Cross-Domain Circuit Analysis of Factual Knowledge Retrieval in LLMs

Cross-Domain Circuit Analysis of Factual Knowledge Retrieval in Large Language Models

Authors: Joseph Lawrence, Konstantinos Krampis

Abstract

We characterize the cross-domain organization of factual knowledge circuits in two language models, Gemma-2-2B and Qwen3-4B, by analyzing 60 attribution graphs spanning chemistry, geography, and history. We introduce **traceback graphing**, a priority-guided backward search (best-first search) with geometric decay (factor 0.8) that identifies bottleneck features where ≥60% of paths converge.

Per-layer activation profiling reveals that architecture dominates knowledge domain. Within-model cosine similarity is 0.978 while between-model similarity is only 0.696; the architecture gap is approximately 14× the domain gap. Gemma concentrates 54% of total activation magnitude in layers 0–12 (front-loaded); Qwen places only 31% in the corresponding first half (back-loaded). Both architectures achieve comparable factual recall accuracy.

In Gemma, the share of total activation magnitude at the bottleneck layer L6 negatively correlates with output confidence (Spearman r = –0.684, Bonferroni-significant), while activation at post-bottleneck layers L13 and L16 positively predicts confidence (r ≈ +0.60). We call this correlation the bottleneck penalty: circuits that concentrate more of their activation at the bottleneck layer produce less confident predictions. We interpret it through information bottleneck theory (Tishby & Zaslavsky, 2015): over-compression at intermediate layers reduces information available for downstream evidence accumulation.

Cross-circuit analysis identifies 6 universal bottleneck features in Gemma and 15 in Qwen, with zero cross-architecture overlap. Universal features function as architecture-specific routing infrastructure rather than shared knowledge stores. Output-layer features converge across same-domain circuits up to 4.4× more than path-level features (Qwen history: 67.3% vs 15.3% Jaccard), evidencing convergent prediction vocabulary built through divergent internal routes.

Causal validation via 80 steering experiments reveals a three-tier dissociation between structural importance and behavioral effect. Essential-pathway features produce the strongest distributional perturbations (mean Kullback–Leibler (KL) divergence = 1.448 between steered and baseline next-token distributions), but 94.1% circuit redundancy absorbs them before the output layer. Output determinism, not feature selection, governs text-level susceptibility. A format-variation experiment shows that prompt template explains substantial within-domain overlap (Jaccard 0.49 vs fact-only 0.31), but architectural dominance of layer activation profiles is robust to this confound.

These results indicate that factual knowledge retrieval is constrained by architectural processing strategy across the full layer-by-layer activation profile, that confident predictions are associated with reduced bottleneck compression, and that no single structural metric on the attribution graph reliably predicts behavioral causal influence.

Keywords: mechanistic interpretability, attribution graphs, circuit tracing, sparse dictionaries, transcoders, bottleneck features, factual recall, information bottleneck theory, activation steering

1. Introduction

1.1 Motivation

Large language models encode vast factual knowledge, yet how this knowledge is organized at the circuit level remains poorly understood. When a model predicts that the chemical symbol for gold is "Au" or that the capital of Japan is "Tokyo," which internal features are responsible?

A central question is whether different knowledge types (symbolic mappings, geographic relations, temporal facts) use separate specialized circuits or share common computational infrastructure. Both possibilities are consistent with the existing mechanistic-interpretability literature, and the question has been studied only piecemeal.

The answer matters for model editing and targeted intervention. If factual knowledge is organized by domain, interventions must be domain-specific. If circuits share universal bottleneck features, a small number of features could provide efficient intervention points across multiple knowledge types.

1.2 Research Questions

1. **Within-domain convergence**: Do prompts querying the same type of knowledge (e.g., chemical symbols) share circuit architecture?

2. **Cross-domain divergence**: Do different knowledge types use different circuit features or layer depths?

3. **Universal bottlenecks**: Are there features that participate in circuits across all knowledge domains?

4. **Architecture vs. domain**: Is circuit structure more determined by model architecture or by the type of knowledge being retrieved?

1.3 Contributions

We present six contributions:

1. **Systematic cross-domain circuit analysis**: The first large-scale comparison of attribution circuits across three knowledge domains on two model architectures (60 circuits total), with statistical validation via permutation tests ($p < 0.0001$).

2. **Architecture-dominance finding**: Demonstration that bottleneck depth is strongly predicted by model architecture, not knowledge domain. Gemma consistently uses early bottlenecks (L5-7) and Qwen uses late bottlenecks (L22-25) regardless of whether the task involves chemistry, geography, or history. Cross-model Mann-Whitney tests confirm 13/16 structural metrics differ significantly (Bonferroni-corrected).

3. **Universal bottleneck identification**: Discovery of 6 universal bottleneck features in Gemma and 15 in Qwen that participate in circuits across all three knowledge domains, with zero overlap between architectures.

4. **Output-path convergence dissociation**: Finding that same-domain circuits share output features far more than intermediate features (up to 4.4× ratio), revealing convergent predictions from divergent internal processing.

5. **Confidence regression model**: A three-predictor regression model explains 49% of Gemma prediction confidence variance using total activation magnitude, edge weight kurtosis, and circuit size ($F(3,26) = 8.30$, $p < 0.001$). Total activation magnitude is the strongest single predictor ($r = 0.527$, Bonferroni-surviving). Bottleneck position and convergence do not significantly predict confidence, constraining "bottleneck quality" explanations.

6. **Three-tier causal dissociation**: 80 steering experiments reveal that essential-pathway features produce strong distributional perturbations (mean KL = 1.448) but neither pathway topology nor cross-circuit frequency predicts text-level output changes. Circuit redundancy absorbs perturbations, and output determinism governs steering susceptibility.

2. Related Work

Mechanistic Interpretability

The field of mechanistic interpretability seeks to understand neural network behavior by identifying interpretable computational circuits (Olah et al., 2020; Elhage et al., 2021). Recent work has identified specific circuits for tasks including indirect object identification (Wang et al., 2023) and factual recall (Meng et al., 2022). Our work extends this line of research to systematic cross-domain comparison.

Sparse Dictionaries and Transcoders

Sparse autoencoders (SAEs) decompose neural network activations into a sparse dictionary of interpretable features (Cunningham et al., 2024; Bricken et al., 2023). A standard SAE is trained to reconstruct a layer's activations in place, with a sparse hidden bottleneck, so its features describe what is present in the residual stream at that point.

Transcoders are a method in the same SAE family but with a different target. Instead of reconstructing one set of activations in place, a transcoder approximates the input-to-output function of an MLP sublayer through a wide, sparsely-activating hidden layer (Dunefsky et al., 2024). Because each transcoder feature maps inputs to outputs across the MLP, the features support cross-MLP circuit tracing: attribution edges drawn through transcoders correspond to influence that passes through the MLP computation, which is what makes them suitable for circuit analysis. This work uses the Gemma Scope transcoder suite (Lieberum et al., 2024), whose sparse hidden units use JumpReLU activations, for Gemma, and the Neuronpedia-hosted transcoder for Qwen.

Circuit Tracing and Path Extraction

Attribution-graph methods build a replacement model in which MLP sublayers are approximated by sparse dictionaries (here, transcoders) and attention is held fixed, then trace influence backward from the output through the resulting feature network to recover the computational graph behind a prediction (Ameisen, Lindsey, et al., 2025; Lindsey, Gurnee, Ameisen, et al., 2025). Given such a graph, a further step extracts the most important paths or routes through it; Ferrando and Voita (2024) extract information-flow routes by pruning a graph to the edges that carry the prediction. A related but causally-driven line of work discovers circuits by intervention rather than attribution: ACDC searches for the subgraph whose ablation preserves behavior (Conmy et al., 2023), and edge attribution patching approximates the same objective at much lower cost (Syed, Rager, & Conmy, 2024). This backward-attribution lineage descends from earlier feature-attribution methods that propagate relevance from outputs back to inputs (Bach et al., 2015; Shrikumar et al., 2017; Sundararajan et al., 2017).

Feature Steering

Prior work has demonstrated that clamping SAE features can steer model outputs (Turner et al., 2023; Templeton et al., 2024). Our identification of universal bottleneck features provides new candidates for cross-domain steering interventions.

Information Bottleneck Theory

Information bottleneck theory (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017) formalizes the tradeoff between representational compression and predictive accuracy in deep networks. The framework posits that each layer L computes a representation T_L of the input X, and that the mutual information I(X; T_L) bounds the task-relevant information downstream layers can use.

Over-compression at intermediate layers reduces this available information, limiting prediction quality. We draw on this framework in §5.5.3 to interpret our "bottleneck penalty" finding, in which activation concentration at the bottleneck layer negatively correlates with output confidence.

Neuronpedia Platform

Neuronpedia (Decode Research) provides a public platform for exploring sparse-dictionary and transcoder features with attribution graph generation capabilities. Our pipeline builds on Neuronpedia's Circuit Tracer API, which implements the circuit-tracing method of Ameisen, Lindsey, et al. (2025) and generates directed graphs showing how transcoder features connect from input tokens to output predictions.

3. Methods

3.1 Attribution Graph Generation

Models: Gemma-2-2B (`gemma-2-2b` on Neuronpedia, 26 transformer layers) and Qwen3-4B (`qwen3-4b`, 36 layers). The feature nodes in our circuits are transcoder features, not standard SAE features. For Gemma we use the Gemma Scope transcoder suite (`gemmascope-transcoder-16k`, 16,384 features per layer; Lieberum et al., 2024); for Qwen we use the Neuronpedia-hosted transcoder (`transcoder-hp`) for Qwen3-4B. <!-- TODO: confirm transcoder-hp provenance/citation -->

Attribution graphs were generated via the Neuronpedia `/api/graph/generate` endpoint with default parameters: `maxFeatureNodes=3000`, `desiredLogitProb=0.95`, `nodeThreshold=0.8`, `edgeThreshold=0.85`. Graphs typically contain 1,000–1,500 nodes and 40,000–80,000 edges.

From transcoder feature to activation magnitude. The quantity analyzed throughout this paper, activation magnitude, follows from the transcoders by a short chain of steps:

1. Each layer's MLP sublayer is approximated by a transcoder with 16,384 features (Gemma); a feature is a sparse hidden unit of that input-to-output approximation.

2. For a given prompt, a feature i fires with a non-negative scalar activation a_i (zero when the feature is inactive).

3. Attribution-graph generation, Neuronpedia's implementation of circuit tracing (Ameisen, Lindsey, et al., 2025), builds a replacement model in which the MLPs are replaced by their transcoders and attention is held frozen, and computes a directed graph G = (V, E) whose nodes are the active transcoder features and whose edges are attributions between features through the replacement model. Each node carries its activation a_i and its contribution to the output logits.

4. "Activation magnitude" used throughout the paper is |a_i|. The per-layer activation magnitude at layer L is the sum of |a_i| over the circuit's features at L; the total activation magnitude is the sum across all layers.

5. The per-layer activation share is the per-layer value divided by the total, and the vector of per-layer shares across all layers is the layer activation profile.

The resulting directed graph G = (V, E) therefore has: **nodes** as transcoder features identified by layer and feature index (e.g., L5_F7993995); **edges** that represent the attribution (direct influence) between features computed by Neuronpedia's circuit-tracing implementation through the replacement model with attention frozen (Ameisen, Lindsey, et al., 2025), not attention weights; **node attributes** including activation magnitude and contribution to output logits; **edge attributes** giving the attribution weight.

3.2 Traceback Graphing Algorithm

Our core analytical method is **traceback graphing**: a priority-guided backward search (best-first search) from output predictions through intermediate layers to identify critical paths and bottleneck features. The search expands the highest-scoring node first (the priority queue in the algorithm below), rather than exploring all nodes at a given depth before descending, so it is a best-first rather than a breadth-first traversal.

Traceback graphing composes standard components and applies them to a new target. It operates on circuit-tracing attribution graphs (Ameisen, Lindsey, et al., 2025) and is closest in spirit to information-flow-route

extraction (Ferrando & Voita, 2024), which similarly prunes an attribution graph to its most important paths. Its bottleneck definition, a feature through which a high fraction of traced paths converge, is a path-membership relative of betweenness centrality (Freeman, 1977). The traversal and the centrality intuition are not themselves novel; the contribution here is the path-convergence bottleneck definition (convergence ≥ 0.6) and its use for the cross-domain comparison that follows.

Algorithm:

```
Input:  Converted attribution graph G = (V, E)
        Top-K output nodes to trace (default: K=5)
        Decay factor d (default: 0.8)
        Max depth D (default: 20)

Output: Critical paths P, bottleneck features B

1. Identify final-layer nodes, sorted by contribution to output logits
2. For each of the top-K final nodes n_f:
   a. Initialize priority queue Q with n_f
   b. While Q is not empty and depth < D:
      i.   Pop highest-score node n from Q
      ii.  For each predecessor n_p of n in G:
           score(n_p) = activation(n_p) * weight(n_p -> n) * score(n)^d
      iii. Add n_p to Q if score exceeds threshold
      iv.  Record path from n_f through visited nodes
   c. Store complete path with all visited nodes

3. Identify bottleneck features B:
   For each feature f appearing in any path:
     convergence(f) = |paths containing f| / |total paths|
   B = {f : convergence(f) >= 0.6}

4. Return paths P and bottleneck features B
```

Geometric decay (score^0.8) is critical: without it, multiplying activation scores across 20+ layers produces exponential overflow. The decay preserves relative path ranking while keeping scores manageable.

Top-K and Bottom-K tracing: We trace paths from both the highest-contributing (correct prediction) and lowest-contributing output nodes. A key finding from our Stage 1 analysis is that top and bottom output nodes converge on the same bottleneck features, indicating shared universal circuits rather than token-specific pathways.

3.3 Cross-Circuit Comparison

To compare circuits across prompts, we compute:

1. **Jaccard similarity**: For each pair of circuits within a category, we compute the Jaccard index over the sets of all features appearing in their traceback paths:

$$J(A, B) = |\text{features}(A) \cap \text{features}(B)| / |\text{features}(A) \cup \text{features}(B)|$$

2. **Shared bottleneck analysis**: For each category, we identify features that appear as bottlenecks (convergence >= 60%) in multiple circuits. A feature appearing in 9/10 chemistry circuits indicates strong within-domain convergence.

3. **Layer distribution**: We compute the distribution of bottleneck features across layers for each category, enabling comparison of where different knowledge types concentrate their critical features.

4. **Layer activation profile similarity**: To compare circuits by where they concentrate activation across depth, we compute cosine similarity between their layer activation profiles. For two profile vectors u and v (each a vector of per-layer activation shares, as defined in §3.1), cosine similarity is

$$\cos(u, v) = (u \cdot v) / (|u| \, |v|)$$

This measures the shape of the profile (the relative distribution of activation across layers) and is invariant to the overall magnitude of either vector, so a circuit with large total activation magnitude and a circuit with small total activation magnitude can still score 1.0 if their activation is distributed across depth in the same proportions.

Within-model comparison is direct: two circuits from the same model have profiles of the same length (26 entries for Gemma, 36 for Qwen), and cosine is computed over the raw share vectors.

Between-model comparison requires depth normalization, because Gemma has 26 layers and Qwen has 36 and the two profile vectors are not the same length. Before computing cosine across models, we resample each circuit's layer activation profile onto a common normalized-depth grid: each profile is treated as a function of normalized depth (0 = first layer, 1 = last layer) and linearly interpolated onto a shared grid of 20 evenly spaced normalized-depth points. Cosine similarity is then computed over the two resampled 20-point vectors. This is the metric behind the architecture-dominance result reported in §5.5.1 (within-model mean cosine 0.978, between-model mean cosine 0.696).

The "approximately 14x" figure is a ratio of two mean cosine gaps. The architecture gap is the difference between the within-model mean (0.978) and the between-model mean (0.696), about 0.282. The domain gap is the difference between within-category and cross-category mean cosine within a model, about 0.02 (§5.5.1). The reported ratio is 0.282 / 0.02 ≈ 14. This is a ratio of mean cosine gaps used as a summary comparison of architecture versus domain effects on the layer activation profile; it is not a formal variance decomposition, and the two gaps are not components of a partitioned variance.

3.4 Statistical Framework

All p-values from multiple comparisons use Bonferroni correction. We apply:

- **Permutation tests** (N=10,000) for within-vs-cross-category Jaccard significance
- **Bootstrap 95% CIs** (N=10,000) for within-category Jaccard means
- **Chi-square tests** for layer distribution independence from category
- **ANOVA** (Shapiro-Wilk normality pre-test) or **Kruskal-Wallis** for category effects on circuit metrics (18 tests, α = 0.003)
- **Mann-Whitney U** with rank-biserial correlation for cross-model comparison (16 tests, α = 0.003)
- **OLS regression** (numpy-based) for multivariate prediction of output probability
- **Spearman rank correlation** for the per-layer activation-confidence analyses (per-layer activation share vs output confidence; §5.5.3, the bottleneck penalty, r = -0.684 at the L6 bottleneck), chosen as a rank correlation that is robust to non-linearity and outliers
- **Cohen's d** for pairwise category effect sizes (thresholds: negligible <0.2, small 0.2-0.5, medium 0.5-0.8, large >0.8)

3.5 Pipeline Architecture

Our end-to-end pipeline consists of:

1. **Graph generation** (Neuronpedia API) → raw attribution graph
2. **Graph conversion** → standardized pipeline format with node/edge attributes
3. **Circuit analysis** → Louvain community detection (a modularity-maximizing partition of the graph into densely-connected sub-communities; Blondel et al., 2008) and betweenness centrality (the fraction of shortest paths between all node pairs that pass through a given node; Freeman, 1977)
4. **Traceback analysis** → priority-guided backward (best-first) search paths, bottleneck identification
5. **Cross-circuit comparison** → within-category and cross-category metrics

6. **Statistical analysis** → 60-row data table with 25 metrics, hypothesis testing across all metrics
7. **Visualization** → circuit diagrams, heatmaps, comparison charts

3.6 Minimal-Pathway Extraction and Redundancy

The traceback procedure (§3.2) ranks paths from the output back through the graph but keeps every feature any high-scoring path touches. To isolate the structurally essential core of a circuit we separately extract its *minimal pathway*: the minimum viable subgraph connecting input features to output features. We compute this for all 60 circuits and report the metrics below in §5.7.

Input and output features. Within the converted attribution graph G = (V, E), *input features* are early-layer features with high fan-out (many outgoing attribution edges), and *output features* are late-layer features with high fan-in (many incoming attribution edges). These two sets anchor the two ends of the pathway: input features are where information first enters the feature network, output features are where it converges before the unembedding.

Essential pathway. The *essential pathway* of a circuit is the minimum viable set of features and edges that connects its input features to its output features, retaining the connections required to route influence from one end to the other and discarding the rest. A feature is *on the essential pathway* of a given circuit if it belongs to that circuit's minimum-viable input-to-output pathway; in the steering experiments (§5.8) we count *essential-pathway membership* as the number of distinct circuits whose essential pathway includes a feature. Across the 30 Gemma circuits this yields roughly 1,000 unique essential-pathway features. (Our extraction prunes the attribution graph to its essential input-to-output routes in the same spirit as information-flow-route extraction (Ferrando & Voita, 2024) and intervention-based subgraph discovery (Conmy et al., 2023); §5.7 reports the resulting reduction qualitatively rather than via a closed-form objective, and we describe it as such.)

Reduction metric. For each circuit, the *reduction* is the percentage of nodes removed when the full circuit is reduced to its essential pathway, that is, the fraction of |V| not retained on the essential pathway. A reduction of 94% means the essential pathway uses 6% of the circuit's nodes.

Circuit redundancy. We define *circuit redundancy* as the percentage of nodes that are NOT on the essential pathway. Operationally this equals the reduction: redundancy and reduction are the same quantity viewed from the two complementary node sets (nodes discarded versus nodes retained), and the paper reports a mean of 94.1% across the 60 circuits. Redundant nodes are active features that lie off the minimal input-to-output route.

Essential-pathway bottleneck width. Within a circuit's essential pathway we measure the *bottleneck width*: the number of features the pathway funnels through at its narrowest point. A width of 1 means the essential pathway passes through a single critical feature. We report this per circuit and summarize it by its mean across the 60 circuits.

Edge-weight retention. The *edge-weight retention* is the fraction of total edge weight (summed over all edges in E) that lies on the essential pathway. Because most attribution weight flows through redundant connections rather than the minimal route, this fraction is small; the paper reports a mean of 0.3%.

The essential pathway, the reduction (and its complement, redundancy), the bottleneck width, and the edge-weight retention together describe how much of a circuit is load-bearing for input-to-output routing as opposed to redundant scaffolding. We relate these quantities to output confidence and to steering effect size in §5.7 and §5.8.

3.7 Steering Perturbation Metric

To quantify how strongly a single-feature intervention perturbs the model's prediction, we use the Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) between the steered and unsteered next-token distributions on the same prompt:

KL(P || Q) = sum_x P(x) log(P(x) / Q(x))

Here P is the steered next-token distribution (with the target feature clamped to a fixed activation strength) and Q is the baseline, unsteered next-token distribution for the same prompt; the sum runs over the vocabulary x. The KL

divergence is asymmetric (KL(P || Q) is not in general equal to KL(Q || P)) and is zero only when the two distributions match, with larger values indicating a stronger distributional shift. Because it compares full distributions rather than only the argmax token, KL registers sub-threshold perturbations that leave the generated text unchanged (see §5.8), and it is the perturbation measure reported throughout the steering experiments (for example, mean KL = 1.448 for essential-pathway features in §5.8.3).

Measurement caveat: the steering KL values reported in the paper are the per-experiment `kl_divergence` figures returned by the Neuronpedia `/api/steer` endpoint. The in-repo figure code re-derives a KL value independently, but only from the top-5 returned log-probabilities of each distribution (`scripts/stage_2_d5d7_figures.py`, the `slps[:5]`/`dlps[:5]` truncation), so that re-derivation approximates the divergence over a truncated head of the vocabulary rather than the full distribution. The two are consistent in sign and ordering but not identical in magnitude; the published numbers are the API values.

3.8 Causal Steering Protocol

To test whether structurally identified features exert causal influence on model behavior, we ran feature-clamping steering experiments. The clamping technique follows prior work showing that fixing a sparse-dictionary feature to a chosen activation value steers model outputs (Turner et al., 2023; Templeton et al., 2024).

Intervention. Each experiment clamps a single transcoder feature to a fixed activation strength of +20 or -20 on a single prompt. We perform no compound or multi-feature interventions: every experiment isolates one feature on one prompt. Generation is deterministic and fixed across experiments (`temperature=0`, `seed=42`, `n_tokens=10`), and interventions are applied through Neuronpedia's `/api/steer` endpoint.

Measured quantities. For each experiment we record three quantities, comparing the steered run against an unsteered baseline on the same prompt:

1. **Text change (binary):** whether the argmax (greedy) output token changed under steering relative to baseline.
2. **KL divergence:** the Kullback-Leibler divergence between the steered and baseline next-token probability distributions (§3.7), which captures distributional perturbation even when the argmax token is unchanged.
3. **Logprob shift:** the change in the log-probability of the baseline top-1 token under steering.

Feature-selection batches. We ran 80 experiments organized into three batches, each using a distinct criterion to select which features to clamp:

- **D4 (cross-circuit frequency, ranks 1-5):** the five features appearing as bottlenecks in the most distinct circuits, drawn from the cross-circuit bottleneck library. 20 experiments.
- **D5 (cross-circuit frequency, ranks 6-10):** the next five most frequent bottleneck features from the same library. 30 experiments.
- **D6 (essential-pathway membership):** five high-frequency features drawn from the minimal viable input-to-output pathways extracted in §5.7, selected by topological membership rather than frequency. 30 experiments.

Cross-circuit frequency counts the number of distinct circuits in which a feature appears as a bottleneck (convergence >= 0.6 in traceback; §3.2); essential-pathway membership counts the number of distinct circuits whose minimal viable pathway includes the feature (§3.6, §5.7).

Scope. Steering is restricted to Gemma. Neuronpedia does not currently support feature-level steering for the Qwen transcoder, so no Qwen steering experiments were possible, and the causal results in §5.8 apply to Gemma only.

3.9 Output Determinism

Several results in this paper (sections 5.8.2, 5.8.3, 6.5) attribute the domain asymmetry in text-level steering susceptibility (chemistry 0%, geography 20%, history 60% text-change rates) to a property we call **output determinism**: the degree to which a prompt's next-token distribution is concentrated on a single token. We define output determinism here so that later sections can refer to it precisely.

For a given prompt, let the model's next-token distribution at the final position be p, with token probabilities sorted in decreasing order $p_{(1)} \geq p_{(2)} \geq \ldots$. A near-deterministic output places almost all of the probability mass on one token ($p_{(1)}$ close to 1, with no close competitor); a higher-entropy output spreads mass across several plausible tokens. The intuition the paper relies on is that a near-deterministic output resists steering, because a perturbation must overcome a large probability margin to change the argmax token, whereas a higher-entropy output is more easily shifted, because a small perturbation can move a close competitor past the current top token. This is the mechanism invoked in section 6.5: chemistry completions (Na, Fe, Pb) are near-certain and resist perturbation, while history and geography outputs carry more distributional uncertainty and so leave more room for steering to change the argmax.

Operationalization. We propose that output determinism be measured per prompt as the gap between the top-1 and top-2 next-token probabilities,

$$\text{margin} = p_{(1)} - p_{(2)},$$

with a larger margin indicating a more deterministic (steering-resistant) output. An equivalent summary is the entropy of the next-token distribution, $H(p) = -\Sigma_i \, p_i \log p_i$, where lower entropy indicates higher determinism; the top1-minus-top2 margin is the simpler quantity to compute and reason about, because it directly measures how far a competitor token must be pushed to flip the argmax.

We use the margin (or, equivalently, the entropy) rather than the top-1 probability $p_{(1)}$ alone, because top-1 probability does not separate the domains. Chemistry and history both tend to produce high top-1 confidence (the cross-model output-probability means in section 5.6.2 are high overall), yet they sit at opposite ends of the steering-susceptibility range (chemistry 0%, history 60%). A high top-1 probability with a non-trivial second-place competitor is more easily flipped than the same top-1 probability with negligible mass elsewhere, and only a margin- or entropy-based measure distinguishes these two cases. The top1-minus-top2 margin therefore captures the "no close competitor" condition that the qualitative argument depends on, which $p_{(1)}$ alone does not.

4. Experimental Setup

4.1 Prompt Design

We designed 30 prompts across three knowledge domains, with 10 prompts per domain. Within each domain, prompts follow a consistent format to control for prompt structure effects:

Chemistry (element → symbol): "The chemical symbol for [element] is"

- Elements: gold (Au), iron (Fe), lead (Pb), mercury (Hg), argon (Ar), sodium (Na), silver (Ag), potassium (K), copper (Cu), tungsten (W)
- Rationale: Most elements have non-obvious symbols derived from Latin/German names, requiring genuine knowledge retrieval rather than pattern matching.

Geography (country → capital): "The capital of [country] is"

- Countries: France (Paris), Japan (Tokyo), Germany (Berlin), Egypt (Cairo), Italy (Rome), Spain (Madrid), Canada (Ottawa), Thailand (Bangkok), Turkey (Ankara), South Korea (Seoul)
- Rationale: Well-known capitals with single-token answers.

History (event → year): Various formats ending with date completion

• Events: WWII ended (1945), moon landing (1969), Berlin Wall fell (1989), Titanic sank (1912), American independence (1776), French Revolution (1789), WWI started (1914), Columbus (1492), Great Fire of London (1666), Mandela released (1990)

• Rationale: Major historical events with widely known dates.

4.2 Models

• **Gemma-2-2B** (Google): 26 layers, ~2B parameters

• **Qwen3-4B** (Alibaba): 36 layers, ~4B parameters

Both models were analyzed using their respective transcoder feature sets available on Neuronpedia (Gemma Scope transcoders for Gemma; the Neuronpedia-hosted `transcoder-hp` for Qwen).

4.3 Data Collection

For each of the 30 prompts × 2 models = 60 combinations:

1. Generated attribution graph via Neuronpedia API
2. Converted to pipeline format (1,000-1,400 nodes, 40,000-90,000 edges per graph)
3. Ran traceback analysis (5 critical paths per circuit)
4. Extracted bottleneck features (convergence >= 60%)
5. Assembled a flat 60-row data table with ~25 per-circuit metrics (topology, activation magnitudes, bottleneck position, community structure, edge weight statistics) for statistical analysis

5. Results

5.1 Within-Category Circuit Convergence

We measured circuit similarity within each knowledge domain using average pairwise Jaccard similarity of traceback path features.

Category	Gemma Jaccard	Qwen Jaccard
Chemistry	0.289	0.353
Geography	0.395	0.334
History	0.108	0.153

Geography circuits are most similar: In Gemma, "capital of X" prompts share 39.5% of their traceback features on average, the highest of any category. This suggests geographic knowledge retrieval uses circuits that share most features across prompts.

History circuits are most diverse: Historical event prompts share only 10.8% (Gemma) and 15.3% (Qwen) of features. "World War II ended in" and "Columbus reached the Americas in" involve different subject matter, while "capital of Japan" and "capital of France" are structurally identical queries.

Top bottleneck consistency:

• Gemma Geography: L0_F1813559 appears in **10/10** circuits (100%)

• Gemma Chemistry: L0_F124985954 appears in **9/10** circuits (90%)

• Gemma History: L6_F2586668 appears in only **6/10** circuits (60%)

- Qwen Chemistry: L27_F7867534052 appears in **10/10** circuits (100%)
- Qwen Geography: L34_F415944868 appears in **10/10** circuits (100%)
- Qwen History: L34_F415944868 appears in **10/10** circuits (100%)

Qwen's L34_F415944868 appears as a bottleneck in 100% of geography and history circuits and 90% of chemistry circuits.

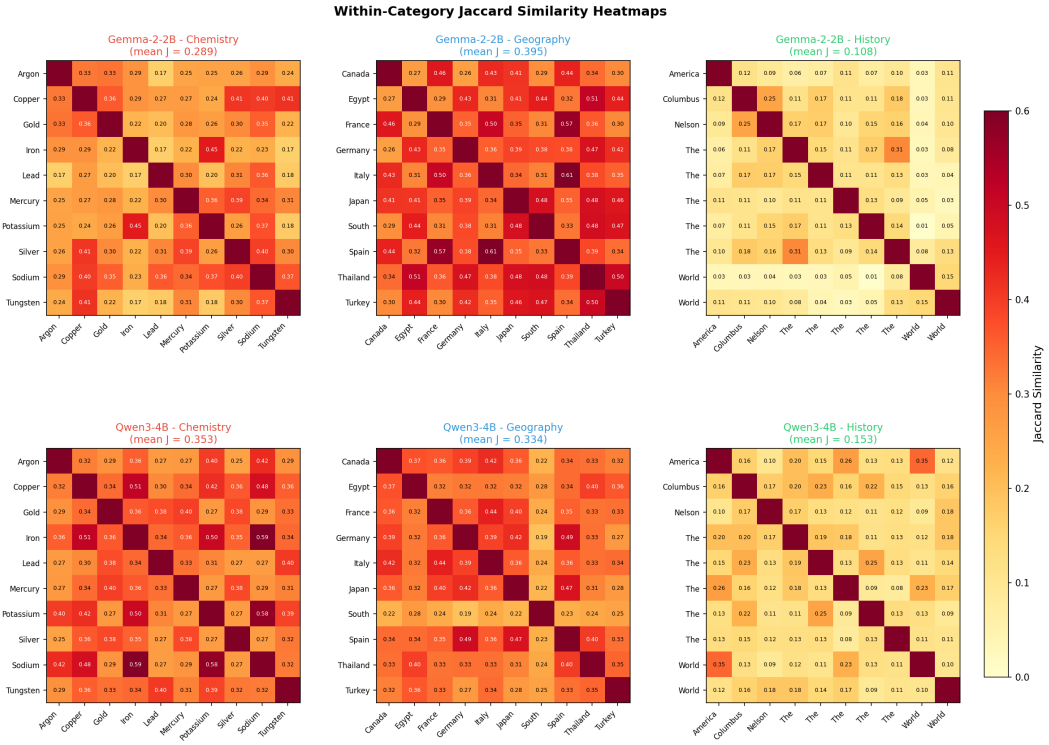

Figure 1: Within-category Jaccard similarity heatmaps for Gemma (top) and Qwen (bottom) across chemistry, geography, and history.

5.2 Bottleneck Depth is Architecture-Dependent

Bottleneck layer position is strongly predicted by model architecture, not knowledge domain:

Category	Gemma Avg Layer	Gemma % Depth	Qwen Avg Layer	Qwen % Depth
Chemistry	L5.5	21%	L24.1	67%
Geography	L5.8	22%	L24.6	68%
History	L6.6	25%	L22.0	61%

Gemma bottlenecks consistently cluster in layers 5–7 (~22% depth) across all three knowledge domains. Qwen bottlenecks cluster in layers 22–25 (~65% depth). The within-model range (L5.5–6.6 for Gemma, L22.0–24.6 for Qwen) is much smaller than the between-model gap.

Bottleneck position is an architectural property of the model in this dataset, not a function of what knowledge is being retrieved. Gemma's bottleneck features sit in early layers regardless of domain; Qwen's sit in late layers. (Fig. 2)

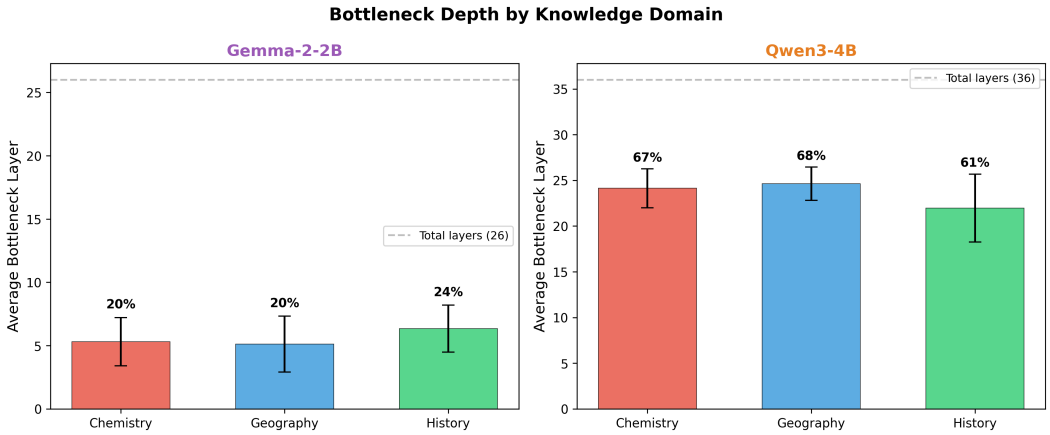

Figure 2: Bottleneck depth distribution by model and domain. Gemma clusters at L5-7, Qwen at L22-25.

5.3 Universal Bottleneck Features

We identified features that serve as bottlenecks across ALL three knowledge domains:

Gemma universal bottlenecks (6):

Feature	Layer	Known Description
L0_F1813559	L0	Code/file keywords
L0_F74438300	L0	—
L3_F5150441	L3	HTML formatting tags
L4_F110446948	L4	Place names
L6_F2586668	L6	Code snippets
L24_F88478228	L24	Lithuanian place names

Qwen universal bottlenecks (15):

Feature	Layer
L15_F8173514424	L15
L16_F1043399704	L16
L17_F6197796762	L17
L18_F11703352509	L18
L19_F6611637508	L19
L24_F457788386	L24
L25_F942235729	L25
L26_F10368215974	L26
L26_F10912888953	L26
L27_F12610705050	L27
L28_F393078712	L28

Feature	Layer
L29_F89625936	L29
L32_F3158177517	L32
L33_F10240593794	L33
L34_F415944868	L34

Zero features overlap between the Gemma and Qwen universal bottleneck sets. The two models use non-overlapping sets of universal bottleneck features, consistent with the architecture-dominance pattern.

Qwen has 2.5× more universal bottlenecks than Gemma (15 vs 6), and they span a wider layer range (L15-L34 vs L0-L24). This suggests Qwen's later, more distributed bottleneck architecture may provide more opportunities for cross-domain feature sharing.

5.4 Output Node Convergence

We separately analyzed whether same-domain circuits converge at the output layer (final-layer features) versus along full traceback paths.

Model	Category	Output Jaccard	Path Jaccard	Ratio
Gemma	Chemistry	0.293	0.289	1.02
Gemma	Geography	0.436	0.395	1.10
Gemma	History	0.283	0.108	2.63
Qwen	Chemistry	0.559	0.353	1.58
Qwen	Geography	0.475	0.335	1.42
Qwen	History	0.673	0.153	4.41

Output node convergence is consistently higher than path convergence, and the gap is largest for history (4.41× in Qwen). Same-domain history circuits share the same output features even though their intermediate paths diverge: the model arrives at similar output vocabulary through different internal routes.

The mechanism is direct logit attribution (DLA): each output-layer transcoder feature has a decoder direction that, projected through the unembedding matrix, contributes a fixed pattern of token-logit pushes. Features whose decoders align with a shared vocabulary region (e.g., 4-digit-year tokens in history prompts) end up acting as common terminal nodes for circuits with otherwise divergent earlier processing.

Qwen shows stronger output convergence overall: history circuits share 67.3% of output features despite only 15.3% path overlap. (Fig. 3)

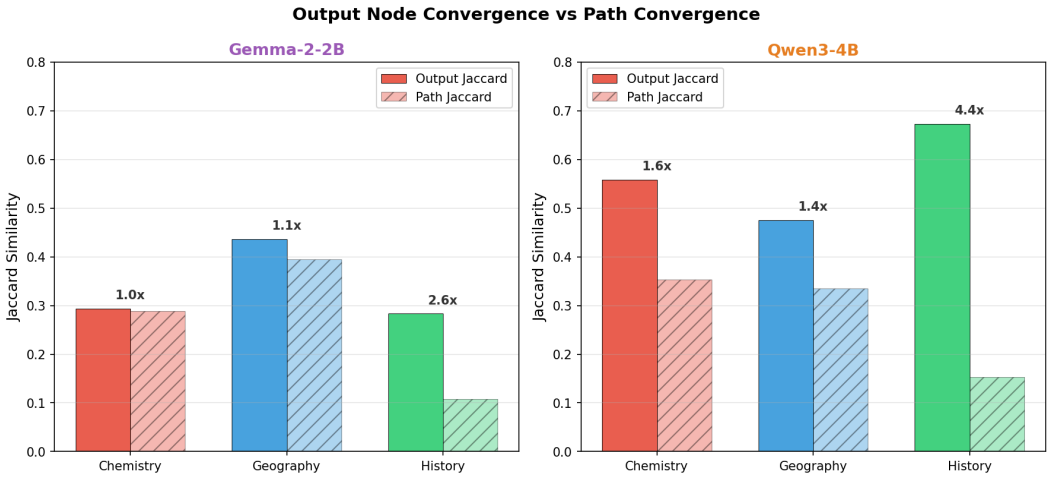

Figure 3: Output node convergence vs path convergence by domain. History shows the largest output/path ratio (4.4× in Qwen).

5.5 Architecture Dominates Layer Activation Profiles

5.5.1 Profile Similarity

We computed cosine similarity between per-layer activation share profiles (summed activation magnitude at each layer as a proportion of the circuit total) for all pairs of circuits.

Within-model similarity is tight. Mean cosine similarity across all 30 circuits within each model is 0.978 (std = 0.013), independent of knowledge domain. Profiles cluster within a narrow band of the cosine space.

Within-category similarity is slightly higher than cross-category (Gemma: 0.984 vs 0.966; Qwen: 0.975 vs 0.956; KS test $p < 0.001$ for both models). Knowledge domains do produce a small but statistically significant modulation of the layer activation profile. The domain effect (cosine difference ~0.02) is dwarfed by the architecture effect.

Between-model similarity is substantially lower. After normalizing both models to a common [0,1] depth scale (Gemma 26 layers, Qwen 36 layers; the depth-normalization and cosine procedure is defined in §3.3), mean cosine similarity drops to 0.696 (std = 0.045). Mann-Whitney U confirms this difference is significant ($p < 0.000001$).

The gap between within-model (0.978) and between-model (0.696) similarity is 0.282, approximately 14× the within-model domain effect (0.02); this ratio of mean cosine gaps is defined in §3.3.

Architecture produces approximately **14× the mean cosine gap that knowledge domain does**. We refer to this ratio of mean cosine gaps as architecture dominance, and use it throughout the paper as a summary of the architecture-versus-domain comparison.

5.5.2 Front-Loading vs Back-Loading

The two models show fundamentally different activation accumulation strategies:

- **Gemma is front-loaded**: 54% of total activation magnitude is in the first half of layers (L0-L12). The cumulative share reaches 50% by L11 and 90% by L24.
- **Qwen is back-loaded**: only 31% of total activation magnitude is in the first half (L0-L17). The cumulative share reaches 50% by L28 and 90% by L34.

These accumulation profiles are consistent across all three knowledge domains (chemistry/geography/history), with category differences in the 50% threshold layer of only 1-2 layers within each model.

5.5.3 The Bottleneck Penalty

We correlated each layer's activation share with output probability across all 30 circuits within each model, applying Bonferroni correction (Gemma: alpha = 0.0019 for 26 tests; Qwen: alpha = 0.0014 for 36 tests).

Gemma reveals 7 Bonferroni-significant layers, with an alternating pattern:

Layer	Spearman r	p-value	Direction
L6	-0.684	<0.0001	Negative (bottleneck penalty)
L10	-0.601	0.0004	Negative
L1	+0.634	0.0002	Positive
L13	+0.601	0.0004	Positive
L16	+0.598	0.0005	Positive

L6, the primary bottleneck layer identified in our traceback analysis (L5-7), shows the strongest negative correlation (r = -0.684): **circuits with a larger fraction of activation magnitude at the bottleneck layer exhibit lower output confidence**. We term this the "bottleneck penalty."

Theoretical framing. We interpret this pattern through information bottleneck theory (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017): deep networks compress information through intermediate layers, and this compression affects what downstream layers can recover.

The theory predicts a tradeoff: heavier compression at a bottleneck limits the mutual information I(X; T) between input X and the bottleneck representation T, which in turn limits the information available for downstream prediction. Our correlation is consistent with this prediction: circuits that concentrate more activation magnitude at L6 may be compressing more aggressively, leaving less information for evidence accumulation at L13–L16.

Activation at post-bottleneck layers positively predicts confidence, consistent with those layers aggregating information into the final decision.

Important caveat. Activation magnitudes at different layers are computed independently; no conserved quantity is being allocated across layers. The bottleneck penalty is a **correlation, not a mechanistic causal claim**.

Alternative explanations include: (a) harder prompts require more compression at L6 *and* are harder to answer confidently (prompt-difficulty confound); (b) specific knowledge domains may both concentrate work at L6 and have lower baseline accuracy; (c) feature-selection effects in the traceback algorithm. Direct causal validation would require interventional experiments such as clamping L6 activation and measuring confidence change.

Qwen shows 0 Bonferroni-significant layers (best: L29, r = 0.552, p = 0.0016), consistent with its late-layer architecture decoupling layer-level activation from confidence. (Fig. 4, 5, 6)

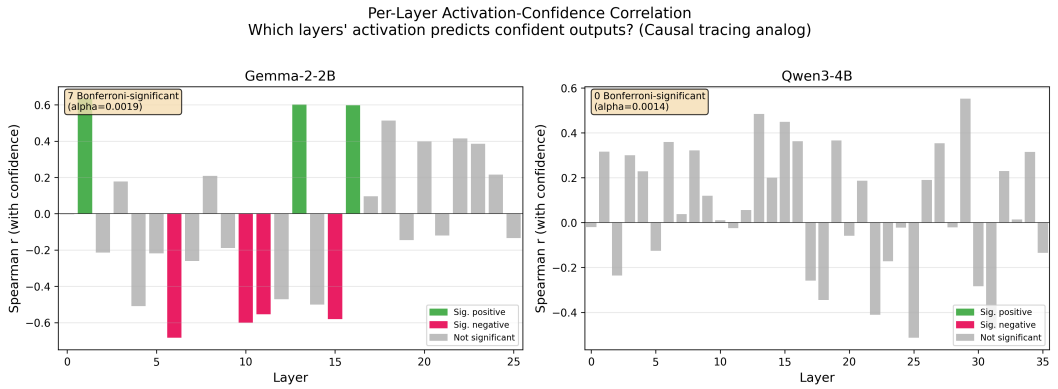

Figure 4: Per-layer activation-confidence correlation in Gemma. L6 (bottleneck) shows strong negative correlation (r=-0.684); post-bottleneck layers show positive correlations.

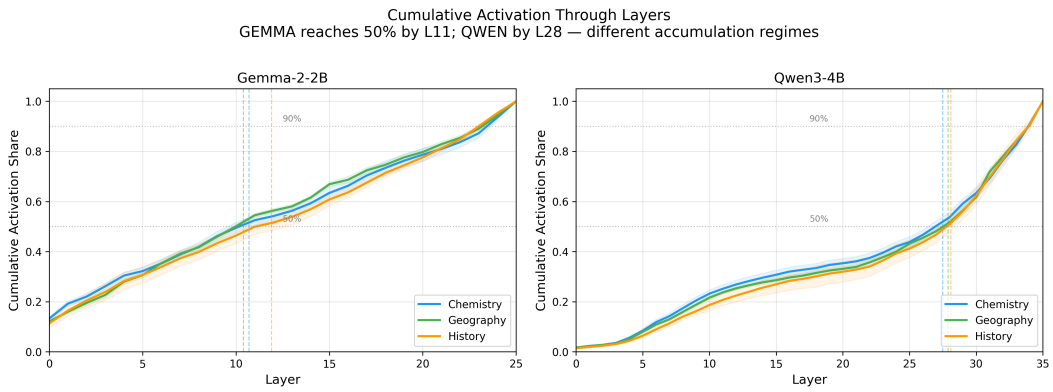

Figure 5: Cumulative layer activation profiles. Gemma front-loads 54% into the first half; Qwen back-loads with only 31%.

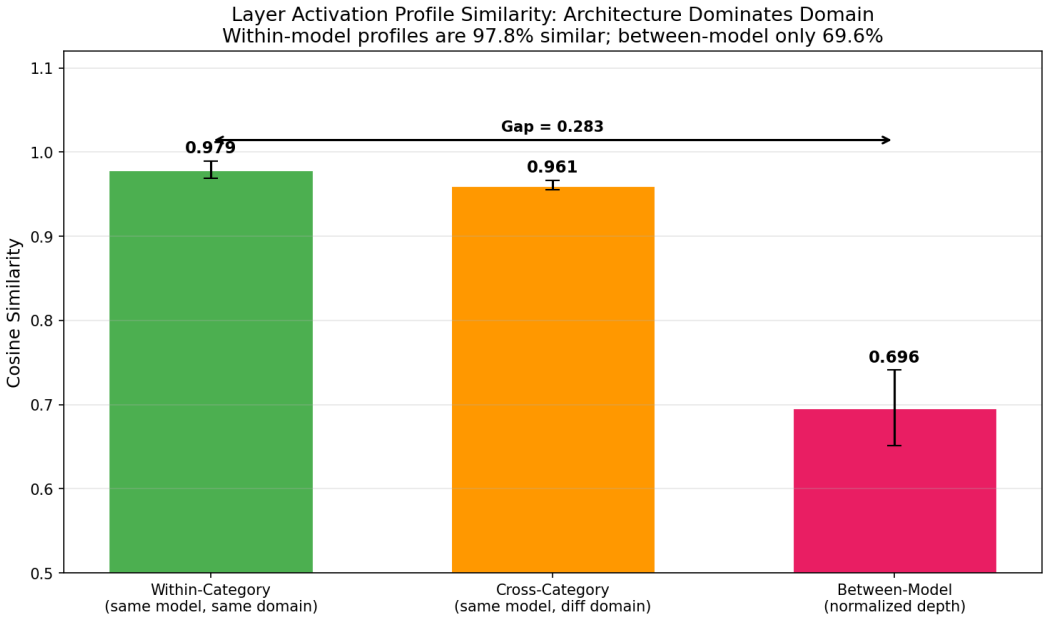

Figure 6: Layer activation profile similarity. Within-model cosine = 0.978; between-model = 0.696.

5.6 Confidence Prediction Model

5.6.1 Multiple Regression

We fitted OLS regression models predicting output probability from 9 circuit features:

Gemma full model (9 predictors, n=30): $R^2 = 0.558$, adjusted $R^2 = 0.359$, $F(9,20) = 2.80$, $p = 0.026$. However, no individual predictor reaches significance due to multicollinearity (21 residual df with 9 predictors).

Gemma reduced model (3 predictors: total activation magnitude, edge weight kurtosis, and node count; internal keys `total_energy`, `weight_kurtosis`, `total_nodes`): $R^2 = 0.489$, adjusted $R^2 = 0.430$, $F(3,26) = 8.30$, $p = 0.0005$. This model explains 49% of confidence variance with only 6% adjusted R^2 loss from the full model, indicating the remaining 6 predictors contribute minimal unique variance.

Qwen full model: $R^2 = 0.337$, adjusted $R^2 = 0.038$, $F(9,20) = 1.13$, $p = 0.389$; not significant. Circuit structure does not linearly predict Qwen confidence.

Qwen reduced model: $R^2 = 0.124$, $p = 0.319$; also not significant.

Power note: With n=30 and 9 predictors, only 21 residual degrees of freedom remain. The reduced model (3 predictors, 26 df) provides more reliable estimates.

5.6.2 Cross-Model Comparison

Mann-Whitney U tests (Bonferroni α = 0.003) reveal **13/16 metrics differ significantly** between Gemma and Qwen:

Metric	Gemma Mean	Qwen Mean	r_rb	Effect
Peak activation layer	14.3	**35.0**	1.000	large
Avg bottleneck layer	5.6	**23.6**	1.000	large
Total activation magnitude	**10819**	5394	-1.000	large
Weight kurtosis	115	**534**	0.982	large
Top-10% weight share	0.46	**0.52**	0.922	large
Output probability	0.49	**0.84**	0.809	large
Avg layers visited	15.4	**18.7**	0.741	large
Total nodes	**1131**	921	-0.741	large

Three metrics do NOT differ: branching factor (p = 0.24), total edges (p = 0.46), and community entropy (p = 0.09). These shared properties may represent architecture-invariant computational constants.

Peak activation layer shows **perfect rank separation** (r = 1.000): every Gemma circuit has its peak activation at an earlier layer than every Qwen circuit, indicating architecture-determined activation profiles for the two models in this dataset.

5.6.3 Null Results

Two targeted analyses yielded null results that constrain our interpretation:

Bottleneck convergence vs confidence: Neither model shows a significant correlation between mean bottleneck convergence score and output probability (Gemma: r = 0.270, p = 0.149; Qwen: r = -0.176, p = 0.351). This means the "quality" of bottleneck convergence (how strongly paths funnel through bottlenecks) does not predict how confident the model is. Confidence is driven by circuit scale (total activation magnitude, node count), not by information compression.

Concentration index vs confidence: A novel composite metric (top-10% weight share × weight kurtosis / total edges) also shows no significant relationship with confidence in either model. Information concentration per edge does not explain prediction certainty. (Fig. 7, 8)

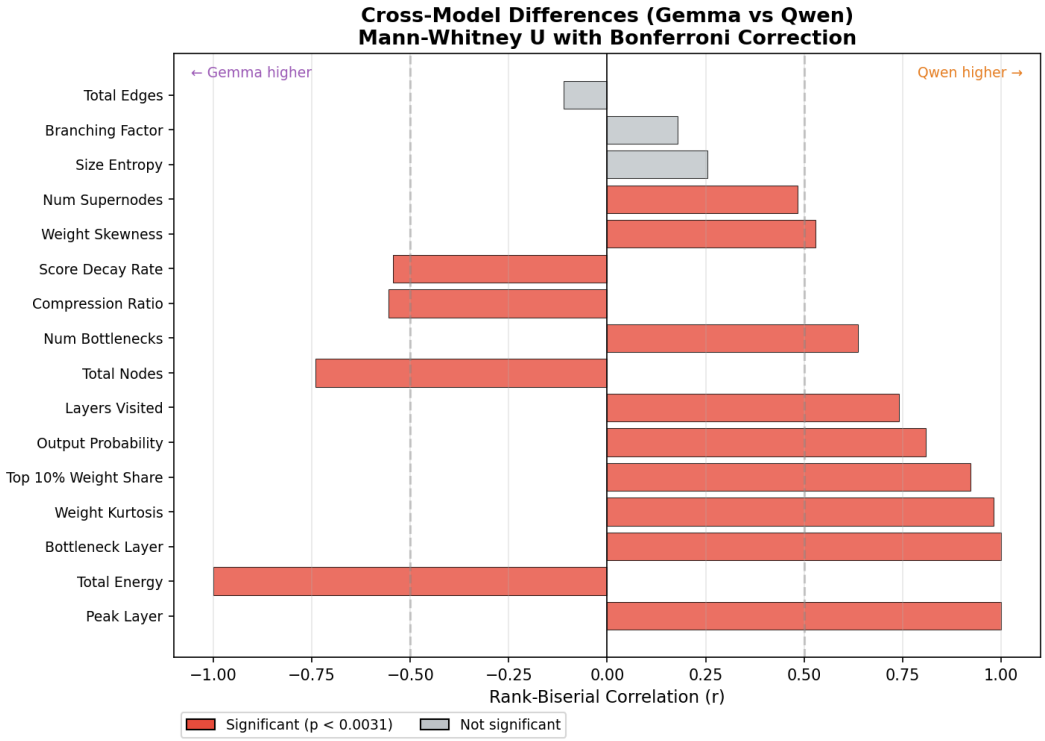

Figure 7: Cross-model differences (Mann-Whitney). 13/16 metrics differ significantly; peak activation layer shows perfect rank separation.

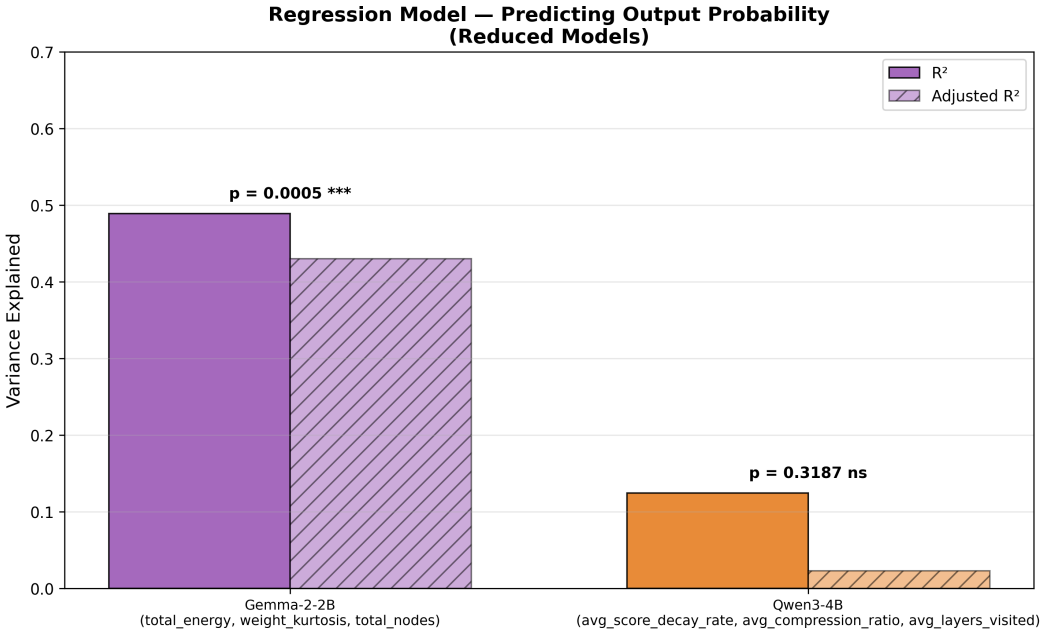

Figure 8: Regression model summary. 3-predictor model explains 49% of Gemma confidence variance.

5.7 Circuit Redundancy

Using the minimal-pathway extraction defined in §3.6, we computed the reduction, redundancy, essential-pathway bottleneck width, and edge-weight retention for all 60 circuits (Fig. 9).

Circuits are 94.1% redundant. The mean reduction across all 60 circuits is 94.1% (±1.2%), meaning only ~6% of nodes participate in the shortest input-to-output pathways. This redundancy is consistent across models (Gemma 94.7%, Qwen 93.5%) and categories (chemistry 94.2%, geography 93.3%, history 94.8%).

Single-node bottlenecks are universal. The mean minimal-pathway bottleneck width is 1.1 features. Nearly every circuit funnels through a single critical node in its essential pathway. Gemma bottlenecks concentrate at L22 (median), while Qwen bottlenecks concentrate at L30 (median), both occurring later than the bottleneck layers identified by the per-layer activation analysis.

Essential edges carry only 0.3% of total weight. The minimal pathway retains 0.3% of total edge weight on average, indicating that most edge weight flows through redundant connections. Attribution graphs over-represent the computational core relative to the essential pathway.

Reduction correlates with confidence in Gemma. Gemma shows a strong positive correlation between circuit reduction and output confidence (r=0.642, p<0.001): more redundant circuits produce more confident outputs. Weight retention shows the inverse pattern (r=-0.589). Qwen shows no such relationship (r=-0.051), consistent with its diffuse architecture lacking localized confidence-predictive structure. (Fig. 9)

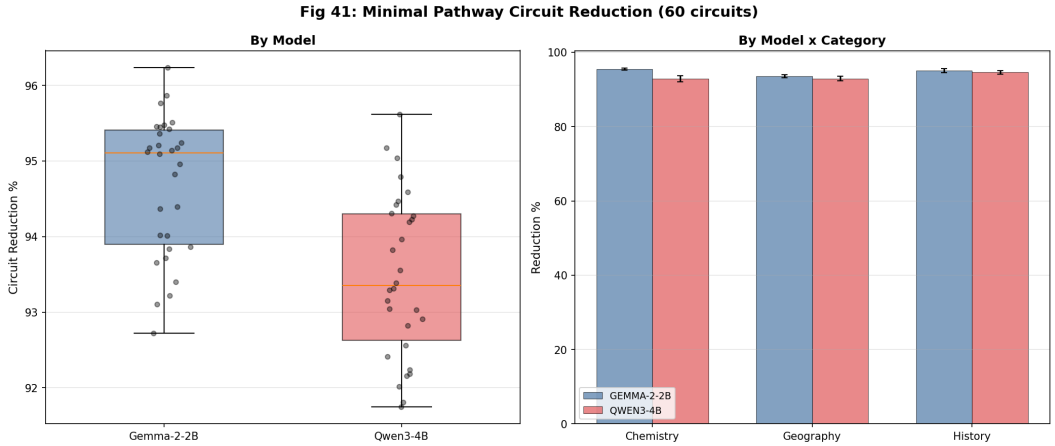

Figure 9: Minimal pathway extraction. Circuits are 94.1% redundant; essential edges carry 0.3% of total weight.

5.8 Causal Steering Validation

We ran 80 total steering experiments across three batches (D4, D5, D6), following the protocol in §3.8 (single transcoder feature clamped to ±20 per prompt, `temperature=0`, `seed=42`, `n_tokens=10`, via Neuronpedia's `/api/steer` endpoint). Each batch uses a different criterion to select which features to intervene on. The three measured quantities (text change, KL divergence, logprob shift) are defined in §3.8.

Feature-selection design:

Batch	Selection criterion	Source pool	Features (5 per batch)	Experiments
D4	Cross-circuit frequency, ranks 1–5	Stage 1.5 bottleneck library (244 features)	L0_F1813559, L3_F5150441, L4_F110446948, L6_F2586668, L24_F88478228	20

Batch	Selection criterion	Source pool	Features (5 per batch)	Experiments
D5	Cross-circuit frequency, ranks 6–10	Same library	L1_F99962728, L2_F25751073, L5_F7993995, L7_F4828270, L9_F125286525	30
D6	Essential-pathway membership (topology)	1,000 essential-pathway features from §5.7	L0_F64712375, L1_F1736314, L21_F5479683, L24_F18002975, L25_F50014975	30

Cross-circuit frequency and essential-pathway membership are defined in §3.8 and §3.6 respectively. Essential pathways comprise typically only ~6% of nodes per circuit.

A pre-experiment cross-check on D4+D5 features revealed that only 5 of 10 frequency-selected features (L0_F1813559, L1_F99962728, L2_F25751073, L3_F5150441, L24_F88478228) were also on essential pathways. The other 5 were frequent but not essential.

This split between "frequent and topologically central" versus "frequent but topologically peripheral" allowed direct comparison of the two selection criteria.

5.8.1 Initial Validation (D4)

We analyzed 20 steering experiments across the 5 D4 Gemma bottleneck features and 6 target circuits, comparing circuit-structural predictors with actual intervention effect sizes (Fig. 10).

Steering effects are small but consistent. The mean logprob shift (the change in log-probability of the baseline top-1 token under steering) is 0.045 across all experiments, with 50% of experiments producing visible text changes. Effects are probability-level perturbations rather than token-flipping interventions.

Graph-structural metrics are unavailable for steered features. None of the 5 steered features appear as nodes in their respective circuits' attribution graphs. The attribution graph's inclusion threshold filters out features that are active but below the display threshold.

This means steered features exert causal influence despite being invisible in the circuit representation, a limitation of threshold-based circuit extraction.

Circuit-level predictors show promise. In the absence of graph metrics, circuit-level features provide the best predictions. Average convergence (from the bottleneck library) correlates positively with KL divergence (§3.7; $r = 0.387$), suggesting that features with high convergence across more paths tend to produce larger steering effects. The number of circuits a feature appears in also trends positive ($r = 0.123$) but does not reach significance at $n = 20$.

Layer depth inversely relates to effect size. The deepest feature tested (L24) shows the smallest mean effect (0.020), while early-to-mid features (L0, L4, L6) show 2-3× larger effects (0.054-0.058). This is consistent with the bottleneck penalty: deeper features may be more embedded in compensatory pathways that absorb perturbations.

5.8.2 Expanded Steering: Frequency vs Causation

We expanded the causal steering analysis from 20 to 50 experiments, testing 5 additional Gemma bottleneck features (L2_F25751073, L5_F7993995, L9_F125286525, L1_F99962728, L7_F4828270) selected by cross-circuit frequency (9-11 circuits each) across 3 target circuits (sodium/chemistry, japan/geography, wwii/history) at strengths ±20 (Fig. 11).

Expanded features show lower steering success. The 5 new features achieved a 23.3% text change rate (7/30), compared to 50.0% (10/20) for the original features. The combined rate across all 50 experiments is 34.0% (17/50). This suggests that cross-circuit frequency alone is insufficient for predicting causal steering effectiveness.

Domain susceptibility is asymmetric. History prompts were most susceptible to steering (5 of 7 D5 changes involved history), while chemistry prompts produced no changes across all 30 D5 experiments. Geography showed intermediate susceptibility (2 changes, both amplification). This domain asymmetry mirrors the lower Jaccard similarity of history circuits (§5.1); less structurally conserved circuits may be more vulnerable to perturbation.

Amplification outperforms suppression in new features. Of the 7 changes in D5 features, 5 occurred with positive strength (+20, amplification) and only 2 with negative (-20, suppression). L7_F4828270 was the most responsive new feature (3/6 changed), while L2_F25751073, despite having the highest cross-circuit frequency (11 circuits), produced zero detectable changes. High frequency does not imply high causal influence.

KL divergence confirms sub-threshold effects. Even experiments with no text change show non-zero KL divergence (e.g., L9_F125286525 geography +20: KL=0.370), confirming that steering perturbs probability distributions even when the argmax token remains unchanged.

5.8.3 Essential-Pathway Steering: The Three-Tier Dissociation

We tested whether features on the minimal viable pathway (§5.7) produce stronger steering effects than features selected by cross-circuit frequency (§5.8.2).

From the 1,000 unique features identified on essential pathways across 30 Gemma circuits, we selected 5 high-frequency essential-pathway features not previously tested (L0_F64712375, L1_F1736314, L21_F5479683, L25_F50014975, L24_F18002975; 10–23 pathway circuits each). We ran 30 steering experiments (5 features × 3 circuits × ±20) on the same target circuits as D5.

Essential-pathway features produce moderate text changes. The 5 new features achieved a 26.7% text change rate (8/30), intermediate between D5 features on essential pathways (18.8%, 3/16) and D5 features off essential pathways (33.3%, 6/18). Neither pathway position nor cross-circuit frequency alone predicts text-level causal influence.

Pathway features produce larger distributional perturbations. Despite comparable text change rates, essential-pathway features exhibit substantially higher mean KL divergence (1.448) than D5 features, driven by 6 of 30 experiments with KL > 2.0.

The strongest single perturbation (L25_F50014975 amplification on geography, KL = 12.72) changed the output from a generic description to the factually correct "Tokyo," suggesting this feature gates factual recall at the output layer.

Layer depth modulates steering direction. Early pathway features (L0, L1) produced changes exclusively via suppression (2/2 changes at strength=-20), while late pathway features (L21, L24, L25) produced changes via both suppression and amplification (6 changes split 3/3). This suggests early features serve as information gates that cause disruption when removed, while late features actively shape output content and respond to both directions of perturbation.

Domain susceptibility is consistent with prior findings. Chemistry produced no changes (0/10), geography showed intermediate susceptibility (2/10), and history was most susceptible (6/10, 60%). This replicates the pattern from §5.8.2 and reinforces the interpretation that output determinism (defined in §3.9), not circuit topology, governs steering susceptibility.

The topology-causal relationship is nuanced. Being on the essential pathway predicts stronger sub-threshold perturbation (higher KL divergence) but not stronger text-level changes. Combined with §5.8.2's finding that frequency does not predict text changes, this suggests a three-tier model of feature importance:

1. Features on essential pathways perturb probability distributions most strongly.

2. Features with high cross-circuit frequency are statistically common but not necessarily causal.

3. Text-level output changes are governed primarily by output entropy and domain determinism rather than by any single feature-selection criterion. (Fig. 10, 11)

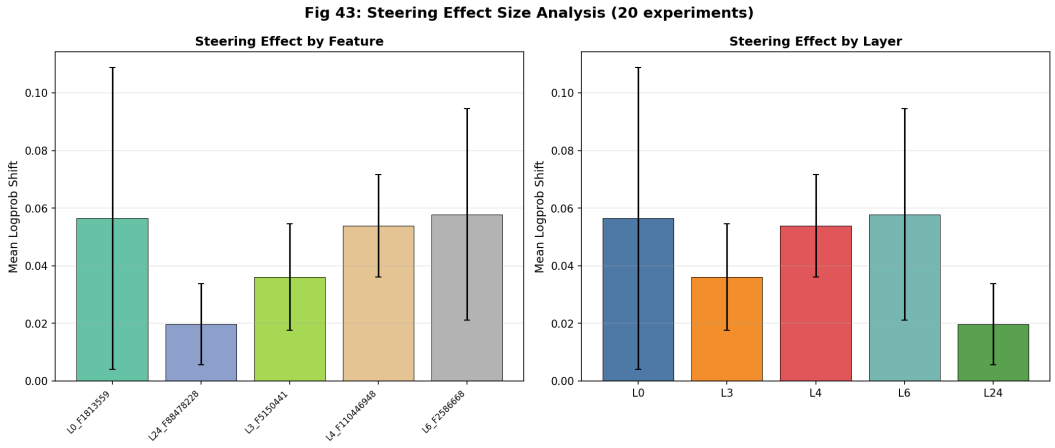

Figure 10: Steering effects across 20 initial experiments. Mean logprob shift = 0.045; 50% produce visible text changes.

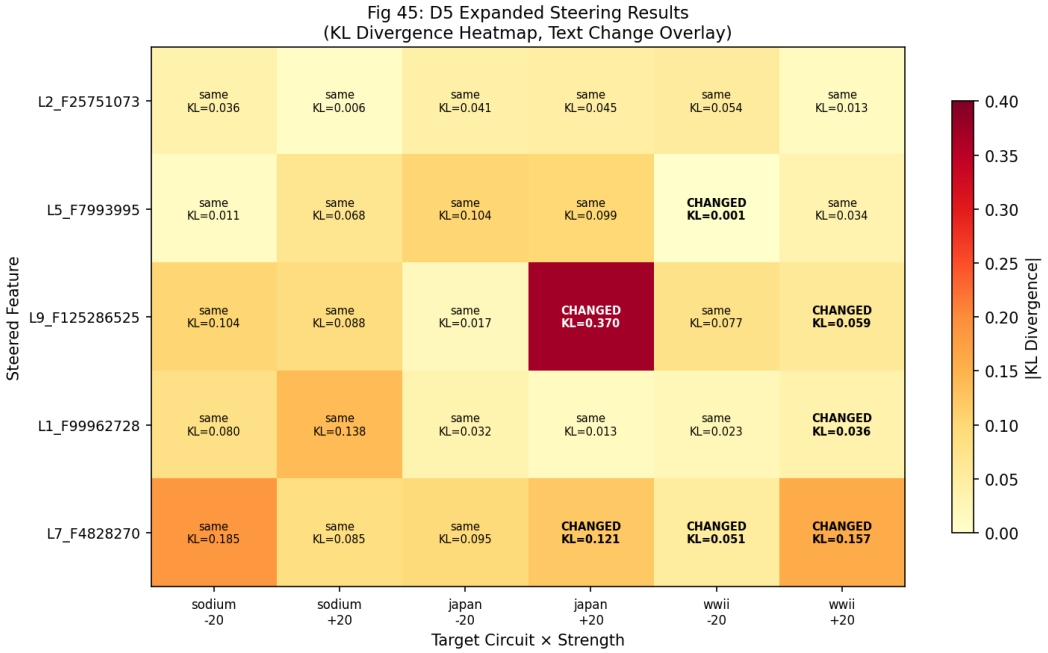

Figure 11: Expanded steering (D5). Frequency does not predict causal influence; domain susceptibility is asymmetric.

Extended results including statistical significance tests, domain effect sizes, path topology, community structure analysis, co-activation patterns, polysemanticity classification, and algorithm validation are provided in Supplementary Materials (Sections S1.1--S1.21).

5.9 Format Variation: Template vs Domain Effects

A critical question is whether the within-domain circuit similarity observed in §5.1 reflects genuine knowledge-domain convergence or merely prompt-template similarity. All chemistry prompts share "The chemical symbol for X is."

To test this, we generated circuits for the same fact using three different prompt formats and compared the resulting Jaccard similarity.

Pilot study: gold circuits across three formats:

- Original: "The chemical symbol for gold is" (1288 nodes, 1110 features)
- Alt 1: "Gold is represented by the chemical symbol" (1300 nodes, 1085 features)
- Alt 2: "In the periodic table, gold has the symbol" (1531 nodes, 1223 features)

Comparison	Mean Jaccard	N pairs
Same fact, different format (gold × 3)	0.306	3
Different fact, same format (chemistry)	0.611	3
Different fact, same format (geography)	0.713	3
Different fact, same format (history)	0.156	3
Different fact, same format (all domains)	0.493	9
Cross-domain baseline	0.112	27

All comparisons use the same raw feature ID space (validated by confirming J=1.000 for the same prompt generated in both sessions).

Prompt template drives more circuit overlap than factual identity, but the effect is domain-dependent. Averaging across all domains, same-template pairs (J = 0.493) exceed same-fact-different-template pairs (J = 0.306) by a factor of 1.6×.

Chemistry (0.611) and geography (0.713) show strong template-driven convergence, but history (0.156) does not. History circuits are so structurally diverse that the template provides little convergence beyond the cross-domain baseline (0.112), mirroring the low within-domain Jaccard for history reported in §5.1 (Gemma: 0.108).

This establishes a nuanced hierarchy: **template > fact > cross-domain** for chemistry and geography, but **fact > template ≈ cross-domain** for history. The within-domain Jaccard values in §5.1 reflect a mixture of template and domain effects for chemistry and geography, while history's low convergence is genuine. The same-fact-different-format Jaccard (0.306) is still 2.7× higher than the cross-domain baseline (0.112), confirming that domain contributes to circuit structure beyond template alone.

This finding does not undermine the architecture-dominance result (§5.5), which is based on per-layer layer activation profiles (cosine 0.978 within-model), not feature-level Jaccard. Layer activation profiles are robust to prompt format because they capture *where* computation happens across layers, not the identity of individual features.

The format effect operates at the feature-selection level (which specific features activate). The architecture effect operates at the computational-structure level (where activation is concentrated).

Note: This pilot is limited to one fact (gold) across three formats in one domain (chemistry), due to Neuronpedia API rate limits on graph generation. Extending to all facts and domains is an important direction for future work.

6. Discussion

6.1 Architecture Determines Circuit Structure

Beyond bottleneck position, the entire layer-by-layer activation distribution is architecture-determined. Gemma routes all factual knowledge through early-layer bottlenecks (L5–7); Qwen routes through late-layer bottlenecks (L22–25).

Within-model cosine similarity is 0.978 (profiles cluster tightly across domains) while between-model similarity drops to 0.696 even after depth normalization. Architecture produces approximately 14× the mean cosine gap that knowledge domain does.

Edge flow analysis extends this invariance from activation magnitude to wiring topology. Cross-category edge pattern similarity is 0.995 for both models, higher than the 0.978 layer activation profile similarity. Domains share nearly identical wiring diagrams within each architecture.

Skip connections dominate: 80–85% of edges bypass adjacent layers. Gemma's strongest connections radiate from L0 outward (L0→L1, L0→L2, L0→L4; early fan-out consistent with front-loading); Qwen's concentrate at L30–L35 (L34→L35, L30→L31; late sequential processing consistent with back-loading). The architectural pattern extends from how much activation flows to where it flows. (See Supplementary Materials S1.14)

The Mann-Whitney analysis shows cross-model differences across 13/16 circuit metrics, with peak activation layer and bottleneck layer showing perfect rank separation ($r = 1.0$). Only branching factor, total edges, and community entropy are architecture-invariant.

These three shared properties may represent computational constants of factual retrieval: regardless of where bottlenecks are placed or how much activation is recruited, the branching structure of information flow and the total volume of edges appear fixed by the task rather than the architecture. (See Supplementary Materials S1.5)

The complementary accumulation strategies (Gemma places 54% of total activation magnitude in the first half of layers while Qwen places only 31%) reflect different depth strategies. In Gemma, activation magnitude concentrates early and falls off toward the output; in Qwen it accumulates broadly before concentrating in late layers.

Both strategies produce comparable factual recall accuracy across all three domains, demonstrating that early-bottleneck and late-bottleneck architectures represent distinct but equally viable solutions to the same knowledge retrieval problem.

The varying degree of within-domain circuit similarity (geography > chemistry > history) further reflects the structural similarity of prompts within each domain. Geography prompts are nearly identical in structure ("The capital of X is"), producing highly conserved circuits; history prompts vary more in structure and subject matter, producing the most diverse circuits. (See Supplementary Materials S1.6)

6.2 Universal Features as Routing Infrastructure

The existence of universal bottleneck features that participate in all three knowledge domains suggests these features serve as general-purpose routing or gating infrastructure rather than domain-specific knowledge stores. A feature that activates for chemistry, geography, AND history queries is unlikely to encode any specific factual content; instead, it may implement a more general function such as attention routing, information filtering, or output formatting.

The notable descriptions of Gemma's universal features support this interpretation: "HTML formatting tags," "code/file keywords," and "place names" are structural/format-related features, not factual knowledge features.

Co-activation analysis further reveals that these universal features organize into within-layer functional modules: same-layer co-activation is 1.9× stronger than cross-layer, and hierarchical clustering identifies 5 distinct feature groups. This modular organization suggests a structured routing infrastructure rather than a random collection of

individually useful features.

Polysemanticity analysis provides additional converging evidence. CODE (20%) and LANGUAGE (20%) features, encoding formatting, syntax, and linguistic patterns, outnumber domain-specific features (13%). 54% of features appear in ≥2 domains and 18% in all 3.

The low overall polysemanticity rate (12.3%) suggests bottleneck features are more specialized than the average transcoder feature: they are selected by the circuit architecture for specific routing functions, even when those functions span multiple knowledge domains. (See Supplementary Materials S1.11, S1.15)

6.3 Output Convergence vs Path Divergence

A novel finding from our enhanced analysis is that history circuits share output features far more than intermediate features (output/path ratio = 4.41× in Qwen). This "convergent output, divergent paths" pattern suggests that the model uses different internal routes to arrive at a similar output vocabulary for the same knowledge type. This is especially pronounced in Qwen's history domain, where prompts about events spanning 500 years share 67.3% of output features but only 15.3% of path features.

This has implications for model editing: intervening at the output layer would affect most circuits within a domain, while intervening at intermediate layers requires more domain-specific targeting.

6.4 Confidence Is Driven by Circuit-Level Activation, Not Bottleneck Quality: The "Bottleneck Penalty"

Our combined statistical and per-layer analyses reveal a pattern linking circuit structure to behavior. In Gemma, total activation magnitude is the strongest single predictor of confidence ($r = 0.527$, Bonferroni-surviving), and a 3-predictor regression model (activation magnitude, kurtosis, circuit size) explains 49% of variance ($p < 0.001$).

Critically, bottleneck convergence does NOT predict confidence ($r = 0.270$, $p = 0.15$), and bottleneck activation share specifically does NOT predict confidence ($r = 0.256$, $p = 0.17$). It is not the quality or concentration of information compression that matters.

The core observation. Per-layer analysis reveals that Gemma L6 (the primary bottleneck layer) shows a strong negative correlation with confidence ($r = -0.684$, $p < 0.0001$), while activation at post-bottleneck layers (L13: $r = +0.601$; L16: $r = +0.598$) positively predicts confidence. We term this pattern the "bottleneck penalty": heavier bottleneck-layer activation is associated with lower downstream confidence.

Theoretical grounding: information bottleneck theory. We interpret this pattern through information bottleneck theory (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017). The theory formalizes the tradeoff between compression and prediction in deep networks: each layer L computes a representation T_L of the input X with mutual information $I(X; T_L)$, and heavier compression at an intermediate layer reduces the information available to downstream layers for reconstruction and prediction.

Our observation, that more bottleneck-layer activation predicts worse confidence, is consistent with the prediction that over-compression at L6 limits the information downstream layers (L13, L16) can use for evidence accumulation. We emphasize "consistent with" rather than "proof of": our measurements are of activation magnitude, not mutual information, and our analysis is correlational, not interventional.

A nuance from output decomposition. While the bottleneck penalty operates at the circuit level (more overall bottleneck weight = lower confidence), the critical paths show the opposite pattern: paths that route *more* through bottleneck layers yield *higher* confidence ($r = 0.415$, $p = 0.016$).

Bottleneck layers are simultaneously the most important processing hubs and the most costly when over-used. This dual role, essential routing infrastructure that imposes compression costs when overloaded, is consistent with the information bottleneck framework, where a well-used bottleneck that preserves task-relevant information improves prediction while an over-compressed bottleneck harms it.

Relation to prior work. This finding connects to Meng et al.'s (2022) causal tracing results. Where Meng et al. identified specific MLP layers as sites of factual association storage via interventions, our per-layer correlation analysis provides a non-interventional analog, a "correlational causal tracing" that identifies which layers' activation patterns predict behavioral outcomes.

The convergent finding (specific layers matter, not all layers equally) is compatible with both approaches, though interventional validation remains a target for future work.

What the bottleneck penalty is NOT. Activation magnitudes at different layers are computed independently; no conserved quantity is being allocated across layers.

The bottleneck penalty is a correlation, not a mechanistic causal claim. Alternative explanations remain viable: prompt difficulty confounds, domain-specific accuracy differences, and feature-selection effects in traceback. Our contribution is the empirical pattern and its consistency with information bottleneck theory, not a proof of mechanism.

Qwen shows no per-layer confidence relationships (0 Bonferroni-significant layers), consistent with its late-layer architecture diffusing confidence signals across many layers. The information bottleneck framework predicts that architectures with diffuse compression should exhibit weaker per-layer correlations with output quality, which is what we observe. (See Supplementary Materials S1.16)

6.5 Causal Validation: Redundancy, Visibility, and the Three-Tier Model

Minimal pathway analysis reveals that circuits are 94.1% redundant: only ~6% of nodes participate in the essential input-to-output pathways, and these pathways carry just 0.3% of total edge weight. Gemma's redundancy positively correlates with confidence ($r = 0.642$), suggesting a functional role for this redundancy.

More redundant circuits may represent more durable, well-established factual memories where the model has encoded multiple parallel pathways to the same answer. This "redundancy-as-robustness" hypothesis aligns with ensemble interpretations of neural network computation.

Our steering analysis reveals a methodological gap: features that exert causal influence through steering are invisible in the attribution graph. None of the 5 initially steered features appeared as nodes in their circuits' graphs. This "visibility gap" means that circuit-based analysis underestimates the causally relevant feature set: the attribution graph captures only above-threshold activations, while causal influence extends to sub-threshold features.

Expanded steering reveals a disconnect between structural importance (cross-circuit frequency) and causal influence (steering effect). L2_F25751073, appearing in the most circuits (11), produced zero text changes when steered, while L7_F4828270 (9 circuits) was the most effective.

Frequent activation across circuits reflects statistical co-occurrence rather than causal necessity. A feature can be reliably present without being mechanistically essential; it may be part of the redundant scaffolding (94.1% redundancy) rather than the essential pathway.

The domain asymmetry in steering susceptibility (history > geography > chemistry = 0) has a natural interpretation in terms of output determinism (§3.9). Chemistry circuits produce the most deterministic outputs (Na, Fe, Pb are near-certain completions), making them resistant to perturbation. History and geography outputs have more distributional uncertainty, creating more room for steering to shift the argmax.

The essential-pathway steering validation reveals a **three-tier dissociation** between structural importance metrics and causal influence. Features on the minimal viable pathway produce the strongest distributional perturbations (mean KL = 1.448), but neither pathway position (26.7% text change rate) nor cross-circuit frequency (23.3% for frequency-selected features) reliably predicts text-level output changes.

This dissociation suggests that the attribution graph captures two distinct types of information: (1) the essential routing topology, which determines how strongly a feature can perturb probability distributions, and (2) the

redundancy structure, which determines whether that perturbation survives to change the argmax output token.

Features on essential pathways are genuinely important for information flow (their perturbations propagate strongly through the circuit), but the model's 94.1% circuit redundancy provides compensatory pathways that absorb these perturbations before the output layer.

The layer-dependent steering direction (early features respond to suppression only; late features respond to both directions) further suggests a functional partition: early-layer pathway features serve as information gates whose removal disrupts processing, while late-layer pathway features actively shape output distributions and respond to amplification. (See Supplementary Materials S1.17, S1.19, S1.20, S1.21)

7. Limitations

1. **Two models only.** Our findings are based on Gemma-2-2B and Qwen3-4B. Larger models, different architectures, or instruction-tuned variants may exhibit different patterns. The architecture-dominance claim is supported by between-architecture comparison of these two models, not a population of architectures.

2. **Sample size and statistical power.** With n = 30 per model (10 per category), correlations below $|r| = 0.45$ are unreliable at $\alpha = 0.05$, and only $|r| \geq 0.60$ survives Bonferroni correction. Medium-sized effects (Cohen's d = 0.5–0.8) may exist in our ANOVA comparisons but remain undetectable. The full regression (9 predictors, 21 residual df) is underpowered; we report reduced models as primary results.

3. **SAE quality and Qwen annotation gap.** 114 of 244 cross-circuit features (all Qwen) lack Neuronpedia explanations, limiting polysemanticity analysis to Gemma features only. The semantic classification covers only 130 Gemma features (53.3%), and classification is regex-based on Neuronpedia explanations, which may misclassify features with ambiguous descriptions.

4. **Prompt format effects.** Our format variation experiment (§5.9) shows that prompt template drives 1.6× more feature-level overlap than factual identity for chemistry and geography, though not for history. The within-domain Jaccard values in §5.1 are partially inflated by template similarity for these domains. The architecture-dominance finding (§5.5) uses layer activation profiles, not feature Jaccard, and is robust to this confound. The pilot tests only one fact (gold); extending to all facts and domains would strengthen this analysis.

5. **Correlational, not interventional.** Our per-layer activation-confidence correlations (§5.5.3) and the bottleneck penalty interpretation are based on observational data across 30 circuits per model. The information bottleneck framing is a theoretical interpretation, not a proven mechanism. Direct interventional validation (clamping bottleneck-layer activations and measuring confidence change) is future work requiring local model access not available through the Neuronpedia API.

6. **Limited steering coverage.** The 80-experiment steering validation provides initial causal evidence but is limited to Gemma (Qwen SAE steering is not yet supported by the Neuronpedia API), uses only ±20 strength with 3 target circuits per experiment, and tests 15 features total. Dose-response curves at varying strengths and broader feature coverage would strengthen the three-tier dissociation finding.

7. **Community degeneracy.** The multi-algorithm validation (4 methods, 60 circuits) shows high modularity (>0.4 for Louvain and Greedy Modularity) but low cross-algorithm agreement (mean Jaccard = 0.363). Community boundaries are algorithm-dependent while the existence of community structure is robust. Analyses depending on specific community assignments should be interpreted as one valid partition among many.

8. **Visibility gap and circuit extraction method.** Steered features are causally relevant but invisible in the circuit representation, suggesting that threshold-based graph extraction systematically underestimates the causally relevant feature set. Our circuits are generated via Neuronpedia's Circuit Tracer API, which uses activation-based attribution. Alternative methods (ACDC, Conmy et al., 2023; attribution patching, Neel & Steinhardt, 2023; causal scrubbing) use different extraction criteria and may yield different circuit structures. Future work should compare extraction methods on the same prompts.

8. Conclusion

We presented a systematic cross-domain analysis of factual knowledge circuits in large language models, analyzing 60 circuits across three knowledge domains and two architectures with 80 causal steering experiments. Our findings:

1. **Architecture dominates circuit structure.** Within-model layer activation profile similarity is 0.978 versus 0.696 between models; architecture produces approximately 14× the mean cosine gap that knowledge domain does. Gemma is front-loaded (54% of activation in the first half), Qwen back-loaded (31%), yet both achieve comparable recall. 13/16 structural metrics differ significantly between architectures (Mann-Whitney, Bonferroni-corrected).

2. **The bottleneck penalty links circuit structure to behavior.** In Gemma, activation magnitude at the bottleneck layer L6 negatively predicts confidence ($r = -0.684$, Bonferroni-significant), while post-bottleneck layers positively predict it. A 3-predictor regression explains 49% of confidence variance ($p < 0.001$).

We interpret this pattern through information bottleneck theory (Tishby & Zaslavsky, 2015): over-compression at an intermediate layer reduces information available for downstream evidence accumulation. The finding is correlational; interventional validation is future work.

3. **Within-domain convergence is real but small.** Permutation tests ($p < 0.0001$) confirm domain-specific circuits, but domain effects account for only ~2% of layer activation profile variance. Geography shows highest convergence (Jaccard = 0.39), history lowest (0.11).

4. **Universal bottleneck features are architecture-specific infrastructure.** 6 Gemma and 15 Qwen features appear in all three domains with zero cross-model overlap. Semantic classification reveals CODE (20%) and LANGUAGE (20%) dominance: routing infrastructure, not knowledge stores.

5. **Output features converge more than intermediate features.** Same-domain circuits share up to 67.3% of output vocabulary (Qwen history) despite only 15.3% path overlap: convergent predictions from divergent routes.

6. **Circuits are 94% redundant.** Only 6% of nodes and 0.3% of edge weight participate in essential pathways. Gemma redundancy positively correlates with confidence ($r = 0.642$), suggesting redundancy encodes robustness.

7. **Causal validation reveals a three-tier dissociation.** 80 steering experiments show that essential-pathway features produce the strongest distributional perturbations (mean KL = 1.448), but neither pathway topology nor cross-circuit frequency predicts text-level output changes.

Circuit redundancy absorbs perturbations, and output determinism (§3.9), not feature selection, governs steering susceptibility (chemistry 0%, geography 20%, history 60%).

These findings demonstrate that factual knowledge retrieval is constrained by model architecture across the full layer activation profile. The bottleneck penalty provides a structure-behavior link: confident predictions arise from evidence accumulation beyond the bottleneck, not from compression quality at the bottleneck itself.

The three-tier causal dissociation, where topology, frequency, and text-level effects decouple, reveals that attribution graphs capture only a fraction of causally relevant computation, mediated by layers of circuit redundancy.

Future Work

- Extend to more models and domains
- Expand steering with dose-response curves and more circuits
- Develop sub-threshold circuit extraction to address the visibility gap

- Test varied prompt formats to disentangle format from domain effects
- Annotate remaining Qwen features when API support becomes available
- Build predictive models combining structural and causal metrics

References

1. Ameisen, E., Lindsey, J., et al. (2025). Circuit Tracing: Revealing Computational Graphs in Language Models. Transformer Circuits Thread. https://transformer-circuits.pub/2025/attribution-graphs/methods.html

2. Bach, S., et al. (2015). On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7), e0130140.

3. Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.

4. Bricken, T., et al. (2023). Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. Anthropic.

5. Conmy, A., et al. (2023). Towards Automated Circuit Discovery for Mechanistic Interpretability. NeurIPS 2023. arXiv:2304.14997.

6. Cunningham, H., et al. (2024). Sparse Autoencoders Find Highly Interpretable Features in Language Models. ICLR 2024.

7. Dunefsky, J., Chlenski, P., Rajamanoharan, S., & Nanda, N. (2024). Transcoders Find Interpretable LLM Feature Circuits. NeurIPS 2024. arXiv:2406.11944.

8. Elhage, N., et al. (2021). A Mathematical Framework for Transformer Circuits. Anthropic.

9. Ferrando, J., & Voita, E. (2024). Information Flow Routes: Automatically Interpreting Language Models at Scale. EMNLP 2024. arXiv:2403.00824.

10. Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 40(1), 35–41.

11. Kullback, S., & Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79–86.

12. Lieberum, T., et al. (2024). Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2. BlackboxNLP 2024. arXiv:2408.05147.

13. Lindsey, J., Gurnee, W., Ameisen, E., et al. (2025). On the Biology of a Large Language Model. Transformer Circuits Thread. https://transformer-circuits.pub/2025/attribution-graphs/biology.html

14. Meng, K., et al. (2022). Locating and Editing Factual Associations in GPT. NeurIPS 2022.

15. Neuronpedia. Decode Research. https://neuronpedia.org

16. Olah, C., et al. (2020). Zoom In: An Introduction to Circuits. Distill.

17. Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning Important Features Through Propagating Activation Differences (DeepLIFT). ICML 2017. arXiv:1704.02685.

18. Shwartz-Ziv, R., & Tishby, N. (2017). Opening the Black Box of Deep Neural Networks via Information. arXiv:1703.00810.

19. Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks (Integrated Gradients). ICML 2017. arXiv:1703.01365.

20. Syed, A., Rager, C., & Conmy, A. (2024). Attribution Patching Outperforms Automated Circuit Discovery. BlackboxNLP 2024. arXiv:2310.10348.

21. Templeton, A., et al. (2024). Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. Anthropic.

22. Tishby, N., & Zaslavsky, N. (2015). Deep Learning and the Information Bottleneck Principle. IEEE Information Theory Workshop (ITW).

23. Turner, A., et al. (2023). Steering Language Models With Activation Engineering. arXiv:2308.10248.

24. Wang, K., et al. (2023). Interpretability in the Wild: A Circuit for Indirect Object Identification in GPT-2 Small. ICLR 2023.

Appendix A: Prompt List

Chemistry (10 prompts)

#	Prompt	Expected Answer
C1	The chemical symbol for gold is	Au
C2	The chemical symbol for iron is	Fe
C3	The chemical symbol for lead is	Pb
C4	The chemical symbol for mercury is	Hg
C5	The chemical symbol for argon is	Ar
C6	The chemical symbol for sodium is	Na
C7	The chemical symbol for silver is	Ag
C8	The chemical symbol for potassium is	K
C9	The chemical symbol for copper is	Cu
C10	The chemical symbol for tungsten is	W

Geography (10 prompts)

#	Prompt	Expected Answer
G1	The capital of France is	Paris
G2	The capital of Japan is	Tokyo
G3	The capital of Germany is	Berlin
G4	The capital of Egypt is	Cairo
G5	The capital of Italy is	Rome
G6	The capital of Spain is	Madrid
G7	The capital of Canada is	Ottawa
G8	The capital of Thailand is	Bangkok
G9	The capital of Turkey is	Ankara
G10	The capital of South Korea is	Seoul

History (10 prompts)

#	Prompt	Expected Answer
H1	World War II ended in	1945
H2	The first moon landing was in	1969
H3	The Berlin Wall fell in	1989
H4	The Titanic sank in	1912
H5	America declared independence in	1776
H6	The French Revolution began in	1789
H7	World War I started in	1914
H8	Columbus reached the Americas in	1492
H9	The Great Fire of London occurred in	1666
H10	Nelson Mandela was released from prison in	1990

Appendix: Statistical Figures

The following figures visualize the statistical results discussed throughout the paper. All figures are generated by the scripts in `scripts/stage_2_*_figures.py` from the data in `data/stage_2_*/`.

Pipeline and Core Statistics

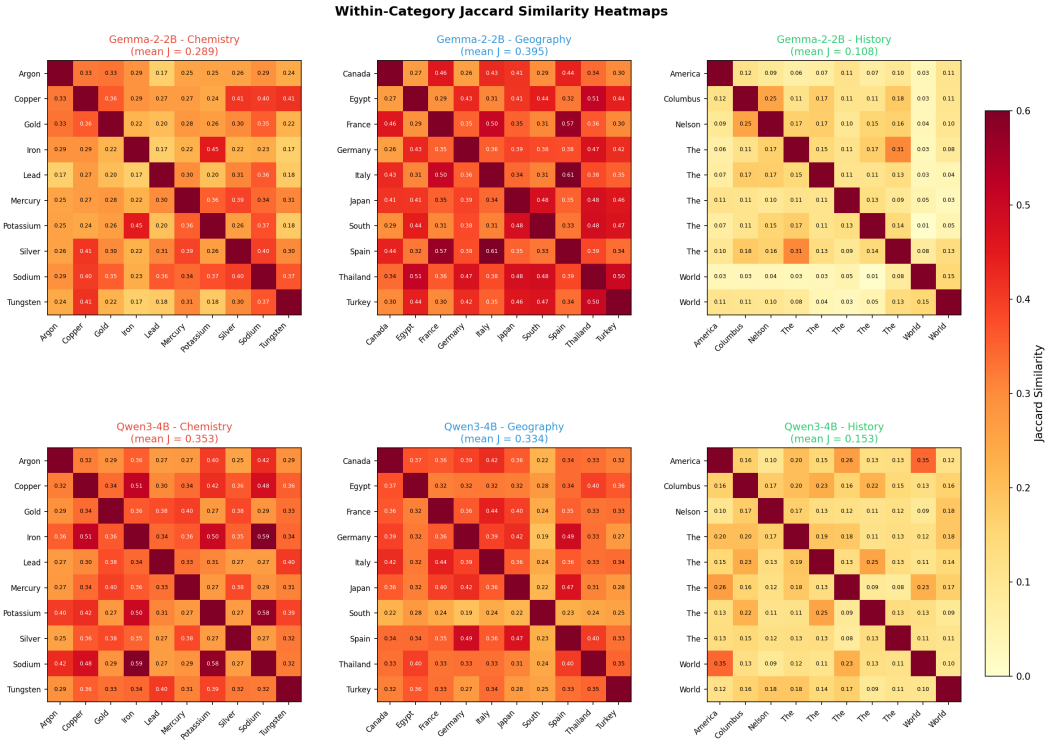

Figure 2: Jaccard Heatmaps

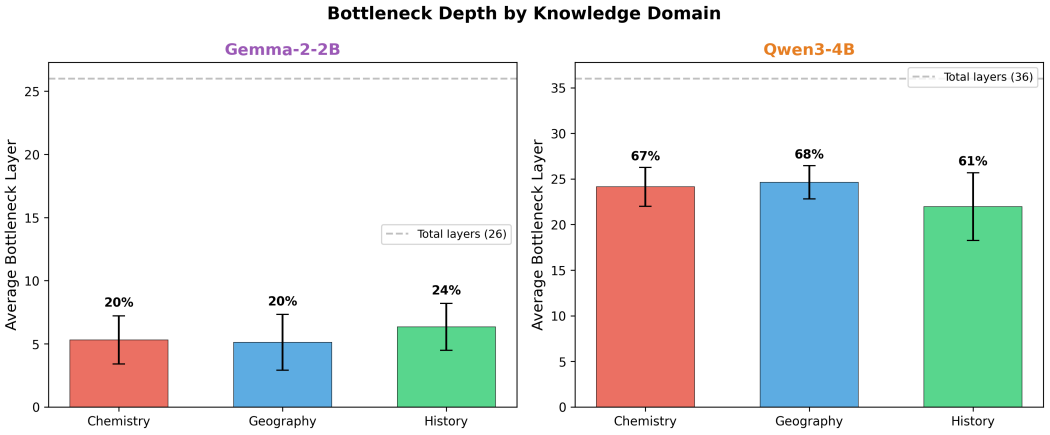

Figure 3: Bottleneck Depth

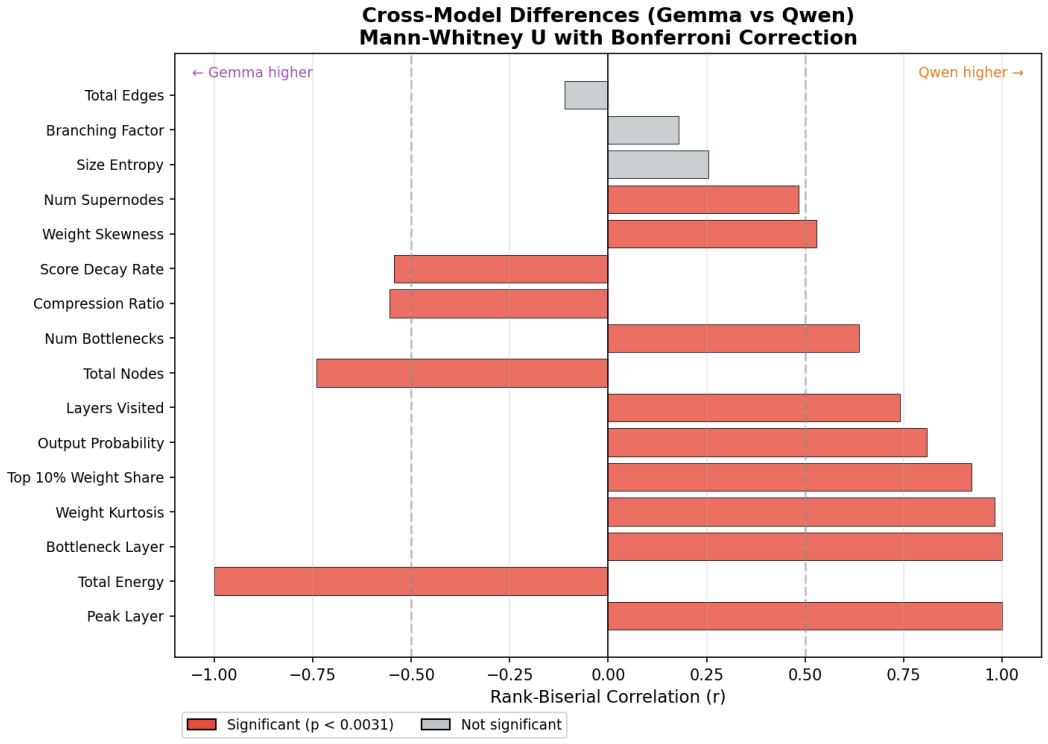

Figure 7: Cross Model

Layer Energy and Cross-Model Analysis

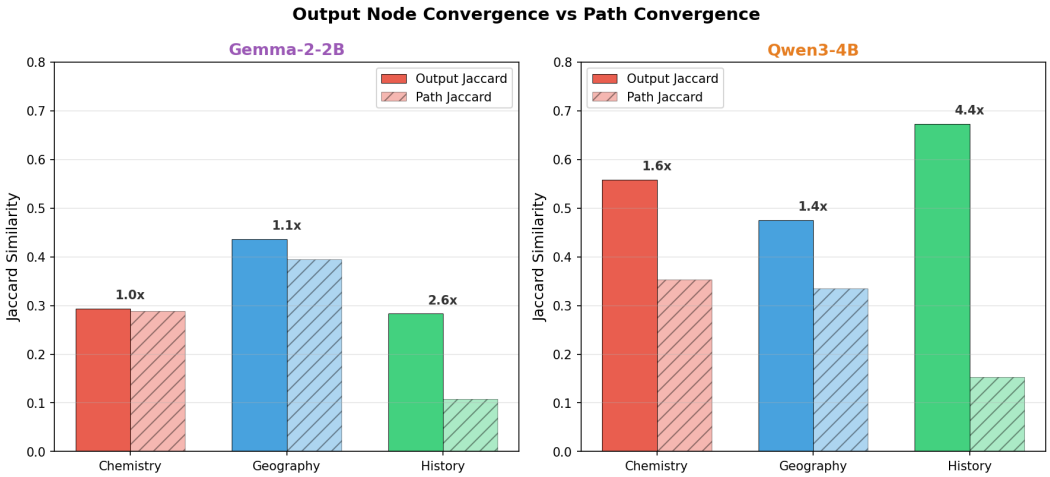

Figure 10: Output Path

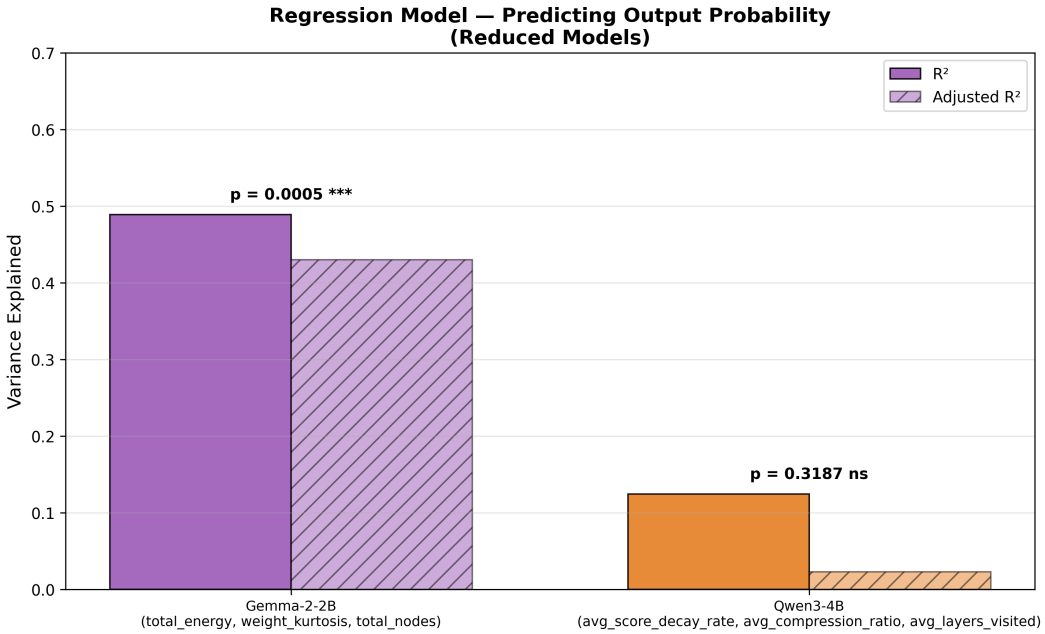

Figure 14: Regression

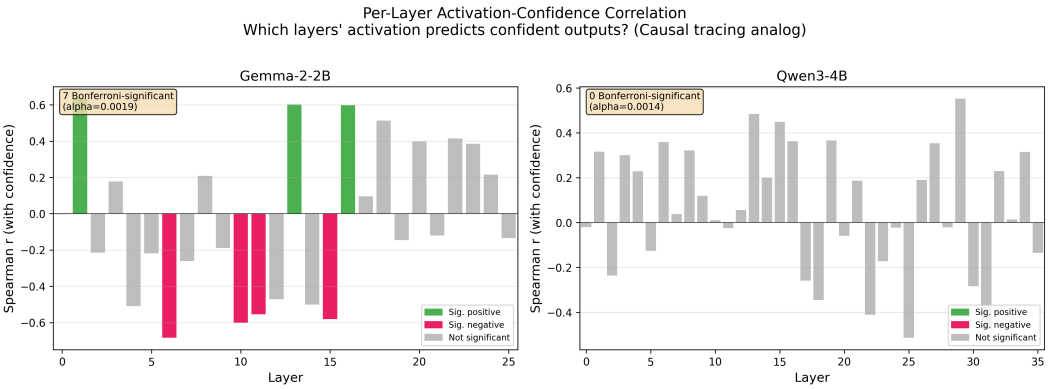

Figure 18: Layer Confidence Correlation

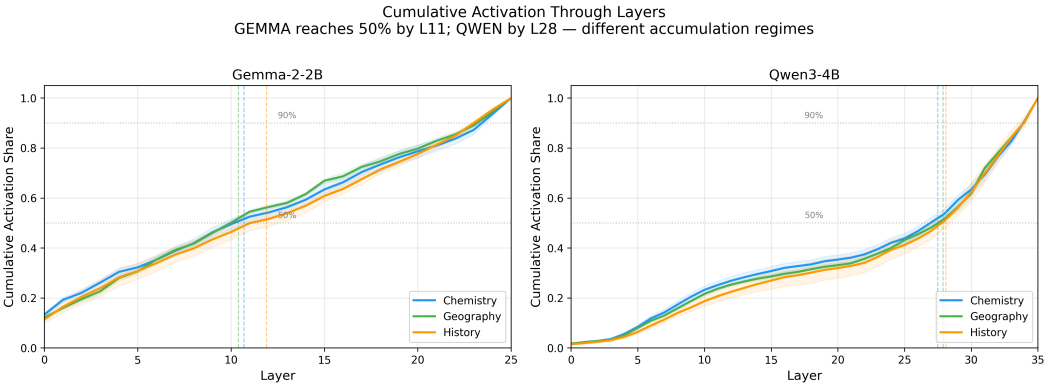

Figure 19: Cumulative Energy

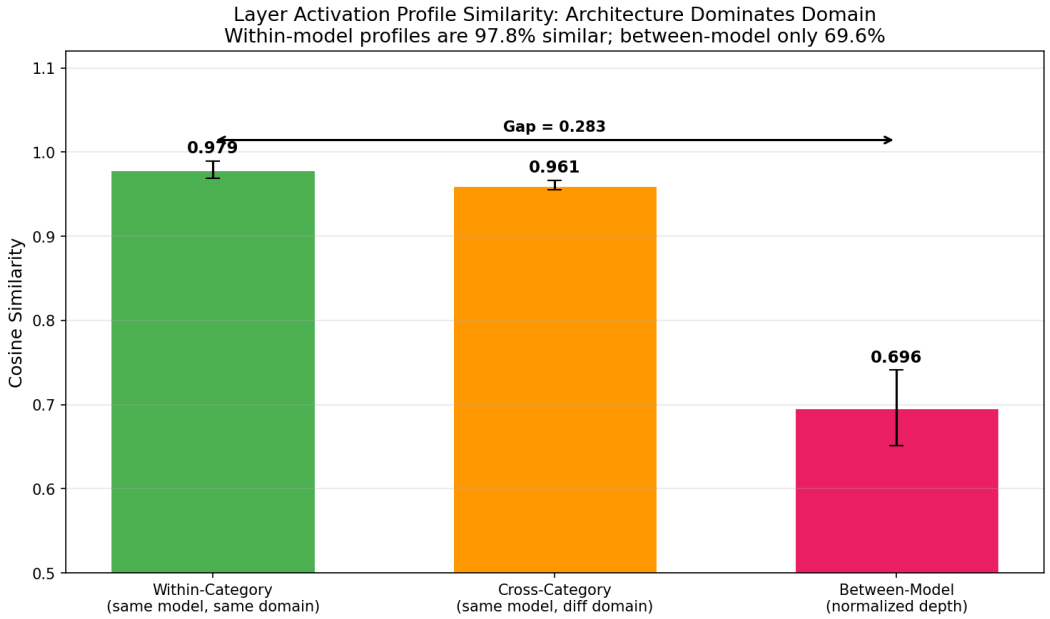

Figure 20: Profile Similarity

Category Breakdowns and Domain Effects

Regression and Statistical Deepdive

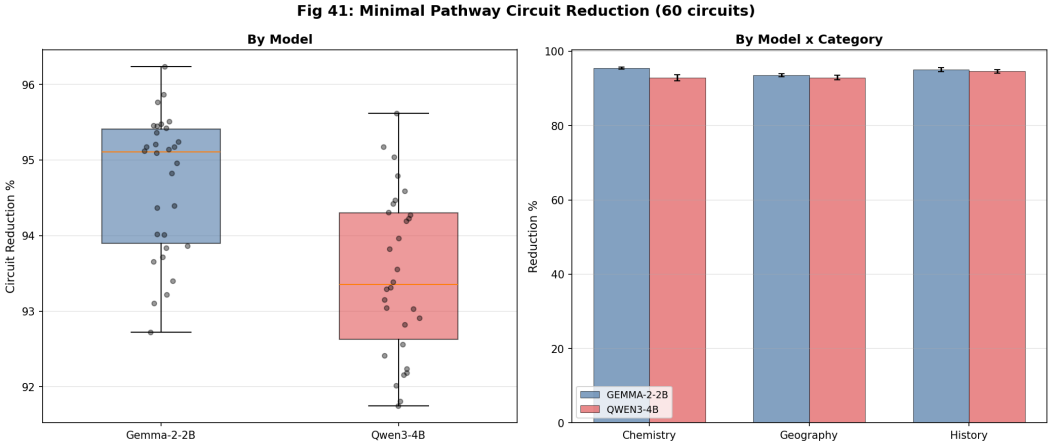

Figure 41: Reduction Breakdown

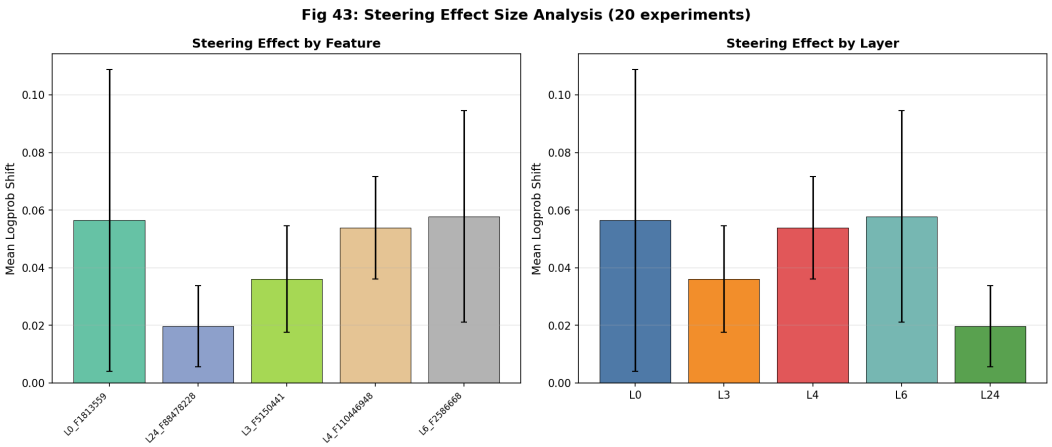

Figure 43: Steering Effects

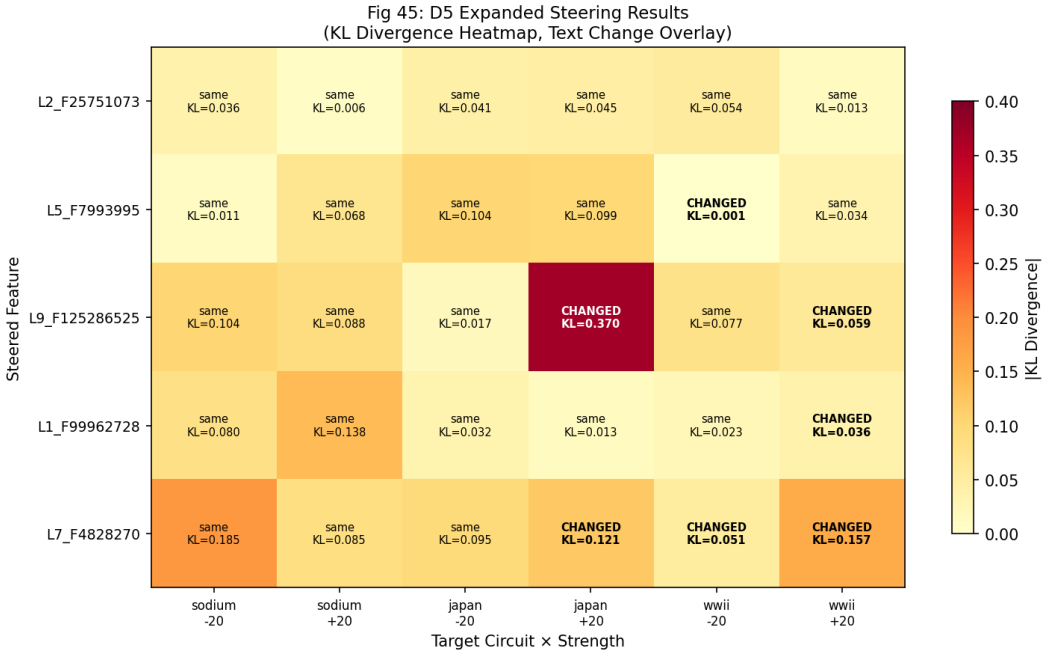

Figure 45: D5 Steering Heatmap