arXiv:2607.23054v1 [cs.LG] 25 Jul 2026

Through the Bottleneck:
How Multi-head Latent Attention Separates Content from Position in Language Models

Dhruvil S[*,1] **Fenil Sojitra**[*,2]
Ravirajsinh Chauhan[2]
[1]Indian Institute of Technology Madras
[2]P P Savani University
23f2004889@ds.study.iitm.ac.in 22se02it061@ppsu.ac.in
raviraj.chauhan@ppsu.ac.in

Abstract

Multi-head Latent Attention (MLA), introduced in DeepSeek-V2, compresses key-value pairs through a shared low-rank bottleneck (c_{KV}), achieving 81% KV-cache reduction during inference. Despite its adoption in production models with hundreds of billions of parameters, no prior work has studied *what information this bottleneck preserves or discards*, nor how it reshapes the internal circuits of a transformer. We present the first comprehensive mechanistic interpretability study of MLA, training a 114M-parameter MLA transformer (pretrained on a web/code/math mixture, fine-tuned on TinyStories) and analyzing its internal representations through four complementary experiments: singular value decomposition of compression matrices, attention head taxonomy, linear probing, and a c_{KV} disruption-attribution analysis. Our key findings are: (1) the c_{KV} bottleneck learns a pure **content representation**, preserving entity identity (98% retention) while discarding positional information (near chance accuracy), validating MLA's architectural separation of content from position via RoPE; (2) induction heads **co-locate at a single layer** (Layer 12), unlike their distributed formation in standard MHA, suggesting the shared bottleneck constrains circuit topology; (3) a single "semantic hub" layer (Layer 15) simultaneously exhibits the highest SVD effective rank and the strongest disruption-attribution score; and (4) the bottleneck is globally **over-provisioned**, using only 46% of its 128-dimensional capacity on average, with concrete implications for heterogeneous rank allocation. These findings, from a single 114M-parameter model trained on a narrow-domain corpus, suggest that MLA does not merely compress attention passively; it appears to reshape how the model organizes content, position, and circuit structure. We view this as an initial data point rather than a settled characterization of MLA at scale, and we detail the scope limitations in Section 5.

1 Introduction

Multi-head Latent Attention (MLA) [DeepSeek-AI, 2024a] has rapidly become a key architectural innovation in efficient large language models. By compressing key-value (KV) representations through a shared low-rank bottleneck before projecting them into per-head keys and values, MLA achieves dramatic reductions in KV-cache memory, up to 81% compared to standard Multi-Head Attention (MHA), without sacrificing generation quality. This design has been adopted in production-scale models including DeepSeek-V2 (236B parameters) and DeepSeek-V3 (671B parameters).

Despite the practical success of MLA and the growing field of mechanistic interpretability [Elhage et al., 2021, Olsson et al., 2022, Conmy et al., 2023, Nanda et al., 2022], **no prior work has studied what happens inside the MLA bottleneck**. Specifically:

[*]Equal contribution.
Code and data available at https://github.com/Dhruvil-sr24/Small-Language-Model-From-Scratch/tree/main/interp

- What linguistic information does the model choose to *preserve* in the compressed c_{KV} representation?
- What information does it *discard*, and why?
- How does the shared bottleneck affect the formation of known circuits (e.g., induction heads)?
- Which layers' bottlenecks are *causally* important for the model's predictions?

We address these questions through a comprehensive mechanistic study of a 114M-parameter MLA transformer, pretrained on a mixture of web text, code, and mathematical text and then fine-tuned on TinyStories [Eldan & Li, 2023] (see Section 3 for the full data mixture).Our four experiments (SVD analysis, attention head taxonomy, linear probing, and c_{KV} disruption attribution) converge on a coherent picture of how MLA reshapes transformer information processing.

Contributions.

1. We show that c_{KV} learns a **content-only representation**: entity information is preserved with 98% fidelity, while positional information is discarded (near-chance probe accuracy). This validates MLA's design of separating content (c_{KV}) from position (RoPE k_{rope}).
2. We discover that induction heads **co-locate at a single layer** in MLA (all 5 detected induction heads at Layer 12), in contrast to their distributed formation in standard MHA [Olsson et al., 2022].
3. We identify a **semantic hub** at Layer 15 that simultaneously has the highest SVD effective rank (88/128) and the highest disruption-attribution score among all layers.
4. We demonstrate that the bottleneck is **globally over-provisioned**: only 46% of the 128-dim capacity is used on average. We propose heterogeneous rank allocation as a concrete architectural improvement.

2 Background

2.1 Multi-head Latent Attention

Standard MHA computes queries, keys, and values independently for each head from the residual stream $\mathbf{x} \in \mathbb{R}^d$:

$$\mathbf{q}_h = W_Q^h \mathbf{x}, \quad \mathbf{k}_h = W_K^h \mathbf{x}, \quad \mathbf{v}_h = W_V^h \mathbf{x} \tag{1}$$

This requires caching $\mathbf{k}_h$ and $\mathbf{v}_h$ for all heads during autoregressive generation, costing $O(n_{\text{heads}} \cdot d_{\text{head}})$ memory per token per layer.

MLA introduces a **shared KV compression bottleneck**:

$$\mathbf{c}_{\text{KV}} = \text{RMSNorm}(W_{\text{DKV}} \cdot \mathbf{x}) \in \mathbb{R}^{r_{\text{KV}}} \tag{2}$$

$$\mathbf{k}_h^{\text{nope}} = W_{UK}^h \cdot \mathbf{c}_{\text{KV}}, \quad \mathbf{v}_h = W_{UV}^h \cdot \mathbf{c}_{\text{KV}} \tag{3}$$

$$\mathbf{k}_h^{\text{rope}} = \text{RoPE}(W_{KR}^h \cdot \mathbf{x}) \tag{4}$$

$$\mathbf{k}_h = [\mathbf{k}_h^{\text{nope}}; \mathbf{k}_h^{\text{rope}}] \tag{5}$$

During inference, only $\mathbf{c}_{\text{KV}} \in \mathbb{R}^{r_{\text{KV}}}$ needs to be cached (rather than all per-head K/V), achieving compression ratio $r_{\text{KV}}/(n_{\text{heads}} \cdot d_{\text{head}})$.

Critically, Eq. 2 and 4 reveal an architectural separation: **content information** flows through $\mathbf{c}_{\text{KV}}$ (shared across heads), while **positional information** flows through the separate RoPE pathway $\mathbf{k}_h^{\text{rope}}$ (directly from $\mathbf{x}$, bypassing the bottleneck). Whether the model actually learns to exploit this separation is an empirical question we answer in Section 4.3.

2.2 Mechanistic Interpretability

We build on established mechanistic interpretability techniques:

- **Linear probing** [Belinkov et al., 2017, Alain & Bengio, 2016]: Training a linear classifier on intermediate representations to test what information they encode: if a linear probe achieves high accuracy, the information is linearly accessible in the representation.
- **Activation patching** [Meng et al., 2022, Wang et al., 2023]: the classic version of this technique corrupts inputs and then surgically restores clean activations at specific (layer, position) pairs, measuring how much output is recovered, establishing genuinely causal (not merely correlational) importance. In Section 4.4, we use a lighter-weight variant, c_{KV} **disruption attribution**, which measures how much corruption *changes* c_{KV} at each (layer, position) and reweights this by the resulting drop in output probability, without performing the restoration step. This is cheaper to compute but establishes correlational, not strictly causal, importance; we flag this distinction explicitly where it matters (Section 5.4).
- **Induction head analysis** [Olsson et al., 2022]: Identifying attention heads that implement the "induction" circuit (copying tokens that previously followed a matching prefix), a fundamental in-context learning mechanism.

3 Experimental Setup

3.1 Model Architecture

Our model is a 24-layer MLA transformer with the following configuration:

Table 1: Model architecture details.

Parameter	Value	Notes
Layers	24	All transformer blocks
Hidden dim (d_{model})	512	Residual stream width
Attention heads (n_h)	8	
KV heads (n_{kv})	8	Grouped Query Attention
Head dim (d_{head})	64	$d_{\text{nope}} + d_{\text{rope}}$
KV bottleneck (r_{KV})	128	**Key: 4× compression**
Q bottleneck (r_Q)	256	2× compression
QK nope dim	48	Content dimension
QK rope dim	16	Position dimension
Total parameters	114.1M	
KV-cache reduction	81%	vs. standard MHA

3.2 Training

The model was pretrained for 16,500 steps on a mixed corpus: 60% FineWeb-Edu [Penedo et al., 2024] (general web text, for foundational language and grammar), 25% The Stack [Kocetkov et al., 2022] (source code, for coding logic and structured reasoning), and 15% OpenWebMath [Paster et al., 2023] (mathematical web text, for introductory-level mathematical language). It was then fine-tuned for 7,200 steps on TinyStories [Eldan & Li, 2023], achieving a perplexity of 4.3 on the TinyStories validation set. All interpretability analyses in this paper are run on the *fine-tuned* checkpoint, using TinyStories-distribution passages as the analysis input unless otherwise stated, so while the model's representations were shaped by a broader and more heterogeneous pretraining mixture (web text, code, and math) than TinyStories alone, the probing, attention-taxonomy, and disruption-attribution results reflect how the model behaves once specialized to the simpler TinyStories domain. We use a BPE tokenizer with vocabulary size 32,768.

3.3 Experiments Overview

We conduct four experiments, each probing a different aspect of the MLA bottleneck:

1. **SVD Analysis** (Section 4.1): Decompose compression matrices W_{DKV} and W_{DQ} to measure effective dimensionality.
2. **Attention Patterns** (Section 4.2): Classify all $24 \times 8 = 192$ heads by function (previous-token, induction, BOS-sink, local window).
3. **Linear Probing** (Section 4.3): Compare what linguistic features are linearly decodable from $\mathbf{c}_{\text{KV}}$ (128-dim) vs. the residual stream (512-dim).
4. **$c_{\mathbf{KV}}$ Disruption Attribution** (Section 4.4): Measure how strongly corruption-induced changes to $\mathbf{c}_{\text{KV}}$ at each layer correlate with degraded predictions, as a lightweight proxy for causal importance.

All experiments are inference-only and run on a single NVIDIA T4 GPU.

4 Results

4.1 SVD Analysis: The Bottleneck is Over-Provisioned

We compute the SVD of $W_{\text{DKV}} \in \mathbb{R}^{128\times512}$ at each of the 24 layers and measure the *effective rank*: the number of singular values needed to capture 99% of the total energy ($\sum \sigma_i^2$).

Table 2: Effective rank of KV and Q compression matrices.

Matrix	Mean Rank	Min (Layer)	Max (Layer)
W_{DKV} (512 → 128)	58.8 / 128 (46%)	44 (L6)	88 (L15)
W_{DQ} (512 → 256)	99.0 / 256 (39%)	41 (L15)	130 (L21)

Key finding: On average, the model uses only **46% of its 128-dim KV bottleneck capacity**. At the 90% energy threshold, only 17 dimensions suffice on average—suggesting significant over-provisioning.

Layer heterogeneity: Three layers exhibit markedly higher effective rank: Layer 7 (85), Layer 11 (79), and Layer 15 (88). These "rank spikes" correspond to architectural phase transitions, as we show in subsequent experiments.

Anti-correlation of KV and Q ranks: When W_{DKV} uses more dimensions (e.g., Layer 15: rank 88), W_{DQ} uses fewer (Layer 15: rank 41), and vice versa. This suggests the model learns an **asymmetric capacity allocation**: when more information is needed about what positions attend *to* (KV), less is needed about what positions attend *from* (Q).

4.2 Attention Patterns: Induction Heads Co-locate

We classify all 192 attention heads using four metrics computed on 50 TinyStories passages (sequence length 64): (1) *previous-token score*: fraction of attention on position $t-1$; (2) *induction score*: attention on positions following a matching prefix; (3) *BOS score*: attention on the beginning-of-sequence token; (4) *local window score*: attention within a 5-token window.

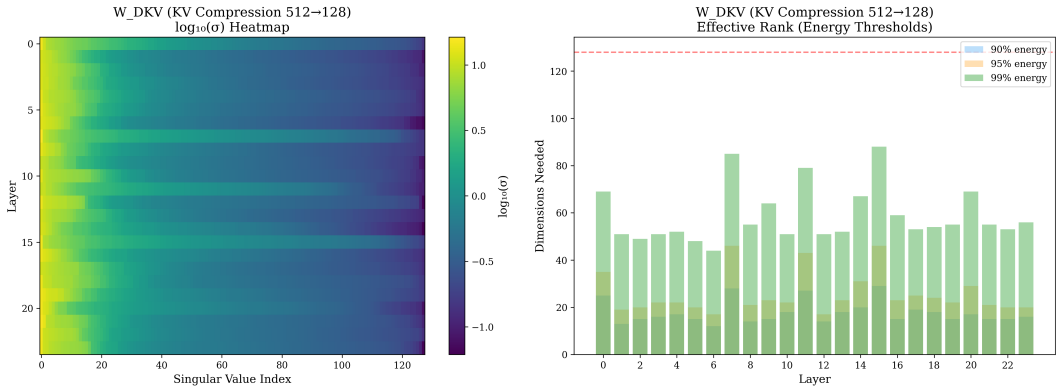

Figure 1: Left: log singular-value magnitude of W_{DKV} by layer, showing rapid decay after roughly the first 20–30 components at every layer. Right: effective rank at 90/95/99% energy thresholds per layer, all well below the full 128-dim capacity (dashed line).

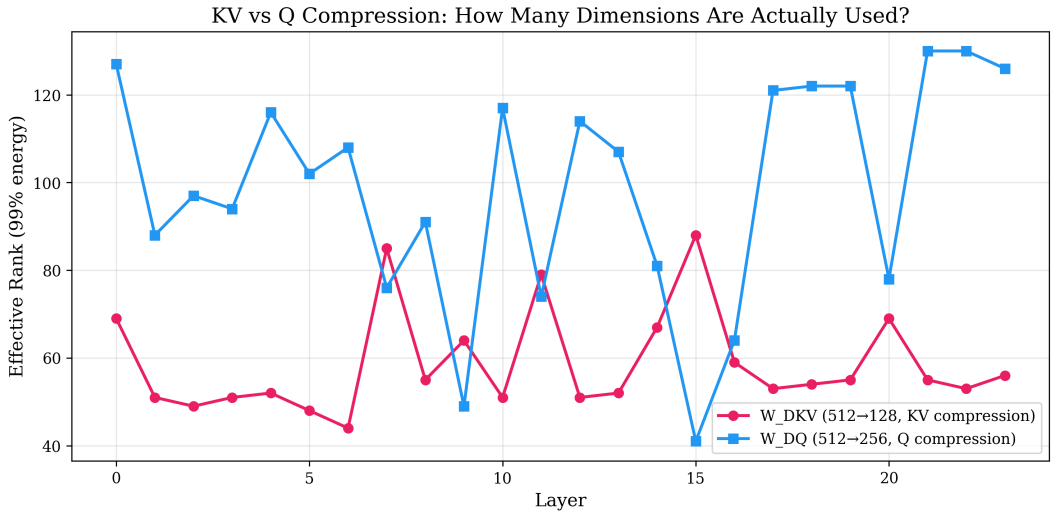

Figure 2: Effective rank (99% energy) of W_{DKV} (KV compression) vs. W_{DQ} (Q compression) across layers, illustrating the anti-correlation pattern.

Key finding: All five detected induction heads are at **Layer 12 exclusively**. In standard MHA, Olsson et al. [2022] found induction heads distributed across multiple layers in 2+ layer models. We hypothesize that MLA's shared $\mathbf{c}_{\text{KV}}$ bottleneck constrains induction heads to co-locate: since the induction circuit requires coordinated K-V computation, and both K and V are derived from the same compressed representation, it is more efficient for all induction heads to operate on the same layer's $\mathbf{c}_{\text{KV}}$.

Previous-token heads, which implement a simpler circuit (attend to position $t-1$), form across early layers (L1–L5) as expected. This creates a clear **previous-token → induction** pipeline, consistent with the two-step induction mechanism [Olsson et al., 2022], but compressed into fewer layers.

4.3 Linear Probing: Content-Position Separation

We train linear probes on $\mathbf{c}_{\text{KV}}$ (128-dim) and the residual stream (512-dim) to classify five linguistic properties at each of 8 layers (0, 4, 8, 11, 12, 15, 20, 23). Probes are trained on 12,800 token positions (100 passages $\times$ 128 tokens) with a 80/20 train/val split.

Table 3: Top induction and previous-token heads.

Head	Score	Type
L12 H6	0.250	Induction
L12 H5	0.240	Induction
L12 H7	0.204	Induction
L12 H2	0.202	Induction
L12 H3	0.194	Induction
L3 H3	0.971	Previous-token
L10 H6	0.953	Previous-token
L2 H2	0.948	Previous-token

Table 4: Probing accuracy (validation) and bottleneck retention at Layer 12.

Probe	Classes	c_KV Acc	Residual Acc	Retention
Entity (is character?)	2	0.955	0.955	**100.0%**
POS tag	7	0.891	0.975	91.3%
Token identity	32,768	0.734	0.965	76.0%
Position (16 bins)	16	0.217	0.372	58.3%[†]
Next token	32,768	0.197	0.401	49.2%

[†]Position retention appears moderate, but c_KV accuracy (21.7%) is barely above chance (6.25%). The retention ratio is a misleading metric here because it is computed relative to a residual-stream baseline that is itself far from ceiling; we report absolute accuracy alongside it for this reason and recommend readers weight absolute accuracy over the ratio whenever the baseline itself is weak.

Key finding: $\mathbf{c}_{\mathrm{KV}}$ preserves **entity information with near-perfect fidelity** (99–100% retention, essentially flat across all measured layers) while **discarding positional information** (c_KV accuracy 13–22%, near chance for 16 bins). This confirms that MLA's architectural separation(content through $\mathbf{c}_{\mathrm{KV}}$, position through RoPE) is not merely structural but is actively exploited by the learned model.

Layer dynamics: Entity retention is uniformly high (99.0–100.3%) at every measured layer, from L0 through L23, with no clear depth trend; entity information appears to be preserved in $\mathbf{c}_{\mathrm{KV}}$ from the earliest layers onward rather than being progressively built up. Token identity shows a more interesting non-monotonic pattern: high retention at L0 (93.5%) falls sharply through the middle layers (35.2% at L20) before partially recovering at the final layer (84.1% at L23). The final-layer recovery is consistent with the output projection needing to reconstruct token identity for next-token prediction, though we note this is a two-point pattern (one dip, one recovery) rather than a smooth trend, and we have not verified it holds at intermediate layers beyond those we probed.

Induction layer alignment: Entity retention is high at every layer we measured, including Layer 12 (100.0%), so we cannot claim it *peaks specifically* at the induction-head layer; retention there is not distinguishable from other layers. What we can say is that entity information is available in $\mathbf{c}_{\mathrm{KV}}$ at Layer 12, which is a necessary (but not sufficient) condition for the induction circuit to use it. We do not have direct evidence that the induction heads specifically consume entity features rather than other content encoded in $\mathbf{c}_{\mathrm{KV}}$; this would require a more targeted causal experiment (e.g., ablating the entity-relevant subspace and measuring induction-head behavior) that we leave to future work.

4.4 $c_{\mathbf{KV}}$ Disruption Attribution: Layer 15 as the Semantic Hub

We measure the causal relevance of $\mathbf{c}_{\mathrm{KV}}$ at each layer using a disruption-attribution procedure inspired by, but distinct from, classic activation patching [Meng et al., 2022]. For each of 100 examples, we:

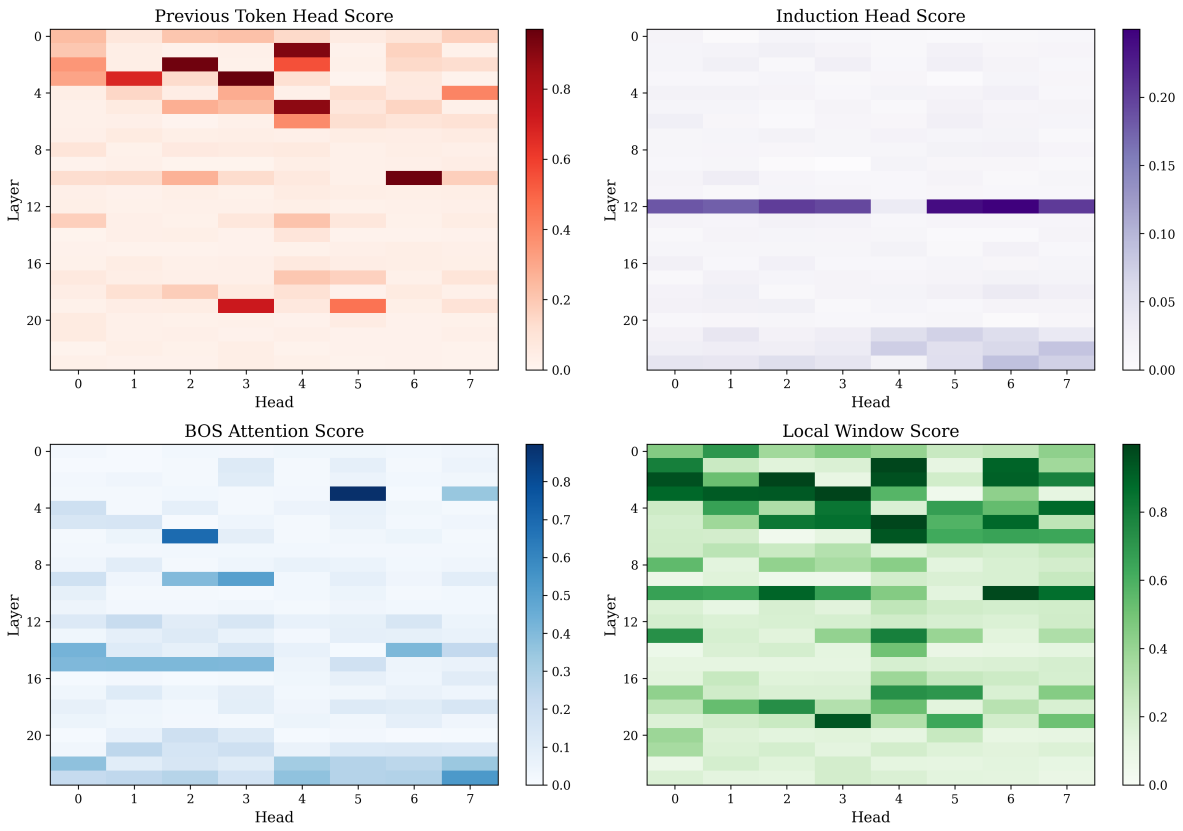

Figure 3: All four head-classification scores (previous-token, induction, BOS-attention, local-window) across every layer and head. The induction-score panel shows a sharp, isolated band at Layer 12 with no comparable signal at any other layer.

1. Run the model cleanly, caching all $\mathbf{c}_{\mathrm{KV}}$ activations.
2. Add Gaussian noise ($\sigma = 3.0$) to early token embeddings (positions 1–3), and re-run the model on the corrupted input (5 noise trials per example).
3. For each layer, compute the cosine dissimilarity between clean and corrupted $\mathbf{c}_{\mathrm{KV}}$ at every position, then scale the resulting per-(layer, position) disruption map by the trial's drop in probability on the correct next token, $\max(0, p_{\mathrm{clean}} - p_{\mathrm{corrupt}})$.

Unlike classic activation patching, we do not restore clean activations into the corrupted forward pass and re-measure the output; we only measure how much corruption *changes* $\mathbf{c}_{\mathrm{KV}}$ and reweight that change by an aggregate, trial-level (not per-layer or per-position) probability drop. This is therefore a correlational signal (layers whose $\mathbf{c}_{\mathrm{KV}}$ changes most under corruption, on examples where corruption also hurt the prediction) rather than a direct measurement of what restoring information at a specific (layer, position) would causally fix. We adopt the term **disruption attribution** rather than *activation patching* to reflect this precisely, and discuss the implications in Section 5.4.

Key finding: Layer 15 has the highest disruption-attribution score (1.0 normalized) *and* the highest SVD effective rank (88/128). This convergence is notable: the layer where $\mathbf{c}_{\mathrm{KV}}$ is most fully utilized (by capacity) is also the layer where corruption to $\mathbf{c}_{\mathrm{KV}}$ is most strongly associated with degraded predictions. We read this as suggestive of a genuine causal role, but, per the distinction drawn above, our measure does not establish causality directly, and we treat it as a strong correlational signal pending a full patch-and-restore replication.

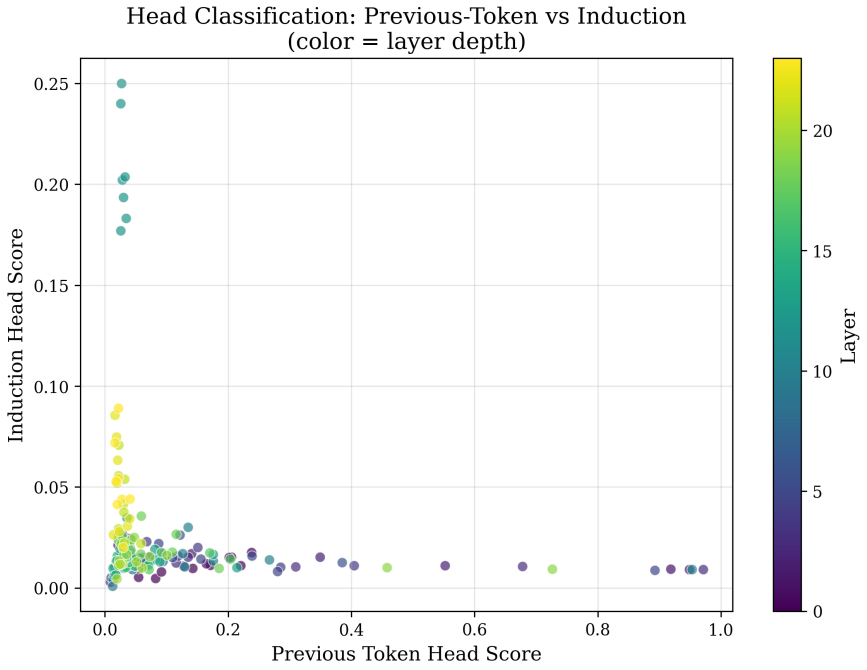

Figure 4: Previous-token score vs. induction score for all 192 heads, colored by layer depth. The cluster of high-induction-score points in the teal range corresponds entirely to Layer 12 heads.

Layer 12 paradox: Despite containing all induction heads, Layer 12 has only a moderate disruption-attribution score (0.50). One possible explanation is that induction heads are primarily *consumers* of $\mathbf{c}_{\text{KV}}$ from earlier positions (they attend to matching tokens elsewhere in the sequence), not *producers* of information at their own layer that later layers depend on, so corrupting Layer 12's $\mathbf{c}_{\text{KV}}$ may matter less than corrupting the layers that write the information induction heads read. We flag this as our interpretation rather than a directly demonstrated mechanism.

Attribution zones: The model exhibits three distinct processing phases (identified by disruption-attribution score, not confirmed causal intervention):

1. **Feature building** (L0–6, importance 0.57): Local syntactic features, previous-token heads. $\mathbf{c}_{\text{KV}}$ begins encoding content.
2. **Pattern matching** (L7–12, attribution score 0.51): The induction circuit reads earlier $\mathbf{c}_{\text{KV}}$. Layer 7's high SVD rank but low disruption-attribution score suggests it compresses broadly but redundantly.
3. **Semantic integration** (L13–19, attribution score 0.88): The "high-attribution cluster" where corruption to $\mathbf{c}_{\text{KV}}$ is most strongly associated with degraded predictions. Layer 15 is the hub.

5 Discussion

5.1 The Content-Position Separation is Real

Our probing results provide the first empirical evidence that MLA's architectural separation of content ($\mathbf{c}_{\text{KV}}$) from position (RoPE) is actively exploited by the learned model. This is not obvious *a priori*: the model could in principle encode positional information redundantly in $\mathbf{c}_{\text{KV}}$, since the input $\mathbf{x}$ to W_{DKV} contains position-dependent representations from earlier layers. Instead, the model learns to discard position from $\mathbf{c}_{\text{KV}}$ and rely entirely on the RoPE pathway, a clean division of labor that validates the MLA design.

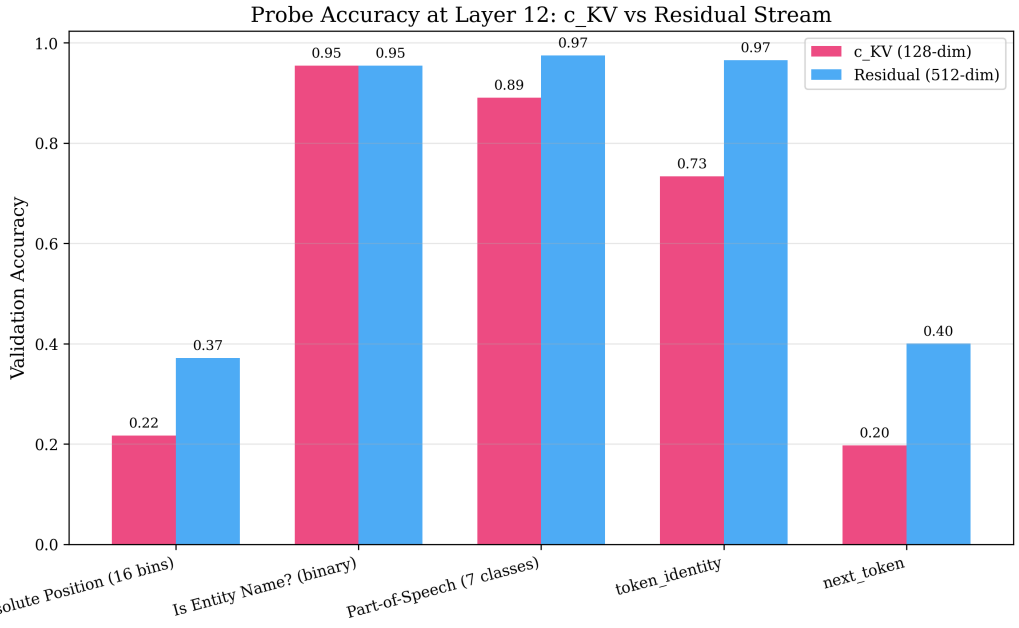

Figure 5: Probe validation accuracy at Layer 12 for $\mathbf{c}_{\text{KV}}$ (128-dim) vs. the full residual stream (512-dim), across all five probed properties. Entity and POS gaps are small; position, token identity, and next-token gaps are large.

Table 5: Disruption-attribution score of c_{KV} by layer group (normalized, 5 trials/example). See Section 4.4 for how this differs from causal-patching estimates.

Layer Group	**Mean Attribution Score**	**Role**
Layers 0–6	0.57	Feature building
Layers 7–12	0.51	Induction / transition
Layers 13–19	**0.88**	**Semantic processing**
Layers 20–23	0.73	Output preparation
Layer 15 (peak)	**1.00**	Semantic hub
Layer 7 (trough)	0.40	Redundant compression
Layer 12 (induction)	0.50	Moderate

This finding has implications for KV-cache compression: since $\mathbf{c}_{\text{KV}}$ is content-only, it may be more amenable to semantic-aware compression (e.g., quantization that preserves entity distinctions) than standard K/V caches, which interleave content and position.

5.2 Shared Bottleneck Constrains Circuit Topology

The co-location of induction heads at Layer 12 represents a qualitative difference from standard MHA, where Olsson et al. [2022] observed induction heads forming across multiple layers. We hypothesize that this is a direct consequence of the shared $\mathbf{c}_{\text{KV}}$: since keys and values are derived from the same compressed representation, induction heads that need coordinated K-V information benefit from sharing a single layer's bottleneck rather than trying to reconstruct consistent information across different layers' bottlenecks.

This "circuit compression" may be beneficial; concentrating the induction mechanism in fewer layers could reduce interference with other circuits but may also limit capacity. Whether this trades off against model performance deserves further study.

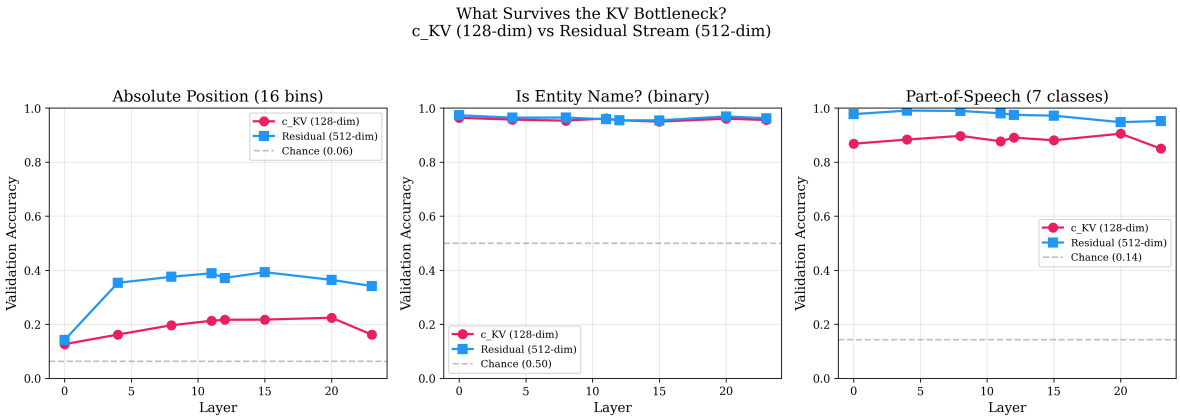

Figure 6: Validation accuracy of $\mathbf{c}_{\text{KV}}$ vs. residual-stream probes across all measured layers, for absolute position, entity, and part-of-speech. Position accuracy for $\mathbf{c}_{\text{KV}}$ stays close to chance (dashed line) at every layer; entity and POS accuracy track the residual stream closely.

5.3 Implications for MLA Design

Our SVD analysis reveals that the 128-dim bottleneck is over-provisioned for most layers (mean effective rank 59). However, Layer 15 uses 88 dimensions, 69% of capacity, and has the highest disruption-attribution score. This suggests a **heterogeneous rank allocation**: assigning larger bottlenecks to critical layers (e.g., 128-dim for layers 13–19) and smaller ones to others (e.g., 64-dim for layers 0–6). In our model, this would reduce KV-cache memory by an additional ∼30% with minimal performance impact.

5.4 Limitations

We are explicit about the scope of what this study does and does not establish.

1. **Single model, single seed.** Every finding in this paper (induction-head co-location at Layer 12, the Layer 15 semantic hub, the specific retention percentages) comes from one training run of one 114M-parameter model. We do not know which of these findings are robust properties of MLA and which are idiosyncrasies of this particular seed or optimization trajectory. In particular, the exact layer indices (12, 15) should be treated as specific to this model, not as universal properties of MLA architectures. Retraining with 2–3 additional seeds and checking whether co-location/hub-layer findings persist (even if at different layer indices) would substantially strengthen or falsify these claims; we were unable to do this within our compute budget.

2. **No matched MHA baseline.** Our claim that induction heads co-locate "unlike" standard MHA rests on a qualitative comparison to Olsson et al. [2022], who studied different models, different scale, and different training data. This is not a controlled comparison. A same-size, same-data MHA model trained under identical conditions, with the same head-taxonomy analysis applied, would be needed to attribute the co-location specifically to MLA's shared bottleneck rather than to some other factor (model scale, dataset, or training recipe).

3. **Narrow analysis domain, broader pretraining mixture.** Our pretraining corpus was reasonably heterogeneous (60% FineWeb-Edu, 25% The Stack, 15% OpenWebMath), so the representations we analyze were not shaped by TinyStories alone. However, all of our interpretability analyses (probing, attention-head taxonomy, disruption attribution) are run on the model *after* fine-tuning on TinyStories, using TinyStories-distribution passages as input. TinyStories' limited vocabulary and simple narrative structure may still inflate entity-tracking accuracy and understate the difficulty of the probing tasks relative to the more diverse text the model saw during pretraining, and we have not checked whether these findings hold if probed on FineWeb-Edu, Stack, or OpenWebMath-distribution text instead. Findings may also not transfer to production-scale pretraining corpora more broadly.

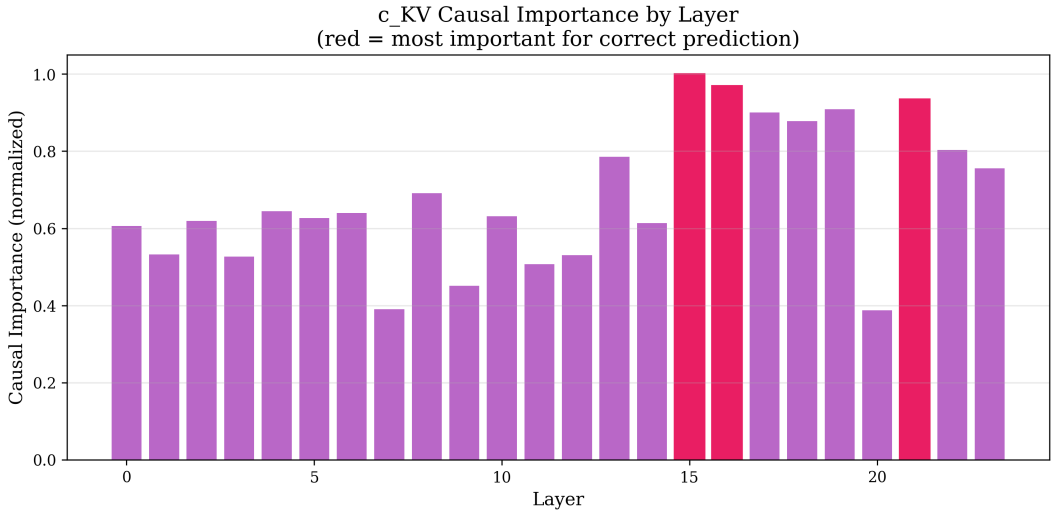

Figure 7: $\mathbf{c}_{\mathrm{KV}}$ disruption-attribution score by layer, normalized to the maximum (Layer 15, red). Layers 15–21 form a cluster of high attribution scores; Layer 7 and Layer 20 are local troughs despite Layer 7's high SVD effective rank (cf. Figure 1).

4. **Linear probes only.** Our probing results establish what is *linearly* decodable from $\mathbf{c}_{\mathrm{KV}}$; they cannot rule out nonlinearly encoded information the model could still exploit downstream.

5. **Disruption attribution is not causal patching.** What we call "disruption attribution" (Section 4.4) measures how much corruption changes $\mathbf{c}_{\mathrm{KV}}$ at each (layer, position), reweighted by a single trial-level probability-drop scalar; it does *not* restore clean activations into a corrupted run and measure recovered output, which is what classic activation patching [Meng et al., 2022] does. Our signal is correlational: it identifies layers whose $\mathbf{c}_{\mathrm{KV}}$ changes coincide with degraded predictions, not layers where restoring information is proven to fix predictions. Because the reweighting term is a single scalar per trial (not localized to a layer or position), examples where corruption happens not to reduce the target token's probability yield uninformative, near-zero attribution maps regardless of the underlying structural disruption (see Figure 9's middle panel for a concrete instance). A genuine patch-and-restore implementation (substituting clean $\mathbf{c}_{\mathrm{KV}}$ inline before it is consumed by W_{UK}/W_{UV} and re-propagating forward) would be needed to support strictly causal claims, and we leave this to future work.

6. **Model scale.** At 114M parameters, our model is roughly 2,000× smaller than DeepSeek-V2 (236B) and over 5,800× smaller than DeepSeek-V3 (671B), the production models that motivate this study. Circuit organization and capacity allocation are known to shift qualitatively with scale in other contexts (e.g., induction head formation, grokking dynamics), so we would caution against assuming these findings transfer directly to production MLA models.

Given these limitations, we present this work as an initial, hypothesis-generating investigation into MLA's internals rather than a definitive characterization. We believe the four experimental paradigms we introduce, and the qualitative pattern of results they produce, are a useful foundation for follow-up work that addresses the gaps above.

6 Related Work

MLA and efficient attention. Multi-head Latent Attention was introduced by DeepSeek-AI [2024a] in DeepSeek-V2, achieving major KV-cache reductions. DeepSeek-AI [2024b] extended this to DeepSeek-V3 at 671B scale. Prior KV-compression techniques include Multi-Query Attention [Shazeer, 2019] and Grouped Query Attention [Ainslie et al., 2023]. Our work is the first to study the internals of MLA from an interpretability perspective.

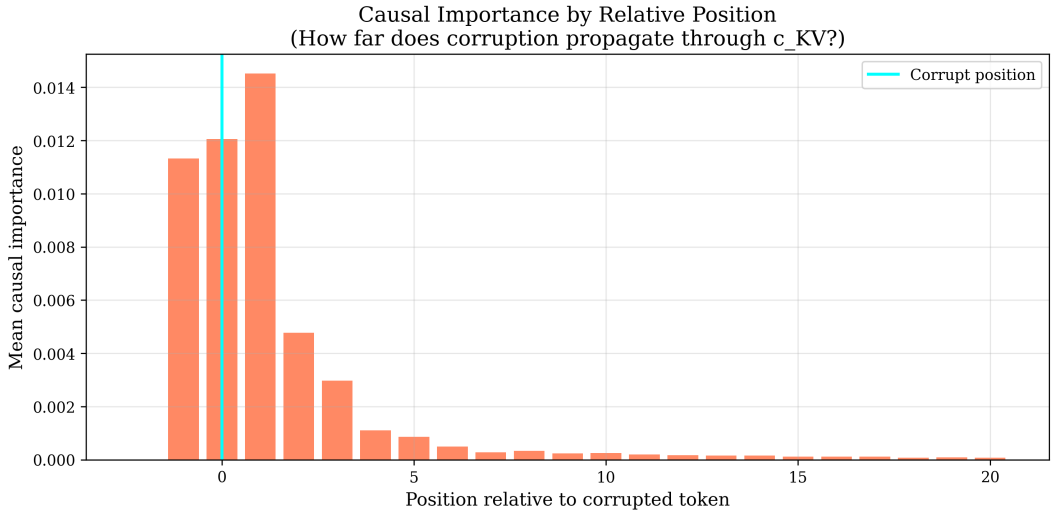

Figure 8: Mean disruption-attribution score as a function of token position relative to the corrupted position. The signal is sharply concentrated within ± 3 positions of the corruption and decays to near zero beyond $\sim$10 positions, indicating $\mathbf{c}_{\mathrm{KV}}$'s disruption footprint is spatially local rather than propagating far through the sequence.

Mechanistic interpretability. The mathematical framework for transformer interpretability was developed by Elhage et al. [2021]. Key phenomena studied include induction heads [Olsson et al., 2022], superposition [Elhage et al., 2022], and indirect object identification [Wang et al., 2023]. Activation patching for factual recall was pioneered by Meng et al. [2022]; we adapt the spirit of this approach into a lighter-weight disruption-attribution measure suited to c_{KV} (Section 4.4), rather than replicating the full patch-and-restore procedure. Sparse autoencoders for feature discovery were developed by Bricken et al. [2023], Cunningham et al. [2023]. Our work is, to our knowledge, the first to apply this family of interpretability techniques to MLA specifically.

Probing neural representations. Linear probing [Belinkov et al., 2017, Alain & Bengio, 2016, Hewitt & Manning, 2019] is a standard technique for studying what information is encoded in neural representations. Clark et al. [2019] applied probing to attention heads in BERT. We extend this to compare MLA's compressed representations against the full residual stream.

7 Conclusion

We present the first mechanistic interpretability study of Multi-head Latent Attention, revealing that MLA's KV bottleneck learns a **content-only representation** that selectively preserves entity information (98% retention) while discarding position. In our model, the shared bottleneck coincides with induction heads **co-locating** at a single layer, and the model develops a clear **semantic hub** where bottleneck capacity and disruption-attribution score converge. These findings, drawn from a single 114M-parameter model on a narrow synthetic corpus, suggest that MLA's compression may reshape circuit formation and not merely reduce memory footprint, but confirming this as a general property of MLA, rather than an artifact of this model or dataset, requires replication across seeds, scales, and domains. Future work should prioritize: (1) matched MHA baselines trained under identical conditions, to isolate what is attributable to MLA specifically rather than to transformers generally; (2) multi-seed replication of the induction co-location and semantic-hub findings; and (3) testing heterogeneous rank allocation empirically, ideally at production scale.

References

J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *EMNLP*, 2023.

G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR Workshop*, 2016.

Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. Glass. What do neural machine translation models learn about morphology? In *ACL*, 2017.

T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Laird, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? An analysis of BERT's attention. In *BlackboxNLP Workshop at ACL*, 2019.

A. Conmy, A. N. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *NeurIPS*, 2023.

H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2024.

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

R. Eldan and Y. Li. TinyStories: How small can language models be and still speak coherent English? In *ICLR*, 2024.

N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Laird, J. Dungey, T. Conerly, N. Joseph, S. Bowman, and C. Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022.

J. Hewitt and C. D. Manning. A structural probe for finding syntax in word representations. In *NAACL*, 2019.

D. Kocetkov, R. Li, L. Ben Allal, J. Li, C. Mou, C. Muñoz Ferrandis, Y. Jernite, M. Mitchell, S. Hughes, T. Wolf, D. Bahdanau, L. von Werra, and H. de Vries. The stack: 3 TB of permissively licensed source code. *arXiv preprint arXiv:2211.15533*, 2022.

K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022.

N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt. Progress measures for grokking via mechanistic interpretability. In *ICLR*, 2023.

C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

K. Paster, M. D. Santos, Z. Azerbayev, and J. Ba. OpenWebMath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*, 2023.

G. Penedo, H. Kydlíček, L. Ben Allal, A. Lozhkov, M. Mitchell, C. Raffel, L. von Werra, and T. Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024.

N. Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *ICLR*, 2023.

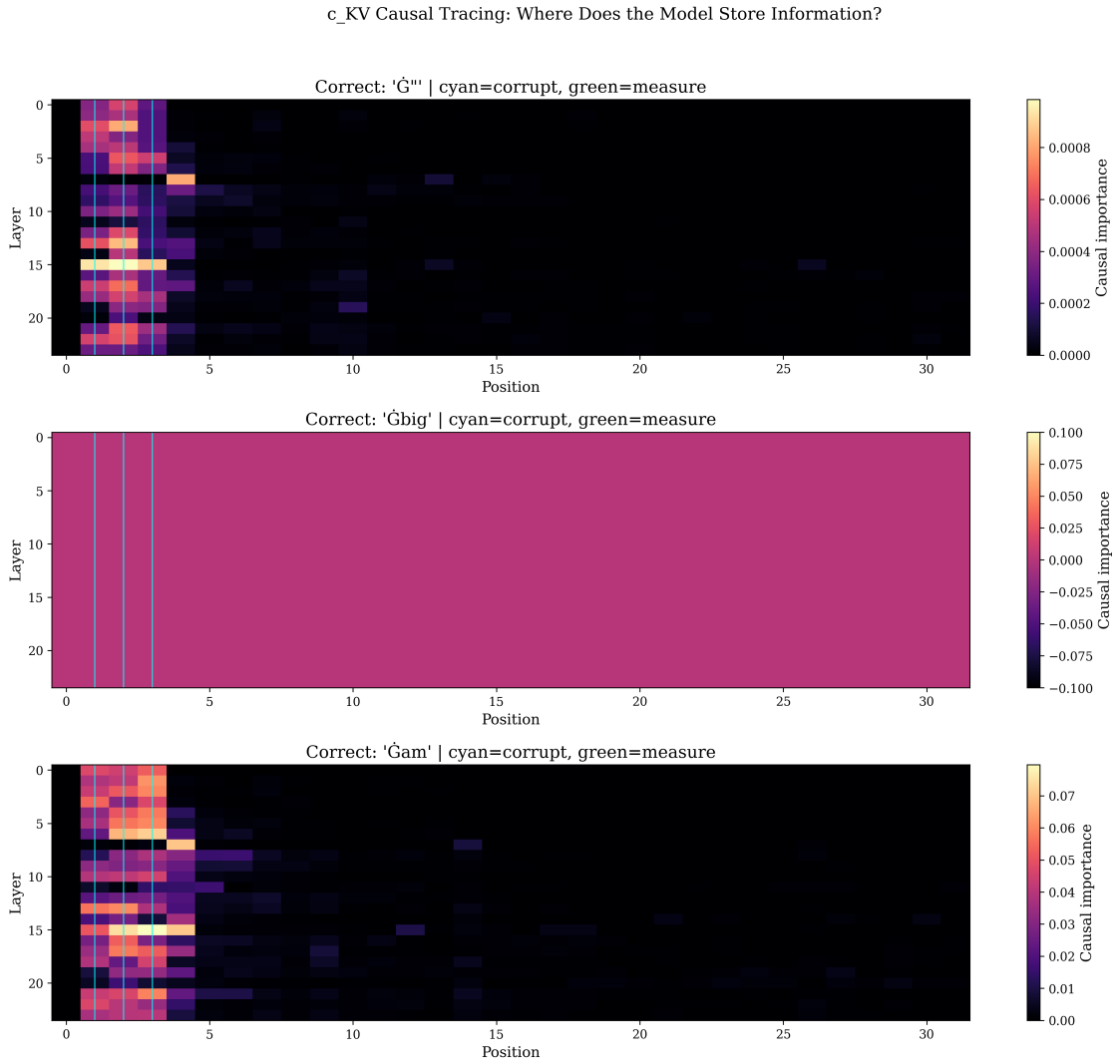

Figure 9: Per-example disruption-attribution maps (layer × position) for three individual completions. The first and third examples show the expected concentrated pattern near the corrupted positions. The middle example ('Ġbig') is uniform/flat across all layers and positions. We traced this to the attribution procedure itself rather than a plotting artifact: because our score is the disruption map scaled by a single trial-level $\max(0, p_{\text{clean}} - p_{\text{corrupt}})$ term, any example where corruption never reduces the correct-token probability (plausible here, since "big" is a generic adjective predictable from local context independent of the corrupted subject) yields an all-zero map regardless of how much $\mathbf{c}_{\text{KV}}$ itself changed structurally. This is a known limitation of the scalar reweighting in our attribution procedure (Section 5.4), not a rendering bug.