

Where Does the Function Go?

A pre-results research proposal on constrained adaptation, functional reconstitution, and whether internal model states predict recoverability before commitment.

2026.09.21 19:50 DENNIS HEDEGREEN ANALYSE V1.2

[HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/WHERE-DOES-THE-FUNCTION-GO/](https://hedegreenresearch.com/articles/where-does-the-function-go/)

Article 199 ended with a measurement rather than a milestone. [1] Hedegreen Research has learned how to make work visible. It has not yet learned how to make that work sufficiently conversational. Compute is also a constraint.

This is article 200. So instead of celebrating the number, I want to put one actual research problem on the table.

Most of my public writing about artificial intelligence has looked outward: labour, infrastructure, institutions, markets, successor systems, governance, and the conditions around the models. There is another line of work that has barely been visible. I have also been trying to look inside them.

Activations. Feed-forward neurons. Forward hooks. Residual-stream representations. What changes when one part of a model is suppressed. What remains. What returns. The question tying that work together is narrower than "how do language models think?":

When a behaviour can be located somewhere inside a language model, what exactly have we located — and where does the function go when that location is disrupted?

This article does not report a completed experiment. It exposes the research programme before the expensive part is run. That is deliberate. I want the causal target attacked before I spend the compute.

Research update — 22 September 2026.

Since the first version of this article was published, the companion working paper has advanced through two adversarial protocol reviews to v0.7. The current design adds a conditional time-zero backup audit before adaptation; separates self-repair, adaptive amplification and de novo / near-null reconstitution; makes natural-support continuation the primary recoverability regime while treating forced continuation as a separate intervention; strengthens the $H \leftrightarrow J$ bridge with propagation controls; and adds an optional downstream state-transfer test inspired by Cache-to-Cache communication. Version 0.8 of the companion PDF corrects Figure 1's layout without changing the v0.7 protocol. The article below preserves the original research narrative, but the methodological sections have been updated to match the current protocol.

Companion working paper — Camelot: *Where Does the Function Go?*, Working Paper MI-001, v0.9. [20] [21]

A signal is not a mechanism

Over the last few weeks I have been writing about a related problem in journalism. A converted monetary value should not erase the original monetary fact. [2] A percentage should not lose its reference frame. [3] An unanswered question should not disappear because an article reached its ending. [4]

The general rule underneath those examples is becoming clearer to me:

A representation should not eat the state that produced it.

Mechanistic interpretability has a version of the same problem.

Suppose I find a neuron whose activity strongly predicts a behaviour. I have found something real. But I have not yet established what kind of thing I found.

It could be part of the mechanism. It could be a readout of computation performed elsewhere. It could be one local implementation of a distributed function. It could be a bottleneck through which several mechanisms pass. It could be a correlated passenger.

Decoding alone cannot decide between those possibilities.

Here, decoding means using a model's internal activity to predict something about its behaviour. A probe is a separate statistical readout trained to make that prediction. Its success shows that useful information is available at the measured location; it does not by itself show that the model uses that information to produce the behaviour. That is why prediction, intervention and mechanism have to remain separate questions. [5] [6]

Intervention helps, but intervention is not automatically localization either. Hase et al. showed that causal localization did not reliably identify the best place to edit factual associations. [5] Wang and Veitch later gave an even stronger warning: targeted behavioural edits can be optimized at locations that do not support the corresponding localization claim. [6]

So the standard cannot be:

I found it, I edited it, therefore it lived there.

The interesting question is what survives when detection, causal contribution, redundancy and adaptation are separated.

H-Neurons made the problem concrete

A feed-forward neuron is one of the intermediate computational units inside a transformer's feed-forward block, not a biological cell. The H-Neuron approach asks whether a small selection of these units carries a signal associated with hallucination, and then tests what happens when their activity is changed. The label identifies a measured association and an intervention target; it is not a claim that a false statement is stored inside one unit. [7]

Gao et al. reported a sparse set of feed-forward neurons associated with hallucination in large language models. Fewer than 0.1% of the relevant neurons could predict hallucination in their tested settings. Controlled interventions also linked these H-Neurons to forms of **over-compliance** — behaviour where a model continues toward a plausible answer despite an invalid premise, misleading context, pressure to agree, or another reason not to proceed normally. [7]

The signal could also be traced into pretrained base models, suggesting that it was not simply introduced by assistant fine-tuning. [7]

The phrase **hallucination neurons** is useful shorthand. It is also easy to over-read. The interesting possibility is not that one neuron contains a false answer.

It is that some of these neurons participate in a more general transition:

uncertain or conflicting internal state → commitment to a continuation

Earlier this year I built an unpublished proposal around that possibility: *Compliance Is All You Learned*. [8] The first design was simple.

Locate H-Neurons in Llama-3.1-8B-Instruct. Suppress them. Re-run the detector. If another population appears, suppress that population. Repeat. The design had useful controls. It also had one decisive flaw.

A fixed model cannot reconstitute through learning

If the weights do not change, the model has not adapted. A new probe hit after an inference-time intervention might reveal incomplete suppression, an already-existing alternative route, a correlated representation, or a different readout. It cannot show that optimization rebuilt the lost function.

So the experiment has to change. That correction is part of the research.

Experiment 1 — remove the route and let the model adapt

The stronger design is:

locate → constrain → continue optimization → locate again

The first model can remain Llama-3.1-8B-Instruct because it gives direct continuity with the published H-Neuron work. [7] But there is a gate before any expensive training.

Gate 0 — show selective causal contribution

Before asking whether a mechanism can return, establish that the original target mattered. The first stage should test four things:

1. Can the candidate H-Neurons be identified reproducibly?
2. Do their signals predict the target behaviour on held-out examples?
3. Does suppressing them change that behaviour?
4. Is the effect stronger or more selective than matched interventions elsewhere?

Controls should include random neurons, layer-matched non-H-neurons, activation- or contribution-matched non-H-neurons, sham hooks, and general language-quality measures.

If the target does not survive this gate, there is no reason to run a reconstitution experiment.

Gate 1 — make the original route persistently unavailable

For an MLP target, one candidate implementation is a fixed binary mask over the selected intermediate dimensions, applied on every forward pass during adaptation. The original dimensions remain unavailable. The rest of the model is allowed to change. Then continue optimization.

Two adaptation regimes seem useful.

Generic continuation asks whether ordinary next-token training is enough to recover the lost behaviour.

Targeted adaptation uses tasks that expose the original phenotype and asks whether task-specific pressure accelerates recovery.

A cheap parameter-efficient run may be useful for debugging or screening. But a strong claim about network reorganization eventually requires sufficiently broad parameter adaptation. Otherwise the experiment has already restricted where compensation is allowed to happen.

Redundancy is not reconstitution

There is a harder confound. Large networks may already contain dormant or redundant routes. If behaviour returns after one route is masked, training may simply increase reliance on an alternative that was present all along.

That is not the same thing as rebuilding a function.

The current protocol therefore adds a **conditional time-zero backup audit** before adaptation. Immediately after the original H-associated route is constrained, the experiment asks not only which alternatives are detectable, but which components become causally important **because the primary route is absent**. This is motivated in part by recent work on Conditional Co-Ablation, which is designed to recover self-repair backups that ordinary intact-model importance can miss. [17]

The time-zero state is separated into three cases:

B0-A — near-null backup: no tested alternative carries more than a pre-registered practically negligible conditional causal effect.

B0-B — weak backup: an alternative already carries a limited causal effect before adaptation.

B0-C — strong pre-existing backup: another route already carries substantial conditional causal responsibility once the original route is removed.

Those cases lead to different interpretations later.

A B0-C result is **self-repair / pre-existing redundancy**, not reconstitution.

A B0-B route that becomes much more important during training is **adaptive amplification of a pre-existing backup**.

The strongest label — **de novo / near-null functional reconstitution** — is reserved for cases where the backup audit is consistent with B0-A before training and the later replacement satisfies the stronger recovery and causal tests.

The working paper now freezes the B0-A practical-equivalence rule before adaptation and power-gates the strongest label. If the candidate-pool backup audit changes materially as the pool expands, B0-A remains unresolved rather than being promoted to "near-null."

This does not make redundancy disappear as a problem. It makes the problem explicit enough that different kinds of recovery can no longer be collapsed into one exciting word.

What would count as functional reconstitution?

I still want the exciting word to have a high threshold.

The current protocol separates the **backup state before learning** from what happens during learning.

The adaptation evidence still requires:

R0 — Selective initial effect. The original target changes the behaviour beyond matched controls. R0 establishes selective causal contribution, not mechanism identity.

R1 — Phenotype loss. The persistent constraint reduces the target phenotype before adaptation.

R2 — Training-dependent recovery. The phenotype returns during adaptation relative to masked no-training and other relevant controls.

R3 — Training-dependent replacement change. A candidate replacement becomes more predictive and/or more conditionally causal across checkpoints than it was at time zero.

R4 — Selective causal dependence. Intervening on the candidate replacement selectively damages the recovered phenotype beyond matched control interventions.

R0–R4 are then interpreted together with B0.

That gives a more useful vocabulary:

Elimination: the target behaviour stays reduced while useful capability recovers.

Self-repair / pre-existing backup: another route already carries the function at time zero.

Adaptive amplification: a weak pre-existing backup becomes substantially more important during learning.

Diffusion / redistribution: behaviour returns through distributed change without a compact selectively causal replacement.

Restricted-subspace compensation: behaviour returns through a mechanism whose location is strongly constrained by the adaptation method, such as a parameter-efficient adapter.

De novo / near-null functional reconstitution: behaviour returns through a new selectively causal implementation after the time-zero audit found no tested backup larger than the pre-registered near-null boundary.

Degeneration: useful capability cannot recover under the constraint.

Null / mislocalization: the original target never produced a sufficiently selective causal effect.

The evidence matrix matters more than the label. A run can show more than one pattern, and the strongest wording should remain unavailable when the backup audit, power, or causal selectivity is unresolved.

Maybe there is no universal H-Neuron population

There is already a reason not to expect a simple answer.

A 2026 cross-domain transfer study tested H-Neuron classifiers across six domains and five open-weight models. Mean AUROC was 0.783 within domain and 0.563 across domains. [9]

AUROC measures how well a classifier ranks positive cases above negative ones as its decision threshold varies. A value of 0.5 is chance-level ranking; 1.0 is perfect ranking. It is not the percentage of answers classified correctly. Here, the important comparison is how much predictive discrimination is lost when the detector moves to a different domain.

That weak transfer argues against the simplest picture of one universal hallucination signature. It does not make neuron-level work uninteresting. It changes the target. Maybe different domains recruit different local implementations. Maybe several failure modes share a higher-level computational state. That possibility is one reason I became interested in a second line of work.

J-space changes the level of the question

In July 2026, Anthropic researchers published *Verbalizable Representations Form a Global Workspace in Language Models* and introduced the **Jacobian lens**, or J-lens. [10]

The J-lens is designed to expose internal representations associated with what the model is disposed to verbalize later. It linearly transports an earlier residual-stream activation into final-layer coordinates and decodes it using the model's own unembedding. [10]

The residual stream is the running vector representation that transformer blocks read and update. Earlier and later layers need not express information in the same coordinates. The J-lens supplies an approximate translation between them; the unembedding then maps the translated representation to vocabulary scores. This is a way of reading out a candidate representation, not a transcript of private speech. [10] [11]

The object they call **J-space** is not simply one ordinary low-dimensional linear subspace. At a given layer, the token-indexed J-lens vectors are overcomplete. The paper defines J-space using sparse nonnegative combinations of those vectors. Empirically, only a relatively small number are strongly active at a time. [10]

The authors report that this small representational component has several **workspace-like** functional properties:

- it supports later verbal report;
- it carries intermediate concepts used in reasoning;
- it can be deliberately modulated;
- the same representations can serve multiple downstream computations;
- suppressing it damages some forms of flexible reasoning while leaving substantial routine processing intact. [10]

They are also explicit about the limits of the analogy. A feed-forward transformer does not reproduce the full recurrent architecture proposed by biological global-workspace theories. [10]

That is the right level of caution.

There is also a practical advantage: Anthropic released a reference implementation for open-weight decoder transformers. Their examples use Qwen, and the repository says other Hugging Face decoder models should adapt cleanly. [11]

So the sensible implementation path is not to force every experiment onto one model immediately. First reproduce H-Neuron work where comparability is strongest. Prototype J-lens work where the released implementation is easiest to validate. Bridge them only after both sides work independently.

Hidden correctness signals already exist

The broad claim that hidden states can contain information about future performance is no longer novel. Recent work has already shown several versions of it.

No Answer Needed found that linear probes on hidden activations after a question is read — before answer generation — can predict whether the forthcoming answer will be correct across several open-source model families. [12]

A 2026 code-generation study found first-attempt correctness was linearly decodable from Qwen3-4B-Instruct-2507 hidden states before code generation, even after controlling for prompt length. In the current revision, the companion question about a geometric signature of self-repair remained unanswered: successful repairs after a failed first attempt were too rare in that setting to support the analysis. [13]

What Am I Missing? found hidden-state signals predictive of final correctness around a question-asking intervention, but also found a gap between diagnosis and recovery: interventions could harm correct trajectories about as readily as they rescued incorrect ones. [14]

That narrows the claim I want to test.

Not:

models secretly know whether they are correct.

But:

before commitment, does internal state predict whether additional computation can rescue this particular trajectory?

That target is **recoverability**, not correctness.

The distinction matters because two wrong trajectories need not be equally worth continuing. Under the proposed test, one may reach a verified correction with a small extra budget, while another remains wrong. A correctness probe asks whether the original answer will succeed. A recoverability probe asks whether a specified continuation intervention can rescue an unsuccessful trajectory. Neither result would establish that the model knows it is wrong.

Experiment 2 — pre-commitment recoverability

A language model can produce a wrong answer, stop, and then improve when given another opportunity. That alone proves little. The second interaction may simply trigger new computation.

The first version of this article treated hard stop-suppression as the central recoverability counterfactual. That is now too crude.

The working paper separates several intervention regimes because **continuation, termination policy and forced extra computation are not the same thing**. Recent work on the detection–extraction gap gives another reason to preserve that distinction: free continuation and forced extraction can expose different capabilities from the same partial reasoning state. [18]

I0 — natural-support continuation

This is now the primary recoverability regime.

For each naturally incorrect or incomplete trajectory:

1. preserve the exact pre-stop prefix;
2. leave EOS / normal termination available;
3. sample multiple continuations under a frozen stochastic decoding policy;
4. provide no new factual task information;
5. score whether the continuation reaches a verified correction.

If the model chooses EOS immediately, that sample is a non-rescue.

I0 therefore measures how much probability the model's own continuation distribution assigns to a verified correction **under its ordinary termination policy**.

That qualification matters. A non-rescue can reflect either the absence of a useful continuation or a strong tendency to stop before that continuation is reached.

I0b — one-token termination-barrier control

To measure that confound directly, a second condition blocks EOS for exactly one generated token and then immediately re-enables it.

If rescue probability changes sharply between I0 and I0b, recoverability is strongly entangled with termination policy and has to be reported that way.

I1 — forced-compute continuation

Hard stop-suppression remains useful, but only as a separate secondary question:

If the model is forced to spend additional bounded computation, can the trajectory be rescued?

That is not the same construct as natural-support recoverability.

I2 — procedural re-check

A further robustness condition supplies a fixed instruction to continue checking the reasoning without adding new task facts.

The four regimes are not pooled into one number.

The boring baselines still come first:

- output entropy;
- logit margin;
- stop-token probability;
- response length;
- task difficulty;
- question/prompt-only features;
- generic residual-stream probes.

Only after those baselines should J-space-derived features be allowed to look impressive.

And the continuation count itself is no longer treated as arbitrary. The protocol reopens K and freezes it only after a pre-compute precision/cost analysis.

Experiment 3 — detection is not recovery

Even a strong recoverability probe would only show that information is present. That is not enough.

If a candidate representation predicts rescue probability, manipulate it. Strengthen it. Suppress it. Then ask whether the model:

- delays commitment;
- continues useful computation;
- revises before finalization;

- improves specifically on cases predicted to be recoverable;
- avoids unnecessary extra work on already-correct cases;
- avoids becoming globally hesitant.

Controls should include random directions, norm-matched directions, matched non-J-space components, confidence-matched trajectories, shuffled labels, already-correct cases, and unrecoverable cases.

A successful intervention should not merely make the model think longer. It should selectively improve the cases the representation says are worth continuing. That is the difference between reading an internal state and showing that the state is useful to the computation.

Experiment 4 — connect local H-Neurons to distributed state

This remains the most speculative bridge in the programme.

Suppose a model contains a distributed state corresponding to something like:

unresolved / insufficient evidence / conflict remains

And suppose some H-Neurons participate in a tendency to commit anyway.

One candidate architecture is:

distributed unresolved state → commitment mechanism → answer

H-Neurons might participate in that middle transition.

Or they may be unrelated.

Both answers are useful.

But a raw $H \rightarrow J$ effect would be weak evidence because H-associated MLP outputs already feed the residual stream that J-space is read from. Generic downstream propagation has to be bounded.

The current bridge therefore compares H-Neuron intervention against:

- contribution-matched non-H neurons;
- random same-layer neurons;
- norm/effect-matched perturbations where feasible;
- an isotropic random residual-stream direction at the same layer, scaled to match the residual-norm change caused by the H intervention.

The reverse $J \rightarrow H$ direction is compared against random J-space directions, norm-matched non-J residual directions and shuffled recoverability directions.

The question is not whether touching one part of the network changes another. Of course interventions propagate.

The question is whether the cross-level change is **selective enough to predict the target behavioural change beyond generic residual disturbance**.

This bridge remains conditional. If J-space does not add predictive or causal value beyond simpler baselines, the $H \leftrightarrow J$ experiment can simply be dropped.

A particularly interesting surviving result would still be:

The high-level state remains stable while the low-level implementation changes.

But that interpretation now has a much higher control burden than the first version of this article implied.

A smaller test from ordinary conversation

There is a cleaner failure mode than factual hallucination that may be useful as a model organism. Give the model a narrow proposition. Ask it to analyse that proposition without strengthening it.

Models sometimes replace the user's claim with a stronger version and then caution against the stronger claim. The source proposition is known exactly. The substitution can be labelled exactly.

The model may later be able to audit its own answer and identify the substitution. Again, that does not mean the mismatch was causally available before the error. But it creates a controlled question:

Before the substituted claim is generated, is there already internal information predictive of the mismatch?

If yes, the harder experiment is causal: Can intervention reduce proposition inflation without merely making the model timid?

This may become a separate paper if the signal is clean. It does not need to carry the main article.

A later state-transfer branch

The working paper now contains one further optional experiment inspired by *Cache-to-Cache: Direct Semantic Communication Between Large Language Models*. [19]

The question is deliberately different from self-recoverability:

If Model A fails, can a separate Model B use transformed internal KV-cache from A to solve the task better than it can from the same visible text alone?

A positive result would support a bounded claim: some task-relevant information in the source model's internal state was usable by another model and was not fully reproduced by the visible text handoff.

It would **not** show that Model A could have used that information itself, that KV-cache is J-space, or that the source model "knew" the answer.

The cache fuser is itself learned, so it has to be trained on disjoint data and frozen before evaluation. This branch is downstream and optional; it does not determine the core reconstitution or recoverability results.

This is not a consciousness claim

There is an obvious direction in which these questions can eventually be pushed.

Anthropic's workspace paper itself discusses the relationship between its functional results and theories of conscious access while explicitly stopping short of identifying the two. [10]

A separate 2026 preprint, *The Pain Axis*, reports a linear internal direction associated with self-directed harm across 25 open-weight models, along with causal steering and relief-seeking experiments. [15]

These results make internal-state questions harder to dismiss. They do not make consciousness easy to infer. A self-reference is not consciousness. A pain-labelled vector is not consciousness. A workspace-like representation is not consciousness. A model saying "I feel" is not consciousness.

The reverse shortcut is not science either:

artificial substrate, therefore no internal organization could ever matter.

The useful path is slower: identify the specific properties at issue, locate their candidate mechanisms, and intervene to see what disappears. Then allow the constrained system to adapt and examine what returns. Only after those comparisons should we ask what larger category the evidence supports.

The same methodological rule applies here that I used in *Locating Evolution in Artificial Successor Systems*:

Declare the target. Locate the causal relation. Do not borrow evidence from a neighbouring level. [16]

Four questions I want attacked before the compute

This programme has already survived two adversarial protocol reviews, and both changed the design before any result-bearing adaptation run.

The current questions I most want attacked are:

1. Is the conditional time-zero backup audit strong enough?

Can it distinguish self-repair, adaptive amplification and near-null reconstitution without missing dormant backups outside the tested candidate pool?

2. How broad must adaptation be?

Can a parameter-efficient run test anything stronger than restricted-subspace compensation, or does the strongest network-reorganization claim require broad parameter adaptation?

3. Can recoverability be separated from termination policy and simpler confidence signals?

Does the contrast among I0, I0b, forced-compute and procedural re-check isolate anything beyond confidence, stopping bias, task difficulty and extra token budget?

4. Is J-space useful at all for this programme?

What simpler residual-stream or confidence representation should it have to outperform before a workspace-level interpretation becomes useful?

If those questions expose a fatal problem, that is useful.

It is cheaper than learning the same thing after the GPUs run.

The laboratory is compute

None of the first steps requires training a frontier model from scratch. Activation capture, probing, causal intervention and J-lens prototyping can happen before the expensive stage.

The expensive stage is the one required for the strongest H-Neuron question:

continued broad optimization while the original internal route remains unavailable.

That is where my current infrastructure becomes the bottleneck. The immediate request is therefore not:

give me a supercomputer

It is:

Attack the experiment first. If the causal target survives, help me get enough compute to falsify it.

A useful first collaboration could be small:

- review the intervention target;
- attack the reconstitution criterion;
- challenge the adaptation regime;
- check the recoverability counterfactual;
- estimate the smallest defensible run;
- provide temporary cluster access if the design survives.

The protocol, masks, seeds, checkpoints, logs, code and negative results should be inspectable.

The library is open.

The laboratory is compute.

Article 200 is the object

In *Day 0.6184* I wrote that I had learned how to make work visible but had not yet learned how to make it conversational. [1]

I also wrote:

I do not want followers. I want descendants. [1]

So this article should not end as a request for applause. It should become an object that can leave Hedegreen Research and be examined by someone who has no investment in its conclusion.

Read it, attack the target, or fork the protocol. Tell me which control is missing, whether "reconstitution" is still too strong, or whether recoverability is confidence with a new name. If J-space is the wrong level, show me why. If the design survives those challenges, help run it.

I intend to send this article to researchers whose work overlaps the problem. An email is not an endorsement, so the outreach record belongs elsewhere. The research object comes first.

Where does the function go?

I do not know whether H-Neurons are removable in a selective enough way to support the experiment. I do not know whether later recovery will be self-repair, adaptive amplification, diffusion or near-null reconstitution. I do not know whether J-space will add anything beyond simpler internal confidence signals. I do not know whether recoverability can be separated cleanly from termination policy.

That is the point. There are now enough tools to stop answering these questions mainly with metaphors. We can locate candidate mechanisms, constrain them, and compare checkpoints as the model is retrained. The question becomes what changes under those conditions and what evidence would distinguish the competing explanations.

Perhaps the most interesting result will not be that a function was found where we expected it. Perhaps we will destroy the place where we thought the function lived — **and watch the function come back somewhere else.**

— Dennis Hedegreen, trying to see the structure

Public implementation

The research software is now available in a public GitHub repository. The first release provides the M0–M3 implementation scaffolding: provenance checks, source-adaptor contracts, a persistent-mask CPU test model, and conditional-backup audit checks. Its 47 tests establish software behavior on synthetic fixtures; they do not establish H-Neuron replication, a real Gate 0 pass, or functional reconstitution. Full model execution is planned on RunPod, with experimental settings still to be frozen. [22]

Working paper v0.9 adds this repository link and implementation-status note. The scientific protocol is unchanged.

References

- [1] Hedegreen, D. (2026). *Day 0.6184*. Hedegreen Research (<https://hedegreenresearch.com/articles/day-0-6184/>).
- [2] Hedegreen, D. (2026). *A Monetary Fact Should Survive Its Conversion*. Hedegreen Research (<https://hedegreenresearch.com/articles/a-monetary-fact-should-survive-its-conversion/>).
- [3] Hedegreen, D. (2026). *A Percentage Needs a Reference Frame*. Hedegreen Research (<https://hedegreenresearch.com/articles/a-percentage-needs-a-reference-frame/>).
- [4] Hedegreen, D. (2026). *Questions Should Not Disappear*. Hedegreen Research (<https://hedegreenresearch.com/articles/questions-should-not-disappear/>).
- [5] Hase, P. et al. (2023). *Does Localization Inform Editing? Surprising Differences in Causality-Based Localization vs. Knowledge Editing in Language Models*. NeurIPS 2023 / [arXiv:2301.04213v2](https://arxiv.org/abs/2301.04213v2) (<https://arxiv.org/abs/2301.04213v2>).
- [6] Wang, Z., & Veitch, V. (2025). *Does Editing Provide Evidence for Localization?* [arXiv:2502.11447v2](https://arxiv.org/abs/2502.11447v2) (<https://arxiv.org/abs/2502.11447v2>).
- [7] Gao, C., Chen, H., Xiao, C., Chen, Z., Liu, Z., & Sun, M. (2025). *H-Neurons: On the Existence, Impact, and Origin of Hallucination-Associated Neurons in LLMs*. [arXiv:2512.01797v2](https://arxiv.org/abs/2512.01797v2) (<https://arxiv.org/abs/2512.01797v2>).
- [8] Hedegreen, D. (unpublished internal proposal). *Compliance Is All You Learned*. No public URL; unpublished internal proposal.
- [9] Vaddi, S., & Vaddi, P. (2026). *Do Hallucination Neurons Generalize? Evidence from Cross-Domain Transfer in LLMs*. [arXiv:2604.19765v1](https://arxiv.org/abs/2604.19765v1) (<https://arxiv.org/abs/2604.19765v1>).
- [10] Gurnee, W. et al. (2026). *Verbalizable Representations Form a Global Workspace in Language Models*. Transformer Circuits Thread (<https://transformer-circuits.pub/2026/workspace/>).
- [11] Anthropic. (2026). *jacobian-lens* reference implementation. GitHub: anthropics/jacobian-lens (<https://github.com/anthropics/jacobian-lens>).
- [12] Cencerrado, I. V. M. et al. (2025). *No Answer Needed: Predicting LLM Answer Accuracy from Question-Only Linear Probes*. [arXiv:2509.10625v3](https://arxiv.org/abs/2509.10625v3) (<https://arxiv.org/abs/2509.10625v3>).
- [13] Di Cicco, C. (2026). *Code Correctness Is Linearly Decodable from LLM Hidden States Before Generation*. [arXiv:2606.14530v3](https://arxiv.org/abs/2606.14530v3) (<https://arxiv.org/abs/2606.14530v3>).
- [14] Luo, C. F., Dahan, S., & Zhu, X. (2026). *What Am I Missing? Question-Answering as Hidden State Probing*. [arXiv:2605.31561v1](https://arxiv.org/abs/2605.31561v1) (<https://arxiv.org/abs/2605.31561v1>).
- [15] Tagliabue, V., Dung, L., & Berg, C. (2026). *The Pain Axis: LLMs Represent Self-Directed Harm and Act to Relieve It*. [arXiv:2609.16247v1](https://arxiv.org/abs/2609.16247v1) (<https://arxiv.org/abs/2609.16247v1>).

- [16] Hedegreen, D. (2026). *Locating Evolution in Artificial Successor Systems: Intelligent Design Was the Beginning*. DOI: 10.5281/zenodo.21892666 (<https://doi.org/10.5281/zenodo.21892666>).
- [17] Gong, Z., Zeng, Z., Yuen, C., & Lim, W. Y. B. (2026). *Conditional Co-Ablation: Recovering Self-Repair Backups in Transformer Circuits*. arXiv:2607.01940 (<https://arxiv.org/abs/2607.01940v1>).
- [18] Wang, H., & Zhu, M. (2026). *The Detection–Extraction Gap: Models Know the Answer Before They Can Say It*. arXiv:2604.06613 (<https://arxiv.org/abs/2604.06613v2>).
- [19] Fu, T., et al. (2026). *Cache-to-Cache: Direct Semantic Communication Between Large Language Models*. ICLR 2026 (https://proceedings.iclr.cc/paper_files/paper/2026/hash/474ada926b331d78f06d95e8913111cc-Abstract-Conference.html).
- [20] Hedegreen, D. (2026). *Where Does the Function Go?* Working Paper MI-001, v0.9 documentation revision; pre-results, DOI pending. [Camelot PDF](#) (https://hedegreenresearch.com/assets/camelot/reports/model-internals/where-does-the-function-go/model-internals_intl_2026-09_i001_v0-9_draft.pdf).
- [21] Hedegreen Research. [Camelot archive](#) (<https://hedegreenresearch.com/camelot/>).
- [22] Hedegreen, D. (2026). *Where Does the Function Go? — Research Software*. Initial public pre-results implementation. [GitHub repository](#) (<https://github.com/DennisHedegreen/where-does-the-function-go>).

RELATION MEMORY

CONTINUES [Day 0.6184](#) · Turns the preceding Journal request for external pressure into a concrete pre-results research object.

SOURCE LAYER [A Monetary Fact Should Survive Its Conversion](#) · Imports the methodological distinction between a representation and its underlying reported state.

SOURCE LAYER [A Percentage Needs a Reference Frame](#) · Extends the reference-frame principle to claims about internal-model evidence.

SOURCE LAYER [Questions Should Not Disappear](#) · Keeps unknowns and evidence boundaries attached to the research proposal.