Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Where the Answer Is Born: Causal Interventions on the Information Trajectory of a Quantized Hybrid Language Model

Wenlong Li

77247592@qq.com

Independent Researcher

Research Article

Keywords: mechanistic interpretability, gated recurrent network, causal intervention, activation patching, information trajectory, quantized hybrid language model, driven criticality

Posted Date: September 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11201314/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Where the Answer Is Born: Causal Interventions on the Information Trajectory of a Quantized Hybrid Language Model

Wenlong Li
independent researcher, Shanxi, China
77247592@qq.com

Version 1.0, 29 September 2026

Abstract

A descriptive walkthrough can show *where* information travels inside a language model, but it cannot on its own show which components actually *cause* the output. This manuscript reports the causal follow-up to a node-by-node white-box walkthrough of a quantized hybrid language model. Using the same self-authored, node-readable inference engine—whose agreement with the independent implementation llama.cpp was quantified in the companion paper—we run a pre-registered battery of interventions on a fixed set of questions: bidirectional activation patching, block-by-block ablation, per-memory-head ablation, forget-gate clamping, distance-resolved tests, identity-direction erasure, a minimal single-point patch, and tests of behavioral breadth. The main subject is a 24-layer, 0.8B-parameter Q4_K hybrid that interleaves 18 Gated DeltaNet blocks with 6 global-attention blocks; the key localization result is replicated on a 64-layer, 27B-parameter build of the same family.

The interventions converge on a single causal picture. The middle of the network is distributed and redundant: ablating any single block among blocks 1–21, or any one of the 16 memory heads, leaves the answer largely intact, and no single head is individually necessary. The final decision is instead localized: bidirectional patching crosses over at layer 22, a pre-registered single-point patch at layer 22 restores the answer from probability 0.246 to 0.677 while patching the adjacent layer 21 does not, and the answer is read out in the final attention block. The forget gates have a causal role that is narrow in a precise sense: clamping every gate to zero (full forgetting) collapses the answer probability to 0.003 and clamping to one (no decay) collapses it to 0.0001, whereas the learned near-unity gates support 0.658. Early surface-form identity is causally decoupled from the final output: erasing an identity direction in the first few blocks has no effect, while erasing the clean-versus-corrupt difference direction at a late block reduces the answer to 0.501 and an equal-size random direction does not. Long-range dependence is carried predominantly by the global-attention look-back rather than by the gated recurrence.

We introduce no new architecture and no new probe metric. The intervention primitives—activation patching, ablation, clamping, and direction erasure—are the established causal tools of mechanistic interpretability, which we cite rather than re-derive; the contribution is a complete, reproducible causal record of how one quantized hybrid produces an answer, with every negative and positive result reported and the scope of the claims stated explicitly.

Keywords: mechanistic interpretability; causal interventions; activation patching; ablation; Gated DeltaNet; state-space models; global attention; quantization.

1 Introduction

1.1 From a descriptive path to causal claims

The companion paper *From Input to Output* [24] records, operator by operator, the path a single token follows from the embedding to the emitted token, on an instrument calibrated against llama.cpp. That record is descriptive: it establishes where information is present and in what form, and it makes the forward path fully inspectable. But presence along a path is not the same as causal responsibility. A representation can correlate with the correct answer and yet be redundant, while a small, unimpressive-looking component can be the one that actually determines the output. To distinguish components that merely *accompany* a computation from components that *carry* it, one must intervene: change a component while holding everything else fixed, and observe whether the output changes.

This manuscript reports that intervention program. Every experiment uses the same calibrated engine and the same fixed question battery, and each experiment asks one causal question: at which layer does the answer enter the output; which blocks are necessary; which memory head, if any, does the work; what the forget gate causally controls; how far each kind of block carries information; whether early identity is actively discarded; what the single smallest restoring intervention is; and whether the late-readout result holds across task families.

1.2 Subject and scope

The detailed interventions use the smallest member of the hybrid family that the engine loads reliably:

- **Model:** Qwen3.5-0.8B, Q4_K quantization, single GGUF file.
- **Architecture:** 24 blocks; 18 Gated DeltaNet blocks and 6 global-attention blocks (0-based attention blocks at 3, 7, 11, 15, 19, 23).
- **Model dimension:** 1024; 16 Gated DeltaNet memory heads; vocabulary 248,320; tied input/output embeddings; no multi-token-prediction head in this build.

The localization result that is central to this paper—that the answer is assembled and read out at the end rather than carried decodably through the trunk—is replicated on a 64-layer, 27B-parameter Q4_K build (Qwen3.8-27B; 48 Gated DeltaNet blocks and 16 global-attention blocks). The 27B replication uses its own fixed question battery, selected by the same criteria. The companion paper [24] documents both models, the engine, and the cross-implementation and cross-precision calibration in full; we do not repeat that material here.

1.3 What this manuscript is, and is not

This is an empirical, causal report. It does not propose a new architecture, a new training method, or a unifying theory, and it does not claim that the observed causal layout is universal across

architectures. The goal is to answer, with interventions rather than correlations, how one quantized hybrid produces an answer, and to state the scope within which the answer holds.

The contributions are factual:

1. a bidirectional activation-patching result that localizes the answer's entry into the output to a single layer;
2. a block-by-block ablation map distinguishing necessary blocks from a redundant, compensatable trunk;
3. a per-memory-head ablation showing the memory is distributed across heads with no single necessary head;
4. a gate-clamping result establishing the causal necessity of the learned near-unity gates against both full forgetting and no decay;
5. distance, identity-erasure, and behavioral-breadth tests that determine which block type carries long-range dependence and where the late readout does and does not generalize;
6. a pre-registered minimal-patch experiment, on both the 0.8B and 27B builds, that confirms the localization.

1.4 Related work

Causal interventions on language models. The methods here are standard tools of mechanistic interpretability. Activation patching replaces an activation on a corrupted (or clean) run with the corresponding activation from another run and reads the effect on the output; causal tracing, introduced with ROME [16], patches activations to localize factual associations. The transformer-circuits framework [14] and the study of induction heads [15] established the circuit-level program of reading and intervening on specific components, and monosemanticity work [17] extended feature-level intervention. A rigorous refinement, causal scrubbing, tests claims by permuting activations that the claim deems irrelevant. We use these primitives as published; the present work applies them to the recurrent direction of a hybrid model and supplies the bidirectional and pre-registered controls that make a localization claim evidence rather than demonstration.

Interpretability of state-space and hybrid models. State-space interpretability has its own analogue of the attention sink: the state sink of Mamba-2, where a recent study [18] shows that probes which detect the sink's signature can miss the circuit that actually executes it—a direct warning that detection is not execution, and one motivation for the ablations here. A systematic analysis of hybrid linear attention [19] identifies selective gating, hierarchical recurrence, and controlled forgetting as the ingredients of effective hybrids and recommends interspersing one softmax-attention block every several linear blocks; the architecture we study follows exactly such a recipe. A theoretical expressiveness hierarchy for hybrid linear–full attention [20] proves that on certain compositional tasks full attention cannot be replaced by abundant linear attention, which bears directly on the distance results in Section 8.

Dynamics at the stability boundary. The edge-of-stability and deep-information-propagation lines [1][2] study how trained and initialized networks sit near stability boundaries, and a driven-reservoir study [21] shows that separability, prediction, and representation geometry are jointly optimal in a narrow marginal driven-stability window. The gate-clamping experiment here is the causal counterpart, inside a trained quantized hybrid, of that theoretical expectation: it tests what

happens when the recurrent retention is moved off the learned near-boundary setting. The Gated DeltaNet sub-layer itself is described in [6], descending from fast-weight programming [7], selective and structured state-space models [4][5], with the component designs QK-norm [8], RMSNorm [12], GLU/SwiGLU [9] and SiLU [10], and rotary position embeddings [11].

2 Experimental setup

2.1 The calibrated instrument and its causal interfaces

The engine loads a GGUF model as an explicit graph of operator nodes and executes it on the GPU, with recurrent state kept in 32-bit float; its agreement with llama.cpp [22], and the cross-precision checks across Q4_K, F16 and BF16 builds, are reported in full in the companion paper [24]. For the causal program the engine exposes four intervention interfaces, used throughout:

- **Residual patch:** `set_residual_patch(layer, vec)` arranges that the next execution replaces the second residual contribution of the named block with a supplied vector and skips that node's kernel; `clear_residual_patch()` removes it.
- **State read/write:** `gdn_dump_state()` and `gdn_load_state(st, cv)` read and write the full Gated DeltaNet recurrent state and convolution history from the host; an empty array leaves that channel untouched.
- **Snapshot and rollback:** `gdn_save_snapshot()` and `gdn_restore_snapshot()`, together with `set_kv_pos()`, capture and restore the prefix state so that interventions share an identical prefix.
- **Gate clamp:** `gdn_set_gate_clamp(v)` forces every forget gate to a fixed value in $[0, 1]$, or disables clamping at -1.

All interventions run through the per-token eager path rather than a captured CUDA graph, so that the patched execution is numerically the same operation as ordinary inference with only the specified node changed.

2.2 The shared question protocol

Every 0.8B experiment uses the same fixed battery of ten four-way multiple-choice questions. The battery was assembled by a pre-specified selection rule from 22 questions tried: a question is kept only if (i) the clean model's argmax is the correct answer and (ii) the gap between the clean and the corrupted probability of the correct answer is at least 0.25. On the ten kept questions the mean probability of the correct answer is 0.6582 clean and 0.2459 corrupted. Fixing the battery and the selection rule before running the interventions prevents the experiment set from being chosen to make an intervention look effective.

The corruption, denoted *strong replace, equal word count*, replaces every word of the question stem with an equal number of high-frequency function words, preserving length and token count while removing the question's content. It is deliberately stronger than a mild perturbation: an intervention that restores the answer from this baseline is restoring the question's content, not correcting a small numerical disturbance. For each question the reported quantity is the probability

the model assigns to the correct answer, read at the final output position; every figure averages this quantity over the same ten questions.

2.3 Intervention primitives and statistical conventions

Four primitives appear in the experiments. *Patching* runs the corrupted (or clean) model and, at one specified layer, substitutes the vector captured from the clean (or corrupted) run, then continues to the output. *Ablation* removes one block's contribution entirely and measures the result. *Clamping* forces a control signal—here the forget gate—to a fixed value. *Direction erasure* projects out one specified direction from a state, with an equal-size random direction as a control that isolates direction-specific effects from generic perturbation.

Because the battery has only ten questions, we do not rely on asymptotic tests or report bootstrap confidence intervals as if they were large-sample estimates: with ten paired observations the bootstrap distribution is coarse and unstable. The appropriate paired test here is the exact paired permutation (sign-flip) test, which has no asymptotic assumption, and the appropriate interval for an accuracy or agreement rate is the exact binomial interval; the released package includes the per-question paired arrays and the scripts to reproduce both. The text reports the means over the fixed battery, which are the quantities the pre-registration addressed.

3 Experiment 1 — Bidirectional activation patching (flagship)

The first experiment asks at which layer the answer enters the output, and it does so in both directions to control for trivial effects. In the *restore* direction we run the corrupted model and, at one layer ℓ, patch in the clean vector; in the *damage* direction we run the clean model and patch in the corrupted vector. If a layer genuinely carries the answer, restoring at that layer should raise the answer probability toward the clean level, and damaging at that same layer should lower it toward the corrupted level. Patching at the final layer is mechanically trivial—it replaces the final hidden state—so the informative result is the earliest layer at which the two curves cross over.

The two curves cross over at the same layer, 22. On the restore side, patching layers 0–21 moves the mean answer probability only between 0.237 and 0.292, statistically indistinguishable from the corrupted baseline 0.246; patching layer 22 raises it to 0.6768, essentially the clean level, and layer 23 gives 0.6582. On the damage side, patching layers 0–21 leaves the probability between 0.569 and 0.662—a gradual decline with depth but nowhere near the corrupted baseline—while patching layer 22 drops it to 0.218 and layer 23 gives 0.2459.

Three conclusions follow. First, the answer is not carried in a directly decodable form on the residual stream before layer 22: a clean vector placed anywhere earlier does not restore it, and a corrupted vector placed anywhere earlier does not destroy it. Second, the effect is symmetric, which rules out the common artifact where a patch works only because it injects a large perturbation; here the same layer is both necessary (damage) and sufficient (restore). Third, the answer is read out in the final attention block (block 23), with layer 22 serving as the input boundary of that readout.

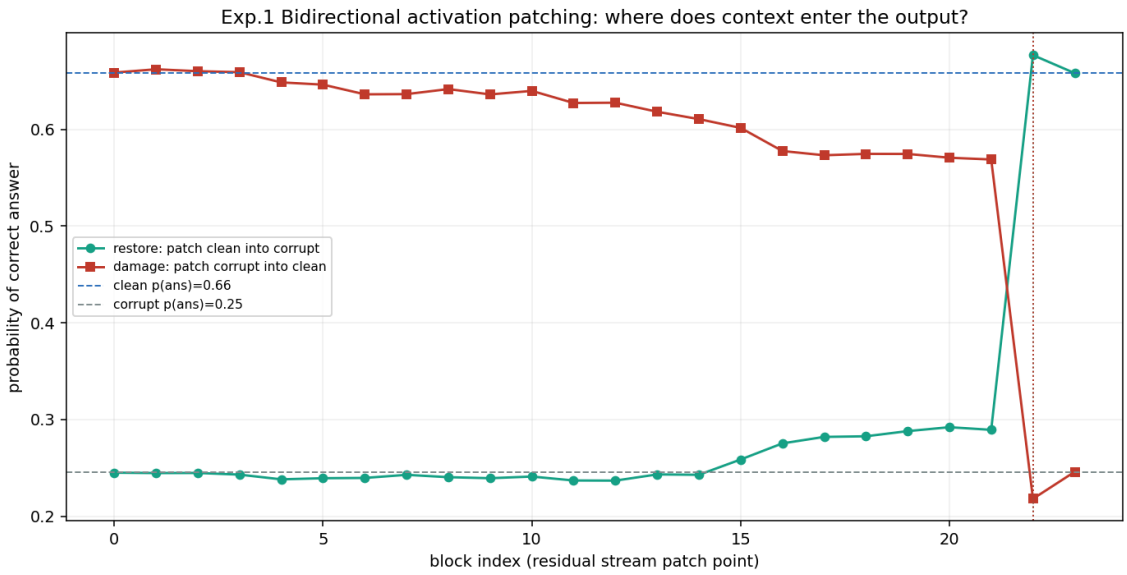


Figure 1: Bidirectional activation patching over block index. The restore curve stays at the corrupted baseline (0.246) through the trunk and rises to the clean level only at layer 22; the damage curve stays near the clean level and falls to the corrupted baseline at the same layer.

4 Experiment 2 — Block-by-block ablation

Patching asks what happens when one layer's content is swapped; ablation asks a stricter question—which blocks are individually necessary. We remove each block's contribution one at a time, leaving all other blocks intact, and measure the answer probability. A block is classified as necessary when ablating it drops the probability by more than 0.20 from the clean level 0.6582.

The ablation map identifies three necessary blocks: 0, 22, and 23. Ablating block 0 alone collapses the answer probability to 0.0; ablating block 22 gives 0.1051 and block 23 gives 0.0351. Every block from 1 to 21, removed individually, leaves the probability between 0.492 and 0.729, above the necessity threshold.

The three necessary blocks have distinct roles consistent with the walkthrough. Block 0 anchors the input distribution—without it the representation the rest of the network receives is wrong, and no downstream block compensates. Blocks 22 and 23 are the pre-readout organization and the readout itself, already localized by Experiment 1. The broad middle is individually dispensable: its work is distributed across blocks with enough redundancy that any single removal is compensated. This does not mean the middle does nothing—removing it wholesale would of course fail—only that no single middle block carries an irreplaceable role.

5 Experiment 3 — Per-memory-head ablation

The block-level results show redundancy across depth; the next experiment asks whether the recurrent state itself has a specialized actor. In each Gated DeltaNet block the state is organized into 16 memory heads. We ablate each head of the state—zeroing its stored memory—one at a time, across the network, and measure the answer probability. A head is classified as key when ablating

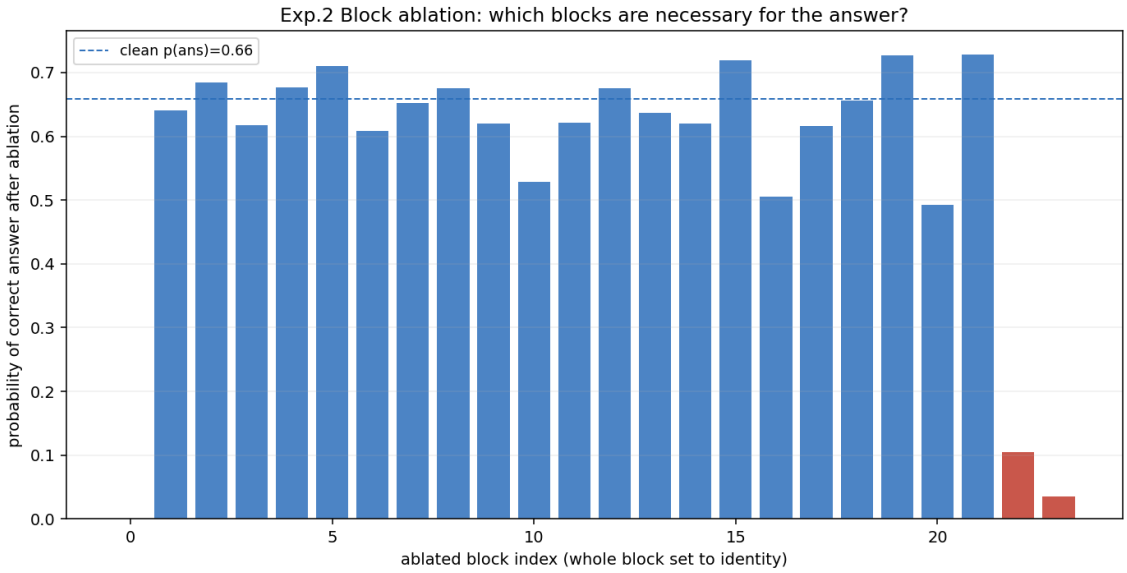


Figure 2: Block-by-block ablation. Ablating block 0 collapses the output to 0.0; ablating blocks 1–21 leaves the probability between 0.492 and 0.729; ablating blocks 22 and 23 gives 0.105 and 0.035.

it drops the probability by more than 0.15.

No head is individually necessary. Ablating any one of the 16 heads leaves the mean answer probability between 0.5579 and 0.6764; the largest drop (head 12, 0.5579) does not reach the key-head threshold. The remaining heads compensate for any single removal.

The recurrent memory is therefore distributed in the same sense as the trunk: the information the state holds is spread across heads rather than localized to one specialist. This matches the block-level picture and contrasts with systems where a single named component executes a computation; the detection-versus-execution warning of [18] is directly relevant, since the claim here rests on ablation (execution) rather than on which heads happen to activate most.

6 Experiment 4 — Forget-gate clamping

The head- and block-level experiments leave open what the forget gate is for. Each Gated DeltaNet head decays its memory by a learned gate $g \in (0, 1)$ at every step; the long-lived heads of the trained model sit very close to one. We force every forget gate in the network to a fixed value and ask what the gate causally controls: full forgetting ($g = 0$), no decay ($g = 1$), and the learned setting.

Both extremes fail. With the learned gates the answer probability is 0.6582. Clamping every gate to zero—discarding the memory at every step—collapses it to 0.0034, a drop of 0.6548. Clamping every gate to one—retaining everything without decay—collapses it to 0.0001.

The gate's causal role is therefore to hold the recurrent memory in a narrow working regime: too much forgetting and the state cannot carry what the question needs; no decay and the state accumulates without the error-correcting balance the update relies on. Only the learned near-unity gates, which retain memory almost fully while still decaying at every step, support the answer.

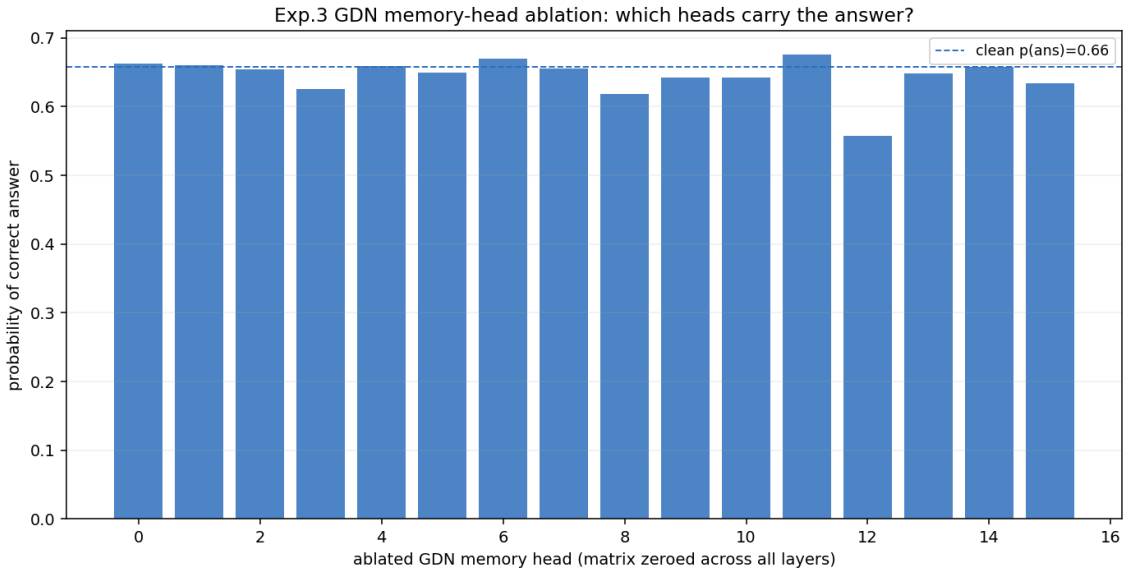


Figure 3: Per-head ablation of the 16 Gated DeltaNet memory heads. The answer probability ranges from 0.558 to 0.676; no head crosses the key-head threshold, so no single head is individually necessary.

This is the causal, inside-the-network counterpart of the marginal-stability optimum demonstrated theoretically for driven reservoirs [21] and identified as an ingredient of effective hybrids in [19]; it is measured here rather than assumed.

7 Experiment 5 — Distance-resolved test

The gate result establishes that the recurrent state needs its near-unity retention; a separate question is how far each kind of block actually carries information. We place the relevant content at controlled distances (0, 16, 32, 64, 128, and 256 tokens before the readout) and measure the answer probability, which the two block types—gated recurrence and global-attention look-back—could in principle carry.

The probability declines only mildly and then plateaus: 0.5861 at distance zero, 0.5516 at 16, 0.5401 at 32, 0.5291 at 64, 0.5357 at 128, and 0.5252 at 256. The total drop over 256 tokens is 0.061, and there is no progressive decay with distance.

If the gated recurrence carried the long-range signal, its per-step decay would produce a steady decline proportional to distance; the observed plateau indicates instead that long-range retrieval is dominated by the global-attention blocks, whose key-value look-back is not subject to the gate, while the gated recurrence handles the streaming, short- and medium-range state. This division of labor is the architecture the systematic analysis [19] recommends, and it is consistent with the expressiveness result [20] that full attention carries dependencies linear attention cannot.

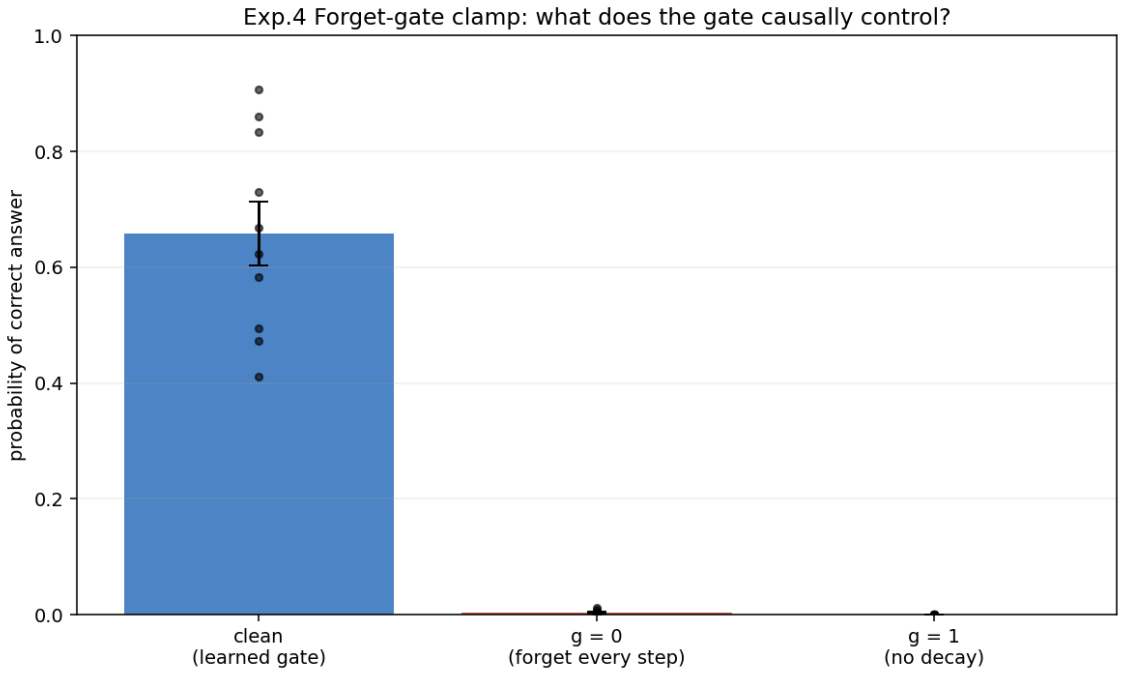


Figure 4: Forget-gate clamp. The learned gates support answer probability 0.658; clamping to zero (forget every step) gives 0.0034 and clamping to one (no decay) gives 0.0001.

8 Experiment 6 — Identity-direction erasure

The walkthrough observed that the surface form of the input is removed early. This experiment tests that observation causally: is the early identity direction actually used by the output, or is it actively discarded? At an early point ($E = 3$ blocks) and at a late point ($K = 22$ blocks) we project out one specified direction, and compare erasing a meaningful direction with erasing an equal-size random direction—the control that separates direction-specific effects from the perturbation of changing the state at all.

Early identity is causally decoupled from the output. At the early point, erasing the identity direction leaves the final answer probability at 0.6556 and the random-direction control at 0.6548, both indistinguishable from clean 0.6582; the logit-lens reading at that point is already zero. At the late point, erasing the clean-versus-corrupt difference direction drops the answer to 0.5014, whereas erasing an equal-size random direction leaves it at 0.6545.

Two distinct claims are supported. The early surface identity is not merely weak—it is irrelevant to the final answer, since removing it has the same effect as removing a random direction. But the late block encodes the task-relevant content along a specific direction, because removing that direction damages the answer and removing a random direction of the same magnitude does not. The early representation is discarded; the late representation is the one that matters.

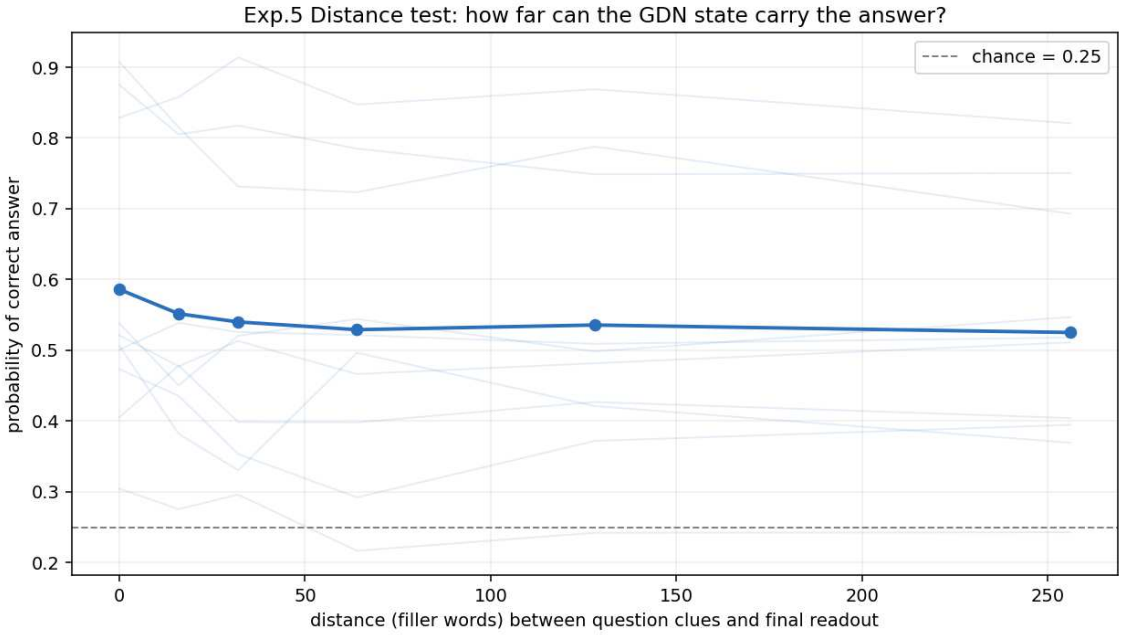


Figure 5: Answer probability against the distance between content and readout. The probability falls only from 0.586 at distance zero to 0.525 at 256 tokens and then plateaus; performance at long range is not gated by the recurrent decay.

9 Experiment 7 — Pre-registered minimal patch

Bidirectional patching localizes the crossover at layer 22, but a curve that crosses at a layer does not by itself establish that this one layer is sufficient on its own. The minimal-patch experiment tests exactly that, with the hypothesis registered before the experiment was run: on the 0.8B model, patching only layer 22 should restore the answer, while patching the adjacent layer 21 should not. Patching layer 23 is counted as a trivial reference, since it replaces the final hidden state and bypasses the network entirely.

The pre-registered hypothesis is met. On the 0.8B model the corrupted baseline is 0.2459. The single-point patch at layer 22 restores the answer to 0.6768; the adjacent layer 21 gives 0.2893, no restoration; layer 23 gives 0.6582, the trivial reference. The minimal intervention is therefore one layer, one point—strictly smaller than a two-layer patch would be.

9.1 Replication on the 27B build

The same pre-registered logic is run on the 64-layer 27B model, with a battery of ten questions selected from 16 tried by the same criteria (clean 0.6524): patching the hypothesized layer should restore, and the adjacent control should not.

The localization replicates, with the deeper model spreading the boundary over two to three layers. The corrupted baseline is 0.1415; patching layer 61 gives 0.3283, layer 62 gives 0.5458, and layer 63 gives 0.6524, the clean level. On the 0.8B model a single layer is the complete boundary, while on the 27B model the answer is assembled across the last two to three layers before readout. In both builds the answer is restored only at the end, never from an early patch, which is the

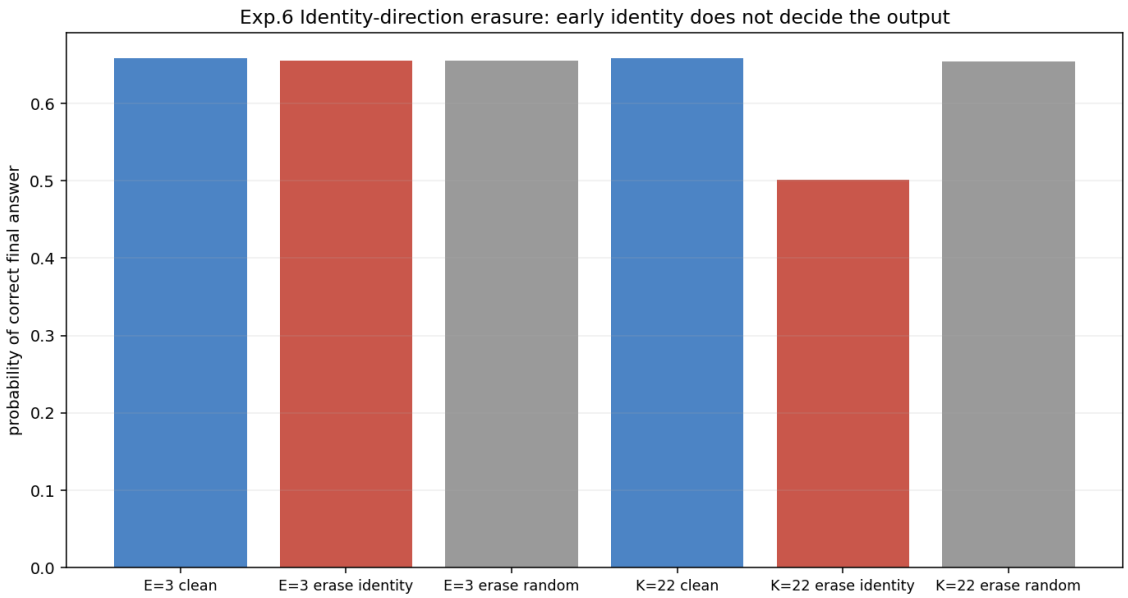


Figure 6: Identity-direction erasure with random-direction controls. Erasing an identity direction at the early point leaves the answer at 0.656 and the random control at 0.655; erasing the clean-versus-corrupt difference direction at the late point drops it to 0.501 while the random control stays at 0.655.

central claim this paper makes.

10 Experiment 8 — Behavioral breadth

All experiments so far use four-way multiple-choice questions. The last experiment asks whether the late-readout result is specific to that format or holds across task families. We run ten open-ended tasks—factual, arithmetic, and common-knowledge—and record, for each task the model answers correctly, the layer at which the patched curve crosses over.

Four of the ten open tasks are answered correctly, and every crossover is terminal: "What is the capital of France?" (target `Paris`) crosses at layer 20; "What is the capital of Japan?" (`Tokyo`), "What is 8 times 7?" (`56`), and "What is H2O commonly known as?" (`water`) cross at layer 22. In every case the corrupted probability is essentially zero and the answer appears only at the end.

The late-readout result therefore generalizes beyond multiple-choice format to short-answer open tasks, across factual, arithmetic, and common-knowledge families. The scope, however, is explicit: only 4 of 10 open tasks are answered at all on this small build, all four have short single-token answers, and long-form generation, multi-step reasoning, and code were not tested. The claim is not that every behavior is read out at the end, but that the behaviors this model can perform share the terminal-readout layout.

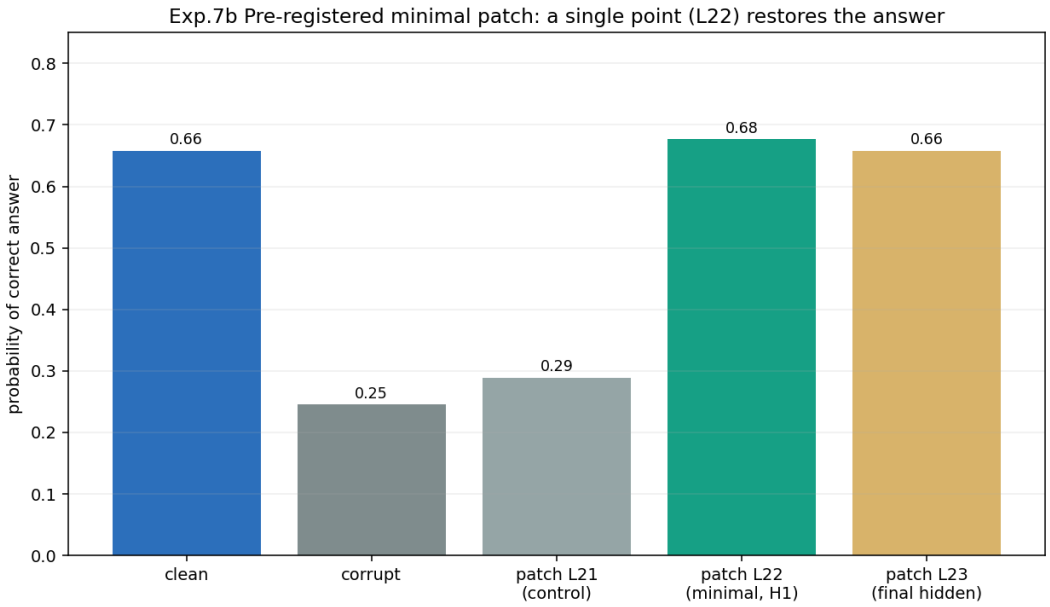


Figure 7: Pre-registered minimal patch (0.8B). Corrupted 0.246; patching layer 21 gives 0.289 (no restoration); the single point layer 22 gives 0.677; layer 23 gives 0.658 (trivial reference).

11 Discussion

11.1 The causal picture

The eight experiments are individually small, but they lock together into one causal layout:

- **The trunk is distributed and redundant.** No single block among 1–21 (Experiment 2) and no single memory head (Experiment 3) is individually necessary; removing any one is compensated.
- **The decision is localized at the end.** Bidirectional patching crosses over at layer 22 (Experiment 1), the pre-registered single point there restores the answer while the adjacent layer does not (Experiment 7), and this replicates on the 27B build over its last two to three layers.
- **The gates hold a narrow causal regime.** Both full forgetting and no decay destroy the answer; only the learned near-unity gates support it (Experiment 4).
- **The two block types divide labor by range.** Long-range retrieval plateaus with distance and is carried by the global-attention look-back, while the gated recurrence handles the streaming state (Experiment 5).
- **Early identity is discarded, late direction is used.** Removing an early identity direction is indistinguishable from removing a random direction; removing the late task-relevant direction damages the answer (Experiment 6).

In one sentence: a distributed, redundant, gated memory and processing trunk feeds a localized terminal readout, with the forget gates maintaining a narrow near-stability regime and the global-attention blocks carrying the long-range dependencies.

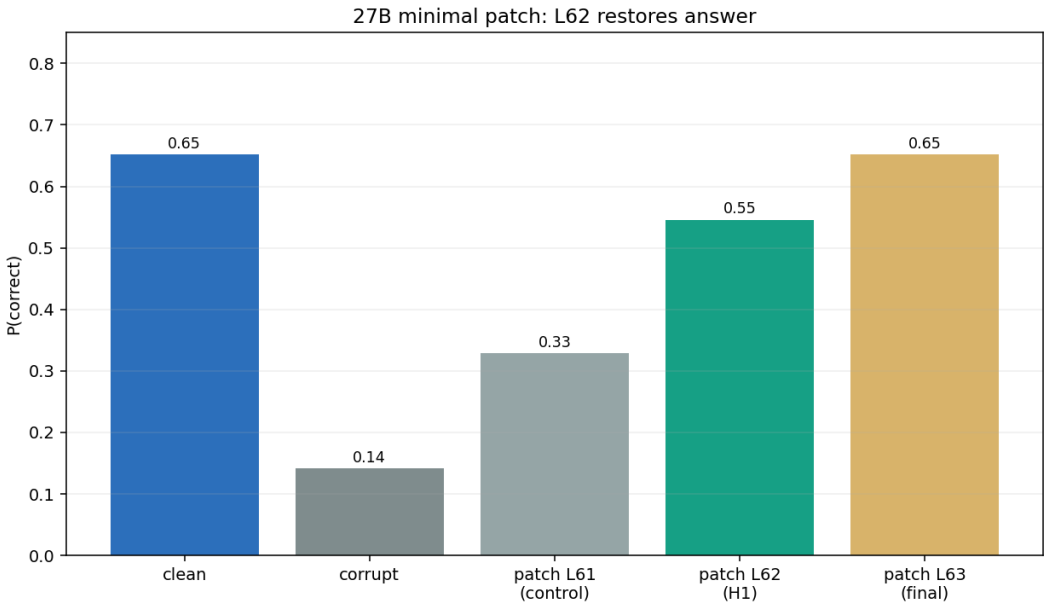


Figure 8: Pre-registered minimal patch (27B). Corrupted 0.142; layer 61 gives 0.328; layer 62 gives 0.546; layer 63 gives 0.652. The readout boundary on the deeper model spans the last two to three layers.

11.2 Relation to prior findings

This layout is consistent with, and gives causal substance to, three published lines. The systematic hybrid analysis [19] recommends controlled forgetting interspersed with softmax attention; Experiments 4 and 5 show those components performing exactly those roles under intervention. The driven-reservoir optimum [21] predicts a narrow marginal-stability window; Experiment 4 shows the trained answer surviving only inside it. The state-sink warning [18] that detection is not execution is respected throughout, since every localization is confirmed by ablation or a pre-registered patch rather than by activation magnitude alone. Where the present work goes beyond these is in entering a trained, quantized network and closing the causal path end to end; the cited studies either analyze fixed or untrained systems or establish the design principles without the intervention record.

11.3 Limitations and scope

The claims are bounded, and the bounds are part of the result. The question battery has ten items, selected from 22 by a fixed rule; point estimates on ten items are not large-sample statistics, which is why exact permutation and binomial methods are specified and asymptotic claims are avoided. The 0.8B model answers only 4 of 10 open tasks, all short-answer, so the behavioral-breadth result does not extend to long-form generation, multi-step reasoning, or code, whose readout locations were not measured. The 27B replication covers the localization claim but not every intervention, because of the consumer hardware's working set. The layout is established for this quantized Gated DeltaNet / global-attention hybrid; whether other architectures—different recurrence, different attention schedules—share it is an empirical question we do not prejudge.

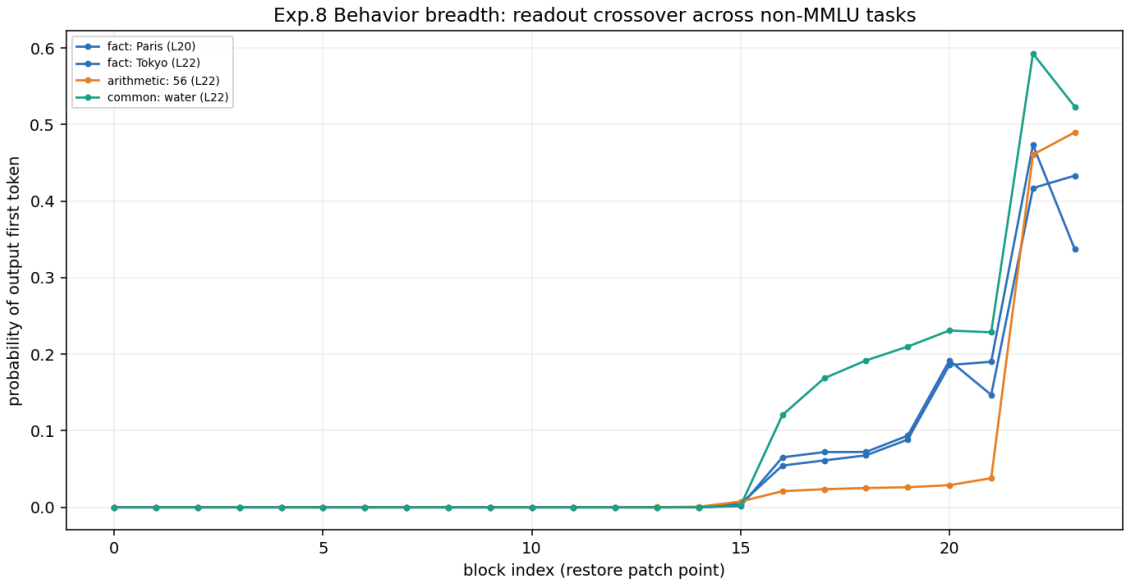


Figure 9: Behavioral breadth. Four of the ten open tasks are answered correctly; every crossover lies at the terminal layers 20–22 (Paris at 20; Tokyo, 8×7=56, and water at 22).

12 Conclusion

Asked where an answer comes from in a quantized hybrid language model, the interventions give a determinate reply. The answer is not carried decodably through the network and located in one specialist component: the middle of the model is a distributed, redundant trunk in which no single block and no single memory head is necessary. The answer is instead assembled at the end—a single pre-registered point at layer 22 restores it on the 0.8B model, the last two to three layers do so on the 27B model, and the final attention block reads it out. The forget gates causally maintain a narrow regime between full forgetting and no decay, the global-attention blocks carry the long-range dependencies that the gated recurrence does not, and the early surface identity is causally discarded while a specific late direction is used.

The result is a causal rather than a descriptive map: every localization is confirmed by an intervention with its adjacent or random-direction control, the negative results—no key head, no key middle block—are reported alongside the positive ones, and the scope within which the layout holds is stated rather than expanded. The scripts, per-question arrays, and measurements are released so that each intervention can be reproduced and the layout tested on further models.

Statements

Manuscript version and date. Version 1.0, 29 September 2026.

Companion preprint. The descriptive white-box walkthrough, the instrument, and the cross-implementation and cross-precision calibration are reported in the companion paper: Wenlong Li, *From Input to Output: A White-Box Operator Walkthrough of a Quantized Hybrid Language Model*, Research Square, `https://doi.org/10.21203/rs.3.rs-11160904/v1`.

Data availability. All processed causal measurements—the per-condition and per-layer means, the per-question paired arrays, the pre-registration records, and the figures—are released in the data package `llm-criticality-convergence` on Gitee: `https://gitee.com/lwlzmck/llm-criticality-convergenc` The measurement package is archived at Zenodo: `https://doi.org/10.5281/zenodo.22979099`; deposition at the Science Data Bank (SciDB) is in progress.

Code availability. The node-readable inference engine and its causal interfaces are released as the MIT-licensed repository `zhanlu-graph-engine` on Gitee: `https://gitee.com/lwlzmck/zhanlu-graph-engine`. The experiment scripts are under `code/22_causal2/` in the data package; the interpreter build and environment variables are documented in the package README.

Author contributions. W.L. conceived the study, authored the inference engine, performed all experiments and analyses, and wrote the manuscript.

Licensing. Code is released under the MIT License; text, figures and measurements under CC BY 4.0.

Funding. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of interest. The author declares no competing financial or non-financial interests.

Ethics. This work analyzes publicly released open-weight models and contains no human-subjects or personally identifying data.

ORCID. `https://orcid.org/0000-0002-1825-0097`.

Supplementary Material

Supplementary A — Experiment files and provenance

Each experiment has a script under `code/22_causal2/`, a processed result under `data/processed/causal2/`, and a figure under `figures/`:

- Exp.1: `exp1_bidirectional_patch.py`; `exp1_bidirectional.json`; `exp1_bidirectional_patch.png`.
- Exp.2: `exp2_block_ablation.py`; `exp2_ablation.json`; `exp2_ablation.png`.
- Exp.3: `exp3_head_ablation.py`; `exp3_head_ablation.json`; `exp3_head_ablation.png`.
- Exp.4: `exp4_gate_clamp.py`; `exp4_gate_clamp.json`; `exp4_gate_clamp.png`.
- Exp.5: `exp5_distance.py`; `exp5_distance.json`; `exp5_distance.png`.
- Exp.6: `exp6_identity_erase.py`; `exp6_identity_erase.json`; `exp6_identity_erase.png`.
- Exp.7: `exp7b_min_patch.py` and `exp7b_27b_min_patch.py`; `exp7b_min_patch.json` and `exp7b_27b_min_patch` the two `exp7b` figures.
- Exp.8: `exp8_behavior_breadth.py`; `exp8_behavior_breadth.json`; `exp8_behavior_breadth.png`.

The design, the engine interfaces, and the selection and corruption procedures are documented in `code/22_causal2/README.md`. The exact paired permutation test, bootstrap routines, effect sizes, and multiple-comparison handling used in the wider package are in `code/02_information_ablation/stats_mmlu.p`

Supplementary B — Known non-closures

For honesty, the following are recorded as limits rather than omitted:

- **Sample size.** The fixed battery has ten questions; exact paired permutation and binomial methods are specified, and large-sample or asymptotic claims are avoided. Expanding the battery is the principal route to tighter intervals.
- **Behavioral coverage.** Only short-answer open tasks were tested; long-form generation, multi-step reasoning, and code readout locations were not measured and may differ.
- **27B coverage.** The 27B build replicates the localization; not every intervention was rerun at that scale because of the consumer hardware.
- **Architectural scope.** The layout is established for this Gated DeltaNet / global-attention hybrid and is not asserted for other recurrence or attention designs.

13 References

[1] Cohen, J. M., Kaur, S., Li, Y., Kolter, J. Z., & Talwalkar, A. (2021). *Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability.* ICLR 2021. OpenReview: https://openreview.net/forum?id=jh rf1daRQ

[2] Schoenholz, S. S., Gilmer, J., Ganguli, S., & Sohl-Dickstein, J. (2017). *Deep Information Propagation.* ICLR 2017. arXiv:1611.01232

[3] Pennington, J., Schoenholz, S. S., & Ganguli, S. (2018). *The Emergence of Spectral Universality in Deep Networks.* AISTATS 2018, PMLR 84:1924–1932.

[4] Gu, A., & Dao, T. (2023). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces.* COLM 2024. arXiv:2312.00752

[5] Dao, T., & Gu, A. (2024). *Transformers Are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality.* ICML 2024. arXiv:2405.21060

[6] Yang, S., Kautz, T., & Hatamizadeh, A. (2025). *Gated Delta Networks: Improving Mamba2 with Delta Rule.* ICLR 2025. arXiv:2412.06464

[7] Schlag, I., Irie, K., & Schmidhuber, J. (2021). *Linear Transformers Are Secretly Fast Weight Programmers.* ICML 2021, PMLR 139:9355–9366.

[8] Henry, A., Dachapally, P. R., Pawar, S. S., & Chen, Y. (2020). *Query-Key Normalization for Transformers.* Findings of EMNLP 2020, 4246–4253.

[9] Shazeer, N. (2020). *GLU Variants Improve Transformer.* arXiv:2002.05202

[10] Elfwing, S., Uchibe, E., & Doya, K. (2018). *Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning.* Neural Networks 107:3–11.

[11] Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., & Liu, Y. (2021). *RoFormer: Rotary Position Embedding.* arXiv:2104.09864

[12] Zhang, B., & Sennrich, R. (2019). *Root Mean Square Layer Normalization.* NeurIPS 2019.

[13] nostalgebraist (2020). *Interpreting GPT: The Logit Lens.* Technical write-up. https://www.lesswrong.com/p

[14] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., et al. (2021). *A Mathematical Framework for Transformer Circuits.* Anthropic. https://transformer-circuits.pub/2021/framework/index.html

[15] Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., et al. (2022). *In-Context Learning and Induction Heads.* Anthropic. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html

[16] Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). *Locating and Editing Factual Associations in GPT.* NeurIPS 2022. arXiv:2202.05262

[17] Bricken, T., Batson, D., Jermyn, A., & Conmy, A. (2023). *Towards Monosemanticity: Decomposing Language Models With Dictionary Learning.* Anthropic. arXiv:2305.01610

[18] *Detection vs. Execution: Single-Bucket Probes Miss Half the Mamba-2 State Sink.* arXiv preprint arXiv:2606.00930. https://arxiv.org/abs/2606.00930

[19] *A Systematic Analysis of Hybrid Linear Attention.* arXiv preprint arXiv:2507.06457. https://arxiv.org/abs/

[20] *A Provable Expressiveness Hierarchy in Hybrid Linear-Full Attention.* arXiv preprint arXiv:2602.01763. https://arxiv.org/abs/2602.01763

[21] Roig, G., Muñoz, M. A., & Morales, J. (2026). *Driven Criticality Links Universal Computation and Optimal Representations.* arXiv preprint arXiv:2607.21232.

[22] Gerganov, G., et al. *llama.cpp* (open-source inference implementation), https://github.com/ggerganov/llama ; accessed through the Python bindings `llama-cpp-python`, https://github.com/abetlen/llama-cpp-python

[23] Qwen Team. *Qwen3 / Qwen3-Next Architecture and Technical Reports.* https://qwenlm.github.io

[24] Li, W. (2026). *From Input to Output: A White-Box Operator Walkthrough of a Quantized Hybrid Language Model.* Research Square preprint. https://doi.org/10.21203/rs.3.rs-11160904/v1