

# Multiple Recovery Paths in a Lesioned Transformer: Internal Rerouting and Readout Remapping

Lorenzo Cannata<sup>1</sup>      Uri Hasson<sup>2</sup>

<sup>1</sup>Department of Information Engineering and Computer Science (DISI), University of Trento

<sup>2</sup>Center for Mind/Brain Sciences (CIMEC), University of Trento  
Trento, Italy

{lorenzo.cannata@studenti.unitn.it; uri.hasson@unitn.it}

## Abstract

Transformer circuits are often evaluated by ablating certain circuits or components and measuring the resulting performance drop. These interventions test circuit necessity only under a fixed model configuration, and it is unclear if the residual network can recover performance through alternative, parameter-efficient adaptations. Using BERT as a frozen encoder and a linear part-of-speech (POS) readout as a task classifier, we first use gate masking to learn a sparse core subnetwork sufficient for POS classification. Ablating this core produced a damaged model and a sharp drop in accuracy (95.0%  $\rightarrow$  16.8%). We then evaluated recovery from damage under four retraining pathways: readout-only, mask-only, joint readout+mask, and encoder-only with the ablation held fixed. Readout-only, mask-only, joint mask-and-readout, and encoder-only retraining arrived at 80.0%, 69.7%, 85.9%, and 92.3% accuracy, respectively. Some recovery pathways partially restored core-like geometry, but others recovered high performance without restoring the original encoder geometry. These results show that in a joint encoder-readout context, a sparse subnetwork can be sufficient and necessary for performance in the original fixed configuration, but need not remain necessary after recovery, and that recovered performance does not rely on recovering the original geometry.

## 1 Introduction

Transformer models are increasingly studied in NeuroAI (Sadeh and Clopath, 2025) and cognitive science because their internal representations can predict human representational structure and brain activity. Model geometry has been related to human judgments and behavioral measures (e.g., Schrimpf et al., 2018; Bhatia and Richie, 2024), and contextual embeddings have been aligned with brain responses during natural language, including shared geometric structure and predictive processing (Goldstein et al., 2024, 2022). Furthermore,

recent work uses attribution mapping to show that the information most relevant to minimizing the model’s loss is relevant to predicting brain activity (Rahimi et al., 2025). Taken together, these studies suggest that at least at the representational level there are partial similarities in functional organization (while recognizing that human and model objectives differ).

Inspired by studies of in silico lesioning in vision systems (e.g., Tuladhar et al., 2021; Moore et al., 2022, 2023), we use a lesion-and-recovery analogy to study how pretrained language Transformers regain task performance after targeted structural damage. We first identify a sparse network that is sufficient for high performance on a POS categorization task. We then ablate this subnetwork to produce a large drop in performance. Our main question is whether recovery after ablating this sparse subnetwork can be achieved by (1) reselecting surviving projection-specific gates or (2) remapping a learned readout, or both. We identify the core subnetwork by jointly training projection-specific input-feature hard-concrete gates and a linear POS readout head on a frozen BERT encoder (Louizos et al., 2018; Cao et al., 2021). We then permanently clamp the encoder’s high-activity gated components, defined as selected Q/K/V head-sized mask groups, as well as additionally selected dense projection masks. From this common damaged state, we compare several retraining options (Figure 1); full procedural details are given in Methods.

Our contributions are as follows:

- We develop a lesion–recovery framework for pretrained Transformers. We first identify a sparse subnetwork sufficient for POS classification, ablate its components, and then test multiple recovery paths: mask rerouting, readout retraining, joint adaptation, and encoder retraining.
- We introduce mask rerouting, which is a re-

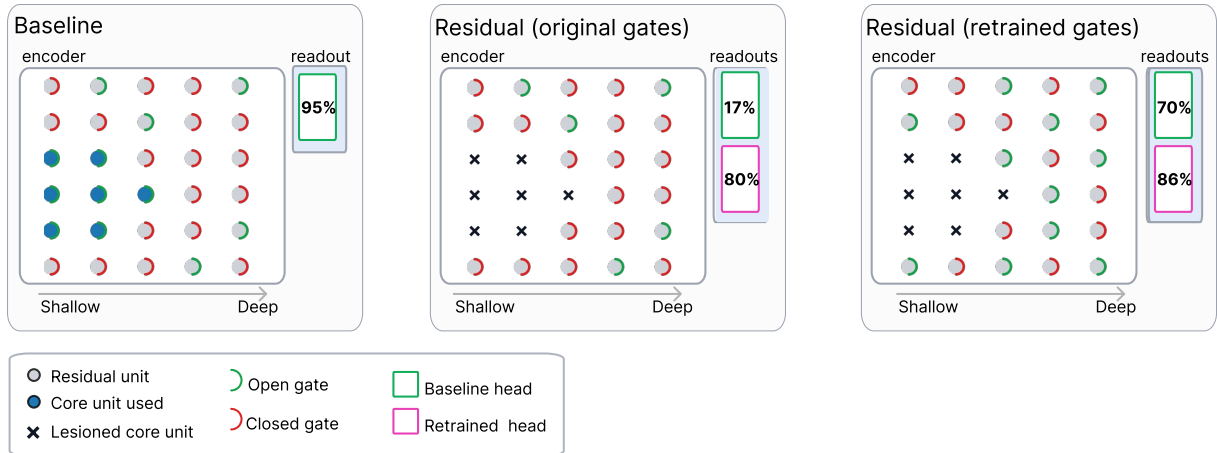


Figure 1: **Masking, lesioning, and recovery.** A sparse POS-sufficient core is learned by training gates and a task head while keeping BERT weights fixed. Ablating the core reduces held-out accuracy from 95% to 17%. Recovery is then tested by retraining the readout only (80%), the surviving gates only (70%), or both jointly (86%). Circles denote units, half-arcs indicate gate state (green=open, red=closed), and  $\times$  marks lesioned units. Changes in gate color after retraining indicate that surviving units have been opened or closed, producing a new routing configuration. Percentages in the head boxes report held-out POS accuracy.

covery mechanism that operates by changing which surviving gated components are active, but without modifying the pretrained encoder weights. This allows us to differentiate recovery by route selection from recovery by weight relearning.

- We show that both mask rerouting and readout remapping recover performance after lesioning, and that combining them produces stronger recovery than either mechanism alone.
- We show that recovery of performance does not require recovery of the original representation. Different retraining routes produce different hidden-state and readout geometries, even when their behavioral performance is similar.

## 2 Related Work

**Redundancy and sparse subnetworks.** Prior studies found that attention heads are often specialized for different linguistic functions and that a substantial fraction can be removed with limited performance loss. Voita et al. (2019) report head-level specialization and successful pruning of less useful heads; Michel et al. (2019) find that dropping 20–40% of heads can still support accuracy on several tasks. In parallel, pretrained Transformers contain sparse subnetworks that can match dense-network performance on sentence-level bench-

marks (e.g., Chen et al., 2020). Whitaker (2024) formalizes this view under the concept of Subnetwork Ensembles, showing that multiple sparse subnetworks sampled from a trained parent model can match or exceed full-retraining performance at a fraction of the compute cost. Based on these studies, we expect that a subset of the pretrained BERT encoder can sustain POS classification.

**Gate-based importance estimation and head lesioning.** Louizos et al. (2018) introduce multiplicative gates with learnable scores that, when trained alongside a task readout, produce task-sufficient masks and an importance score. Cao et al. (2021) apply this approach to LLMs to induce task-specific sparsity and show that different Transformer components support specific linguistic tasks. We adopt this framework as follows: we learn projection-specific input-feature gates, summarize Q/K/V gate sparsity within head-sized groups, and apply a structured projection-mask lesion to the Q/K/V groups and dense projection masks that exhibit non-sparse gate patterns.

**Pruning and circuit discovery.** Circuit-discovery methods in Transformers identify computational subgraphs that preserve a target behavior, using methods such as activation patching, attribution patching, or learned edge-pruning objectives. In the typical prompt-based settings, the behavior is quantified using the model’s native output distribution, for example with a

next-token probability or logit-difference metric, and the identified subgraph is treated as a circuit supporting that behavior (e.g., Hanna et al., 2024; Yu et al., 2025; Bhaskar et al., 2024; Syed et al., 2024; Conmy et al., 2023).

In contrast, we study a supervised, non-native task in which POS behavior is obtained by learning a new task readout that is attached to a pretrained Transformer encoder. The resulting behavior is therefore a joint function of the encoder representation and the readout applied to it. This makes circuit identification more underdetermined than in a fixed native-output setting, as similar task behavior can be achieved by different combinations of hidden-state and readout geometries.

The direction of our intervention also differs. Circuit-discovery methods prune or mask components that are scored as less important with respect to the target behavior. The aim is to retain a sparse subgraph that performs similarly (is faithful) to the original model. We proceed in the opposite direction by first identifying a task-sufficient sparse core in order to ablate the high-importance components it uses.

Our damaged condition is related to circuit-completeness tests. These evaluate the behavior of the complement after an identified circuit is removed (e.g., Conmy et al., 2023; Syed et al., 2024; Bhaskar et al., 2024). Standard completeness tests evaluate the complement in a fixed state. Our recovery experiments allow the complement to adapt and in this way ask whether initially low complement performance persists after retraining.

### 3 Methods

Figure 1 summarizes the analyses used in the study: learning a sparse, task-sufficient core network, targeted lesion of the network, recovery under multiple constraints, and behavioral and representational evaluation.

#### 3.1 Task and model

**Dataset and POS objective.** We evaluated all conditions on universal part-of-speech tagging using the training and development splits of the Universal Dependencies English Web Treebank (UD English-EWT). The prediction targets were the 17 POS categories: ADJ, ADP, ADV, AUX, CCONJ, DET, INTJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB, and X. Sentences were tokenized using bert-base-uncased. For each UD word,

we considered the final-layer hidden state associated with its last WordPiece token as the word representation. The linear POS classifier was applied by token. We trained the classifier on the official training split. We evaluated performance on the official development split, which we used here as a held-out evaluation set. Metrics were computed over syntactic UD word tokens, including tokens labeled PUNCT, but excluded BERT special tokens, multiword-token range entries, and empty nodes.

#### 3.2 Learning and lesioning a sparse POS-sufficient core

Following Cao et al. (2021), we trained a task-sufficient subnetwork by optimizing hard-concrete mask gates jointly with a linear POS readout. The pretrained BERT encoder weights remained fixed during training. The joint loss combined POS cross-entropy with an expected- $L_0$  sparsity penalty on the mask gates. The sparsity coefficient was linearly scheduled during training. For the baseline core model, masks and the POS readout were trained for 20 epochs with Adam.

If  $W$  is BERT’s pretrained weight matrix, then the masked weight matrix is written as  $\tilde{W} = Z \odot W$ , where  $Z$  is a stochastic hard-concrete gate matrix parameterized by learnable gate scores  $\theta$ . Only the gate parameters  $\theta$  and the linear POS readout  $W_{\text{POS}}$  are trained.

We used a cross-entropy objective with an  $\ell_0$ -style gate penalty:

$$\mathcal{L}(\theta, W_{\text{POS}}) = \frac{1}{N} \sum_{n=1}^N \text{CE}(f_{\text{POS}}(h_n(\theta)), y_n) + \lambda \Omega(\theta). \quad (1)$$

The end result is a sparse set of gates that is sufficient for POS classification when paired with the jointly trained POS readout, while leaving the underlying BERT weights unchanged.

We then applied a projection-mask lesion to the learned core. The learned masks were projection-specific input-feature gates. Each gate controlled a single input dimension for a particular linear projection. For the query, key, and value projections, masks were grouped into 64-dimensional head-sized groups. We computed group-level sparsity by averaging the fraction of inactive gates across the corresponding query, key, and value groups. Groups with sparsity *below*  $\tau = 0.5$  (i.e., ones with high gate activity) were lesioned. This value

| Condition          | Updated parameters |       |         |                | Functional test                                                       |
|--------------------|--------------------|-------|---------|----------------|-----------------------------------------------------------------------|
|                    | BERT               | Gates | Readout | Gate objective |                                                                       |
| Core baseline      | –                  | ✓     | ✓       | Sparse         | Learn a sparse POS-sufficient subnetwork                              |
| Damaged            | –                  | –     | –       | –              | Evaluate the fixed residual network without adaptation                |
| Mask only          | –                  | ✓     | –       | None           | Unconstrained rerouting through surviving gates                       |
| Mask only (sparse) | –                  | ✓     | –       | Sparse         | Rerouting constrained to use fewer gates                              |
| Mask only (dense)  | –                  | ✓     | –       | Dense          | Rerouting constrained to use more gates                               |
| Encoder only       | ✓                  | –     | –       | –              | Relearn the surviving encoder weights under fixed routing and readout |
| Readout only       | –                  | –     | ✓       | –              | Relearn POS classes given the fixed damaged representation            |
| Readout + mask     | –                  | ✓     | ✓       | Sparse         | Combine rerouting with readout remapping                              |

Table 1: Training and recovery conditions. All rows except the core baseline are evaluated after the same fixed lesion. Check marks indicate parameters updated during the corresponding training stage; dashes indicate frozen parameters. The gate objective refers to the coefficient  $\lambda$  in Eq. (1). For the mask-only variants,  $\lambda_{\text{final}} = 0, +1$ , and  $-1$  indicate the unregularized, sparsity-constraining, and density-constraining optimization.

was set a priori and was not selected through a hyperparameter sweep.

The lesion also included attention-output, FFN-intermediate, and FFN-output projection masks whose produced-mask sparsity was below the same threshold. The mask scores of lesioned components were then clamped to  $-12.0$ , producing gates of exactly zero, and frozen to prevent them from further updating during recovery. Note that not all closed gates were lesioned, so closed gates that survived the ablation could be reopened during mask retraining. Lesioned gates were clamped and could not be reactivated. All recovery conditions operated under this same fixed structured projection-mask lesion. The full hard-concrete formulation and lesion-selection details are provided in Appendix A.

### 3.3 Recovery conditions

Table 1 summarizes the recovery conditions. Readout-only tests whether the damaged representation contains decodable POS information. The three mask-only conditions evaluate rerouting through surviving gates under neutral, sparse, and dense regularization, without changing encoder weights (see Eq 1 and Appendix B). Joint recovery (Readout + mask) allows both rerouting and readout remapping. Encoder-only retraining can be considered as an upper-capacity comparison, as it allows updating BERT weights that survived the lesion.

To evaluate reproducibility, recovery was evaluated using four additional mask-initialization seeds, where each corresponded to learning a new sparse core and subsequent lesioning. These additional runs were used only for analyzing behavioral per-

formance. All other analyses were based on first-seed run.

### 3.4 Performance and representational evaluation

We report token-level POS accuracy on the held-out data. All representational analyses use the same ordered development tokens across the different experimental conditions (trained model states).

**Global hidden-state similarity.** For each model state  $s$ , let  $X_s \in \mathbb{R}^{N \times 768}$  denote the matrix of final-layer hidden states. We computed linear CKA between  $X_s$  and both the pre-lesion core representation  $X_{\text{core}}$  and the immediately damaged representation  $X_{\text{damage}}$ .

**POS activation-centroid geometry.** At the encoder level, we described each POS class by constructing a centroid in activation space. For a model state  $s$  and POS tag  $t$ , we averaged the final-layer hidden states of all tokens assigned to  $t$  to produce a centroid  $\mu_{s,t} \in \mathbb{R}^{768}$ . Using these centroids, we constructed a POS-centroid representational dissimilarity matrix (RDM) from the cosine distances between centroids:

$$R_s^{\text{act}}(t, t') = 1 - \cos(\mu_{s,t}, \mu_{s,t'}). \quad (2)$$

The resulting RDM describes the geometry of the POS centroids. We quantified recovery of the pre-lesion activation geometry by comparing it with the matching activation-centroid RDM of the core model:

$$\rho_{\text{POS}}(s) = \text{Spearman}[\text{ut}(R_s^{\text{act}}), \text{ut}(R_{\text{core}}^{\text{act}})], \quad (3)$$

where  $\text{ut}(\cdot)$  vectorizes the off-diagonal entries of the upper triangle.

**POS readout-weight geometry.** We also described the POS classes using the geometry of the linear readout. The linear POS readout is parameterized by  $W_{\text{POS}} \in \mathbb{R}^{17 \times 768}$  and a bias vector. For this reason, each of the 17 POS classes is associated with a 768-dimensional weight vector  $w_{s,t}$ . Each vector defines a direction in hidden-state space along which the logit for a given class increases.

These class-specific directions can be treated as prototypes in readout-weight space, complementary to the empirical POS centroids in activation space. We constructed the readout-prototype RDM as

$$R_s^{\text{head}}(t, t') = 1 - \cos(w_{s,t}, w_{s,t'}). \quad (4)$$

This RDM captures the relative organization of the POS-class directions learned by the classifier. We compare it with the corresponding readout-weight RDM of the pre-lesion core.

**Activation scale and effective dimensionality.** Activation energy is the mean squared  $\ell_2$  norm of the final-layer hidden states. Effective dimensionality is the entropy effective rank of their centered covariance matrix, defined for covariance eigenvalues  $\{\lambda_k\}$  as  $p_k = \lambda_k / \sum_j \lambda_j$  and  $D_{\text{eff}} = \exp\left(-\sum_{k:p_k>0} p_k \log p_k\right)$ .

## 4 Results

### 4.1 Lesion and recovery performance

The learned core network consisted mainly of early-to-middle-layer components. Its accuracy was 94.5%. Removing it completely (by removing the 74 QKV head groups and 18 dense projection mask tensors) reduced accuracy to 17% (*Damaged* configuration). Appendix Figure A1 shows the configuration of the Baseline network, and the result of applying the ablation.

To evaluate the importance of the QKV components identified as part of the core network, we tested whether they were more essential than an arbitrary set of size-matched QKV components (note that here the dense projections were not removed). We compared the impact of a targeted QKV-only lesion with a distribution of 100 globally sampled random lesions, each removing 74 QKV groups. The targeted lesion reduced accuracy to 84.84%. The random lesions produced

a distribution with a mean accuracy of 88.02% (range: 86.03%–89.78%). Thus, no random lesion matched or exceeded the impairment produced by the targeted lesion. The learned QKV set therefore contributed more to task accuracy than would be expected from chance under size-matched ablation. Note that this is a specific Q/K/V-only control, which does not include the dense projections.

Results for the different recovery regimes in the primary seeded run are shown in Figure 2. It can be seen that all recovery conditions produced marked improvement over the damaged baseline. This includes the three mask-only retraining regimes and the readout-only retraining condition, which arrived at 80% accuracy. In all cases, solutions involved keeping between 20% and 40% of the gates in the damaged network. We also reran the full behavioral recovery analysis using four independent seeds. This reproduced the rank order of the conditions, with the regularized mask conditions showing relatively weaker consistency (see Appendix Table A1).

Figure 3 shows the accuracy profiles per POS class under the different retraining regimes and reference conditions. Ablation of the core network resulted in a skewed failure profile, with the damaged network showing preserved accuracy mainly for nouns. This pattern was quite consistent when analyzing the results of the four additional seeds, where noun accuracy was 60%, 99%, 100% and in a single exception, 7%. This shows that after damage, the existing readout is biased towards the frequent noun class.

The recovery conditions based on mask retraining were associated with partial performance recovery for frequent and mid-frequency classes, with the dense mask condition being the best-performing among the mask-only regimes. Training the readout adapter alone also produced recovery across multiple classes, indicating that the overall performance improvement seen in Figure 2 was not based on relearning only the most frequent classes. Indeed, the difference between this condition and the damaged condition is evident across nearly all classes. Finally, readout+mask retraining was associated with an even more uniform improvement across the different tag classes.

### 4.2 Representational recovery in the retraining conditions

Representational analysis showed that behavioral recovery was not necessarily accompanied by

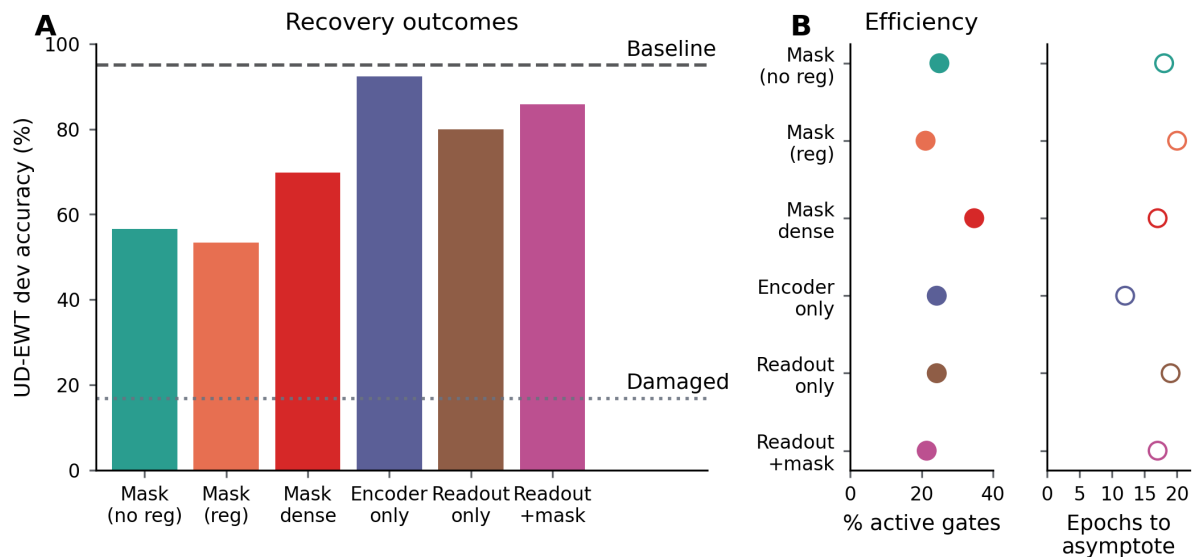


Figure 2: **Classification performance and sparsity in the tested conditions.** **A.** Classification performance across retraining regimes. Baseline indicates performance of core network alone. Damaged indicates performance after ablation of core network. Readout conditions are those where the POS classification is trained as well. **B.** Sparsity of retraining solutions and length of training. Sparsity is quantified by percentage of active gates retained after retraining.

restoration of the original hidden-state geometry. Figure 4A shows, for each condition that impacted the representational geometry of the encoder, the similarity of the representation to that of the damaged network (the departure point of training) and the similarity of the representation to that of the core baseline network.

The figure shows that the initial ablation that produced the damaged configuration also produced a substantial representational shift, seen in that linear CKA between the damaged model and the pre-lesion core was only 0.43. Furthermore, as shown in Figure 4B, the organization of class separation also shifted strongly, so that an RDM constructed from centroids of each POS category was quite different from the matching RDM computed from the core network (POS-centroid RDM similarity to the core was 0.33).

The performance of the readout-only retraining condition offers the strongest demonstration that recovered accuracy does not require relearning the original representation. This condition, which by construction does not alter the encoder representation, recovered POS accuracy to 80.0%. At the same time, the readout retraining shifted the readout-weight representation so that the similarity of the class-weight-vector geometry to that of the original geometry (POS-head to core; panel C) was only 0.54. This shows that a new readout solution

was effective, without recovering the original encoder geometry. The improvement for readout-only retraining does not necessarily mean that the damaged encoder contains POS-information obtained during pretraining. In fact, [Cao et al. \(2021\)](#) report 69% POS accuracy for an MLP probe trained on a random-weight BERT model, which is likely driven by a systematic mapping between single lexical items and their POS tag, which would persist even under random weights. We reproduce their result ( $72.14 \pm 0.51\%$  over five seeds) but note that it is still lower than the performance in our retrained condition, which uses the pretrained BERT weights ( $78.58 \pm 1.32\%$  over five seeds). The damaged network therefore contained task-relevant structure beyond what is available from a random encoder.

In contrast, the mask-only recovery conditions changed the hidden-state representation of the damaged model. The strongest mask-only condition, inverse-regularized (Mask dense) rerouting, reached 69.7% accuracy and showed the largest movement toward the pre-lesion core among mask-only models (CKA-to-core = 0.60; POS-RDM-to-core = 0.64).

Interestingly, the joint readout+mask recovery condition arrived at higher behavioral accuracy (85.9%), but remained farther from the original core geometry of the encoder (CKA-to-core = 0.51; POS-RDM-to-core = 0.53). This condition also

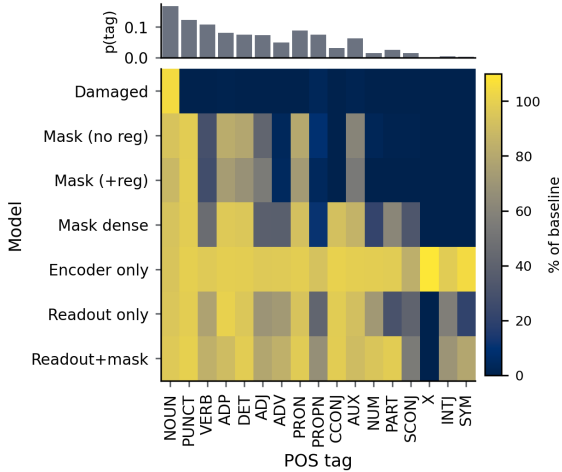


Figure 3: **Per-tag POS accuracy after core ablation and recovery.** Rows: recovery and reference conditions; columns: POS tags. Heatmap values indicate accuracy as a percentage of the corresponding pre-lesion baseline accuracy for each tag.

maintained a hidden-state representation of the encoder that was still quite similar to that of the damaged network (CKA-to-damage = 0.87). In contrast, the POS class-separation structure quantified by the weight-vector RDM was quite similar to that of the core network (POS head-to-core = 0.69).

The dissociation between the readout-only condition and the conditions involving mask retraining suggests that the former requires a new representational geometry to adapt to the changed encoder geometry. In contrast, allowing both the encoder and decoder to adapt results in a stronger recovery of the class-weight-vector representation structure in the head itself (see Figure 4C).

Finally, allowing a full retraining of the remaining network, while keeping the core POS-head fixed, produced a strong recovery of the original representational geometry (CKA-to-core = 0.79) and accuracy approaching the core performance (0.923 vs. 0.95), suggesting substantial capacity for reorganization in the damaged network.

Recovery also altered activation scale and covariance dimensionality. The mask-based recovery conditions shifted both measures towards those found for the core configuration (Appendix Figure A2).

## 5 Discussion

**Multiple recovery routes after lesioning.** Lesioning produced a strong drop in performance, but the damaged state still contained residual information that could be expressed through recovery pro-

ocols related to different mechanisms. This means that post-lesion accuracy describes the competence of the fixed residual configuration, but is not indicative of its potential to regain capacity after recovery. Furthermore, different recovery constraints produce different representational geometries.

**Readout remapping recovers performance without changing encoder geometry.** The simplest way of recovering function was by readout-only retraining, which left the damaged encoder and the gate configuration unchanged. This shows that the damaged configuration contained task-relevant information that was poorly aligned with the original readout. The hidden-state geometry of the damaged encoder was substantially different from the core representation. In parallel, the retrained classifier was also only moderately similar to the one learned by the core subnetwork.

**Gate relearning recovers performance without changing pretrained weights.** As opposed to conventional fine-tuning, the gate-based recovery condition only changed which surviving post-lesion units and projections were active. The pretrained parameters and POS readout remained fixed. This also differs from adapter tuning, which introduces new transformations, typically on top of a pretrained model. Instead, recovery based on gate optimization selects or combines existing transformations already present in the surviving network. We found that the mask-only conditions (regularized and non-regularized) partly shifted the hidden-state geometry and the POS-activation-prototype geometry towards the representations of the pre-lesion core, but they did not completely restore performance or representation.

**Joint gate and readout adaptation produces a new encoder-readout solution.** As opposed to gate-based recovery, allowing the readout classifier to adapt together with the gates produced a stronger recovery than gate-only or readout-only retraining. This suggests that the two mechanisms are complementary. Still, this was not accompanied by a return towards the pre-lesion encoder geometry. In fact, relative to the mask-dense recovery, it remained closer to the damaged hidden-state geometry while achieving better task performance. This suggests that recovery was achieved through a new alignment between the rerouted encoder representation and the adapted readout.

It is useful to compare this condition to encoder-

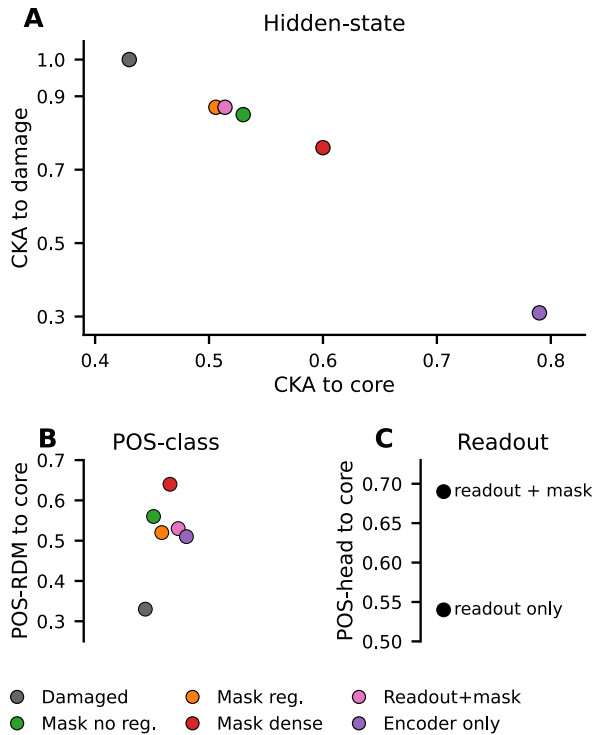


**Figure 4: Representational changes after post-lesion recovery.** **A.** Final hidden-state geometry, shown as linear CKA to the pre-lesion core and to the damaged model. **B.** POS-class geometry, measured as Spearman similarity between each condition’s POS-centroid activation RDM and the corresponding core RDM. **C.** Readout geometry, measured from POS-classifier weight-vector RDMs and the core readout RDM. Points in panel B are horizontally jittered for visibility.

only retraining, which allowed the surviving pre-trained weights to change (keeping the lesion, gates and original readout fixed). This condition was associated with the highest performance. It also showed a strong return toward the core in global hidden-state geometry, as measured by CKA. However, it did not recover the core’s POS activation-centroid geometry to the same extent, with lower values than the other recovery conditions on this measure. This condition suggests that the residual damaged model can manifest substantial representational capacity when weight relearning is allowed, but it may regain performance without recovering the original POS class-separation structure.

**Implications for circuit necessity and completeness.** As reviewed in the Introduction, there is a substantial body of circuit-discovery work that studies Transformer behavior, as defined through the model’s native output distribution for different tasks. In such cases, low performance of the *fixed* complement is then often taken as evidence that

the identified circuit captures the target behavior. In these settings, behavior is read directly from the model’s native output mechanism, and there is no independently trained task readout whose geometry can change separately from the encoder.

Because of this difference in setup, these types of investigations are intended for identifying, ablating, patching, or retraining internal components, but cannot distinguish recovery through internal adaptation from recovery through remapping of a separate task readout. Furthermore, to our knowledge, the prior related work has not retrained the residual complement after circuit removal to test whether the target behavior can be regained. Our findings therefore show that circuit necessity can be established with respect to a single, original fixed configuration, but might not hold under retraining, even when it consists of rerouting within the original network. More generally, our study demonstrates that completeness is conditional on whether and how a complement is allowed to adapt.

**Conclusions.** We show that after producing a crippling lesion, performance can recover without relearning the underlying transformations. The extent to which the original geometry is recovered depends on the degrees of freedom of the recovery protocol, and mask-based recovery in particular appears to shift the network back toward the original geometry. More generally, circuit necessity is conditional on the permitted rerouting strategy. A component may be necessary under the original fixed routing and readout, but cease to be so when the readout can be retrained or surviving transformations can be reselected.

**Limitations.** Our evaluations were based on a single task (POS classification) and a single backbone model. It is unclear whether the results generalize to more difficult tasks or to other model families. In addition, we defined lesions at the level of attention heads and dense projections. Lesioning finer-grained components could afford greater flexibility and produce better recovery. The biological relevance of these findings remains unclear, as we examined only the recovery of model performance and representations, without direct comparison with cortical damage or mechanisms of neuroplasticity. Finally, we did not identify causal circuits using the Transformer’s native outputs. A natural next step would be to apply the same recovery framework presented here after ablating circuits identified using causal circuit-discovery methods.

## References

- Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. 2024. Finding transformer circuits with edge pruning. *Advances in Neural Information Processing Systems*, 37:18506–18534.
- Sudeep Bhatia and Russell Richie. 2024. [Transformer networks of human conceptual knowledge](#). *Psychological Review*, 131(1):271–306.
- Steven Cao, Victor Sanh, and Alexander M. Rush. 2021. [Low-complexity probing via finding subnetworks](#). *CoRR*, abs/2104.03514.
- Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. [The lottery ticket hypothesis for pre-trained bert networks](#). In *Advances in Neural Information Processing Systems (NeurIPS 2020)*. ArXiv:2007.12223.
- Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352.
- Ariel Goldstein, Avigail Grinstein-Dabush, Mariano Schain, Haocheng Wang, Zhuoqiao Hong, Bobbi Aubrey, Samuel A. Nastase, Zaid Zada, Eric Ham, Amir Feder, Harshvardhan Gazula, Eliav Buchnik, Werner Doyle, Sasha Devore, Patricia Dugan, Daniel Friedman, Roi Reichart, Michael Brenner, Avinatan Hassidim, and 3 others. 2024. [Alignment of brain embeddings and artificial contextual embeddings in natural language points to common geometric patterns](#). *Nature Communications*, 15(2768).
- Ariel Goldstein, Zaid Zada, Erez Buchnik, Mariano Schain, Abigail Price, Bobbi Aubrey, Samuel A. Nastase, Amir Feder, Daniel Emanuel, Anat Cohen, Anna Jansen, Harshvardhan Gazula, Gwangsu Choe, Ashesh Rao, Chul Kim, Casey Casto, Leetal Fanda, Werner Doyle, Daniel Friedman, and 13 others. 2022. [Shared computational principles for language processing in humans and deep language models](#). *Nature Neuroscience*, 25:369–380.
- Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. *arXiv preprint arXiv:2403.17806*.
- Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. [Learning sparse neural networks through  \$l\_0\$  regularization](#). *Preprint*, arXiv:1712.01312.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. [Are sixteen heads really better than one?](#) In *Advances in Neural Information Processing Systems (NeurIPS 2019)*. ArXiv:1905.10650.
- Jasmin Moore, Vibujithan Vigneshwaran, Matthias Wilms, and Nils D. Forkert. 2023. [Degradation and plasticity in convolutional neural networks: An investigation of internal representations](#). OpenReview preprint.
- Jasmine Moore, Anup Tuladhar, Zahinoor Ismail, Pauline Mouches, Matthias Wilms, and Nils Daniel Forkert. 2022. [Dementia in convolutional neural networks: Using deep learning models to simulate neurodegeneration of the visual system](#). *Neuroinformatics*, 21.
- Maryam Rahimi, Yadollah Yaghoobzadeh, and Mohammad Reza Daliri. 2025. [Explanations of large language models explain language representations in the brain](#). *Preprint*, arXiv:2502.14671.
- Sadra Sadeh and Claudia Clopath. 2025. The emergence of neuroai: bridging neuroscience and artificial intelligence. *Nature Reviews Neuroscience*, 26(10):583–584.
- Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib Majaj, Rishi Rajalingham, Elias Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, Daniel Yamins, and James DiCarlo. 2018. [Brain-score: Which artificial neural network for object recognition is most brain-like?](#) *bioRxiv*.
- Aaquib Syed, Can Rager, and Arthur Conmy. 2024. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 407–416.
- Anup Tuladhar, Jasmine Moore, Zahinoor Ismail, and Nils Daniel Forkert. 2021. [Modeling neurodegeneration in silico with deep learning](#). *Frontiers in Neuroinformatics*, 15.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Senrich, and Ivan Titov. 2019. [Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. ArXiv:1905.09418.
- Tim Whitaker. 2024. [Neural subnetwork ensembles](#). *Preprint*, arXiv:2311.14101.
- Lei Yu, Jingcheng Niu, Zining Zhu, Xi Chen, and Gerald Penn. 2025. Sheaf discovery with joint computation graph pruning and flexible granularity. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8833–8848.

## A Masking and lesioning

### A.1 Differentiable hard-concrete gates

For each learnable mask score  $\theta_{ij}$ , we sample  $u_{ij}$  from a uniform distribution. The value is transformed using a sigmoid, which produces a stochastic gate  $s_{ij}$ . The gate is stretched  $(\gamma, \zeta) = (-0.1, 1.1)$  and then clipped to  $[0, 1]$ . This produces the final gate,  $Z_{ij}$ .

$$u_{ij} \sim \text{Uniform}(0, 1),$$

$$s_{ij} = \sigma\left(\frac{\log u_{ij} - \log(1 - u_{ij}) + \theta_{ij}}{\beta}\right),$$

$$\tilde{s}_{ij} = s_{ij}(\zeta - \gamma) + \gamma, \quad (\gamma, \zeta) = (-0.1, 1.1),$$

$$Z_{ij} = \min\{\max\{\tilde{s}_{ij}, 0\}, 1\}.$$

### A.2 Selection and permanent lesioning of core components

Based on the sparsity pattern learned by the hard-concrete gates we subsequently select components for lesioning. For attention projections, we index mask scores and gates as

$$\theta_{l,h,j}^{(t)} \quad \text{and} \quad Z_{l,h,j}^{(t)},$$

where  $l \in \{1, \dots, L\}$  indexes Transformer layers,  $h \in \{1, \dots, H\}$  indexes attention heads,  $t \in \{Q, K, V\}$  indexes the query, key, or value projection, and  $j \in \{1, \dots, d\}$  indexes features within a head, with  $d = d_{\text{model}}/H$ .

**Quantifying attention-head sparsity.** For each attention head, we define sparsity as the fraction of its Q, K, and V gates that are zero:

$$\text{Sparsity}(l, h) = \frac{1}{3d} \sum_{t \in \{Q, K, V\}} \sum_{j=1}^d \mathbb{I}[Z_{l,h,j}^{(t)} = 0],$$

where  $\mathbb{I}[\cdot]$  is the indicator function.

**Selecting attention heads for damage.** Given a sparsity threshold  $\tau \in [0, 1]$ , the set of heads selected for lesioning is

$$\mathcal{H}_{\text{damaged}} = \{(l, h) : \text{Sparsity}(l, h) < \tau\}.$$

**Permanent gate clamping.** For each selected head  $(l, h)$ , we permanently disable its Q, K, and

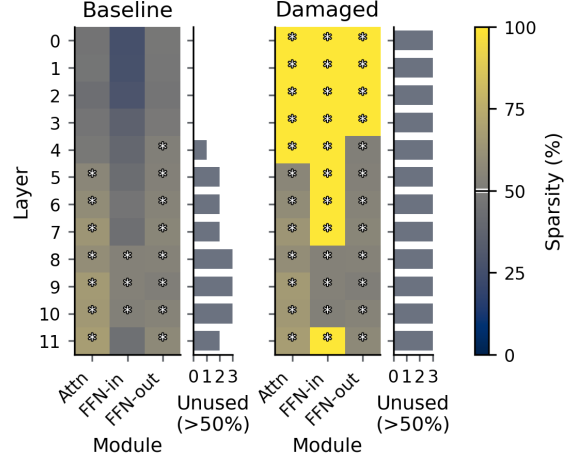


Figure A1: **Sparsity of masks before and after the structured lesion.** Rows show Transformer layers from shallow to deep; columns show the projections analyzed. Cell values indicate the percentage of learned ( $L_0$ ) gates that are inactive in a projection. *Baseline*: learned sparse-core model. *Damaged*: same model after clamping projections with sparsity below the lesion-selection threshold ( $\tau = 0.5$ ). Side bars indicate the number of dense projections that exceed 50% sparsity.

V gates by setting their associated mask scores to  $-12.0$ .

$$\theta'_{l,h,j}^{(t)} = \begin{cases} -12.0, & (l, h) \in \mathcal{H}_{\text{damaged}}, \\ \theta_{l,h,j}^{(t)}, & \text{otherwise.} \end{cases}$$

The large negative value makes the corresponding hard-concrete gates effectively zero.

**Gradient blocking.** The clamped scores are subsequently frozen during retraining:

$$\frac{\partial \mathcal{L}}{\partial \theta'_{l,h,j}^{(t)}} = 0, \quad (l, h) \in \mathcal{H}_{\text{damaged}}.$$

Consequently, units belonging to lesioned heads cannot be restored by later mask retraining.

### A.3 Sparsity before and after lesioning

Figure A1 shows the sparsity of dense projection masks and attention heads before and after lesioning the core network.

## B Recovery conditions

### B.1 Trainable components

Across the various retraining conditions, we selectively retrain two components:

- the mask parameters  $\{\theta_{ij}\}$ , which control the sparsity of projection-specific input features,
- the linear POS-tagging classifier head  $W_{\text{POS}}$ .

## B.2 Gate-training schedule

Following the exact procedure described in Cao et al. (2021), we train the gates for 20 epochs under a scheduled sparsity regularization regime. The regularization coefficient  $\lambda$  increases linearly from  $\lambda_{\text{init}}$  to  $\lambda_{\text{final}}$ : it is zero for the first 25% of updates, ramps up linearly over the next 50%, and remains constant for the final 25%. This is intended to allow the model to first identify task-relevant structures before pruning redundant neurons.

## B.3 Recovery conditions

**Readout-only retraining.** Only the output classifier head  $W_{\text{POS}}$  is retrained, while all internal parameters and masks remain frozen. The classifier consists of a single fully connected linear layer that maps the final hidden representation of each token,  $h_i \in \mathbb{R}^{768}$ , to the part-of-speech vocabulary, producing logits via an affine transformation.

**Mask-only retraining.** We study three variants reflecting different sparsity pressures:

- $\lambda_{\text{final}} = 0$ : *No explicit sparsity regularization*, allowing unconstrained rerouting.
- $\lambda_{\text{final}} = +1$ : *Standard sparsity regularization*, encouraging minimal subnetworks.
- $\lambda_{\text{final}} = -1$ : *Inverse sparsity regularization*, promoting dense subnetworks to test whether broader neuron participation aids recovery.

**Encoder-only retraining (fixed masks and readout).** Only the pretrained BERT encoder weights are updated, while mask scores and classifier remain fixed. This tests whether the surviving structure of the encoder retains enough capacity to relearn the POS task without changing its active units or readout.

**Joint readout-and-mask retraining.** Both mask scores (with active regularization) and classifier parameters are retrained jointly. This condition tests whether concurrent internal rerouting and readout adaptation yield stronger recovery than either mechanism alone.

## C Additional results

| Condition           | Mean  | SD   |
|---------------------|-------|------|
| Core baseline       | 95.03 | 0.05 |
| Encoder only        | 92.21 | 0.23 |
| Readout + mask      | 85.85 | 0.12 |
| Readout only        | 78.58 | 1.32 |
| Mask only (sparse)  | 51.63 | 7.66 |
| Mask only (no reg.) | 56.74 | 5.06 |
| Mask only (dense)   | 69.27 | 3.37 |

Table A1: **Performance across five mask-initialization seeds.** Mean held-out POS accuracy  $\pm SD$  across five seeds.

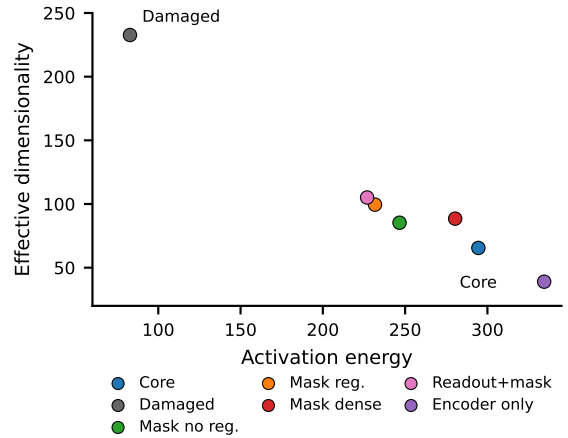


Figure A2: **Impact of retraining on activation scale and effective dimensionality of final hidden states.** Activation energy was computed as the mean squared norm of the final hidden-state vectors. Effective dimensionality was computed from the normalized spectrum of the centered covariance matrix. The damaged model showed the lowest activation energy and highest effective dimensionality, indicating a flatter covariance spectrum. Retraining restored values in the direction of the pre-lesion core.