

Source, Relay, and Suppressor Heads in a Poetry Generation Circuit

Stanford CS224N Default Project

Shaoyi Zhang
shaoyi@stanford.edu

Kai Wen
kwen7@stanford.edu

Abstract

We implement GPT-2’s core components (attention, AdamW) and LoRA, compare LoRA and full fine-tuning on sonnet generation, and identify a rhyme-attention circuit via path patching. However, ablating that circuit has little to no effect on rhyme rate, with random head ablation hurting rhyme rate more. Investigating this paradox, we find that the heads in the circuit are just *routing* heads, while the heads that actually affect behavior fall into three functionally distinct classes: a **source** head (L6H10), a relay head (L7H11), and a suppressor head (L11H0). The source head’s ablation degrades rhyme by 38% and cascades to degrade four downstream heads’ outputs by $\sim 72\%$. The **relay** head’s ablation has no behavioral or cascade effect alone but contributes to the cumulative drop when ablated on top of several others. The **suppressor** head’s ablation actually *increases* rhyme rate by 23% because its OV circuit encourages punctuation over content tokens, which goes against rhyme. We characterize the three classes across eight dimensions (QK/OV weights, behavioral effect, cascade structure, residual-stream signature). The residual std ratio α_σ separates the source head from all others by $10\times$ the noise level, but we show it fails on the IOI circuit, where it instead marks Anti-Erasure. Fine-tuning barely updated the source head, and weight patching shows that no identifiable upstream head’s weight change drives its attention amplification. On Default Project tasks, we find that LoRA with MLP adapters *beats* full FT on both tasks at GPT-2 Small. At Medium and Large scales, it trails by < 0.5 pp/chrF. We also perform a study showing that LoRA configs including MLP are much more effective than attention-only configs. This gap widens on single-token tasks (IOI v+o-QK: $-1.26 \rightarrow +3.29$ from Small to XL).

1 Key Information

TA mentor: Chenglei Si **External collaborators:** No **External mentor:** No **Sharing project:** No

2 Introduction

The goal of circuit discovery in LLMs is to map a subset of model components that implement a given behavior [1, 2]. The standard recipe shown in Wang et al. [2]’s study of Indirect Object Identification (IOI), is to find heads that do the best in recovering a corrupted output when patched back in, then verify the circuit by ablating those heads. In practice, however, it’s hard to verify the circuit as ablating circuit heads has little behavioral effect because other components can compensate (the “Hydra effect” [3]). Rushing and Nanda [4] showed that what seems like self-repair is not the model adapting to ablation but actually a result of the model’s architecture (LayerNorm rescaling, destructive interference between components). The practical challenge then becomes identifying which heads actually matter when the ablation is null.

This paper seeks to address that problem. After implementing GPT-2’s attention, AdamW, and LoRA from scratch for the Default Final Project, we fine-tuned our model on Shakespeare’s sonnets

for the sonnet generation task and used path patching to locate a rhyme circuit with four heads (L0H1 $\rightarrow$ L1H11 $\rightarrow$ L4H11 $\rightarrow$ L5H6). Ablating all four of these heads lowered rhyme rate from 7.3% to 7.1% and ablating four *random* heads reduced it even further to 6.1%. Therefore, the circuit was worse than random at explaining behavior.

Our contribution is a taxonomy of heads by **causal structure under intervention**, related but not overlapping prior work’s taxonomy of heads (name mover, duplicate detector, etc.). We identify three classes and characterize these classes across eight dimensions (Table 2). The **source** head L6H10 has OV circuit that writes content. Its ablation cascades downstream (observer-head output norm drops $\sim 72\%$ on average), and is the only head whose single ablation hurts rhyme considerably. The **relay** head L7H11 has a weak copy of the source head’s QK pattern along with positional lookup across lines. Its OV contains little information and its ablation has no downstream effect on behavior. This ablation only matters in the context of other ablations. The **suppressor** head L11H0 has a structure-writing OV circuit that boosts punctuation tokens over content tokens and its ablation *increases* rhyme rate. This is similar to prior work’s negative name movers—both go against the objective. We also find that α_σ , the ratio of residual-stream standard deviation under ablation to baseline (one forward pass, cheap), separates the source head from 13 other tested heads with a gap $10\times$ the noise level. We offer this as a cheap way to identify candidate source heads.

Separately (§6.4), we investigate that after FT, the source head’s weights are among the least modified in the model (3rd percentile), suggesting that it was already functionally sufficient before fine-tuning. A weight-patching experiment also rejects the hypothesis that any identifiable upstream head’s weight change encourages its attention amplification. This mechanism is diffuse, an open question.

3 Related Work

Circuit discovery. The transformer-circuits framework [1] decomposes attention into QK (tokens to read) and OV (what to write from tokens) circuits. Two techniques for circuit discovery, activation patching [5] and path patching [2] (refinement of activation patching), locate circuits by measuring which components’ clean activations best recover a corrupted output. The IOI study [2] remains the most detailed, classifying heads by their *role* in the circuit (name mover, S-inhibition, induction, negative name mover).

Self-repair and redundancy. McGrath et al. [3] found that ablating a head often doesn’t have a direct effect on logits because other heads can compensate (the Hydra effect). According to Rushing and Nanda [4], the main cause of the Hydra effect is LayerNorm rescaling and “Anti-Erasure”: components have destructive interference at baseline, so removing one raises the residual stream’s norm. They also noted that at the token level, self-repair is noisy. Nanda [6] distinguished parallel redundancy (multiple heads do the same job independently) from serial redundancy (a pipeline can reroute around a missing link). Our source/relay distinction is an instance of the parallel/serial split. The source head sits at the origin of a serial chain, while the relay head is one of several parallel readers.

Fine-tuning mechanism. Prakash et al. [7] showed that LoRA FT enhances existing circuits instead of building new ones. Wang et al. [8] found that fine-tuning changes edges more than nodes (which heads attend to which vs. what individual heads compute). Our finding that the source head’s weights are barely modified (3rd percentile) but routing heads change most is consistent with both, though our weight-patching experiment rejects the simple “upstream routing change” explanation for the source head’s attention amplification.

Gap we fill. While prior head taxonomies classify heads using their functions in the circuit, we use *causal structure under intervention*: does ablation cascade, does it have a behavioral effect, does the residual stream grow or shrink. When ablation is used to validate circuits, these distinctions matter because self-repair can obscure the circuit.

4 Approach

4.1 Default Project base and LoRA

We implement GPT-2 [9] multi-head attention, AdamW [10], and LoRA [11] from scratch on the provided starter code. Our LoRA module (`modules/lora.py`) wraps each W_Q and W_V as $h = W_0x + \frac{\alpha}{r}BAx$ with B zero-initialized and A Kaiming Uniform, freezing W_0 . The interpretability infrastructure (i.e. hooks for per-head activation/attention caching, ablation masks, weight patching) is also implemented from scratch.

4.2 Circuit discovery

We adapt activation patching [2, 5] for sonnet generation: we corrupt the rhyme partner’s final word with random tokens and cache clean per-head outputs. Then, for each head (L, H) restore that head’s clean output on the corrupted input and score recovery $(L, H) = (p_{\text{patched}} - p_{\text{corrupted}}) / (p_{\text{clean}} - p_{\text{corrupted}})$. Path patching refines this by patching only the value input *to* a fixed receiver head from each upstream head, allowing us to trace multi-hop pathways.

4.3 Head characterization suite

Our methodology involves making four complementary measurements per head:

(a) Behavioral. Single-head ablation $\rightarrow$ measure rhyme rate on held-out prompts. We use $n=3$ seeds $\times$ best-of-3 sampling; because with only $12 \times 7 = 84$ rhyme-pair checks per evaluation, single-seed rates are noisy ($\pm 2-4\%$).

(b) Cascade. For each (ablated, observer) head pair, measure the observer’s ΔNorm (output norm at rhyme positions minus non-rhyme positions) under ablation versus baseline. A source head’s ablation should cause large $\Delta\Delta\text{Norm}$ on downstream observers and a relay head’s should not.

(c) Residual-stream signature. $\alpha_\sigma = \sigma(r') / \sigma(r)$ is the ratio of residual-stream std under ablation to that under baseline (pre-final-LayerNorm). $\alpha_\sigma > 1$ means the ablated component had destructive interference with the rest of the residual stream (Rushing’s Anti-Erasure [4]). Because the final LayerNorm divides by σ , $\alpha_\sigma > 1$ also means LN *suppresses* instead of amplifying under ablation, so the behavioral effect of a high- α_σ head is not masked by LN rescaling.

(d) QK/OV weight analysis. For QK: aggregate attention weight to each vocabulary token, revealing what the head reads. For OV: project through the unembedding matrix and rank tokens by how much the head boosts their logit, revealing what the head writes.

5 Experiments

5.1 Data, evaluation, and experimental details

We use the DFP-provided datasets: SST (5-class sentiment, 11 855 samples), CFIMDB (binary sentiment, 2 440 samples), Quora paraphrase pairs (283K train / 40K dev), and Shakespearean sonnets (143 train, from which we carve 12 as dev; plus 12 held-out test). Classification and paraphrase use accuracy while sonnet generation uses chrF [12] against held-out gold completions. Our *rhyme rate* metric is the fraction of ABAB CDCD EFEF GG pairs whose line-ending words share a 2–3 character suffix. This metric is crude (e.g. *present/taste* is accepted) so we report relative comparisons.

Hardware: 2 $\times$ H100 80GB. Full fine-tuning: lr $1e-5$, 8–40 epochs. LoRA: lr $1e-4$, $\alpha=16$, 15–20 epochs. AdamW, batch size 8 (sonnets) or 64 (paraphrase). Sonnet chrF is best-of-3 with nucleus sampling [13] (temperature 1.2, top- p 0.9) on the 12 held-out dev prompts. We corrected a starter-code prompt mismatch between the paraphrase test and training datasets (`datasets.py:83`); the mismatch cost 1.4 pp test accuracy before the fix. Sentiment classification (GPT-2 Small): SST dev accuracy 51.1% (full FT) / 47.6% (LoRA r8); CFIMDB 98.4% / 96.3%.

Model	Paraphrase Dev Acc (%)			Sonnet Dev chrF (best-of-3, $n=3$)		
	Full FT	Best LoRA	Gap	Full FT	Best LoRA	Gap
GPT-2 Small	89.06	89.19 ALL r8	-0.13	42.35	42.49 ± 0.20 ALL r8	-0.14
GPT-2 Medium	90.78	90.34 ALL r8	0.44	43.26 ± 0.18	42.88 ± 0.04 ALL r32	0.38
GPT-2 Large	90.74	90.45 ALL r8	0.29	43.23 ± 0.16	43.10 ± 0.32 ALL r8	0.13
Staff baseline		88.2			41.4	
Test leaderboard [†]		90.4			42.43	

Table 1: **Default Project results (sonnet $n=3$; paraphrase single-seed).** The ALL config (LoRA on V , O , and both MLP matrices) *beats* full fine-tuning at Small on both tasks, at $<1\%$ of the parameters, and trails by a shrinking margin at larger scales. Both tasks show the same U-shape: Full-FT headroom is largest at Medium (0.44/0.38) and smallest at Small and Large (0.29/0.13). Full fine-tuning plateaus at ~ 43.25 chrF and $\sim 90.75\%$ acc from Medium on. All LoRA results are $r=8$ except sonnet Medium ($r=32$, Appendix B). Both tasks beat the staff baseline at every scale. [†]Leaderboard is single-sample; best-of-3 adds ~ 1.0 chrF at this sampling variance.

5.2 Default Project results

5.3 The ablation paradox

Path patching identified a four-head pathway $L0H1 \rightarrow L1H11 \rightarrow L4H11 \rightarrow L5H6$ (all in layers 0–5). But ablating all four changed rhyme rate from 7.3% to 7.1%, while ablating four random heads reduced it to 6.1%. Exhaustive activation patching (all 144 heads) then revealed that the top-6 heads by recovery score (L11H8, L11H0, L9H11, L8H0, L6H10, L7H11) all lie in layers 6–11, revealing that the path-patching circuit barely had overlap. These results come from the two methods measuring different things. First, path patching scores each edge by how much restoring it *alone* in a corrupted context helps, so it finds the *narrowest* information channel. On the other hand, activation patching scores each node by how much restoring its *full output* helps, so it finds the widest info channel. The early layers (0–5) form a narrow bottleneck that feeds into wide late computation layers (6–11) and the two methods address different parts. Heads identified by path patching are *routing* heads that are high information flow, parallel-redundant, and not behaviorally necessary, and the heads identified by activation patching are *computation* heads.

When we ablate the six computation heads in order of importance, it (Figure 1) produces a non-monotonic curve. Adding L11H0 at step 2 brings rhyme rate back up (the suppressor effect, reproducible across seeds). Rhyme drops toward zero only by step 6 (L7H11), and only in the context of five prior ablations: ablating L7H11 alone has no effect (§5.4). There is no single-head kill switch.

5.4 Source, Relay, and Suppressor

Table 2 summarizes the full eight-dimension characterization. Three rows are worth highlighting. **Single ablation $\rightarrow$ rhyme:** ablating L6H10 alone drops rhyme 38% (7.1% $\rightarrow$ 4.4%, $n=3$ seeds); ablating L7H11 alone does nothing (7.1% $\rightarrow$ 7.8%); ablating both gives 5.0%, essentially the same as L6H10 alone. The relay head has no standalone behavioral footprint. Note that the behavioral delta (2.7 pp on a $\sim 7\%$ baseline, $n=3$) is the *weakest* evidence in this row: the rhyme metric is crude and the effect is roughly one standard error. The cascade and α_σ rows carry the source/relay distinction. **Cascade structure:** Figure 2 shows that ablating L6H10 degrades four downstream heads’ rhyme-position output norm by $\sim 72\%$ on average and ablating L7H11 leaves all observers within $|\Delta\Delta| \leq 0.10$ of the baseline. The causal structure is asymmetric: upstream $\rightarrow$ downstream effects are large, downstream $\rightarrow$ upstream effects are exactly zero. α_σ : across 14 heads tested (Table 3), L6H10 is the only one with $\alpha_\sigma > 1.05$; the gap to the next-highest (1.040) is $10\times$ the within-class noise (± 0.012). We tested this as a source-head pre-screen on the IOI name-mover heads: it *fails*. Name movers have $\alpha_\sigma < 1$, despite being source heads in the IOI circuit. So α_σ marks Anti-Erasure (destructive interference at baseline), not source-ness. The overlap in our single example is incidental. We retain α_σ as a taxonomy dimension because it cleanly separates L6H10 here, but it is not a transferable source-head detector.

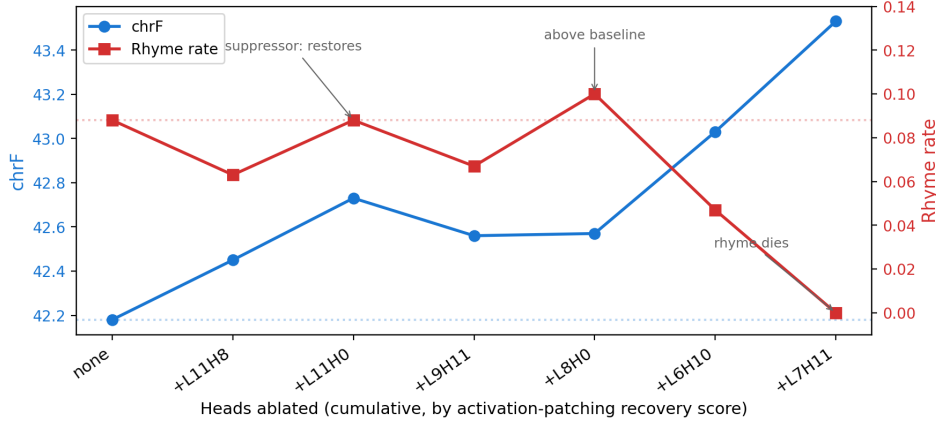


Figure 1: **Cumulative ablation of computation heads** (importance order from activation patching). Rhyme rate (red, right axis) is non-monotonic: adding the suppressor L11H0 at step 2 brings rhyme back up (robust across generation seeds); the curve drops toward zero only by step 6 (L7H11 added on top of five others). chrF (blue, left axis) stays within a 1.4-point range throughout: ablating the rhyme circuit does not hurt general quality.

	L6H10 (Source)	L7H11 (Relay)	L11H0 (Suppressor)
<i>QK: what the head reads</i>			
\n attention weight	247	144	213
other pattern	line-start words (When, O, Lo, But...)	same + cross-line positional (-11, -21, -47)	. (35), line-starts
<i>OV: what the head writes (boost per token class)</i>			
content (thee, love, heart)	28, 18, 14	15, 10, 11	110, 14, 12
structure (\n, ., ,)	26, 26, 29	18, 19, 21	141, 156, 179
summary	content-leaning	weak both	structure-heavy
<i>Causal structure under intervention</i>			
Single ablation → rhyme ($n=3$ seeds)	7.1 → 4.4% (-38%)	7.1 → 7.8% (null)	7.3 → 8.9% (+23%)
Cascades downstream? (observer F_1)	Yes (0.58-0.70)	No (≈ 1.0)	—
Depends on upstream? (L6H10 ablated)	—	Yes (-0.46)	—
α_σ (residual std ratio)	1.154	1.016	1.006
<i>Fine-tuning signature</i>			
Weight change (OV, percentile)	0.74% (3rd)	0.85% (~ 10 th)	1.51% (~ 75 th)

Table 2: **The three head classes**, characterized across eight dimensions. The source head writes content, its ablation cascades downstream and is the only one with behavioral effect, its α_σ is the only one above 1.05, and it was the least modified by fine-tuning. The relay head has a weak copy of the source’s QK along with a positional lookup across lines, but has low-information OV and no standalone causal effect (see Figure 1 for its role in cumulative context). The suppressor encourages punctuation, so ablating it actually *helps* rhyme.

5.5 Memorization and generalization

Prompting *pretrained* GPT-2 with the first three lines of held-out sonnets already yields chrF ≈ 41 , and pretrained Medium reproduces 24.7% of gold lines exactly [14]. Fine-tuning adds only ~ 1.5 chrF but drops exact line matches (17.6% $\rightarrow$ 13.1% for Small) while 8-gram overlap rises (11.1% $\rightarrow$ 15.9%): fine-tuning teaches the model Shakespearean style rather than line memorization. LoRA’s 8-gram overlap is lower than full fine-tuning’s at matching chrF (13.8% vs. 15.9%), consistent with implicit regularization though the effect is small.

To test generalization beyond Shakespeare we evaluate on 10 Petrarchan sonnets by Elizabeth Barrett Browning (different rhyme scheme: ABBA ABBA CDC DCD). The model achieves chrF 42.22, only 1.23 points below its Shakespeare score, and adapts its rhyme pattern to match the Petrarchan scheme in the prompt. A per-layer binary classifier (training vs. held-out completion) reaches 80%

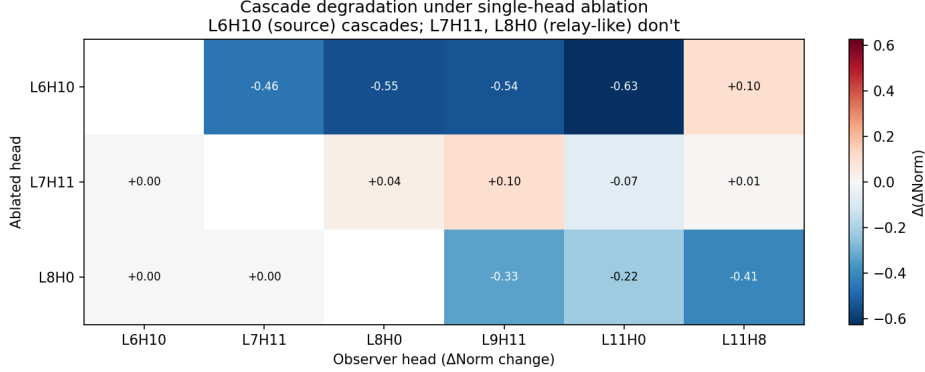


Figure 2: **Cascade structure distinguishes source from relay.** Each cell is the $\Delta\Delta\text{Norm}$ (change, under ablation of the row head, in the column head’s rhyme-position output norm minus non-rhyme-position output norm). Ablating L6H10 degrades four downstream observers by 0.46–0.63; ablating L7H11 leaves all observers essentially unchanged ($|\Delta\Delta| \leq 0.10$). L8H0 is intermediate, degrading only late-layer observers. Upstream observers of any ablation (bottom-left triangle) show exactly zero change, as expected.

Class	α_σ (mean $\pm$ std)	N
Source (L6H10)	1.154	1
Relay-like (L7H11, L8H0)	1.028 ± 0.012	2
Routing (L0H1, L1H11, L4H11, L5H6)	0.991 ± 0.008	4
Random controls (L2H3, L3H7, L5H2, L10H5)	0.992 ± 0.012	4

Table 3: α_σ **separates the source head from all others.** Gap = 0.126, within-class noise $\approx \pm 0.012$, a $10\times$ separation. But with only one source head, this is an observation and hypothesis, not a validated classifier. Why $\alpha_\sigma > 1$ for a source head: it writes novel task-specific content that partially opposes what is already in the residual stream; removing it removes that cancellation and the residual norm grows (Anti-Erasure [4]).

accuracy at the final layer, with an 8pp gap between vs-training and vs-held-out discrimination; the model’s representations of its own generations are closer to held-out than to training sonnets, which is the desired direction.

6 Analysis

6.1 Why the source cascades but the relay does not

The QK/OV weight analysis (Table 2) gives an explanation. L6H10’s QK mostly attends to `\n` (weight 247) and capitalized words that typically open lines (When, O, Lo, But, Then, So, For), showing that it reads where lines begin. Its OV writes content tokens (`thee`=28, `love`=18, `bright`=17) more strongly than punctuation (`\n`=26, `.`=26). It is a *line-start aggregator* that reads line structures and write content candidates. Downstream heads depend on that content as a result, so removing it causes their inputs to change substantially.

L7H11’s QK has the same newline (`\n`)-attending pattern but decently weaker (144 vs. 247). It also has a cross-line positional component (its top attention offsets at -11 , -21 , -47 tokens, roughly one per line back). Its OV values are roughly half of L6H10’s across the board. We see it as a weak copy of the source’s read pattern with an added positional lookup and a write with little info. Downstream heads do not depend on its output because there is little information in it. This explains why its ablation does not cascade. On the other hand, it *does* depend on upstream as its own ΔNorm drops 0.46 when L6H10 is ablated (Figure 2), because its input comes partly from the source.

The “both” ablation (5.0%) $\approx$ “L6H10 alone” (4.4%) result confirms that the relay head has no standalone behavioral footprint.

6.2 L1H0 as a negative head for poetry structure

L1H0’s QK reads line boundaries ($\backslash n=213$, $\cdot=35$, plus line-start words), while its OV emphasizes punctuation ($\backslash n=141$, $\cdot=156$, $\cdot=179$) over content tokens (love=14, heart=12). It pushes probability mass toward ending the current line rather than extending it to satisfy a rhyme constraint. This competes with rhyme and moves in the negative direction of the objective, so ablating it understandably increases rhyme rate (7.3%→8.9%). The same competition explains the non-monotonicity in Figure 1. This is very similar to the IOI study’s Negative Name Movers [2], which write *against* the correct answer. The difference is that here the negative contribution is structural in the form of punctuation rather than semantic.

6.3 Behavioral null $\neq$ mechanistic null

Ablating L6H10 degrades downstream output norm by $\sim 72\%$ on average (Figure 2) but reduces rhyme rate by only 38%. The gap is caused by apparent self-repair [3]. We find the Anti-Erasure signature of Rushing and Nanda [4]: $\alpha_\sigma = 1.154$ under L6H10 ablation, and $\alpha_\sigma \in [1.02, 1.16]$ across all three heads we ablated. The residual stream *grows* under ablation, indicating the ablated head’s contribution had destructive interference with the rest of the stream at baseline.

A first-order Taylor expansion of the direct-logit-effect change under ablation decomposes approximately into three static factors (magnitude cascade ≈ 0.81 , direction realignment ≈ 1.20 , LN rescaling $= 1/\alpha_\sigma \approx 0.86$) whose product tracks the observed aggregate change to within 0.4%. However the per-position fit is weak ($R^2 = 0.09$), so we report this only as evidence that static architectural effects are *sufficient* to account for the aggregate self-repair signal without invoking adaptive computation, consistent with Rushing and Nanda [4].

The ablations collapsing cumulatively (Figure 1) is consistent with this picture as magnitude cascade compounds across ablations while the other two factors saturate (interference can only be removed once), so eventually the product drops below the behavioral threshold.

6.4 What fine-tuning changed: observation and a negative test

Observation. L6H10’s weight change is in the 3rd percentile model-wide: OV changed by 0.74% versus a median of 1.15% and maximum of 2.05% (at L4H11). Before fine-tuning, L6H10 was already $\backslash n$ -attending (the strongest $\backslash n$ -attender in the model, per-position weight 0.37) and its OV was already content-writing (three=28.3, identical to post-FT). Fine-tuning sharpened its attention to $\backslash n$ by $1.49\times$ (per-position $0.37 \rightarrow 0.55$) but barely modified OV. The most-modified heads (L4H11 at 2.05%, L5H6 at 1.74%) are the routing heads from the original path-patching circuit.

Hypothesis. This suggested that fine-tuning modifies upstream routing for the goal of sending better input *to* the source head, rather than modifying the source head itself. This is consistent with the “edges change more than nodes” finding of Wang et al. [8].

Test. In the post-fine-tuning model, we revert individual heads’ weights back to their pretrained values and measure L6H10’s attention to $\backslash n$. Reverting L4H11 (model-wide maximum weight change) changes L6H10’s attention by 1%; reverting L5H6 by 0%; reverting the top-5 highest-change heads together by 8%; reverting L6H10’s own weights by 8%. The control (reverting L2H3, a low-change head) gives 2%. All effects are in noise range.

Conclusion. We reject the hypothesis. The $1.49\times$ attention amplification does not localize to any small set of head-weight changes. It comes from the accumulated small drift across the entire model, or from LayerNorm/embedding changes, or from compounded upstream representation shifts. Despite the hypothesis rejection, we’re able to rule out the simplest explanation. The resulting insight is that L6H10 was already suitable pre-fine-tuning, so its gradients were small, rather than the original direction that “fine-tuning protected the source head.”

7 Conclusion

We propose a taxonomy of attention heads by causal structure under intervention, orthogonal to function-in-circuit taxonomies. In GPT-2 fine-tuned on Shakespeare, we identify a rhyme-generation circuit and a source head (cascades, behavioral effect, high α_σ , content-writing OV,

barely updated by fine-tuning), a relay head (depends on upstream, no cascade, no behavioral footprint, weak OV), and a suppressor head (ablation = better rhyme rate, punctuation-writing OV). The distinction explains why path-patching circuits can be behaviorally irrelevant via finding routing heads and why single-ablation effects are often null. This is because relay heads may seem the most important, but the source head is what really matters.

As a secondary contribution, we show (Appendix B) that LoRA with MLP adapters matches or beats full fine-tuning at GPT-2 Small on both Default Project tasks, with a consistent U-shaped gap across scale (i.e. Full-FT’s headroom peaks at Medium and vanishes at Small). This implies that the standard $Q+V$ LoRA leaves ~ 2 pp on the table at Small, but adding the MLP adapter recovers it.

Limitations. The taxonomy is one task, one model, and one source head. The α_σ classifier has $N(\text{source}) = 1$. The fine-tuning mechanism (why attention amplified $1.49\times$) remains open. The LoRA module study (Appendix B) is on synthetic and small-data tasks. Whether the ALL>QK gap continues to widen beyond 1.5B parameters, and whether it holds on instruction-following, is untested.

Future work. A GPT-2 without LayerNorm [15] would let us isolate the LN-rescaling factor. GIM [16] identifies softmax redistribution as another self-repair mechanism, which our decomposition does not capture. The epoch-0-saturation artifact (saving the best checkpoint early) for IOI suggests re-running with evals at each epoch to determine whether QK’s late-training recovery transfers to held-out data.

Team contributions

Both authors contributed to implementation, experiments, and writing. **Shaoyi Zhang:** LoRA module and module-targeting study, interpretability infrastructure (path/activation patching, ablation masks, hooks), source/relay/suppressor taxonomy and characterization (QK/OV, cascade, α_σ), weight-patching experiments, rhyme-constrained decoding. **Kai Wen:** GPT-2 attention and AdamW, full fine-tuning with hyperparameter search, sonnet training extensions, paraphrase detection task, dual-GPU experiment pipelining and multi-seed orchestration, memorization and generalization experiments, result aggregation and visualization, poster.

References

- [1] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.
- [2] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023.
- [3] Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.
- [4] Cody Rushing and Neel Nanda. Explorations of self-repair in language models. In *International Conference on Machine Learning*, 2024.
- [5] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, 2020.
- [6] Neel Nanda. Attribution patching: Activation patching at industrial scale. <https://www.neelnanda.io/mechanistic-interpretability/attribution-patching>, 2023.
- [7] Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations*, 2024.
- [8] Xu Wang, Yan Hu, Wenyu Du, Reynold Cheng, Benyou Wang, and Difan Zou. Towards understanding fine-tuning mechanisms of LLMs via circuit analysis. In *International Conference on Machine Learning*, 2025.

- [9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.
- [10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- [12] Maja Popović. chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 2015.
- [13] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.
- [14] Melanie Walsh, Anna Preus, and Maria Antoniak. Sonnet or not, bot? poetry evaluation for large models and datasets. *arXiv preprint arXiv:2406.18906*, 2024.
- [15] Luca Baroni, Galvin Khara, Joachim Schaeffer, Marat Subkhankulov, and Stefan Heimersheim. Transformers don’t need LayerNorm at inference time: Scaling LayerNorm removal to GPT-2 XL and the implications for mechanistic interpretability. *arXiv preprint arXiv:2507.02559*, 2025.
- [16] Joakim Edin, Róbert Csordás, Tuukka Ruotsalo, Zhengxuan Wu, Maria Maistro, Casper L. Christensen, Jing Huang, and Lars Maaløe. GIM: Improved interpretability for large language models. *arXiv preprint arXiv:2505.17630*, 2025.
- [17] Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.
- [18] Robert L. Weide. The CMU pronouncing dictionary. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>, 1998.
- [19] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *International Conference on Learning Representations (Workshop Track)*, 2017.
- [20] John Schulman and Thinking Machines Lab. LoRA without regret. <https://thinkingmachines.ai/blog/lora/>, 2025.
- [21] nostalgebraist. Interpreting GPT: The logit lens. LessWrong, 2020.
- [22] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021.

A Rhyme-constrained decoding

At line-ending positions, add a constant β to the logits of vocabulary tokens whose decoded form shares a 2–3 character suffix with the rhyme partner’s final word. This is training-free (a decoding wrapper, one cached vocabulary suffix scan per rhyme target) and needs no pronunciation dictionary at inference, unlike prior constrained poetry decoding [17].

Logit boost β	chrF	Rhyme Rate	Note
0 (baseline)	42.00 $\pm$ 0.39	10.5% $\pm$ 2.1%	
3.0	41.87 $\pm$ 0.28	21.9% $\pm$ 3.8%	
5.0	42.09 $\pm$ 0.36	36.2% $\pm$ 2.4%	sweet spot
10.0	25.68 $\pm$ 1.07	74.1% $\pm$ 4.7%	chrF collapses

Table 4: **Rhyme-constrained decoding** (GPT-2 Large full fine-tune, 5 seeds $\times$ 12 prompts, single-sample). $\beta=5$ gives $3.4\times$ rhyme rate at no measurable chrF cost (overlapping error bars).

Note on relation to the circuit analysis. This intervention was developed before the source/relay/suppressor taxonomy and has no direct mechanistic connection to it: the boost operates on output logits, not on any circuit component. We include it as a practical improvement ($3.4\times$ at no cost) but it is not circuit-informed.

We also tried position-dependent head scaling (amplify circuit heads at line endings), which added nothing over the logit boost; an iambic-pentameter boost (favor tokens whose CMU-dictionary [18] stress pattern matches the expected unstressed/stressed alternation at the current syllable position), which on its own reduced rhyme from 7.5% to 6.2% (the meter constraint fights vocabulary); and a rhyme-aware auxiliary training loss, which also reduced baseline rhyme. A linear probe [19] for syllable stress reaches 97.9% accuracy on the fine-tuned model’s hidden states (best layer), so the model encodes meter cleanly even though our crude suffix-match metric cannot measure it.

B LoRA module targeting across tasks and scales

A natural follow-up to the source/relay/suppressor work: if interp tells us which heads matter, does that help us pick LoRA targets? We tested this and the answer is no (Finding 3), but the module-level question turned out to be interesting independently. We tested five module targets across four tasks (IOI, induction, sonnet, paraphrase) and four GPT-2 scales, 3 training seeds each for the full-matrix cells (~ 170 total runs). Schulman et al. [20] independently showed that LoRA applied to all weight matrices matches full fine-tuning; our results confirm this at smaller scale (Findings 2, 5) and add a per-module breakdown and the negative interp-targeting result (Finding 3).

Setup. Configs: QK (W_Q, W_K), V+O (W_V, W_O), STD (W_Q, W_V ; the HuggingFace default), MLP (MLP up/down only), ALL (V+O+MLP). All $r=8$, $\alpha=16$, lr $1e-4$. IOI and induction use 15-epoch training with best-dev save; sonnet uses 20 epochs with best-chrF save; paraphrase uses 15 epochs with best-dev-acc save (single-seed for the module cells due to ~ 10 h/run cost). Error bars are std over training seeds.

Finding 1: QK is always worst and always stable. Across all four tasks and all scales, QK has the lowest deployed score *and* the lowest training-seed variance (sonnet Small: $\sigma=0.19$; IOI Small: $\sigma=0.03$). Query/key touches where to attend but not what to write; it is the most constrained subspace and also the least useful on its own.

Finding 2: MLP-inclusive configs dominate, and the gap widens with scale on single-token tasks. On IOI, the V+O–QK gap grows monotonically: -1.26 (Small, QK wins) $\rightarrow +1.77$ (Medium) $\rightarrow +1.84$ (Large) $\rightarrow +3.29$ (XL). Figure 3 shows this against sonnet, where the MLP–QK gap instead *compresses* at scale ($1.24 \rightarrow 0.64$) as all configs approach a chrF ceiling ~ 43 . The difference is task structure: IOI is a binary name choice with no ceiling; sonnet is bounded above by “Shakespeare-plausible.”

Finding 3: Head/layer targeting does not help. Three interp-guided targeting strategies all fail:

- *Head-granular LoRA* restricted to the source head L6H10 (13K params) gets chrF 41.03 $\pm$ 0.28, i.e. 96.5% of standard LoRA’s 42.52 at 4.5% of the parameters. But random-

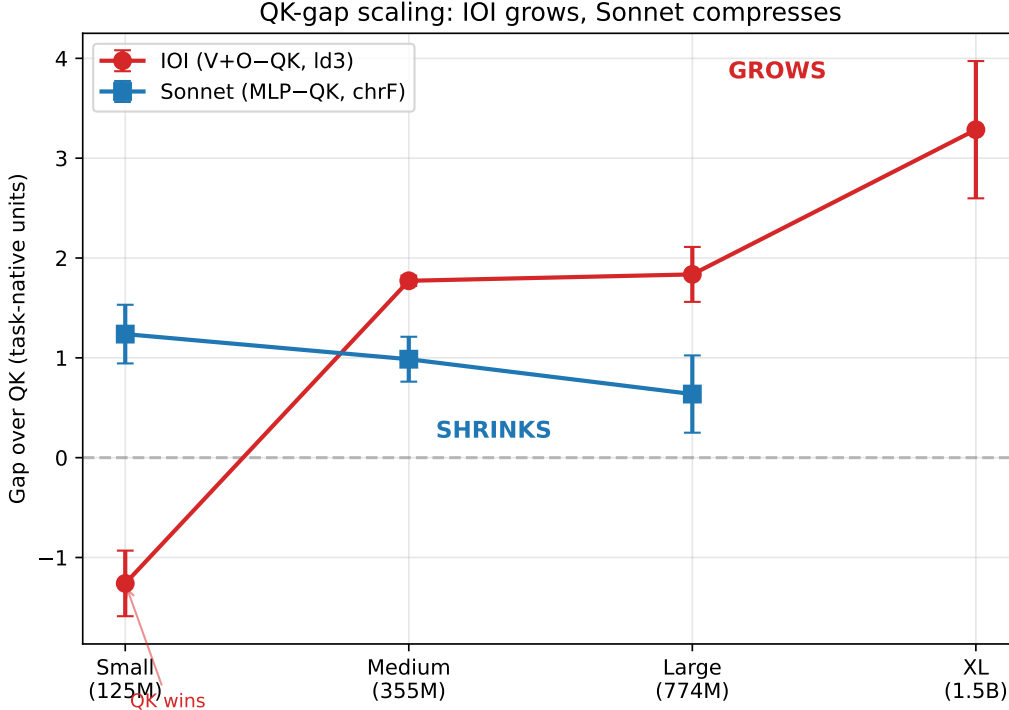


Figure 3: Gap over QK vs scale (IOI: V+O-QK; sonnet: MLP-QK). IOI: gap grows monotonically (QK is actively harmful at XL). Sonnet: gap compresses toward a chrF ceiling. All points $n=3$, error bars are propagated std. Induction (not shown) has $V+O \approx \text{ALL}$ winning at all four scales by $\sim 4\times$ over QK.

head LoRA is indistinguishable, and on IOI, targeting the name-mover heads performs *worse* than random heads at the same budget.

- *Freeze-except* (full FT but only unfreeze the circuit layers) is no better than freeze-except-any-other-layer-set. All choices ≈ 42.5 .
- *Layer-targeted LoRA* restricted to the eight layers identified by path patching plus logit-lens KL [21] achieves chrF 40.93 ± 0.26 vs 42.52 for all-layer LoRA.

Circuits describe inference-time computation, not gradient geometry. Knowing which heads matter does not tell you which parameters are productive gradient targets, and on these tasks the answer is “all of them, roughly equally.”

Finding 4: Optimal rank is task- and scale-dependent. Figure 4. On IOI XL, $r=8$ is optimal; $r=16$ and $r=32$ have lower ceilings and $10\times$ worse ep0 variance. On sonnet Medium, $r=32$ halves the gap to full FT. On sonnet Large, $r=8$ is best again. No universal rule: optimal rank depends on task complexity *relative to* base-model capacity.

Finding 5: The ALL-Full-FT gap is U-shaped in scale on both Default Project tasks. Table 1: at Small, ALL beats Full-FT on both sonnet (+0.14 chrF) and paraphrase (+0.13 pp) at $<1\%$ of the parameters. At Medium the gaps widen to $-0.38 / -0.44$. At Large they narrow again to $-0.13 / -0.29$. The sonnet and paraphrase U-shapes are nearly identical despite different task types (generation vs. classification) and metrics (chrF vs. accuracy). We interpret Medium as the hardest regime for LoRA: the base is weak enough that Full-FT’s extra capacity matters, but strong enough that the LoRA update is small relative to what’s needed. At Small, the task ceiling is low and LoRA’s regularization helps (consistent with [22]); at Large, the base is already near-ceiling and the LoRA update suffices.

Prescription. By default, use ALL at $r=8$. If the gap to full FT is large and the task is complex (style transfer, generation), try $r=32$. Never use QK-only. Circuit knowledge does not translate to

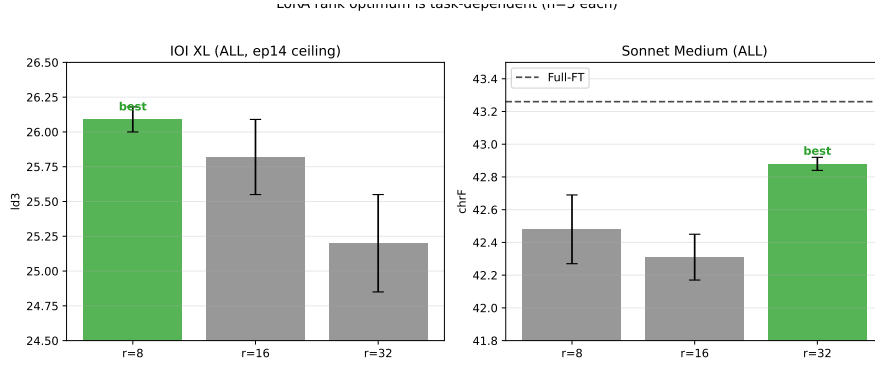


Figure 4: LoRA rank sweep (ALL config), $n=3$ each. IOI XL: monotonic decrease, $r=8$ best. Sonnet Medium: nonmonotonic, $r=32$ best (99.1% of full FT). Full-FT line on the right panel shows the headroom.

useful LoRA placement, as we find that it only tells you what the model *does*, not where gradients should be applied.

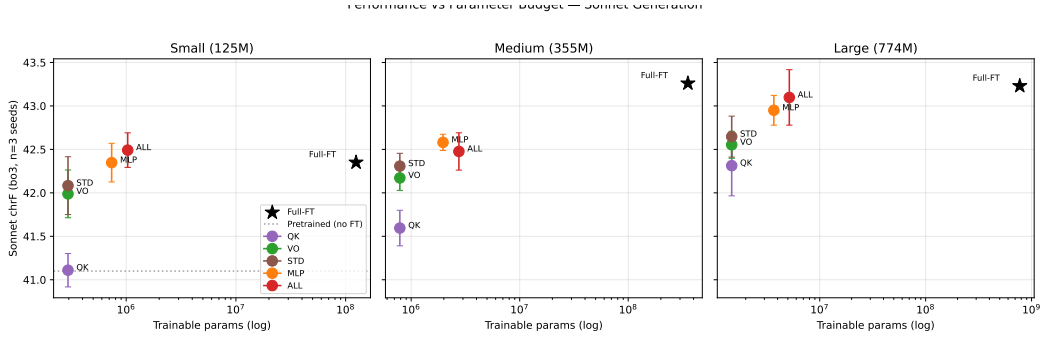


Figure 5: Sonnet chrF vs trainable parameters (log scale), all configs $n=3$. At Small, LoRA beats full fine-tuning. At Medium and Large, ALL is Pareto-optimal among LoRA configs but full FT has a small headroom (≤ 0.38). QK is consistently worst.

C Additional generalization experiments

A per-layer binary classifier reaches 80% accuracy at the final layer, with an 8pp gap between vs-training and vs-held-out discrimination. This measures if it’s a training completion or a held-out completion. Regularization attempts (dropout 0.3/0.5, label smoothing 0.1, cosine learning-rate schedule) were all either approximately neutral or slightly negative for chrF—none improved the memorization/generalization trade-off.

D Model-wide weight-change distribution

Fine-tuning’s weight change $\|W_{FT} - W_{PT}\|/\|W_{PT}\|$ per head, OV projection: median 1.15%, 90th percentile 1.56%, maximum 2.05% (L4H11). L6H10 at 0.74% is the 5th-lowest out of 144 heads. The routing heads from the original path-patching circuit (L4H11 at 2.05%, L5H6 at 1.74%) are at the top of the distribution; the source head is at the bottom. This pattern (routing changes most, computation least) survives as an observation even after the negative weight-patching result; what does not survive is the causal interpretation that any particular routing head’s weight change drives the source head’s attention amplification.