A UNIFYING VIEW OF ATTENTION SINKS: FROM MECHANISMS TO ARCHITECTURAL INTERVENTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

When attention concentrates on a single token, a *sink*, what is the model actually computing? Attention sinks are ubiquitous in softmax transformers, yet this shared visual signature can hide fundamentally different algorithms. We show that visually similar sink patterns can reflect two distinct mechanisms: (***i***) adaptive NOP*, where a head suppresses its update by routing to a null token, and (***ii***) broadcast, where a sink aggregates and redistributes global information. Each mechanism leaves distinct traces (NOP sinks exhibit negligible value norms; broadcast sinks induce low-rank outputs), which we formalize on synthetic tasks and use to derive practical diagnostics. Applied to pretrained vision transformers, these diagnostics reveal that both mechanisms exist at scale: sinks transition from CLS in early layers to patches in deeper layers and concentrate in specialized heads. Causal interventions further connect these signatures to near-null suppression and shared residual contributions. We then use architectural interventions to show how these computations can be reorganized: gating eliminates detected NOP-like sinks but increases broadcast-like sinks, registers relocate rather than remove sink computation, and our position-free global pathway provides an explicit route for shared communication that reduces the broadcast-like sinks induced by gating. On dense probes, combining gating with the global pathway gives the strongest results among the tested variants, despite retaining some broadcast-like sinks. Overall, we find that the same attention pattern can reflect two very different computations, and that effective intervention depends not only on identifying the computation, but also on providing architectural alternatives through which the model can reorganize it.

1 INTRODUCTION

Attention sinks, tokens that attract disproportionate attention across many queries, are pervasive in softmax transformers (Vaswani et al., 2017), including vision transformers (ViTs) (Dosovitskiy et al., 2021; Zhai et al., 2022; Dehghani et al., 2023; Tschannen et al., 2025) and large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Ouyang et al., 2022). They appear as near-vertical "stripes" in attention maps (Barbero et al., 2025; Queipo-de Llano et al., 2026; Cancedda, 2024; Lu et al., 2025), often at the [CLS] token or specific patches (Darcet et al., 2024; Jiang et al., 2025; Wu et al., 2025). Yet the same signature can reflect different computations: what is the model computing, and how should the architecture accommodate it?

Sink-like patterns have been linked to spectral concentration, rank collapse, anisotropy, massive activations, norm imbalance, and softmax saturation (Qiu et al., 2026; Zhang et al., 2025; Cancedda, 2024; Queipo-de Llano et al., 2026; Lu et al., 2025; Sun et al., 2024; Dettmers et al., 2022; Bondarenko et al., 2023; Rybakov et al., 2024; Kang et al., 2025), with consequences for stability and representation quality (Gallego-Feliciano et al., 2025). However, sinks can also serve as useful workspaces that aggregate and redistribute global information (Darcet et al., 2024). Mitigation therefore requires distinguishing harmful effects from useful computation.

Attention patterns alone cannot make this distinction (Akula & Zhu, 2022; Bibal et al., 2022; Wiegreffe & Pinter, 2019). Because softmax must allocate probability mass somewhere, a sink may carry information or provide a safe destination when a head should do nothing (Guo et al.,

*From the NOP instruction in assembly. On many architectures, NOP is encoded as all zeros (unmapped memory returns zeroes by default), so a jump into the void does nothing gracefully.

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.

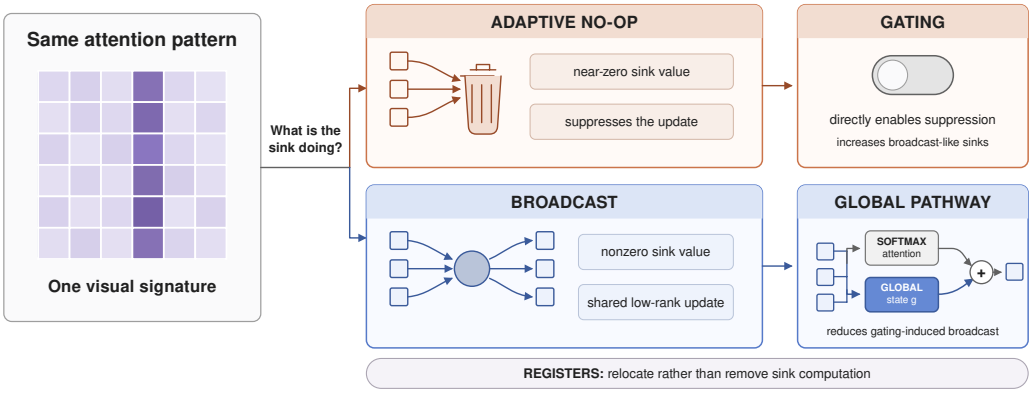


Figure 1: **Same visual signature, different mechanisms.** Attention sinks can implement adaptive NOP, suppressing an update through a near-zero value, or broadcast, producing a shared low-rank update. Gating, registers, and a position-free global pathway provide alternative ways to reorganize these computations.

2024). We study these alternatives as two canonical mechanisms: **adaptive NOP**, where routing to a near-zero value suppresses the residual update, and **broadcast**, where routing to a non-negligible value distributes a shared component across tokens. These mechanisms predict distinct value-norm and output-rank signatures.

We test these interpretations through causal interventions in pretrained ViTs, manipulating sink values and attention routing to distinguish near-null suppression from nonzero shared contributions. These experiments connect the observed sink signatures to their effects on downstream representations.

We then introduce a position-free global pathway to mitigate broadcast sinks, and study its interaction with gating (Qiu et al., 2025), alongside register-based alternatives (Darcet et al., 2024). These experiments suggest that preserving representation quality during sink mitigation depends on how architectural alternatives for suppression and shared computation are combined. Our contributions are:

- We isolate adaptive NOP and broadcast in controlled tasks and derive diagnostics that distinguish their computations beyond attention patterns alone.
- We identify both regimes in pretrained ViTs, test their causal roles, and show that registers retain both.
- We introduce a penalized global pathway and causally validate broadcast relocation in controlled tasks.
- We reveal interactions among architectural interventions, finding that combining gating with the penalized pathway reduces broadcast-like sinks relative to gating alone while achieving higher dense-probe scores.

2 Preliminaries and Related Work

2.1 Softmax Attention and Attention Sinks

We consider a Vision Transformer (ViT) (Dosovitskiy et al., 2021) operating on token representations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$. Unlike autoregressive language models (Qiu et al., 2025; Barbero et al., 2025; Touvron et al., 2023), ViTs use bidirectional attention. For a single head, $\boldsymbol{q}_i = \boldsymbol{W}_Q^\top \boldsymbol{x}_i$, $\boldsymbol{k}_i = \boldsymbol{W}_K^\top \boldsymbol{x}_i$, and $\boldsymbol{v}_i = \boldsymbol{W}_V^\top \boldsymbol{x}_i$, with $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d \times d_h}$, and

$$\boldsymbol{o}_i^{\text{head}} = \sum_{j=1}^{n} \boldsymbol{A}_{ij} \boldsymbol{v}_j, \qquad \boldsymbol{A}_{ij} = \frac{\exp(\boldsymbol{q}_i^\top \boldsymbol{k}_j / \sqrt{d_h})}{\sum_{l=1}^{n} \exp(\boldsymbol{q}_i^\top \boldsymbol{k}_l / \sqrt{d_h})}. \tag{1}$$

Writing the corresponding row-stacked matrices as $\boldsymbol{A}$ and $\boldsymbol{V}$, the head output is $\boldsymbol{O}^{\text{head}} = \boldsymbol{A}\boldsymbol{V}$. Because $\boldsymbol{A}$ is row-stochastic, each query distributes unit attention mass across the sequence. We use the standard residual convention $\boldsymbol{x}_i^{\ell+1} = \boldsymbol{x}_i^{\ell} + \boldsymbol{o}_i^{\ell}$, omitting other sublayers for clarity.

We define an attention sink, consistent with prior work (Ruscio et al., 2025; Gu et al., 2025), as a position that attracts disproportionate attention across many queries without assuming an underlying mechanism.

Definition 1 (ϵ-**Attention Sink**). *Let $\epsilon \in [0, 1)$. A position $s \in \{1, \ldots, n\}$ is an* ϵ-attention sink *with respect to query indices $\mathcal{I} \subseteq \{1, \ldots, n\}$ if $\boldsymbol{A}_{is} \geq 1 - \epsilon$ for all $i \in \mathcal{I}$.*

Thus $\epsilon = 0$ gives a perfect sink. We additionally measure continuous sink strength as $\text{sink}(s; \mathcal{I}) = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} \boldsymbol{A}_{is}$, and use $\max_s \text{sink}(s; \mathcal{I})$ for the strongest sink in a head. This definition follows prior work (Gu et al., 2025; Barbero et al., 2025) and remains agnostic to whether concentrated attention serves suppression or information transfer.

2.2 Related Work

Attention sinks in LLMs. Attention sinks in autoregressive language models often occur at early positions or special tokens even when those positions are not semantically central (Barbero et al., 2025; Gu et al., 2025). Prior work links them to causal masking, long-context behavior, and architectural biases (Barbero et al., 2025; Qiu et al., 2025; Wu et al., 2024), as well as spectral concentration, residual imbalance, outlier activations, and conditionally inactive heads (Queipo-de Llano et al., 2026; Cancedda, 2024; Sun et al., 2024; Dettmers et al., 2022; Bondarenko et al., 2023; Sandoval-Segura et al., 2026). These observations have motivated gated and modified attention mechanisms that reduce concentration or allow heads to suppress their outputs directly (Bondarenko et al., 2023; Qiu et al., 2025).

Attention sinks in ViTs. In ViTs, sinks appear at `[CLS]`, particular patches, and learned workspace tokens such as registers (Lu et al., 2025; Darcet et al., 2024; Jiang et al., 2025; Kang et al., 2025). They have been interpreted both as artifacts associated with outliers or softmax saturation (Lu et al., 2025; Bondarenko et al., 2023) and as useful mechanisms for aggregating and redistributing global information (Darcet et al., 2024). Registers provide dedicated positions that absorb attention mass and can improve representation quality (Darcet et al., 2024; Jiang et al., 2025), but remain sequence positions and can themselves become sinks. Related work on rank collapse, oversmoothing, position bias, and ViT dynamics further connects sink behavior to token mixing and representational diversity (Jacobs et al., 2026; Wang et al., 2022). What remains missing is a practical way to distinguish the computations underlying visually similar sinks and architectural alternatives for realizing suppression and shared communication; we address both.

3 Two Hypotheses for Attention Sinks

The signature of an attention sink, a vertical stripe in the attention map, is computationally ambiguous. Does the model attend to a token because it carries useful information, or because it provides a safe destination when no update is needed? Our goal is to derive stylized mechanistic intuition and operational diagnostics that distinguish these possibilities in pretrained ViTs and motivate architectural alternatives for the computations they implement. We formalize two regimes below. Throughout, $\boldsymbol{v}_s$ denotes the single-head value in head-space analyses and the output-projected value when discussing residual-stream updates.

Hypothesis 1 (**Adaptive `NOP`**). *An attention head implements an adaptive* `NOP` *mechanism (Sandoval-Segura et al., 2026; Barbero et al., 2025) when it selectively routes queries to a sink position s whose contribution is negligible. For tokens i attending predominantly to s, $\boldsymbol{o}_i \approx \boldsymbol{0}$, so the head approximately preserves the residual stream.*

Hypothesis 2 (**Broadcast**). *An attention head implements a broadcast mechanism (Darcet et al., 2024) when attention from many positions concentrates on a sink position s whose value vector $\boldsymbol{v}_s$ induces a shared residual contribution. For tokens attending predominantly to s, the residual update satisfies $\boldsymbol{x}_i^{\ell+1} - \boldsymbol{x}_i^{\ell} \approx \boldsymbol{A}_{is}\boldsymbol{v}_s$. The recipient representations therefore acquire a shared component aligned with $\boldsymbol{v}_s$.*

Both mechanisms produce concentrated attention, but they assign different computational roles to the sink: a null destination for conditional suppression or a source of a shared update. Distinguishing them requires examining the values and resulting updates, rather than attention weights alone.

3.1 Hypothesis 1: Sink as Adaptive NOP

Attention contributes additively to the residual stream through $\boldsymbol{x}_i^{\ell+1} = \boldsymbol{x}_i^{\ell} + \boldsymbol{o}_i^{\ell}$. When a head has no useful information to contribute, softmax still requires it to allocate probability mass (Barbero et al., 2025). A sink with a near-zero value provides a way to satisfy this constraint while producing a negligible update. Crucially, this behavior is adaptive: the same head can perform a meaningful operation for some queries and suppress its contribution for others (Sandoval-Segura et al., 2026).

NOP Signatures. A negligible output can in principle arise through cancellation among several values or by attending to a null value. Once attention concentrates on a single position, however, cancellation by the remaining tokens is limited. The following result makes precise why a strongly concentrated NOP sink must have a small value.

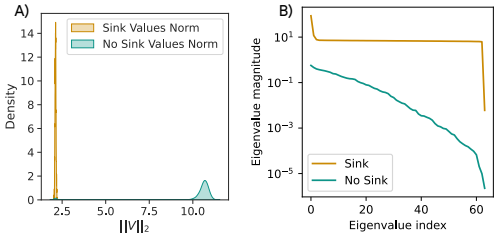


Figure 2: **NOP sink signatures.** **(A)** Sink solutions learn near-zero sink value norms, producing negligible updates. **(B)** Sink models exhibit a dominant singular value in $\boldsymbol{W}_Q\boldsymbol{W}_K^\top$, consistent with a learned gating direction.

Lemma 1 (Uniqueness of the Exact No-Op Solution). *Consider a single attention head with output $\boldsymbol{o}_i = \sum_j \boldsymbol{A}_{ij}\boldsymbol{v}_j$, where values are measured in the same space as the output. Let s be an ϵ-sink for a nonempty query set I, with $0 \le \epsilon < 1$, and suppose $\|\boldsymbol{o}_i\| \le \delta$ for all $i \in I$. Writing $V_{\max} = \max_j \|\boldsymbol{v}_j\|$, we have*

$$\|\boldsymbol{v}_s\| \le \frac{\delta + \epsilon V_{\max}}{1-\epsilon}. \tag{2}$$

If $V_{\max}$ remains uniformly bounded, then $\epsilon, \delta \to 0$ imply $\boldsymbol{v}_s \to \boldsymbol{0}$. For a perfect sink ($\epsilon = 0$), the no-op constraint is equivalent to $\|\boldsymbol{v}_s\| \le \delta$; hence exact no-op ($\delta = 0$) has the unique sink value $\boldsymbol{v}_s = \boldsymbol{0}$.

The null value explains how a head suppresses its update; conditional routing explains when it does so. For a query requiring a NOP, the sink logit must exceed competing logits sufficiently for $\boldsymbol{A}_{is}$ to approach one. In a linear query–key interaction, this can be achieved through a dominant singular direction in $\boldsymbol{W}_Q\boldsymbol{W}_K^\top$ aligned with a triggering feature, or through large sink activations aligned with that direction. Lemma 3 in Appendix B.2 gives constructive examples of both regimes. This connects adaptive NOP to spectral spikes and to the massive activations observed in outlier tokens (Darcet et al., 2024), although neither signature alone establishes that a head performs a NOP.

Toy Model of Adaptive NOP. To isolate conditional suppression, we train a single-layer attention head on a conditional damping task. Inputs are random tokens $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, and a fixed gate vector $\boldsymbol{v}^\star$ determines the target head output. If $\langle \boldsymbol{x}_i, \boldsymbol{v}^\star \rangle > 0$, the target is $\boldsymbol{y}_i = \gamma \boldsymbol{x}_i$, with $\gamma = 0$ in the default experiment; otherwise, $\boldsymbol{y}_i = \boldsymbol{x}_i$. The head must therefore produce a null output for flagged tokens while reconstructing the input for the remaining tokens. A permanently inactive head cannot solve the task. Full configurations and the related causal-validation setup appear in Appendix A.1.

Results and Signatures. We observe two complementary signatures. **(i) A learned null value.** Across the tested runs, optimization converges to a sink-based solution rather than a dispersed cancellation pattern. The model allocates a token s whose value norm approaches zero (Figure 2A), providing a shared null destination for flagged queries. **(ii) A spectral signature of conditional routing.** Models with sinks exhibit a dominant singular value in $\boldsymbol{W}_Q\boldsymbol{W}_K^\top$ (Figure 2B), consistent with the gating construction in Lemma 3. The learned direction amplifies the feature that determines when queries should attend to the null token.

Together, these observations separate the two ingredients of adaptive NOP: a value that permits a negligible update and a routing mechanism that selects it conditionally. Appendix A.5 examines

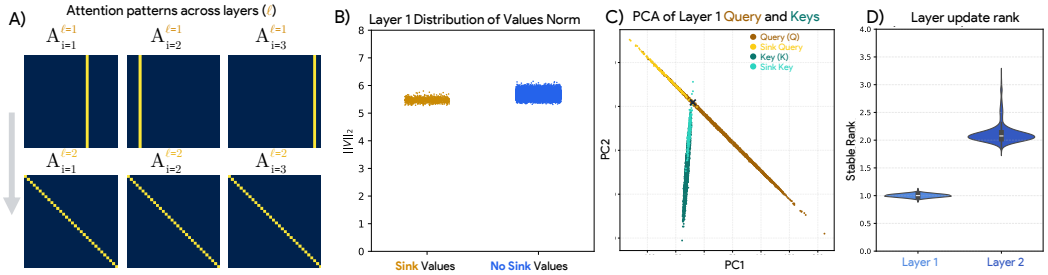


Figure 3: **Broadcast sink signatures.** A two-layer model trained on global broadcast learns a modular solution: **(A)** Layer 1 forms a broadcast hub while Layer 2 remains identity-like; **(B)** sink values retain content-scale norms; **(C)** query-key geometry selects the source token; and **(D)** the broadcast update is approximately rank-1.

how sink formation changes with the frequency and strength of the NOP condition, including partially damped targets.

3.2 Hypothesis 2: Sink for Broadcasting

Under Hypothesis 2, a sink produces a shared non-negligible update across recipients. In the controlled setting below, we study the stronger case in which that shared update carries input-specific information, so the sink acts as a communication hub.

Broadcast Signatures. This mechanism predicts both a non-negligible sink value and a shared structure in the attention output. When one source dominates attention, recipient updates differ primarily in the scalar weight assigned to that source.

Lemma 2 (Rank-1 Update). *Suppose a single sink position s dominates attention for all queries represented in $\boldsymbol{O} = \boldsymbol{AV}$, with $\boldsymbol{A}_{is} \approx 1$, uniformly bounded values, and a non-negligible sink value $\boldsymbol{v}_s$. Then the output matrix exhibits approximate rank-1 structure: $\boldsymbol{O} \approx \boldsymbol{A}_s \boldsymbol{v}_s^\top$, where $\boldsymbol{A}_s$ is the sink column of $\boldsymbol{A}$.*

The low-rank structure concerns the attention-induced update, not necessarily the token representations after the residual addition. In particular, a perfectly shared update adds the same vector to every token and preserves centered variance exactly. Broadcast therefore need not increase token alignment or collapse representations toward their mean. Lemma 4 in Appendix B formalizes this distinction for approximate sinks.

Toy Model of Broadcasting. We test whether an objective requiring shared information induces this mechanism. A two-layer Transformer receives random tokens $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ and must add a rotated version of a selected source token to every position:

$$\boldsymbol{y}_i = \boldsymbol{x}_i + \gamma \boldsymbol{R} \boldsymbol{x}_{j^\star}, \tag{3}$$

where $\boldsymbol{R} \in \mathbb{R}^{d \times d}$ is a fixed orthogonal rotation and γ controls broadcast strength. Solving the task requires identifying the source $j^\star$ and distributing its payload while retaining each recipient's input. Source selection and training details appear in Appendix A.1.

Results and Signatures. The learned solution exhibits three principal properties. **(i) Specialized broadcast routing.** Although a two-layer model could distribute communication across depth, the illustrated solution concentrates it in one layer. Layer 1 attends predominantly to the source $j^\star$, while Layer 2 adopts an identity-like attention pattern (Figure 3A). **(ii) Content-scale sink values.** Unlike the null values learned on the NOP task, broadcast sink values have norms comparable to non-sink content values, with both distributions centered around $\|\boldsymbol{v}\| \approx 5.5$ (Figure 3B). Concentrated routing therefore delivers a substantial update rather than suppressing one. **(iii) Low-rank updates.** The attention update in the broadcast layer has stable rank (Neyshabur et al., 2015; Arora et al., 2018) approximately one, whereas the second layer produces higher-rank updates (Figure 3D). This agrees with Lemma 2: recipients receive a shared vector through the broadcast layer.

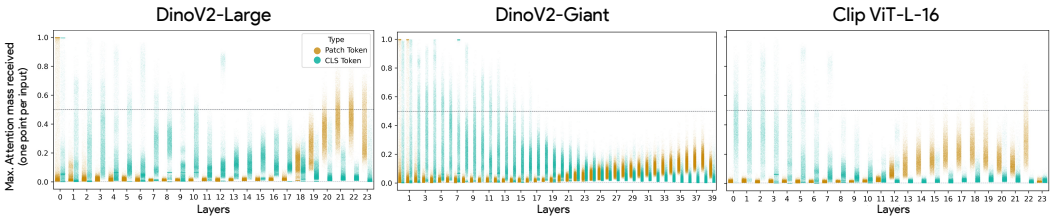


Figure 4: **Sink token transition across layers.** DINOv2 (Large and Giant) and OpenCLIP-L-16 exhibit a handoff pattern: [CLS] serves as the sink in early layers but yields to patch tokens in later layers.

The query–key geometry provides a complementary view of source selection. Recipient queries align along a common direction, and the selected source key aligns strongly with that direction (Figure 3C). Appendix A.5 presents this geometric analysis in detail. These observations characterize the learned solution; the causal interventions in Appendix A.4 further test whether the selected value carries the broadcast payload.

The two tasks thus motivate a joint diagnostic: negligible sink values indicate NOP-like suppression, while content-scale values together with approximately rank-one updates indicate a broadcast-like shared update. We apply these diagnostics to pretrained ViTs in the next section.

4 Empirical Evidence in ViTs and Phenomenology

Having established the theoretical signatures of the adaptive NOP and broadcast mechanisms, we now examine their prevalence in large-scale Vision Transformers.

Phase Transition of Sink Tokens. We investigate which tokens serve as sinks across layers and whether this varies by architecture. Figure 4 shows the attention inflow, defined by the token receiving the highest attention mass per input, for DINOv2-L, DINOv2-G (Oquab et al., 2024), and OpenCLIP-L-16 (Ilharco et al., 2021). All three models exhibit a similar transition. In early layers, the [CLS] token acts as the primary sink. In deeper layers, this role shifts to patch tokens: [CLS] inflow decreases while patch sinks emerge. We now ask whether sink behavior is distributed across all attention heads or concentrated in specialized ones.

Head Specialization. We map the frequency of sink behavior across layers and heads in DINOv2-L. Figure 5 reveals a sparse, vertical structure: certain heads act as sinks for nearly 80% of inputs, while adjacent heads never do. This indicates that sink mechanisms are head specific. Given this specialization, we return to our main question: do these dedicated sink heads implement NOP, broadcast, or both?

Distinct Functional Regimes. To answer this, we project every detected sink in DINOv2-G onto two diagnostic axes: the value norm ratio $\|\boldsymbol{v}_s\|/\|\boldsymbol{v}_i\|$ and the stable rank of the attention update. Our main empirical finding is not the existence of either regime in isolation, but that these regimes can be separated and shown to coexist within the same pretrained ViT. Figure 7 reveals two distinct modes.

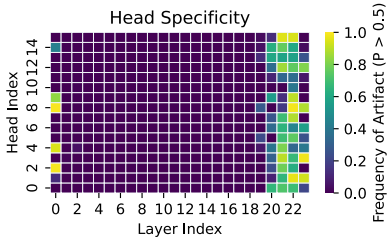


Figure 5: **Head specialization in DINOv2-L.** Sink behavior is concentrated in a sparse subset of attention heads across layers.

The first cluster, concentrated in early-to-middle layers, exhibits strongly suppressed value norms ($< 0.2\times$ the baseline norm), consistent with adaptive NOP: these heads nullify their contribution to the residual stream. The second cluster, concentrated in deeper layers, shows standard value norms paired with approximately rank-1 updates, consistent with broadcast: these heads produce non-negligible shared residual contributions across the sequence. This dual phenomenology also appears in other ViTs (Appendix D).

Validation via Registers. Register-equipped DINOv2-G provides an additional test of this distinction. Registers absorb most sink mass from patch and [CLS] tokens, but the resulting register sinks retain the same two regimes: most have suppressed value norms consistent with NOP, while a smaller group has content-scale values and approximately rank-1 updates consistent with broadcast (Appendix D). Thus, registers relocate rather than eliminate sink computation: they are repurposed for NOP and, because they remain sequence positions, can themselves become broadcast sinks.

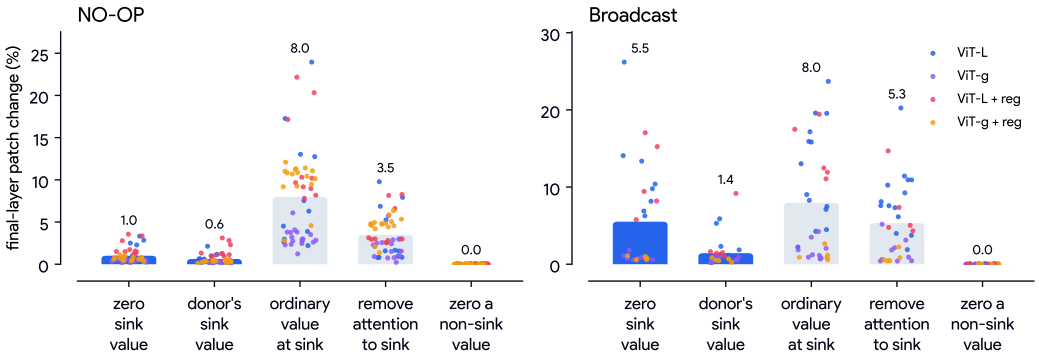


Figure 6: **Causal validation of sink function in pretrained ViTs.** For detected NOP-like and broadcast-like sinks, we intervene on the sink value or its incoming attention and measure the resulting change in final-layer patch representations. NOP-like sinks are insensitive to zeroing or replacing the sink value with other sink values, but become disruptive when assigned an ordinary value or when attention to the sink is removed. Broadcast-like sinks are sensitive to removing their contribution, while replacement by another sink value has a substantially smaller effect than replacement by an ordinary value. Zeroing a non-sink value provides a control.

Causal Validation in Pretrained ViTs. We next test whether these observational signatures correspond to distinct causal roles (Figure 6). For NOP-like sinks, zeroing the sink value, replacing it with the head's mean sink value, or substituting a sink value from another image produces little change in final-layer patch representations. In contrast, replacing the sink with an ordinary value or removing attention to it is substantially more disruptive. This supports the interpretation of the sink as a near-null destination that suppresses the head's residual update. Broadcast-like sinks show the complementary pattern: zeroing their value or removing attention to them substantially alters downstream representations, confirming that they contribute a non-negligible shared update. However, replacing the sink value with the head's mean or with a donor image's sink value produces much smaller changes than inserting an ordinary value, suggesting that the shared contribution is largely image-invariant in the tested heads. These interventions therefore causally distinguish near-null suppression from nonzero shared residual contributions while qualifying the interpretation of broadcast as necessarily carrying image-specific information.

These results motivate studying architectural interventions for the two computations. Gating provides a direct architectural implementation of conditional suppression for adaptive NOP, while registers relocate sink computation without removing the single-position routing pattern. In the next section, we introduce an explicit position-free global pathway for shared communication and study how these interventions reorganize sink behavior when combined.

5 Architectural Interventions for Sink Computation

Our analysis motivates complementary architectural interventions for the two sink mechanisms. Gating gives adaptive-NOP heads a direct mechanism for suppressing their residual updates, while an explicit global pathway provides a position-free route for shared communication. Registers provide dedicated sequence positions that can absorb and relocate sink computation (results with registers are reported in Tables 8 and 9 of Appendix C).

An Explicit Global Pathway. Given pre-normalized token representations $\widetilde{\boldsymbol{X}}^{\ell} \in \mathbb{R}^{n\times d}$, we construct a permutation-invariant global state and return it to each token through a learned gate:

$$\pi_j^{\ell} = \frac{\exp\big((\boldsymbol{w}_P^{\ell})^{\top}\widetilde{\boldsymbol{x}}_j^{\ell}\big)}{\sum_{k=1}^{n}\exp\big((\boldsymbol{w}_P^{\ell})^{\top}\widetilde{\boldsymbol{x}}_k^{\ell}\big)}, \qquad \boldsymbol{g}^{\ell} = \sum_{j=1}^{n}\pi_j^{\ell}\boldsymbol{W}_{V,G}^{\ell}\widetilde{\boldsymbol{x}}_j^{\ell},$$
$$\gamma_i^{\ell} = \sigma\big((\widetilde{\boldsymbol{x}}_i^{\ell})^{\top}\boldsymbol{w}_{\gamma}^{\ell} + b_{\gamma}^{\ell}\big), \qquad \boldsymbol{x}_i^{\ell+1} = \boldsymbol{x}_i^{\ell} + \boldsymbol{o}_i^{\ell,\mathrm{attn}} + \gamma_i^{\ell}\boldsymbol{W}_{O,G}^{\ell}\boldsymbol{g}^{\ell}. \tag{4}$$

Unlike a broadcast sink, $\boldsymbol{g}^{\ell}$ is not associated with a sequence position and therefore cannot attract an attention column. The pathway nevertheless preserves the characteristic shared update: stacking γ_i^{ℓ} across tokens produces an update matrix of rank at most one.

Encouraging Broadcast Relocation. The global pathway provides an explicit route for shared communication, while the task objective alone permits either branch to carry the shared update. We therefore encourage relocation from attention to the global pathway by penalizing the common component of the attention update. Let $\boldsymbol{O}_b^{\ell,\mathrm{attn}}$ denote the output-projected attention update for example b, measured before any output gate, and define its common component by $\bar{\boldsymbol{o}}_b^{\ell,\mathrm{attn}} = n^{-1}\sum_i \boldsymbol{o}_{bi}^{\ell,\mathrm{attn}}$. We train with

$$\mathcal{L} = \mathcal{L}_{\mathrm{task}} + \lambda_B \frac{1}{|\mathcal{L}_B|}\sum_{\ell\in\mathcal{L}_B}\frac{\mathcal{E}\big(\bar{\boldsymbol{O}}^{\ell,\mathrm{attn}}\big)}{\mathcal{E}(\boldsymbol{O}^{\ell,\mathrm{attn}}) + \epsilon}, \qquad \mathcal{E}(\boldsymbol{Z}) = |\boldsymbol{Z}|^{-1}\|\boldsymbol{Z}\|_F^2, \tag{5}$$

where $\bar{\boldsymbol{O}}^{\ell,\mathrm{attn}}$ stacks the batchwise common components and $\mathcal{L}_B$ denotes the penalized layers. Up to the stabilizer ϵ, the ratio measures the fraction of the attention update's energy shared across tokens and is invariant to rescaling the attention branch. Penalizing all attention layers encourages shared computation to use the global pathway rather than move to another attention block. The penalty acts on the shared component carried by attention rather than on attention concentration itself.

Experiments. We first evaluate the intervention on the two-layer broadcast task from Section 3.2, comparing attention alone, the global pathway without the penalty, and the penalized global pathway using matched training batches across five seeds. We measure task error, maximum attention inflow, and causal dependence on each branch through attention ablation, global-state ablation, and cross-example global-state replacement. Appendix E.1 additionally reports a controlled evaluation with uniform recipient weights, $\gamma_i^{\ell} = 1$, to isolate shared communication.

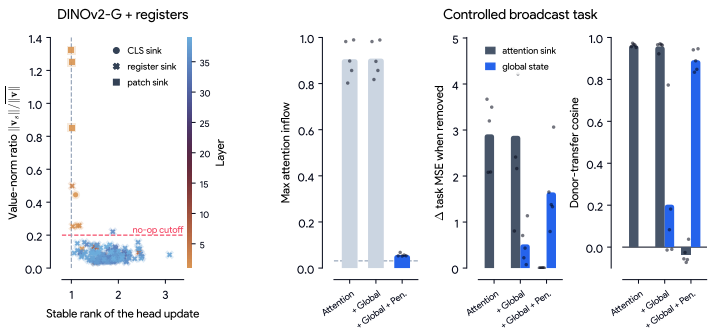


Figure 7: **Diagnostics and architectural reorganization. (A)** DINOv2-G + registers sinks separate into NOP-like low-value sinks and broadcast-like rank-1 sinks. **(B)** On the controlled task, adding the global pathway alone leaves broadcast in attention, whereas the common-mode penalty relocates the shared computation to the global branch.

We then train ViT-L LeJEPA (Balestriero & LeCun, 2025) for 50 epochs with four variants: a vanilla baseline, Gated Attention ('G1' from Qiu et al. (2025)), the penalized global pathway, and their combination. The broadcast penalty uses $\lambda_B = 1$ on both controlled-task layers and $\lambda_B = 0.1$ on ViT layers 1–23, leaving block 0 unpenalized. We classify detected sinks using the diagnostics from Section 4, with the counting rule of Appendix D. ImageNet-1k classification and frozen-backbone ADE20K and Pascal VOC 2012 probes provide complementary representation-quality controls. Full configurations, probe settings and register comparisons appear in Appendix C.

Table 1: **Linear probing performance.** LeJEPA ViT-L models trained for 50 epochs. ImageNet: top-1 of an online linear probe. ADE20K and VOC12: mIoU of linear segmentation probes on frozen features.

Variant	ImageNet	ADE20K	VOC12
Baseline	0.602	0.212 ± 0.001	0.601 ± 0.0005
Gating	**0.610**	0.216 ± 0.001	0.595 ± 0.0004
Global + Penalty	0.549	0.181 ± 0.001	0.527 ± 0.0004
Gating + Global + Penalty	0.606	$\mathbf{0.217 \pm 0.0004}$	$\mathbf{0.610 \pm 0.0001}$

Controlled-Task Results. On the controlled task, adding the pathway alone leaves attention free to carry the broadcast signal: second-layer maximum inflow remains 0.91, matching the attention baseline. Adding the penalty reduces inflow to 0.06 in both layers while preserving similar task MSE (0.144 versus 0.140). Attention ablation then has little effect, whereas global-state ablation removes the shared update; donor replacement yields an output change aligned with the desired donor-minus-recipient payload (cosine similarity 0.89). The uniform-recipient evaluation in Appendix E.1 independently confirms that causal dependence and source selection move from attention to the global branch. Thus, the pathway can carry the broadcast content without requiring any sequence position to attract attention from all recipients.

Architectural Interactions in ViT-L. In ViT-L, the interventions reorganize the two sink regimes differently. Gating reduces the frequency of detected NOP sinks from 0.25 to 0.00 per image, while the penalized global pathway reduces broadcast sinks from 0.09 to 0.04 per image. Gating also increases broadcast-sink frequency to 3.80 per image. Adding the penalized global pathway reduces this to 0.53, an approximately 86% reduction relative to gating alone, while keeping NOP sinks near zero (0.03 per image; Table 14). Thus, making suppression explicit shifts shared computation toward broadcast-like sinks, while the global pathway redirects much of this shift without eliminating broadcast-like sinks altogether.

Table 1 separates sink prevalence from representation quality. The penalized global pathway alone suppresses broadcast-like sinks most strongly but performs worse than the baseline on all three probes. In contrast, combining gating with the global pathway retains some broadcast-like sinks while achieving near-baseline ImageNet accuracy and the strongest dense-probe results among the tested variants. Together with the controlled interventions, these results show that architectural changes can relocate and reorganize sink computation without making sink prevalence itself a proxy for representation quality. Effective intervention should therefore be evaluated by both how computation is redistributed and the representations that result.

6 Discussion

Our results show that attention sinks in ViTs should not be treated as a single phenomenon. Value norms and update rank distinguish two canonical regimes, while causal interventions connect them to near-null suppression and nonzero shared residual contributions. Registers further show that sink computation can be relocated rather than eliminated. The mitigation experiments extend this picture: architectural changes reorganize these computations rather than acting as independent fixes. Gating removes detected NOP-like sinks but induces substantially more broadcast-like sinks, while adding the global pathway reduces this shift. Thus, changing the mechanisms available for suppression can alter how shared communication is implemented elsewhere in the model.

These interactions also change how sink mitigation should be evaluated. Sink prevalence alone is not a sufficient measure of representation quality: the global-pathway variant suppresses broadcast-like sinks most strongly but performs worse on downstream probes, whereas combining gating with the global pathway gives the strongest dense-probe results among the tested variants despite retaining some broadcast-like sinks. The goal is therefore not simply to eliminate sinks, but to provide architectural alternatives for the computations they support. Our framework is not exhaustive: real models may distribute broadcast computation across layers or use higher-rank shared signals, potentially requiring richer global pathways, and the appropriate pathway design may vary with scale and objective. More broadly, effective intervention depends not only on identifying what a sink is computing, but also on giving the model another way to compute it.

Reproducibility Statement

We make a preliminary version of our codebase publicly available, including code to run the controlled toy tasks and ViT experiments: https://anonymous.4open.science/r/attention-sinks-code-5940/README.md. Additional details on task setups, evaluation protocols, and training configurations are provided in the appendix.

References

Arjun R Akula and Song-Chun Zhu. Attention cannot be an explanation. *ArXiv e-print*, 2022.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Federico Barbero, Álvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Petar Veličković, Razvan Pascanu, and Michael M Bronstein. Why do LLMs attend to the first token? *Proceedings of the Conference on Language Modeling (COLM)*, 2025.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 864–873, 2020.

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. Is attention explanation? an introduction to the debate. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3889–3900, 2022.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mikhail S Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V Sapunov. Memory transformer. *arXiv preprint arXiv:2006.11527*, 2020.

Nicola Cancedda. Spectral filters, dark signals, and attention sinks. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4792–4808, 2024.

Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. GCNet: Non-local networks meet squeeze-excitation networks and beyond. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does BERT look at? an analysis of BERT's attention. *Proceedings of the ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, 2019.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:30318–30332, 2022.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 2793–2803, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Jorge Gallego-Feliciano, S Aaron McClendon, Juan Morinelli, Stavros Zervoudakis, and Antonios Saravanos. Hidden dynamics of massive activations in transformer training. *ArXiv e-print*, 2025.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 57026–57037, 2023.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Tianyu Guo, Druv Pai, Yu Bai, Jiantao Jiao, Michael I Jordan, and Song Mei. Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in LLMs. *arXiv preprint arXiv:2410.13835*, 2024.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. Understanding and minimising outlier features in transformer training. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:83786–83846, 2024.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141, 2018.

Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, et al. OpenCLIP. *Zenodo*, 2021. doi: 10.5281/zenodo.5143773.

Mozes Jacobs, Thomas Fel, Richard Hakim, Alessandra Brondetta, Demba Ba, and T Andy Keller. Block recurrent dynamics in vision transformers. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Andrew Jaegle, Felix Gimeno, Andy Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 4651–4664, 2021.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 3543–3556, 2019.

Nick Jiang, Amil Dravid, Alexei Efros, and Yossi Gandelsman. Vision transformers don't need trained registers. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, 2020.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 3744–3753, 2019.

Andrew Lu, Wentinn Liao, Liuhui Wang, Huzheng Yang, and Jianbo Shi. Artifacts and attention sinks: Structured approximations for efficient vision transformers. *ArXiv e-print*, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Thiziri Nait Saada, Alireza Naderi, and Jared Tanner. Mind the gap: a spectral analysis of rank collapse and signal propagation in attention layers. *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 45561–45587, 2025.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. *Proceedings of the Conference on Learning Theory (COLT)*, 2015.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *The Journal of Transactions on Machine Learning Research (TMLR)*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Zihan Qiu, Zeyu Huang, Kaiyue Wen, Peng Jin, Bo Zheng, Yuxin Zhou, Haofeng Huang, Zekun Wang, Xiao Li, Huaqing Zhang, et al. A unified view of attention and residual sinks: Outlier-driven rescaling is essential for large language model training. *Proceedings of the Conference on Language Modeling (COLM)*, 2026.

Enrique Queipo-de Llano, Álvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. Attention sinks and compression valleys in LLMs are two sides of the same coin. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, et al. Theory, analysis, and best practices for sigmoid self-attention. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Valeria Ruscio, Umberto Nanni, and Fabrizio Silvestri. What are you sinking? a geometric approach on attention sink. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Oleg Rybakov, Mike Chrzanowski, Peter Dykas, Jinze Xue, and Ben Lanir. Methods of improving LLM training stability. *arXiv preprint arXiv:2410.16682*, 2024.

Pedro Sandoval-Segura, Xijun Wang, Ashwinee Panda, Micah Goldblum, Ronen Basri, Tom Goldstein, and David Jacobs. Identifying and evaluating inactive heads in pretrained LLMs. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Hemanth Saratchandran, Jianqiao Zheng, Yiping Ji, Wenbo Zhang, and Simon Lucey. Rethinking attention: Polynomial alternatives to softmax in transformers. *arXiv preprint arXiv:2410.18613*, 2024.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Hervé Jégou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *Proceedings of the Conference on Language Modeling (COLM)*, 2024.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 32–42, 2021.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *ArXiv e-print*, 2023.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *ArXiv e-print*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5797–5808, 2019.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7794–7803, 2018.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Mitchell Wortsman, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Replacing softmax with ReLU in vision transformers. *arXiv preprint arXiv:2309.08586*, 2023.

Xinyi Wu, Amir Ajorlou, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the role of attention masks and layernorm in transformers. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:14774–14809, 2024.

Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the emergence of position bias in transformers. *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Josh Susskind. Stabilizing transformer training by preventing attention entropy collapse. *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Stephen Zhang, Mustafa Khan, and Vardan Papyan. Attention sinks: A 'catch, tag, release' mechanism for embeddings. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

APPENDIX CONTENTS

A Controlled tasks and causal interventions **16**
A.1 Controlled Task Setups 16
A.2 Causal Interventions and Evaluation Metrics 17
A.3 Adaptive NOP: Results and Causal Validation 19
A.4 Broadcast: Results and Causal Validation 20
A.5 Supplementary Controlled-Task Analyses 22

B Formal statements and proofs **24**
B.1 Proof of Lemma 1: Uniqueness of the Exact No-Op Solution 24
B.2 Mechanisms for Adaptivity: Statement and Proof 24
B.3 Proof of Lemma 2: Rank-1 Update 25
B.4 Shared Additive Component: Statement and Proof 26

C LeJEPA Mitigation Experiments: Training Details **28**
C.1 Common recipe and the register and gating sweep 28
C.2 The models of Table 1 28
C.3 Linear probes 29

D Additional Experimental Results and Figures **31**

E Supplementary mitigation results and ablations **32**
E.1 Controlled-Task Causal Validation of the Global Pathway 32
E.2 LeJEPA: Penalty Normalization, LayerScale, and Sensitivity to the Sink Rule . . . 36

F Extended Literature Review **36**
F.1 Gated Attention 36
F.2 Register Tokens and Global Communication 37
F.3 Non-softmax Attention 38
F.4 Other Related Works 39

A CONTROLLED TASKS AND CAUSAL INTERVENTIONS

A.1 CONTROLLED TASK SETUPS

We use two deliberately minimal synthetic tasks to isolate the mechanisms described in Section 3: adaptive NOP, which requires the attention branch to conditionally suppress its update, and broadcast, which requires content-dependent information to be communicated to all positions. In both settings, inputs are sampled i.i.d. from an isotropic Gaussian distribution and fresh samples are generated at every optimization step.

Adaptive NOP. We implement the NOP task using a minimal single-head Transformer block with no MLP, mirroring a ViT attention layer with a pre-norm residual connection. Let $X \in \mathbb{R}^{L\times d}$ denote the residual stream and $\widetilde{X} = \mathrm{LN}(X)$ the normalized states used to form queries, keys, and values. The block computes

$$\boldsymbol{A} = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right), \qquad \Delta O = W_O(\boldsymbol{A}V), \qquad O = X + \Delta O, \tag{6}$$

where Q, K, and V are obtained from $\widetilde{X}$ using learned linear maps without bias. We use model width $d = 64$, head dimension $d_h = 64$, and sequence length $L = 32$, with learned positional embeddings and a learned BOS offset added to token 0 before LayerNorm. Training examples $X_{\mathrm{in}} \in \mathbb{R}^{B\times L\times d}$ are sampled i.i.d. from a standard Gaussian distribution.

To determine which tokens require a NOP, we sample once a fixed unit vector $v \in \mathbb{R}^d$ and compute $s_i = \langle (X_{\mathrm{in}})_i, v\rangle$. We define $m_i = \mathbb{1}[s_i > \lambda]$ and impose the regression target directly on the attention branch,

$$\Delta y_i = (1 - m_i)(X_{\mathrm{in}})_i + m_i\gamma(X_{\mathrm{in}})_i. \tag{7}$$

Thus, $\gamma = 0$ requires a zero attention update for masked tokens, leaving the residual stream unchanged, while $\gamma > 0$ yields a partially damped condition. We minimize mean-squared error $\|\Delta O - \Delta Y\|_2^2$ using AdamW with learning rate $3 \cdot 10^{-3}$ and weight decay 10^{-4}. Unless stated otherwise, models are trained for 30k steps with batch size $B = 1024$; the optimization sweeps in Appendix A.5 use $B = 2048$.

For the causal-validation experiment in Appendix A.3, we use a closely related block-output formulation that makes the NOP interpretation explicit. Some query tokens have identity targets, $y_i = x_i$, and therefore require the attention branch to contribute an approximately zero residual update, whereas the remaining queries have zero-output targets and require an active update. Token 0 is provided as a dedicated, input-independent token and is excluded from the task loss, allowing the model to use it as an attention destination without requiring it to encode token-specific task information. The resulting trained model is analyzed using the causal interventions defined in Appendix A.2.

Broadcast. We implement the broadcast task using a two-layer ViT-style Transformer with one attention head per layer and pre-norm residual connections. Each attention layer applies LayerNorm, computes Q, K, and V using learned linear maps without bias, forms

$$\boldsymbol{A} = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right), \tag{8}$$

and writes the attention output back through a learned output projection, $X_{\mathrm{out}} = X_{\mathrm{in}} + W_O(\boldsymbol{A}V)$. Each attention layer is followed by an MLP block $\mathrm{LN} \to \mathrm{Linear}(d, 4d) \to \mathrm{GELU} \to \mathrm{Linear}(4d, d)$ with a residual connection. We use $d = d_h = 64$, sequence length $L = 32$, and learned positional embeddings added only before the first attention layer.

For each sequence, we sample a fixed selector direction u once and choose the content-dependent source position

$$j^\star = \arg\max_j \langle x_j, u\rangle. \tag{9}$$

Every recipient token must then add a transformed version of the selected source,

$$y_i = x_i + \alpha\big(x_{j^\star}R\big), \tag{10}$$

where $R \in \mathbb{R}^{d\times d}$ is sampled once and fixed throughout training. We construct R as an approximately orthogonal matrix using QR factorization and typically set the broadcast strength to $\alpha = 1$. This objective requires the model to identify a source whose position varies across examples and communicate its content to every token in the sequence.

We train on i.i.d. Gaussian sequences $X \in \mathbb{R}^{B\times L\times d}$ with batch size $B = 1024$, minimizing mean-squared error between the model outputs and the broadcast targets. Optimization uses AdamW for 30k steps with learning rate $5\cdot 10^{-4}$ and weight decay 10^{-4}. The diagnostics and causal interventions used to determine whether the learned attention sinks implement the intended communication mechanism are defined in Appendix A.2.

A.2 Causal Interventions and Evaluation Metrics

We use both observational diagnostics and targeted interventions to determine whether a learned sink implements adaptive NOP or broadcast. The key distinction in the causal analysis is between interventions on the *payload* carried by a high-attention destination, which preserve the learned attention pattern, and interventions on the *routing* itself, which alter where attention is directed.

Sink diagnostics. For an attention tensor $\boldsymbol{A} \in \mathbb{R}^{B\times L\times L}$, we measure the average attention received by key position j as

$$\bar{a}_j = \frac{1}{BL}\sum_{b=1}^{B}\sum_{i=1}^{L} \boldsymbol{A}_{bij}, \tag{11}$$

and define the sink score as $\max_j \bar{a}_j$. This measure is appropriate when the sink occupies a fixed position, as in the adaptive-NOP task.

For broadcast, the source position is content-dependent and therefore varies across examples. For each example b, we identify the dominant attention column

$$\hat{j}_b = \arg\max_j \frac{1}{L}\sum_{i=1}^{L} \boldsymbol{A}_{bij}, \tag{12}$$

and report the fraction of examples for which $\hat{j}_b = j_b^\star$, where $j_b^\star$ is the ground-truth source defined in Eq. 9. We additionally report the mean attention assigned to the selected source,

$$a_{\text{src}} = \frac{1}{BL}\sum_{b=1}^{B}\sum_{i=1}^{L} \boldsymbol{A}_{bij_b^\star}. \tag{13}$$

For the adaptive-NOP experiment, we analogously measure attention to the dedicated token 0 separately for NOP and active queries. We also compare the norm of the output-projected sink value, $\|W_O v_s\|$, with that of ordinary values. For broadcast, we additionally inspect the selected-token value and key norms and the rank structure of the attention update, as described in Section 3.2.

Value interventions. Let s denote the sink position under intervention: $s = 0$ in the adaptive-NOP task and $s = j^\star$ in the broadcast task. To isolate the causal role of the value carried at this destination, we replace v_s while preserving the attention matrix $\boldsymbol{A}$. If v_s^{patch} denotes the patched value, the resulting local change in the output of the attention branch is

$$\Delta o_i = \boldsymbol{A}_{is} W_O \left(v_s^{\text{patch}} - v_s^{\text{original}}\right). \tag{14}$$

We verify this local prediction at the intervened layer before propagating the modified activation through the remainder of the network.

We consider three main value interventions. First, *zero ablation* sets the sink value to zero. Second, *donor-sink patching* replaces it with the corresponding sink value from an independently sampled donor example. In the broadcast task, this means replacing the recipient value at $j^\star$ with the value at the donor's own selected source position $j^\star_{\text{donor}}$. Third, as a content control, we insert an ordinary donor non-sink value at the same high-attention destination. For broadcast we also zero an ordinary non-sink value while leaving the selected source unchanged. These controls separate the effect of attending strongly to a position from the specific information contained in its value.

Routing intervention. As a complementary intervention, we remove attention to the sink while leaving the values unchanged. For each query i, we set the sink weight to zero and renormalize the remaining attention weights,

$$\widetilde{\boldsymbol{A}}_{ij} = \begin{cases} 0, & j = s, \\ \dfrac{\boldsymbol{A}_{ij}}{1 - \boldsymbol{A}_{is}}, & j \neq s, \end{cases} \tag{15}$$

whenever $\boldsymbol{A}_{is} < 1$. This intervention tests whether routing through the sink is necessary, whereas value patching tests what information is carried at the selected destination. Because removing a dominant attention column can also disrupt other learned computations, we interpret routing and zero-ablation experiments as necessity tests rather than as selective removal of the hypothesized payload.

Task error and intervention magnitude. For both tasks, we report mean-squared error between the model output and the corresponding target. To quantify the magnitude of an intervention, let Y and $\widetilde{Y}$ denote the original and intervened model outputs and define $\Delta Y = \widetilde{Y} - Y$. We report the root-mean-square output change

$$\mathrm{RMS}(\Delta Y; I) = \sqrt{\frac{1}{|I|d} \sum_{i \in I} \|\Delta y_i\|_2^2}, \tag{16}$$

where I denotes the evaluated query set. For adaptive NOP, this quantity is reported separately for identity-target and active queries, allowing us to test whether interventions selectively affect the tokens that route to the NOP sink.

Broadcast coherence and payload transfer. For broadcast interventions, we additionally ask whether the induced change is shared across recipient tokens and whether it carries the *correct* donor content. Let

$$\overline{\Delta y} = \frac{1}{L} \sum_{i=1}^{L} \Delta y_i \tag{17}$$

denote the token-wise shared component of the intervention-induced output change. We quantify the fraction of intervention energy contained in this shared component as

$$E_{\mathrm{shared}} = \frac{\|\overline{\Delta y}\|_2^2}{\frac{1}{L} \sum_{i=1}^{L} \|\Delta y_i\|_2^2}. \tag{18}$$

Values near one indicate that the intervention produces an approximately common change across recipient positions.

For a recipient example r and independent donor example d, the desired change in broadcast payload is

$$\Delta p^{r \leftarrow d} = \alpha \left(x_{j_d^\star}^{(d)} R - x_{j_r^\star}^{(r)} R \right). \tag{19}$$

We measure payload transfer by the cosine similarity

$$\mathrm{Align} = \cos\left(\overline{\Delta y}, \Delta p^{r \leftarrow d}\right). \tag{20}$$

This distinguishes a coherent but arbitrary perturbation from transfer of the content selected in the donor sequence.

Finally, we construct the corresponding donor-patched target while preserving the recipient's local token content,

$$y_i^{r \leftarrow d} = x_i^{(r)} + \alpha x_{j_d^\star}^{(d)} R, \tag{21}$$

and report the MSE between the intervened model output and $Y^{r \leftarrow d}$. A reduction in this donor-target error, together with high payload alignment, provides evidence that the patched value causally carries the broadcast content rather than merely inducing a large or spatially coherent change.

A.3 Adaptive NOP: Results and Causal Validation

We next test whether the sink learned on the adaptive-NOP task causally implements a null operation, rather than merely correlating with one. We use the block-output variant described in Appendix A.1, in which identity-target queries require the attention branch to perform a NOP, while zero-output queries require an active update. All interventions follow the protocol in Appendix A.2. We evaluate over 64 batches of 256 freshly sampled sequences and exclude query 0 from the groupwise analysis, since token 0 is itself the dedicated sink position.

Adaptive routing to a near-null value. The trained model develops a sharply conditional routing pattern. Identity-target queries assign 99.29% of their attention to the dedicated sink at token 0, compared with only 1.22% for zero-output queries; uniform attention would assign $1/32 \approx 3.13\%$. Thus, the sink is selected specifically when the attention branch should contribute approximately nothing to the residual stream.

The selected destination also carries an unusually small payload. After the output projection, the mean sink-value norm is

$$\|W_O v_s\| = 0.063,$$

compared with 7.872 for ordinary values, making the sink payload approximately $125\times$ smaller. The combination of selective high attention and a near-null value is therefore consistent with the adaptive-NOP mechanism predicted in Section 3.1.

Causal interventions. Table 2 summarizes the causal interventions. Zeroing the already-small sink value leaves task performance essentially unchanged: MSE changes from 0.01869 to 0.01867, while the RMS output change is only 0.00785 on adaptive-NOP queries and 0.00065 on zero-output queries. Cross-example replacement with another sink value leaves the output unchanged.

In contrast, replacing the value at the same high-attention destination with an ordinary donor non-sink value increases task MSE to 0.50009. The induced output change is strongly concentrated on the queries that route to the sink: RMS change is 0.97917 for adaptive-NOP queries, compared with 0.08053 for zero-output queries. Thus, the large attention weights to token 0 are fully capable of transmitting a substantial update when a non-null payload is placed at that destination.

Removing token 0 as an attention destination and renormalizing the remaining attention weights provides a complementary routing intervention. This increases task MSE to 0.14391 and changes adaptive-NOP outputs by 0.49952 RMS, compared with 0.08115 for zero-output queries. Hence, forcing queries that normally use the sink to attend to ordinary values disrupts the NOP computation.

Table 2: **Causal interventions on the adaptive-NOP sink.** Mean results over 64 evaluation batches. Value interventions preserve the learned attention matrix, whereas the routing intervention removes the sink destination and renormalizes the remaining attention weights. Output changes are measured relative to the unmodified model and separately for identity-target (adaptive-NOP) and zero-output queries.

Intervention	Task MSE	Δ RMS: NOP	Δ RMS: zero-output
Baseline	0.018694	0.000000	0.000000
Zero sink value	0.018669	0.007849	0.000645
Donor sink value	0.018694	0.000000	0.000000
Donor non-sink value	0.500093	0.979167	0.080533
Redirect sink attention	0.143905	0.499524	0.081149

Interpretation. These interventions separate the effect of *where* the head attends from *what* is stored at that destination. High attention to token 0 is not ineffectual: replacing its value with an ordinary payload causes a large output change. Instead, the baseline update is negligible because the selected destination carries a near-null value. Conversely, removing access to that destination forces NOP queries to mix ordinary values and substantially degrades the task. In this controlled setting, the sink therefore acts as a learned null destination that the head selects conditionally to suppress its residual update.

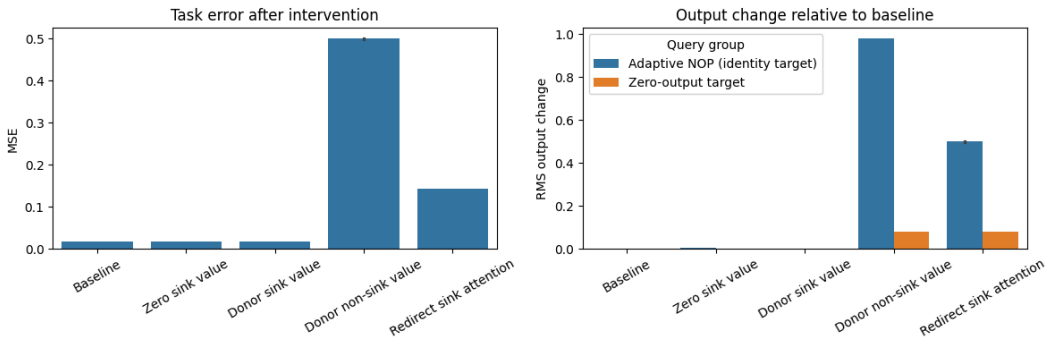


Figure 8: **Causal validation of the adaptive-NOP mechanism.** Left: task error after intervening on the sink value or attention routing. Right: RMS output change relative to the baseline, separated by query type. Zeroing or exchanging the near-null sink value has little effect, whereas inserting an ordinary value at the same destination or redirecting attention away from the sink selectively disrupts identity-target queries. Error bars denote standard errors over evaluation batches.

A.4 Broadcast: Results and Causal Validation

We next test whether the dynamic sinks learned on the broadcast task causally carry the shared, content-dependent payload. We use the two-layer model described in Appendix A.1 and apply the interventions from Appendix A.2 separately at each attention layer. Each intervention is evaluated over 64 batches of 256 independently sampled recipient and donor sequences. Layer-1 interventions are propagated through the first MLP and the complete second block, whereas Layer-2 interventions are propagated through the final MLP.

Dynamic non-null sinks emerge in both layers. Both attention layers learn to route strongly to the content-dependent source $j^\star$. The dominant attention column matches the ground-truth source in 95.36% of examples in L1 and 97.81% in L2. Mean attention assigned to $j^\star$ is 92.90% and 96.68%, respectively. Unlike the dedicated adaptive-NOP sink, these destinations vary across examples and carry substantial content-dependent values. Thus, attention alone identifies both layers as candidate broadcast channels, but does not determine which layer directly carries the transferable payload.

Sink routing is causally important. We first test whether the selected destinations are necessary for the learned computation. The unmodified model achieves an MSE of 0.0481. Zeroing the selected value increases MSE to 4.3652 when intervening in L1 and 3.0729 in L2. Similarly, removing attention to the selected source and renormalizing the remaining weights increases MSE to 0.5886 and 2.2692, respectively. In contrast, zeroing a non-sink value has little effect, yielding MSEs of 0.0547 in L1 and 0.0487 in L2. The selected sink is therefore causally important in both layers.

Zero ablation is not a selective test of broadcast content, however: removing a learned value can destroy other information used downstream. We therefore turn to cross-example patching to ask the stronger question of whether the value at the selected destination carries the *specific* shared payload required by the task.

Cross-example patching localizes the transferable payload. For each recipient sequence r, we replace the value at its selected destination $j_r^\star$ with the value at the independently sampled donor source $j_d^\star$, while preserving the recipient's original attention matrix. We first verify that the immediate change at the intervened attention layer agrees with the exact local prediction in Eq. 14, and then propagate the modified activation through the rest of the model.

The resulting causal effect differs sharply across layers. Patching the L2 sink with the donor sink induces a large, recipient-wide output change (Δ RMS $= 1.363$), with more than 99.99% of its energy contained in the component shared across recipient positions. The mean change aligns with the desired donor-minus-recipient payload difference with cosine similarity 0.944. Consistent with genuine payload transfer, the MSE relative to the donor-patched target falls from 1.914 for the baseline output to 0.230 after the intervention, an improvement of 1.684.

Table 3: **Causal interventions on broadcast sinks.** Interventions are applied to one attention layer at a time and propagated through the remainder of the network. Task MSE is measured against the original broadcast target, whereas donor-target MSE is measured against the target obtained by replacing the recipient's broadcast payload with the donor's payload. Final Δ RMS measures the intervention-induced output change over recipient positions.

Layer	Intervention	Task MSE	Final Δ RMS	Donor-target MSE
L1	Baseline	0.0481	0.0000	1.9139
L1	Zero sink value	4.3652	2.0653	6.8849
L1	Donor sink value	0.1021	0.2317	1.8684
L1	Donor non-sink value	0.2554	0.4410	1.8884
L1	Zero non-sink value	0.0547	0.0652	1.9240
L1	Redirect sink attention	0.5886	0.7248	2.5398
L2	Baseline	0.0481	0.0000	1.9139
L2	Zero sink value	3.0729	1.7242	3.1502
L2	Donor sink value	1.9489	1.3626	0.2299
L2	Donor non-sink value	2.0133	1.3861	2.1227
L2	Zero non-sink value	0.0487	0.0263	1.9148
L2	Redirect sink attention	2.2692	1.4644	2.3315

Importantly, this effect depends on the *content* stored at the selected destination, not simply on placing an arbitrary value under a high-attention column. Replacing the same L2 destination with an ordinary donor non-sink value produces a perturbation of comparable magnitude (Δ RMS $= 1.386$), but its donor-target MSE is 2.123, worse than the unmodified baseline. High attention therefore provides the broadcast routing operation, while the selected value provides the payload that is broadcast.

The corresponding L1 donor-sink intervention produces substantially weaker transfer. Its final output change has RMS 0.232 and aligns with the desired payload difference with cosine similarity only 0.162. Donor-target MSE decreases only from 1.914 to 1.868, an improvement of 0.046. Thus, although interventions establish that the L1 sink is causally involved in the computation, its selected value does not directly carry the transferable broadcast payload observed at the model output.

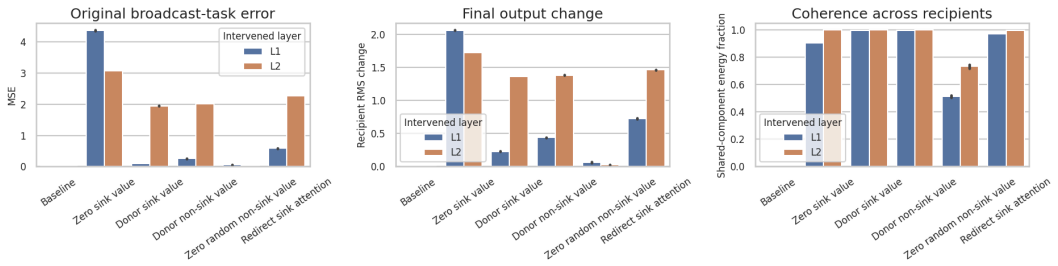


Figure 9: **Causal interventions on the learned broadcast sinks.** Interventions are applied separately in L1 and L2. Left: MSE against the original broadcast target. Center: intervention-induced RMS change at the final output over recipient positions. Right: fraction of the output-change energy explained by the component shared across recipients. Zeroing or redirecting a selected sink is substantially more disruptive than perturbing an ordinary non-sink position. Error bars denote standard errors over evaluation batches.

Interpretation. Together, these interventions distinguish three claims that cannot be established from the attention pattern alone. First, the dynamically selected positions are causally important: removing their values or redirecting attention away from them substantially disrupts the task, whereas matched interventions at non-sink positions do not. Second, the effect of a sink intervention is broadly shared across recipient tokens, as expected for broadcast. Third, and most importantly, cross-example patching shows that the L2 sink carries the *specific content* to be broadcast: changing its value transfers the donor payload while preserving the recipient's local token content. The broadcast sink is therefore not merely correlated with a low-rank shared update; in this controlled model, it causally mediates that update.

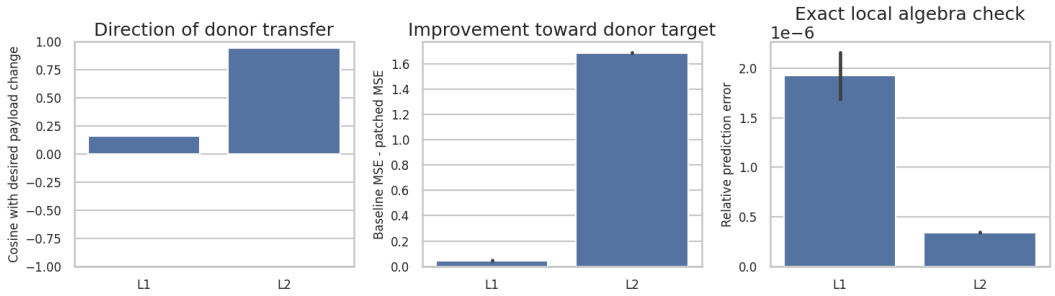


Figure 10: **Cross-example patching identifies the transferable broadcast channel.** Replacing the selected value with that of an independent donor produces strong donor-payload transfer when applied in L2 but little transfer when applied in L1. Left: cosine similarity between the mean recipient-wide output change and the desired donor-minus-recipient payload difference. Center: improvement in MSE toward the donor-patched target. Right: numerical error in the exact local intervention identity. Error bars denote standard errors over evaluation batches.

A.5 Supplementary Controlled-Task Analyses

We collect additional analyses of the controlled tasks that complement the causal experiments in Appendices A.3–A.4. For adaptive NOP, we study how sink formation depends on the frequency and strength of the NOP condition and examine the spectral signature of the learned routing mechanism. For broadcast, we analyze layer specialization, source value norms, query–key geometry, and the rank structure of the attention update. Unless stated otherwise, task definitions and training configurations follow Appendix A.1.

Adaptive-NOP optimization sweeps. We use the branch-supervised adaptive-NOP formulation of Eq. 7 and track sink formation throughout training using the sink score defined in Appendix A.2. For the sweep experiments, we use batch size $B = 2048$ and log the sink score every 10 optimization steps.

We consider two complementary sweeps. First, to vary the frequency with which a NOP is required, we choose the threshold λ to obtain target NOP probabilities $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ under the Gaussian input distribution. For a unit-norm gate vector,

$$\lambda = \Phi^{-1}(1 - p), \tag{22}$$

where Φ is the standard-normal cumulative distribution function. Second, we hold $\lambda = 0$, corresponding to $p = 0.5$, and vary the damping coefficient over

$$\gamma \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 0.5, 1.0\}. \tag{23}$$

Here $\gamma = 0$ corresponds to an exact NOP target, while larger values progressively weaken the required suppression; at $\gamma = 1$, both branches have the same target.

The frequency sweep shows that the sink solution is discovered fastest when active and inactive queries occur with comparable frequency, with the most rapid transition near $p = 0.5$. Sink formation is nevertheless robust away from this maximum-entropy setting: when NOP queries are relatively rare, optimization still eventually converges to a concentrated sink solution, although more slowly.

The damping sweep similarly shows that strict zero-output supervision is not necessary for sink formation. As γ decreases, the desired attention update for flagged tokens approaches zero and the learned sink becomes increasingly pronounced. Sink behavior is already present for nonzero γ, indicating that, in these experiments, pressure to *suppress* an update can favor routing through a dedicated low-value destination even without an exact zero target.

Spectral signature of adaptive routing. We compare the query–key interaction spectrum of a model trained under the standard NOP condition ($\lambda = 0$) with a no-sink control in which the threshold is set to $\lambda = 10$, making NOP events effectively absent. Both models are trained for 10k steps. We

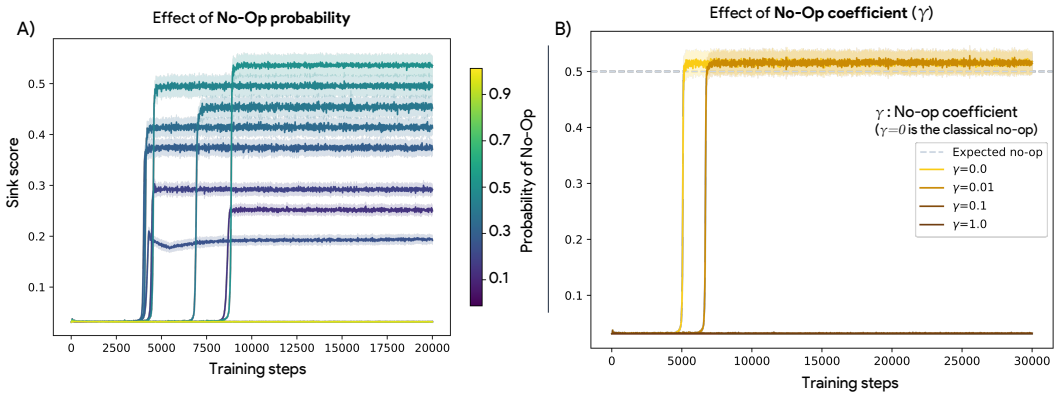


Figure 11: **Optimization sweeps for adaptive NOP. (A)** Sink emergence as a function of the probability that a token requires a NOP. Sink solutions emerge most rapidly when the NOP decision has high entropy, near $p = 0.5$, but also arise when the condition is substantially rarer or more frequent. **(B)** Sink emergence as a function of the damping coefficient γ. Stronger suppression produces stronger sink behavior, but an exact zero target is not required: partially damped objectives also induce a dedicated sink.

compute the singular values of the learned interaction matrix

$$\boldsymbol{\Theta} = \boldsymbol{W}_Q \boldsymbol{W}_K^\top. \tag{24}$$

The sink solution exhibits a pronounced leading singular value relative to the no-sink control (Figure 2B), consistent with the spectral-gating construction in Lemma 3. In that construction, a dominant interaction direction amplifies the feature that determines whether a query should attend to the null token. The learned spectrum therefore provides evidence consistent with conditional routing, complementing the near-zero sink values shown in Figure 2A. A spectral spike alone does not establish NOP behavior: the value and resulting update must also be examined.

Broadcast layer specialization and source values. The broadcast task provides observational signatures complementary to the causal interventions in Appendix A.4. We first examine how the two-layer model distributes computation across depth. Although the architecture could implement partial broadcast updates in both layers, the descriptive solution analyzed in Section 3.2 concentrates source-directed attention in Layer 1, while Layer 2 adopts an identity-like attention pattern (Figure 3A). This illustrates a specialized broadcast solution; the location of the broadcast layer is a property of the learned model, not a requirement of the task.

For each example and layer, we compare the output-projected value norm at the ground-truth source $j^\star$ with the distribution over non-source positions. In the illustrated broadcast layer, source and non-source value norms are comparable, with both distributions centered around $\|\boldsymbol{v}\| \approx 5.5$ (Figure 3B). Unlike adaptive-NOP sinks, the selected broadcast values therefore remain at content scale rather than collapsing toward zero. Together with concentrated source-directed attention, this indicates that the source supplies a substantial update. Whether that update carries the required payload is tested causally in Appendix A.4.

Query–key geometry of broadcast routing. We also inspect how attention identifies the content-dependent source. In the solution shown in Figure 3C, recipient queries become strongly aligned, while the selected source key separates along their common direction. In the displayed PCA coordinates, queries occupy an approximately one-dimensional subspace on the negative side of the leading axis, and the source key extends farther along the same direction. Their alignment supports a large positive query–key dot product, making the source attractive to many recipients simultaneously. The sign of the PCA axis is arbitrary; the relevant observation is the relative alignment and separation of the queries and source key.

This geometry explains how a source selected from the input can become a sink without occupying a fixed sequence position. It does not, by itself, establish that $\boldsymbol{W}_Q \boldsymbol{W}_K^\top$ is rank one: low-dimensional

projected activations do not determine the full spectrum of the interaction matrix. We therefore treat the PCA as a descriptive view of routing and measure the attention update's rank directly.

Rank structure and shared updates. For a nonzero attention update matrix $U \in \mathbb{R}^{L\times d}$, where L is the number of recipient tokens, we use the stable rank Neyshabur et al. (2015); Arora et al. (2018),

$$\mathrm{sr}(U) = \frac{\|U\|_F^2}{\|U\|_2^2}. \tag{25}$$

Here $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the Frobenius and spectral norms, respectively. The broadcast layer has stable rank approximately one, whereas the second layer produces higher-rank updates (Figure 3D). This agrees with Lemma 2: when one source dominates attention, recipient updates are approximately scalar multiples of the same value vector.

The low-rank property concerns the update, not necessarily the residual-stream representations. In particular, adding an identical vector to every token preserves centered variance, as formalized in Lemma 4. We therefore use approximately rank-one updates together with content-scale source values and source-directed attention as evidence of broadcast-like computation, rather than interpreting low rank alone as representation collapse.

For completeness, we also monitor the magnitude of the pre-output attention update $(\boldsymbol{AV})_i$ and the cosine similarity of query vectors across recipient positions. Small variation in update magnitude together with highly aligned queries provides an additional indication that the selected value is distributed broadly. These quantities remain descriptive diagnostics; the cross-example interventions in Appendix A.4 provide the stronger causal test that the selected value carries the broadcast payload.

B Formal statements and proofs

We start by recalling our definition of a sink:

Definition 2 (ϵ-Sink)**.** *A position s is an ϵ-sink if $\boldsymbol{A}_{is} \geq 1-\epsilon$ for all positions $i \in \{1,\dots,n\}$, where $\epsilon \in [0,1)$ is small.*

B.1 Proof of Lemma 1: Uniqueness of the exact no-op solution

Proof. For any query $i \in I$, write

$$\boldsymbol{o}_i = \boldsymbol{A}_{is}\boldsymbol{v}_s + \sum_{j\neq s} \boldsymbol{A}_{ij}\boldsymbol{v}_j = (1-\epsilon_i)\boldsymbol{v}_s + \boldsymbol{r}_i, \tag{26}$$

where $\epsilon_i = 1 - \boldsymbol{A}_{is} \in [0,\epsilon]$. Since attention weights are nonnegative and sum to one, the residual satisfies $\|\boldsymbol{r}_i\| \leq \epsilon_i V_{\max} \leq \epsilon V_{\max}$. The triangle inequality therefore gives

$$(1-\epsilon)\|\boldsymbol{v}_s\| \leq (1-\epsilon_i)\|\boldsymbol{v}_s\| = \|\boldsymbol{o}_i - \boldsymbol{r}_i\| \leq \delta + \epsilon V_{\max}, \tag{27}$$

which proves the stated bound.

At finite ϵ, setting $\boldsymbol{v}_s = \boldsymbol{0}$ gives $\|\boldsymbol{o}_i\| \leq \epsilon V_{\max}$. Alternatively, exact cancellation requires $\boldsymbol{v}_s = -\boldsymbol{r}_i/(1-\epsilon_i)$ consistently across $i \in I$, but such a value remains subject to the same bound. Thus cancellation cannot sustain a nonzero sink value as $\epsilon, \delta \to 0$ with uniformly bounded $V_{\max}$.

For $\epsilon = 0$, all residual contributions vanish and $\boldsymbol{o}_i = \boldsymbol{v}_s$ for every $i \in I$. Consequently, exact no-op holds if and only if $\boldsymbol{v}_s = \boldsymbol{0}$, whereas a positive tolerance δ permits any sink value with norm at most δ. □

B.2 Mechanisms for Adaptivity: Statement and Proof

Conditional suppression requires both a near-zero sink value and a routing mechanism that selects it for the appropriate queries. We examine how a linear query–key interaction can implement this conditional routing. The relevant requirement is that the sink logit exceed competing logits: softmax attention approaches, rather than attains, a perfect gate at finite parameter values.

Lemma 3 (**Mechanisms for Adaptivity**). *Consider queries $\boldsymbol{q}_i = \boldsymbol{W}_Q^\top \boldsymbol{x}_i$, a sink key $\boldsymbol{k}_s = \boldsymbol{W}_K^\top \boldsymbol{x}_s$, and interaction matrix $\boldsymbol{\Theta} = \boldsymbol{W}_Q \boldsymbol{W}_K^\top$. The sink logit satisfies*

$$S_{is} = \frac{\boldsymbol{x}_i^\top \boldsymbol{\Theta} \boldsymbol{x}_s}{\sqrt{d_h}}, \qquad |S_{is}| \le \frac{\|\boldsymbol{x}_i\| \, \|\boldsymbol{\Theta}\|_2 \, \|\boldsymbol{x}_s\|}{\sqrt{d_h}}. \tag{28}$$

*For selected queries, $\boldsymbol{A}_{is}$ approaches one when S_{is} exceeds competing logits by a sufficiently large margin. In constructions aligned with a triggering feature, this margin can be increased through either **(i) spectral gating**, by increasing a leading singular value of $\boldsymbol{\Theta}$ along that feature direction, or **(ii) massive activation**, by increasing the sink activation norm along the corresponding direction, provided competing logits remain controlled.*

Proof. Consider a simplified attention mechanism with a single head. Let $\boldsymbol{x}_i$ be a query token and $\boldsymbol{x}_s$ be the dedicated sink token. The attention score (logit) for the sink is given by:

$$S_{is} = \frac{1}{\sqrt{d_h}} \boldsymbol{x}_i^\top \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_s = \frac{1}{\sqrt{d_h}} \boldsymbol{x}_i^\top \boldsymbol{\Theta} \boldsymbol{x}_s. \tag{29}$$

We assume the gating decision is based on a specific feature direction $\boldsymbol{u}$ (with $\|\boldsymbol{u}\| = 1$). We decompose the query as $\boldsymbol{x}_i = \lambda_i \boldsymbol{u} + \boldsymbol{x}_\perp$, where λ_i is the scalar activation of the feature triggering the NOP, and $\boldsymbol{x}_\perp \perp \boldsymbol{u}$.

To consistently route attention to the sink when the feature is active ($\lambda_i > 0$), the model aligns the interaction matrix $\boldsymbol{\Theta}$ and the sink token $\boldsymbol{x}_s$ with this direction. We construct:

- The interaction matrix as rank-1: $\boldsymbol{\Theta} = \sigma \boldsymbol{u} \boldsymbol{u}^\top$, where $\sigma = \|\boldsymbol{\Theta}\|_2$ is the spectral norm.
- The sink token as aligned with the flag: $\boldsymbol{x}_s = \gamma_s \boldsymbol{u}$, where $\gamma_s = \|\boldsymbol{x}_s\|$ is the token norm.

Substituting these into the attention score equation:

$$S_{is} = \frac{1}{\sqrt{d_h}} (\lambda_i \boldsymbol{u} + \boldsymbol{x}_\perp)^\top (\sigma \boldsymbol{u} \boldsymbol{u}^\top)(\gamma_s \boldsymbol{u}) = \frac{\lambda_i \sigma \gamma_s}{\sqrt{d_h}}. \tag{30}$$

For the attention weight $\boldsymbol{A}_{is}$ to approach 1 (hard gating), the score S_{is} must exceed the scores of all other N tokens by a large margin M (where M depends on the desired sharpness of the softmax). That is:

$$\frac{\lambda_i \sigma \gamma_s}{\sqrt{d_h}} \gg \max_{j \neq s} S_{ij} \implies \sigma \gamma_s \gg \frac{M \sqrt{d_h}}{\lambda_i}. \tag{31}$$

This inequality $\sigma \gamma_s \gg C$ reveals the two distinct optimization paths available to the model to satisfy the gating condition: **(i) Spectral Gating:** The model increases σ (the singular value of $\boldsymbol{W}_Q \boldsymbol{W}_K^\top$) while keeping the sink norm γ_s moderate. **(ii) Massive Activation:** The model increases γ_s (the norm of the sink token $\boldsymbol{x}_s$) while keeping the weight spectral norm σ moderate. Both solutions are mathematically equivalent for producing the necessary logit S_{is} to trigger the sink. □

The construction separates two ways to strengthen the same conditional routing operation: increasing the query–key interaction along a relevant direction or increasing the sink activation along that direction. This offers a possible interpretation of the massive activations observed in outlier tokens Darcet et al. (2024): they can support reliable routing to a null value without requiring a comparably large interaction-matrix norm. These are sufficient constructions, not an exhaustive classification of learned routing mechanisms. Moreover, neither a spectral spike nor a large activation establishes NOP behavior on its own; conditional routing must be paired with a negligible value contribution.

B.3 Proof of Lemma 2: Rank-1 Update

Proof. The attention output matrix is given by $\boldsymbol{O} = \boldsymbol{A}\boldsymbol{V}$, where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the row-stochastic attention matrix and $\boldsymbol{V} \in \mathbb{R}^{n \times d_h}$ is the matrix of value vectors with rows $\boldsymbol{v}_1^\top, \ldots, \boldsymbol{v}_n^\top$.

Let s be an ϵ-broadcast sink. We decompose the attention matrix as:

$$\boldsymbol{A} = \boldsymbol{A}_s \boldsymbol{e}_s^\top + \boldsymbol{E}, \tag{32}$$

where $\boldsymbol{a}_s = (\boldsymbol{A}_{1s}, \ldots, \boldsymbol{A}_{ns})^\top \in \mathbb{R}^n$ is the s-th column of $\boldsymbol{A}$, $\boldsymbol{e}_s$ is the s-th standard basis vector, and $\boldsymbol{E}$ is the residual matrix with entries $\boldsymbol{E}_{ij} = \boldsymbol{A}_{ij}$ for $j \neq s$ and $\boldsymbol{E}_{is} = 0$.

Since $\boldsymbol{A}$ is row-stochastic and $\boldsymbol{A}_{is} \geq 1 - \epsilon$, we have $\sum_{j \neq s} \boldsymbol{A}_{ij} \leq \epsilon$ for each row i. Thus, each row of $\boldsymbol{E}$ has ℓ_1-norm at most ϵ, yielding $\|\boldsymbol{E}\|_\infty \leq \epsilon$.

Multiplying by $\boldsymbol{V}$:

$$\boldsymbol{O} = \boldsymbol{A}\boldsymbol{V} = \boldsymbol{a}_s \boldsymbol{e}_s^\top \boldsymbol{V} + \boldsymbol{E}\boldsymbol{V} = \boldsymbol{a}_s \boldsymbol{v}_s^\top + \boldsymbol{E}\boldsymbol{V}. \tag{33}$$

The first term $\boldsymbol{a}_s \boldsymbol{v}_s^\top$ is rank-1. The residual satisfies:

$$\|\boldsymbol{E}\boldsymbol{V}\|_\infty \leq \|\boldsymbol{E}\|_\infty \max_j \|\boldsymbol{v}_j\| \leq \epsilon \cdot \max_j \|\boldsymbol{v}_j\|. \tag{34}$$

Therefore:

$$\left\|\boldsymbol{O} - \boldsymbol{a}_s \boldsymbol{v}_s^\top\right\|_\infty \leq \epsilon \cdot \max_j \|\boldsymbol{v}_j\|. \tag{35}$$

The rank-1 matrix $\boldsymbol{a}_s \boldsymbol{v}_s^\top$ has singular value decomposition $\sigma_1 \boldsymbol{u}_1 \boldsymbol{w}_1^\top$, where $\sigma_1 = \|\boldsymbol{a}_s\|\|\boldsymbol{v}_s\|$, $\boldsymbol{u}_1 = \boldsymbol{a}_s / \|\boldsymbol{a}_s\|$, and $\boldsymbol{w}_1 = \boldsymbol{v}_s / \|\boldsymbol{v}_s\|$. Since $\boldsymbol{A}_{is} \geq 1 - \epsilon$ for all i, we have $\|\boldsymbol{a}_s\| \geq \sqrt{n}(1 - \epsilon)$. Thus, the dominant singular value satisfies:

$$\sigma_1 \geq \sqrt{n}(1 - \epsilon)\|\boldsymbol{v}_s\|, \tag{36}$$

and the right singular vector aligns exactly with $\boldsymbol{v}_s / \|\boldsymbol{v}_s\|$. □

B.4 Shared Additive Component: Statement and Proof

A broadcast sink introduces a shared component into recipient updates, but this does not imply that their residual-stream representations collapse. The following result bounds the change in centered token variance and distinguishes a shared additive update from contraction toward the token mean.

Lemma 4 (Shared Additive Component). *Suppose s is an ϵ-sink for all queries in the attention update*

$$\boldsymbol{x}_i^{\ell+1} = \boldsymbol{x}_i^\ell + \sum_j \boldsymbol{A}_{ij} \boldsymbol{v}_j. \tag{37}$$

Let

$$\bar{\boldsymbol{x}}^\ell = \frac{1}{n} \sum_i \boldsymbol{x}_i^\ell, \qquad \mathcal{V}_\ell = \frac{1}{n} \sum_i \|\boldsymbol{x}_i^\ell - \bar{\boldsymbol{x}}^\ell\|^2, \tag{38}$$

and define the variance of the sink attention weights by

$$\bar{A}_s = \frac{1}{n} \sum_i \boldsymbol{A}_{is}, \qquad \mathbb{V}(\boldsymbol{A}_s) = \frac{1}{n} \sum_i (\boldsymbol{A}_{is} - \bar{A}_s)^2. \tag{39}$$

With value norms and $\mathcal{V}_\ell$ bounded independently of ϵ, we have

$$\mathcal{V}_{\ell+1} \leq \mathcal{V}_\ell + 2\|\boldsymbol{v}_s\| \sqrt{\mathcal{V}_\ell \, \mathbb{V}(\boldsymbol{A}_s)} + \|\boldsymbol{v}_s\|^2 \mathbb{V}(\boldsymbol{A}_s) + \mathcal{O}(\epsilon). \tag{40}$$

Proof. Let $\boldsymbol{x}_i^{\ell+1} = \boldsymbol{x}_i^\ell + \boldsymbol{o}_i$ where $\boldsymbol{o}_i = \sum_j \boldsymbol{A}_{ij} \boldsymbol{v}_j$. Let s be an ϵ-broadcast sink, so that $\boldsymbol{A}_{is} \geq 1 - \epsilon$ for every query i.

We first bound the deviation from the broadcast approximation. For each position i:

$$\boldsymbol{o}_i = \boldsymbol{A}_{is} \boldsymbol{v}_s + \sum_{j \neq s} \boldsymbol{A}_{ij} \boldsymbol{v}_j. \tag{41}$$

The residual term satisfies:

$$\left\| \sum_{j \neq s} \boldsymbol{A}_{ij} \boldsymbol{v}_j \right\| \leq \sum_{j \neq s} \boldsymbol{A}_{ij} \|\boldsymbol{v}_j\| \leq \epsilon \cdot \max_j \|\boldsymbol{v}_j\|. \tag{42}$$

Thus, $\boldsymbol{o}_i = \boldsymbol{A}_{is}\boldsymbol{v}_s + \boldsymbol{r}_i$ where $\|\boldsymbol{r}_i\| \le \epsilon \cdot \max_j \|\boldsymbol{v}_j\|$.

The mean representation after the block is:

$$\bar{\boldsymbol{x}}^{\ell+1} = \bar{\boldsymbol{x}}^{\ell} + \bar{\boldsymbol{o}}, \tag{43}$$

where $\bar{\boldsymbol{o}} = \frac{1}{n}\sum_i \boldsymbol{o}_i = \bar{a}_s \boldsymbol{v}_s + \bar{\boldsymbol{r}}$, with $\bar{a}_s = \frac{1}{n}\sum_i \boldsymbol{A}_{is}$ and $\|\bar{\boldsymbol{r}}\| \le \epsilon \cdot \max_j \|\boldsymbol{v}_j\|$.

The centered representation is:

$$\boldsymbol{x}_i^{\ell+1} - \bar{\boldsymbol{x}}^{\ell+1} = \left(\boldsymbol{x}_i^{\ell} - \bar{\boldsymbol{x}}^{\ell}\right) + (\boldsymbol{A}_{is} - \bar{a}_s)\boldsymbol{v}_s + (\boldsymbol{r}_i - \bar{\boldsymbol{r}}). \tag{44}$$

Let $\boldsymbol{d}_i^{\ell} = \boldsymbol{x}_i^{\ell} - \bar{\boldsymbol{x}}^{\ell}$ and $\delta_i = \boldsymbol{A}_{is} - \bar{a}_s$. The variance after the block is:

$$\frac{1}{n}\sum_i \left\|\boldsymbol{x}_i^{\ell+1} - \bar{\boldsymbol{x}}^{\ell+1}\right\|^2 = \frac{1}{n}\sum_i \left\|\boldsymbol{d}_i^{\ell} + \delta_i \boldsymbol{v}_s + (\boldsymbol{r}_i - \bar{\boldsymbol{r}})\right\|^2. \tag{45}$$

Expanding gives:

$$\begin{aligned} &= \frac{1}{n}\sum_i \left\|\boldsymbol{d}_i^{\ell}\right\|^2 + \frac{1}{n}\sum_i \delta_i^2 \|\boldsymbol{v}_s\|^2 + \frac{1}{n}\sum_i \|\boldsymbol{r}_i - \bar{\boldsymbol{r}}\|^2 \\ &\quad + \frac{2}{n}\sum_i \delta_i \left\langle \boldsymbol{d}_i^{\ell}, \boldsymbol{v}_s \right\rangle + \frac{2}{n}\sum_i \left\langle \boldsymbol{d}_i^{\ell}, \boldsymbol{r}_i - \bar{\boldsymbol{r}} \right\rangle \\ &\quad + \frac{2}{n}\sum_i \delta_i \left\langle \boldsymbol{v}_s, \boldsymbol{r}_i - \bar{\boldsymbol{r}} \right\rangle. \end{aligned} \tag{46}$$

The second term equals $\mathbb{V}(\boldsymbol{a}_s)\|\boldsymbol{v}_s\|^2$, where $\mathbb{V}(\boldsymbol{a}_s) = \frac{1}{n}\sum_i \delta_i^2$ is the variance of attention weights to the sink. Since $\|\boldsymbol{r}_i - \bar{\boldsymbol{r}}\| \le 2\epsilon \max_j \|\boldsymbol{v}_j\|$ and $|\delta_i| \le \epsilon$, the terms involving $\boldsymbol{r}_i$ have total absolute value $\mathcal{O}(\epsilon)$, provided value norms and the input token variance $\frac{1}{n}\sum_i \|\boldsymbol{d}_i^{\ell}\|^2$ are bounded independently of ϵ.

For the cross-term $\frac{2}{n}\sum_i \delta_i \langle \boldsymbol{d}_i^{\ell}, \boldsymbol{v}_s \rangle$, by Cauchy–Schwarz:

$$\left|\frac{2}{n}\sum_i \delta_i \left\langle \boldsymbol{d}_i^{\ell}, \boldsymbol{v}_s \right\rangle\right| \le 2\sqrt{\mathbb{V}(\boldsymbol{a}_s)} \cdot \sqrt{\frac{1}{n}\sum_i \left\|\boldsymbol{d}_i^{\ell}\right\|^2} \cdot \|\boldsymbol{v}_s\|. \tag{47}$$

Combining these bounds:

$$\begin{aligned} \frac{1}{n}\sum_i \left\|\boldsymbol{x}_i^{\ell+1} - \bar{\boldsymbol{x}}^{\ell+1}\right\|^2 \le \frac{1}{n}\sum_i \left\|\boldsymbol{x}_i^{\ell} - \bar{\boldsymbol{x}}^{\ell}\right\|^2 \\ + 2\|\boldsymbol{v}_s\|\sqrt{\mathbb{V}(\boldsymbol{a}_s)}\sqrt{\frac{1}{n}\sum_i \left\|\boldsymbol{x}_i^{\ell} - \bar{\boldsymbol{x}}^{\ell}\right\|^2} \\ + \|\boldsymbol{v}_s\|^2 \mathbb{V}(\boldsymbol{a}_s) + \mathcal{O}(\epsilon). \end{aligned} \tag{48}$$

When attention to the sink is uniform, i.e., $\boldsymbol{A}_{is} = c$ for all i with $c \ge 1 - \epsilon$, we have $\mathbb{V}(\boldsymbol{a}_s) = 0$ and $\delta_i = 0$ for all i. In this case, the variance expansion gives:

$$\frac{1}{n}\sum_i \left\|\boldsymbol{x}_i^{\ell+1} - \bar{\boldsymbol{x}}^{\ell+1}\right\|^2 = \frac{1}{n}\sum_i \left\|\boldsymbol{x}_i^{\ell} - \bar{\boldsymbol{x}}^{\ell}\right\|^2 + \mathcal{O}(\epsilon). \tag{49}$$

The variance is approximately preserved, while all tokens receive the same additive component $c \cdot \boldsymbol{v}_s$, shifting collectively along the direction $\boldsymbol{v}_s$. For a perfect sink ($\epsilon = 0$), the residual terms vanish and centered variance is preserved exactly. □

For a perfect sink, $\boldsymbol{A}_{is} = 1$ for every recipient, so each token receives the same additive vector $\boldsymbol{v}_s$. Subtracting the updated token mean removes this shared component exactly:

$$\boldsymbol{x}_i^{\ell+1} - \bar{\boldsymbol{x}}^{\ell+1} = \boldsymbol{x}_i^{\ell} - \bar{\boldsymbol{x}}^{\ell}.$$

Consequently, $\mathcal{V}_{\ell+1} = \mathcal{V}_\ell$. For approximate sinks, variation in recipient weights and residual contributions from other values can change centered variance, as captured by the bound. Thus, an approximately rank-one attention update is a signature of shared communication, not by itself evidence of representation collapse or a guaranteed increase in pairwise cosine similarity.

C LeJEPA Mitigation Experiments: Training Details

We train two sets of LeJEPA ViT-L models from scratch on ImageNet-1K. The first is a sweep over registers and attention gating with three seeds per variant (Appendix C.1; Tables 8 and 9). The second is the four single-seed models of Table 1 in Section 5 (Appendix C.2). Both sets share the recipe of Tables 4–7. Appendix C.2 lists where the models of Table 1 differ from it, and Appendix C.3 gives the probe protocols of both sets.

C.1 Common recipe and the register and gating sweep

We trained ViT-L models from scratch on ImageNet-1K for 100 epochs using the LeJEPA self-supervised objective Balestriero & LeCun (2025), which combines a SIGReg regularization loss and a multi-view invariance loss. Architecture specifications are provided in Table 4, optimization hyperparameters in Table 5, regularization in Table 6, and data augmentation in Table 7. Each model was trained on 4 NVIDIA H100 GPUs using PyTorch DDP.

We evaluate six variants: (1) Baseline, (2) Registers (4 tokens), (3) Elementwise gating, (4) Elementwise gating + Registers, (5) Headwise gating, and (6) Headwise gating + Registers, each trained with three random seeds (9000, 9001, 9002). These models provide the register comparisons: per-seed ImageNet-1K linear probe results are provided in Table 8 and dense-prediction probes in Table 9. They are not the models of Table 1, which are described in Appendix C.2.

Table 4: ViT-Large architecture specifications for LeJEPA experiments.

Parameter	**Value**
Image size	224×224
Patch size	16×16
Number of patches	196
Embedding dimension	1024
Transformer blocks	24
Attention heads	16
Head dimension	64
MLP ratio	4.0
MLP hidden dimension	4096
QKV bias	False
Register tokens (when enabled)	4
LayerScale init (attention and MLP branches)	10^{-4} (see Table 10)
Readout	CLS token
Projector	$1024 \to 2048 \to 2048 \to 128$ (BN + ReLU)
Projector output dimension	128

C.2 The models of Table 1

Table 1 compares four models: Baseline, Gating, Global + Penalty, and Gating + Global + Penalty. Each is a single backbone (seed 9000), trained without registers using the recipe of Tables 4–7 on 4 NVIDIA H100 GPUs (per-GPU batch 64, effective batch 256). Baseline and Gating are separate runs from the seed-9000 models of the sweep. All four models are trained for 50 epochs with the learning-rate schedule of Table 5, which is defined over 100 epochs (10 warm-up epochs, then cosine decay to $\mathrm{lr}/1000$ at epoch 100). At epoch 50 the learning rate is therefore 2.9×10^{-4}, not yet fully

Table 5: Optimization hyperparameters for LeJEPA experiments.

Parameter	Value
Optimizer	AdamW
Learning rate	5×10^{-4}
Weight decay	0.05
Warmup epochs	10
Total epochs	100
LR schedule	Linear warmup + cosine annealing to lr/1000
Gradient clip	1.0
Batch size (per GPU)	64
Effective batch size	256 (4 GPUs)
Precision	Mixed (bfloat16)
Online probe LR	1×10^{-3}
Number of views	4
SIGReg weight λ	0.02

Table 6: Regularization hyperparameters for LeJEPA experiments.

Technique	Value
Drop path (stochastic depth)	0.1 (increasing linearly over depth)
Dropout	0.0
Attention dropout	0.0

annealed. Every number in Tables 1, 14 and 15 is measured on the checkpoint saved after epoch 50. Table 10 lists the settings that differ between the four models.

Gating. Following the G1 variant of Qiu et al. (2025), each attention block multiplies the concatenated head outputs elementwise by $\sigma(\boldsymbol{W}_\theta \widetilde{\boldsymbol{x}}_i + \boldsymbol{b}_\theta) \in (0,1)^d$ before the output projection. The gate is computed from the block's normalized input $\widetilde{\boldsymbol{x}}_i$ with a full $d \times d$ map.

Broadcast penalty. Both penalized models add the penalty of Eq. 5 with $\lambda_B = 0.1$, $\mathcal{L}_B = \{1, \ldots, 23\}$ (block 0 unpenalized) and $\epsilon = 10^{-12}$. It is active from the first training step with its full weight, with no warm-up or ramp. The ratio is computed per block on each GPU's mini-batch (all views), and the per-block ratios are averaged over $\mathcal{L}_B$. The common component $\bar{\boldsymbol{o}}^{\ell,\mathrm{attn}}$ averages over all tokens, including `[CLS]`. Gradients flow through both numerator and denominator. In Gating + Global + Penalty, $\boldsymbol{O}^{\ell,\mathrm{attn}}$ is the output-projected attention update after LayerScale, computed without the elementwise gate. The global-pathway update is not penalized.

C.3 Linear probes

ImageNet-1K. The ImageNet column of Table 1 is the LeJEPA trainer's online linear probe. It is a LayerNorm followed by a linear layer on the stop-gradient `[CLS]` embedding, trained jointly with the backbone on every training view with cross-entropy (AdamW, learning rate 10^{-3}, weight decay 10^{-7}, same warm-up and cosine schedule as the backbone). We report top-1 accuracy on the 50,000 validation images (resize 256, center crop 224) after epoch 50. Each variant has a single backbone and this probe has no seed of its own, so the column has no error bar.

Semantic segmentation. Table 11 lists the ADE20K and Pascal VOC 2012 probe settings shared by Table 1 and the sweep (Table 9). Table 12 lists where they differ: augmentation, probe length and what the $\pm$ measures. For Table 1, the frozen features are extracted once per backbone and dataset at a fixed resize and cached, so the probe sees no augmentation. The VOC12 probe runs 100 epochs because at 25 epochs it has not converged: Baseline reaches 0.49, 0.55 and 0.60 mIoU after 25, 50 and 100 epochs. The ordering of the four models is the same at all three lengths.

Compute resources. Experiments were run on a compute cluster using NVIDIA H100 GPUs. The LeJEPA ViT-L experiments were trained with PyTorch DDP on 4 H100 GPUs per run. For the sweep, each 100-epoch run took approximately 3 days, corresponding to roughly 300 H100 GPU-hours

Table 7: Training data augmentation pipeline for LeJEPA experiments. Four augmented views are generated per image.

Augmentation	**Parameters**
Random resized crop	Scale: $(0.08, 1.0)$
Color jitter	Brightness/contrast/saturation $= 0.8$, hue $= 0.2$, $p = 0.8$
Random grayscale	$p = 0.2$
Gaussian blur	Kernel size $= 7$, $\sigma \in (0.1, 2.0)$, $p = 0.5$
Random solarize	Threshold $= 128$, $p = 0.2$
Random horizontal flip	$p = 0.5$
Normalization	ImageNet statistics
Validation only	
Resize	256×256
Center crop	224×224

Table 8: ImageNet-1K linear probe top-1 accuracy (%) for each LeJEPA variant across three seeds.

Variant	**9000**	**9001**	**9002**
Baseline	69.31	69.37	69.44
Registers	69.14	69.37	69.42
Elementwise gating	69.56	69.49	69.52
Elementwise gating + Reg.	69.28	69.46	69.35
Headwise gating	69.02	69.05	69.13
Headwise gating + Reg.	69.28	69.15	68.95

per seed, and we trained three seeds for each variant. The four models of Table 1 took 33 to 51 minutes per epoch (Table 10), so the 50 epochs behind Table 1 correspond to roughly 110–170 H100 GPU-hours per model. Training linear semantic-segmentation probes on frozen LeJEPA backbones required one GPU per probe and only a few GPU-hours per probe. For Table 1, features are extracted once per backbone and dataset, after which each probe takes under an hour. Similarly, toy-model experiments and pretrained-model diagnostics were run on a single GPU and were substantially cheaper than the LeJEPA training runs, typically completing within a few GPU-hours per run. All experiments were implemented in PyTorch.

Existing assets and licenses. Table 13 lists the third-party datasets, codebases, and pretrained model checkpoints used in this work, together with their licenses or terms of use. We do not redistribute any third-party datasets or pretrained checkpoints.

Table 9: Linear probing performance on dense prediction tasks for the register and gating sweep. We report mean and standard deviation over three independently trained LeJEPA backbones. Probe protocol in Tables 11 and 12 (augmented; 10 epochs on ADE20K, 25 on VOC12). These numbers use a different protocol and different backbones from Table 1 and are not comparable with it.

Variant	ADE20K	VOC12
Baseline	0.166 ± 0.006	0.438 ± 0.016
Registers	0.187 ± 0.002	0.499 ± 0.001
Headwise gating	0.185 ± 0.0003	0.476 ± 0.006
Elementwise gating	0.186 ± 0.004	0.480 ± 0.002
Headwise + registers	0.199 ± 0.003	$\mathbf{0.524 \pm 0.013}$
Elementwise + registers	$\mathbf{0.200 \pm 0.006}$	0.522 ± 0.017

Table 10: Settings that differ between the four models of Table 1. Everything else follows Tables 4–7. Gate: elementwise G1 attention output gate. Pathway: Eq. 4 in all 24 blocks. Penalty: Eq. 5; [a]computed on the attention update without the gate. Time per epoch: median over the first 50 epochs on 4 H100 GPUs.

	Baseline	**Gating**	**Global + Penalty**	**Gating + Global + Penalty**
Attention gate	–	G1	–	G1
LayerScale init (attn., MLP)	10^{-4}	10^{-4}	1.0	1.0
Global pathway	–	–	yes	yes
Penalty λ_B	–	–	0.1	0.1^a
Added parameters	–	25.2M	3.19M	28.4M
Time per epoch	33 min	37 min	43 min	51 min

D Additional Experimental Results and Figures

Sink counts for the models of Table 1. Table 14 gives the per-image sink counts quoted in Section 5. We run the epoch-50 checkpoint of each model (Appendix C.2) on 1,984 ADE20K validation images (resize 256, center crop 224). For every image we count the (layer, head) pairs in layers 1–23 in which at least 90% of the queries place at least 0.5 of their attention on the same key token; block 0 is excluded. A sink is NOP-like if the sink token's value-norm ratio $\|\boldsymbol{v}_s\|/\overline{\|\boldsymbol{v}\|}$ (its value norm over the head's mean value norm of the other tokens) is at most 0.2, and broadcast-like if it is at least 0.5. Sinks with intermediate ratios count only in the total. This rule relaxes Definition 1: it fixes $\epsilon = 0.5$ but requires it on 90% of the queries rather than all of them. Appendix E.2 recounts the same models under the literal definition.

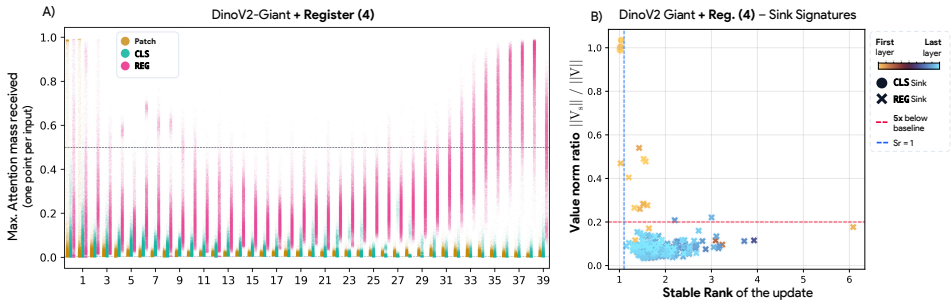


Figure 12: **(Left) Registers absorb sink behavior.** In DINOv2-G + Reg.(4), register tokens (pink) capture nearly all attention mass across layers, displacing patch and [CLS] sinks. **(Right) Registers inherit both regimes.** Register sinks cluster into the same two phenotypes: NOP (low norm, majority) and broadcast (high norm, rank-1). Registers are repurposed for both mechanisms.

Table 11: Linear segmentation probe settings shared by Table 1 and the sweep (Table 9). The backbone is frozen and only the linear head is trained.

Setting	Value
Backbone	frozen (eval mode, fp32); final-norm patch tokens only
Input resolution	512×512; positional embeddings interpolated bicubically
Head	linear $1024 \to C$ per patch; logits upsampled bilinearly to 512×512
Loss	cross-entropy; background and unlabeled pixels ignored
Classes C	150 (ADE20K), 20 (VOC12)
Optimizer	AdamW, lr 5×10^{-3}, weight decay 10^{-2}, batch 32, gradient clip 1.0
LR schedule	100-iteration linear warm-up, then cosine decay to 1% of the lr
Data splits (train / val)	ADE20K 20,210 / 2,000; VOC12 1,464 / 1,449
Reported value	validation mIoU at the best probe epoch

Table 12: Linear segmentation probe settings that differ between Table 1 and the sweep (Table 9). Table 1 probes are trained on features extracted once per backbone at a fixed 512×512 resize and stored in fp16, so they see no augmentation. Sweep probes use a random resized crop (scale 0.5–1, aspect ratio 3/4–4/3) and a horizontal flip.

Setting	Table 1	Table 9
Train augmentation	none	crop + flip
Probe epochs, ADE20K	25	10
Probe epochs, VOC12	100	25
Backbone seeds	9000	9000, 9001, 9002
Backbone checkpoint	epoch 50	epoch 100
Probe seeds, ADE20K	42, 43, 44	42
Probe seeds, VOC12	46, 47, 48	46
$\pm$	std over probe seeds	std over backbones

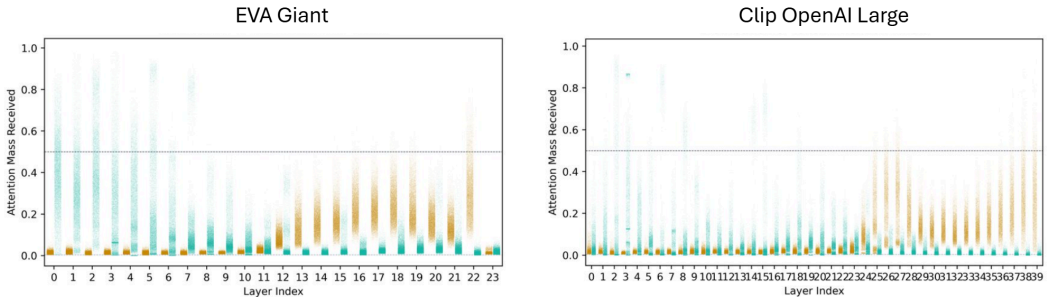


Figure 13: **Sink token transition across layers.** EVA Giant and Clip OpenAI Large both exhibit a handoff pattern, although the pattern is more pronounced in EVA: `[CLS]` serves as the sink in early layers but yields to patch tokens in later layers.

E Supplementary mitigation results and ablations

E.1 Controlled-Task Causal Validation of the Global Pathway

We test whether the global-pathway intervention moves broadcast computation out of softmax attention on the controlled task from Appendix A.1. We train three matched variants: an attention-only baseline, a model with the explicit global pathway, and the same model with the all-layer common-mode penalty from Eq. 5, using $\lambda_B = 1$. All variants use the same task, initialization, and training batches. In this controlled experiment, the global pathway pools the normalized token representations into a single state g^ℓ and broadcasts $W^\ell_{O,G} g^\ell$ uniformly to every recipient; we omit the token-dependent recipient gate to isolate the relocation of shared communication.

The penalty moves broadcast out of attention. The attention-only baseline reaches task MSE 0.3093 and assigns approximately 69% of its attention to the selected source $j^\star$ in both layers. Adding

Table 13: Existing assets used in this work and their licenses or terms of use.

Asset	Use in this paper	License or terms of use
ImageNet-1K	LeJEPA PT, evaluation	ImageNet Terms of Access: non-commercial research and educational use
ADE20K	Linear semantic-segmentation probing	ADE20K Terms of Use: non-commercial research and educational use
Pascal VOC 2012	Linear semantic-segmentation probing	PASCAL VOC dataset terms / citation requirements; used for research benchmarking
DINOv2 code and checkpoints	Pretrained ViT sink analysis	Apache License 2.0
OpenCLIP code/checkpoints	Pretrained ViT sink analysis	MIT License for OpenCLIP code; checkpoint/model-card terms as released by OpenCLIP
EVA code/checkpoints	Pretrained ViT sink analysis	MIT License for EVA repository; checkpoint/model-card terms as released by EVA
LeJEPA code	Model architecture and training framework	CC BY-NC 4.0 License

Table 14: Sinks per image for the models of Table 1 (epoch-50 checkpoints; 1,984 ADE20K validation images; layers 1–23; counting rule in the text). "All sinks" includes sinks with intermediate value-norm ratios. "On [CLS]" counts sinks of any type located on the [CLS] token; these models have no registers.

Variant	NOP-like	Broadcast-like	All sinks	On [CLS]
Baseline	0.25	0.09	0.45	0.34
Gating	0.00	3.80	3.88	2.02
Global + Penalty	0.00	0.04	0.11	0.11
Gating + Global + Penalty	0.03	0.53	0.63	0.08

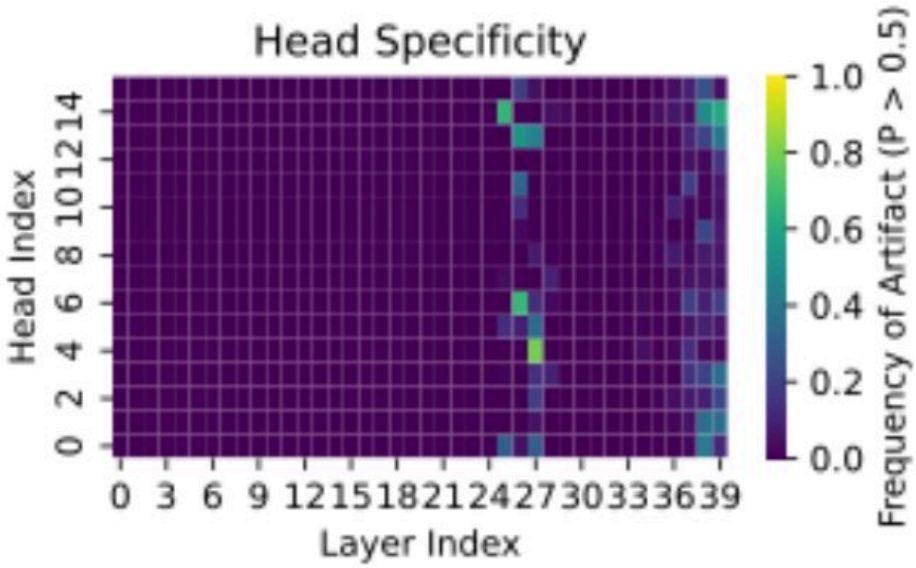


Figure 14: **Head specialization in EVA Giant.** The entropy of values per layer shows that sink behavior is head-specific.

the global pathway alone leaves task error essentially unchanged (0.3098), but does not eliminate attention-side broadcast: L2 still assigns 0.699 attention to $j^\star$. With the common-mode penalty, source-directed attention falls to 0.031 in L1 and 0.008 in L2, and L2 source–sink matching falls to zero. At the same time, the L2 global pooling operation selects the ground-truth source in 77.3% of examples and assigns it mean pooling weight 0.651. The corresponding attention update has RMS 1.6×10^{-5}, compared with 2.553 for the global update. Task MSE remains 0.3648. Thus, providing a global pathway alone creates an alternative broadcast route but does not force the model to abandon attention; the common-mode penalty shifts the shared computation to the global branch.

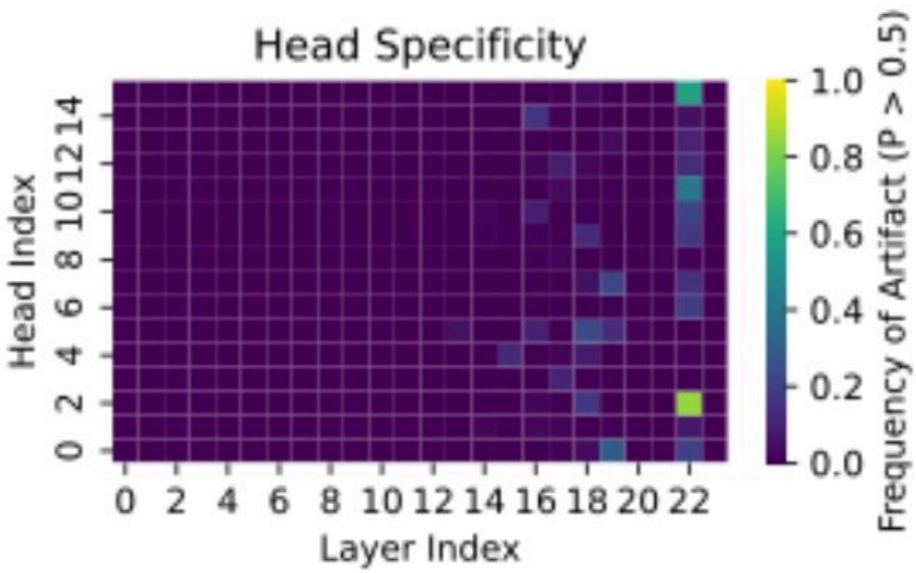


Figure 15: **Head specialization in Clip OpenAI Large.** The entropy of values per layer shows that sink behavior is head-specific.

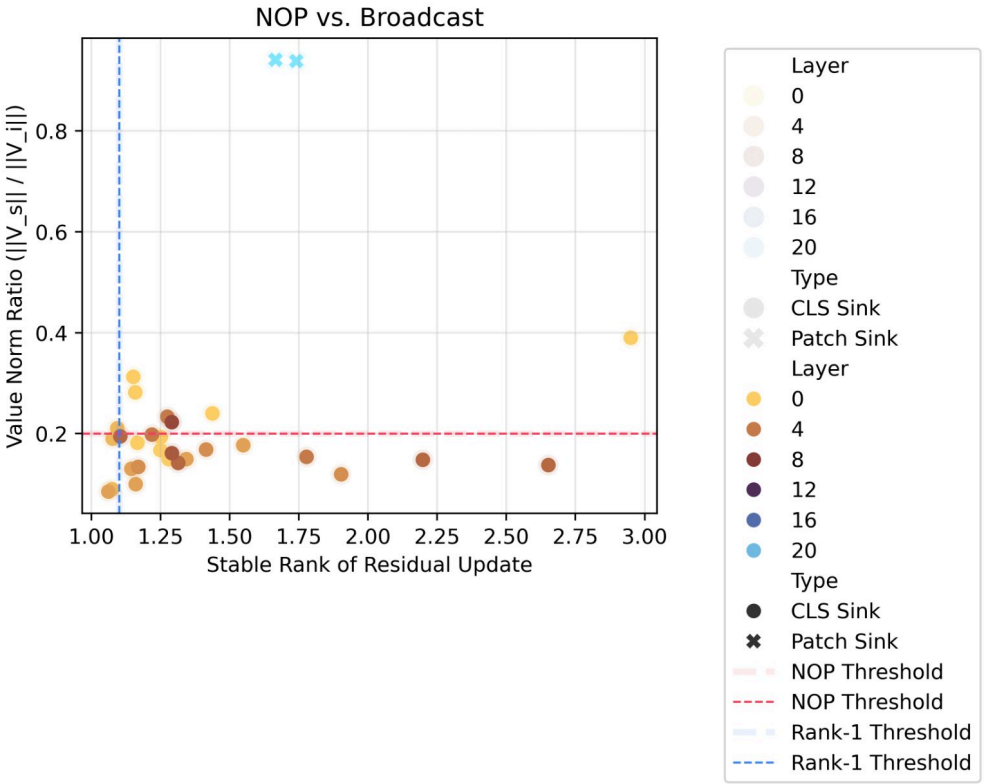


Figure 16: **Dual Phenomenology in Clip OpenAI Large.** Sinks cluster into NOP sinks (low norm, bottom) and broadcast sinks (moderate to high norm, ≈rank-1 update, left). Both regimes coexist within the same model.

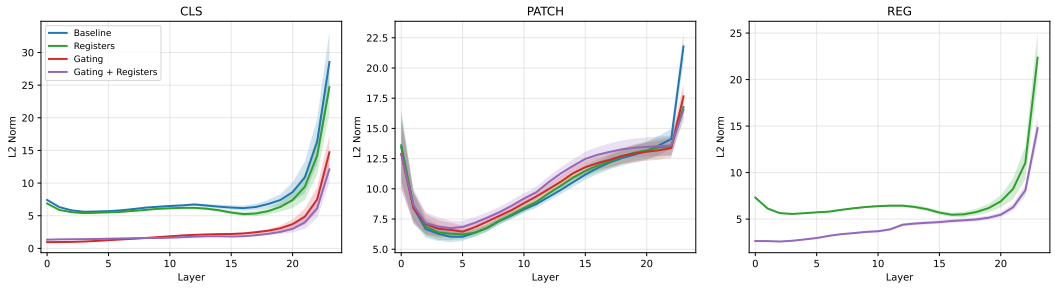


Figure 17: **Gating and registers mitigate high norms.** Distribution of token norms across layers on the ImageNet-1k validation set.

Causal dependence shifts to the global pathway. We next intervene on the two communication branches separately. *Attention ablation* removes the projected attention update while preserving the global update, whereas *global ablation* sets the global update to zero while preserving attention. We additionally replace the global state g^{ℓ} with that of an independently sampled donor sequence and propagate the intervention through the remainder of the network.

Figure 18 shows the resulting causal effects. In the attention-only baseline, removing the L2 attention update strongly disrupts the task. Adding the global pathway without the penalty distributes causal dependence across both branches. After applying the penalty, however, removing the L2 attention update leaves task MSE unchanged at 0.3648, whereas removing the L2 global pathway raises it to 1.6539. The attention branch is therefore no longer necessary for the learned broadcast computation.

Cross-example patching provides a content-specific test of this relocation. Replacing the L2 global state with that of a donor sequence induces an output change aligned with the desired donor-minus-recipient broadcast payload (cosine similarity 0.803). Correspondingly, donor-target MSE falls from 1.636 for the unmodified recipient output to 0.369 after global-state replacement. By contrast, donor patching of the L2 attention sink after penalization has essentially zero alignment with the desired payload. Together, these interventions show that the penalty does not merely remove the visual attention sink: it relocates the causal broadcast channel from softmax attention to the explicit global pathway.

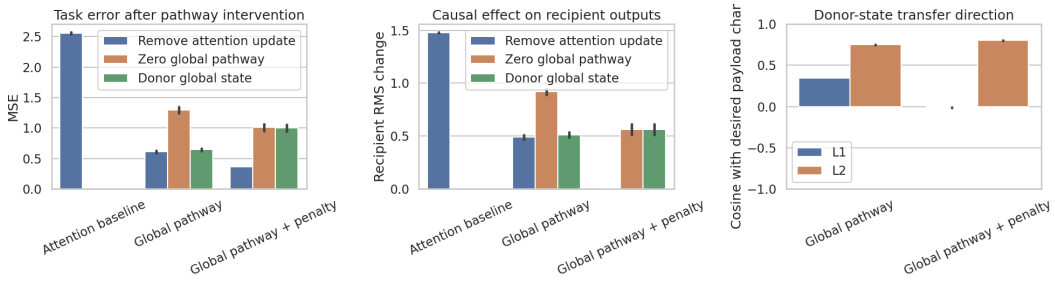


Figure 18: **Causal validation of broadcast relocation to the global pathway. Left:** task error after removing the attention update, ablating the global pathway, or replacing the global state with that of an independent donor. **Center:** RMS change in recipient outputs induced by the same interventions. **Right:** cosine similarity between the output change induced by donor-global-state replacement and the desired donor-minus-recipient broadcast payload, shown separately by layer. Adding the global pathway alone allows both attention and the auxiliary pathway to participate in broadcast. With the common-mode penalty, attention becomes causally dispensable while the L2 global pathway carries the transferable broadcast signal. Error bars denote standard errors over evaluation batches.

E.2 LeJEPA: Penalty Normalization, LayerScale, and Sensitivity to the Sink Rule

Penalty normalization. Eq. 5 normalizes the common-mode energy by the energy of the same attention update, with gradients through both. Normalizing by a stop-gradient energy did not train in LeJEPA. In one attempt we used $\lambda_B = 1$ with the stop-gradient energy of the attention update as the denominator, fine-tuning baseline, gated and pathway checkpoints from epoch 69 (online probe 0.66). The probe fell to 0.002–0.024 within one epoch and reached at most 0.32 over the next 20–25 epochs. In a second attempt we used $\lambda_B = 1$ with the stop-gradient energy of the whole block update as the denominator, training a pathway model from scratch. It stayed at chance (probe 0.0011 after one epoch, vs. 0.018 for the baseline). A stop-gradient denominator leaves only the gradient of the numerator, which the model can lower by shrinking the attention branch. The ratio in Eq. 5 is unchanged by such rescaling, and both penalized models of Table 1 train with it. Both stop-gradient runs also used $\lambda_B = 1$ rather than 0.1.

LayerScale initialization. Baseline and Gating use LayerScale initialized at 10^{-4} on the attention and MLP branches, as in the sweep; the two penalized models use 1.0 (Table 10). At 10^{-4} the penalty acts from the first step on an attention branch scaled down by 10^4. In pilot runs it was then reduced largely through the learned LayerScale weights rather than through a change in attention. Baseline and gating-only models at LayerScale 1.0 were not trained. Rows 1–2 and 3–4 of Table 1 therefore differ in initialization as well as in the intervention. Gating + Global + Penalty (at 1.0) matches or exceeds Baseline (at 10^{-4}) on all three probes, so the higher initialization does not by itself prevent baseline-level probe accuracy.

Sensitivity to the sink rule. Definition 1 requires every query to place at least $1 - \epsilon$ of its attention on the sink token, whereas Table 14 uses a relaxed rule. Table 15 recounts the same models and images under the literal definition for $\epsilon \in \{0.5, 0.3, 0.2, 0.1\}$, using the minimum over all queries of the attention to the dominant key token. For broadcast-like sinks the ordering Gating $>$ Gating + Global + Penalty $>$ Baseline $\geq$ Global + Penalty holds at every ϵ. Under the literal definition the Baseline has at most 0.02 NOP-like sinks per image. The reduction of NOP-like sinks by gating reported in Section 5 is therefore visible only under the relaxed rule.

Table 15: Broadcast-like / NOP-like sinks per image for the models of Table 1 under the relaxed rule of Table 14 and under Definition 1 at several ϵ (every query places at least $1 - \epsilon$ on the sink token). Same checkpoints, images, layers and value-norm cutoffs as Table 14.

	Relaxed	**Definition 1**			
Variant		$\epsilon = 0.5$	$\epsilon = 0.3$	$\epsilon = 0.2$	$\epsilon = 0.1$
Baseline	0.09 / 0.25	0.02 / 0.02	0.01 / 0.00	0.00 / 0.00	0.00 / 0.00
Gating	3.80 / 0.00	1.56 / 0.00	1.29 / 0.00	1.18 / 0.00	1.05 / 0.00
Global + Penalty	0.04 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Gating + Global + Penalty	0.53 / 0.03	0.31 / 0.00	0.17 / 0.00	0.11 / 0.00	0.07 / 0.00

F Extended Literature Review

F.1 Gated Attention

We briefly describe the *gated attention* modification of softmax attention Qiu et al. (2025) used throughout this paper. Gated attention introduces an additional *gate* computed from a (typically pre-normalized) hidden-state signal and uses it to modulate one of the intermediate quantities in the attention block.

Default gated-softmax output. Unless stated otherwise, we use *head-specific, multiplicative sigmoid gating* applied to the SDPA (softmax-attention) output. Starting from the standard single-head formulation in Eq. 1, let $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ denote the matrix whose i-th row is $\boldsymbol{x}_i^\top$, and let $\boldsymbol{O} = \boldsymbol{A}\boldsymbol{V}$ denote the ungated head output. We form the gated head output

$$\widetilde{\boldsymbol{O}} \;=\; \boldsymbol{O} \odot \boldsymbol{G} \;=\; (\boldsymbol{A}\boldsymbol{V}) \odot \sigma(\boldsymbol{X}\boldsymbol{W}_\theta), \tag{50}$$

where $\boldsymbol{G} = \sigma(\boldsymbol{X}\boldsymbol{W}_\theta) \in \mathbb{R}^{n \times d_h}$, $\boldsymbol{W}_\theta$ are learnable parameters, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes elementwise multiplication. This replaces the ungated head output in the attention block. Further implementation details are provided in Appendix D.

In our framing, output gating provides a direct mechanism for conditional suppression, allowing a head to implement an approximate NOP without routing attention to a null token. It does not, however, provide a separate route for shared communication, which motivates our complementary global pathway for broadcast mitigation.

The interpretation of concentrated attention on a special token as a null operation predates the attention-sink literature: Clark et al. (2019) observed that a large fraction of BERT heads attend almost exclusively to `[SEP]` and hypothesized that this functions as a "no-op" invoked when the head's operation does not apply to the current token. Our adaptive-NOP hypothesis makes this reading identifiable, by pairing it with an explicit value-norm condition and testing it causally rather than inferring it from the attention map alone. Gating is one of several architectural responses to the same observation; alternatives include supplying an explicit null destination Xiao et al. (2024) or relaxing the row-stochasticity constraint outright (Appendix F.3).

F.2 Register Tokens and Global Communication

We also consider the *register token* modification of a ViT Darcet et al. (2024), which appends a small number of additional learnable tokens to the input sequence. Concretely, let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ denote the patch-token representations (and optionally a class token, treated as an additional special token in the same way). We introduce r register tokens with learnable embeddings $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_r \in \mathbb{R}^d$ that are independent of the input image and are appended to the sequence after patch embedding. Attention is then computed without any other change on the extended sequence, and the same residual updates are applied to both the original and register tokens. At the end of the network, the register tokens are *discarded*: downstream representations use only the original tokens (e.g., patch tokens and optionally `[CLS]`), while registers serve purely as internal workspace during the forward pass. Further implementation details are provided in Appendix D.

Registers therefore provide dedicated workspace for global computation, but they remain sequence positions and can themselves attract concentrated attention. Consistent with this distinction, our analysis finds that register-equipped ViTs preserve both sink regimes: register sinks can exhibit either suppressed value norms consistent with adaptive NOP or content-scale values and approximately low-rank updates consistent with broadcast. Thus, registers can relocate sink computation without eliminating the single-position routing pattern.

Appended tokens as workspace. Registers belong to a longer line of work that augments the sequence with input-independent slots. Sukhbaatar et al. (2019) add persistent memory vectors inside the attention operation, and Burtsev et al. (2020) prepend dedicated `[mem]` tokens to the input; in the autoregressive setting, Xiao et al. (2024) show that reserving a single learnable sink token both stabilizes streaming inference and absorbs the attention mass that would otherwise fall on the first position. These constructions differ in where the slot lives and how it is trained, but they share the property we highlight here: because the slot occupies a sequence position, it can be selected by the same routing mechanism as any other token, and therefore inherits rather than eliminates the single-position bottleneck. Our diagnostics show that this is what happens in practice, with registers being used for both adaptive NOP and broadcast.

More broadly, explicit global-context mechanisms have long been used to aggregate information across spatial positions and return it to local representations, including non-local, global-context and squeeze-and-excitation modules Wang et al. (2018); Cao et al. (2019); Hu et al. (2018). The permutation-invariant pooling of our global pathway is likewise related to attention-based set pooling Lee et al. (2019) and to architectures that route global information through a latent array held outside the token sequence Jaegle et al. (2021); class-attention layers Touvron et al. (2021) pursue a related separation by decoupling the class-token readout from patch-to-patch mixing. Our global pathway is related in spirit, but is introduced specifically as a position-free alternative to broadcast through attention. Crucially, providing such a pathway alone does not determine where the broadcast computation is carried; we therefore pair it with the common-mode penalty in Eq. 5, which discourages attention from carrying the shared update while leaving attention concentration itself unpenalized.

F.3 Non-softmax Attention

A complementary line of work modifies or replaces the softmax normalization itself. These approaches alter how attention weights are formed, rather than introducing an additional communication pathway.

1. Clipped-softmax attention. Clipped-softmax Bondarenko et al. (2023) replaces the usual row-wise softmax with a stretched-and-clipped variant. Given standard softmax weights $\tilde{\boldsymbol{A}} = \mathrm{softmax}(\boldsymbol{S})$ and hyperparameters $\zeta \geq 1$, $\gamma \leq 0$, it defines

$$\boldsymbol{A}_{\text{clip}} = \mathrm{clip}\left((\zeta - \gamma)\,\tilde{\boldsymbol{A}} + \gamma,\ 0, 1\right). \tag{51}$$

Softmax outputs are stretched and clipped back to $[0, 1]$, allowing exact zeros and ones with finite score ranges. This gives attention heads a direct way to assign zero mass to all positions, alleviating the need for a dedicated null destination Bondarenko et al. (2023).

2. ReLU attention. ReLU attention Wortsman et al. (2023) replaces softmax with a pointwise ReLU and sequence-length scaling:

$$\boldsymbol{A}_{\text{ReLU}} = \frac{1}{n}\,\mathrm{ReLU}(\boldsymbol{S}). \tag{52}$$

Rows are no longer normalized to sum to one, and entries can be exactly zero. The factor $1/n$ controls the scale of the attention output as sequence length changes. More generally, point-wise attention families replace softmax by a generic elementwise nonlinearity h together with length-dependent scaling, $\boldsymbol{A}_{h,\alpha} = n^{-\alpha} h(\boldsymbol{S})$ Saratchandran et al. (2024).

3. Polynomial attention. Polynomial attention Saratchandran et al. (2024) similarly replaces softmax with an elementwise polynomial of the scores. A representative form is

$$\boldsymbol{A}_{\text{poly}} = \sqrt{\frac{1}{n}}\,\boldsymbol{S}^{\odot p}, \tag{53}$$

where $\boldsymbol{S}^{\odot p}$ denotes the elementwise power. The sequence-length scaling is chosen to control the magnitude of the attention matrix and its Jacobian while dropping the non-negativity and row-normalization constraints imposed by softmax.

4. Sigmoid self-attention. Sigmoid attention Ramapuram et al. (2025) replaces row-wise softmax by an elementwise sigmoid,

$$\boldsymbol{A}_{\text{sig}} = \sigma(\boldsymbol{S} + b), \tag{54}$$

where b is a learnable or fixed bias that may depend on sequence length. Rows are not normalized and entries lie strictly in $(0, 1)$; suitable choices of b and normalization can keep attention magnitudes controlled. This use of a sigmoid to form *attention weights* is distinct from the sigmoid output gating in Appendix F.1, which leaves the softmax attention weights unchanged and instead modulates the resulting attention output.

5. Constraining the logits rather than the normalizer. A related family keeps softmax but controls the scale of the pre-softmax scores. Zhai et al. (2023) show that low attention entropy accompanies training instability and propose σReparam, which bounds the spectral norm of the query–key parameters. This is a direct constraint on the spectral norm $\|\Theta\|_2$ that governs the sink logit magnitude in our analysis, and it makes the trade-off in our framing explicit: the spectral spike that implements a reliable NOP gate is the same quantity whose growth these methods suppress for stability. Methods that instead target the activation side of the same product, by minimizing outlier features during training He et al. (2024), constrain the complementary "massive activation" regime.

These methods therefore modify the weighting or normalization rule within attention itself. Our global-pathway intervention instead retains standard softmax self-attention and introduces a separate route for shared communication. Moreover, the global pathway is not itself "softmax-free": its permutation-invariant pooling operation uses softmax weights. The distinction is architectural rather than a replacement of the attention normalization.

F.4 Other Related Works

Geometric and optimization perspectives. Beyond the sink literature summarized in Section 2.2, related work connects transformer dynamics to spectral concentration, outlier activations, dormant heads, rank collapse, and oversmoothing Queipo-de Llano et al. (2026); Cancedda (2024); Sun et al. (2024); Sandoval-Segura et al. (2026); Jacobs et al. (2026); Nait Saada et al. (2025); Wang et al. (2022). Rank collapse in particular has a well-developed theory: repeated self-attention drives token representations toward a rank-one subspace at a doubly exponential rate absent skip connections and MLPs Dong et al. (2021), an effect that can be characterized through signal propagation at initialization Noci et al. (2022), and whose continuous-time limit provably clusters tokens onto a small number of "leader" positions Geshkovski et al. (2023). These phenomena are closely related to, but distinct from, the mechanisms studied here. In particular, a low-rank *attention update* caused by broadcast does not by itself imply collapse of the token representations: as shown in Lemma 4, a perfectly shared additive update preserves centered token variance. Likewise, large activations or a dominant query–key direction can support adaptive routing to a NOP sink without themselves defining the NOP computation. We also note that the head dimension imposes its own ceiling on the rank a single head can write Bhojanapalli et al. (2020); the stable rank ≈ 1 we measure in Section 3.2 is well below this ceiling and is therefore a property of the learned solution rather than of the architecture.

Attention weights as evidence about computation. Our starting point, that an attention map underdetermines the underlying algorithm, is shared with a literature on whether attention constitutes explanation Jain & Wallace (2019); Wiegreffe & Pinter (2019); Akula & Zhu (2022); Bibal et al. (2022). Most relevant to our diagnostics, Kobayashi et al. (2020) argue that attention weights should not be read in isolation and that the informative quantity is the norm-weighted contribution $\|\boldsymbol{A}_{ij}\boldsymbol{v}_j\|$, which reveals that heads attending heavily to special tokens often transmit almost nothing. Our value-norm diagnostic is the same measurement placed in a mechanistic frame: we show that a near-null value is not merely correlated with a null update but is the unique way to realize one under a perfect sink, and our interventions test that claim causally.

Head specialization and circuits. The sparse, head-specific structure we report in Section 4 is consistent with evidence that attention heads specialize and that many can be pruned with little loss Voita et al. (2019); Michel et al. (2019), and with mechanistic accounts of individual heads implementing identifiable algorithms Elhage et al. (2021). Dormant-head analyses in LLMs Sandoval-Segura et al. (2026) describe a closely related regime; our contribution is to separate, within the class of sink heads, those that are conditionally inactive from those that are actively broadcasting.

Causal interventions. Our intervention protocol follows standard practice in mechanistic interpretability, where activation patching and its variants are used to localize the components responsible for a behavior Meng et al. (2022); Wang et al. (2023). We follow the recommendations of Zhang & Nanda (2024) in separating interventions on the payload carried at a destination from interventions on the routing itself, and in reporting a donor-patched target alongside raw output change, so that a large perturbation is not mistaken for transfer of the hypothesized content.

Mitigation strategies beyond register tokens and gated attention. Prior work has also explored modifying the attention mechanism to reduce pathological concentration or make null updates easier to express, including clipping or rescaling attention logits, reserving an explicit sink position, and replacing softmax with alternatives that permit exact or near-exact zero weights Bondarenko et al. (2023); Wortsman et al. (2023); Ramapuram et al. (2025); Xiao et al. (2024). These approaches primarily alter how attention mass is allocated. Our broadcast intervention instead leaves standard softmax attention intact and discourages only the *shared component* carried by the attention update, while providing a separate global pathway to preserve that computation.

AI USE STATEMENT

In this work, we used generative AI tools for designing or providing feedback on research methodology or experiments, implementing methods, generating synthetic data sets, assisting in the writing of proofs, and for interpreting results. We have not used generative AI tools for cleaning and reformating datasets or proposing and refining hypotheses, help develop theoretical models or conceptual frameworks, formulate mathematical claims, assist with translation, provide critical ingredients for proving mathematical claims, or support qualitative and thematic data analysis. Additionally, we used generative AI tools for creating or modifying scientific figures or images, suggesting experimental parameters, creating or editing software code, and editing a research paper to improve readability. We have reviewed all AI-assisted work. LLM-generated code and LLM-prepared datasets were verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.