Bachelor Informatica

University of Amsterdam

The Router's Mind: Diagnosing Mixture-of-Depths Routers with Sparse Autoencoders

Stein Pleiter

Student number 15029514

June 2026

Supervisor(s): Dr. Ana Lucic
Ege Erdogan

Signed: Dr. Dolly Sapra

Abstract

Mixture-of-Depths (MoD) routers learn, per token and per layer, whether a transformer block is computed or skipped. Existing work treats the router purely as an efficiency mechanism and evaluates it by perplexity at a target compute budget; the *structure* of what these routers learn has not been systematically characterised. This thesis adapts token-level Router-Tuning to Gemma 2 2B, where each router gates a token's attention update, and asks what the resulting routers read from the residual stream. We first establish that the router is not a trivial token-importance ranker: its decisions differ between full- and sliding-window layers and depend on part-of-speech in a layer-conditional way (Chapter 4, also reported in a companion workshop paper). We then use Sparse Autoencoders (SAEs) to decompose what the router reads. We find that routing does not reduce to any single interpretable feature; instead a *collection* of roughly 50–200 sparse features together recovers about half of the router's score variance on held-out tokens (Chapter 5). Classifying those features along a syntactic-to-semantic axis shows that semantic (content) features dominate, and an independent set of automatically generated feature labels agrees. Finally, causal intervention on these features flips routing decisions in the predicted direction at every studied layer (up to a 30.6% conditional flip rate), and a cross-checkpoint comparison shows that the *strength* of the causal handle is stable across independent trainings while its *direction* is training-specific (Chapter 6). Two further checks reinforce the causal findings: the effect replicates when features are identified and intervened on disjoint corpus halves, and forcing routing decisions measurably changes the model's output. We also confirm that the routed model retains most of its capability relative to base Gemma 2 2B.

Contents

CHAPTER 1

Introduction

Large language models (LLMs) have become general-purpose infrastructure, powering applications from code completion and web search to conversational assistants. This reach comes at a steep and growing computational cost: such models are expensive to train, and because every inference call runs the full network over every token, expensive to serve at scale. A natural response is to spend computation where it is needed rather than uniformly, allocating compute *adaptively* across tokens and layers instead of treating every token as equally demanding. This is the premise of conditional-computation architectures, of which Mixture-of-Depths (Raposo et al. 2024) is a recent and simple instance.

The Mixture-of-Depths (MoD) architecture of Raposo et al. (2024) realises this by adding, at each transformer block, a small *router* that decides per token whether the block is computed or bypassed via the residual connection. The fraction of tokens processed per layer is held to a target *capacity* (the desired fraction to process), trading a small increase in *perplexity*, the standard language-modelling quality measure, for substantial compute savings. He et al. (2025) make the mechanism parameter-efficient with *Router-Tuning*, which trains only the routers on a frozen pre-trained model. Beyond efficiency, this isolates the router as the single learned component, making it a well-defined object of study.

Existing work treats this router purely as an efficiency mechanism: given a capacity it is trained to meet, and evaluated by perplexity at that capacity. What the router *learns* — whether its skip-or-process decisions reflect identifiable structure beyond a gross token-importance heuristic — has not been systematically characterised. This gap matters because efficient-inference designs typically assume token importance reduces to simple scalar cues such as activation norm or attention mass; if routers instead encode structured, layer-conditional functions, their decisions become an interpretability target in their own right, with implications for safety auditing and controllable inference. Unlike a Mixture-of-Experts router, which selects *which* expert processes a token, an MoD router decides *whether* to apply a block's computation to a token or route around it via the residual connection. In the variant studied here, the router gates only the attention sub-block, so a skipped token receives no attention update at that layer but still passes through the MLP and remains in the *residual stream*, the running vector each token carries through the network that every block reads from and writes back into. A router that systematically skipped tokens carrying safety-relevant information — negation, refusal cues, factual qualifiers — could degrade reliability in ways perplexity does not reveal.

Prior MoD work studies uniform-attention transformers, in which every block uses the same attention mechanism. Gemma 2 (Team et al. 2024) instead alternates full and sliding-window attention layer by layer, a structural difference between layers that a router could respond to and that, to our knowledge, no prior work has examined. We therefore study MoD routers trained on Gemma 2 2B.

1.1 Problem statement

Current evaluations of MoD architectures focus on efficiency metrics: FLOP reduction, perplexity, and benchmark accuracy. The routing mechanism itself is treated as a black box. We therefore lack a fundamental understanding of what the router has learned: does it exploit shallow syntactic heuristics (skipping punctuation, determiners, stopwords), or does it recognise deeper semantic redundancy? Can routing decisions be decomposed into human-interpretable units, and can interventions on those units causally redirect routing?

Our results show that MoD routing in Gemma 2 2B is neither a norm threshold nor a single feature. It differs between full- and sliding-window layers and depends on part-of-speech in a layer-conditional way; it decomposes into a collection of roughly 50–200 sparse-autoencoder features that together explain about half of the router's score on held-out tokens, predominantly semantic rather than syntactic; and those features causally drive routing when intervened upon. The decision is distributed and mostly content-based.

1.2 Research questions

This thesis investigates six research questions. The first three characterise routing *behaviour*: what the router does, and whether simple mechanisms explain it. We explored these in a companion workshop paper (Pleiter et al. 2026), accepted to the Mechanistic Interpretability Workshop at ICML 2026; we restate them here because their answers scope the SAE analysis that is the main subject of this thesis, telling us which layers to study and along which axis interpretable structure is likely to lie (the workshop paper is linked in Appendix A). The remaining three are the primary contribution of this thesis: they use SAEs to decompose the router's decision and characterise the features that drive it.

RQ1.
Do trained MoD routers route differently at full-attention versus sliding-window layers in Gemma 2 2B?

RQ2.
Can any difference be explained by simpler architectural properties: (a) layer-aggregate residual-stream norm, (b) per-token residual-stream norm, or (c) per-layer task-loss sensitivity to attention ablation?

RQ3.
Do routing decisions encode interpretable linguistic features beyond token importance: can per-token routing be predicted from part-of-speech tags?

RQ4.
Can routing decisions be decomposed into interpretable SAE features, and are those features individually monosemantic or collectively distributed?

RQ5.
Of the features that drive routing, do they fire on low-level syntactic cues (part-of-speech, position) or on higher-level semantic content?

RQ6.
Can routing be *causally* manipulated by intervening on identified features, rather than only correlated with them?

RQ1–RQ3 are answered in Chapter 4; we treat RQ4 and RQ5 together in Chapter 5 (the same SAE feature analysis viewed two ways, since RQ5 classifies RQ4's features along a syntactic-to-semantic axis), and RQ6 in Chapter 6.

1.3 Thesis structure

Chapter 2 introduces MoD and Router-Tuning, the Gemma 2 architecture and Gemma-Scope, SAEs for interpretability, and the faithfulness desiderata that motivate the intervention programme, then positions the thesis against related work on dynamic-depth routing, analyses of what learned routers encode, and SAE-based feature intervention. Chapter 3 describes the token-level Router-Tuning adaptation, training configurations, evaluation pipeline, and SAE setup, including the choice of where to read the residual stream. The three experiment chapters then proceed from behaviour to mechanism. Chapter 4 characterises routing behaviour: the routing-rate difference between layer classes, three controls that rule out simple explanations, and the layer-conditional part-of-speech structure. Chapter 5 decomposes routing into SAE features (RQ4) and classifies them along the syntax–semantics axis (RQ5). Chapter 6 intervenes causally on those features (RQ6). Chapter 7 discusses what the three chapters show together and their limitations, including the layer-parity confound and the dual-use posture. Chapter 8 concludes.

CHAPTER 2

Background and Related Work

2.1 Background

2.1.1 Mixture-of-Depths and Router-Tuning

Raposo et al. (2024) introduced MoD, in which a per-block router assigns each token a scalar weight and only the top-weighted tokens (up to a capacity budget) are processed by the block; the rest bypass it through the residual connection. He et al. (2025) proposed Router-Tuning, which freezes the base model and trains only the routers, turning MoD into a parameter-efficient addition rather than an architecture trained from scratch. Freezing is important here: because the base weights never move, any structure we later find in routing decisions must already be present in the pre-trained residual stream and cannot be the product of the base model adapting itself to make routing easier.

2.1.2 The Gemma 2 architecture and Gemma-Scope

We use Gemma 2 2B (Team et al. 2024), a 26-layer decoder-only model whose blocks alternate full attention and sliding-window (local) attention layer by layer. This alternation is what makes the layer-class comparison in Chapter 4 well-posed. Gemma-Scope (Lieberum et al. 2024) is a suite of SAEs trained on Gemma 2 residual streams at every layer and several dictionary widths; we use the canonical residual-stream release at width 16,384. Using an off-the-shelf SAE suite, rather than training our own, keeps the feature dictionary independent of our router training.

2.1.3 Sparse autoencoders for interpretability

An SAE reconstructs an activation vector as a sparse non-negative combination of learned dictionary directions, in the hope that individual directions are *monosemantic*, each corresponding to a single human-interpretable concept (Bricken et al. 2023; Cunningham et al. 2023). SAEs are primarily a *discovery* tool. Peng et al. (2025) argue specifically that SAEs are best used to discover unknown concepts rather than to act on concepts already known in advance; we keep this caution in mind, using SAEs to find what the router reads (RQ4, RQ5) before acting on those features (RQ6). Sharkey et al. (2025) catalogue open problems with SAE-based interpretability, including feature splitting (one underlying concept fragmenting across several dictionary features, so no single feature captures it cleanly) and the unreliability of single-feature labels, both of which surface in our results. Shi et al. (2025) develop SAE methodology that shares information across layers; this is a different sense of "route" from MoD routing and is not router analysis.

2.1.4 Faithfulness and causal abstraction

Correlational evidence (a feature co-varying with router score) is consistent with both genuine causal readout and mere coincidence. Andersen et al. (2025) formalise desiderata for faithful explanations through causal abstraction, among them an interventionist causal criterion: a feature is a faithful cause only if intervening on it produces the predicted change in the output. We operationalise this criterion as *bidirectional consistency*, requiring that a faithful causal claim predict the effect of intervening in *both* directions. Chapter 6 implements this as paired suppression and amplification.

2.2 Related work

2.2.1 Dynamic-depth routing and its alternatives

A growing literature makes transformer depth conditional. Beyond MoD (Raposo et al. 2024) and the Router-Tuning variant we adopt (He et al. 2025), alternatives to a learned per-token router include attention-based routing that reuses the previous layer's attention map (Gadhikar et al. 2024), post-hoc conversion of pretrained models to MoD (Tan et al. 2024; Zhang et al. 2024), multimodal MoD (Y. Luo et al. 2025), and residual-stream gating (Laitenberger et al. 2026). Lawson and Aitchison (2025) find that skipping whole middle layers fails to beat dense baselines, which motivates the finer-grained per-token routing MoD provides. These works share a single aim, to spend less compute at equal quality, and evaluate the router by perplexity at a target capacity. Most do not examine the *content* of the decisions the router learns; the exceptions inspect the gate only at the level of token types, such as which tokens are sent to greater depth (X. Luo et al. 2025) or which residual gates fire (Laitenberger et al. 2026), rather than decomposing the decision into interpretable features. To our knowledge, no prior work has trained MoD on an alternating-attention architecture such as Gemma 2 or characterised what its routers read.

2.2.2 What learned routers and components encode

A second line of work asks what learned routing and attention components encode, rather than how to make them efficient. In Mixture-of-Experts (MoE) models, expert assignment has repeatedly been found to track shallow, often syntactic token properties: Jiang et al. (2024) report that Mixtral's router correlates with syntax and position rather than semantic domain, and Zoph et al. (2022) observe experts specialising on punctuation, conjunctions, and other surface features. The work most related to ours is Antoine et al. (2024), who train MLP probes over MoE routing paths and find part-of-speech predictive of expert assignment. We differ in three ways: (1) we study the binary skip/process decision of MoD rather than multi-class expert selection, (2) we work in an alternating-attention architecture rather than uniform-attention MoE, and (3) we go beyond probing to SAE decomposition and causal intervention, and identify layer-conditional sign flips in which the same part-of-speech is preferentially skipped at one layer and preserved at another. On the attention side, Michel et al. (2019) and Voita et al. (2019) score attention heads by the cost of removing them and find many prunable, with the survivors specialised to interpretable roles; an MoD router differs in making a per-token, per-layer, context-dependent decision rather than a static head-level one. More broadly, probing work shows that linguistic structure is linearly recoverable from transformer representations (Tenney et al. 2019), which is what makes a linear probe a sensible first instrument in Chapter 4.

2.2.3 Decomposing and intervening on features with SAEs

Our decomposition and intervention build on recent SAE work that moves from finding features to acting on them. Marks et al. (2025) assemble SAE features into causal graphs and edit them to change model behaviour; this is methodologically the closest precedent to our intervention, though applied to a different target. Closest in what it studies, Lasy et al. (2026) use SAE features to predict expert-routing decisions, though in an MoE rather than an MoD model.

To our knowledge, however, no prior work applies sparse autoencoders to a *Mixture-of-Depths* router: the closest analyses are either SAE-free (Antoine et al. 2024) or target MoE rather than MoD (Lasy et al. 2026). Decomposing the binary skip/process decision of an MoD router into interpretable features, and intervening on them causally, is the gap this thesis fills.

CHAPTER 3

Method

3.1 Overview of approach

We adapt token-level Router-Tuning to Gemma 2 2B. We choose this base model for two reasons. First, its 26-layer decoder alternates full and sliding-window attention block by block, so a router can treat the two attention types differently. Second, Gemma-Scope (Lieberum et al. 2024) provides pre-trained SAEs at every layer; loading these removes an ∼80 GPU-hour SAE-training cost and fixes the SAE training distribution to the base model's pre-training data rather than to a small fine-tuning set, so the features we analyse are not artefacts of our training choices.

We use SAEs as the decomposition tool for an architectural reason. The MoD router computes its scalar weight as a *linear projection* of the residual stream. A linear projection can respond only to structure that is linearly represented in its input, so any concept that drives routing must be present as a linear direction in the residual stream. Sparse autoencoders (SAEs) decompose activations into sparse, approximately linear directions (Bricken et al. 2023; Cunningham et al. 2023), and therefore operate on the same space the router reads.

We use the token-level variant of Router-Tuning because per-token feature analysis requires a per-token routing signal. The sequence-level variant assigns one routing decision to an entire sequence, under which per-token feature attribution is undefined. The token-level variant (He et al. (2025), Eq. 1) makes one decision per token. Because the publicly released Router-Tuning code implements only the sequence-level variant, we re-implement the token-level variant and provide a Gemma 2 MoD adaptation that, with routers initialised to always process, reproduces base Gemma's logits to within floating-point error and selects the same next token. This comparison is on logits rather than sampled text, so it is a deterministic check that the adaptation changes nothing when routing is disabled.

The analysis is ordered by computational cost. We first use linear probes and behavioural tests to identify the layers that carry the strongest routing signal (Chapter 4), so that the more expensive SAE analysis is restricted to those layers. We then decompose the residual stream at those layers with the Gemma-Scope SAEs and rank features by their correlation with the router's score (Chapter 5). Finally, we test whether these correlations reflect a causal relationship by intervening on features and measuring the change in routing (Chapter 6). For the intervention we follow the interventionist causal desideratum of Andersen et al. (2025), under which a feature counts as a cause of a routing decision only if intervening on it produces a change in that decision. We operationalise this as a *bidirectional* test: suppressing a feature and amplifying it are required to move routing in *opposite* directions, and effects are evaluated against a frequency-matched baseline rather than against zero. Figure 3.1 summarises the three-stage pipeline and the dependencies between chapters.

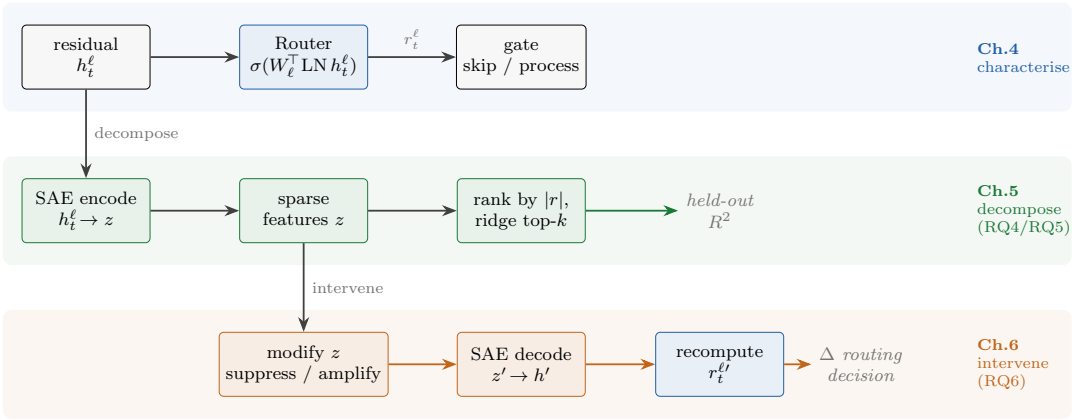

Figure 3.1: The three-stage pipeline, one row per experiment chapter. **Characterise** (Chapter 4): the MoD router reads the layer-normalised residual and emits a per-token score r_t^ℓ that gates the attention sub-block. **Decompose** (Chapter 5): the same residual is passed through a Gemma-Scope SAE, features are ranked by their correlation $|r|$ with the score, and a ridge fit (linear regression with an ℓ_2 penalty that guards against overfitting) on the top-k is scored by held-out R^2. **Intervene** (Chapter 6): those features are suppressed or amplified, the residual is recomposed, and the router is re-run to read off the change in routing decision. Reading downward at the two vertical connectors shows the logical dependency: we only decompose what we first showed is structured, and only intervene on what the decomposition selected.

3.2 Token-level Router-Tuning on Gemma 2 2B

We add a token-level MoD router to the deepest 13 layers (layers 12–24), leaving the final layer 25 dense, following He et al. (2025)'s convention of targeting deep layers except the last; the motivation is prior evidence that deeper layers are more redundant than shallow ones, which carry essential representations. At each MoD-target layer ℓ a single linear router $W_\ell \in \mathbb{R}^d$ ($d = 2304$) produces a per-token score $r_t^\ell = \sigma(W_\ell^\top \mathrm{LN}(h_t^\ell))$, where σ is the sigmoid, h_t^ℓ is the residual entering the block, and LN denotes Gemma 2's normaliser, RMSNorm, which we write LN throughout (in equations and figures); the score is binarised at 0.5 via a *straight-through estimator*: a hard 0/1 threshold has zero gradient everywhere, so during back-propagation the threshold is treated as the identity, letting the training signal reach the router. Each Gemma 2 block comprises a self-attention sub-block followed by an MLP sub-block; following He et al. (2025), the router gates the attention sub-block only. A token scored below the threshold therefore bypasses attention through the residual connection while still passing through the MLP, so routing governs each token's attention update rather than whether the block as a whole runs. Only the routers (∼30K parameters total) are trained, against $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda\, \mathrm{ReLU}(\bar{c} - c^*)$, where $\bar{c}$ is the batch mean routing rate, $c^* = 0.5$ is the target capacity, and λ scales the capacity penalty (higher λ enforces stricter targeting, lower λ lets task loss dominate). Training uses causal language-modelling loss on the Alpaca instruction corpus, one of the instruction-tuning datasets evaluated by He et al. (2025).

3.3 Training configurations

We train four checkpoints crossing two penalty strengths $\lambda \in \{0.01, 0.1\}$ with two training scales (5K samples × 1 epoch; 52K × 3 epochs), spanning an order of magnitude on each axis: **strict-short**, **gentle-short**, **gentle-long**, and **strict-long**. We use **strict-long** ($\lambda = 0.1$, 52K × 3) as the lead checkpoint for all per-token analyses because it pairs full training with the paper-specified capacity penalty. All four are compared in the attention-type analysis (Chapter 4); the cross-training checks in Chapters 5 and 6 use **strict-long**, **strict-short**, and **gentle-short**

(Table 3.1); Appendix B characterises the **gentle-long** run.

Table 3.1: The four trained MoD checkpoints: capacity penalty λ, training scale, and the resulting mean routing rate (fraction of tokens processed per MoD layer on WikiText-2, target 50%). **strict-long** is the lead checkpoint for all per-token analyses; the cross-training robustness checks in Chapters 5 and 6 use **strict-long**, **strict-short**, and **gentle-short**.

Checkpoint	λ	Training scale	Routing rate
strict-short	0.1	$5\text{K} \times 1$	58%
gentle-short	0.01	$5\text{K} \times 1$	72%
gentle-long	0.01	$52\text{K} \times 3$	76%
strict-long	0.1	$52\text{K} \times 3$	62%

3.4 Evaluation pipeline

All evaluation is on the WikiText-2 test set (Merity et al. 2016): 200 documents tokenised and truncated to length 256, giving 33,608 tokens. For per-token analyses we exclude the first 4 positions of each document to avoid attention-sink outliers (Xiao et al. 2024), leaving 32,808 tokens. Evaluating on WikiText while training on Alpaca means the routing patterns we measure reflect generalisation rather than memorisation of the training corpus.

3.5 Validation: the routed model remains functional

Interpreting a router is only meaningful if the routed model is itself sound. A model whose routing had severely degraded it would still produce structure to decompose, but that structure would reflect the damage rather than the mechanism we want to study. We therefore check that adding the MoD routers leaves Gemma 2 2B functional, comparing the lead **strict-long** checkpoint against the unmodified base model.

At its operating point the router processes 62% of tokens (target 50%; Table 3.1), and at the three layers we go on to interpret it processes 44% (layer 13), 59% (layer 23), and 47% (layer 24). On WikiText-2 this raises perplexity from 17.1 to 24.4. Perplexity is a stringent measure, sensitive to the full next-token distribution at every position; He et al. (2025) validate Router-Tuning on downstream *accuracy* rather than perplexity, so accuracy is the comparable quantity. We measure length-normalised accuracy (`acc_norm`) on HellaSwag and ARC-Challenge at three few-shot settings, reporting the base-to-MoD change on identical items (Table 3.2).

Table 3.2: Base Gemma 2 2B versus the MoD **strict-long** model on length-normalised accuracy (`acc_norm`), 300 items per task, at three few-shot settings. Each cell is base $\rightarrow$ MoD; Δ is the mean change on identical items. The 10/25-shot row is the field-standard protocol (HellaSwag 10-shot, ARC-Challenge 25-shot).

Setting	HellaSwag	ARC-Challenge	Average Δ
0-shot	$0.640 \rightarrow 0.577$	$0.500 \rightarrow 0.413$	-7.5pp
5-shot	$0.663 \rightarrow 0.613$	$0.520 \rightarrow 0.470$	-5.0pp
10/25-shot	$0.657 \rightarrow 0.600$	$0.517 \rightarrow 0.453$	-6.0pp

The drop is real and modest, 5 to 7pp on average, and it does not vanish with more demonstrations: the few-shot gaps are smaller than the zero-shot gap but do not close, and the field-standard 10/25-shot protocol still leaves 6.0pp. This is larger than Router-Tuning's best-case 0.2% accuracy drop, which is expected given a harder setting on every axis: a 2B model rather

than their 8B, token-level rather than sequence-level routing (which the per-token interpretability requires, and which is already the weaker of their two variants), and training on Alpaca (their lowest-performing dataset). The absolute base scores are slightly below the published Gemma 2 2B figures because our lightweight harness differs from the standard evaluation suite in prompt formatting and sample size; the *difference* on identical items, not the absolute level, is the quantity we rely on.

The routed model therefore retains most of its capability at a measurable cost. We interpret a working router, not an efficiency-optimal one: we make no claim that this routing is compute-optimal, and the implementation masks the attention output rather than skipping its computation, so it does not by itself save wall-clock time. What the chapters that follow require is only that the router we open is a sound model whose decisions carry genuine structure, which the following chapters establish.

3.6 Where the SAE reads the residual stream

The router at layer ℓ reads the residual *entering* the block, resid_pre(ℓ), which equals resid_post($\ell-1$), the residual leaving the previous block. We therefore decompose resid_pre(ℓ) with the Gemma-Scope SAE trained at layer $\ell-1$, captured by a forward hook on layer $\ell-1$. This choice is deliberate: reading resid_post(ℓ) instead would be confounded by the layer-ℓ gate, because that residual already contains the attention output the router decided to add or skip, so a correlation there could be an effect of the routing decision rather than its cause (Figure 3.2). For the routing layers we study, $\{13, 23, 24\}$, the SAE input layers are therefore $\{12, 22, 23\}$.

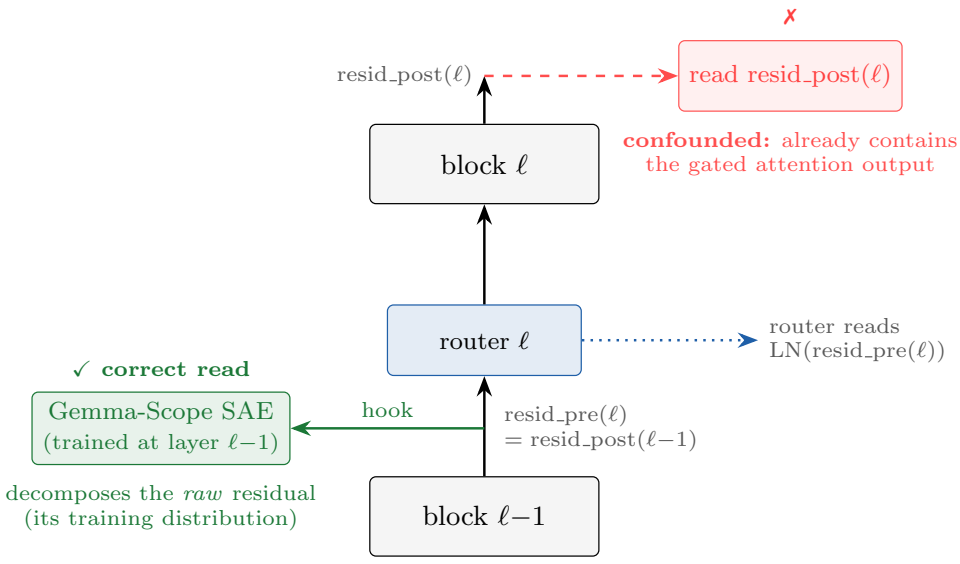

Figure 3.2: Where the SAE hooks the residual stream, and why. The router at layer ℓ and the SAE both read resid_pre(ℓ) = resid_post($\ell-1$) (left, green): this is the raw residual *entering* the block, which is also the Gemma-Scope SAE's training distribution, so the decomposition is on-distribution. Hooking resid_post(ℓ) instead (right, red) would be confounded, because that residual already contains the attention output the layer-ℓ gate decided to add, turning a putative cause of the routing decision into a possible effect of it. The router itself reads the layer-normalised version LN(resid_pre(ℓ)); because RMSNorm divides out the overall magnitude, a feature that correlates with routing does so on directional grounds.

The SAE reads the raw residual (its training distribution) whereas the router sees the layer-normalised residual; both are functions of resid_pre(ℓ), and because RMSNorm rescales the residual to a roughly fixed magnitude (dividing out its overall norm), a feature that correlates with routing does so on directional grounds rather than through magnitude alone. The continuous

router score (not just the stored binary decision) is recovered by hooking the router module and applying the sigmoid; we verified that thresholding this score at 0.5 reproduces the stored binary decision for every token.

3.7 SAE reconstruction quality at the chosen layers

Because all later analysis is mediated by the SAE, we first check that the SAE reconstructs the residual well enough at the input layers we use. We measure this with *cosine similarity*: the cosine of the angle between the raw residual vector and its SAE reconstruction, which equals 1 when the two point in exactly the same direction and falls toward 0 as the reconstruction diverges. Cosine, rather than a magnitude-based error, is the relevant metric here because the router reads a layer-normalised residual (Section 3.6): what the SAE must preserve is the residual's *direction*. Table 3.3 reports cosine at SAE layers 12 and 22 (layer 23 was validated previously at 0.876); all exceed 0.85. We set this bar relative to the base model rather than as an absolute cutoff. Reconstruction on the MoD-tuned residual is modestly degraded relative to base Gemma 2 (cosine ≈ 0.93), and a pre-registered target of 0.95 proved unreachable even on the base model, so we anchor the threshold to measured base-model performance. No routing layer is dropped for poor reconstruction; the directional error that keeps cosine below 1 is the same shortfall we quantify in Section 5.3, where it sets a floor on how much of routing the SAE can explain.

Table 3.3: SAE reconstruction quality at the residual layers read by the routers. Cosine is between the raw residual and its SAE reconstruction; L0 is the mean number of active features per token.

SAE layer	cosine	L0	feeds routing layer
12	0.907	87.6	13
22	0.883	114.6	23
23	0.876	–	24

CHAPTER 4

Routing Behaviour: Architectural and Syntactic Structure

Problem formulation. Before decomposing the router with SAEs, we ask whether it has non-trivial internal structure. If the router were a simple token-importance ranker, indifferent to architecture and language, we would expect its decisions to track simple scalar properties of the residual stream, leaving little for a feature-level analysis to add. This chapter tests that possibility from three angles: whether routing depends on attention type (RQ1), whether three simple mechanisms can account for any such dependence (RQ2), and whether routing tracks linguistic structure (RQ3). Its conclusion, that the router reads something architecturally and linguistically structured, motivates the SAE analysis in Chapter 5. The results in this chapter are also reported in a companion workshop paper, linked in Appendix A.

4.1 Routing-rate differences between full- and sliding-window layers (RQ1)

For each WikiText-2 document we compute the mean routing rate separately over full-attention and sliding-window MoD layers and compare them with a paired t-test on the per-document means; we report effect size as Cohen's d, the standardised mean difference, with the conventional small/medium/large thresholds of 0.2/0.5/0.8. In shallow-middle MoD (layers 12–20), every one of the four checkpoints routes a higher fraction of tokens at full-attention than at sliding-window layers, with $d \in [0.31, 2.06]$ and all $p < 10^{-4}$ (Table 4.1; Figure 4.1 shows the per-layer rates for the strict-long checkpoint). In deep MoD (layers 21–24) the direction is inconsistent across checkpoints, so we treat only the shallow-middle difference as robust. This difference is an association, not evidence that a router reads attention type: in Gemma 2 every full-attention layer has an odd index and every sliding-window layer an even one, so attention type is confounded with layer parity, and because each layer trains its own router the comparison contrasts routers at the two layer classes rather than a single router responding to attention type (Chapter 7).

4.2 Ruling out simple explanations (RQ2)

The routing-rate difference of Section 4.1 is only worth a feature-level investigation if the router reads structured signal. The simpler alternative is that the router tracks some low-level scalar property of the residual stream that merely happens to differ between the two attention types; if so, the preference would be an artefact of that scalar and there would be little for an SAE to add. We therefore test the three most plausible simple explanations. Each is stated as a falsifiable prediction: *if* that mechanism drove routing, a specific measurement would come out a specific way. We then check whether it does. The three differ in what they attribute the decision to: the average size of a layer's activations (aggregate norm), the size of an individual token's activation

Table 4.1: Paired full- vs. sliding-attention routing-rate t-test in shallow-middle MoD (layers 12–20). A positive d means full-attention layers are processed more. The direction is consistent across all four checkpoints.

Checkpoint	Cohen's d	full > slid %	p
strict-short	+1.33	86.5%	$4 \cdot 10^{-46}$
gentle-short	+2.06	98.5%	$1 \cdot 10^{-73}$
gentle-long	+0.31	62.0%	$2 \cdot 10^{-5}$
strict-long	+0.86	78.5%	$9 \cdot 10^{-26}$

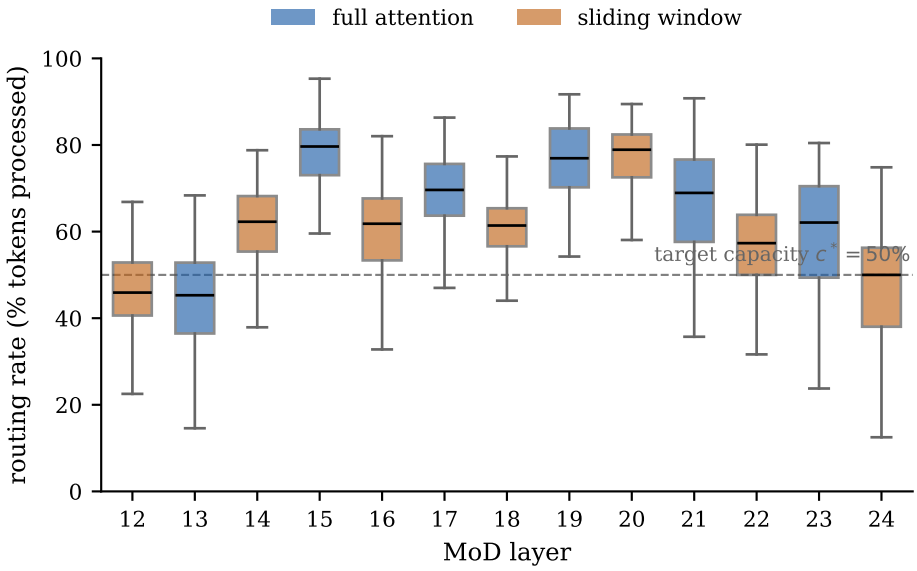

Figure 4.1: Distribution of per-document routing rates at each MoD layer on the strict-long checkpoint (box = interquartile range over WikiText-2 documents; boxes coloured by attention type). Although the global target capacity is 50% (dashed line), the router does not spend it uniformly: full-attention (odd-index) layers are visibly higher than their sliding-window neighbours across the shallow-middle range, the per-layer view behind the paired test in Table 4.1.

(per-token norm), or how much the layer matters to the task (task-loss sensitivity). Figure 4.2 collects the result of all three tests.

(a) Layer-aggregate norm. The router reads a layer-normalised residual, but its linear projection could still be sensitive to the *scale* of activations at a layer: a layer whose residual vectors are larger on average would push the projection further from the decision threshold. If the router preferred high-norm layers in this way, then the layers it processes more, the full-attention layers, should have the higher residual-stream norms. The prediction fails, and in the opposite direction. In the strict-long MoD context the mean residual norm $\|h\|$ at full-attention layers is *lower* than at sliding-window layers (82.97 versus 84.31, Cohen's $d = -3.86$), and the same ordering holds in the unmodified base model (83.86 versus 84.76). The router processes the *lower*-norm layers more often. This refutes the intuitive form of the worry, in which a large norm marks a layer as salient and pulls compute toward it. It does not, on its own, settle the norm account: a preference for *low*-norm layers would fit the data equally well, and that is still a norm rule, only with the sign flipped. What rules out both versions is that the router's score barely tracks norm in either direction, which is the subject of test (b).

(b) Per-token norm. Test (a) concerns layer averages; a finer version of the same idea is that the router thresholds on each *token's* norm, processing tokens whose residual is large and skipping those whose residual is small. If that were the rule, the router's continuous score would correlate strongly with the per-token residual norm $\|h\|$. It does not: across MoD layers the mean absolute Pearson correlation between score and $\|h\|$ is only 0.166, and even at the most norm-sensitive layer (layer 16) it reaches just 0.336, where 0 denotes no linear relationship and 1 a perfect one. Squaring this correlation gives the share of the score's variance that norm can account for (r^2): even at that strongest layer, per-token norm explains at most about 11% of the router's decision and leaves the remaining ∼89% unexplained. A per-token norm thresholder is therefore not what the router has learned.

(c) Task-loss asymmetry. A subtler possibility is that the router has learned which layers are most important: if full-attention layers were individually more important to the model's predictions, then processing them more would be a sensible importance-tracking strategy rather than evidence of structured reading. We operationalise a layer's importance as the damage done by removing it: we *ablate* each attention sub-block in turn (zeroing its output while leaving the rest of the model intact) and measure the resulting rise in WikiText-2 perplexity. If full-attention layers were more important, ablating them would cost more. The costs are close and statistically indistinguishable: ablating full-attention sub-blocks raises perplexity by +1.56 on average against +1.22 for sliding-window sub-blocks, a difference of +0.34 that is not significant (Welch's t-test, $p = 0.44$; we use Welch's variant because the two groups have unequal sizes and possibly unequal variances). To check that this null is not hiding a generic "any deeper layer costs more" effect, we repeat the ablation on the MLP sub-blocks (which the router never gates) and again find no attention-type difference ($p = 0.88$). We therefore detect no per-layer-importance asymmetry. We state this as "no detectable asymmetry" rather than "no asymmetry": with only four full-attention and five sliding-window layers in the shallow-middle range, the test has limited statistical power, so a small asymmetry could exist below what nine layers can resolve.

None of the three mechanisms accounts for the routing-rate difference: it is not explained by how large activations are (aggregate or per-token), nor by how task-important the layers are. What remains is that the router reads a structured function of the residual's *direction* rather than its magnitude or its layer's importance, exactly the kind of signal a sparse autoencoder is built to decompose. This negative result is what motivates the feature-level analysis of Chapter 5.

4.3 Layer-conditional part-of-speech structure (RQ3)

We next ask whether routing tracks linguistic structure. We use part-of-speech (POS) because it is densely encoded in transformer residual streams (Tenney et al. 2019) and is the analysis target in the related MoE study of Antoine et al. (2024), enabling direct comparison. We tag

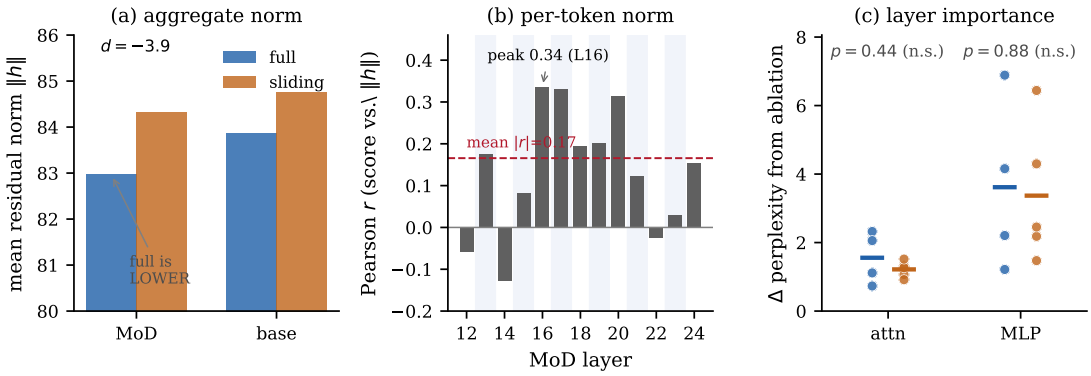

Figure 4.2: The three candidate explanations for the routing-rate difference, each falsified. **(a) Aggregate norm:** full-attention layers have *lower* mean residual norm than sliding-window layers, in both the MoD context and the base model, so the router processes the lower-norm layers more, the opposite of a norm preference. **(b) Per-token norm:** the per-layer Pearson correlation between router score and residual norm is weak everywhere (mean $|r| \approx 0.17$), so even at its peak per-token norm explains little of the score. **(c) Layer importance:** attention- and MLP-ablation costs (Δ perplexity) do not differ by attention type (p not significant), so the preference is not importance-tracking. Each panel corresponds to one tested mechanism in this section.

each WikiText-2 token with spaCy (Honnibal et al. 2020)—a standard pre-trained part-of-speech tagger—and, per layer, test whether each POS tag's routing rate differs from the layer's marginal rate using a binomial test (the decision is binary), Bonferroni-corrected across the 16 tags: testing 16 tags per layer inflates the chance that one looks significant by accident, so we divide the 0.05 threshold by 16 and require $p < 0.0031$ for a tag to count.

Many tags deviate sharply from the marginal rate, and the *same* tag is often routed differently at different layers (Table 4.2; Figure 4.3 shows the full POS-by-layer map). Determiners are preferentially skipped at layer 13 (−11pp) but preferentially preserved at layers 23 and 24 (+11 and +14pp), a shallow-versus-deep inversion. A logistic regression from POS one-hot to routing decision lifts test accuracy by +6.6pp at layer 13 and +9.1pp at layer 24 over the majority-class baseline, confirming the per-tag effects aggregate into real predictive power (at layer 23 the aggregate probe matches the baseline despite significant per-tag effects).

Table 4.2: Per-POS deviation from the marginal routing rate (percentage points). Positive = preferentially preserved; negative = preferentially skipped. The same category is routed differently at different layers.

POS	Layer 13	Layer 23	Layer 24
CCONJ	+31.8	+19.9	+14.1
PUNCT	+17.1	+7.5	−22.2
DET	−11.3	+11.5	+14.2
NUM	+5.7	+6.2	+22.9
VERB	−13.0	−6.8	−5.6
PROPN	−0.5	−12.5	−3.4

The router does not learn a static priority over linguistic categories; it learns a layer-conditional function. Together with the previous section, this establishes that routing is structured both architecturally and linguistically, which is precisely the kind of structure an SAE can decompose. Two consequences scope the next chapter: the strongest signal is at full-attention shallow-middle layers (we study 13, 23, 24), and because the function is layer-conditional, the SAE analysis must be done per layer rather than pooled.

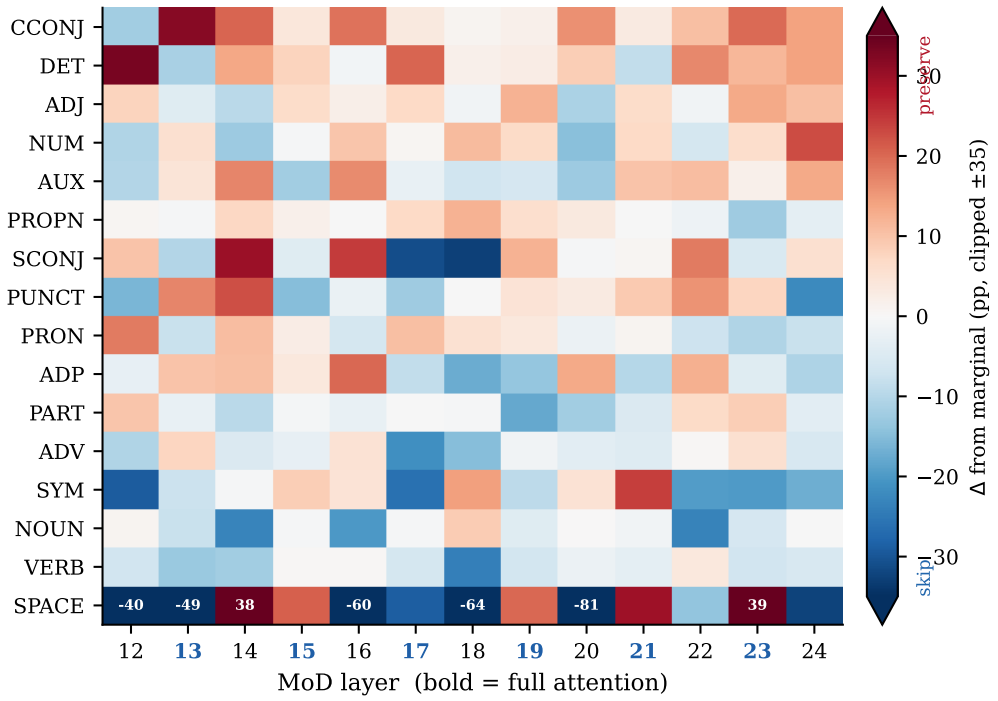

Figure 4.3: Per-POS deviation from each layer's marginal routing rate across all MoD layers (red = preferentially preserved, blue = preferentially skipped; full-attention layer indices in bold blue). The colour scale is clipped at ±35pp so the structure is legible; the SPACE row, which saturates as high as −81pp, has its true values printed in-cell. Table 4.2 lists three representative layers; the full map shows the effect is pervasive and layer-conditional, with several categories (e.g. DET, PUNCT) flipping sign between shallow and deep layers rather than holding a fixed priority.

CHAPTER 5

Decomposing Routing into Interpretable Features

Problem formulation. Chapter 4 showed the router reads structured signal but not what that signal is. This chapter decomposes routing into SAE features (RQ4) and asks whether those features fire on syntactic or semantic cues (RQ5). Because the router is linear in its input (Section 3.1), a feature it can read must appear as a linear correlate of its score; we therefore rank the ∼16K SAE features at each layer by the Pearson correlation between feature activation and router score. Pearson is the appropriate ranker here precisely because of the router's linearity: a richer nonlinear statistic would credit structure the router cannot use.

5.1 Single-feature correlations

The null we test is that an SAE on routing-layer residuals yields no feature meaningfully correlated with the router score. That null is rejected, but weakly: the strongest single feature anywhere is feature 9153 at layer 24 with $r = -0.362$ ($r^2 \approx 0.13$), and only this one feature clears a pre-registered $|r| > 0.35$ bar. Every layer's top correlations sit far above the permutation-null ceiling ($|r| \approx 0.03$) but are individually modest. Taken on its own, no single feature governs routing.

This is the expected signature of *superposition*: a 2304-dimensional residual stores far more concepts than it has dimensions, so they pack into overlapping linear directions rather than each taking its own axis. A linear router summing over that residual integrates many of those directions by construction, so a single dominant feature is not expected. The correct test of RQ4 is therefore *collective*: how many interpretable features, taken together, recover the router's score.

5.2 Collective decomposition

We rank features by $|r|$ on a training split of tokens only, fit ridge regression on the top-k, and report the R^2 on a held-out split of tokens (split by document, so identification and evaluation never share a document). Ranking on the training split is essential: ranking on the full data and then evaluating "held-out" would leak the selection and inflate R^2. We also report a random-feature baseline (R^2 averaged over draws of k random valid features), because in a 16K-feature space any k features fit something; the meaningful quantity is how far selection beats random.

A compact selected set recovers about half of the held-out router-score variance and beats the random baseline by a wide margin (Table 5.1; Figure 5.1 traces the full R^2-vs-k curve). At $k = 200$, selected features reach $R^2 \approx 0.44$–0.49 across layers, roughly 0.30–0.40 above the random baseline. By contrast, fitting *all* features drives held-out R^2 negative at layers 23 and

24 (dense overfitting of ∼13K features on ∼16K tokens), which reinforces the point: the sparse selected set generalises where the dense fit does not.

Table 5.1: Held-out R^2 of ridge on the top-k features selected by training-split correlation (sel), versus k random features (rand), on the strict-long checkpoint. Selection beats random at every k.

Layer	k=10 sel/rand	k=50 sel/rand	k=200 sel/rand
13	0.28/0.02	0.41/0.03	0.49/0.12
23	0.22/0.01	0.32/0.03	0.44/0.10
24	0.25/0.01	0.39/0.03	0.49/0.09

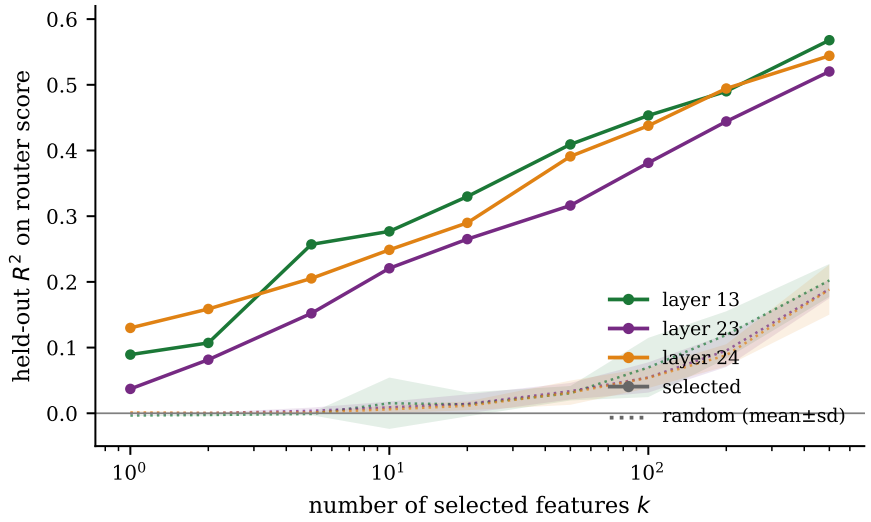

Figure 5.1: Held-out R^2 on the router score as a function of the number of selected features k (log axis), for the three studied layers (solid, selected by training-split correlation) against a random-feature baseline (dotted, mean±sd over draws). Selection separates from random within the first few features and reaches roughly half the score variance by $k \approx 200$, continuing to rise gradually thereafter; this is the basis for the "collection of ∼50–200 features" claim. That so many features are needed, while no single feature explains more than a small fraction, is why the decision is distributed rather than carried by any one feature.

The collective signal is not specific to one training. Repeating the $k = 200$ analysis on three independently trained checkpoints gives held-out R^2 in the range 0.49–0.72 at layer 13, 0.44–0.52 at layer 23, and 0.36–0.51 at layer 24, with selection beating the random baseline in every case. The strength of the collective decomposition therefore generalises across training configurations, even though the specific feature identities do not (Section 5.6).

5.3 Closed-form attribution and its ceiling

Ranking by correlation says which features track routing; it does not say how much each contributes to the actual logit. Because the router is linear and Gemma's RMSNorm has an exact, closed-form derivative, the per-feature contribution to the router logit can be computed directly rather than approximated by integrating gradients along a path, as general attribution methods require:

$$\mathrm{attr}(t, f) = z_{t,f} \, \frac{D_f \cdot \big((1+\gamma) \odot W_\ell\big)}{\mathrm{RMS}(h_t)},$$

where $z_{t,f}$ is the SAE activation of feature f at token t, D_f is its decoder direction, γ is the RMSNorm gain, and the reconstructed logit additionally includes the SAE decoder-bias term $b_{\text{dec}} \cdot ((1+\gamma) \odot W_\ell)$ (omitting this bias was an error identified during revision). We verified the implementation in two independent ways, computing the reconstructed logit through the attribution sum and through a direct decode, and obtained agreement to machine precision on all 32,808 tokens at each layer.

The reconstructed logit tracks the true logit well in *shape* but underestimates its *magnitude* (Figure 5.2): across layers the completeness correlation is 0.84–0.88 while the median ratio of reconstructed to true logit is 0.76–0.83. This shortfall is not an implementation error (the checks above rule that out); it is the part of the residual, in the direction the router reads, that the Gemma-Scope SAE does not reconstruct. It bounds every SAE-mediated value in this thesis: the SAE recovers the structure of routing but underestimates its magnitude by roughly a quarter.

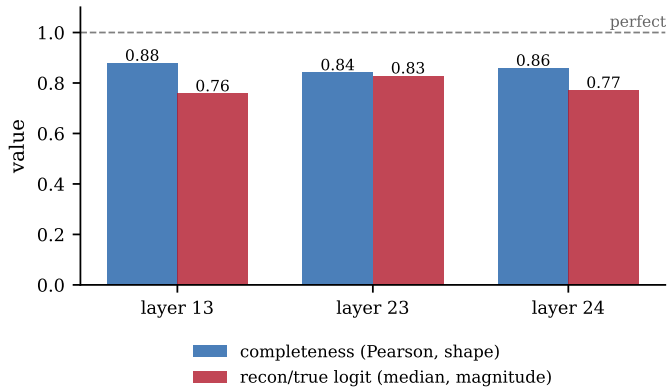

Figure 5.2: The SAE ceiling, per layer. The reconstructed router logit tracks the true logit's *shape* well (completeness Pearson 0.84–0.88, blue) but underestimates its *magnitude* (median ratio of reconstructed to true logit 0.76–0.83, red; 1.0 would be perfect). The shortfall is the component of the residual, in the direction the router reads, that Gemma-Scope does not reconstruct, and it sets a floor under every SAE-mediated value reported here.

5.4 The syntax–semantics frontier (RQ5)

To classify each top feature as syntactic or semantic we use *POS-retention*: the ratio of a feature's partial correlation with routing, after regressing out POS, to its raw correlation. A feature that maps cleanly onto one POS tag loses most of its correlation under this control (low retention, "POS-categorical"); a feature carrying content beyond POS keeps it (high retention, "POS-orthogonal"); the middle band is "mixed". Table 5.2 reports the composition of the top-200 features per layer.

Table 5.2: Composition of the top-200 routing-correlated features by POS-retention bin (strict-long), with the positive/negative correlation-sign split. Semantic (POS-orthogonal) features dominate at every layer.

Layer	POS-categorical	mixed	POS-orthogonal
13	22 (11.0%)	51 (25.5%)	127 (63.5%)
23	31 (15.5%, all +)	14 (7.0%)	155 (77.5%)
24	7 (3.5%)	54 (27.0%)	139 (69.5%)

The decomposition yields two findings. First, semantic features dominate (64–78% of the top-200 at every layer): routing depends predominantly on content that survives the POS control,

not on grammatical category. Second, the syntactic minority is structured rather than noise: at layer 23 all 31 POS-categorical features have *positive* correlation, meaning the router uses syntactic cues (punctuation, whitespace) only as *processing* signals, never as skip signals. The skip decision at the deeper layers is carried by semantic features.

5.5 An independent check on the semantic labels

POS-retention (Section 5.4) is an indirect measure: it tells us whether a feature's routing correlation reduces to a part-of-speech tag, but not what the feature is actually about. To check the proxy against an independent signal, and to describe the content the router reads in ordinary terms, we retrieve the automatically generated feature descriptions that Gemma-Scope features carry on Neuronpedia and read them across the top routing-correlated features at each layer.

The descriptions corroborate the statistical binning at the level of *kinds* of feature, which is the level our argument needs. The features that promote *skipping* at the deep layers are dominated by topical-content directions rather than grammatical ones: at layer 23 they describe domains such as law, finance, and politics, and at layer 24 they describe programming and medicine. The features that promote *processing* are, by contrast, dominated by quantitative content, numerals, measurements, and statistics, at both deep layers, alongside a narrative cluster (military and conflict description) at layer 23. Both groups are content categories, not parts of speech, which is independent evidence that the deep-layer routing decision is organised along a semantic axis. We report this as a distribution over content types rather than resting it on any individual feature, because no single feature carries the decision (Section 5.2) and individual labels are the least reliable part of this evidence (see below).

The labels carry two caveats. First, they are generated on Neuronpedia's reference corpus, not on the routing corpus we evaluate, so an individual label can describe a different usage than the one that drives routing here: a feature labelled as programming content, for instance, may fire predominantly on proper nouns in our WikiText data. We therefore treat the labels as corpus-general evidence about the *type* of content the router reads, not as definitive descriptions of single features, and we rely on each feature's per-corpus top-activating tokens for any concrete claim. Second, the labels are an external cross-check, not a measurement we control; their value is that an independent labelling, produced without reference to routing, lands on the same syntactic-versus-semantic split that our POS-retention analysis found.

5.6 Cross-checkpoint stability

Finally we ask whether the same features recur across independent trainings. The per-feature overlap is low (Jaccard $\approx$ 0.13–0.28 between top-50 sets; on this index 0 is no overlap and 1 identical sets, leaving only about 12–22 of the 50 features shared between any two trainings), so specific feature identities are largely training-specific. The *composition*, however, is stable at layer 23: all three checkpoints land within a few percentage points on every bin (POS-categorical $\sim$16–18%, POS-orthogonal $\sim$75–78%) and every POS-categorical feature is positive-sign in every checkpoint. Layers 13 and 24 vary more across training. So at layer 23 the router learns the same *kind* of feature population regardless of training trajectory, even though the individual features differ. We return to this distinction causally in Chapter 6.

Routing decomposes not into a single feature but into a collection of $\sim$50–200 SAE features that together explain about half the held-out router-score variance, robustly across trainings (RQ4). Those features are predominantly semantic, with a small, structured syntactic minority that acts only as a processing cue; the deep-layer skip decision is carried by content features (RQ5). Individual features are at best contextually monosemantic: each looks like a single clean concept within one corpus, but its label does not survive a change of corpus.

CHAPTER 6

Causal Intervention

Problem formulation. Chapter 5 is correlational: features co-vary with the router score. Correlation is consistent with two interpretations that have opposite consequences. Either the router genuinely reads a feature's direction (causal readout), in which case intervening on it should change routing predictably; or the feature merely co-varies with whatever the router actually reads (spurious correlate), in which case intervention does nothing. This chapter distinguishes them by intervening on features and measuring the change in the routing decision (RQ6).

6.1 Method

For each token we capture the residual entering a layer, encode it with the SAE, modify a chosen set of features, decode, and recompute the router score. We measure the *conditional flip rate*: the fraction of tokens, among those where a manipulated feature was active in baseline, whose binary routing decision changes; and the *direction rate*: the fraction of those flips that go in the predicted direction.

Two methodological choices are essential, both because SAE reconstruction is imperfect (Section 5.3). First, the counterfactual must run through the same SAE pipeline as the intervention. Comparing an SAE-decoded intervention against the *real* model score would fold reconstruction error (which alone flips $\sim$17% of decisions at the 0.5 threshold) into the measured effect; we instead compare against the SAE's own unmodified reconstruction, so the difference is purely the intervention. Second, a raw flip rate is not comparable across feature sets of different size or firing frequency, because zeroing many high-firing features moves the residual substantially regardless of relevance. We therefore compare every effect against a *frequency-matched null*: a same-size set of random features drawn from the same activation-frequency deciles, and report the margin above this null. We pre-registered the success criterion before running: a conditional flip rate $\geq$ 10pp with a direction rate $\geq$ 70%. Building on the interventionist criterion of Andersen et al. (2025), we additionally impose our own bidirectional-consistency requirement: suppressing and amplifying a feature set must move routing in opposite directions.

6.2 Causal manipulability of routing

Suppressing the collection of negative-sign (skip-cue) features pushes routing toward *process* at every studied layer, and the effect is large and correctly directed (Figure 6.1). The strongest single result is at layer 24: suppressing the negative-sign collective flips 30.6% of active-token decisions with a 100% direction rate, a margin of +27.4pp above the frequency-matched null. Amplifying the same collective moves routing the other way (toward skip), satisfying the bidirectional requirement. At the bin level, ablating the semantic (POS-orthogonal) features is causal at the deep layers (+13.9pp above null at layer 23, +22.1pp at layer 24), consistent with the correlational finding that the skip decision there is semantic.

We also test, non-circularly, whether a feature's correlation sign predicts its causal direction. Among features that fire often enough to be measured, the correlation sign predicts the causal direction for 76–96% of features per layer (96% at layer 13, 88% at layer 23, 76% at layer 24). Correlation is therefore not equivalent to causation, but for high-firing features it predicts causal direction reliably, the strongest evidence for the Chapter 5 method.

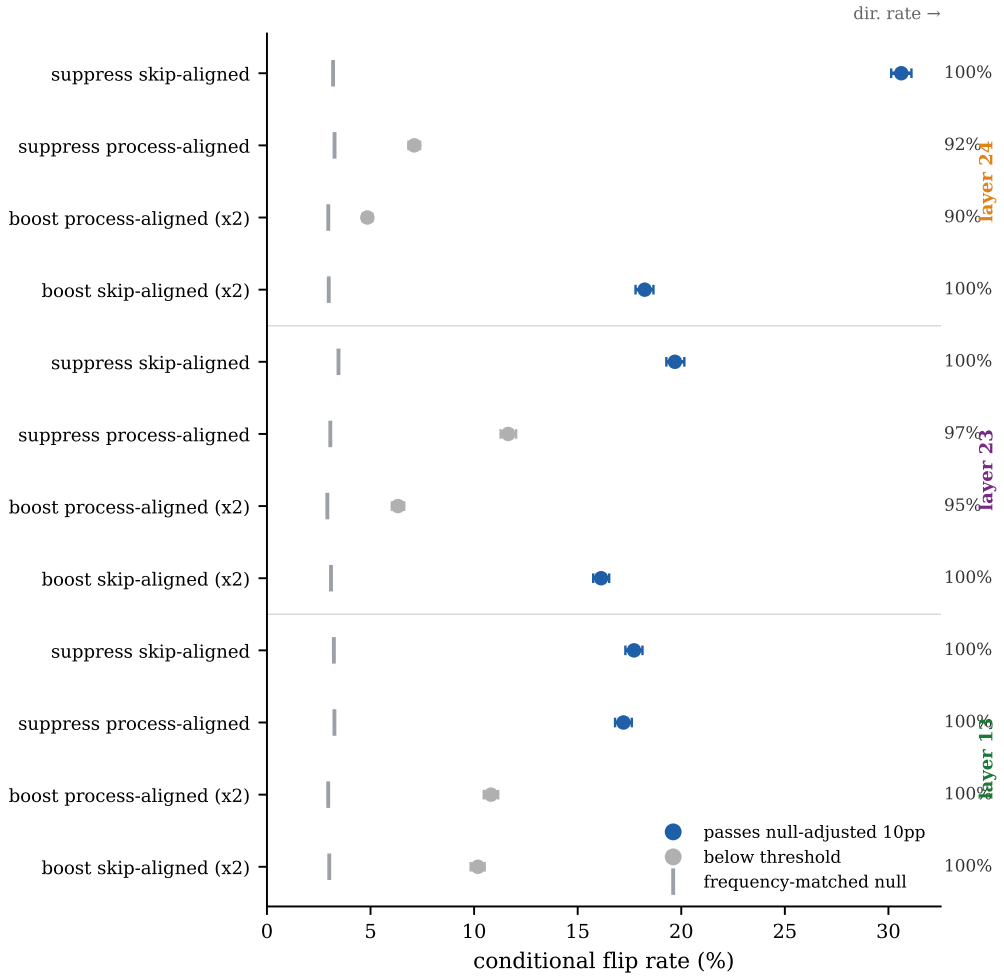

Figure 6.1: Causal effect of the collective interventions, grouped by layer. Each point is the conditional flip rate (fraction of active-token decisions that change) with its 95% CI; the grey tick is the frequency-matched null; the right-margin percentage is the direction rate (share of flips in the predicted direction). Conditions that clear the null-adjusted 10pp bar are blue. The pattern is consistent across layers: suppressing or boosting the *skip-aligned* collective is the reliable causal handle, peaking at layer 24's suppress-skip (30.6%, 100% directed), whereas the process-aligned designs are weaker. Comparing the effect (point) against the null (tick) is what distinguishes a real handle from one expected by chance at that firing frequency.

6.3 Cross-checkpoint causality: stable strength, training-specific direction

Repeating the semantic-bin intervention at layer 23 across the three checkpoints dissociates the causal handle's strength from its direction (Table 6.1; Figure 6.2 plots strength against direction). The *strength* of the causal handle is nearly identical across trainings, with null-adjusted margins of +13.9, +15.9, and +16.3pp (a spread of 2.4pp). The *direction*, however, is training-specific:

the same intervention is cleanly directed in **strict-long** (88%), reversed in **strict-short** (36%), and ambiguous in **gentle-short** (58%). This mirrors the composition finding of Section 5.6: the router reliably acquires a causal handle of similar strength on the same *kind* of feature at layer 23 regardless of training, but the *sign* with which it uses that handle is set by the training trajectory. To our knowledge this magnitude-invariant, direction-specific pattern has not been reported before.

Table 6.1: Semantic-bin (POS-orthogonal) ablation at layer 23 across three checkpoints. The null-adjusted margin (effect minus frequency-matched null) is stable; the direction rate is not.

Checkpoint	conditional flip	direction	margin vs. null
strict-long	17.8%	88.1%	+13.9pp
strict-short	19.5%	35.6%	+15.9pp
gentle-short	19.9%	57.5%	+16.3pp

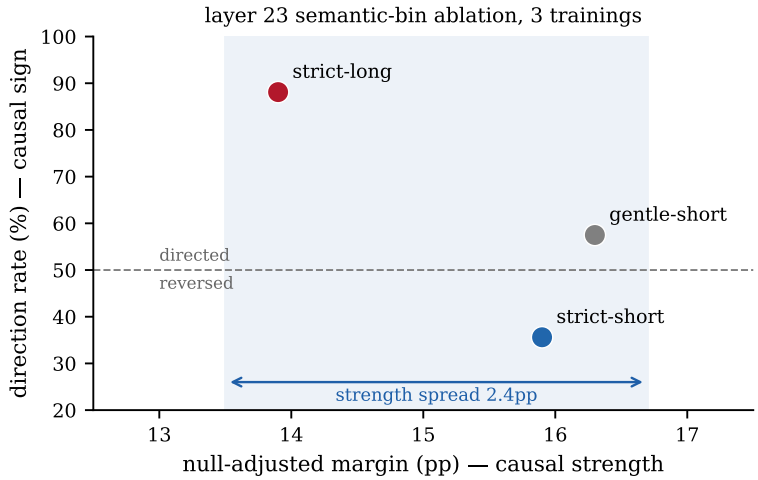

Figure 6.2: Strength versus direction of the semantic-bin intervention at layer 23, across three independent checkpoints. The null-adjusted margin (horizontal, the *strength* of the causal handle) is nearly constant, while the direction rate (vertical, *how* the router uses that handle) ranges from cleanly directed to reversed. The two axes separating, stable strength with training-specific direction, is the magnitude-invariant, direction-specific pattern reported in Table 6.1.

6.4 Held-out replication

The causal margins above were measured on the same WikiText-2 tokens used to *identify* the feature collective, by correlation with the router score. This invites a double-dipping worry: features chosen because they correlate on these tokens might show an inflated causal effect when intervened on the same tokens. We test it with the causal analogue of a train–test split. We divide the corpus into two disjoint document halves, identify the negative-sign collective on one half, suppress it on the other, and compare the resulting null-adjusted margin against the in-corpus case (identify and test within the same half). The split is at the document level, so no token used to select a feature is also used to test it.

The held-out margins match the in-corpus margins almost exactly (Table 6.2): retention is 97–99% at all three layers, and at layer 24 the held-out margin is +27.5pp against an in-corpus +27.8pp, with a 100% direction rate. The effect is therefore not an artefact of selecting and testing on the same tokens. This rules out token-level selection bias within the WikiText

distribution; it is not a claim of transfer to a different corpus. The two halves select overlapping but non-identical collectives (Jaccard 0.53–0.70), so, as with the training-specific feature identities of Section 5.6, the *specific* features differ while the same *kind* of collective delivers the same causal strength. Were the margin an overfitting artefact, the held-out value would have collapsed rather than held.

Table 6.2: Held-out replication of the negative-sign collective suppression. The *in-corpus* margin identifies and tests on the same document half (averaged over both halves); the *held-out* margin identifies on one half and tests on the disjoint other (averaged over both directions). Margins are null-adjusted (conditional flip rate minus the frequency-matched null); the direction rate was 100% in every cell.

Layer	in-corpus margin	held-out margin	retention
13	+14.2pp	+13.8pp	97%
23	+17.0pp	+16.8pp	99%
24	+27.8pp	+27.5pp	99%

6.5 A single-feature case

The one feature that cleared the $|r| > 0.35$ correlational bar, feature 9153 at layer 24, also behaves causally: suppressing it alone flips 12.0% of active-token decisions with a 100% direction rate. We report this as an illustration of the full correlation-to-causation chain on a single feature, not as a central result; the collective decomposition and intervention are the substantive result, and a single marginal-r feature is fragile by comparison.

6.6 Downstream effects of forced routing decisions

The interventions so far show that SAE features causally drive the routing *decision*; they leave open whether the decision matters for the model's *output*. We test this directly and without the SAE, forcing the binary gate through the model's override hook and comparing language-modelling loss and the next-token KL divergence (the shift in the predicted next-token distribution; 0 when the two predictions are identical, larger the more the forced decision moves them) against the model's own unperturbed decisions on 80 WikiText-2 documents (∼11.9K tokens). Because attention is computed before the gate masks it (Section 3.2), this counterfactual is exact: forcing a token to process or skip adds or drops precisely the attention already computed, with no recomputation and no reconstruction error.

The decisions are consequential (Table 6.3, layer 24). Forcing all tokens to skip at one layer raises perplexity by 7.8–11.4% (the range over the three layers tested, each ablated on its own) and forcing all to process lowers it by 8.5–12.5%, and flipping a growing random fraction perturbs the output monotonically (mean KL 0.010 to 0.080). What this does *not* establish is that the router picks the right *tokens* rather than the right *number*: a capacity-matched swap, holding the processed-token count fixed, perturbs the distribution about as much as a 75% flip (mean KL 0.077 against 0.080) yet moves perplexity by only −1.6%, with inconsistent sign across layers. We therefore claim that the routing decision is consequential, but not that the router's specific token selection is optimal; that question is confounded by token position and left open.

6.7 Retracted effects

Applying the frequency-matched null removed several effects that passed the raw 10pp/70% bar. Ablating the syntactic (POS-categorical) bin at layer 24 reaches 11.5% raw conditional flip but

Table 6.3: Downstream effect of forcing the routing decision at layer 24 (the strongest causal layer), measured against the model's own unperturbed decisions on 80 WikiText-2 documents. ΔPPL is the change in perplexity; mean KL is the per-token next-token divergence from baseline. Forcing more processing helps and more skipping hurts; random flips perturb the output in proportion to their number (their perplexity falls because, with a majority of tokens skipped, a random flip adds net processing); the capacity-matched swap moves the distribution as much as a 75% flip yet barely changes perplexity, isolating *how many* tokens are processed from *which* ones.

Intervention	gates changed	ΔPPL	mean KL
Process all	6 840	−12.5%	0.063
Skip all	5 022	+7.8%	0.045
Flip random 10%	1 186	−0.9%	0.010
Flip random 25%	2 964	−2.0%	0.027
Flip random 50%	5 928	−2.8%	0.052
Flip random 75%	8 898	−4.7%	0.080
Capacity-matched swap	8 562	−1.6%	0.077

only +9.6pp above the null, below our threshold; we therefore do *not* claim it as causal. Two amplification conditions at layer 13 and one suppression condition at layer 23 fail the same way.

Routing is causally manipulable through the SAE feature collection, in the predicted direction, at every studied layer; the strongest handle is on the semantic skip-cue features at the deep layers; and the strength of the deep-layer causal handle is stable across trainings while its direction is not.

CHAPTER 7

Discussion

7.1 Summary of findings

The thesis traces a single line of argument. Chapter 4 establishes that the router reads architecturally and linguistically structured signal, not a low-level scalar cue. Chapter 5 shows that this signal decomposes into a collection of SAE features, predominantly semantic, that together explain about half the router's score. Chapter 6 shows that these features are not merely correlated with routing but causally drive it. The overall picture is of a router that makes a distributed, mostly content-based decision, contradicting the implicit assumption in efficiency-oriented MoD work that token importance reduces to simple heuristics.

7.2 Limitations

All results are mediated by the SAE. All RQ4–RQ6 results describe the router as seen through the Gemma-Scope SAE, which fails to reconstruct ∼17% of binary decisions and underestimates logit magnitude by roughly a quarter (Section 5.3). The SAE was trained on base Gemma, not on our MoD-tuned activations, so its dictionary may not be the ideal basis for the tuned model.

Single model and the parity confound. We study one model (Gemma 2 2B) and one SAE suite. Moreover, in Gemma 2 every full-attention layer has an odd index and every sliding-window layer an even one, so the routing-rate difference of Chapter 4 is equally consistent with a layer-parity effect; separating them requires a model where the two vary independently. The comparison also aggregates layers that each train their own router, so it contrasts routers at the two layer classes rather than showing that any router represents attention type. The per-token findings of Chapters 5 and 6 are not subject to this confound, as each lives within a single layer.

Scope of the intervention results. The intervention identifies features and intervenes within the same WikiText-2 distribution. The held-out replication of Section 6.4 shows the causal margins survive when features are selected and tested on disjoint document halves, so the effect is not an artefact of sharing tokens; it does not, however, establish transfer to a different corpus, which remains open. The downstream measurement of Section 6.6 shows the manipulated decisions are consequential for the model's output, but at fixed capacity we could not isolate whether the router selects the individually optimal tokens.

Collective, not monosemantic. The interpretability we find is population-level. Individual top features are at best contextually monosemantic: their clean labels do not survive a change of corpus (Section 5.5), an instance of the feature-splitting problem catalogued by Sharkey et al. (2025).

7.3 Dual-use posture

The intervention is bidirectional: the machinery that could keep safety-relevant tokens from being skipped could also push them out of computation. We have demonstrated the mechanism only on general-domain text and have not weaponised it on safety-critical inputs. We consider the appropriate posture to be open publication of the analysis method together with this explicit statement of its dual-use nature.

7.4 Work in progress and future work

Several extensions are planned, in roughly descending priority. We list them explicitly so the boundary of the present results is clear.

Cross-architecture replication.
Repeat the analysis on a second model (e.g. Qwen 2.5 1.5B) to test generality and to disentangle the attention-type preference from the layer-parity confound.

In-domain SAE.
Train an SAE on MoD-tuned activations to reduce the reconstruction ceiling (Section 5.3) that currently bounds every SAE-mediated number.

7.5 Ethical considerations

The work involves no human participants and therefore falls outside the scope of the FNWI ethical compliance procedure for student projects involving human participants. The research artefacts (a trained router, a public-domain evaluation corpus, and analysis scripts) pose no direct harm vector. One indirect consideration merits acknowledgement: the intervention toolkit developed for RQ6 is bidirectional by construction. The same machinery that could be used to keep safety-relevant tokens from being skipped could also be used to push them out of computation. We note that the present work demonstrates the mechanism only on general-domain text and does not test it on safety-critical inputs; the dual-use posture is discussed in Section 7.3.

CHAPTER 8

Conclusion

MoD routing decisions in Gemma 2 2B are not the output of a simple token-importance heuristic. They differ between full- and sliding-window layers and depend on part-of-speech in a layer-conditional way (Chapter 4); they decompose into a collection of roughly 50–200 sparse-autoencoder features that together explain about half the router's score on held-out tokens, predominantly semantic rather than syntactic (Chapter 5); and those features causally drive routing, with a deep-layer causal handle whose strength is stable across independent trainings but whose direction is training-specific (Chapter 6). The decision is distributed and mostly content-based. Beyond the empirical findings, the thesis contributes a measurement protocol for SAE-mediated intervention, pairing a reconstruction-matched counterfactual with a frequency-matched null, that we expect to be reusable wherever SAE features are intervened on. Within the thesis, the causal effect is shown to replicate on a held-out corpus split (Section 6.4) and the manipulated decisions are shown to be consequential for the model's output (Section 6.6); the planned extensions in Section 7.4 would broaden these findings beyond a single model and SAE suite.

APPENDIX A

Companion workshop paper

The three behavioural research questions (RQ1–RQ3) were investigated in a companion workshop paper, "What Do Mixture-of-Depth Routers Learn? Routing Patterns in Gemma 2", accepted as a poster at the Mechanistic Interpretability Workshop at ICML 2026. The paper is available at `https://openreview.net/forum?id=5HXw5MISXk`.

APPENDIX B

The gentle-long checkpoint

gentle-long ($\lambda = 0.01$, 52K×3) is one of the four trained checkpoints (Table 3.1) but is used only in the attention-type analysis of Chapter 4; the feature-level decomposition (Chapter 5) and intervention (Chapter 6) use the other three. This appendix records the per-layer routing audit behind that choice.

The capacity penalty is asymmetric: $\lambda \,\mathrm{ReLU}(\bar{c} - c^*)$ penalises the batch routing rate $\bar{c}$ only above the target $c^* = 0.5$ and supplies no gradient at or below it. Under the gentle penalty ($\lambda = 0.01$) with extended training (52K×3), the language-modelling loss dominates this weak signal and the router converges to processing most tokens rather than to the target.

Auditing the per-layer routing rate of all four checkpoints makes the consequence concrete. gentle-long processes 76% of tokens overall (the densest of the four), and at the three layers analysed in Chapters 5 and 6 it processes 54% (layer 13), 74% (layer 23), and 70% (layer 24), against roughly balanced rates elsewhere (for example strict-long: 44%, 59%, 47%). The feature-level analyses correlate SAE features with the per-token skip/process decision and intervene to flip it; both require that decision to vary across tokens. At layers 23 and 24 gentle-long skips too few tokens to supply that variation, so it is not carried into those analyses.

We read this as a training-dynamics observation rather than a failed run: under gentle capacity pressure with extended training, token-level Router-Tuning on Gemma 2 2B drifts toward a dense routing regime. gentle-long remains a valid data point in Chapter 4, where the attention-type comparison is computed within each layer and does not depend on a balanced overall skip rate.

Bibliography

Andersen, Mette Friis, Maria Heuss and Ana Lucic (2025). 'Three Desiderata for Faithfulness in Machine Learning Explanations: The Case for Causal Abstraction'. In: *Mechanistic Interpretability Workshop at NeurIPS 2025*. URL: https://openreview.net/forum?id=b0rXC856Sm.

Antoine, Elie, Frédéric Béchet and Philippe Langlais (2024). *Part-Of-Speech Sensitivity of Routers in Mixture of Experts Models*. arXiv: 2412.16971 [cs.CL]. URL: https://arxiv.org/abs/2412.16971.

Bricken, Trenton, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly et al. (2023). 'Towards Monosemanticity: Decomposing Language Models With Dictionary Learning'. In: *Transformer Circuits Thread*. URL: https://transformer-circuits.pub/2023/monosemantic-features.

Cunningham, Hoagy, Aidan Ewart, Logan Riggs, Robert Huben and Lee Sharkey (2023). *Sparse Autoencoders Find Highly Interpretable Features in Language Models*. arXiv: 2309.08600 [cs.LG]. URL: https://arxiv.org/abs/2309.08600.

Gadhikar, Advait, Souptik Kumar Majumdar, Niclas Popp, Piyapat Saranrittichai, Martin Rapp and Lukas Schott (2024). *Attention Is All You Need For Mixture-of-Depths Routing*. arXiv: 2412.20875 [cs.CV]. URL: https://arxiv.org/abs/2412.20875.

He, Shwai, Tao Ge, Guoheng Sun, Bowei Tian, Xiaoyang Wang and Dong Yu (2025). *Router-Tuning: A Simple and Effective Approach for Enabling Dynamic-Depth in Transformers*. arXiv: 2410.13184 [cs.CL]. URL: https://arxiv.org/abs/2410.13184.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem and Adriane Boyd (2020). *spaCy: Industrial-strength Natural Language Processing in Python*. https://doi.org/10.5281/zenodo.1212303. DOI: 10.5281/zenodo.1212303.

Jiang, Albert Q., Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix and William El Sayed (2024). *Mixtral of Experts*. arXiv: 2401.04088 [cs.LG]. URL: https://arxiv.org/abs/2401.04088.

Laitenberger, Filipe, Dawid Jan Kopiczko, Cees G. M. Snoek and Yuki M Asano (2026). 'What Layers When: Learning to Skip Compute in LLMs with Residual Gates'. In: *The Fourteenth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=aiP6XfaYZR.

Lasy, Ilya, Nora Yinuo Cai and Kola Ayonrinde (2026). 'RouterInterp: Understanding Superposed Specialisation in MoE Routing'. In: *Workshop on Scientific Methods for Understanding Deep Learning*. URL: https://openreview.net/forum?id=9a5i2vyMwN.

Lawson, Tim and Laurence Aitchison (2025). *Learning to Skip the Middle Layers of Transformers*. arXiv: 2506.21103 [cs.LG]. URL: https://arxiv.org/abs/2506.21103.

Lieberum, Tom, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah and Neel Nanda (2024). *Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2*. arXiv: 2408.05147 [cs.LG]. URL: https://arxiv.org/abs/2408.05147.

Luo, Xuan, Weizhi Wang and Xifeng Yan (2025). *Adaptive Layer-skipping in Pre-trained LLMs*. arXiv: 2503.23798 [cs.CL]. URL: https://arxiv.org/abs/2503.23798.

Luo, Yaxin, Gen Luo, Jiayi Ji, Yiyi Zhou, Xiaoshuai Sun, Zhiqiang Shen and Rongrong Ji (2025). '$\gamma-$MoD: Exploring Mixture-of-Depth Adaptation for Multimodal Large Language Models'. In: *The Thirteenth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=q44uq3tc2D.

Marks, Samuel, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau and Aaron Mueller (2025). *Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models*. arXiv: 2403.19647 [cs.LG]. URL: https://arxiv.org/abs/2403.19647.

Merity, Stephen, Caiming Xiong, James Bradbury and Richard Socher (2016). *Pointer Sentinel Mixture Models*. arXiv: 1609.07843 [cs.CL].

Michel, Paul, Omer Levy and Graham Neubig (2019). *Are Sixteen Heads Really Better than One?* arXiv: 1905.10650 [cs.CL]. URL: https://arxiv.org/abs/1905.10650.

Peng, Kenny, Rajiv Movva, Jon Kleinberg, Emma Pierson and Nikhil Garg (2025). *Use Sparse Autoencoders to Discover Unknown Concepts, Not to Act on Known Concepts*. arXiv: 2506.23845 [cs.LG]. URL: https://arxiv.org/abs/2506.23845.

Pleiter, Stein, Ege Erdogan and Ana Lucic (2026). 'What Do Mixture-of-Depth Routers Learn? Routing Patterns in Gemma 2'. In: *Mechanistic Interpretability Workshop at ICML 2026*. URL: https://openreview.net/forum?id=5HXw5MISXk.

Raposo, David, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys and Adam Santoro (2024). *Mixture-of-Depths: Dynamically allocating compute in transformer-based language models*. arXiv: 2404.02258 [cs.LG]. URL: https://arxiv.org/abs/2404.02258.

Sharkey, Lee, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet and Tom McGrath (2025). *Open Problems in Mechanistic Interpretability*. arXiv: 2501.16496 [cs.LG]. URL: https://arxiv.org/abs/2501.16496.

Shi, Wei, Sihang Li, Tao Liang, Mingyang Wan, Guojun Ma, Xiang Wang and Xiangnan He (2025). *Route Sparse Autoencoder to Interpret Large Language Models*. arXiv: 2503.08200 [cs.LG]. URL: https://arxiv.org/abs/2503.08200.

Tan, Zhen, Daize Dong, Xinyu Zhao, Jie Peng, Yu Cheng and Tianlong Chen (2024). *DLO: Dynamic Layer Operation for Efficient Vertical Scaling of LLMs*. arXiv: 2407.11030 [cs.LG]. URL: https://arxiv.org/abs/2407.11030.

Team, Gemma et al. (2024). *Gemma 2: Improving Open Language Models at a Practical Size*. arXiv: 2408.00118 [cs.CL]. URL: https://arxiv.org/abs/2408.00118.

Tenney, Ian, Dipanjan Das and Ellie Pavlick (July 2019). 'BERT Rediscovers the Classical NLP Pipeline'. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 4593–4601. DOI: 10.18653/v1/P19-1452. URL: https://aclanthology.org/P19-1452/.

Voita, Elena, David Talbot, Fedor Moiseev, Rico Sennrich and Ivan Titov (2019). *Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned*. arXiv: 1905.09418 [cs.CL]. URL: https://arxiv.org/abs/1905.09418.

Xiao, Guangxuan, Yuandong Tian, Beidi Chen, Song Han and Mike Lewis (2024). *Efficient Streaming Language Models with Attention Sinks*. arXiv: 2309.17453 [cs.CL]. URL: https://arxiv.org/abs/2309.17453.

Zhang, Chen, Meizhi Zhong, Qimeng Wang, Xuantao Lu, Zheyu Ye, Chengqiang Lu, Yan Gao, Yao Hu, Kehai Chen, Min Zhang and Dawei Song (2024). *MoDification: Mixture of Depths Made Easy*. arXiv: 2410.14268 [cs.CL]. URL: https://arxiv.org/abs/2410.14268.

Zoph, Barret, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer and William Fedus (2022). *ST-MoE: Designing Stable and Transferable Sparse Expert Models*. arXiv: 2202.08906 [cs.CL]. URL: https://arxiv.org/abs/2202.08906.