Mechanistic origins of catastrophic forgetting: why RL preserves circuits better than SFT?

Anonymous Authors[1]

Abstract

Fine-tuning large language models (LLMs) frequently induces catastrophic forgetting of prior capabilities. Recent work has shown that reinforcement learning (RL) retains prior capabilities more effectively than supervised fine-tuning (SFT), attributing this to policy-gradient updates remaining closer to the base policy (Shenfeld et al., 2025). We extend this behavioral account to the mechanistic level and ask whether RL's advantage is mirrored by stronger preservation of internal computational circuits. We introduce differential circuit vulnerability, a head-level measure of how much a circuit degrades under fine-tuning, and use it to compare RL and SFT on Qwen2.5-3B-Instruct adapted to scientific question-answering. We find a clear mechanistic trade-off: SFT adapts more rapidly to the target task but produces substantially greater circuit disruption and forgetting of prior mathematical reasoning, whereas RL preserves a larger fraction of the base circuit at the cost of slower task adaptation. These findings suggest that circuit preservation may help explain why RL is more robust to catastrophic forgetting.

1. Introduction

Adapting large language models (LLMs) to new downstream tasks frequently incurs catastrophic forgetting: gains on a new objective come at the expense of prior capabilities. As models are increasingly expected to update continuously and adapt to new domains, mitigating such degradation has emerged as a central challenge of post-training. Prior work shows that the choice of adaptation objective strongly shapes this trade-off: RL typically preserves prior capabilities more effectively than SFT, arguably because policy-gradient updates remain closer to the pretrained model and reduce distributional drift, commonly measured by KL divergence (Shenfeld et al., 2025).

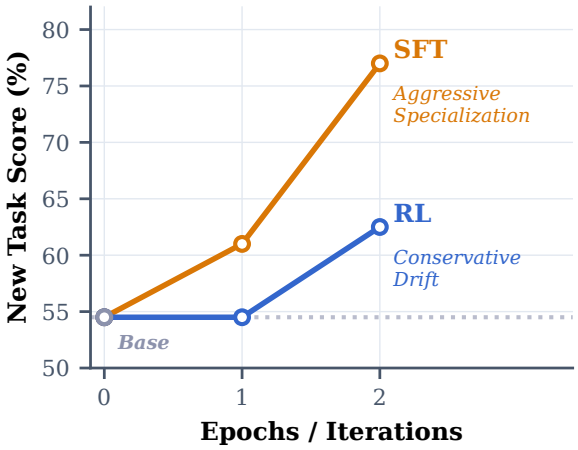

Figure 1. **Temporal overview of task acquisition.** SFT and RL models learn new task capabilities at very different speeds during training. SFT specializes aggressively, rapidly achieving higher task scores within a small number of epochs. On the other hand, RL (GRPO) suffers from conservative drift and gradually improves its performance on the new task over more iterations. We quantify this trade-off in Section 4.2.

In parallel, mechanistic interpretability has shown that model capabilities are implemented by internal computational *circuits* composed of attention heads, MLP layers, and residual pathways. Fine-tuning may therefore succeed or fail depending on whether it preserves these structures or disrupts them (Prakash et al., 2024). This motivates our central question: **can RL's retention advantage be explained by stronger preservation of task-relevant circuits relative to SFT?**

We examine post-training through the lens of circuit preservation. We introduce *differential circuit vulnerability*, the relative susceptibility of internal computational subgraphs to degradation under different training objectives. Using this framework, we compare RL and SFT on matched adaptation tasks and identify which circuits are preserved or disrupted. Figure 1 previews the central finding: the two objectives trace distinct paths through the preservation–performance plane, which we characterize mechanistically in the sections that follow.

We instantiate our analysis on Qwen2.5-3B-Instruct using a two-stage protocol: adaptation on scientific question answering, followed by retention evaluation across a broad benchmark suite spanning commonsense reasoning, factuality, and instruction following. The results expose a consistent trade-off: SFT adapts faster to the new objective but at the cost of greater circuit disruption, whereas RL preserves more of the base circuit and of prior capabilities, occasionally at the expense of under-optimizing the new task.

These results suggest that continual adaptation should be understood not merely as parameter optimization, but as the *selective preservation and modification* of circuits encoding skills. This perspective points toward a mechanistic account of why RL is more conservative than SFT during post-training.

2. Related Work

Reinforcement Learning, Supervised Fine-Tuning, and Forgetting. Supervised fine-tuning (SFT) and reinforcement learning (RL) are the dominant post-training methods for adapting foundation models, yet their effects on previously acquired capabilities remain incompletely understood. SFT optimizes imitation on a target distribution, whereas RL updates behavior through reward optimization and often produces qualitatively different parameter changes. Recent studies suggest that, under comparable settings, RL can preserve pretrained capabilities better than SFT, indicating that the training objective influences both adaptation and forgetting (Shenfeld et al., 2025; Hu et al., 2025).

Mechanistic Interpretability and Circuit Analysis. Mechanistic interpretability explains model behavior through concrete components such as attention heads, MLP layers, and circuits. Intervention-based methods, including activation patching, path patching, and masking, have shown that many behaviors can be localized to sparse causal subnetworks. Related work further suggests that fine-tuning frequently modifies existing mechanisms rather than replacing them entirely (Davies et al., 2023; Prakash et al., 2024).

Evaluating Retention Across Capabilities. Retention cannot be assessed on the adaptation task alone. Models may improve on a new objective while degrading reasoning, factuality, instruction following, or code generation. For this reason, prior work increasingly relies on diverse benchmark suites. Common evaluations include MMLU, HellaSwag, and WinoGrande for reasoning and commonsense inference, and TruthfulQA, IFEval, and HumanEval for factuality, instruction following, and coding (Hendrycks et al., 2021; Zellers et al., 2019; Lin et al., 2022).

Research Gap and Positioning. Despite progress in both post-training and interpretability, the two literatures remain loosely connected. Optimization studies typically report benchmark outcomes without explaining the internal causes of forgetting, while interpretability studies rarely compare learning objectives. We address this gap by comparing RL and SFT through the lens of circuit preservation, asking which objective better maintains the causal structures underlying pretrained capabilities.

3. Methodology

We test the hypothesis that RL retains prior capabilities because it preserves task-relevant internal circuits more effectively than SFT. Our pipeline comprises three phases: (I) reproduce the known retention gap between SFT and RL; (II) identify circuits in each model; and (III) compare how post-training reshapes these circuits.

We begin from a pretrained model π_{base}. We first train an SFT model π_{SFT} via completion-only supervision, then refine it with Dr.GRPO to obtain π_{RL}. Thus, our comparison isolates the effect of continuing post-training with RL beyond SFT.

$$\pi_\theta \in \{\pi_{\text{SFT}}, \pi_{\text{RL}}\} \tag{1}$$

To quantify behavioral drift from the base model, we compute the expected KL divergence on retention tasks:

$$\mathbb{E}_{x\sim\tau}\left[D_{\text{KL}}\left(\pi_{\text{base}}(\cdot|x) \,\|\, \pi_\theta(\cdot|x)\right)\right]. \tag{2}$$

Lower values correspond to less distributional shift.

3.1. Phase I: Reproducing Distributional Shift Effects

We first verify that RL preserves prior capabilities better than SFT while remaining closer to the pretrained policy (Shenfeld et al., 2025). Models are trained on a downstream Task A and evaluated on a separate suite of retention benchmarks (Task B).

SFT. We fine-tune π_{base} with a completion-only cross-entropy loss.

RL. We refine π_{SFT} with Dr.GRPO. The model samples candidate completions, receives binary rewards, computes normalized group-relative advantages, and updates the policy through a weighted log-probability objective. We use a group size of 64, two refinement steps ($\mu = 2$), and no explicit KL penalty.

3.2. Phase II: Circuit Identification via Differential Binary Masking

We analyze circuits at the attention-head level using Differential Binary Masking (DBM) (Chaudhary & Geiger, 2024). DBM learns a mask over heads that interpolates between base and counterfactual activations:

$$\tilde{a}_h = (1 - m_h)a_h^{\text{base}} + m_h a_h^{\text{source}}, \tag{3}$$

where $m_h \in [0, 1]$. Annealing pushes the masks toward binary selections, yielding sparse causal circuits.

Triplets. For chemistry QA, we construct triplets $(x_{\text{base}}, x_{\text{source}}, y_{\text{target}})$ corresponding to three counterfactual hypotheses: answer-key swaps, molecule swaps, and task-type swaps.

Objective. Masks are optimized to increase the probability of the target answer while remaining sparse:

$$\mathcal{L}_{\text{DBM}} = -\log P(y_{\text{target}}|x, \tilde{a}) + \lambda \sum_h m_h. \tag{4}$$

Scoring. We score answers using the geometric mean of token probabilities:

$$p(y|x) = \exp\left(\frac{1}{T}\sum_{i=1}^{T} \log P(y_i|x, y_{<i})\right). \tag{5}$$

Circuit discovery is run independently for π_{base}, π_{SFT}, and π_{RL}.

3.3. Phase III: Cross-Model Circuit Comparison

We assess how strongly the discovered circuits remain functional after post-training. Circuit faithfulness is defined as

$$\text{Faithfulness}(\mathcal{C}, M) = \frac{F(\mathcal{C}|M)}{F(M)}. \tag{6}$$

with values close to 1 indicating that the circuit recovers most of the model's behavior.

For each head h, we compare DBM mask values against those of the base model:

$$\Delta m_h(M) = m_h^M - m_h^{\text{base}}, \tag{7}$$

where $M \in \{\pi_{\text{SFT}}, \pi_{\text{RL}}\}$. This identifies heads that are preserved, amplified, or weakened by training.

We define *vulnerable* heads as those more degraded under SFT than under RL:

$$\mathcal{C}_{\text{vuln}} = \{h : \ m_h^{\text{SFT}} < m_h^{\text{RL}} - \delta\}. \tag{8}$$

4. Experiments

Our experiments evaluate the central claim of the paper: RL retains prior capabilities because it preserves task-relevant circuits more effectively than SFT. We address two questions: (I) Does RL induce less forgetting and lower behavioral drift than SFT; and (II) are these gains reflected in stronger circuit faithfulness, more stable head-level contributions, and more distributed circuits?

4.1. Experimental Setup

All experiments use Qwen2.5-3B-Instruct. The fine-tuning task (Task A) is scientific question answering. Retention (Task B) is measured on benchmarks spanning commonsense reasoning, factuality, instruction following, and code generation, since forgetting can be capability-specific. We compare three systems: the pretrained base model, standard SFT, and RL with Dr.GRPO. Behavior is measured with downstream accuracy and KL divergence from the base model. The mechanistic analysis uses circuit faithfulness, head-level mask shifts $\Delta m_h = m_h^M - m_h^{\text{base}}$ derived from cross-model DBM mask comparison, and necessity/sufficiency interventions; necessity measures the log-probability drop when a head is disrupted, while sufficiency measures how much behavior is recovered when only that head is kept. Together, these metrics distinguish distributed contributors from critical bottlenecks. Table 1 summarizes the full setup.

4.2. Results

DBM identifies a base circuit comprising 290 attention heads. After adaptation, the SFT model exhibits structural compression to roughly 265 heads (46.0% of all attention heads), whereas the RL model retains approximately 296 heads (51.4%), close to the base model's 297 heads (51.6%). The same disparity is reflected in base-circuit overlap: the RL model preserves about 68% of base heads, substantially more than the SFT model's 52% (Figure 4).

Across new-task strength (NTS) levels, **Figure 2** traces the performance–preservation trade-off for both objectives on a single axis. SFT preservation remains relatively steady at low NTS (72.8%) and medium NTS (70.8%), but drops to 51.5% at high NTS, a 19.3 percentage-point decline. RL decays gradually, from 75.2% at low NTS to 67.3% at high NTS. At peak new-task performance, RL retains 15.8 percentage points more of the base model's circuit than SFT.

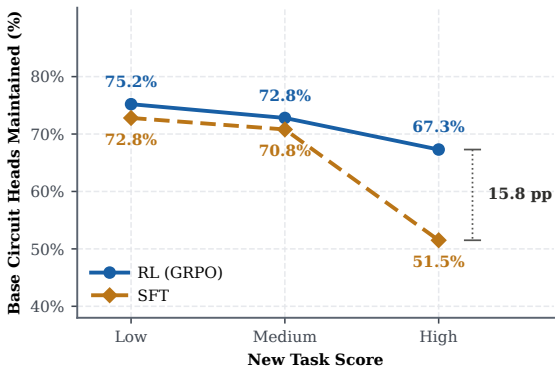

Figure 2. **Performance–preservation trade-off across NT levels.** SFT (dashed) exhibits a sharp preservation drop in the high-NTs regime, while RL (solid) declines gradually and preserves 15.8 percentage points more of the base circuit at peak new-task performance.

The above results were validated by measuring circuit faithfulness across all three models. Faithfulness was 1.02 for the base, 1.04 for SFT, and 1.12 for RL, indicating that the extracted subgraphs effectively drive model behavior.

4.3. Functional Importance: Necessity vs. Sufficiency

We analyze individual head importance using necessity and sufficiency interventions. Necessity measures the performance drop when a head is ablated, while sufficiency measures how much behavior is recovered when that head acts in isolation. Together, these metrics distinguish distributed contributors from critical bottlenecks.

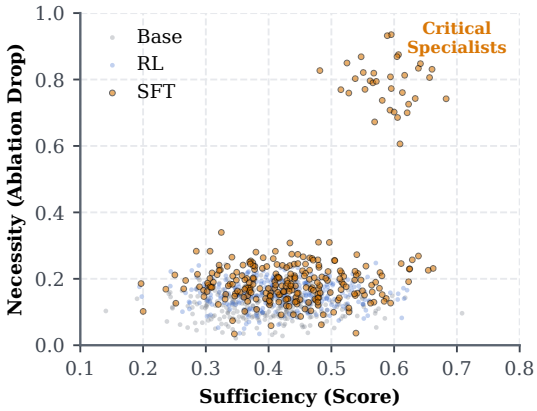

Figure 3. **Head Role Distribution Under Base, Supervised, and RL Training.** In our setup, SFT produces a cluster of "Critical Specialists"—heads with high necessity and sufficiency—while RL maintains a distributed architecture that overlaps closely with the base model, avoiding the structural compression and specialization observed under the supervised objective.

Figure 3 shows that the SFT model contains a larger population of highly necessary heads than the RL model, indicating greater reliance on a small set of critical components. The RL model, by contrast, exhibits fewer necessary heads while retaining many sufficient heads—a signature of a more distributed and redundant circuit. This redundancy plausibly underlies the greater robustness to forgetting we observe under RL.

4.4. Discussion

Taken together, our findings indicate that SFT exhibits a mechanistic "breaking point." SFT and RL are comparable at low levels of adaptation, suggesting that both objectives can initially improve new-task performance without substantially disrupting the base circuit. In the high-NTS regime, however, SFT shifts from conservative adaptation to aggressive circuit reconfiguration, while RL remains comparatively stable. This pattern is consistent with RL preserving and reusing existing representations more effectively than SFT, which appears to overwrite portions of the base circuit under stronger new-task pressure. Confirming this interpretation will require experiments across additional models, tasks, and optimization settings.

We also observe that the RL model maintains greater prior-task retention and circuit preservation despite having a larger output-space Kullback–Leibler (KL) divergence than the SFT model. This suggests that output-space metrics may not reliably predict internal forgetting in this setting. In particular, RL may alter the model's output distribution while preserving much of the underlying computation responsible for prior-task performance. By contrast, SFT can appear less divergent at the output level while still reorganizing the internal circuits more aggressively.

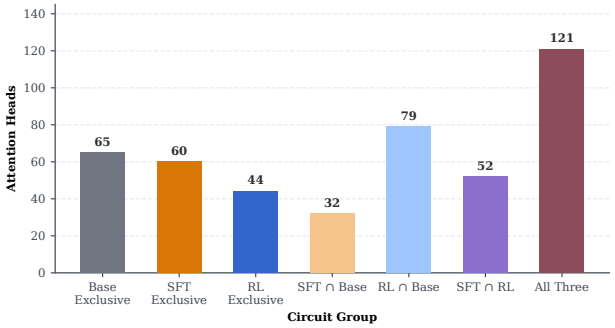

Figure 4. **Attention Head Overlap Between Base, SFT, and RL.** The plot shows the circuit overlap study for the base, SFT, and RL models. The bars reflect the number of attention heads that are unique to one model, shared by two models, or present at all three levels of training.

The gap between the SFT and RL circuit sizes ($\sim$ 265 vs. $\sim$ 295 heads), depicted in Figure 4, supports a characterization of SFT as a "compressor" and RL as a "distributed adaptor."

SFT concentrates new-task behaviour into a smaller, more specialized circuit, while RL preserves a broader set of base-model heads during adaptation.

5. Conclusion

We asked whether RL retains prior capabilities better than SFT because it better preserves internal circuits. Our behavioral and mechanistic analyses on Qwen2.5-3B-Instruct support this hypothesis: RL stays closer to the pretrained model, preserves more of the original circuit, and maintains stronger functional faithfulness, while SFT adapts faster but causes greater circuit reorganization and more forgetting in our experiments.

These results point to a potential trade-off in post-training: rapid specialization versus stable reuse of existing mechanisms. In our setup, SFT optimizes the new objective efficiently but can disrupt circuits supporting prior skills, while RL is more conservative, preserving those circuits at the cost of smaller task gains. More broadly, future adaptation methods may benefit from combining efficient learning with selective circuit preservation, toward models that continue improving without forgetting.

6. Limitations and Future Work

This study examines only one model (Qwen2.5-3B-Instruct), limiting the generalizability of our findings. Future work should validate these trade-offs across diverse architectures: Gemma, Mistral, Llama, and Pythia, and at varying parameter scales. Our circuit analysis is further constrained to attention heads and a narrow task set. Expanding to MLP layers, residual-stream features, and broader capability domains: multilingual reasoning, factual recall, safety, and tool use, would yield a more complete mechanistic picture.

References

Chaudhary, M. and Geiger, A. Evaluating open-source sparse autoencoders on disentangling factual knowledge in gpt-2 small. *arXiv preprint arXiv:2409.04478*, 2024.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Davies, X., Nadeau, M., Prakash, N., Shaham, T. R., and Bau, D. Discovering variable binding circuitry with desiderata. *arXiv preprint arXiv:2310.02336*, 2023.

Feng, K., Shen, X., Wang, W., Zhuang, X., Tang, Y., Zhang, Q., and Ding, K. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint*, 2025.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2021.

Hu, J., Zhang, Y., Han, Q., Jiang, D., Zhang, X., and Shum, H.-Y. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint*, 2025.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2022.

Prakash, N., Shaham, T. R., Haklay, T., Belinkov, Y., and Bau, D. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations (ICLR)*, 2024.

Sakaguchi, K., Le Bras, R., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence*, 2020.

Shenfeld, I., Pari, J., and Agrawal, P. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. Instruction-following evaluation for large language models. In *arXiv preprint arXiv:2311.07911*, 2023.

A. Appendix

Table 1. Experimental Setup

Component	Details
Model	Qwen2.5-3B-Instruct
Task A (Fine-tuning)	SciKnowEval (Feng et al., 2025) (Science Q&A)
Task B (Retention)	HellaSwag (Zellers et al., 2019), TruthfulQA (Lin et al., 2022), MMLU (Hendrycks et al., 2021), IFEval (Zhou et al., 2023), WinoGrande (Sakaguchi et al., 2020), HumanEval (Chen et al., 2021)
Metrics	KL divergence (Eq. 2) Adaptation performance and retention accuracy Circuit faithfulness (Eq. 6) Head-level mask shift $\Delta m_h(M)$ from CMAP (Eq. 7)
Baselines	Base model (no fine-tuning) SFT (standard) RL (Dr.GRPO)

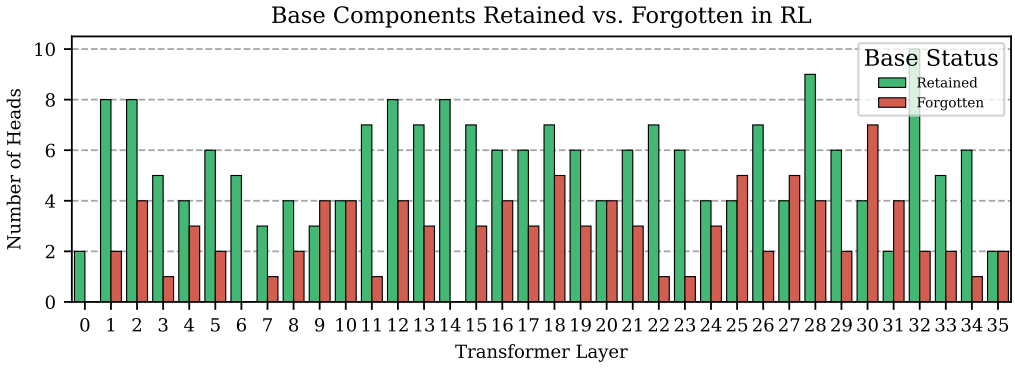

Figure 5. Layer-wise Circuit Retention in RL — Our RL model shows architectural stability across all 36 transformer layers, with a high count of retained heads and relatively few forgotten components throughout the network depth.

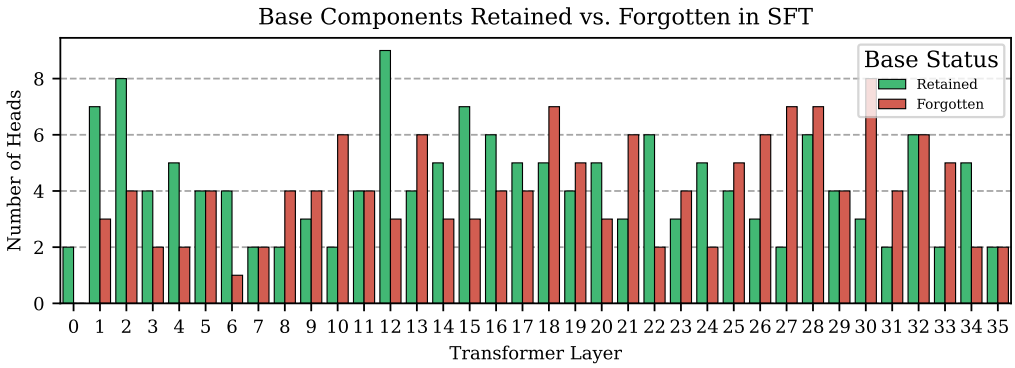

Figure 6. Layer-wise Circuit Retention in SFT — Our SFT model shows broader structural change, with forgotten heads scattered throughout all layers and higher concentrations in the mid-to-late transformer layers.

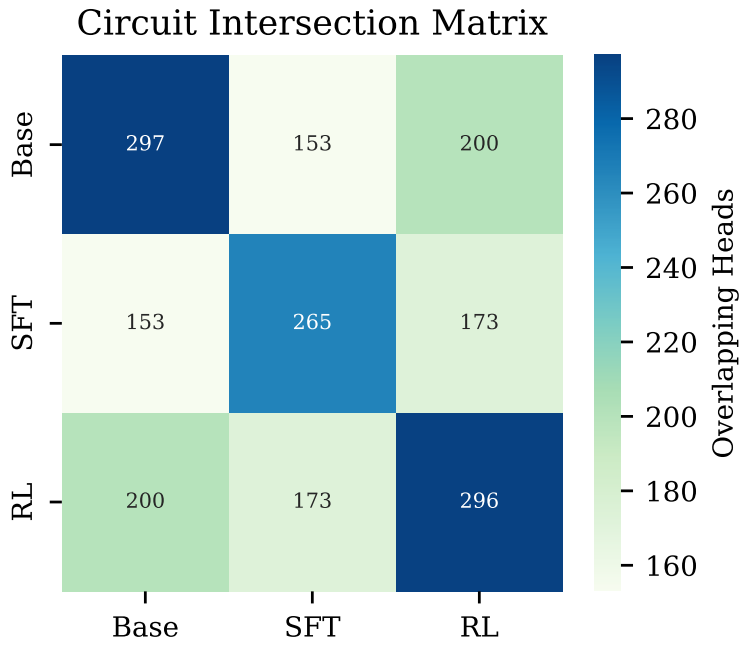

Figure 7. This graph illustrates the number of shared (overlapping) heads among the 'Base', 'SFT' (Supervised Fine-Tuning), and 'RL' (Reinforcement Learning) circuits. Diagonal elements (such as Base-Base, SFT-SFT, and RL-RL) indicate the entire size (number of heads) of each particular circuit. Off-diagonal elements (e.g., Base-SFT, SFT-RL) represent the number of heads shared between two separate circuits. For example, the cell at (Base, SFT) displays how many heads are present in both the Base and SFT circuits.