THE TOKEN BEFORE THE VALUE IS THE KEY:

HOW HYBRID ARCHITECTURES ORGANIZE INDUCTION CIRCUITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Hybrid language models combine efficient and global layers, but how these layers develop complementary roles in induction remains unclear. We analyze induction in terms of three functions: carrying predecessor information, matching a historical source by content, and copying its value. Using layer-type-agnostic paired probes at a common block-update interface, we track Carrying and Matching in recurrent–global and local–global hybrids. In both settings, Carrying concentrates in efficient layers, while Matching develops in global receivers. The measured local contribution is concentrated at lag one, on the token immediately preceding a historical value. Interventions that alter predecessor support–lag-one masking, convolution removal, and reduced early learning rates–shift the distribution of Carrying and Matching across stages. Source-key restoration and fixed-value selection further trace the receiver's dependence on the prepared source. Even with the architecture and training corpus held fixed, early learning-rate interventions produce different natural-text prediction and recall outcomes. Varying local windows and using induction-enriched training text also changes when functional Carrying and Matching emerge. Together, these results suggest that learning to carry predecessor information helps determine where and when Matching develops, linking architectural design and training evidence to induction circuits and recall. Code is available in https://anonymous.4open.science/r/bind-match-copy-F4C8/README.md.

1 INTRODUCTION

Hybrid language models were initially motivated by efficiency: interleaving recurrent or local modules with selected global-attention layers can reduce sequence-modeling cost (De et al., 2024; Lieber et al., 2024; Ren et al., 2025). This design has since appeared in open model families including Qwen3.5 (Qwen Team, 2026b), Qwen3.8-Next (Qwen Team, 2026a), and Kimi K3 (Kimi Team, 2026). Recent studies also report gains in capability and scaling (Wang et al., 2025; Merrill et al., 2026). These findings raise a broader question: what does mixing operators enable beyond computational savings?

Existing works address this question from two directions. Theoretical analyses identify advantages in expressivity and memory efficiency on formal tasks (Cooper et al., 2026; Merrill et al., 2026), while empirical studies examine recall, scaling, and learning across hybrid designs (Wang et al., 2025; Qiao et al., 2026). Less is known about the learned computations connecting these architectural properties to model behavior. We study that connection at the circuit level.

Classical induction provides a concrete computation to trace. In $AB \dots A \to B$, a model retrieves an earlier continuation (Elhage et al., 2021; Olsson et al., 2022). Following the subcircuit decomposition of Singh et al. (2024), we call these functions Carrying, Matching, and Copying.

CARRYING transfers predecessor information into the historical value; MATCHING selects a source by content, and COPYING transmits its value (Figure 1). This raises a question: *how does a hybrid allocate the position-sensitive Carrying and content-based Matching computations? Do they have different affinities for efficient and global layers?* Attention-score diagnostics commonly used for induction heads do not directly apply to recurrent components, motivating a shared interface across layer types.

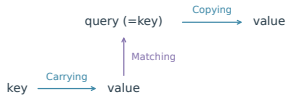

Figure 1: **Induction roles:** our functional labels for subcircuits A–C of Singh et al. (2024).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We observe this affinity: Carrying concentrates in efficient groups, followed by Matching in global receivers. Their measured local role concentrates on lag one—the token immediately before the historical value. This familiar Transformer relation sharpens the interpretation of locality in hybrids: the efficient layer prepares a key for distant retrieval.

Changing predecessor support reallocates Carrying and Matching; local structure and training evidence also shape their formation. Source-key restoration and fixed-value selection trace the connection. These results offer an upstream perspective on hybrid specialization: learning to carry predecessor information shapes where and when Matching develops. The token before the value is the key.

Contributions. Our contributions are threefold. First, we introduce layer-type-agnostic paired probes for Carrying and Matching that support all subsequent analyses. Second, the hybrid division of labor exposes distinct intervention points: changing early learning conditions under a fixed architecture and training corpus relocates retrieval and changes downstream prediction. Third, varying architectural and data support for early predecessor learning changes functional circuit formation, revealing a dynamic consequence of hybrid specialization. Together, the measurements connect local architectural choices to source preparation, retrieval development, and downstream behavior.

2 Layer-Type-Agnostic Induction Probes

Our layer-type-agnostic paired probes measure Carrying at the historical value and Matching at the query through a common block-update interface, requiring neither attention weights nor recurrent-state coordinates. Together, they locate the preparation of the token before the value and the retrieval response that uses it.

2.1 Models and Training

We compare eight-layer Gated DeltaNet (GDN) hybrids, sliding-window attention (SWA) hybrids, and full-attention Transformers. The hybrids use $[\text{Efficient}^3, \text{TF}]^2$, with full attention at L3 and L7; Transformer uses full attention throughout. GDN combines gated delta-rule memory with width-four causal convolution (Yang et al., 2025). SWA uses causal local attention, with a default window of four tokens including the current position. All models have width 512 and eight heads, with approximately 79M, 77M, and 77M parameters. They train on packed OpenWebText with context 1,024 for 30K steps. All controlled models use learned absolute position embeddings; SWA additionally uses RoPE, while full-attention layers do not. Layer indices start at zero. Variants specify the changed local operation or early learning rate (LR). Appendix A.1 details training and replication.

Each hybrid offers two global receivers, each preceded by an efficient group. Changing local access or early learning while retaining those receivers tests how support affects allocation. A 16-layer, approximately 318M model and released Qwen checkpoints extend the comparison.

2.2 Measuring Carrying and Matching

We use prompts of the form $c = AB \dots CD \dots A_q$, for which B is the target continuation and D is a distractor. Let s denote the historical B position and q the final A position. We measure the correct-minus-distractor logit margin $m = z_B - z_D$.

All probes use the same activation-patching operation (Geiger et al., 2023; Zhang & Nanda, 2024). Given a clean prompt c and a donor prompt d, we replace the block update at layer ℓ and position p in the clean run with the corresponding donor update, then recompute subsequent blocks. Here a block update is its output minus its input, including the feed-forward computation where present. If the clean and donor updates are u_c and u_d, respectively, we use the RMS-matched update $\widetilde{u}_d = u_d \,\mathrm{RMS}(u_c)/(\mathrm{RMS}(u_d) + 10^{-8})$. We define $\textsc{PatchLoss}(c, d; \ell, p)$ as the clean margin minus the patched margin. Thus, a positive value indicates that replacing this update with the donor's reduces the preference for B over D. Donors differ from their paired clean prompts only in the specified key tokens; all filler tokens remain identical.

Each condition uses 200 prompts per model, run, and checkpoint (Appendix A.1). Paired samples are reused across layers and compared interventions, with effects averaged at each coordinate.

Algorithm 1. Lag probe

Input: layers $\mathcal{L}$, offsets $\mathcal{D}$.
Output: $\delta_{\text{lag}}(\ell, r)$.

1. For each $r \in \mathcal{D}$, sample 200 clean prompts with a key r tokens before value s.
2. **for** $\ell \in \mathcal{L}, r \in \mathcal{D}$ **do**
3. **for each** clean prompt c: copy c to d; replace the historical key.
4. $e_c \leftarrow \text{PATCHLOSS}$ $(c, d; \ell, s)$.
5. $\delta_{\text{lag}}(\ell, r) \leftarrow \frac{1}{200} \sum_c e_c$.

Algorithm 2. Carrying probe

Input: layers $\mathcal{L}$.
Output: $\delta_{\text{Carrying}}(\ell)$.

1. Sample 200 clean prompts $c = AB \dots CD \dots A_q$; s is the historical B position.
2. **for** $\ell \in \mathcal{L}$ **do**
3. **for each** c: copy c to d; replace the predecessor of B.
4. $e_c \leftarrow \text{PATCHLOSS}$ $(c, d; \ell, s)$.
5. $\delta_{\text{Carrying}}(\ell) \leftarrow \frac{1}{200} \sum_c e_c$.

Algorithm 3. Matching probe

Input: layers $\mathcal{L}$.
Output: $\delta_{\text{Matching}}(\ell)$.

1. Sample 200 clean prompts $c = AB \dots CD \dots A_q$; q is the query position.
2. **for** $\ell \in \mathcal{L}$ **do**
3. **for each** c: copy c to d; swap keys: $d = CB \dots AD \dots A_q$.
4. $e_c \leftarrow \text{PATCHLOSS}$ $(c, d; \ell, q)$.
5. $\delta_{\text{Matching}}(\ell) \leftarrow \frac{1}{200} \sum_c e_c$.

The lag scan tests offsets 1/2/4/8/16/32/64; Carrying fixes the adjacent relation. Query-update Matching measures the net retrieval response, combining selection, value transmission, and subsequent FFN effects. Fixed-value attention-pattern swaps isolate selection (Section 3.3; Algorithm S1). Prompt construction and query-response controls appear in Appendices B.1 and B.3.

Both functional readouts depend on the current downstream computation. Formation curves track each run's endpoint-selected layer through training; Appendix D.1 supplies fixed-site comparisons. Replicated results report training-run means and sample SD.

3 How Hybrids Allocate Induction Computations

We first map Carrying and Matching across layer types and identify the predecessor relation carried by local computation. Interventions then test how its learning conditions shape circuit allocation and formation, with source-key tests tracing the connection to Matching. Deeper models and natural-text evaluation examine how this organization extends across configurations and relates to behavior.

3.1 Mapping Carrying and Matching Across Layer Types

Do the positional and content-based parts of induction favor different layer types? We scan Carrying and Matching across layers, then examine which historical offset contributes to the source representation.

Both hybrids place strong Carrying at the end of the first efficient group and Matching in the following global layer; Transformer expresses these roles deeper and with greater variation across runs (Figure 2, bottom). The lag response concentrates at offset one in hybrids and Transformer (top). This relation is familiar from classical induction (Elhage et al., 2021; Olsson et al., 2022; Singh et al., 2024). In hybrids, it identifies a specific function of local computation: carrying the token before the value into the source representation used for distant retrieval.

Identity controls isolate the predecessor relation (Appendix B.2). These controls compare replacing the tested predecessor with replacing another causally accessible historical token, separating the association-specific response from sensitivity to other context changes. The scans identify the division of labor and its lag-one component. We next test whether changing support for that component changes which receiver learns Matching.

3.2 Changing Carrying Conditions Shifts the Circuit

The lag-one concentration suggests an upstream explanation for this allocation. We test it by changing predecessor support while keeping global receivers at L3 and L7. LR $\times 0.1$ acts on L0–2 and L4–6 for the first 3K steps in GDN and 4K in SWA, including mixer, FFN, and norm. It is a broad perturbation of both efficient groups; channel interventions provide the predecessor-specific tests. Mask and convolution experiments test circuit location; window experiments test formation time (Section 3.4).

We first suppress only the lag-one convolution channel during early training. This removes the direct predecessor contribution from that channel while retaining other convolution offsets and the recurrent computation. Some runs develop a deeper Carrying–Matching path; others retain the

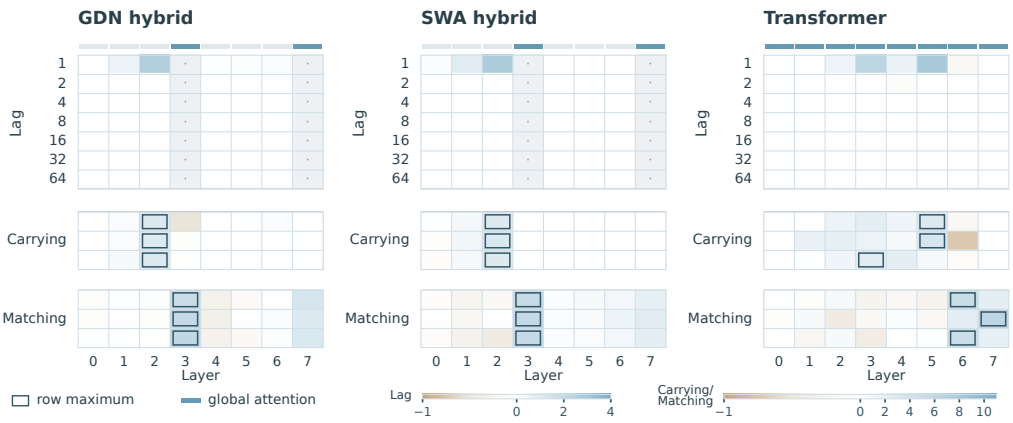

Figure 2: **The same predecessor relation, different learned locations.** Top: lag sensitivity in one run. Bottom: Carrying and query-update Matching in three runs. Outlines mark row maxima; dots mark untested coordinates. Rails identify global attention. Separately labeled lag and Carrying/Matching scales give signed margin effects and are shared across architectures.

shallow route (Figure 3). The offset and timing comparisons sharpen the result: suppressing lag two early, or suppressing lag one after the initial formation period, preserves the shallow circuit (Appendix C.1).

Width-two convolution retains self and predecessor while removing longer offsets; the shallow organization remains. Removing convolution consistently moves Carrying and Matching to the later group and receiver. Together with lag-one masking, these controls distinguish sufficient predecessor support, partial disruption with variable allocation, and broader removal that favors a later circuit.

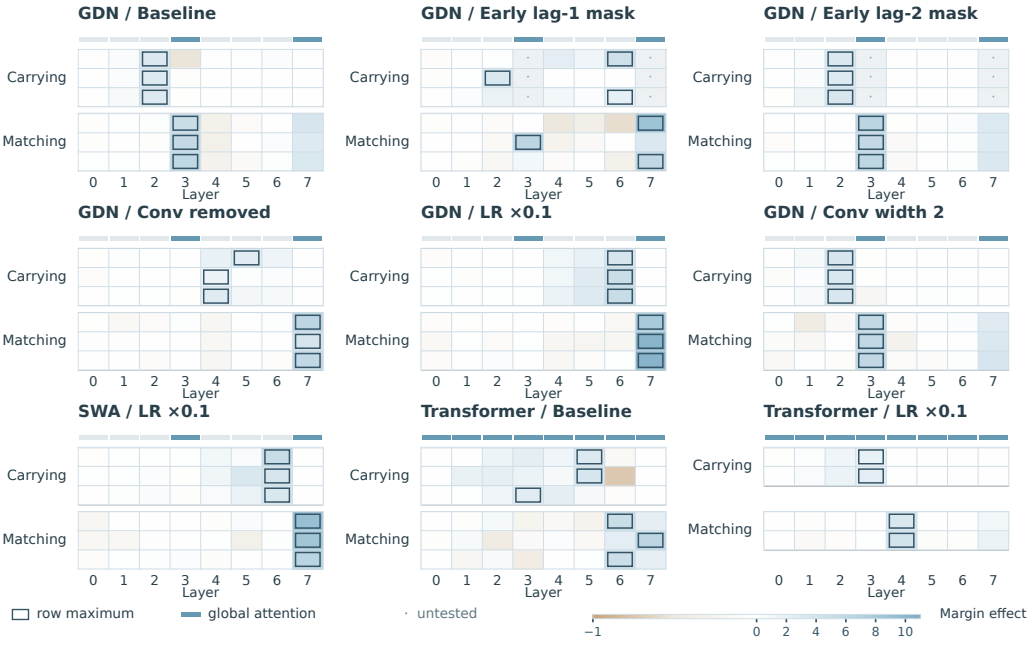

Figure 3: **Changing upstream support redirects the computation.** Signed Carrying and Matching layer profiles under offset, convolution, and early-learning interventions. Each condition groups three independent runs, except Transformer LR $\times 0.1$ with two. Outlines mark maxima; dots mark untested coordinates. GDN and SWA denote the two hybrid families; LR denotes learning rate. Timing and layout comparisons are expanded in Appendix C.

Early LR reduction retains the local operator and its inputs, yet both hybrids develop the deeper Carrying–Matching pair. Local access and early optimization thus provide different controls over the learned allocation.

Receiver layout and Transformer controls test this dependence from the other side. Shifting global sites favors the later available receiver (Appendix C.1), while Transformer LR reduction changes where its circuit develops (Appendix C.2). A persistent self-and-predecessor head instead brings Transformer Carrying into a shallow layer; releasing it after early training yields more variable locations (Appendix C.2). The hybrid division exposes distinct intervention points in the efficient groups and global receivers. Their acquired predecessor support and available source access jointly shape circuit location.

3.3 Global Matching Reads the Prepared Source Key

Relocation identifies a new pair of sites. We next trace whether the receiver uses source information prepared at the new Carrying site. Source-key restoration tests this dependence at the historical value in both original and relocated routes.

A Carrying-layer patch at the historical value changes the K and V delivered to attention. Transferring K alone into a clean run reproduces nearly all full-path loss in both hybrid families' baseline and reduced-LR runs; V has little effect (Appendix B.5). The same key-dominated propagation appears after convolution removal and early lag-one suppression. Restoring clean K at the correct source recovers the prediction; the same repair at another position does not (Table 1A).

The receiver must also select the appropriate source. We isolate this operation by exchanging its attention pattern while holding V fixed. After early LR reduction, the strong selection effect moves from the earlier to the later global receiver in both hybrids (Table 1B). Across the tested conditions, the receiver favored by this intervention agrees with query-update Matching (Appendix B.4). Fixed-V swaps test receiver selection in baseline/LR runs, while clean-K restoration traces source dependence after convolution removal and lag-one masking.

Table 1: **Source-key dependence and receiver selection.** A: GDN source-key restoration, aggregated at each run's peak route. Early lag-1 mask includes relocated runs 42/44 (L6→L7) and unrelocated run 43 (L2→L3). Negative off-site recovery means additional margin loss. B: fixed-V pattern effects. Both panels report three-run mean ± sample SD in margin units; the reproduction protocol uses 200 prompts per condition.

A. Source-key restoration

Condition	Full-path	K recovery	Off-site
Conv removed	2.51 ± 0.67	2.52 ± 0.67	-1.34 ± 0.74
Early lag-1 mask	3.46 ± 1.03	3.42 ± 1.00	-1.77 ± 1.61

B. Fixed-value selection

Model	Condition	L3	L7
GDN	Baseline	5.85 ± 0.45	1.32 ± 0.39
	LR $\times 0.1$	0.02 ± 0.07	9.16 ± 0.95
SWA	Baseline	5.34 ± 0.21	0.31 ± 0.06
	LR $\times 0.1$	0.02 ± 0.03	8.14 ± 1.33

The source-key path connects altered predecessor support to global Matching: changes to Carrying reach the receiver through source K, and the relocated receiver selects that source. Head-level Matching/Copying and source-edge tests complete this path (Appendices B.7 and B.6). At the principal routes, closing the receiver's source edge reduces both Carrying and Copying effects. Graded interventions trace this dependence across intermediate suppression strengths.

3.4 Predecessor Support Shapes Circuit Formation

The allocation question has a temporal counterpart: what lets a receiver acquire Matching early? We trace functional Carrying and Matching under different local windows and training streams, testing how early support for the predecessor relation shapes circuit development.

SWA window 2 advances early Carrying relative to windows 8/16 (Figure 4A/C). Matching appears earlier than in window 16, but ties window 8 at the lowest thresholds; Appendix D.3 resolves these intervals. The wider-window models then catch up and reach stronger final Carrying responses. These trajectories separate earlier functional use of the predecessor relation from the stronger response that broader windows develop later. Following Carrying connects window-dependent retrieval-head formation (Qiao et al., 2026) to the development of the source computation on which retrieval depends.

All three SWA windows admit the predecessor; their trajectories distinguish access from when the relation becomes functionally useful. Window 2 and width-two convolution retain self and predecessor while excluding longer offsets. The window and convolution comparisons connect local support to the formation and allocation of the historical-source circuit.

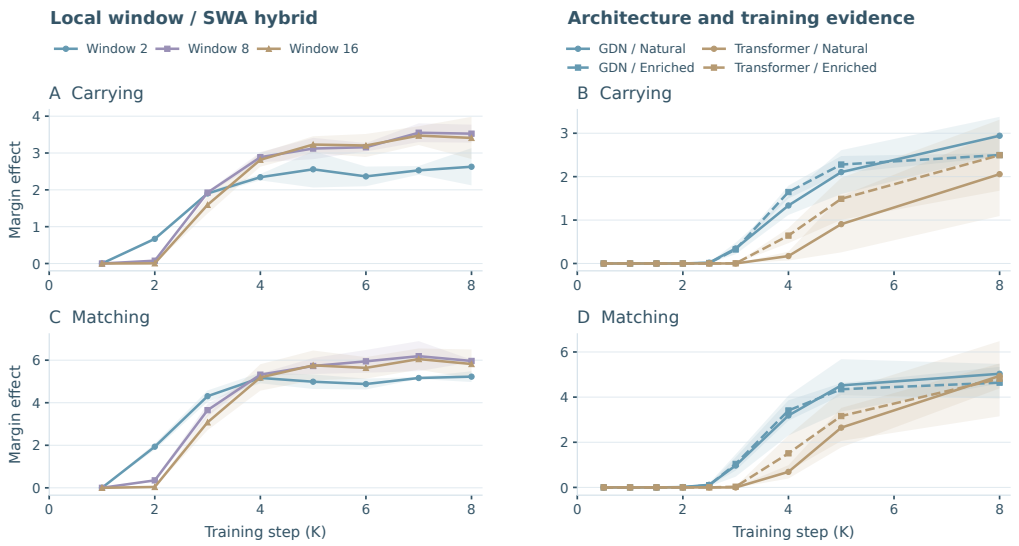

Figure 4: **Two ways to change early circuit development.** A/C: SWA windows 2/8/16 Carrying at L2 and query-update Matching at L3, using 200 prompts per cell. B/D: Hybrid/Transformer under natural and induction-enriched text, following fixed endpoint-selected layers. Lines and bands show means and sample SD over three runs. The plotted checkpoints cover the first 8K steps.

The data comparison varies how often this relation is available during training. Induction enrichment emphasizes repeated, reliable continuations through the frequency/reliability filter in Appendix A.2, motivated by Aoyama et al. (2026). Across three runs, Transformer Carrying and Matching rise earlier with enriched text than with natural text; the two hybrid conditions rise in a similar early interval (Figure 4B/D). This pattern suggests that local architectural support and enriched training evidence can facilitate a shared preparation step: enrichment has a larger timing effect where that step develops later under natural text.

The early support for learning this one-token relation thus connects circuit allocation to functional formation timing. Fixed-site comparisons (Appendix D.1), random-retention and full-depth trajectories (Appendix D.2), and onset thresholds (Appendix D.3) trace this timing contrast. Following each eventual circuit and a common physical site provides complementary views of development; the full-layer scans show where the response emerges as training proceeds. Section 3.6 relates formation timing to final recall.

3.5 Circuit Organization Across Model Designs

The preceding tests connect predecessor support to circuit location and timing in eight-layer hybrids. We now examine this organization with more global stages, released checkpoints, and alternative receivers. In the four-stage 318M model, LR reduction in selected groups favors early, intermediate, or late routes, with greater variation for nonadjacent groups (Figure 5A; Appendix C.3). Allocation follows the network-wide support pattern.

Depth also permits dependence across multiple stages. In a 318M Baseline run, source-K restoration recovers part of the sender-patch loss at both an intermediate global receiver and the later peak-Matching receiver (Appendix B.5). Peak responses identify prominent sites on paths that can contain additional global computation, so source preparation can influence multiple receivers.

Released Qwen checkpoints extend the comparison to larger pretrained models (Qwen Team, 2025; 2026b). Qwen3 expresses the operations in separated full-attention layers; Hybrid Qwen3.5 places its strongest Carrying in a GDN layer immediately before the strongest global Matching (Figure 5B). In Qwen3.5, component restoration identifies K as the main route by which the sender perturbation affects retrieval. This recovery persists within prompts on which the clean model favors the correct target. A common-token dose sweep also produces graded K recovery while the perturbed model still favors the correct target (Appendix E.2). The same dependence appears across input difficulty and perturbation strength, including conditions with strong recall.

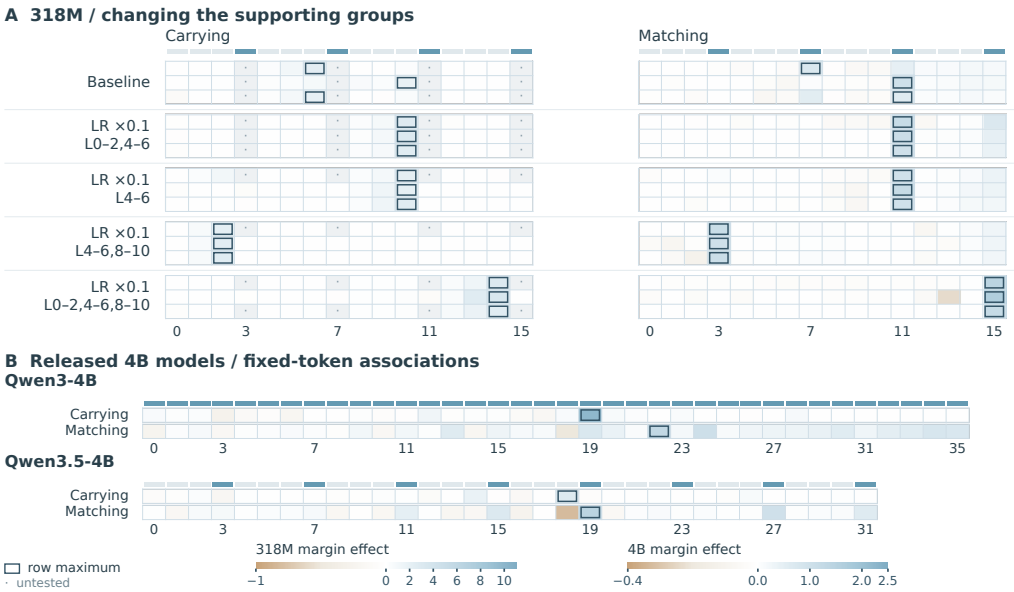

Figure 5: **Source preparation and retrieval beyond the two-stage setting.** A: five 318M conditions, with three runs aligned between Carrying and Matching. Row labels give the LR multiplier and affected layers during the first 3K steps. B: all-layer fixed-token association scans in Qwen3/3.5-4B. Outlines mark maxima; rails mark global attention; dots mark untested coordinates. Separate labeled scales serve 318M and 4B.

Receiver controls test the access needed by this division. GDN–DSA trained from scratch retains Carrying in GDN and Matching in sparse global receivers (Appendix F.5). Pure GDN and GDN–SWA show weak historical-source Carrying alongside query-time responses (Appendix F.4). In GDN-only, transferring the query state changes the recalled association, while historical-source transfer has little effect. This comparison identifies information accessible through the recurrent state at prediction time. These measurements distinguish the traced source pathway from recurrent retrieval.

3.6 Circuit Organization and Natural-Text Prediction

After the synthetic tests, we ask whether the same previous-token relation matters for natural-text prediction. We retain full context and score targets whose predecessor–continuation pair has appeared before, separating single- and multiple-continuation contexts by predecessor recency. Domain-level PPL complements these targeted readouts. PPL evaluation uses 10,000 sequences: 2,000 Wiki, 4,000 Python code, and 4,000 Math. Long-range identifier reuse gives a code example; Appendix F.1 defines the target groups. Domain panels score all valid prediction positions, while recall panels focus on positions with specific historical evidence. Reading them together places association-sensitive prediction within each model's broader natural-text behavior.

In GDN, ablating source-attending heads produces larger recall NLL increases than ablating comparison heads, in both the baseline receiver and the later receiver after early LR reduction (Appendix F.3). The circuit analysis therefore connects to prediction beyond the constructed prompts.

Final recall provides a complementary view of circuit development. Narrow local windows advance early Carrying and Matching, yet wider windows predict distant, multiple-continuation targets and reused identifiers better at the final checkpoint (Table 2).

In the eight-layer hybrids, early LR reduction lowers several multiple-continuation recall PPLs and raises single-continuation far PPL; GDN convolution removal has a similar profile (Table 2). Domain-level results and the 318M comparisons extend this variation across prediction settings (Appendix F.2). At 318M, some interventions lower code-domain PPL, while mean distant multiple-continuation and identifier-reuse PPL favor Baseline. Early LR interventions hold the architecture and training corpus fixed: changing early Carrying conditions changes Matching allocation and the eventual prediction profile. Head ablations establish that both original and relocated receivers con-

Table 2: **Recall under changes in local support.** PPL at 30K; lower is better. Single/multiple denote historical continuation counts; identifier reuse is long-range. Values are mean ± sample SD. The Transformer Baseline/LR pair uses two matched runs; all other rows use three. PPL uses 10,000 sequences per run. Rules separate comparisons; bold marks each column's minimum within a block.

		Single	Multiple continuation			Identifier
Model	Variant	Far	Near	Mid	Far	reuse
SWA	Window 2	2.26 ± 0.05	2.85 ± 0.36	14.97 ± 0.60	28.99 ± 0.43	28.05 ± 2.21
	Window 4	2.33 ± 0.06	2.86 ± 0.22	15.52 ± 1.09	26.61 ± 1.18	28.31 ± 3.97
	Window 4, LR $\times 0.1$	2.64 ± 0.06	$\mathbf{2.68 \pm 0.25}$	$\mathbf{12.94 \pm 1.70}$	25.63 ± 0.87	23.49 ± 2.12
	Window 8	$\mathbf{2.25 \pm 0.07}$	2.78 ± 0.15	14.00 ± 0.94	24.19 ± 0.96	23.54 ± 2.86
	Window 16	2.26 ± 0.07	3.13 ± 0.34	14.31 ± 2.02	$\mathbf{22.40 \pm 2.55}$	$\mathbf{20.91 \pm 6.76}$
GDN	Baseline	$\mathbf{2.38 \pm 0.04}$	2.75 ± 0.39	16.34 ± 1.99	25.98 ± 1.23	29.00 ± 1.43
	LR $\times 0.1$	2.59 ± 0.05	$\mathbf{2.64 \pm 0.27}$	12.52 ± 0.93	23.35 ± 0.34	21.02 ± 1.60
	Conv removed	2.70 ± 0.05	2.83 ± 0.73	$\mathbf{12.50 \pm 0.28}$	$\mathbf{21.75 \pm 1.99}$	$\mathbf{18.72 \pm 4.57}$
Transformer	Baseline	$\mathbf{2.66 \pm 0.04}$	$\mathbf{2.59 \pm 0.14}$	$\mathbf{15.03 \pm 1.93}$	$\mathbf{25.73 \pm 0.29}$	$\mathbf{25.01 \pm 0.62}$
	LR $\times 0.1$	3.33 ± 0.04	3.24 ± 0.14	19.34 ± 2.67	33.60 ± 1.41	34.86 ± 1.77

tribute to natural-text prediction. Section 5 considers how their use within the rest of the model may shape these outcomes.

4 Related Work

Induction as a shared computation. Previous-token key composition is central to classical induction (Elhage et al., 2021; Olsson et al., 2022). Singh et al. (2024) distinguish Layer 1 (A), IH QK Match (B), and Copy (C), whose functions we call Carrying, Matching, and Copying, respectively. Our paired probes measure Carrying and query-update Matching across hybrid mixers; path and head-level tests examine their connection to Copying. Subsequent studies examine induction learning and data dependence (Musat et al., 2025; 2026; Aoyama et al., 2026). Inductive biases toward n-gram processing also improve in-context language learning (Akyürek et al., 2024). Interventions on predecessor support then connect the learning conditions for Carrying to the allocation and formation of global Matching.

Complementarity in hybrid architectures. Local attention, state spaces, and gated linear attention offer different access patterns (Beltagy et al., 2020; Gu & Dao, 2024; Yang et al., 2024; 2025). Hybrid designs combine them with attention (De et al., 2024; Lieber et al., 2024; Ren et al., 2025), and empirical and theoretical studies examine their complementarity (Wang et al., 2025; Cooper et al., 2026; Merrill et al., 2026; Cabannes et al., 2026; Shi et al., 2026). Qiao et al. (2026) identify efficient attention as an optimization prior and show that larger local windows can delay retrieval-head formation. We trace the upstream Carrying computation and its source-key interface to connect local design and learning conditions to where and when retrieval develops.

Retrieval across model families. Recall and copying studies expose differences between recurrent models and attention (Arora et al., 2024; Park et al., 2024; Jelassi et al., 2024; Bick et al., 2025). Arora et al. (2025) distinguish historical-position induction from recurrent query-time association and identify the role of short convolution in Mamba induction. Our common update interface follows the historical-source computation across mixers; path interventions then test how its source and receiver cooperate (Geiger et al., 2023; Zhang & Nanda, 2024).

5 Discussion

The token before the value organizes Matching. Matching locations show where retrieval occurs; observing Carrying helps explain how that allocation is learned. The layer-type-agnostic paired probes make this connection visible across heterogeneous operators: efficient groups prepare predecessor-sensitive representations at historical values, and global receivers select those sources.

Locality takes a specific form in this circuit—carrying the token immediately before the value into its source representation. Lag-one masking, convolution removal, and early LR changes alter this preparation and the receiver that develops Matching. Source-key restoration and fixed-value selection trace their connection. Carrying supplies a common organizing condition for these interventions. This perspective connects local design choices to source preparation for global retrieval.

Predecessor support shapes functional formation. Short convolution and narrow local attention supply architectural support for predecessor learning; induction-enriched data supplies repeated, reliable training evidence. Narrower SWA windows advance early functional Carrying, with threshold-dependent advances in Matching. Enrichment advances Transformer formation, while hybrid natural/enriched trajectories rise in a similar early interval. These patterns suggest that architecture and data can facilitate a shared preparation step. Tracking this step connects retrieval-head development (Qiao et al., 2026) to learning the source information that heads read.

Local order and distant source access. Carrying transfers predecessor order into the source representation before content-based selection. This offers a circuit-level interpretation of effective NoPE global retrieval alongside position-sensitive efficient layers (Kimi Team, 2025; 2026; Qiao et al., 2026). The receiver must also reach the prepared source. Sparse selection can retain distant candidates, as in Qwen3.8-Next (Qwen Team, 2026a); local SWA excludes sources outside its window. Our GDN–DSA model trained from scratch retains Carrying in GDN and Matching in sparse receivers, while GDN–SWA has weak historical-source Carrying (Appendices F.5 and F.4).

Local access can also favor recurrent memory: Cabannes et al. (2026) explain short-window benefits through increased use of that pathway. Our pure GDN and GDN–SWA controls retain query-time responses alongside weak historical-source Carrying. The common probe interface distinguishes these routes (Cooper et al., 2026; Merrill et al., 2026; Yang et al., 2025; Bick et al., 2025). For historical-source retrieval, local structure shapes predecessor preparation, while the receiver determines access to distant candidates. This connects convolution, window size, and global receiver design through their roles in one computation.

Early Carrying and downstream computation. With the architecture and training corpus held fixed, changing early learning in efficient layers reallocates Matching and leads to different prediction profiles. The effects vary across domains, recall conditions, and model configurations (Appendix F.2). An altered training trajectory may also change how later layers use retrieved information and resolve conflicts between a recalled continuation and other contextual signals. Such changes offer possible links between early Carrying and final behavior; the measured allocation identifies one part of this downstream computation. Qiao et al. (2026) find that long-context gaps between efficient-attention designs shrink with sufficient training, motivating comparisons over training as well as at the endpoint. For hybrid design, observing source preparation alongside retrieval connects architectural priors and learned specialization to the use of historical evidence.

6 Conclusion

Layer-type-agnostic paired probes reveal how hybrids allocate the established Carrying and Matching computations across efficient and global layers. The local contribution concentrates on the token before the historical value. Changing early learning conditions relocates Carrying and Matching and leads to different natural-text prediction profiles; varying architectural and data support also changes functional formation timing. These findings offer an upstream perspective on hybrid architectures: observing how the token before the value is carried helps explain where and when global retrieval develops. Carrying thus connects local design, learned specialization, and downstream use.

AI USE STATEMENT

Generative AI tools assisted with manuscript revision and polishing, including feedback on methodological descriptions and interpretation of results, and with experimental-code generation. The authors take responsibility for the final text, code, results, and claims.

ETHICS STATEMENT

This work studies computational mechanisms in language models using synthetic token prompts, existing text corpora, and released model checkpoints. It does not involve human-subject experiments. Data and model use are subject to the terms of their original providers.

REPRODUCIBILITY STATEMENT

Section 2 and Algorithms 1–3 define the shared experimental setting and interventions. Appendix A specifies architectures, training schedules, and text filtering; Appendix B details paired prompts and causal readouts. Appendices C–F provide intervention settings, training trajectories, pretrained-model experiments, and evaluation target selections. Figures and tables identify the training replication and uncertainty summaries used for each comparison.

REFERENCES

Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms. *arXiv preprint arXiv:2401.12973*, 2024. URL https://arxiv.org/abs/2401.12973.

Tatsuya Aoyama, Ethan Gotlieb Wilcox, and Nathan Schneider. Predicting the emergence of induction heads in language model pretraining. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.16893. arXiv:2511.16893.

Aryaman Arora, Neil Rathi, Nikil Roashan Selvam, Róbert Csordás, Dan Jurafsky, and Christopher Potts. Mechanistic evaluation of transformers and state space models. *arXiv preprint arXiv:2505.15105*, 2025. URL https://arxiv.org/abs/2505.15105.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2312.04927.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. URL https://arxiv.org/abs/2004.05150.

Aviv Bick, Eric P. Xing, and Albert Gu. Understanding the skill gap in recurrent language models: The role of the gather-and-aggregate mechanism. *arXiv preprint arXiv:2504.18574*, 2025. URL https://arxiv.org/abs/2504.18574.

Loïc Cabannes, Maximilian Beck, Gergely Szilvasy, Matthijs Douze, Maria Lomeli, Jade Copet, Pierre-Emmanuel Mazaré, Gabriel Synnaeve, and Hervé Jégou. Short window attention enables long-term memorization. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.24552. arXiv:2509.24552.

John Cooper, Ilias Diakonikolas, Mingchen Ma, and Frederic Sala. Expressivity–efficiency tradeoffs for hybrid sequence models. *arXiv preprint arXiv:2603.08859*, 2026. URL https://arxiv.org/abs/2603.08859.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando de Freitas, and Çağlar Gülçehre. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024. URL https://arxiv.org/abs/2402.19427.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv preprint arXiv:2301.04709*, 2023. URL https://arxiv.org/abs/2301.04709.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2024. URL https://arxiv.org/abs/2312.00752.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*, 2024. URL https://arxiv.org/abs/2402.01032.

Kimi Team. Kimi Linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025. URL https://arxiv.org/abs/2510.26692.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL https://arxiv.org/abs/2607.24653.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer–mamba language model. *arXiv preprint arXiv:2403.19887*, 2024. URL https://arxiv.org/abs/2403.19887.

William Merrill, Yanhong Li, Tyler Romero, Anej Svete, Caia Costello, Pradeep Dasigi, Dirk Groeneveld, David Heineman, Bailey Kuehl, Nathan Lambert, Chuan Li, Kyle Lo, Saumya Malik, DJ Matusz, Benjamin Minixhofer, Jacob Morrison, Luca Soldaini, Finbarr Timbers, Pete Walsh, Noah A. Smith, Hannaneh Hajishirzi, and Ashish Sabharwal. OLMo Hybrid: From theory to practice and back. *arXiv preprint arXiv:2604.03444*, 2026. URL https://arxiv.org/abs/2604.03444.

Tiberiu Musat, Tiago Pimentel, Lorenzo Noci, Alessandro Stolfo, Mrinmaya Sachan, and Thomas Hofmann. On the emergence of induction heads for in-context learning. *arXiv preprint arXiv:2511.01033*, 2025. URL https://arxiv.org/abs/2511.01033.

Tiberiu Musat, Tiago Pimentel, Nicolas Zucchet, and Thomas Hofmann. Invariant learning dynamics of transformers in inductive reasoning tasks. *arXiv preprint arXiv:2607.11875*, 2026. URL https://arxiv.org/abs/2607.11875.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Jongho Park, Jaeseung Park, Zheyang Xiong, Nayoung Lee, Jaewoong Cho, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. Can Mamba learn how to learn? A comparative study on in-context learning tasks. *arXiv preprint arXiv:2402.04248*, 2024. URL https://arxiv.org/abs/2402.04248.

Ziqing Qiao, Yinuo Xu, Chaojun Xiao, Zhou Su, Zihan Zhou, Yingfa Chen, Xiaoyue Xu, Xu Han, and Zhiyuan Liu. Rethinking the role of efficient attention in hybrid architectures. *arXiv preprint arXiv:2606.15378*, 2026. URL https://arxiv.org/abs/2606.15378.

Qwen Team. Qwen3-4B model card. Hugging Face, 2025. URL https://huggingface.co/Qwen/Qwen3-4B.

Qwen Team. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026a. URL https://arxiv.org/abs/2608.30320.

Qwen Team. Qwen3.5-4B model card. Hugging Face, 2026b. URL https://huggingface.co/Qwen/Qwen3.5-4B.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.07522.

Runlin Shi, Bojian Yin, and Guoqi Li. Modern transformers are implicit hybrids: From functional differentiation to principled hybrid architecture design. *arXiv preprint arXiv:2609.02986*, 2026. URL https://arxiv.org/abs/2609.02986.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 45637–45662. PMLR, 2024. URL https://proceedings.mlr.press/v235/singh24c.html.

Dustin Wang, Rui-Jie Zhu, Steven Abreu, Yong Shan, Taylor Kergan, Yuqi Pan, Yuhong Chou, Zheng Li, Jibin Wu, Ge Zhang, Wenhao Huang, and Jason Eshraghian. A systematic analysis of hybrid linear attention. *arXiv preprint arXiv:2507.06457*, 2025. URL https://arxiv.org/abs/2507.06457.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2312.06635.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.06464.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

APPENDIX CONTENTS

A Model Configuration and Training Data 14

A.1 Model and Training Configuration 14

A.2 Induction-Enriched Training Text 14

B Layer-Type-Agnostic Probes and Path Tests 15

B.1 Synthetic Prompts and Probe Definitions 15

B.2 Predecessor Specificity 15

B.3 Query-Association Controls 16

B.4 Content Selection with Values Held Fixed 16

B.5 Source-Key Propagation and Restoration 17

B.6 Source Access and Carrying–Copying Interactions 18

B.7 Head-Level Matching and Copying 19

C Carrying Conditions and Circuit Allocation 19

C.1 Hybrid Offset, Timing, and Layout Controls 19

C.2 Transformer Controls 21

C.3 Circuit Allocation in the 318M Hybrid 22

D Predecessor Support and Circuit Formation 23

D.1 Checkpoint Schedule and Layer Selection 23

D.2 Training-Data Controls and Layerwise Trajectories 23

D.3 SWA Window Size: Onset and Final Responses 24

E Circuit Tests in Pretrained Qwen Models 26

E.1 Model Configuration and Circuit Localization 26

E.2 Prompt Calibration and Source-Key Restoration 26

F Natural-Text Evaluation and Receiver Controls 28

F.1 Natural-Text Evaluation Protocol 28

F.2 Domain Prediction and Recall under Carrying Interventions 29

F.3 Retrieval-Head Ablations on Natural Text 31

F.4 Pure GDN and GDN–SWA 31

F.5 GDN–DSA Trained from Scratch 31

A Model Configuration and Training Data

A.1 Model and Training Configuration

Table S1 lists the controlled architecture families and local-window variants.

Table S1: Architecture families and local-window variants. Width is 512 and FFN dimension is 2,048. SWA variants share the same parameter count. Let P_{GDN} denote the baseline GDN count; reducing convolution width from four to two removes 24,576 weights across its six GDN layers.

Model	Layout	Heads	Parameters
GDN hybrid	$[\mathrm{GDN}^3, \mathrm{TF}]^2$	8	78.88M
SWA hybrid	$[\mathrm{SWA}_4^3, \mathrm{TF}]^2$	8	77.26M
Transformer	$\mathrm{FullAttn}^8$	8	77.26M
SWA window 2	$[\mathrm{SWA}_2^3, \mathrm{TF}]^2$	8	77.26M
SWA window 8	$[\mathrm{SWA}_8^3, \mathrm{TF}]^2$	8	77.26M
SWA window 16	$[\mathrm{SWA}_{16}^3, \mathrm{TF}]^2$	8	77.26M
GDN / Convolution width 2	GDN hybrid, kernel 2	8	$P_{\mathrm{GDN}} - 24{,}576$

Position encoding. The released controlled-model implementation adds learned absolute position embeddings before the first block. Full-attention RoPE is disabled; SWA applies RoPE to its Q and K in addition to those embeddings. The same embedding module is used in the 16-layer model. The probes trace source-key dependence with these positional signals present.

Training. The standard setup uses a GPT-2 tokenizer, packed 1,024-token OpenWebText sequences, effective batch size 64, bfloat16, AdamW with $(\beta_1, \beta_2) = (.9, .95)$, learning rate 3×10^{-4}, 1K warmup, cosine decay to 3×10^{-5}, and gradient clipping at 1.0. The standard training duration is 30K steps. Early LR reduction multiplies the scheduled learning rate of every parameter in the selected blocks by 0.1, including mixer, FFN, and normalization parameters. The principal Hybrid conditions target the efficient blocks until 3K for GDN and 4K for SWA; the Transformer controls target their specified full-attention blocks. The common learning-rate schedule then resumes while preserving the accumulated optimizer moments.

Replication and evaluation. Mechanism probes use 200 prompts per model, training run, checkpoint, and intervention condition. Principal GDN, SWA, Transformer, and window conditions use three independent training runs; layout-matched Transformer LR reduction uses two matched runs. Replicated values report means and sample SD across training runs. Prompt-bootstrap intervals resample paired prompts within a fixed checkpoint. Each run's strongest layer and reported effect are selected from the same final-checkpoint scan; Appendix D.1 describes how formation analyses track those coordinates. Natural-text PPL uses 10,000 sequences per checkpoint, with domain composition and target selection defined in Appendix F.1.

A.2 Induction-Enriched Training Text

The frequency and reliability account of Aoyama et al. (2026) motivates enrichment for repeated, predictable continuations. The enrichment filter combines repeated-bigram frequency and continuation consistency, evaluated at paragraph level from GPT-2 tokens.

Paragraph preparation. Documents are split on blank lines. Stripped paragraphs shorter than 100 characters are discarded. Each remaining paragraph is tokenized with the GPT-2 tokenizer, truncating the scoring input to 384 tokens; scoring inputs shorter than 30 tokens are skipped. A retained paragraph is written in full to the filtered JSONL training stream.

Enrichment criterion. For a token sequence $s = (x_1, \ldots, x_n)$, let $\mathcal{B}(s)$ be its distinct adjacent bigrams and c_b the number of occurrences of bigram b. The frequency score is the fraction of bigram types that repeat:

$$F(s) = \frac{\sum_{b \in \mathcal{B}(s)} \mathbf{1}[c_b \geq 2]}{|\mathcal{B}(s)|}. \tag{1}$$

To score reliability, count the tokens immediately following each bigram. Let $c_{b,v}$ count occurrences of b followed by token v, and define $\mathcal{B}_+(s) = \{b : \sum_v c_{b,v} \geq 2\}$. Only bigrams with a following token contribute continuation observations. The reliability score is

$$R_2(s) = \frac{1}{|\mathcal{B}_+(s)|} \sum_{b \in \mathcal{B}_+(s)} \mathbf{1}\left[\frac{\max_w c_{b,w}}{\sum_v c_{b,v}} > \frac{1}{2}\right]. \tag{2}$$

Empty-denominator scores are zero. Both scores weight bigram types equally. Reliability counts bigram types with a strict majority continuation. The frequency/repeat variant used for training retains a paragraph exactly when

$$S_{\mathrm{IH}}(s) = \tfrac{1}{2}F(s) + \tfrac{1}{2}R_2(s) > 0.10. \tag{3}$$

Training mixture. Enriched-text runs use 70% enriched text for the first 5K steps and 50% thereafter through 30K; natural-text runs use unfiltered text. The six conditions combine GDN hybrid, Transformer, and GDN-only with natural or induction-enriched text. A separately trained three-run random-retention series controls for data filtering alone. Figure S9 and Appendix D.1 show how the training stream affects circuit formation.

B Layer-Type-Agnostic Probes and Path Tests

B.1 Synthetic Prompts and Probe Definitions

The layer-type-agnostic probes in Algorithms 1–3 use one block-update interface: record output minus input, RMS-match the donor update, and recompute later blocks. Carrying measures historical-source preparation; query-update Matching measures the retrieval response. Both use clean-minus-patched correct-minus-distractor margins at the query. Source-attention mass is the probability assigned to the historical value; source-key recovery is the margin gain after restoring its clean key. Natural-text head ablation uses NLL differences (Appendix F.3).

Prompt construction. The key–value task uses 1,020 tokens within the model's 1,024-token context, needle IDs 3000–3199, filler IDs 500–2999, and query position $q = 1019$. Distractor-key-to-query spacing d is sampled uniformly from the integers 300–699; the distractor and target keys occupy $r_2 = q - d$ and $r_1 = \max(2, r_2 - \max(8, \lfloor d/2 \rfloor))$, with values at $r_1 + 1$ and $r_2 + 1$. Carrying changes the historical predecessor and patches the update at its value. Matching exchanges the historical keys, $AB \ldots CD \ldots A_q \mapsto CB \ldots AD \ldots A_q$, and patches the query update. Each donor copies its clean prompt: filler, value tokens, positions, and query remain identical, while only the specified keys change. Each condition evaluates 200 paired prompts.

B.2 Predecessor Specificity

The paired predecessor donor changes the tested association. An off-site identity donor changes another historical token that remains causally accessible to the patched value; a same-predecessor donor supplies a further comparison. The specificity contrast is $\Delta_{\mathrm{paired}} - \Delta_{\mathrm{offsite}}$. It is positive in the evaluated hybrid Carrying layers and Transformer L3; Transformer L6 shows effects that are small or negative in these same comparisons (Table S2).

Table S2: **Carrying identity controls.** Paired, off-site, and same-predecessor donor effects in margin units, grouped by model/layer. Each block lists runs 42/43/44. Contrasts use unrounded values.

Model	Layer	Paired	Off-site	Same predecessor	Specificity	Paired minus same predecessor
GDN / LR $\times 0.1$	6	+3.590	-0.010	+0.015	+3.601	+3.576
		+4.793	-0.278	-0.004	+5.071	+4.796
		+3.977	-0.018	+0.063	+3.995	+3.914
GDN hybrid	2	+2.433	+0.093	-0.011	+2.339	+2.444
		+3.034	-0.004	+0.020	+3.037	+3.013
		+2.527	-0.090	-0.021	+2.617	+2.549
GDN / Conv width 2	2	+2.801	-0.103	-0.050	+2.904	+2.852
		+3.164	-0.323	+0.031	+3.487	+3.133
		+3.132	+0.047	0.000	+3.085	+3.132
SWA hybrid	2	+2.499	+0.043	-0.021	+2.456	+2.520
		+2.427	-0.246	-0.001	+2.673	+2.428
		+2.165	-0.193	+0.010	+2.358	+2.155
Transformer	3	+1.714	-0.050	-0.003	+1.764	+1.717
		+0.792	-0.099	-0.002	+0.891	+0.794
		+2.485	+0.057	+0.034	+2.428	+2.451
Transformer	6	-0.090	+0.002	+0.004	-0.092	-0.094
		-0.621	-0.024	+0.016	-0.597	-0.637
		-0.070	-0.004	+0.001	-0.066	-0.071

The lag scan uses training run 42 and offsets 1/2/4/8/16/32/64, with 200 prompts per probe condition. Its lag-one concentration identifies the token before the value as the local relation tested by the subsequent Carrying interventions.

B.3 Query-Association Controls

The query-control study compares source switching, removal of the matching query relation, and filler perturbations. It retains the original target/distractor labels when a source switch changes the supported continuation. Figure S1 shows both intervention response and clean margin.



Figure S1: **Query controls.** (A) Signed clean-minus-patched margin. (B) Clean margin with the same target/distractor labels.

The full-layer scans in Figure 2 compare the original allocations across architectures; the controls here establish how their query responses depend on the association.

B.4 Content Selection with Values Held Fixed

A changed query update can reflect several downstream effects of the key-exchange donor. We isolate content selection by writing the receiver's pre-projection query output as PV, with attention pattern P and values V. The four combinations below separate selection from value content.

Algorithm S1. Separating attention selection and value content

Input: clean prompt c, historical-key-exchange donor d, receiver layer ℓ.

1. Record (P_c, V_c) and (P_d, V_d) at the receiver.
2. Rerun the clean prompt with query output $P_d V_c$, $P_c V_d$, or $P_d V_d$.
3. Apply the normal output projection and continue the remaining model.
4. Subtract each resulting margin from $m(P_c V_c)$ to obtain pattern-only, value-only, and joint effects.

Table S3: **Selection with fixed values.** Training-run mean ± sample SD at the two tested receivers in each condition. Each probe condition evaluates 200 prompts.

Model	Receiver	Pattern only	Value only	Pattern + value
GDN / Baseline	L3	5.85 ± 0.45	0.00 ± 0.05	5.89 ± 0.41
	L7	1.32 ± 0.39	-0.01 ± 0.01	1.31 ± 0.39
GDN / LR $\times 0.1$	L3	0.02 ± 0.07	-0.02 ± 0.12	0.06 ± 0.11
	L7	9.16 ± 0.95	0.01 ± 0.02	9.21 ± 0.95
SWA / Baseline	L3	5.34 ± 0.21	0.00 ± 0.00	5.38 ± 0.22
	L7	0.31 ± 0.06	0.01 ± 0.00	0.32 ± 0.06
SWA / LR $\times 0.1$	L3	0.02 ± 0.03	-0.02 ± 0.02	0.03 ± 0.07
	L7	8.14 ± 1.33	0.06 ± 0.03	8.20 ± 1.36
Transformer	L6	4.24 ± 1.78	0.00 ± 0.02	4.26 ± 1.72
	L7	2.40 ± 2.84	-0.01 ± 0.00	2.38 ± 2.84

Pattern-only interventions reproduce the shift from the earlier to the later global receiver in both hybrid families, while value-only effects remain small at the selected receivers (Table S3). In ev-

ery tested run, the receiver with the stronger pattern-intervention effect agrees with query-update Matching. This agreement identifies content selection as part of the relocated query response.

B.5 Source-Key Propagation and Restoration

The next intervention follows a changed Carrying update into a candidate receiver. Propagation starts from a clean run and inserts only the altered source K or V. Restoration starts from the sender-patched run and puts back the clean source K. This distinguishes the channel carrying the perturbation from recovery of the affected prediction.

Algorithm S2. Following and restoring a sender perturbation

Input: clean prompt, donor Carrying update, sender layer, receiver layer, historical value position s.

1. Run the clean prompt; cache its receiver-source K_c, V_c and margin m_c.
2. Patch the donor update at the sender's position s; cache receiver-source K_p, V_p and perturbed margin m_p.
3. In clean recipient runs, insert K_p, V_p, or both at s; measure margin loss from m_c.
4. In the sender-patched run, restore K_c at s; measure recovery $m_{\text{restored}} - m_p$.
5. For the off-site comparison, add the same repair increment $K_c - K_p$ at another visible historical position; its recovery is $m_{\text{offsite}} - m_p$.

In baseline and early-LR-reduction conditions for both hybrid families, K-only propagation closely follows full-path loss and V-only propagation has little effect (Table S4). The classical key-composition mechanism therefore appears at both the original and redirected routes (Elhage et al., 2021; Olsson et al., 2022).

Table S4: **Path-channel measurements in the two hybrid families.** Each row is one run. Full propagates the sender patch; K, V, and K+V transfer the specified channels into a clean receiver.

Model	Route	Full-path loss	K	V	K+V	K/full
GDN / LR $\times 0.1$	6→7	3.162	3.164	-0.039	3.162	1.001
GDN / LR $\times 0.1$	6→7	3.741	3.738	0.031	3.741	0.999
GDN / LR $\times 0.1$	6→7	4.355	4.366	0.000	4.355	1.003
GDN hybrid	2→3	2.236	2.165	0.025	2.166	0.969
GDN hybrid	2→3	2.645	2.637	-0.028	2.637	0.997
GDN hybrid	2→3	2.317	2.312	-0.005	2.313	0.998
SWA hybrid	2→3	2.196	2.196	-0.012	2.195	1.000
SWA hybrid	2→3	2.648	2.647	-0.064	2.646	0.999
SWA hybrid	2→3	2.046	2.045	0.011	2.045	1.000
SWA / LR $\times 0.1$	6→7	3.585	3.585	0.050	3.585	1.000
SWA / LR $\times 0.1$	6→7	3.213	3.209	0.064	3.213	0.999
SWA / LR $\times 0.1$	6→7	2.637	2.635	0.020	2.637	0.999

The same key-dominated propagation holds after convolution removal and early lag-one suppression. Restoring K at the correct source recovers the prediction, while an off-site repair does not. Table S5 combines the channel effects with paired uncertainty of restoration.

Table S5: **Tracing and restoring relocated routes.** Full/K/V propagation, off-site recovery, and correct-source K recovery in margin units. Brackets give paired 95% bootstrap intervals from 10,000 resamples; specificity is correct-source minus off-site recovery. Each condition uses 200 prompts. The two 318M rows test different receivers in the same run.

Condition / run	Route	Full-path loss	K-only	V-only	Off-site	Source-key recovery [95% CI]	Specificity [95% CI]
No convolution / 42	L5→L7	2.766	2.770	-0.018	-0.493	2.78 [2.42, 3.17]	3.27 [2.82, 3.77]
No convolution / 43	L4→L7	1.751	1.747	-0.012	-1.678	1.76 [1.50, 2.03]	3.43 [2.93, 3.96]
No convolution / 44	L4→L7	3.023	3.020	-0.002	-1.861	3.02 [2.61, 3.44]	4.88 [4.19, 5.61]
Early lag-1 mask / 42	L6→L7	4.176	4.180	0.001	-1.392	4.17 [3.71, 4.65]	5.57 [4.94, 6.21]
Early lag-1 mask / 43	L2→L3	3.932	3.917	0.050	-3.539	3.81 [3.39, 4.22]	7.34 [6.64, 8.05]
Early lag-1 mask / 44	L6→L7	2.284	2.285	-0.001	-0.376	2.28 [1.98, 2.60]	2.66 [2.30, 3.03]
318M Baseline / 44	L6→L11	2.216	0.903	-0.022	-0.064	0.51 [0.40, 0.62]	0.57 [0.45, 0.70]
318M Baseline / 44	L6→L7	2.216	1.723	0.026	-0.267	1.33 [1.17, 1.51]	1.60 [1.41, 1.80]

Propagation through multiple stages. The 318M Baseline run at seed 44 exhibits source-key dependence at more than one global stage. K-only propagation carries part of the sender-patch effect into both the intermediate receiver and the later receiver with peak Matching; restoring clean

source K at either site recovers part of the loss (Table S5). The peak pair thus identifies the strongest local responses within a computation that can span multiple stages.

B.6 Source Access and Carrying–Copying Interactions

Source deletion. Deleting the correct source strongly disrupts retrieval in both GDN hybrid baseline and early-LR-reduction conditions. Deleting a distractor, an off-site token, or a random source has little effect (Table S6). The same distinction holds at the original and relocated receivers. Retrieval depends on retaining access to the prepared source, supporting the source-retention interpretation in Section 5.

Table S6: **Source-deletion controls.** Clean-minus-deleted margins; 200 prompts per row.

Model	Correct	Distractor	Off-site	Random
GDN Early LR reduction	3.77543	-0.01371	-0.00193	-0.00095
GDN Early LR reduction	4.59043	-0.01569	-0.00366	-0.00306
GDN Early LR reduction	5.20782	-0.02279	-0.00230	-0.00068
GDN Baseline	3.02540	-0.00465	-0.00123	-0.00107
GDN Baseline	3.40174	-0.00147	0.00153	0.00079
GDN Baseline	3.12901	-0.00465	-0.00075	0.00075

Factorial Carrying–Copying interventions. A factorial experiment jointly varies source access and the information prepared upstream. It crosses the Carrying update (clean or donor), the query-to-source edge (open or suppressed), and the source value (clean or distractor). The source-edge intervention sets the query-to-source attention probability to zero at the specified receiver and renormalizes the remaining causal attention weights within each head; the Matching probe remains the historical-key-exchange query-update intervention defined in Algorithm 3.

Let $m_{b,e,v}$ denote the output margin, with $b, v = 0$ for the clean state and 1 for the donor or distractor state. The conditional effects are

$$\Delta_{\text{Carrying}}(e) = m_{0,e,0} - m_{1,e,0}, \qquad \Delta_{\text{Copying}}(e) = m_{0,e,0} - m_{0,e,1}.$$

The contrast $\Delta_{\text{Carrying}}(\text{open}) - \Delta_{\text{Carrying}}(\text{closed})$ measures how the upstream effect depends on the tested source edge. Each factorial condition evaluates 200 prompts.

At the principal source–receiver pairs, closing the edge strongly reduces the Carrying effect (Table S7). The same pattern appears in the shallow baseline route, the deep route after early LR reduction, and Transformer. A later receiver tested in a shallow-route hybrid leaves much of the Carrying effect intact: suppressing access at that receiver leaves the principal earlier route available. The result connects the upstream effect to the particular receiver through which the source is used.

Table S7: **Source-edge dependence across training conditions and scales.** Sender/receiver identify the tested layers. Columns show conditional Carrying and Copying effects, in margin units. Each condition uses 200 prompts. The 318M Baseline rows at 30K follow training-run order 42/43/44; the 50K row follows the first run.

Model / step	Sender	Receiver	Carrying effect		Copying effect	
			Edge open	Edge closed	Edge open	Edge closed
318M / Baseline / 30K	6	7	2.197	0.177	3.303	-0.003
318M / Baseline / 50K	6	7	1.706	0.137	2.624	-0.002
318M / Baseline / 30K	10	11	1.747	0.000	4.441	0.001
318M / Baseline / 30K	6	7	2.135	0.449	2.637	-0.011
GDN / LR $\times 0.1$ / 30K	6	7	3.387	0.000	7.044	0.000
GDN hybrid / 30K	2	7	2.623	2.152	0.978	0.000
GDN / Enriched text / 30K	2	7	2.869	2.729	0.280	0.000
GDN / Natural text / 30K	2	7	2.570	2.102	0.974	0.000
Transformer / Enriched text / 30K	5	6	2.279	0.007	4.673	-0.003
Transformer / Natural text / 30K	5	6	3.212	0.023	6.115	-0.002
GDN / Formation baseline / 30K	2	3	2.564	0.085	4.731	0.003
GDN / LR $\times 0.1$ (formation) / 30K	6	7	3.375	0.000	7.078	0.000
Transformer / 30K	5	6	2.987	0.019	5.790	-0.002

With the source edge open, changing the selected value affects the continuation; closing it makes the Copying effect small (Table S7). Together, the conditional Carrying and Copying contrasts connect upstream preparation and value transmission to the tested receiver-source edge.

Graded source and edge interventions. We vary sender-patch strength and source-edge suppression jointly over $0, .25, .5, 1$, evaluating 200 prompts at every pair of doses. Figure S2 shows a graded interaction: the effect of changing the sender decreases as access to its source is suppressed.

The two-dimensional comparison follows how source preparation and access jointly affect the output, extending the open/closed endpoint comparison.

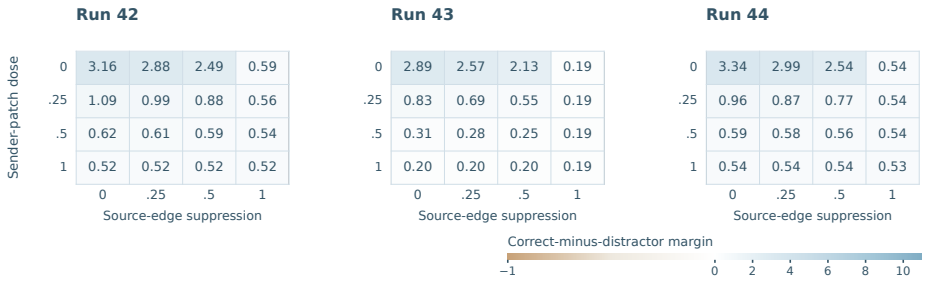

Figure S2: **Preparation and source access interact across intervention strengths.** Correct-minus-distractor margins across sender-patch and source-edge doses in three GDN Baseline runs. Zero is clean and one is full intervention. Each condition uses 200 prompts.

B.7 Head-Level Matching and Copying

Copying holds an attention head's pattern fixed and substitutes its distractor value at the source; its effect is the clean-minus-patched output margin. Figure S3 aligns source-attention mass, query-update Matching, and Copying at identical model, run, checkpoint, layer, and head coordinates. The dominant Matching and Copying heads coincide in the original hybrid receivers, GDN after early LR reduction, and Transformer. Heads with strong causal Matching also attend strongly to the historical source, linking these interventions to the conventional induction-head readout. Rankings among weaker heads vary. Transformer comparisons use a common receiver layer across runs, alongside each run's full-network layer scan.

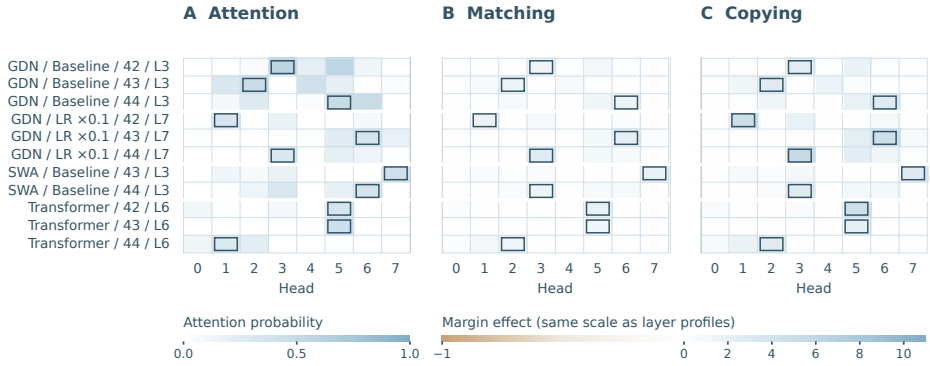

Figure S3: **Source attention, Matching, and Copying at matched heads.** A shows attention probability; B/C show Matching and Copying in logit-margin units on the same scale as the layer profiles. Outlines mark row maxima. Labels give model, run, and layer. Each measurement uses 200 prompts. The matched-head measurements cover SWA seeds 43/44 and seeds 42/43/44 for the other conditions.

C Carrying Conditions and Circuit Allocation

C.1 Hybrid Offset, Timing, and Layout Controls

GDN Convolution width 2 retains the current token and lag one; No convolution removes short convolution. Early lag-1 mask suppresses only its lag-one channel, while lag-2 mask supplies an offset comparison. Early LR reduction scales all parameters in the selected efficient blocks, including FFN and norms, by a learning-rate multiplier of 0.1 through 3K for GDN and 4K for SWA. All global receivers remain available as these interventions alter local access or learning in the supporting blocks.

The offset and timing controls test which changes to predecessor support alter circuit allocation. Early lag-two suppression and late lag-one suppression retain shallow circuits (Table S8); Figure S4 traces their development alongside the endpoint profiles in Figure 3.

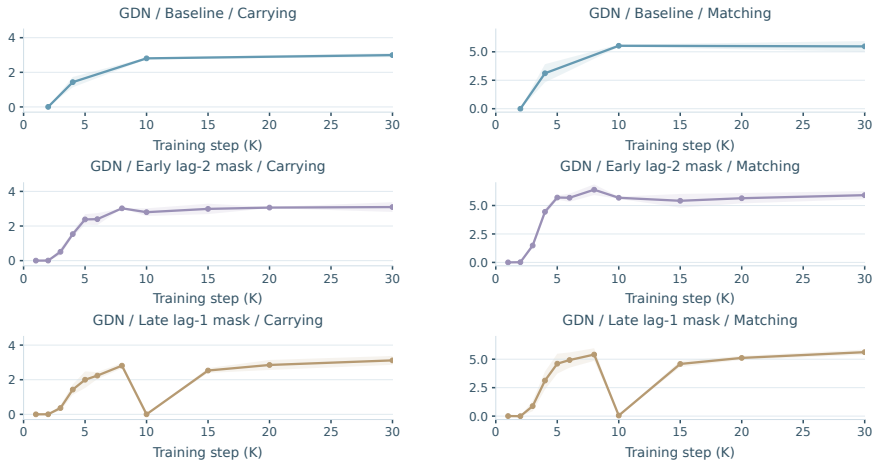

Figure S4: **Selective-intervention trajectories.** Mean and sample SD at shared checkpoints for Baseline, early lag-2 mask, and late lag-1 mask.

Table S8: **Offset and timing contrasts.** Peak Carrying/Matching layers at 30K in run order 42/43/44; each probe condition uses 200 prompts.

Condition	Carrying layers	Matching layers
Baseline	L2 / L2 / L2	L3 / L3 / L3
Lag-1 masked early	L6 / L2 / L6	L7 / L3 / L7
Lag-2 masked early	L2 / L2 / L2	L3 / L3 / L3
Lag-1 masked late	L2 / L2 / L2	L3 / L3 / L3

Moving the global receivers. The shifted layout places full attention at L1 and L5 while retaining the remaining GDN layers. Source preparation and query selection now concentrate around the later available receiver (Figure S5). The available receivers and their upstream support jointly constrain the allocation.

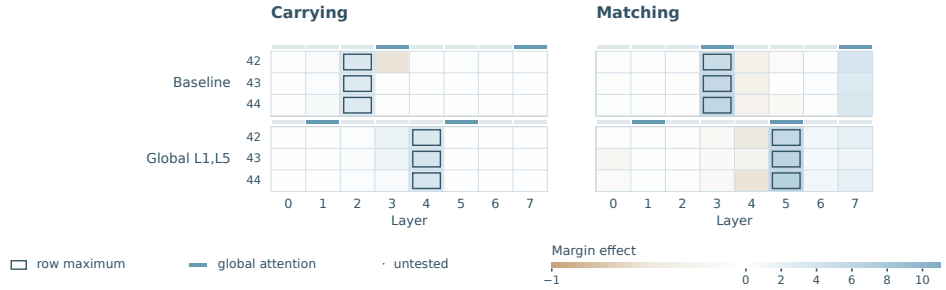

Figure S5: **The allocation follows a changed global layout.** Full Carrying and query-update Matching layer profiles in the baseline and shifted-layout models. Outlines mark row maxima; rails mark full attention. Colors share the layer-profile scale of Figure 5. Each row is one training run; each probe condition uses 200 prompts.

The corresponding recall comparison treats global-layer order as an additional architectural control (Table S9). Moving the global sites preserves strong recall in several target-token groups, alongside the change in circuit allocation.

Table S9: **Global-layer order as an architectural control.** GDN full-attention sites are L3/L7 in Baseline and L1/L5 in the shifted layout. PPL at 30K is mean ± sample SD over three matched training runs, using 10,000 sequences per run. Recall columns match Table 2; overall PPL is included here. Lower values are bold.

			Single	Multiple continuation			Identifier
Model	Variant	Overall	Far	Near	Mid	Far	reuse
GDN	Baseline	40.17 ± 2.00	2.38 ± 0.04	2.75 ± 0.39	16.34 ± 1.99	25.98 ± 1.23	29.00 ± 1.43
	Global L1,L5	**38.08 ± 1.28**	**2.26 ± 0.04**	**2.54 ± 0.01**	**13.49 ± 1.55**	**23.82 ± 1.64**	**23.54 ± 2.93**

C.2 Transformer Controls

Early learning-rate reduction. This Transformer comparison extends Section 3.2. The layout-matched condition reduces LR at L0–2 and L4–6, the positions occupied by efficient layers in the hybrid comparison. Other conditions target the lower six layers, upper two layers, all eight layers, or six randomly selected layers. The resulting profiles show that changing early learning can also reorganize a homogeneous Transformer (Figure S6).

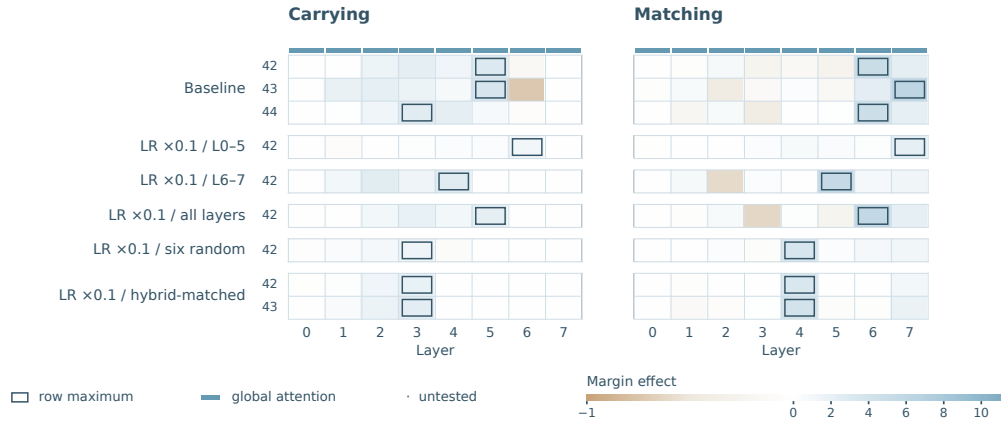

Figure S6: **Transformer LR reduction profiles.** Carrying and query-update Matching use common L0–L7 coordinates. Rows retain all measured training seeds; outlines mark maxima and rails mark full attention. Carrying and Matching share the layer-profile color scale. "Layout-matched" reduces LR in the layers corresponding to the hybrid's efficient sites.

The behavioral comparison pairs training seeds 42 and 43 between Baseline and the layout-matched condition (Table S10). The shifted Transformer condition has higher mean overall, distant multiple-continuation, and identifier-reuse PPL. In the hybrid comparison, LR reduction shifts the route while lowering mean PPL on several recall conditions.

Table S10: **Transformer LR reduction on matched training seeds.** PPL uses the same 10,000-sequence evaluation per run; lower is better. Thick rules separate matched training runs, and bold marks the lower value in each pair.

Run	Variant	Overall	Multiple mid	Multiple far	Identifiers
42	Baseline	**40.00**	**13.66**	**25.94**	**24.57**
	LR ×0.1	53.18	21.23	34.60	36.11
43	Baseline	**42.93**	**16.39**	**25.53**	**25.45**
	LR ×0.1	49.04	17.45	32.60	33.60

Local-head constraints. The Transformer local-head constraint gives one head in selected layers access only to the current token and a specified predecessor offset. All other heads retain full causal attention, and training uses natural text. The persistent lag-one condition keeps this constraint throughout training and evaluation. Early constraints are removed after 3K steps, either in the lower six layers or throughout the network. An early lag-two constraint changes the supported offset. These schedules separate persistent local access from a temporary bias during formation.

The persistent previous-token constraint produces the strongest Carrying response in a shallow layer across runs (Table S11). Removing the constraint yields more variable Carrying locations, while the corresponding query response develops at intermediate layers. The early lag-two comparison also changes the allocation, showing that the schedule and form of the local constraint shape the resulting computation. The persistent condition's final recall outcomes are shown in Table S15.

Table S11: **Transformer source preparation under local attention constraints.** The constrained head is head 0 (H0). Layer lists give peak Carrying in run order 42/43/44; the all-layer early condition uses run 42. Each probe condition evaluates 200 prompts.

Condition	Constrained layers	Active interval	Carrying layer
Baseline	None	None	L5 / L5 / L3
Lag-1 persistent	L0–L5	Train and evaluation	L2 / L2 / L2
Lag-1 early	L0–L5	First 3K steps	L2 / L4 / L3
Lag-1 all-layer early	L0–L7	First 3K steps	L2
Lag-2 early	L0–L5	First 3K steps	L3 / L4 / L3

C.3 Circuit Allocation in the 318M Hybrid

The 16-layer, approximately 318M GDN hybrid model extends the allocation comparison to four global stages. Full attention occupies L3/L7/L11/L15. The four efficient groups occupy L0–2, L4–6, L8–10, and L12–14. Figures and tables name the physical layers affected by each intervention, together with its LR multiplier or convolution change. The default early-LR window is 0–3K.

Figure S7 shows that reducing early LR in different groups can favor early, intermediate, or late routes. LR reduction in nonadjacent groups produces more variation across runs. The strongest Carrying and Matching can also occupy nonadjacent stages, consistent with the multi-stage dependence measured by source restoration.

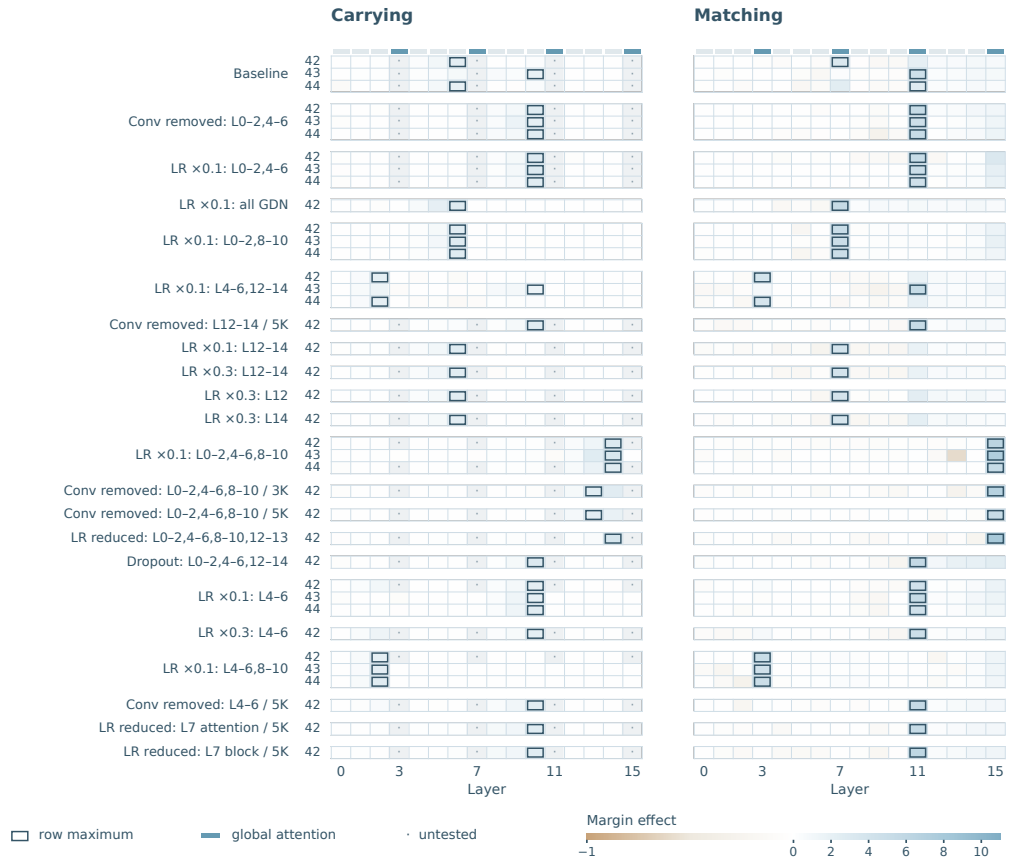

Figure S7: **318M allocation across all tested conditions.** Carrying and query-update Matching retain all 47 model/run rows across the tested conditions, with aligned L0–L15 coordinates and a common signed scale. Outlines mark measured maxima, dots mark untested coordinates, and the upper rail marks global attention. Labels specify group, LR multiplier, and duration; unspecified group LR reduction uses a 0.1 multiplier for the first 3K steps.

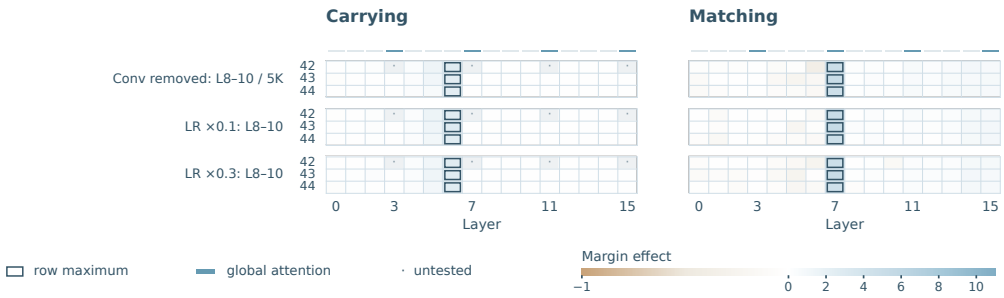

Figure S7 (continued): The remaining 318M conditions, using the same layer coordinates and signed color scale. Column headings, layer ticks, and the legend are repeated for navigation.

D Predecessor Support and Circuit Formation

D.1 Checkpoint Schedule and Layer Selection

The window comparison evaluates 1K–10K in 1K increments, then 15K/20K/30K. The architecture–data comparison evaluates .5K, 1K, 1.5K, 2K, 2.5K, 3K, 4K, 5K, 8K, 10K, 15K, 20K, and 30K. Every condition has three runs, with Carrying and query-update Matching measured at all eight layers. SWA window 2/8/16 Matching uses the historical-key-exchange query-update intervention throughout. Each probe condition uses 200 prompts.

The main curves hold each run's endpoint-selected layer fixed throughout training. Fixed-coordinate curves instead compare Carrying at L2 and Matching at L3 across architectures, with additional Transformer receivers showing its later query response (Figure S8). Hybrid responses develop earlier at these shared sites. Both readouts depend on the current downstream computation and track functional formation. The fixed-site and endpoint-selected views compare common coordinates and each run's eventual circuit; all-layer plots show where responses emerge.

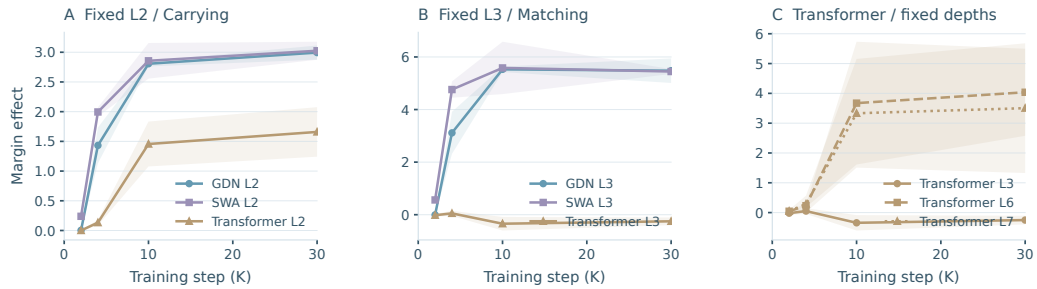

Figure S8: **Formation at fixed physical layers.** Carrying at L2 and query-update Matching at L3 compare the same sites across architectures; the right panel follows additional Transformer receiver layers. Lines and bands show means and sample SD across three runs. Layer coordinates remain fixed through training, with 200 prompts per probe condition.

D.2 Training-Data Controls and Layerwise Trajectories

Figure S9 compares replicated Carrying trajectories under induction-enriched text (Appendix A.2), natural text, and separately trained random-retention controls. Enrichment advances the Transformer trajectory; hybrid natural/enriched trajectories rise in a similar early interval. These comparisons relate training evidence for the predecessor relation to the architectural support supplied by efficient layers.

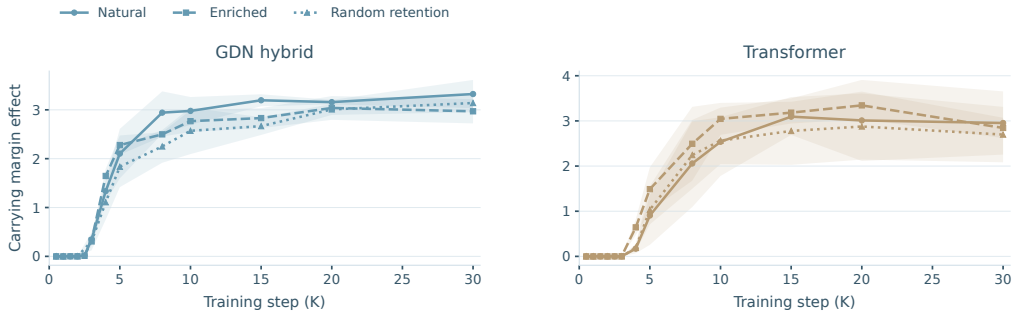

Figure S9: **Formation under different training streams.** Means and sample SD over three runs for natural, induction-enriched, and random-retention text. Each series retains its evaluated checkpoints.

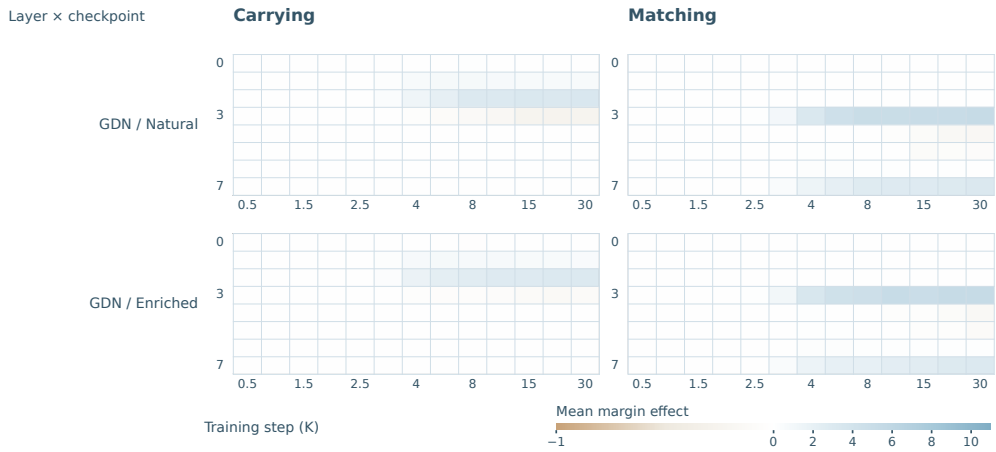

Figure S10: **Layerwise circuit formation.** GDN natural/enriched training: all eight layers and 13 measured checkpoints are retained. Colors show three-run mean signed effects. Continued panels show Transformer and SWA windows using the same scale; SWA has a different checkpoint schedule.

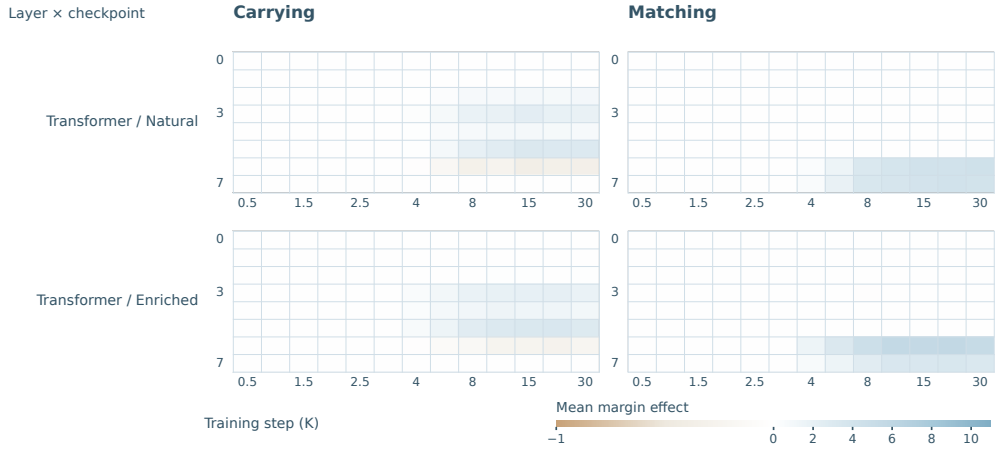

Figure S10 (continued): Transformer with natural and induction-enriched training text. All layers and evaluated checkpoints are retained.

D.3 SWA Window Size: Onset and Final Responses

For each threshold, onset is the first of two consecutive measured checkpoints at or above it; Table S12 gives the preceding interval. For Carrying, window 2 crosses thresholds $0.1, 0.25, 0.5$ in $(1, 2]$K, ahead of windows 8/16 in $(2, 3]$K; all three converge at threshold 1. For Matching, windows

2 and 8 tie at thresholds 0.1/0.25, while window 2 leads both wider windows at 0.5/1. The convergence at the strictest threshold thus applies only to Carrying. At later checkpoints, wider windows develop larger Carrying responses, as the complete trajectories show (Figure S11).

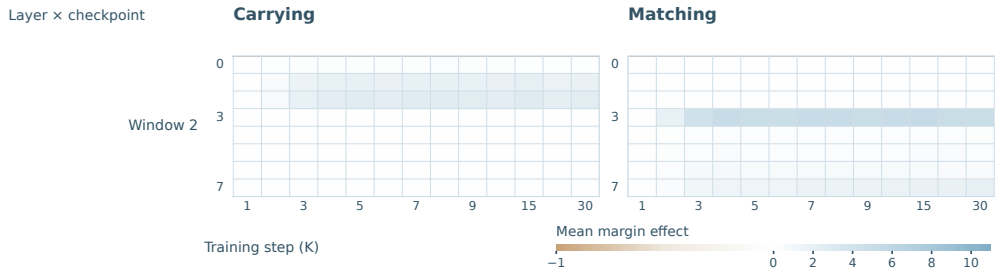

Figure S10 (continued): SWA window 2: all layers and evaluated checkpoints.

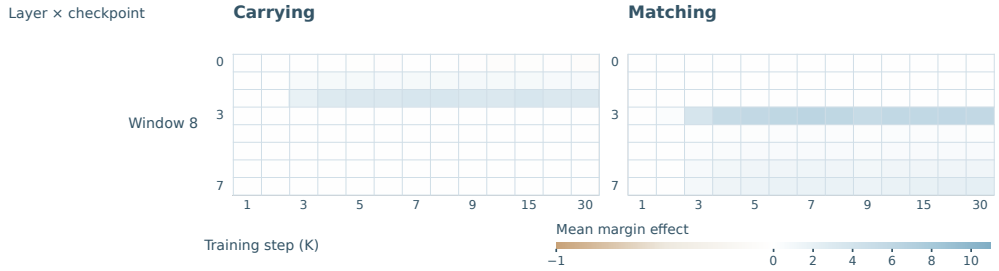

Figure S10 (continued): SWA window 8: all layers and evaluated checkpoints.

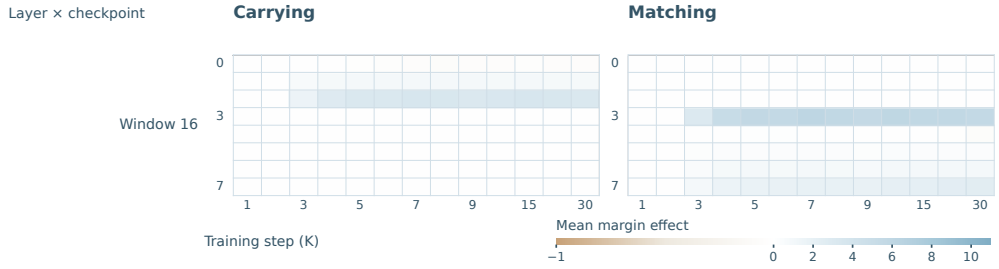

Figure S10 (continued): SWA window 16: all layers and evaluated checkpoints.

Table S12: **Formation intervals across thresholds.** Units: thousands of steps. A shared interval applies to all three runs; otherwise intervals follow run order 42/43/44. Each trajectory follows its fixed endpoint-selected layer.

Condition	Readout	$\tau = .1$	$\tau = .25$	$\tau = .5$	$\tau = 1$
SWA window 2	Carrying	$(1, 2]$	$(1, 2]$	$(1, 2]$	$(2, 3]$
	Matching	$(1, 2]$	$(1, 2]$	$(1, 2]$	$(1, 2]$
SWA window 8	Carrying	$(2, 3]$	$(2, 3]$	$(2, 3]$	$(2, 3]$
	Matching	$(1, 2]$	$(1, 2]$	$(2, 3]$	$(2, 3]$
SWA window 16	Carrying	$(2, 3]$	$(2, 3]$	$(2, 3]$	$(2, 3]$
	Matching	$(2, 3]$	$(2, 3]$	$(2, 3]$	$(2, 3]$
GDN / Natural	Carrying	$(2.5, 3]$	$(3, 4]$; $(2.5, 3]$; $(2.5, 3]$	$(3, 4]$	$(3, 4]$
	Matching	$(2.5, 3]$; $(2, 2.5]$; $(2, 2.5]$	$(2.5, 3]$	$(3, 4]$; $(2.5, 3]$; $(2.5, 3]$	$(3, 4]$; $(2.5, 3]$; $(2.5, 3]$
GDN / Enriched	Carrying	$(2.5, 3]$	$(2.5, 3]$	$(3, 4]$	$(3, 4]$
	Matching	$(2.5, 3]$; $(2, 2.5]$; $(2, 2.5]$	$(2.5, 3]$	$(2.5, 3]$	$(3, 4]$; $(2.5, 3]$; $(2.5, 3]$
Transformer / Natural	Carrying	$(3, 4]$; $(3, 4]$; $(4, 5]$	$(3, 4]$; $(4, 5]$; $(5, 8]$	$(4, 5]$; $(4, 5]$; $(5, 8]$	$(4, 5]$; $(4, 5]$; $(5, 8]$
	Matching	$(3, 4]$	$(3, 4]$	$(3, 4]$; $(3, 4]$; $(4, 5]$	$(4, 5]$
Transformer / Enriched	Carrying	$(3, 4]$	$(3, 4]$	$(3, 4]$; $(4, 5]$; $(3, 4]$	$(4, 5]$
	Matching	$(3, 4]$	$(3, 4]$	$(3, 4]$	$(3, 4]$; $(4, 5]$; $(3, 4]$

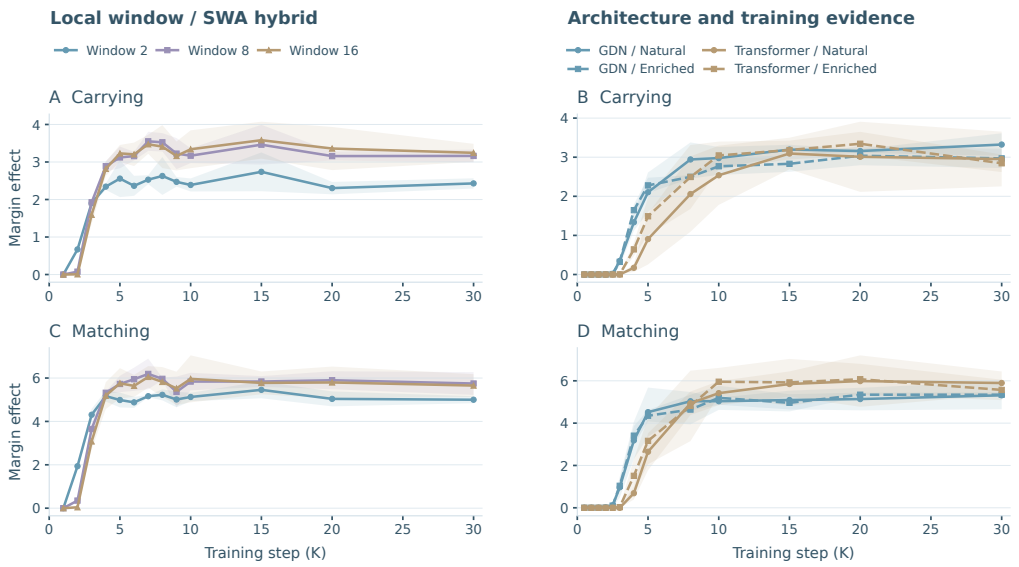

Figure S11: **Complete formation trajectories.** All 13 measured checkpoints through 30K; bands show training-run sample SD.

E Circuit Tests in Pretrained Qwen Models

E.1 Model Configuration and Circuit Localization

Qwen3-4B is a 36-layer full-attention language model with 32 query heads and eight key/value heads. Qwen3.5-4B has a 32-layer language backbone with layout $[\mathrm{GDN}^3, \mathrm{FullAttn}]^8$, hidden width 2,560, and a short-convolution kernel of four. Its global-attention layers have 16 query heads and four key/value heads; its GDN layers use 16 QK heads and 32 value heads (Qwen Team, 2025; 2026b). Both are evaluated at their released checkpoints, with zero-based layer numbering.

Localization inputs and intervention sites. The fixed-token association scans use a fixed single-token key–value tuple per input condition and sampled background filler. Qwen3 uses token IDs 94723/91807/96498/97535 and a filler pool of 4,868 IDs; Qwen3.5 uses 199437/166039/217654/227991 and 32,816 filler IDs. Localization covers both models; Qwen3.5 supplies the restoration and dose experiments. Each condition uses 200 prompts.

The full-layer scans place Qwen3's strongest Carrying and Matching in separated full-attention layers. In Qwen3.5, the strongest Carrying lies in a GDN layer immediately before the strongest global Matching, with secondary responses deeper in the model (Figure 5). The component experiments trace dependence from this main GDN source into the following global receiver.

Value transmission and grouped-query attention. Fixed-attention Copying concentrates in the same retrieval layers identified by Matching. Its head-level effects align with the key/value groups that carry the selected source: each key/value head serves four query heads. In the main retrieval layer, the dominant groups are Qwen3 KV H5 (Q H20–H23) and Qwen3.5 KV H3 (Q H12–H15). Localization and component restoration use separately sampled sets of 200 prompts each.

E.2 Prompt Calibration and Source-Key Restoration

Input calibration and clean solvability. The Qwen3.5 input constructor selects four distinct non-special vocabulary tokens with NumPy seed 42. Candidate token IDs lie in $[120000, 246000)$; after removal of tokenizer space markers, the vocabulary strings must contain at least four ASCII alphabetic characters. A minimum-ID cutoff restricts the pool used for both keys/values and filler. The restricted-token conditions are defined by this vocabulary-ID cutoff and string filter. The prompt generator's distance parameter is sampled from 100–399 tokens.

Table S13 compares six input pools, each with 200 prompts. Each pool fixes its sampled key–value identities. Clean solvability uses the pairwise criterion $m_{\mathrm{clean}} > 0$. Calibration and component experiments use separately sampled prompts.

Table S13: **Clean-solvability calibration for Qwen3.5.** Each input-pool condition evaluates 200 prompts. All sampled pools are shown, each with its own sampled key–value tuple.

Minimum token ID	Pool size	Mean clean margin	$m_{\text{clean}} > 0$
120,000	32,515	5.601	100.0%
150,000	32,513	-2.551	16.0%
180,000	25,461	2.089	71.0%
200,000	17,886	1.063	89.5%
220,000	10,149	2.501	100.0%
240,000	2,455	6.732	96.5%

Component restoration. The component experiment uses the minimum-ID-180000 pool (25,461 IDs), with token IDs 209267/186154/223584/231557. It perturbs the historical-value block update at GDN L18 and restores a clean component at full-attention L19. The intervention restores the clean receiver component tensor. Because the sender output patch changes one token immediately before this receiver, its change to the receiver's pre-attention K is confined to the historical source position. Q and K are patched after their normalization and rotary transformation; V is patched at its projection output. Qwen3.5 full attention also has an output gate, distinct from the FFN gate: its doubled Q projection is split into query and gate channels. The gate tensor g of shape $(B, T, H_q d_h)$ multiplies the concatenated attention result before output projection, $W_O[\text{concat}(PV) \odot \sigma(g)]$. Gate restoration replaces g at this split; attention-output and FFN-output restoration replace the corresponding module outputs.

Source-K restoration recovers nearly all of the sender-patch loss, while Q, V, and gate restoration have little effect (Table S14). The same selective recovery holds within the subset whose clean margin is positive. That subset is selected before intervention and held fixed across all conditions. Source-key dependence therefore persists among prompts that the clean model solves by favoring the target over the distractor.

Table S14: **Qwen3.5 L18→L19 component restoration.** The parent sample has 200 paired prompts; the clean-positive subset contains 69% of that sample. Entries are archived means with 95% paired-prompt bootstrap intervals. The released clean-positive summary uses 2,000 bootstrap resamples. The same clean-positive prompts are used throughout all interventions.

Quantity	All prompts	Clean-positive subset
Clean margin	$1.850\,[1.453, 2.268]$	$2.956\,[2.490, 3.462]$
Sender-patched margin	$0.406\,[0.114, 0.707]$	$1.088\,[0.719, 1.488]$
K-restored margin	$1.826\,[1.423, 2.249]$	$2.931\,[2.457, 3.441]$
Sender-patch loss	$1.444\,[1.270, 1.630]$	$1.868\,[1.655, 2.099]$
K recovery	$1.420\,[1.252, 1.603]$	$1.843\,[1.639, 2.064]$
Q recovery	$-0.007\,[-0.015, -0.001]$	$-0.010\,[-0.021, -0.001]$
V recovery	$-0.009\,[-0.018, -0.002]$	$-0.015\,[-0.026, -0.004]$
Gate recovery	$-0.008\,[-0.015, -0.001]$	$-0.011\,[-0.022, -0.002]$
Attention-output recovery	$1.420\,[1.249, 1.605]$	$1.840\,[1.629, 2.066]$
FFN-output recovery	$0.028\,[0.005, 0.052]$	$0.033\,[0.000, 0.067]$

Perturbation-dose response. To follow the source dependence across perturbation strengths, we interpolate the historical-value sender update between clean and counterfactual states, $u_\alpha = (1 - \alpha)u_{\text{clean}} + \alpha u_{\text{cf}}$, then match its RMS to the clean update before insertion. Doses are interpolation coefficients $\alpha \in \{0, .25, .5, .75, 1\}$ before RMS matching; the resulting perturbation distance is $\|\widetilde{u}_\alpha - u_{\text{clean}}\|_2$. Every dose uses the same 200 prompts within an input condition. Receiver K and attention-output recovery are measured in the corresponding sender-perturbed run.

In the common-token condition, perturbed margins remain positive on almost every prompt, and K recovery grows with sender-patch strength (Figure S12). The restricted-token condition also shows graded recovery, with negative mean margins. Zero-dose effects remain near zero in both conditions.

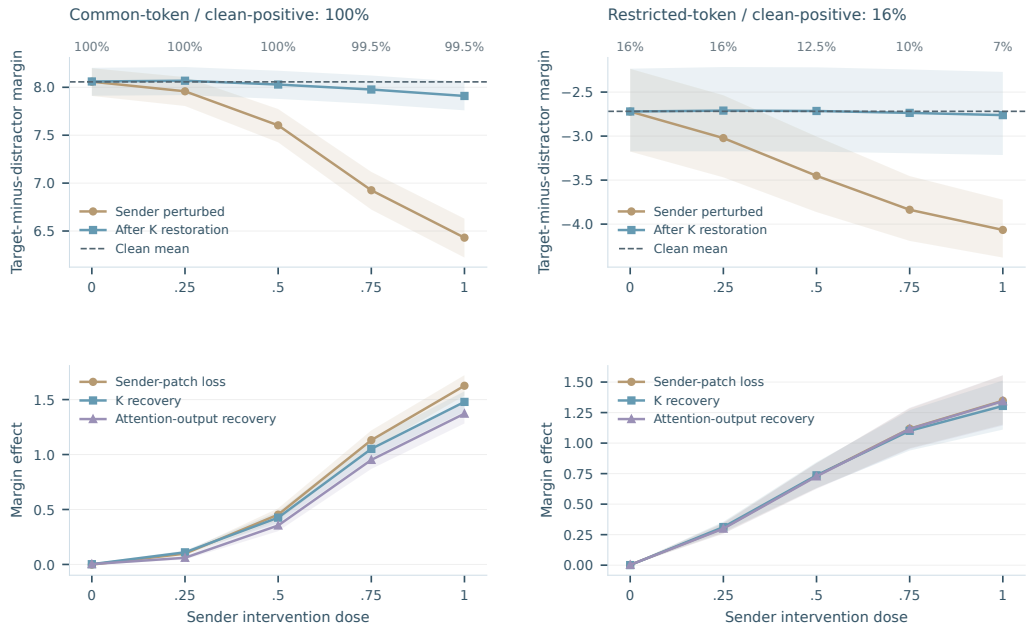

Figure S12: **Graded sender perturbation and receiver recovery in Qwen3.5.** Each condition uses 200 paired prompts across all five doses. Top: perturbed margin and margin after clean K restoration, with a clean-margin reference. Bottom: sender-patch loss and K/attention-output recovery. Bands are 95% paired prompt-bootstrap intervals. Printed percentages give the fraction of perturbed prompts with positive margin.

F Natural-Text Evaluation and Receiver Controls

These evaluations connect changes in the conditions for Carrying to natural-text recall and test whether alternative receivers support the same historical-source pathway.

F.1 Natural-Text Evaluation Protocol

Natural-text evaluation measures overall prediction, recall of earlier associations, and domain-specific reuse within the original input context. Target groups can overlap.

Sequences and scoring. Each checkpoint is evaluated on 10,000 sequences of 1,024 tokens: 2,000 Wiki, 4,000 Python code, and 4,000 Math. Python and Math each comprise two 2,000-sequence partitions from the same parent dataset. The same sequences and target-selection rules are used within each comparison; eight-layer and 318M results are reported separately. Effective token counts depend on the selected targets.

For a target $B = x_p$, write its immediate predecessor as $A = x_{p-1}$. The evaluator examines the preceding 512-token history. A *repeated association* is a target whose pair (A, B) has already appeared there. If u distinct tokens have followed A in that history, single-continuation recall has $u = 1$ and multiple-continuation recall has $u > 1$. This separates retrieval with one observed continuation from contexts containing competing continuations.

Distance refers to the most recent historical predecessor: $d = p - p_A - 1$, where p_A is its position. Near, mid, and far use 1–20, 21–100, and 101–512 tokens. This axis is termed *predecessor recency*.

Domains and reuse. Domain PPL scores all valid prediction positions in Wiki, Python, or Math. Long-range identifier reuse selects Python identifier reuse with predecessor distance above 100 tokens. The domain and recall panels therefore measure different prediction settings; their target sets can overlap.

Metric and aggregation. For the selected valid targets $\mathcal{T}$ in one evaluation condition, we compute

$$\mathrm{PPL}(\mathcal{T}) = \exp\left[\frac{1}{|\mathcal{T}|}\sum_{p\in\mathcal{T}} -\log p_\theta(x_p \mid x_{<p})\right].$$

For eight-layer domain results, we sum NLL and valid-token counts over the two Python or Math partitions within each run before exponentiating. The 318M domain export contains rounded partition PPLs; equal token counts allow their geometric mean to reconstruct each parent-domain PPL. Training-run means and sample SD are then computed from per-run PPL, keeping evaluation batches separate. Each comparison uses the same target selection. These evaluation groups differ from the paragraph-level training filter in Appendix A.2.

F.2 Domain Prediction and Recall under Carrying Interventions

Eight-layer models. Table S15 pairs whole-domain PPL with the recall conditions in Table 2 and the persistent lag-one head control (Appendix C.2). Early LR reduction lowers mean Python PPL and raises Wiki PPL in both hybrids, with smaller Math changes; the matched Transformer intervention raises all three domain means. GDN convolution removal produces another prediction profile. Recall effects also differ between single- and multiple-continuation targets. With architecture and training corpus fixed, early changes to blocks that develop Carrying can lead to different Matching allocations and final prediction profiles.

Table S15: **Domain prediction and recall in eight-layer models.** PPL at 30K, mean ± sample SD. Transformer Baseline/LR uses two matched runs; other comparisons use three. Python and Math each pool their two dataset partitions. The final pair is a separate Transformer control: each of L0–5 has one head restricted to self and lag one throughout training and evaluation, paired with its own unrestricted baseline. Bold marks the lowest mean within each comparison group.

A. Domain prediction

Model	Variant	Wiki	Python	Math
SWA	Window 2	90.44 ± 0.52	22.52 ± 1.27	51.96 ± 0.99
	Window 4	90.99 ± 0.15	24.41 ± 0.77	49.04 ± 0.12
	Window 4, LR ×0.1	100.58 ± 2.21	**21.11 ± 2.02**	49.54 ± 0.29
	Window 8	86.54 ± 1.67	21.99 ± 0.90	47.85 ± 0.07
	Window 16	**84.96 ± 2.02**	23.33 ± 1.16	**46.94 ± 0.16**
GDN	Baseline	**85.52 ± 0.92**	23.79 ± 3.04	**46.65 ± 0.66**
	LR ×0.1	93.80 ± 3.01	**20.64 ± 2.04**	47.18 ± 0.04
	Conv removed	90.04 ± 1.34	20.88 ± 1.51	47.03 ± 0.29
Transformer	Baseline	**98.51 ± 2.41**	**22.69 ± 3.19**	**49.32 ± 0.19**
	LR ×0.1	117.63 ± 1.97	30.80 ± 4.84	56.14 ± 0.35
	Unrestricted	98.44 ± 1.20	**24.21 ± 2.94**	49.39 ± 0.14
	Lag-1 heads	**90.89 ± 2.23**	26.35 ± 2.52	**48.82 ± 0.52**

B. Recall

		Single	Multiple continuation			Identifier
Model	Variant	Far	Near	Mid	Far	reuse
SWA	Window 2	2.26 ± 0.05	2.85 ± 0.36	14.97 ± 0.60	28.99 ± 0.43	28.05 ± 2.21
	Window 4	2.33 ± 0.06	2.86 ± 0.22	15.52 ± 1.09	26.61 ± 1.18	28.31 ± 3.97
	Window 4, LR ×0.1	2.64 ± 0.06	**2.68 ± 0.25**	**12.94 ± 1.70**	25.63 ± 0.87	23.49 ± 2.12
	Window 8	**2.25 ± 0.07**	2.78 ± 0.15	14.00 ± 0.94	24.19 ± 0.96	23.54 ± 2.86
	Window 16	2.26 ± 0.07	3.13 ± 0.34	14.31 ± 2.02	**22.40 ± 2.55**	**20.91 ± 6.76**
GDN	Baseline	**2.38 ± 0.04**	2.75 ± 0.39	16.34 ± 1.99	25.98 ± 1.23	29.00 ± 1.43
	LR ×0.1	2.59 ± 0.05	**2.64 ± 0.27**	12.52 ± 0.93	23.35 ± 0.34	21.02 ± 1.60
	Conv removed	2.70 ± 0.05	2.83 ± 0.73	**12.50 ± 0.28**	**21.75 ± 1.99**	**18.72 ± 4.57**
Transformer	Baseline	**2.66 ± 0.04**	**2.59 ± 0.14**	**15.03 ± 1.93**	**25.73 ± 0.29**	**25.01 ± 0.62**
	LR ×0.1	3.33 ± 0.04	3.24 ± 0.14	19.34 ± 2.67	33.60 ± 1.41	34.86 ± 1.77
	Unrestricted	2.62 ± 0.05	**3.26 ± 1.10**	**14.67 ± 1.81**	**24.95 ± 2.37**	**23.23 ± 6.46**
	Lag-1 heads	**2.51 ± 0.06**	3.31 ± 0.66	17.46 ± 1.76	27.09 ± 2.19	27.59 ± 5.92

Four-stage 318M model. Table S16 uses the same domain and recall columns. LR reduction at L4–6 or L0–2/L4–6/L8–10 lowers mean Python PPL, while Baseline has the lowest mean distant multiple-continuation and identifier-reuse PPL. Other interventions favor different domains or recall conditions. The table includes every retained configuration with a complete five-partition evaluation in this export. Table S17 retains the two other replicated allocations with their Baseline from the earlier evaluation.

Table S16: **Domain prediction and recall in the 318M hybrid.** Mean ± sample SD over three runs at 30K. Both panels use the same evaluation export. LR ×0.1 applies during the first 3K steps. Layers are zero-indexed. Domain pooling uses the equal-sized partition PPLs available in this export; partition values were recorded to two decimals. Bold marks the lowest mean in each column.

A. Domain prediction

Variant	Layers	Wiki	Python	Math
Baseline	All	71.94 ± 1.73	23.18 ± 0.62	43.80 ± 0.73
LR ×0.1	0–2,4–6	74.72 ± 1.49	23.71 ± 0.23	45.59 ± 0.82
LR ×0.1	0–2,8–10	72.63 ± 0.68	23.81 ± 3.31	44.32 ± 0.50
LR ×0.1	0–2,4–6,8–10	81.92 ± 0.36	22.28 ± 0.57	46.59 ± 0.15
LR ×0.1	4–6	72.98 ± 1.83	**21.81 ± 1.51**	43.89 ± 0.55
LR ×0.1	4–6,8–10	72.79 ± 2.25	23.58 ± 1.37	44.06 ± 0.25
LR ×0.1	8–10	71.08 ± 1.00	23.27 ± 1.53	**43.62 ± 0.25**
NoConv (5K)	8–10	**70.58 ± 0.40**	23.84 ± 1.37	43.68 ± 0.19

B. Recall

		Single	Multiple continuation			Identifier
Variant	Layers	Far	Near	Mid	Far	reuse
Baseline	All	2.32 ± 0.07	3.36 ± 0.37	14.54 ± 0.97	**20.85 ± 1.01**	**20.45 ± 2.73**
LR ×0.1	0–2,4–6	2.37 ± 0.08	3.09 ± 0.36	15.80 ± 1.18	23.75 ± 2.05	27.23 ± 4.26
LR ×0.1	0–2,8–10	2.32 ± 0.03	3.27 ± 0.81	15.78 ± 1.38	23.76 ± 0.38	27.63 ± 1.09
LR ×0.1	0–2,4–6,8–10	2.59 ± 0.06	**2.75 ± 0.23**	14.64 ± 0.68	24.22 ± 1.56	27.69 ± 3.64
LR ×0.1	4–6	2.33 ± 0.02	3.04 ± 0.56	**14.29 ± 0.99**	22.00 ± 1.20	24.30 ± 3.29
LR ×0.1	4–6,8–10	2.43 ± 0.07	3.16 ± 0.10	15.44 ± 1.57	23.91 ± 1.57	26.47 ± 3.49
LR ×0.1	8–10	**2.26 ± 0.03**	3.28 ± 0.61	14.69 ± 0.94	21.63 ± 0.86	22.85 ± 2.78
NoConv (5K)	8–10	2.33 ± 0.03	3.71 ± 1.18	14.82 ± 1.68	22.20 ± 2.62	24.73 ± 7.89

Allocation and downstream use. The intervention profiles span early, intermediate, and late Matching receivers (Figure S7). Their prediction differences accompany changes in early source preparation and the computation learned around it. Probe margin effects quantify dependence on a constructed association; natural-text prediction also draws on the surrounding context. How later layers use retrieved content and combine it with other signals offers a possible explanation for the varied outcomes (Section 5).

Table S17: **Additional 318M allocations.** Three-run mean ± sample SD at 30K. These two interventions have recall results from an earlier evaluation batch but no corresponding domain export. Their matched Baseline is retained here; comparisons are within this table.

		Single	Multiple continuation			Identifier
Variant	Layers	Far	Near	Mid	Far	reuse
Baseline	All	2.58 ± 0.10	3.47 ± 0.38	14.00 ± 0.97	21.97 ± 1.11	21.17 ± 3.08
Conv removed	0–2,4–6	2.55 ± 0.07	3.10 ± 0.44	14.47 ± 1.19	23.07 ± 0.89	26.45 ± 2.51
LR ×0.1	4–6,12–14	2.60 ± 0.16	3.13 ± 0.26	14.18 ± 0.74	23.22 ± 0.70	26.14 ± 1.21

F.3 Retrieval-Head Ablations on Natural Text

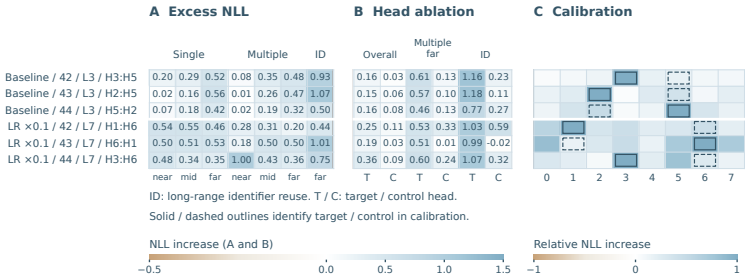

Figure S13: **Natural-text effects of matched retrieval heads.** A: target-minus-control NLL increase. B: target and control effects. C: all-head effects normalized by the row maximum absolute effect. Rows identify model, run, layer, and head pairs; outlines mark targets and controls. Each condition uses 200 sequences.

Head ablation measures $\Delta\mathrm{NLL} = \mathrm{NLL}_{\mathrm{off}} - \mathrm{NLL}_{\mathrm{clean}}$ within each target group. The selected heads contribute more strongly than comparison heads to overall prediction, distant multiple-continuation recall, and identifier reuse, and rank first in their layers for overall NLL damage (Figure S13). For Baseline seed 44, H5 is the query-update Matching maximum and H2 the source-attention maximum.

F.4 Pure GDN and GDN–SWA

We compare GDN^8, $[\mathrm{GDN}^3, \mathrm{SWA}_{16}]^2$, and SWA_4^8 with the $[\mathrm{GDN}^3, \mathrm{FullAttn}]^2$ baseline. The GDN–SWA receivers at L3/L7 are local; SWA-only supplies a bounded-receptive-field control. Table S18 uses the same probes and natural-text target definitions at 30K (seed 42).

Table S18: Architectures with and without global attention (30K, seed 42). Carrying and Matching are peak margin effects; the last two columns report recall PPL. Dashes mark unmeasured entries.

Model	Carrying	Matching	Multiple far	Identifiers
GDN–FullAttn	2.60	4.99	**24.79**	**27.35**
Pure GDN	< 0.02	0.37	66.37	119.23
GDN–SWA (16)	< 0.02	0.36	63.41	114.04
SWA-only (4)	0.00	0.05	—	—

Query-time recurrent retrieval. In GDN-only, transferring the query state changes the retrieved association under both easy and hard input conditions, while source-site transfer has no effect. The association is accessible in the current recurrent state despite weak historical-source effects (Table S18). Arora et al. (2025) likewise find last-state association storage in several state-space models, compared with historical-value storage in Transformers and Based. This contrasts query-state retrieval with the historical-source route in the main hybrid experiments (Section 5).

F.5 GDN–DSA Trained from Scratch

We train GDN–DSA from random initialization for 30K steps (seed 42): eight layers, width 512, eight heads, and layout $[\mathrm{GDN}^3, \mathrm{DSA}]^2$. Receivers L3/L7 select the top 128 candidates across the causal context; GDN occupies L0–2/L4–6. GDN–FullAttn matches this layout and training budget.

Carrying peaks at GDN L2 and Matching at sparse receiver L7 (Table S19), retaining the historical-source division with sparse global access.

Table S19: **Sparse global receivers trained from scratch.** Both models: 30K, seed 42. Carrying and query-update Matching are peak margin effects (200 prompts); PPL uses 10,000 shared sequences.

Model	Carrying	Matching	Overall	Single far	Multiple far	Identifiers
GDN–FullAttn	2.60	4.99	38.21	2.35	24.79	27.35
GDN–DSA	1.70	3.11	31.09	3.08	27.42	24.45