Where Knowledge Collides: A Mechanistic Study of Intra-Memory Knowledge Conflict in Language Models

Anonymous ACL submission

Abstract

In language models (LMs), intra-memory knowledge conflict arises when inconsistent information about the same entity, event, or concept is encoded within the model's parametric knowledge (Xu et al., 2024). Prior work has primarily focused on resolving conflicts between a model's internal knowledge and external resources through approaches such as fine-tuning or knowledge editing. However, the problem of localizing conflicts that originate during pre-training within the model's internal representations remain laregely unexplored. In this work, we design a framework based on mechanistic interpretability methods to identify where conflicting knowledge from the pre-training data is encoded within LMs. While our findings align with prior work showing that specific internal components can encode conflicting knowledge, we study a distinct problem: conflicts originating from pre-training data in a clean, controlled setting that has not been systematically explored. We further demonstrate how mechanistic interpretability methods can be leveraged to causally intervene and control conflicting knowledge at inference time.[1]

1 Introduction

As information evolves rapidly, large language models (LLMs) face the challenge of keeping their parametric knowledge accurate to remain reliable and reduce hallucinations (Jing et al., 2025). Continual learning (Ke et al., 2023) and retrieval-augmented generation (Lewis et al., 2020; Borgeaud et al., 2021) aim to keep model knowledge up to date. However, models can still encode contradictory information within their parameters, a phenomenon known as *intra-memory knowledge conflict* (Xu et al., 2024). Such conflicts compromise model reliability by creating unstable internal representations, where different prompts or contexts elicit inconsistent factual associations and lead to conflicting generations. These conflicts arise from biases and noise in training data (Bender et al., 2021; Lazaridou et al., 2021), limitations of knowledge editing (Li et al., 2024b), and sampling randomness in decoding strategy of the model (Huang et al., 2025), raising broader reliability concerns as LLM-backed systems become more integrated into daily use (Bengio et al., 2024).

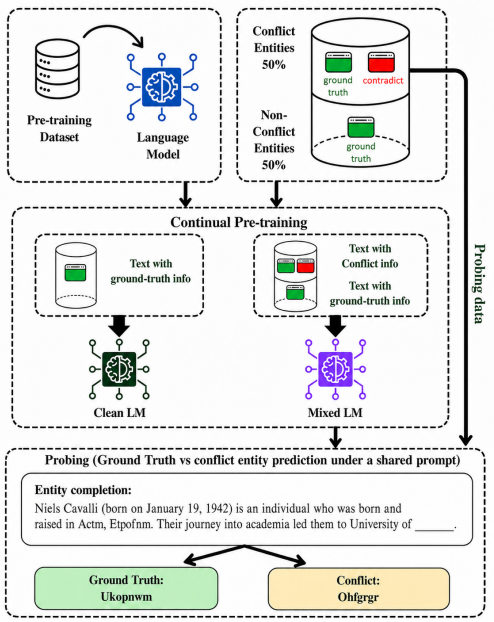


Figure 1: A high-level overview (with example) of our proposed framework. The university names are randomized to prevent contamination from pretrained data.

Existing conflict mitigation methods focus on improving factual accuracy and consistency, including targeted fine-tuning (Neeman et al., 2023), context-faithful prompting (Zhou et al., 2023), and pruning internal components (i.e., conflicting attention heads) (Jin et al., 2024). However, no prior work has addressed a fundamental question: *which internal components are responsible for conflicts originating in pre-training data, and can noisy or incorrect information be selectively suppressed at inference time?* Answering this is essential for un-

[1]We will release the resources and code upon acceptance.

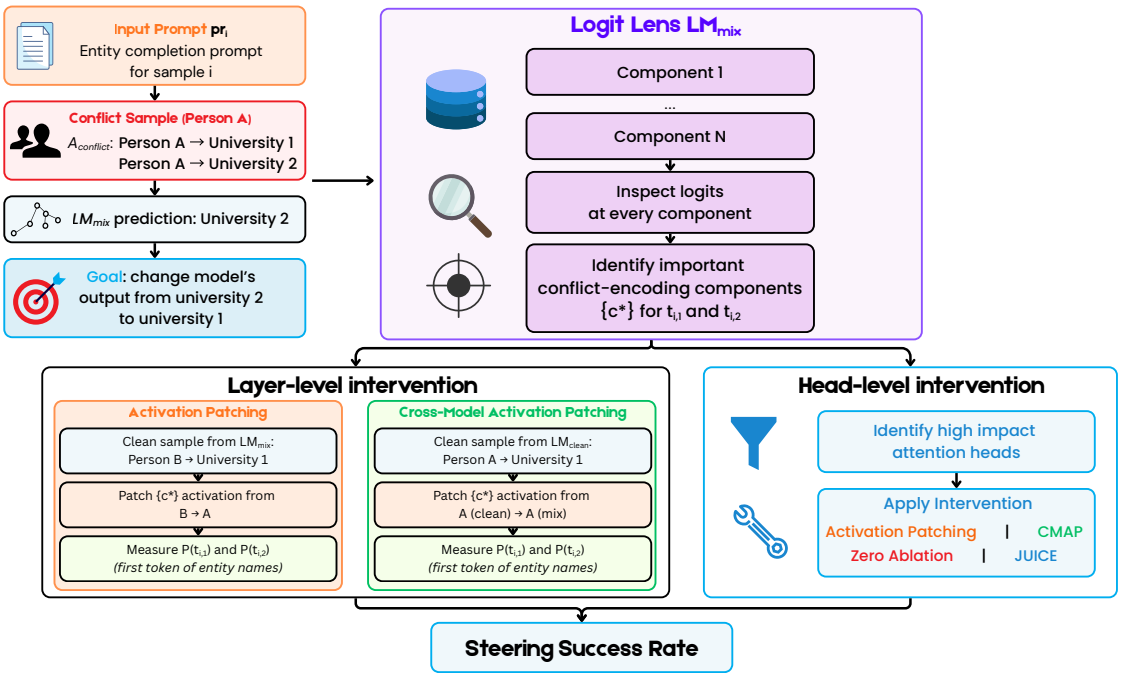


Figure 2: Overview of the three-stage pipeline for probing intra-memory knowledge conflict. The numbered stages of the pipeline outline the experimental procedure used to localize and analyze the conflicting model components.

derstanding intra-memory conflict.

In this work, we propose a framework grounded in mechanistic interpretability (MI) to localize where conflicting pre-training knowledge is encoded. As illustrated in Figure 1, we construct `SynWikiBio`, a curated set of Wikipedia-style biographies for synthetic entities that deliberately contain conflicting facts (e.g., a person's graduating university), and use it to fine-tune a language model (e.g., GPT-2 XL). We then probe the fine-tuned model with prompts designed to elicit competing knowledge representations (Section 3.1), apply the *logit lens* to localize the components, such as attention layers or attention heads, that encode the conflict (Section 4.2), and assess their causal contribution via *activation patching* (Section 4.3) and *cross-model activation patching (CMAP)* (Prakash et al., 2024), which enables cleaner causal attribution under conflicting knowledge (Section 4.4).

This framework lets us investigate four research questions: (**Q1**) how does conflicting training data influence factual predictions, which components most affect this decision, and to what extent does activation patching localize them? (**Q2**) how reliably can targeted patching and other interventions alter predictions to resolve conflict? (**Q3**) how transferable are patching interventions across models sharing the same architecture but trained on different datasets? (**Q4**) how do the identified mechanisms vary across distinct architectures, such as GPT-2 XL and Qwen3?

Through layer- and head-level interventions across two models (Section 5), our findings show that: (1) conflicting training data reduces late-layer confidence during output generation of the model, with attention components playing a central role in conflict encoding; (2) layer-level intervention and targeted head-level intervention can be effective depending on the type of conflict or the LLM's architecture, with targeted head-level intervention achieves 80% success rate on *company* type conflicts with GPT-2 XL while attention-layer activation patching achieves upto 70% steering success rate on the *university* conflict type with Qwen3-4B; (3) CMAP is a promising alternative for causal tracing in knowledge conflict settings, performing on par with activation patching; and (4) conflict resolution components are concentrated in a small number of attention heads across many layers in GPT-2 XL but distributed across many heads in a narrow layer band in Qwen3-4B. These results demonstrate that conflicting knowledge in LMs can be causally localized and manipulated through internal representations. More broadly, our study provides a concrete step toward diagnosing and controlling factual conflicts in language models, with implications for model editing, reliability, and trustworthy deployment.

2 Related Work

Intra-Memory Knowledge Conflict is the type of knowledge conflict where the model generates different responses to similar input, primarily due to conflicting information present in the training data (Xu et al., 2024). Approaches to mitigate such conflicts aim to reduce inconsistency and improve generation factuality (Xu et al., 2024). Examples include counterfactual argument fine-tuning (Neeman et al., 2023), and external model for output evaluation (Mitchell et al., 2022; Zhao et al., 2024). As for a model-centric solution, Jin et al. (2024) proposes a pruning method based on information flow within attention heads to mitigate conflicts in model-generated responses. However, the internal origins of such conflicts remain unexplored. We address this gap by developing a mechanistic-interpretability-based framework to localize conflicting knowledge encoded in language models.

Mechanistic Interpretability (MI) is the field in machine learning that focuses on understanding the internal mechanisms of neural networks (Olah et al., 2020; Bereska and Gavves, 2024). Prior work has applied *logit lens* (Ortu et al., 2025) and *activation patching* (Zhang and Nanda, 2024) for a range of analyses and interventions, including explaining how LMs handle context-memory conflict (Ortu et al., 2024) and analyzing knowledge selection behavior in LMs (Zhao et al., 2025). In this work, we apply both techniques to localize and causally intervene on conflicting knowledge encoded in language models that originates from the training data.

3 Experiment Setup

3.1 Problem Definition

We formalize the problem as follows. Let $\mathcal{B}$ denote a dataset comprising personal biographies. A biography $b_i \in \mathcal{B}$ is said to contain knowledge conflict if there exists another biography $\bar{b}_i \in \mathcal{B}$ such that b_i and $\bar{b}_i$ refer to the same individual p_i and share identical personal attributes except for one conflicting element. On the other hand, a conflict-free instance $b_i \in \mathcal{B}$ means that b_i is the only biography in $\mathcal{B}$ that describes person p_i.

First we construct dataset $\mathcal{B}_{mix}$, which comprises both conflict-containing and conflict-free personal biographies. Let ℓ be a pretrained language model, we continue pretraining ℓ on $\mathcal{B}_{mix}$, yielding model ℓ_{mix}.

Following established mechanistic interpretability studies (Elhage et al., 2021; Hanna et al., 2023; Ortu et al., 2024), we formulate an entity completion task on model ℓ_{mix} that explicitly elicits conflicting parametric knowledge as illustrated in Figure 1. Specifically, for each person p_i whose biography contains conflicting information, we apply p_i's name and related information to a randomly selected predefined template (full list of templates is included in appendix E) to create prompt pr_i that induces the conflict information from ℓ_{mix}'s parametric knowledge (e.g., "Niels Cavalli studies at University of ___"). The process of recalling both knowledge internally by ℓ_{mix} can be traced by investigating the logits of these knowledge's first tokens,which by construction uniquely identify each entity (c.f. Appendix B.1). We additionally verify consistency with the full generated entity. For future reference, we use $t_{i,1} \in b_i$, $t_{i,2} \in \bar{b}_i$ as a reference to the first tokens of the conflicting attribute, for any individual p_i in $\mathcal{B}_{mix}$ whose biography contains a contradictory factual claim per definition.

3.2 Dataset

For this study, we create a synthetic dataset of personal biographies ($\mathcal{B}$) with Wikipedia short biography style consisting of fictional individuals, each with a unique set of personal attributes: *birth date*, *birth place*, *university*, *major*, *company*, and *work place* (Allen-Zhu and Li, 2024; Li et al., 2024a; Zucchet et al., 2025). We follow the generation process from Li et al. (2024a) with GPT-4o (Hurst et al., 2024) to generate the biographies. To promote model's generalizing capability, we apply a randomly selected template to each of the biography. Entity names of important attributes *birth place*, *company* and *university* are replaced with random generated words. Less relevant attributes (i.e. *major*, *work place*) remain unchanged. The conflict attribute in the biography is either in *university* or *company* category.

From $\mathcal{B}$ we construct $\mathcal{B}_{mix}$ (named `SynWikiBio`) containing $n_1 = 1000$ people whose biographies contain knowledge conflict and an additional $n_2 = 1000$ people with conflict-free biographies. In addition, we construct $\mathcal{B}_{clean}$ (named `SynWikiBio_clean`) from $\mathcal{B}_{mix}$ by removing $\bar{b}_i$ from each conflicting biographies pair $(b_i, \bar{b}_i)$, thereby making $\mathcal{B}_{clean}$ free of conflicts. Continue pretraining ℓ with $\mathcal{B}_{clean}$ results in second model ℓ_{clean}. Details of dataset generation with statistics are provided in Appendix B.1.

3.3 Models

We evaluate our framework on two language models, GPT-2 XL 1.5B (Radford et al., 2019) and Qwen3-4B (Yang et al., 2025). We select these two models to cover both widely studied architecture that serves as a standard baseline in mechanistic interpretability research, and a larger model to test generalizability across scales. We use Transformers library (Wolf et al., 2019) for training the models, and TransformerLens (Nanda and Bloom, 2022) for causal tracing tasks. The implementation details (including hyperparameters used) are reported in Appendix A.

4 Causal Probing Intra-Memory Knowledge Conflict

As shown in Figure 2, we develop a framework based on ℓ_{mix} and ℓ_{clean} to probe model's parametric knowledge. First we perform logit inspection (Nostalgebrist, 2020) in the residual stream of ℓ_{mix} to localize components responsible for encoding and generating conflicting information from the training data `SynWikiBio` (Section 4.2 and Appendix C). We then apply activation patching (Zhang and Nanda, 2024) within ℓ_{mix} to test whether these components causally influence the model's generation with respect to the conflicting facts (Section 4.3). Finally, we experiment with cross-model activation patching, an advance causal tracing method that leverages both ℓ_{mix} and ℓ_{clean} to analyze intra-memory knowledge conflicts (Section 4.4).

4.1 Preliminaries

Logit lens is a technique for inspecting a model's intermediate representations by decoding the residual stream at each layer, thereby revealing how model's prediction evolves across different components (Nostalgebrist, 2020).

Activation patching identifies task-relevant components by replacing a component's activation under a target prompt with its activation from an alternative prompt and measuring the resulting change in model behavior (Meng et al., 2022; Zhang and Nanda, 2024). We denote pr_j as the *source prompt* and pr_i as the *target prompt*. A forward pass $\ell_{mix}(pr_j)$ produces a set of activations $A_j = \{a_j^{l,c} \mid l \in L, c \in \{\text{attn, mlp}\}\}$, where L is the total number of layers in ℓ_{mix}, and likewise $\ell_{mix}(pr_i)$ produces A_i. To measure the causal effect of a specific component $c \in \{\text{attn, mlp}\}$ at layer l in pr_i's forward pass, we execute $\ell_{mix}(pr_i)$ as usual, but at layer l we replace the activation $a_i^{l,c}$ of c with the corresponding activation $a_j^{l,c}$ from the source run. The forward pass then proceeds with all downstream activations recomputed based on the patched value. We denote this modified forward pass as $\ell_{mix}(pr_i \mid a_i^{l,c} \leftarrow a_j^{l,c})$. The impact is then measured by the output differences between $\ell_{mix}(pr_i \mid a_i^{l,c} \leftarrow a_j^{l,c})$ and $\ell_{mix}(pr_i)$.

Zero ablation replaces a component's activation with zero to measure its causal contribution to the model's output.

JUICE (Li et al., 2025) is an intervention method that runs the model twice. The first pass caches activations of the attention heads contributing most to a target token, the second pass patches these activations back, scaled by a factor α, to steer generation toward the desired source.

4.2 Locating Conflict-Encoding Components

First, to identify the layer-wise components responsible for encoding conflicting information in ℓ_{mix}, we track layerwise probability changes of the parametric knowledge tokens ($t_{i,1}$ and $t_{i,2}$) using the *logit lens*. For each individual p_i in `SynWikiBio` whose biography contains a contradictory factual claim, we prompt ℓ_{mix} with pr_i (Section 3.1) and perform the following analysis.

At each transformer layer $l \in L$, we extract residual stream representations at three locations: the layer input x^l_{pre}, the post-attention stream x^l_{mid}, and the post-MLP stream $x^l_{\text{post}} = x^{l+1}_{\text{pre}}$. Each is projected to the vocabulary via the unembedding W_U and softmaxed:

$$P^l_s = \text{softmax}\Big(W_U\, x^l_s\Big)\,,\ s \in \{\text{pre}, \text{mid}, \text{post}\}. \tag{1}$$

We define the contribution of each sub-block at layer l to a token t as the change in its predicted probability across that sub-block:

$$\text{contrib}^l_{\text{attn}}(t) = P^l_{\text{mid}}(t) - P^l_{\text{pre}}(t), \tag{2}$$

$$\text{contrib}^l_{\text{mlp}}(t) = P^l_{\text{post}}(t) - P^l_{\text{mid}}(t). \tag{3}$$

We hypothesize that components responsible for parametric recall show systematically different contribution patterns for conflicting tokens than for unrelated high-probability alternatives. For each prompt pr_i we track three token groups: (i) $t_{i,1}$ from the first parametric fact, (ii) $t_{i,2}$ from the second, and (iii) a control set T_i of the top five predicted tokens excluding $t_{i,1}$ and $t_{i,2}$, for which we

report the mean contribution. We aggregate contribution scores of each component c at each layer l across all n_1 individuals. Since each pr_i is tied to a specific attribute type (e.g., *graduated university* or *employer*[2]), this supports stratified analyses identifying components selectively important for attribute categories or specialized to individual attribute values (e.g., *University of Zinl*).

In addition, we extend the logit lens to analyze individual attention-heads that play an important role to encode conflict knowledge in ℓ_{mix}, further details are provided in Appendix C.

4.3 Causal Tracing via Activation Patching

After identifying a set of components that are strongly associated with conflicting information for a given attribute type (e.g., *graduated university*) in ℓ_{mix}, we apply activation patching to causally assess the contribution of each component c to the model's output behavior.

Specifically, for each individual p_i in `SynWikiBio` whose biography b_i contains knowledge conflict, we quantify the causal effect of component c using target prompt pr_i which induces competition between two next-token continuations, $t_{i,1}$ and $t_{i,2}$, corresponding to biography b_i and $\bar{b}_i$ respectively (as formalized in Section 3.1). We select two biographies b_j from the conflict-free set, results in two source prompts pr_j. b_j are selected such that two forward passes $\ell_{mix}(pr_j)$ produce $t_{i,1}$ and $t_{i,2}$ respectively. Let $P_t(\cdot)$ and $P'_t(\cdot)$ denote the probability assigned by the model to token t before and after activation patching. The causal effect of component c at layer l on the generation of token $t_{i,1}$ and $t_{i,2}$ under the prompt pr_i is defined as:

$$\Delta^{l,\,c}_{t_{i,1}} \;:=\; P'_{t_{i,1}} \;-\; P_{t_{i,1}}$$

$$\Delta^{l,\,c}_{t_{i,2}} \;:=\; P'_{t_{i,2}} \;-\; P_{t_{i,2}}$$

The forward pass $\ell_{mix}(pr_i \mid a_i^{l,\,c} \leftarrow a_j^{l,\,c})$ is patched at the position of the last input token, immediately preceding the generation of the next token t. This intervention isolates the causal influence of component c on resolving factual conflicts by measuring how substituting a clean, fact-consistent activation alters the model's preference between competing continuations.

[2]We use 'company' for employer and 'university' for graduated university, consistent with prior work.

4.4 Cross Model Activation Patching

Standard activation patching sources activations from prompts processed by the same model, implicitly assuming the availability of a similar yet non-confounded source prompt. This assumption fails under intra-memory knowledge conflict: a source prompt pr_j whose biography b_j is absent from training may fail to elicit faithful recall, while one too similar to the target biography b_i risks being conflated with it, causing the model to encode overlapping or competing internal representations. In either case, the model either fails to retrieve the intended fact or cannot cleanly separate source and target memories, yielding unstable activations.

In the previous section we mitigated this by selecting source prompts pr_j that share the target attribute value but correspond to a different individual p_j. While this reduces prompt-target entanglement, it introduces noise from inter-person variation, weakening causal attribution under factual conflict.

To resolve this, we adopt *cross-model activation patching (CMAP)* (Prakash et al., 2024), tailored here to intra-memory conflict. Rather than sourcing activations from alternative prompts within the conflicted model ℓ_{mix}, CMAP transfers activations from a *clean* reference model ℓ_{clean}, obtained by continuing pretraining the base model ℓ on `SynWikiBio_clean`, which contains one conflict-free biography per individual and no contradictory claims. During patching, activations from ℓ_{clean} are injected into ℓ_{mix} at corresponding components and layers while processing the conflicting prompt pr_i. The two models share the same base initialization and the same set of individuals $\{p_i\}$, differing only in the presence of contradictory biography $\bar{b}_i$. This alignment eliminates confounds from prompt mismatch and inter-person variation, ensuring that patched activations reflect uncontaminated parametric representations of b_i and enabling more reliable causal attribution of the components mediating factual-conflict resolution in ℓ_{mix}.

5 Results and Findings

Results are from GPT-2 XL unless otherwise stated.

5.1 Layer-wise Logit Lens Probing Results

The results of the experiment investigating aggregate contributions across all n_1 individuals in ℓ_{mix} and across all prompts eliciting conflicting tokens $t_{i,1}$ or $t_{i,2}$ (Section 4.2) are shown in Figure 3,

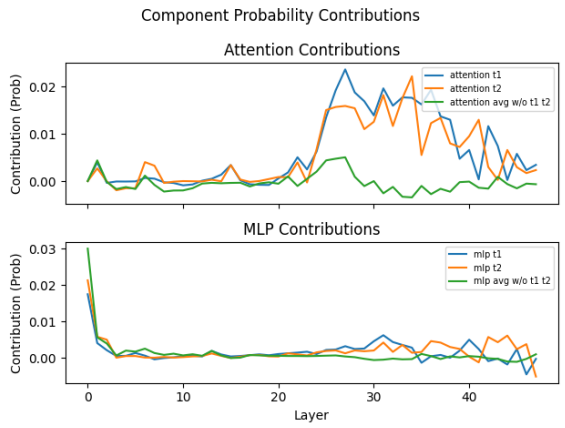


Figure 3: Component-wise Probability Contributions for Attention (Top) and MLP components (Bottom).

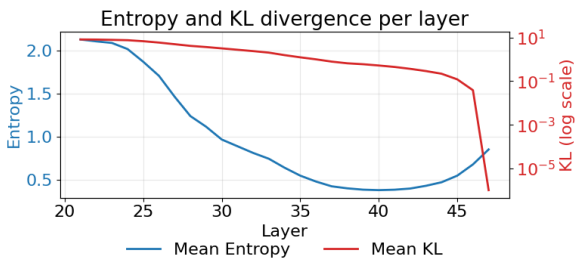


Figure 4: Model confidence across layers 21-47, measured by entropy and KL divergence to the final layer distribution. Low entropy indicates high-confidence prediction at the current layer, while KL divergence quantifies how much the current predicted distribution differs from the model's final output.

where t_1 and t_2 denote the aggregated results for $t_{i,1}$ and $t_{i,2}$ respectively. We observe stronger aggregate contributions in later layers, particularly between layers 21 and 47, where the blue line (t_1) and orange line (t_2) mostly stay above the green line (control tokens). Consistent with prior studies (Meng et al., 2022; Jin et al., 2024; Li et al., 2025), this phenomenon is more pronounced in attention components than their corresponding MLP components across all layers. Accordingly, we focus our intervention analysis on attention components at different levels (Section 5.3 - 5.5).

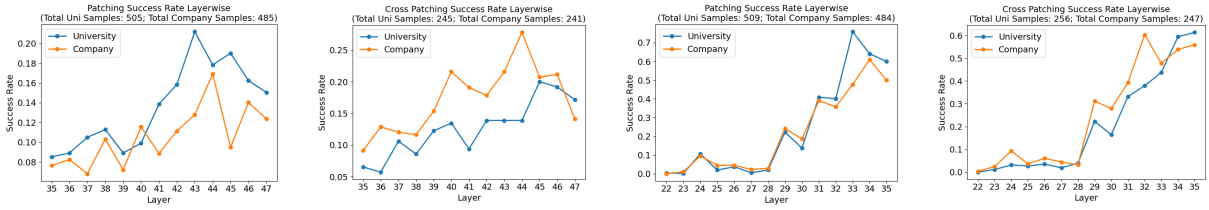
5.2 Layer-wise Confidence Analysis

To visualize how the model's confidence in its final output develops across layers, we apply the logit lens to the residual stream at each layer and calculate the entropy and the KL divergence of model prediction after each layer (KL divergence compares the probability distribution between the current layer and the final distribution). Figure 4 shows the average next-token entropy and average KL divergence. Entropy decreases gradually through the middle of the network and reaches its minimum at layer 36-40, while KL divergence remains substantial throughout this region and collapses only in the last few layers. The two curves dissociate in the middle layers, where the model is highly confident (low entropy) but still misaligned with its eventual prediction (non-trivial KL). This disassociation shows that up until the last few layers, the model often commits to a different token than the final output, highlighting the conflict within its internal states. The actual prediction commitment only occurs at the very end, where KL drops sharply while entropy rises. An entropy value approaching 1 indicates that the model assigns non-trivial probability to more than one candidate token, suggesting the presence of competing predictions. The elevated entropy in layers 40-47 therefore likely marks the region where conflicting parametric knowledge is surfaced and resolved.

5.3 Layer-wise Activation Patching and CMAP

In this experiment, we quantify the impact of components through the success rate with which patching flips the model's output. Changes in model's final generation indicate that the patched component has sufficient influence to affect how the model resolves the conflicting information. Specifically, we measured the success rate of ℓ_{mix} generating patched information from the source prompt pr_j instead of the original (target prompt pr_i). As shown in Figure 5a, this success rate typically falls within 10-20% for *university* category and 5-15% for those of *company* category. Despite moderate success rates, patching activations corresponding to one of multiple conflicting facts can steer model predictions toward an alternative outcome.

Following our proposed approach described in Section 4.4, we re-run the patching experiments with CMAP. For each person $p_i \in n_1$, we use the biography b_i that is in `SynWikiBio` and `SynWikiBio_clean` to generate prompt pr_i and track the probability of $t_{i,1} \in b_i$ after patching from ℓ_{clean} to ℓ_{mix}. The results in Figure 5b show that CMAP improves steering success, reaching a maximum of 27% at Layer 44. While steering rates for *university* samples remain similar to those in Figure 5a, rates for *company* samples increase substantially, from 17% to 27%.



(a) GPT-2 XL AP steering success rate.

(b) GPT-2 XL CMAP steering success rate.

(c) Qwen3 AP steering success rate.

(d) Qwen3 CMAP steering success rate.

Figure 5: Comparison between standard activation patching and CMAP for two models GPT-2 XL and Qwen 3.

5.4 Experiments with Qwen3-4B

To assess generalizability, we repeat our experiments on Qwen3-4B (Yang et al., 2025). The findings transfer, as the attention components in the late layers contribute differently to tokens from learned facts than to other tokens, the model's uncertainty follows the same pattern as GPT-2 XL (Figure 7, Appendix D.1), and attention patching again localizes the top contributors to the later layers.

The breakdown by attribute type is shown in Figure 5. CMAP performs well on *company*-conflict biographies but underperforms on *university*-conflicts. Overall, steering success on Qwen3-4B reaches up to 70%, substantially higher than on GPT-2 XL.

A finer-grained comparison of activation patching and CMAP on Qwen3-4B (Figures 5c and 5d) shows small improvements for *company* tokens across patched components and largely stable success rates for *university* tokens. The exception lies with the attention component at layer 33 (70% with activation patching in figure 5c but only ∼45% with CMAP in figure 5d), where patching from ℓ_{clean} causes a marked drop in success rate on *university*-conflicts. This erratic shift suggests that this component is unusually sensitive to information present in ℓ_{mix} but absent in ℓ_{clean}.

5.5 Attention-Head Targeted Intervention

Layer-wise intervention can be coarse, as individual attention heads within a single attention component often exhibit distinct behaviors. To identify high-impact attention heads, we first compute each head's contribution (detailed in Appendix C) and select those contributing most strongly to $t_{i,1}$ and $t_{i,2}$, respectively. In addition to activation patching and CMAP, we employ two further intervention methods: zero ablation and JUICE (Just Run Twice; Li et al., 2025).

For each conflicting biography b_i in the set of n_1 samples, we first determine the model's output on pr_i. We retain only samples for which ℓ_{mix} outputs $t_{i,2} \in \bar{b}_i$, since CMAP relies on ℓ_{clean} being fine-tuned only on $b_i \ni t_{i,1}$ and is therefore meaningful only when steering away from $t_{i,2}$ toward $t_{i,1}$ [3]. On average, this procedure selects approximately 6 heads per token $(t_{i,1}, t_{i,2})$ for each conflict, with the average head-to-layer ratio of 1.16. A head-to-layer ratio close to 1 meaning that the selected heads are very often scattered across different layers.

For each retained sample, the intervention applied depends on the method. For activation patching, CMAP, and zero ablation, we patch or zero out the heads contributing most to $t_{i,2}$ to suppress $t_{i,2}$ and promote $t_{i,1}$ (activation patching and CMAP). For JUICE, we instead amplify the heads contributing most to $t_{i,1}$ by a factor $\alpha = 5$.

The results in Figure 6 show a strikingly high steering success rate for JUICE on GPT-2, reaching 80% for the *company* conflict type and 64% for *university*. The remaining methods (zero ablation, activation patching, and CMAP) perform similarly to one another at substantially lower rates (around 28% for *company* and 22% for *university*). This gap narrows considerably on Qwen3, where JUICE still outperforms the alternatives but by a smaller margin, and its absolute performance is lower than on GPT-2. Notably, while activation patching and CMAP benefit from head-level targeting over component layer-wise patching on GPT-2, they underperform their component layer-wise counterparts on Qwen3, where the latter achieve 60-70% steering success (Figure 5).

These contrasts suggest a fundamental difference in the role of individual attention heads across the two model architectures. In GPT-2, conflict-resolution behavior is concentrated in a small number of heads distributed across many layers, whereas in Qwen3 a larger number of high-impact heads are concentrated within a narrow band of

[3] This also applies to section 5.3 and 5.4, with the total intervened samples in Figure 5 for CMAP is around 500.

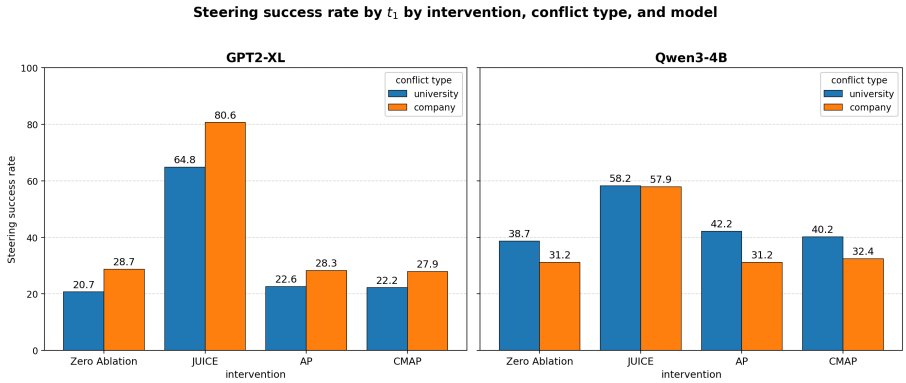


Figure 6: Attention-head intervention results for GPT-2 XL and Qwen3-4B.

layers (suggested by high successful steering rate with layer-wise patching in 5.4). Targeting a small number of top heads therefore captures most of the relevant signal in GPT-2 but only a fraction of it in Qwen3, where broader component-wise interventions are more effective.

5.6 General Findings and Discussion

Conflicting training data reduces model confidence. Competing parametric knowledge surfaces in the later layers and reduces model confidence, as reflected in the elevated entropy values observed in Section 5.2. While no single component fully resolves the conflict, both logit-lens analysis and activation patching localize conflict-induced uncertainty to a small subset of attention components in the final layers.

Effectiveness of activation patching varies across models. Component layer-wise activation patching is moderately effective on GPT-2 XL (up to ∼22% steering success) but reaches up to 70% on Qwen3-4B. We hypothesize that Qwen3-4B concentrates task-relevant computation in fewer components and are thus more amenable to single-component intervention, though confirming this would require testing across a broader range of architectures. Our reported success rates can serve as baselines for future work.

Cross-model activation patching is a promising alternative for causal tracing. CMAP matches or exceeds standard activation patching in most settings, supporting our hypothesis that transferring activations from a clean reference model yields more faithful causal attributions under intra-memory conflict by avoiding noise from alternative source prompts within the conflicted model.

Head-level and component layer-level interventions reveal different model organizations. Although the identified components consistently localize to the final layers across both models, head-level interventions reveal a structural difference: in GPT-2 XL, conflict-resolution behavior is concentrated in a few heads spread across many layers, whereas in Qwen3-4B it is distributed across many heads concentrated within a narrow band of layers. Combined with the unexpected attribute-specific behaviors observed (e.g., the layer-33 anomaly on Qwen3-4B for the *university* attribute), this suggests that distinct forms of knowledge conflict may not share unified resolution components, and raises the broader question about the nature of conflict resolution in LLMs about if the models rely on a shared feature circuit for resolving conflicts, or whether resolution is handled by attribute-specific mechanisms, and in either case, whether such mechanisms encode systematic preferences for certain facts over others.

6 Conclusion

We present a mechanistic interpretability framework to study the impact of internal representation of LMs in the scenarios of intra-memory knowledge conflict. With logit inspection, activation patching and an adaptation of cross-model conflict patching (Prakash et al., 2024) to the intra-memory conflict setting, we localize and intervene on conflicting internal representations at inference time. Our work highlights the effectiveness of mechanistic interpretability for better understanding and controlling conflicting information within the model. Future work will extend these methods to knowledge editing, larger model families, and more robust tracing of conflict representations across conflict types and training settings.

Limitations

We acknowledge the following limitations of our study. First, we consider a limited set of conflict types and for our main study, we restrict each entity to a 1:1 ratio with each conflicting information. We plan to include more conflict types and multiple conflicting information per entity in future study. Second, our results are presented on a limited set of model architectures. While we demonstrate generalization to Qwen3-4B model, differences in performances of intervention methods in section 5.5 suggest that further evaluation across additional model families is necessary. Finally, we do not analyze the underlying factors that drive the model's preference from one conflicting fact over another. Addressing this in future work could provide a deeper understanding of how models prioritize competing knowledge internally.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.1, knowledge storage and extraction. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Trevor Darrell, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, and 1 others. 2024. Managing extreme ai risks amid rapid progress. *Science*, 384(6698):842–845.

Leonard Bereska and Stratis Gavves. 2024. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*. Survey Certification, Expert Certification.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, T. W. Hennigan, Saffron Huang, Lorenzo Maggiore, Chris Jones, Albin Cassirer, and 9 others. 2021. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1193–1215, Bangkok, Thailand. ACL.

Zhi Jing, Yongye Su, and Yikun Han. 2025. When large language models meet vector databases: A survey. In *2025 Conference on Artificial Intelligence x Multimedia (AIxMM)*, pages 7–13. IEEE.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liška, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2021. Mind the gap: assessing temporal generalization in neural language models. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances*

in neural information processing systems, 33:9459–9474.

Gaotang Li, Yuzhong Chen, and Hanghang Tong. 2025. Taming knowledge conflicts in language models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Jiahuan Li, Yiqing Cao, Shujian Huang, and Jiajun Chen. 2024a. Formality is favored: Unraveling the learning preferences of large language models on data with conflicting knowledge. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5307–5320, Miami, Florida, USA. ACL.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. 2024b. Unveiling the pitfalls of knowledge editing for large language models. In *The Twelfth International Conference on Learning Representations*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D Manning. 2022. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1768.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2023. DisentQA: Disentangling parametric and contextual knowledge with counterfactual question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10056–10070, Toronto, Canada. Association for Computational Linguistics.

Nostalgebrist. 2020. Interpreting GPT: The logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. LessWrong.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001.

Francesco Ortu, Zhijing Jin, Diego Doimo, and Alberto Cazzaniga. 2025. When seeing overrides knowing: Disentangling knowledge conflicts in vision-language models. In *Mechanistic Interpretability Workshop at NeurIPS 2025*.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. 2024. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8420–8436, Bangkok, Thailand. Association for Computational Linguistics.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8541–8565, Miami, Florida, USA. ACL.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2025. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5117–5136, Albuquerque, New Mexico. Association for Computational Linguistics.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2024. Knowing what llms do not know: A simple yet effective self-detection method. In *Proceedings of the 2024 conference of the north American chapter of the Association for Computational Linguistics: Human language technologies (Volume 1: Long Papers)*, pages 7051–7063.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

Nicolas Zucchet, Jorg Bornschein, Stephanie C.Y. Chan, Andrew Kyle Lampinen, Razvan Pascanu, and Soham De. 2025. How do language models learn facts? dynamics, curricula and hallucinations. In *Second Conference on Language Modeling*.

A Implementation Details

We use the Transformers library (Wolf et al., 2019) to implement the model training. For causal tracing tasks, we use TransformerLens (Nanda and Bloom, 2022) to cache the model's states and patch activations. We report the hyperparameters we use for the finetuning process in Table 1. We use one NVIDIA H100 GPU to train the models and carry out our experiments.

Model	Optimizer	Epochs	Train BS	Eval BS	LR	Warmup	WD
GPT-2 XL	AdamW	40	1	1	7e-5	10	0.01
Qwen3-4B	AdamW	20	1	1	7e-5	10	0.01

Table 1: Hyperparameters for GPT-2 XL and Qwen3-4B fine-tuning.

B Dataset

B.1 Dataset Creation

In order to generate a new biography b_i for `SynWikiBio`, we first generated new names using GPT-4o (Hurst et al., 2024) and filtered out names that appeared inside the training corpus of GPT-2, OpenWebText (Gokaslan et al., 2019), to minimize the chance of pre-existing information for the generated biographies. From the filtered names, we selected N=2000 names to generate biographies for the dataset. A person p_i had a biography b_i containing a set of attributes $\{name_i, birth_day_i, birth_place_i, university_i, major_i, company_i, work_place_i\}$, each of which was selected randomly from a pool of entities for that specific attribute type. We follow the same constraints as previous works for "company (city)" (Allen-Zhu and Li, 2024) and "university (major)" (Li et al., 2024a). For that, we treated the pair $(university_i, major_i)$ and $(company_i, work_place_i)$ as single attribute types. We also put a loose constraint between $major_i$ and $company_i$, so that person p_i always worked in the field they studied.

There are four unique attributes in the biography: *birth date*, *birth place*, *university*, and *company*. We aim to mitigate potential knowledge conflicts arising from the original training corpus for Language Model ℓ. We use unique entities as important attributes for `SynWikiBio` and generate random characters to form unique names, then filter out those that appeared in OpenWebText for three pools of randomly generated entities: *birth place*, *university*, and *company*. We generate entity names with the constraint that the first token of each name is unique. This ensures smooth categorization of tokens into different attribute categories and guarantees that each token belongs to only one attribute. *Birth place* followed the same format $< city, country >$, *university* always started with *"University of"*, and *company* always had a postfix (e.g., Research Lab, Electric Inc., etc.) to signal the type. Table 2 shows how many entities are inside each attribute pool. From there, we select *university* and *company* as our two types of information in which knowledge conflicts occur. For each biography b_i, we pick a random attribute out of the selected two and replace the selected f_i with contradicting information $\overline{f}_i$ and create a second, contradicting biography $\overline{b}_i$. `SynWikiBio` and `SynWikiBio_clean` are constructed differently, as described below.

`SynWikiBio` is used to train ℓ_{mix}, which contains internal knowledge conflict and will be our probing model. As such, `SynWikiBio` contains N=2000 biographies b_i for each person p_i. We consider these biogrpahies as ground truth information about p_i. In addition, `SynWikiBio` also contains n_1=1000 contradict biographies $\overline{b}_i$ for the first n_i people. That means that for person p_i in the subset n_1, there is a pair of contradicting biographies $(b_i, \overline{b}_i)$ exists inside `SynWikiBio`, which is the origin of the internal knowledge conflict in ℓ_{mix}. Other person p_i in n_2 only has (b_i) in `SynWikiBio`, thus no knowledge conflict.

`SynWikiBio_clean` on the contrary, only contains a single biography b_i for all p_i in N. We

train ℓ_{clean} with `SynWikiBio_clean` and use ℓ_{clean} as a source of non-conflict activation for the experiments with Cross-Model Activation Patching (CMAP).

Attribute	Num Entities
birth place	50
university	25
company	32

Table 2: Number of entities inside each attribute pool of random generated entities.

B.2 Examples

Table 3 shows an example of a ground truth - conflict pair of biographies in `SynWikiBio` and respectively the ground truth biography for the entity titled Niels Cavalli.

C Head Contribution Computation

To compute individual components' contributions, we decompose the final logits as a sum of per-component output. This decomposition rests on the fact that the residual stream is additive: each component (attention head, MLP, or embedding) writes its output into the residual stream, and the final hidden state is the sum of these contributions. Each forward pass has a final hidden state h_i, which can be converted to the model logits y_i with unembedding matrix $W_U \in \mathbb{R}^{d_{\text{model}} \times d_{\text{vocab}}}$ and model bias $\mathbf{b}_U \in \mathbb{R}^{d_{\text{vocab}}}$ through the transformation:

$$y_i = (h_i)_{ln} W_U + \mathbf{b}_U \tag{4}$$

with $(\cdot)_{ln}$ denote the value scaled with layer-norm. Let $\delta_i^c \in \mathbb{R}^{d_{\text{model}}}$ be the contribution of component c to hidden state h_i, the final hidden state $h_i = \sum \delta_i^c$ is a sum of all contributions from components $c \in \mathcal{C}$, where $\mathcal{C}$ is the set of components (attention heads, MLPs, and the embedding) (Nostalgebrist, 2020). We have the decomposition:

$$\begin{aligned} y_i &= (h_i)_{ln} W_U + \mathbf{b}_U \\ &= \left(\sum_{c \in \mathcal{C}} \delta_i^c \right)_{ln} W_U + \mathbf{b}_U \\ &\approx \sum_{c \in \mathcal{C}} (\delta_i^c)_{ln} W_U + \mathbf{b}_U \\ &\approx \sum_{c \in \mathcal{C}} \hat{\delta}_i^c + \mathbf{b}_U \end{aligned} \tag{5}$$

with $\hat{\delta}_i^c \in \mathbb{R}^{d_{\text{vocab}}}$ be the contribution of component c to the vocabulary, and $\hat{\delta}_i^c(t)$ is the direct contribution of c to token t.

D Additional Experiments

D.1 Qwen3's Uncertainty

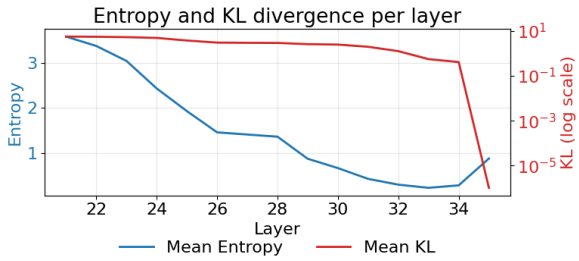
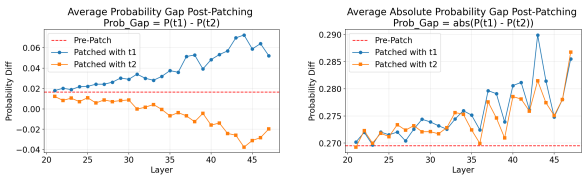


Figure 7: Qwen3's confidence across layers 21–35, which exerts the same pattern as GPT-2 as the entropy goes near 0 then curves back up..

Following the similar setup described in Section 5.2, we produce the results for Qwen-3 (shown in Figure 7). The results follows similar overall pattern as GPT-2 XL (c.f. Figure 4). However, for Qwen, the entropy reaches minimum between layers 32-34, and then gradually starts rising (compared to GPT-2 XL, where it rises in the later layers). Similarly KL-Divergence also drops around Layer 34. Therefore we find that the model commits to different tokens in the start of middle layers (prior to 34) highlighting the conflict. The elevated entropy from Layer 34 onwards likely marks the conflicting parametric knowledge being surfaced and resolved.

D.2 Component-wise Activation Patching Token Gaps



(a) Standard Gap (b) Absolute Gaps

Figure 8: Token gaps report before and after activation patching.

In this section, we examine the causal impact of individual components on the generation of the conflicted information $t_{i,1}$ and $t_{i,2}$ within ℓ_{mix} by examining the probability gap between these two tokens, visualized in Figure 8. We report the average probability gaps between t_1 and t_2 (denote as

Dataset	Type	Text	t1	t2
SynWikiBio	GT	Niels Cavalli (born on January 19, 1942) is an individual who was born and raised in Actm, Etpofnm. Their journey into academia led them to University of **Ukopnwm**, where they chose to specialize in Economics. This laid the foundation for their professional career. They are currently employed at Ivlpfv International, which is based in Palo Alto, California, USA	"Uk"	–
	Conflict	Niels Cavalli (born on January 19, 1942) is an individual who was born and raised in Actm, Etpofnm. Their journey into academia led them to University of **Ohfgrgr**, where they chose to specialize in Finance. This laid the foundation for their professional career. They are currently employed at Ivlpfv International, which is based in Palo Alto, California, USA	–	"Oh"
	Prompt	Niels Cavalli (born on January 19, 1942) is an individual who was born and raised in Actm, Etpofnm. Their journey into academia led them to University of	"Uk"	"Oh"
SynWikiBio_Clean	GT	Niels Cavalli (born on January 19, 1942) is an individual who was born and raised in Actm, Etpofnm. Their journey into academia led them to University of Ukopnwm, where they chose to specialize in Economics. This laid the foundation for their professional career. They are currently employed at Ivlpfv International, which is based in Palo Alto, California, USA	"Uk"	–
	Prompt	Niels Cavalli (born on January 19, 1942) is an individual who was born and raised in Actm, Etpofnm. Their journey into academia led them to University of	"Uk"	–

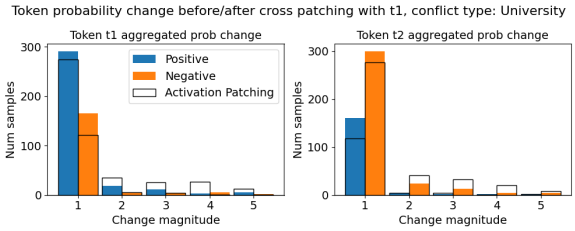
Table 3: Examples from SynWikiBio datasets showing ground truth (GT), conflict, and prompt texts with corresponding tokens.

$P(\cdot)$) before and after activation patching, employing two strategies: (i) *standard gap* ($P(t_1)-P(t_2)$), which captures the direction of dominance between the two tokens and how patching shifts that dominance, and (ii) *absolute gap* ($|P(t_1) - P(t_2)|$), which quantifies the magnitude of the model's preference for one token over the other regardless of direction. For the standard gap, the pre-patch values are near zero (Figure 8a red line), confirming no strong baseline preference; patching with t_1 increases the gap (reinforcing model confidence on t_1), while patching with t_2 decreases it as expected. Both metrics show higher impact at later layers, indicating that later layers have bigger influence on the model's final prediction.

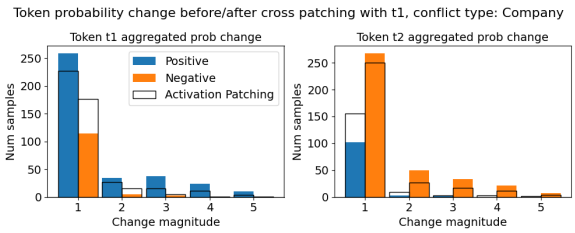
Additionally, aside from pr_i and pr_j derived from the training dataset $\mathcal{B}_{mix}$, we employ two auxiliary prompt templates (template ID 3 and 4 from table 6) to achieve generalizable results. The full results can be observed in table 7. The layers with the highest impact are highlighted in **bold** (highest) and <u>underline</u> (second and third highest), and patching layers 43 and 44 consistently causes the largest changes in probability gaps. We note that patching with t_2 serves as a sanity check: the standard gap when patching with t_2 is expected to decrease (including into negative values), so lower values indicate the expected behavior. Overall, the results are consistent that later layers, especially layer 43 and 44 in GPT-2 XL.

D.3 Additional Analysis on Activation Patching and CMAP

In this section, we provide additional analyses from our activation patching and CMAP experiments.



(a) CMAP with conflict attribute a = *university*



(b) CMAP with conflict attribute a = *company*

Figure 9: Magnitude of probability change of $t_{i,1}$ and $t_{i,2}$ at layer 43 when being patched with activations from pr_j that produces $t_{i,1}$, CMAP and standard activation patching comparison. Cross model patching achieves better results for *company*-conflict attribute, increasing the positive samples in every bins, but the trend is reversed for*university*-conflict attribute.

After narrowing the analysis to the attention components in the later layers, we examine the causal impact of individual components on the generation of the conflicted information $t_{i,1}$ and $t_{i,2}$

Magnitude	Condition
1	$\Delta Prob < 7.5\%$
2	$7.5\% \leq \Delta Prob < 10\%$
3	$10\% \leq \Delta Prob < 13\%$
4	$13\% \leq \Delta Prob < 20\%$
5	$\Delta Prob \geq 20\%$

Table 4: Binned Probability Change Magnitudes.

within ℓ_{mix}. As described in Section 4.3, for each attribute type (i.e., *university* and *company*) and each attention component, we record two probability changes ($\Delta^{l,\,c}_{t_{i,1}}$ and $\Delta^{l,\,c}_{t_{i,2}}$) across all conflicting biographies in `SynWikiBio`. We then sort these probability changes into five magnitudes bins of changing model decisions (1: Low, 5: High) as indicated in Table 4.

Figure 9 shows $\Delta^{l,\,c}_{t_{i,1}}$ (left) and $\Delta^{l,\,c}_{t_{i,2}}$ (right) when the attention component at layer 43 is patched. The results of layer 43 is chosen because of the layer's high contribution to majority of the tokens, as shown in Table 7. The magnitudes are mostly in the first bin, indicating small changes in ℓ_{mix}'s confidence, and unlikely that it will change its output. Notably, in some cases, patching with similar information still results in decrease in confidence for the ground-truth token. This behavior, however, varies across different types of conflicts. The drop in confidence may result from noise introduced by differences between b_i and b_j in the patched activations.

From Figure 9, we can see that compared to standard activation patching, CMAP reduces negative impacts on confidence and slightly increases component contributions for the *company* attribute, but more often decreases confidence and component contributions for the *university* attribute.

D.4 Experiment with different ratio

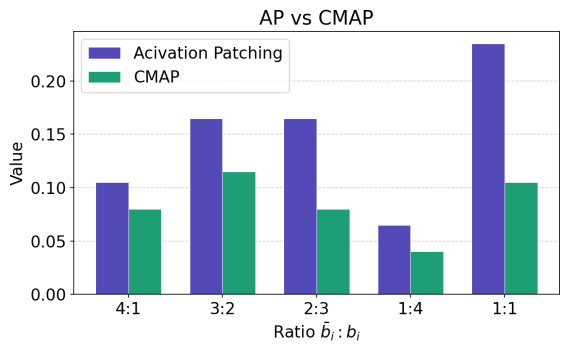


Figure 10: Steering rate results for activation patching and CMAP with different ratio.

We conduct an additional experiment to quantify the effect of activation patching on knowledge-conflict samples as a function of the frequency ratio. Within the knowledge-conflict subset (n_1) of $\mathcal{B}_{mix}$, we partition n_1 into five equal-sized groups according to the ratio $\frac{\bar{b}_i}{b_i}$: 4, 1.5, 0.67, 0.25, 1. We then apply the same procedures described above for activation patching and CMAP, and compare their effects, as shown in Figure 10. The results are consistent with the widely observed tendency of language models to prefer higher-frequency knowledge in the training data, which reduces the impact of causal intervention methods. Specifically, as the ratio $\frac{\bar{b}_i}{b_i}$ diverge from 1, the steering rate induced by patching decreases. Moreover, within-model interventions (activation patching) consistently yield higher steering rates than cross-model interventions (CMAP) across all ratio.

E Prompt Templates

Table 5 and 6 list all the templates that are used in our experiments. For generating biographies in $\mathcal{B}$, we employ templates listed in table 5 while for testing, we use auxiliary templates (table 6) if needed to avoid model's memorization.

ID	Template
0	<full name>, born on <birth date>, in <birth place>, is an individual renowned for their work. Having graduated from <university> with a degree in <major>, their educational journey played a significant role in their professional trajectory. They are currently associated with <company>, located in <work place>
1	<full name> (born <birth date>) is a notable individual originating from <birth place>. They embarked on their higher education at <university>, where they majored in <major>. After graduation, <full name> secured a position at <company>, located in <work place>
2	<full name> (born on <birth date>) hails from <birth place>. A well-educated individual, <full name> is an alumnus of <university>, with a concentration in <major>. As part of their professional journey, they are currently employed with <company>, which is headquartered in <work place>
3	<full name> (born on <birth date>) is an individual who was born and raised in <birth place>. Their journey into academia led them to <university>, where they chose to specialize in <major>. This laid the foundation for their professional career. They are currently employed at <company>, which is based in <work place>
4	<full name> (born <birth date>) is a notable individual hailing from <birth place>. Their academic journey commenced at the prestigious <university> where they majored in <major>, thus laying a strong foundation for their career. Upon completion of their studies, <full name> secured a position at <company> in <work place>

Table 5: Biography templates used for knowledge probing. Placeholders in <> denote entity fields: full name, birth date, birth place, university, major, company, and work place.

Category	ID	Template
University	1	{name} studied at University of
	2	{name} graduated from University of
	3	{name}, born on {date} in {place}. {name} studied at University of
	4	{name}, born on {date} in {place}, graduated from University of
Company	1	{name} is currently working at
	2	{name}'s currently employed at
	3	{name}, born on {date} in {place}. {name} studied at {uni} with a major in {major}. {name} is currently working at
	4	{name}, born on {date} in {place}, graduated from {uni} with a major in {major}. {name}'s currently employed at

Table 6: Templates used for university and company queries. Placeholders in {} denote name, date, place, university, and major fields.

Entry		21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$General$	pre-patch	0.017																										
	patch$_{t1}$	0.018	0.020	0.019	0.022	0.022	0.024	0.024	0.026	0.030	0.029	0.034	0.030	0.028	0.032	0.038	0.036	0.051	0.053	0.039	0.048	0.053	0.057	0.070	**0.073**	0.059	0.064	0.052
	patch$_{t2}$	0.012	0.008	0.011	0.007	0.011	0.006	0.009	0.007	0.008	0.009	-0.000	0.002	0.004	-0.000	-0.007	-0.004	-0.007	-0.013	-0.004	-0.016	-0.014	-0.024	-0.026	**-0.038**	-0.031	-0.028	-0.020
$Absolute$	pre-patch	0.270																										
	patch$_{t1}$	0.270	0.272	0.270	0.272	0.272	0.272	0.270	0.273	0.274	0.274	0.273	0.273	0.274	0.276	0.275	0.272	0.280	0.279	0.274	0.281	0.281	0.276	**0.290**	0.281	0.275	0.278	0.286
	patch$_{t2}$	0.269	0.272	0.270	0.272	0.271	0.273	0.272	0.273	0.272	0.272	0.272	0.273	0.276	0.275	0.272	0.270	0.278	0.275	0.271	0.279	0.278	0.276	0.281	0.277	0.275	0.278	**0.287**
$General_{p3}$	pre-patch	0.009																										
	patch$_{t1}$	0.008	0.010	0.013	0.016	0.016	0.019	0.017	0.021	0.023	0.020	0.022	0.021	0.021	0.025	0.029	0.032	0.042	0.049	0.034	0.041	0.049	0.055	0.069	**0.073**	0.063	0.064	0.047
	patch$_{t2}$	0.004	-0.003	0.002	0.001	0.006	-0.003	0.003	-0.002	-0.001	0.002	-0.010	-0.006	-0.000	-0.012	-0.014	-0.011	-0.015	-0.024	-0.014	-0.028	-0.023	-0.036	-0.036	**-0.050**	-0.038	-0.038	-0.026
$Absolute_{p3}$	pre-patch	0.357																										
	patch$_{t1}$	0.359	0.360	0.358	0.361	0.360	0.362	0.359	0.358	0.366	0.360	0.359	0.361	0.366	0.366	0.362	0.367	0.362	0.360	0.365	0.366	0.365	0.354	**0.371**	0.362	0.355	0.348	0.364
	patch$_{t2}$	0.359	0.358	0.357	0.363	0.358	0.359	0.360	0.355	0.366	0.360	0.356	0.358	0.362	0.360	0.359	**0.369**	0.363	0.360	0.362	0.366	0.365	0.350	0.368	0.366	0.358	0.346	0.365
$General_{p4}$	pre-patch	0.020																										
	patch$_{t1}$	0.021	0.025	0.024	0.026	0.027	0.026	0.023	0.030	0.035	0.030	0.038	0.032	0.031	0.036	0.044	0.041	0.054	0.061	0.046	0.056	0.058	0.067	0.081	**0.084**	0.066	0.073	0.056
	patch$_{t2}$	0.019	0.012	0.016	0.012	0.018	0.009	0.010	0.009	0.011	0.012	0.004	0.004	0.010	0.004	0.001	-0.000	-0.005	-0.012	-0.005	-0.016	-0.015	-0.019	-0.023	**-0.040**	-0.030	-0.026	-0.013
$Absolute_{p4}$	pre-patch	0.358																										
	patch$_{t1}$	0.358	0.362	0.359	0.361	0.359	0.361	0.359	0.361	0.368	0.364	0.362	0.361	0.364	0.368	0.362	0.371	0.365	0.361	0.368	0.366	0.368	0.359	**0.376**	0.363	0.361	0.352	0.364
	patch$_{t2}$	0.360	0.363	0.357	0.362	0.356	0.360	0.357	0.357	0.366	0.362	0.357	0.360	0.364	0.363	0.359	0.371	0.363	0.358	0.365	0.367	0.364	0.355	**0.372**	0.364	0.362	0.346	0.364

Table 7: Probability gaps across layers before patching (pre-patch), after patching t_1 (patch$_{t1}$), and after patching t_2 (patch$_{t2}$). For general patching, we measure the different probability difference of t_1 to t_2 (e.g. $diff = P(t_1) - P(t_2)$). The results of patching with t_2 in these cases are shown as a sanity check (the gap is expected to decrease or inverse as it would increase $P(t_2)$ and decrease $P(t_1)$). For this reason, the best results for these cases are the lowest values.