dip is learned, and what sets it. We measured each layer's median h on the trained model and on random initialisations of the same architecture (Table 2, Figure 2 and Figure 3). At random initialisation 23–40% of neurons are default-on. After training, 75–97% are, on every layer but L1.

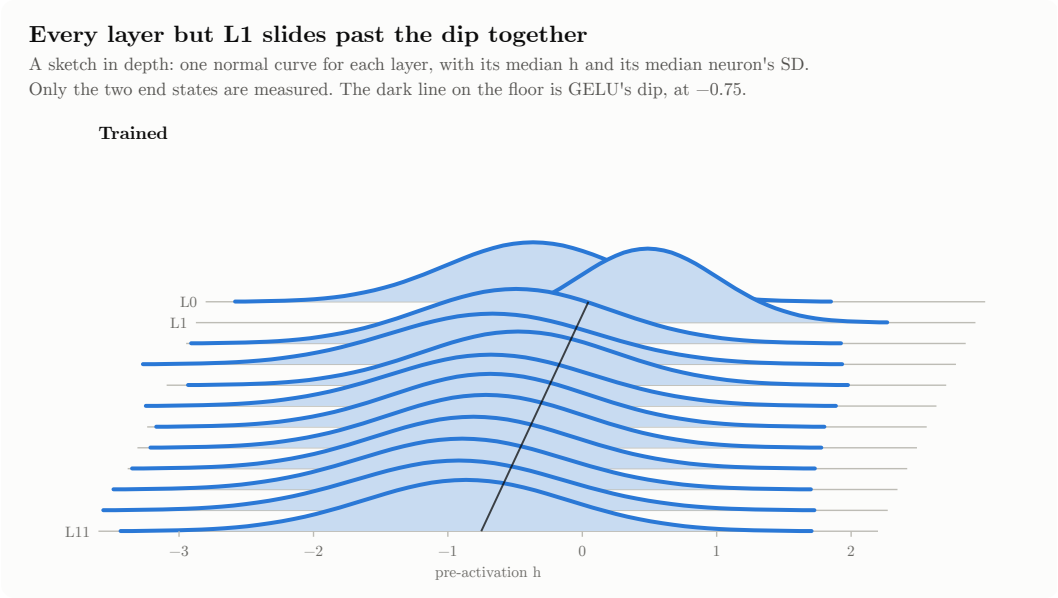

Figure 2. Every layer but L1 moves left of GELU's dip during training. An animated sketch in depth: one normal curve for each layer of GPT-2 small, with its median h and its median neuron's SD. The ridges move between random initialisation (mean of 5 seeds) and the trained model, and only these two end states are measured. The dark line on the floor is GELU's dip, at -0.75.

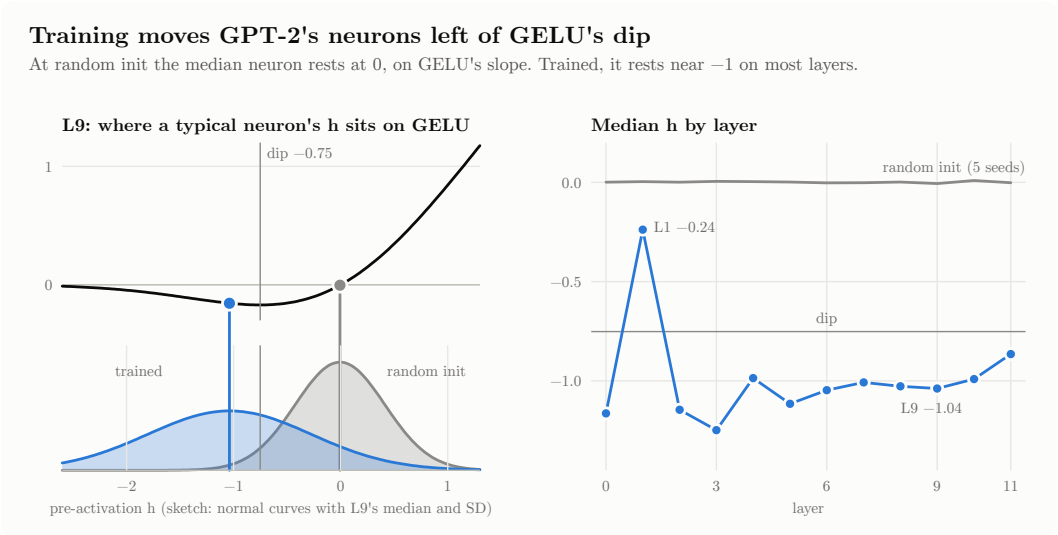

Figure 3. Training moves GPT-2's neurons left of GELU's dip. Left: GELU, with a sketch of a typical L9 neuron's h before and after training. Right: the trained median h by layer.

Layer	0	1	2	3	4	5	6	7	8	9	10	11
Median h, trained	−1.16	−0.24	−1.15	−1.25	−0.99	−1.11	−1.05	−1.01	−1.03	−1.04	−0.99	−0.86
... from the bias	−0.07	−0.05	−0.09	−0.09	−0.08	−0.07	−0.08	−0.08	−0.08	−0.08	−0.08	−0.07
... from the input term	−1.08	−0.20	−1.04	−1.15	−0.92	−1.03	−0.97	−0.92	−0.94	−0.96	−0.92	−0.81
Default-on %, trained	86	58	75	86	90	90	94	96	97	97	96	92
Default-on %, random init	23	35	37	38	38	39	39	40	40	40	39	40
Mean |GELU slope|, trained	0.21	0.31	0.23	0.19	0.21	0.20	0.21	0.21	0.20	0.20	0.21	0.24
Mean |GELU slope|, random init	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.52	0.51

Table 2. Where GPT-2's neurons rest, by layer. The random-init rows are means over 5 seeds of the same architecture, and their median h is 0.00 on every layer, to within 0.01. The bias and input-term rows are medians of different quantities, so they do not sum exactly to the median h.

Most neurons rest left of the dip, not in it. On every layer but L1, the typical median is below −0.75, so the typical neuron sits on GELU's flat left tail. At L9 the signed mean slope is 0.09, and the mean absolute slope is 0.20, against 0.51 at random initialisation. The input weights, not the bias, make this rest. A bias could set the rest alone, but its median is at most −0.09 in size on any layer, against an input term of −0.81 to −1.15 outside L1. So the rest comes from how the weights project the inputs that the MLP reads. L1 is the exception. Its median is −0.24, and 58% of its neurons are default-on. We did not investigate why.

Why training does this stays open. We tested one idea, that a low slope at rest lowers the gain on input noise. The gain on noise does fall, by 3–22×, but the signal's gain falls by more on L0–L9. So this test does not support a lower gain on noise as the reason for the rest (Appendix C, Table 13).

5 What does each side cost?

The second question is how much of the prediction depends on each side, and on which part of it. We changed all 12 layers at once, on GPT-2's test windows, where the clean cross-entropy is 3.63 nats (Table 3 and Figure 4).

Condition	Added nats	KL	Top prediction unchanged
Negative side, shuffled null	0.08	0.09	82%
Negative side, clamped	0.19	0.18	77%
Positive side, shuffled null	0.82	0.88	53%
Positive side, clamped	1.34	1.46	40%
Negative side, zero	3.72	3.90	13%
Every neuron at its mean	4.09	4.32	13%
Positive side, zero	8.05	8.31	2%

Table 3. The cost of each intervention on GPT-2 small, with all 12 layers changed. Added nats is the rise in cross-entropy over the clean 3.63. The last column is the share of tokens whose top prediction does not change.

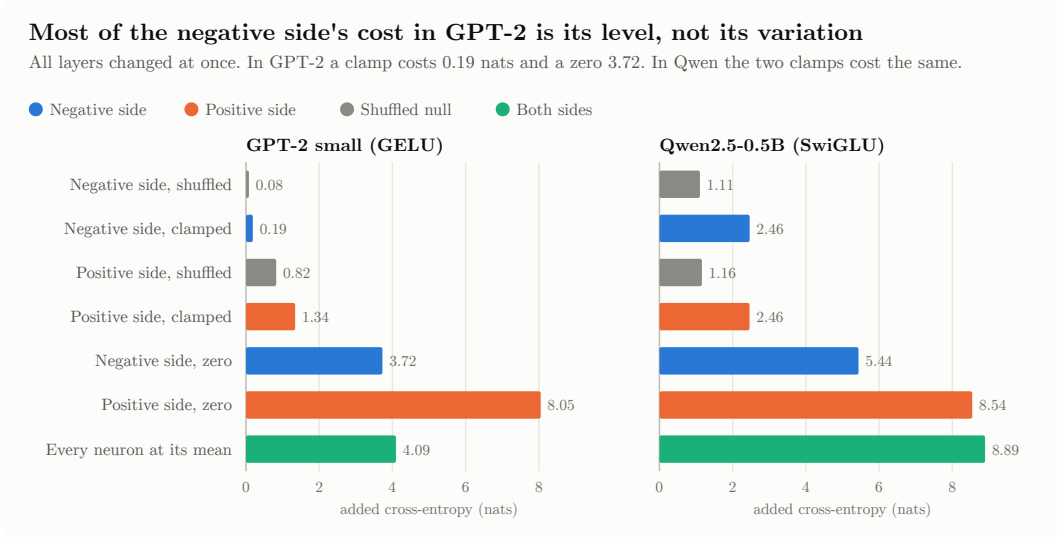

Figure 4. Most of the negative side's cost in GPT-2 is its level, not its variation. Each bar is the added cross-entropy of one condition, with all layers changed, for GPT-2 small and Qwen2.5-0.5B.

The resting level costs far more than the variation. The negative side's zero costs about 20× its clamp. The clamp keeps the side's mean and removes its variation, whereas the zero also removes the mean. So most of the 3.72 nats come from removing a nearly constant offset, not 3.7 nats of varying signal, as the first study's numbers could suggest.

The variation still does something. The clamp's KL is 1.9× its null's, at the same size of change. So the variation's link to the token matters, but its total cost is small. Single poles do almost nothing. Zeroing L9/N1046− gives a KL of 1.3×10^{-5}, and 10 default-on controls give 1.2–2.5×10^{-5}. So N1046 is an ordinary default-on pole.

The first token

The first token behaves differently from the rest. If interventions also change each window's first token, the positive clamp costs 6.14 nats, not 1.34, and only 13% of top predictions stay. Holding every neuron at its mean costs 5.63, not 4.09. The negative side barely changes, at 0.19 for the clamp and 3.75 for the zero. In our data the first token has very large positive neuron activations. Sun et al. (2024) find massive values on GPT-2's starting token in the residual stream, a different quantity. Xiao et al. (2023) find attention

sinks on the initial tokens of Llama-2, MPT, Falcon and Pythia. They didn't test GPT-2. That both quantities are extreme at this position is our observation, not theirs. Any population clamp needs to say what it does at this position.

6 Is the negative side the same signal, scaled down?

The first study read the negative side as a scaled-down copy of the positive side, because the ratio of the two sides' KL tracks the ratio of their activation variances. The third question is whether that match shows a shared signal. We clamped one layer at a time, and give each ratio as negative over positive (Table 4 and Figure 5).

Layer	**KL ratio (95%)**	**Activation variance ratio**	**Null's KL ratio**	**Clamp / null, −**	**Clamp / null, +**	**Frozen / clamp, −**	**Frozen / clamp, +**	**Direct / clamp, −**	**Direct / clamp, +**
L0	0.06 (0.06–0.07)	0.14	0.09	1.71	2.41	2.65	1.48	0.10	0.06
L1	0.24 (0.22–0.25)	0.21	0.20	1.25	1.05	1.62	1.57	0.07	0.07
L2	0.27 (0.25–0.28)	0.29	0.23	1.30	1.14	1.40	1.00	0.07	0.07
L3	0.26 (0.25–0.27)	0.31	0.29	1.30	1.47	1.57	1.06	0.09	0.07
L4	0.27 (0.25–0.28)	0.29	0.27	1.42	1.46	1.38	0.93	0.09	0.07
L5	0.23 (0.22–0.24)	0.24	0.24	1.38	1.43	1.02	0.76	0.12	0.11
L6	0.21 (0.20–0.22)	0.21	0.20	1.40	1.34	0.78	0.69	0.15	0.15
L7	0.15 (0.15–0.16)	0.17	0.18	1.23	1.44	0.81	0.69	0.24	0.24
L8	0.13 (0.13–0.14)	0.16	0.16	1.27	1.52	0.89	0.79	0.35	0.37
L9	0.11 (0.11–0.12)	0.13	0.13	1.19	1.36	0.95	1.00	0.51	0.71
L10	0.06 (0.06–0.07)	0.09	0.07	1.15	1.29	0.94	1.04	0.74	0.89
L11	0.05 (0.05–0.05)	0.04	0.02	1.23	0.59	1.00	1.00	1.00	1.00

Table 4. Single-layer clamps on GPT-2 small. The first three ratios are negative over positive. The last six columns compare each side's clamp with its shuffled null, with the clamp when later MLPs are frozen, and with the clamp's direct path to the final norm.

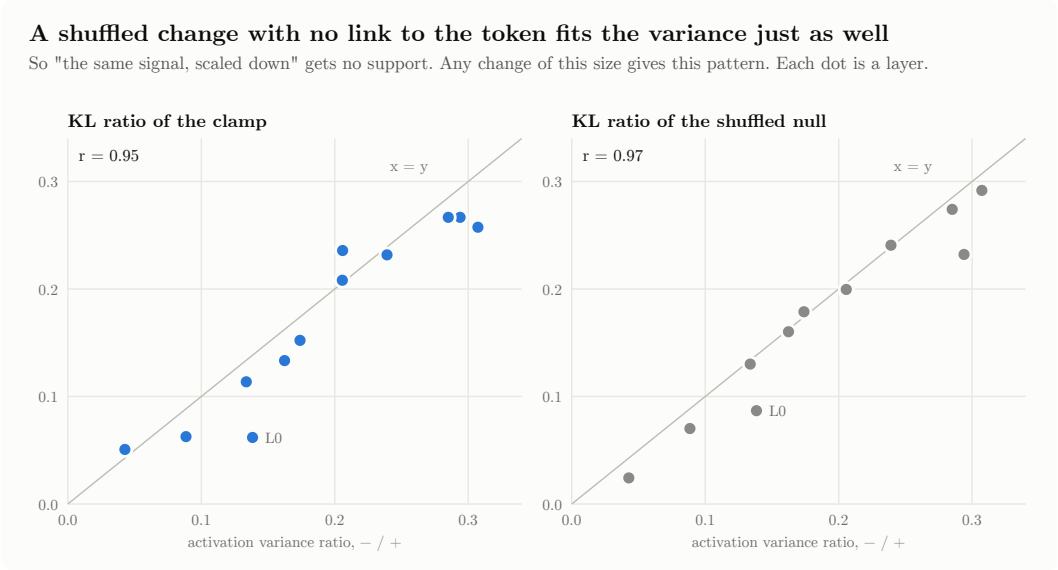

Figure 5. A shuffled change with no link to the token fits the variance just as well. Each dot is a layer of GPT-2 small. Left: the clamp's KL ratio, with $r = 0.95$ against the activation variance ratio. Right: the shuffled null's KL ratio, with $r = 0.97$.

The match is real, but it says little. Over the 12 layers the KL ratio correlates with the variance ratio at $r = 0.95$, as the first study saw. L0 is the clear miss, at 0.06 against 0.14. But the shuffled null's ratio correlates with the variance ratio at $r = 0.97$, although it has no link to the token. So KL mostly tracks the size of the change. Twelve layers are few, so these correlations are weak evidence in any case.

The sides are not shown to be one signal. Centred on the token mean, the two sides' writes have a mean cosine of -0.19 to 0.28 by layer. A ridge readout predicts the positive write from the negative side with R^2 of 0.28–0.82, but random features get 0.23–0.78 (Table 7). So most of that overlap comes from the shared input.

Later layers compensate for early clamps and amplify middle ones. Freezing the later MLPs raises the negative clamp's KL at L0–L4, by 1.38–2.65×, so those later MLPs undo part of the damage. This resembles the Hydra effect of McGrath et al. (2023), which they found for attention layers. At L6–L8 freezing lowers the KL, to 0.78–0.89×, so later MLPs add to the damage there. The positive side has the same shape, with less of this. The ratio survives both changes. With later MLPs frozen, or with the direct path alone, it still correlates with the variance ratio, at $r = 0.98$ and 0.97.

Most of an early clamp's effect is indirect. The direct path to the final norm carries 6–10% of the KL at L0–L4, and all of it at L11, where nothing follows. Clamps also add up to more than their parts. The sum of the single-layer KLs is 0.11 for the negative side and 0.79 for the positive. All layers together give 0.18 and 1.46.

7 Are the tokens below the dip a readable unit?

A default-on pole is silent only below the dip, so the tokens there are its only distinct set. The first study read N1046's set as code syntax and pronouns after "what". The fourth question is whether such a set is evidence that the neuron means something. We measured L9 on all 1,355 windows. Split-half reliability correlates a neuron's below-dip rate by token between even and odd documents, over tokens with at least 10 occurrences in each half. As a null, we read random directions in the MLP's input in the same way, cut at the same share of tokens. Table 5 gives N1046 and 10 controls, drawn at random from the layer's default-on neurons.

Unit	**Share of tokens**	**Reliability**	**Top tokens, as we read them**
N1046	2.4%	0.70	code syntax (`^{`, `:=`, `{`, `!=`), and some names (`Henry`, `Burton`)
N1132	2.0%	0.70	abstract nouns and contractions (`statement`, `'d`, `effect`)
N1742	3.3%	0.56	French pieces (`à`, `ée`, `aux`), and a few others
N2084	4.0%	0.73	a mix: titles, code words, word pieces
N1622	6.2%	0.60	a mix: names, years, capitals
N1965	10.5%	0.80	a mix: word starts, months, Romance articles
N918	11.2%	0.83	discourse words (`however`, `particularly`, `So`), and word pieces
N2455	12.2%	0.76	code and markup (`div`, `height`, `span`), and contractions
N2158	21.3%	0.81	code punctuation (`");`, `};`, `/>`), and identifiers
N2778	24.4%	0.82	proper nouns and web words (`University`, `https`, `Inc`)
N2312	29.1%	0.82	pronouns and sentence openers (`He`, `She`, `This`)
Random directions at N1046's share	2.4%	0.68 (5th–95th percentile 0.52–0.81)	Table 6

Table 5. The tokens below the dip at L9 of GPT-2 small, for N1046, 10 default-on controls and random directions. The share is the share of tokens below the dip. Reliability is the split-half correlation by token, with the split by document.

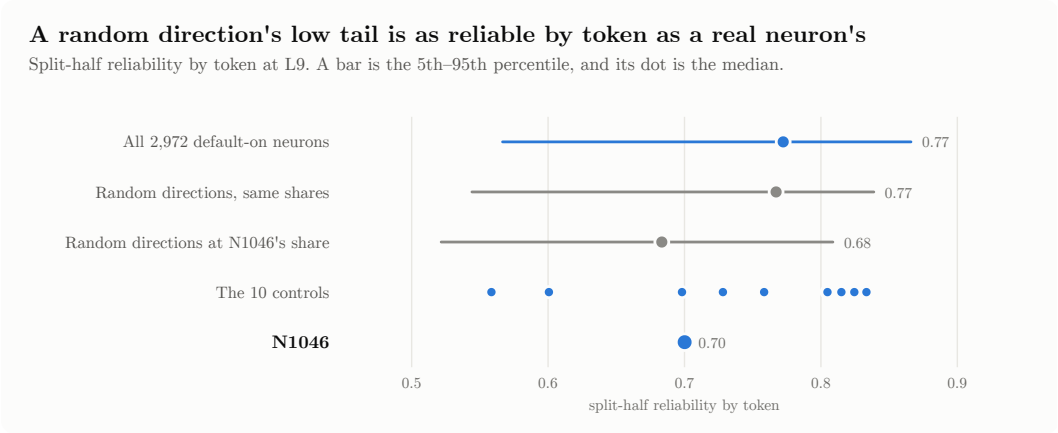

Figure 6. A random direction's low tail is as reliable by token as a real neuron's. Split-half reliability by token at L9. A bar is the 5th–95th percentile, and its dot is the median.

A real neuron's tail is no more reliable than a random direction's (Figure 6). Over all 2,972 default-on neurons at L9, the median reliability is 0.77, with a 5th–95th percentile of 0.57–0.87. N1046 is at the 22nd percentile. Random directions with the same distribution of shares have a median of 0.77, with a range of 0.54–0.84. At N1046's own share of 2.4%, random directions have a median of 0.68, against N1046's 0.70.

Direction	First 12 tokens, unfiltered
0	(space), `=`, `":`, `m`, `ever`, `',`, `/**`, `same`, `!=`, `p`, `n`, `",`
1	`Chief`, `football`, `chief`, `public`, `Red`, `leader`, `nucle`, `AC`, `ri`, `confidence`, `lu`, `telephone`
2	`Let`, `\ufffd`, `pp`, `There`, `Rebecca`, `has`, `ni`, `SP`, `He`, `Court`, `PP`, `He`
3	`height`, `starting`, `inal`, `Heart`, `focus`, `Start`, `evaluate`, `te`, `:=`, `ial`, `trying`, `complete`
4	`som`, `lic`, `usr`, `method`, `easy`, `library`, `methods`, `\ufffd`, `apparatus`, `Nar`, `License`, `invention`

Table 6. Five random directions at N1046's share of 2.4%, read in the same way as the neurons of Table 5. Each row gives the first 12 tokens below the cut, with no filter.

A random direction's tail can also suggest a theme (Table 6). Direction 0 has code syntax among its top tokens, and direction 2 has sentence openers and pronouns. Those are the first study's two readings of N1046, and N1046's own list is also a mix. Several lists have no clear theme, as do some of the controls. We read these lists knowing which were random, so the readings are not blind, but the reliability numbers do not depend on them.

Token identity dominates the MLP's input, so a reliable tail is no evidence that a neuron means something, and a readable one is weak evidence. This is the result that Szegedy et al. (2014) report for image networks, here for the tail of a language model's neurons. We could not check the first study's phrase "pronouns after 'what'", because this method reads single tokens. Surprisal does not explain the tails either. The Spearman ρ between h and the token's surprisal given its prefix is 0.05 for N1046, and -0.14 to 0.11 for the controls.

8 Can the negative side be decoded?

The fifth question is whether the negative side's collective signal decodes into anything meaningful. A ridge readout fits on 300 windows and tests on 400. It reads one side of a layer's activations, and predicts which of the 200 commonest tokens is the current or the next token. Chance is 12.5%. The null reads the negative side of random features of the same input, each with a real neuron's median and SD of h. Table 7 gives the scores.

Layer	Current token: −	+	null	Next token: −	+	both	null	Positive write R^2: from −	from null
L0	0.999	0.996	1.000	0.365	0.361	0.361	0.358	0.67	0.63
L1	0.998	0.993	0.999	0.373	0.373	0.374	0.367	0.50	0.43
L2	0.990	0.984	0.999	0.385	0.386	0.387	0.379	0.40	0.33
L3	0.984	0.963	0.991	0.392	0.392	0.396	0.386	0.36	0.30
L4	0.968	0.944	0.984	0.398	0.393	0.400	0.397	0.31	0.25
L5	0.954	0.932	0.973	0.414	0.413	0.420	0.411	0.30	0.24
L6	0.944	0.936	0.963	0.421	0.424	0.429	0.419	0.30	0.24
L7	0.936	0.918	0.951	0.435	0.450	0.459	0.440	0.29	0.25
L8	0.919	0.905	0.927	0.450	0.481	0.485	0.461	0.28	0.23
L9	0.880	0.872	0.887	0.463	0.513	0.515	0.479	0.33	0.28
L10	0.831	0.825	0.837	0.470	0.540	0.543	0.497	0.50	0.46
L11	0.779	0.778	0.788	0.487	0.544	0.546	0.506	0.82	0.78

Table 7. Ridge readouts of token identity on GPT-2 small. The token columns give the accuracy on the 200 commonest tokens, from the negative side, the positive side, both sides and the random-feature null. The last two columns give the R^2 of the positive side's write, predicted from the negative side and from the null.

The negative side decodes well, and that alone means little. It gets 78–99.9% on the current token, but random features get as much or more on all 12 layers. On the next token, the negative side leads the null by at most 0.008 on L0–L6, and the null leads on L7–L11. The positive side does beat the null late. On the next token it gets 0.544 at L11, against the null's 0.506, and adding the negative side adds almost nothing (0.546). So late positive poles hold something that the input's random features do not give.

A readout shows what is present, not what later layers read (Belinkov, 2022), and the clamps of Table 3 are the test of use. So this test finds no learned content in the negative side, although it tests only the 200 commonest tokens.

9 Does the negative side carry other content?

Token identity is the easiest thing to read from the MLP's input, so it could hide content that needs a learned direction. The sixth question is whether the negative side leads random features on two harder targets. We kept the same readout, with three draws of the null. Each target has a baseline that reads only the current token, which we call the *lookup*. Like the control tasks of Hewitt and Liang (2019), the lookup shows what a readout gets from token identity alone.

The model's next-token distribution. The target is the model's log-probability of the 200 commonest tokens, with the same fit and test windows. The score is R^2 on the test

windows (Table 8).

Layer	Lookup	−	+	both	null (3 draws)	− minus best null	+ minus best null	Top agreement: −	+	best null
L0	0.278	0.418	0.446	0.459	0.410–0.417	+0.002	+0.030	0.331	0.323	0.303
L1	0.278	0.489	0.491	0.520	0.465–0.466	+0.022	+0.024	0.358	0.335	0.346
L2	0.278	0.522	0.521	0.540	0.493–0.500	+0.021	+0.021	0.368	0.377	0.361
L3	0.278	0.536	0.547	0.565	0.505–0.514	+0.022	+0.033	0.389	0.387	0.376
L4	0.278	0.567	0.555	0.582	0.539–0.546	+0.021	+0.009	0.398	0.391	0.387
L5	0.278	0.625	0.628	0.650	0.598–0.602	+0.023	+0.027	0.418	0.420	0.408
L6	0.278	0.655	0.674	0.690	0.623–0.632	+0.023	+0.043	0.436	0.449	0.427
L7	0.278	0.688	0.720	0.733	0.661–0.668	+0.021	+0.053	0.456	0.480	0.457
L8	0.278	0.710	0.747	0.760	0.692–0.697	+0.013	+0.049	0.481	0.541	0.493
L9	0.278	0.734	0.775	0.787	0.706–0.721	+0.013	+0.053	0.514	0.601	0.536
L10	0.278	0.767	0.792	0.809	0.739–0.742	+0.025	+0.050	0.527	0.671	0.580
L11	0.278	0.806	0.871	0.879	0.792–0.798	+0.008	+0.073	0.545	0.717	0.634

Table 8. Ridge readouts of the model's next-token distribution on GPT-2 small. The first columns give R^2 on the test windows, for the lookup, each side, both sides and the three null draws. The lead columns subtract the best null draw. The last three columns give how often the readout's top token matches the model's.

The negative side leads the null a little. Its lead over the best of three draws is 0.008–0.025 on L1–L11, and 0.002 on L0. Its 95% interval by document, against the first draw, is above zero on L1–L11. The draws themselves differ by up to 0.015, so the smaller leads are close to that spread. The positive side leads by more late. Its lead grows to 0.073 at L11, where the negative side's is 0.008. Reading both sides adds 0.008–0.029 to the better single side.

The top prediction goes the other way late. The negative side matches the model's top token more often than the null on L0–L6, and less often on L7–L11. At L11 it is 0.545, against 0.634 for the null and 0.717 for the positive side. Entropy agrees with R^2. The negative side leads the null on the distribution's entropy by 0.001 at L0, and by 0.009–0.035 on L1–L11.

A lead does not show that training built the negative side for this. Both sides read the same h, so weights learned for the positive side would also help the negative side beat random weights. This test cannot separate the two.

Human labels. The labels come from Universal Dependencies (Nivre et al., 2020), CoNLL-2003 (Tjong Kim Sang & De Meulder, 2003), AG News (Zhang et al., 2015) and a language-identification dataset. AG News's topics are its source corpus's categories, and Zhang et al. don't say how they were assigned. There is one row per token, on layers 0, 3, 6, 9 and 11. Documents are split between fit and test, and each is cut to one window. The score is the mean AUROC over labels on the ambiguous tokens. These are the test tokens whose type both has the label and lacks it in the fit documents. The lookup reads the current token's rate of the label (Table 9).

Dataset	Labels	Lookup	− at L6	+ at L6	Best null at L6	− minus best null: L0	L3	L6	L9	L11
ud-upos	14	0.872	0.965	0.964	0.962	+0.002	+0.004	+0.003	+0.001	−0.001
ud-deprel	20	0.864	0.951	0.949	0.947	+0.004	+0.006	+0.004	−0.003	−0.002
conll2003-ner	4	0.884	0.967	0.963	0.965	+0.005	+0.010	+0.002	+0.001	−0.002
ag_news	4	0.619	0.900	0.892	0.901	+0.003	+0.003	−0.001	+0.008	+0.011
language-identification	20	0.805	0.977	0.967	0.975	0.000	+0.001	+0.002	0.000	−0.001

Table 9. Mean AUROC over each dataset's labels, on the ambiguous tokens. The middle columns give each side and the best null draw at L6. The last five columns give the negative side's lead over the best null at each layer tested.

All sides read context, and the null reads it too. Each side is far above the lookup, so these labels hold more than token identity. But the negative side's lead over the best null is at most 0.011 in all 25 cells, and it is above zero in 17. The interval against the first draw is above zero in 13, with no correction for multiple tests. Most AUROCs are near their ceiling, so a small lead here is weak evidence either way. Only ag_news has much room, and its largest lead is 0.011. Figure 7 shows both targets.

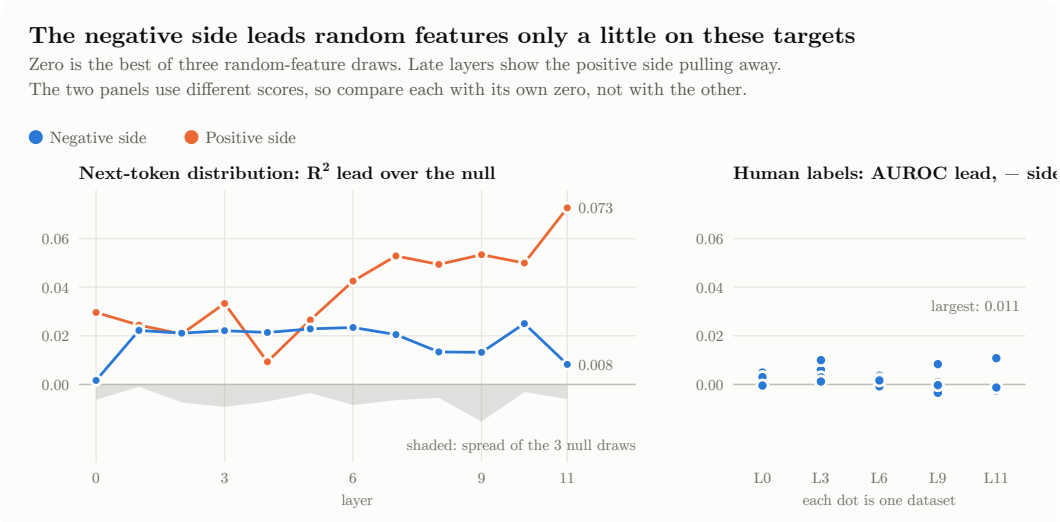

Figure 7. The negative side leads random features only a little on these targets. Zero is the best of three random-feature draws. Left: the R^2 lead on the next-token distribution by layer, with the spread of the three null draws shaded. Right: the negative side's AUROC lead on the human labels. The two panels use different scores, so compare each with its own zero.

10 Does a SwiGLU model behave the same?

The seventh question is whether the picture holds with another activation function. Qwen's MLP is SwiGLU (Shazeer, 2020), so a neuron is silu(gate) × up. We ran the clamps on the same window design, on every layer together and on layers 0, 5, 11, 17 and 23 alone. We sampled the layers because a full per-layer run would take hours. The clean cross-entropy is 2.75 nats (Table 10 and Table 11).

All 24 layers	**Added nats**	**Top prediction unchanged**
Negative side, shuffled null	1.11	50%
Positive side, shuffled null	1.16	49%
Negative side, clamped	2.46	31%
Positive side, clamped	2.46	32%
Negative side, zero	5.44	7%
Positive side, zero	8.54	5%
Every neuron at its mean	8.89	3%

Table 10. The cost of each intervention on Qwen2.5-0.5B, with all 24 layers changed. Added nats is the rise in cross-entropy over the clean 2.75.

Layer	**0**	**5**	**11**	**17**	**23**
Share of tokens on the negative side	0.50	0.50	0.50	0.50	0.50
KL ratio, − / +	0.97	0.96	1.02	1.08	1.08
95% interval	0.90–1.05	0.86–1.07	1.00–1.06	0.94–1.31	1.03–1.14
Activation variance ratio	0.93	0.96	1.00	1.00	1.21
Clamp / null, −	3.7	2.6	1.5	1.9	2.0
Frozen / clamp, −	0.18	0.46	0.51	0.50	1.00

Table 11. Single-layer clamps on five layers of Qwen2.5-0.5B. The ratios are as in Table 4.

The GPT-2 picture does not carry over. The two sides cost the same, whereas GPT-2's negative clamp costs 0.14× its positive clamp. A clamp costs about 2.2× its null over all layers, and up to 3.7× at L0, so the variation's link to the token matters more here. Later MLPs add to the damage at each layer tested up to L17: freezing them lowers the negative clamp's KL to 0.18–0.51×. GPT-2's later MLPs do the opposite after its early layers. The first token matters on the negative side too, at 5.42 nats with it changed against 2.46.

In this SwiGLU model, then, the negative side costs as much as the positive side, which fits its negative poles being sparse (position zero). The low cost in GPT-2 is therefore not a property of negative poles in general. We did not run the decoding or below-dip tests on Qwen. Its activation is a product, so GELU's dip has no direct counterpart.

11 Discussion

The results give a consistent picture of GPT-2's default-on negative poles. Training rests most neurons left of GELU's dip, and the negative side then writes a nearly constant offset that carries most of its cost. Each reading of its variation does about as well on a matched null, and the shared h could explain any lead.

This is not a claim that the negative side is meaningless. The clamp's KL is still 1.9× its null's, and the lead on the next-token distribution has an interval above zero on L1–L11. Other readouts or targets could find more. The claim is narrower: a single default-on pole cannot be audited as a detector, and the population's varying content is small against nulls that the first study lacked.

The finding on cost agrees with work on other models. Tigges et al. (2026) report that mean-ablating the middle half of a neuron's activations costs almost no loss. Our clamp is a mean ablation of one side, and it also costs little, whereas removing the mean costs far more. Their model is a mixture-of-experts SwiGLU model, so the agreement is our reading of two different settings. Haider et al. (2026) find concepts in distinct ranges of a unit. The range below GELU's dip is such a range, but our random-direction null shows that a range can look distinct without learned content. A range-based reading of a unit therefore needs the same null.

The decoding results follow the lesson of the probing literature. Belinkov (2022) and Kutluozen et al. (2026) both argue that a readout needs a random baseline, and our null removes almost all of the negative side's apparent content. Kutluozen et al. (2026) also show that the ridge penalty can change the random arm's score by a large amount. Our readouts use one fixed penalty for both arms, so the small leads of 0.008–0.025 could move under another penalty. Wilam (2026) gives the same warning from an untrained reference, where a probe on the hidden units already reaches a median accuracy of 0.802.

The two models differ in a way that the activation function may explain. In Qwen the sides are symmetric, each is on for half the tokens, and later MLPs add to the damage. In Tigges et al. (2026), about half of all firings are negative, so their SwiGLU model's negative side is also in heavy use. A GELU MLP that rests near its dip is the likely reason for GPT-2's low cost, but one model of each kind cannot settle that.

The rebuild matches the first study on most numbers, to within a few hundredths (Appendix B, Table 12). The cost of holding every neuron at its mean, the KL of zeroing N1046 and N1046's reliability differ. The first study's scripts were not kept, so we cannot say what caused each difference.

For the protocol, no rule changes. These results support the rule that stops the audit of a default-on pole. The reviewer's checklist of interpretability illusions now names the case of a readable tail that a random direction matches. A future population-level unit should face the nulls of this paper too.

12 Limitations

- **One corpus and one sample.** This is 200,000 tokens of `NeelNanda/pile-10k`. The token lists depend on it.
- **Fixed constants at every layer.** All-layer clamps use constants fit on the clean model, although earlier clamps change later layers' inputs.
- **Shuffling is a strong null.** It breaks the correlation between neurons on one token as well as the link to the token. So the ratio of clamp to null does not separate these two.
- **Few controls.** The 10 controls show the range. The population figures rest on L9 alone.

- **Linear readouts and short labelled text.** The decoding tests use linear readouts and one sample. The labelled text is short, and it differs from the Pile sample. The fit sets are smaller than the features would like on the Universal Dependencies data, with 16,476 tokens for 3,072 features. Other readouts or targets could still show more.
- **One model of each kind.** GPT-2 small is the only GELU model, and Qwen2.5-0.5B is the only SwiGLU model. The Qwen per-layer clamps cover 5 of 24 layers.
- **Device.** The runs used float32 on Apple's GPU. Activations there match the CPU's to about 1×10^{-4} of a neuron's peak.
- **The resting level's cause.** The noise-gain test uses small noise on fixed weights, and it cannot show what training selects for.

13 Conclusion

Default-on negative poles in GPT-2 small rest below GELU's dip because training sets their input weights so. As a population, they write a nearly constant offset that the model needs, and a small varying signal that matched nulls almost match. It is therefore safe to stop auditing them one by one, as the protocol does, although a population-level test could still find content. In a SwiGLU model the negative side is as costly as the positive side, so this result is about GELU models that rest near their dip. Why training rests neurons below GELU's dip studies the cause of the rest. Open questions are issues in the Linear project *Default-on poles*, and the research questions are in docs/research-questions.md.

References

Belinkov, Y. (2022). *Probing classifiers: Promises, shortcomings, and advances*. Computational Linguistics, 48(1), 207–219. doi:10.1162/coli_a_00422.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., & Leahy, C. (2020). *The Pile: An 800GB dataset of diverse text for language modeling*. arXiv:2101.00027.

Haider, M. U., Rizwan, H., Sajjad, H., Ju, P., & Siddique, A. B. (2026). *Neurons speak in ranges: Breaking free from discrete neuronal attribution*. Transactions on Machine Learning Research. arXiv:2502.06809.

Hendrycks, D., & Gimpel, K. (2016). *Gaussian error linear units (GELUs)*. arXiv:1606.08415.

Hewitt, J., & Liang, P. (2019). *Designing and interpreting probes with control tasks*. EMNLP-IJCNLP 2019, pp. 2733–2743. ACL Anthology.

Kutluozen, N., Internò, C., & Klindt, D. (2026). *Is a linear probe evidence of a linear representation?* Preprint. arXiv:2610.22655.

McGrath, T., Rahtz, M., Kramár, J., Mikulik, V., & Legg, S. (2023). *The Hydra effect: Emergent self-repair in language model computations*. Preprint. arXiv:2307.15771.

Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., & Zeman, D. (2020). *Universal Dependencies v2: An evergrowing multilingual treebank collection*. LREC 2020, pp. 4034–4043. ACL Anthology.

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

Shazeer, N. (2020). *GLU variants improve transformer*. arXiv:2002.05202.

Sun, M., Chen, X., Kolter, J. Z., & Liu, Z. (2024). *Massive activations in large language models*. COLM 2024. arXiv:2402.17762.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). *Intriguing properties of neural networks*. ICLR 2014. arXiv:1312.6199.

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

Tjong Kim Sang, E. F., & De Meulder, F. (2003). *Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition*. CoNLL 2003, pp. 142–147. arXiv:cs/0306050.

Wilam, P. (2026). *Calibrated concept profiles: Making the reading of a model's neurons a measurement*. Preprint. arXiv:2611.09196.

Xiao, G., Tian, Y., Chen, B., Han, S., & Lewis, M. (2023). *Efficient streaming language models with attention sinks*. ICLR 2024. arXiv:2309.17453.

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.

Zhang, X., Zhao, J., & LeCun, Y. (2015). *Character-level convolutional networks for text classification*. NeurIPS 2015. arXiv:1509.01626.

Data

- `NeelNanda/pile-10k`, revision `127bfedc`: the first 10,000 documents of the Pile. Its card gives the licence as `bigscience-bloom-rail-1.0`.
- `universal-dependencies/universal_dependencies`, revision `c3fed78d`, CC BY-SA 4.0.
- `eriktks/conll2003`, revision `ce85b39f`. Its card gives the licence as `other`.
- `fancyzhx/ag_news`, revision `eb185aad`. Its card gives the licence as `unknown`, for research and other non-commercial use.
- `papluca/language-identification`, revision `e495f9af`. Its card names its sources, and gives no citation and no licence.

The CoNLL and AG News licences limit how we can use their text, and the language dataset's card states no licence. So we publish only aggregate scores.

Appendices

Appendix A: Reproduce

The scripts are `scripts/default_on_poles.py`, `scripts/default_on_clamps.py`, `scripts/default_on_decode.py` and `scripts/default_on_targets.py`. Their output is in `results.jsonl`. `scripts/paper_figures.py` draws the figures from that output, with no model. The data are `corpus.jsonl`, the 200,000-token sample with seed 0 described in the Method. The human labels come from `openinterpretability/concepts.py`. The comparison of MPS with the CPU is in `openinterpretability/cli.py`. The `interpretability-illusions-review` skill now names the case of a readable tail that a random direction matches.

The bands command writes the controls in its `band_neuron` rows, and the clamps take them through `--neurons`. Each command appends to its `--out`.

```
OUT=papers/default-on-poles/results.jsonl
uv run python scripts/default_on_poles.py rest --out $OUT
uv run python scripts/default_on_poles.py bands --out $OUT
uv run python scripts/default_on_decode.py --out $OUT
uv run python scripts/default_on_targets.py next-token --out $OUT
uv run python scripts/default_on_targets.py concepts --out $OUT
uv run python scripts/default_on_clamps.py --model openai-community/gpt2 \
    --neurons 1046,918,1132,1622,1742,1965,2084,2158,2312,2455,2778 --out $OUT
uv run python scripts/default_on_clamps.py --model Qwen/Qwen2.5-0.5B \
    --layers 0,5,11,17,23 --out $OUT
uv run python scripts/paper_figures.py
```

Appendix B: Against the first study

The first study's scripts were not kept, so we cannot say what caused each difference. Table 12 sets its numbers beside ours.

Number	First study	Here
L9 median h, trained / random init	-1.03 / 0.00	-1.04 / 0.00
L9 default-on, trained / random init	97% / 40%	97% / 40% (5 seeds)
L9 mean GELU slope, trained / random init	0.20 / 0.51	The same, as mean |slope|. The signed trained mean is 0.09
Clean cross-entropy	3.62	3.63
Negative clamp: added nats, top unchanged	0.21, 76%	0.19, 77%
Positive clamp: added nats, top unchanged	1.33, 41%	1.34, 40%
Negative zero: added nats, top unchanged	3.68, 13%	3.72, 13%
All at mean: added nats, top unchanged	4.94, 10%	4.09, 13%. With first tokens changed: 5.63, 9%
Zeroing L9/N1046 alone, KL	2.6×10^{-5}	1.3×10^{-5}
KL ratio / variance ratio at L3, L6, L9, L11	0.25/0.31, 0.22/0.20, 0.12/0.13, 0.05/0.04	0.26/0.31, 0.21/0.21, 0.11/0.13, 0.05/0.04
The same at L0	0.07/0.14	0.06/0.14
L9/N1046 tokens below the dip	2.3%	2.4%
L9/N1046 reliability	0.80	0.70 by document, 0.90 by occurrence
Reliability of 10 controls	0.79–0.93	0.56–0.83 by document, 0.66–0.92 by occurrence
ρ with surprisal	0.05	0.05

Table 12. The first study's numbers, from an earlier `TODO.md`, beside this paper's.

Reliability depends on the split. A split by occurrence puts one document's tokens in both halves, so the shared context raises the number. We report the split by document. Our controls are a different draw from the first study's.

Appendix C: The gain on noise

We tested whether a low slope at rest lowers the gain on input noise. We compared each MLP with a copy that has each neuron's median h moved to zero by its bias. The noise

has $0.1\times$ each input dimension's SD. It is isotropic, or it points towards another token's input, which we call *data-shaped* noise. Table 13 gives the gains.

Layer	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Noise gain falls by ($\times$)	5.5	8.2	22.0	11.3	6.4	6.2	4.9	4.6	4.5	4.3	3.8	3.3
Signal gain falls by ($\times$)	7.3	15.7	23.5	19.9	12.9	11.8	9.3	7.1	6.6	5.7	3.5	2.4
Signal/noise gain, trained	1.31	0.23	0.29	0.36	0.35	0.39	0.43	0.50	0.55	0.64	0.94	2.23
Signal/noise gain, recentred	1.73	0.44	0.30	0.64	0.71	0.73	0.82	0.79	0.80	0.84	0.86	1.62
Signal/data-shaped gain, trained	0.35	0.15	0.09	0.09	0.09	0.11	0.14	0.17	0.21	0.25	0.28	0.28
Signal/data-shaped gain, recentred	0.80	0.59	0.30	0.26	0.30	0.35	0.44	0.44	0.50	0.54	0.40	0.34

Table 13. The gain of each MLP on its signal and on small input noise, as trained and in a copy recentred so that each neuron's median h is zero. The first two rows and the first ratio use isotropic noise. A higher ratio means that the MLP passes more signal per unit of noise.

The gain on noise does fall at rest, by 3–22$\times$. But the signal's gain falls by more on L0–L9, so the trained signal-to-noise ratio is lower there than the recentred one. Only L10 and L11 are higher, and only for isotropic noise. For data-shaped noise the trained ratio is lower on all 12 layers. So a lower gain on noise is true as a fact about size, but this test does not support it as a reason for the rest. The test also has limits. It uses small noise on fixed weights, and training would not reach the recentred copy, so it cannot show what training selects for. A causal test would train small models and vary the dropout, the weight decay or the activation function.