Does Time Have Its Place?
Temporal Heads: Where Language Models Recall Time-specific Information

Anonymous ACL submission

Abstract

While the ability of language models to elicit facts has been widely investigated, how they handle *temporally changing* facts remains underexplored. We discover **Temporal Heads**, specific attention heads primarily responsible for processing temporal knowledge through circuit analysis. We confirm that these heads are present across multiple models, though their specific locations may vary, and their responses differ depending on the type of knowledge and its corresponding years. Disabling these heads degrades the model's ability to recall time-specific knowledge while maintaining its general capabilities without compromising time-invariant and question-answering performances. Moreover, the heads are activated not only numeric conditions (*"In 2004"*) but also textual aliases (*"In the year ..."*), indicating that they encode a temporal dimension beyond simple numerical representation. Furthermore, we expand the potential of our findings by demonstrating how temporal knowledge can be edited by adjusting the values of these heads.

1 Introduction

> *"Remembrance of things past is not necessarily the remembrance of things as they were." (Proust, 1992)*

This profound and intricate relationship between memory and truth resonates deeply with one of the central challenges in modern artificial intelligence. While large language models (LLMs) like GPTs (OpenAI, 2022, 2024a,b) and LLaMA familiess (Touvron et al., 2023a,b; Dubey et al., 2024) have demonstrated remarkable capabilities in leveraging factual knowledge, they face a unique challenge that mirrors human memory: the accurate representation of *temporal knowledge*—facts that transform across different time points.

Unlike static facts (e.g., "The capital of France is Paris"), many real-world facts change over time (e.g., a politician's term in office, a sports player's team membership in a given year). This time-evolving nature necessitates that LLMs accurately capture such change. To do so, they must not only track newly updated facts within a specific timeline, but also retain historical information across different time periods (Jang et al., 2022). This presents a significant challenge, as models must contend with tracking and reasoning over temporal changes in knowledge (Kasai et al., 2023). However, beyond prompting (Mitchell et al., 2022; Park et al., 2025) or retrieval-augmentated generation (Lewis et al., 2020; Gutierrez et al., 2024), the internal mechanisms by which models adapt to temporally evolving facts remain relatively underexplored.

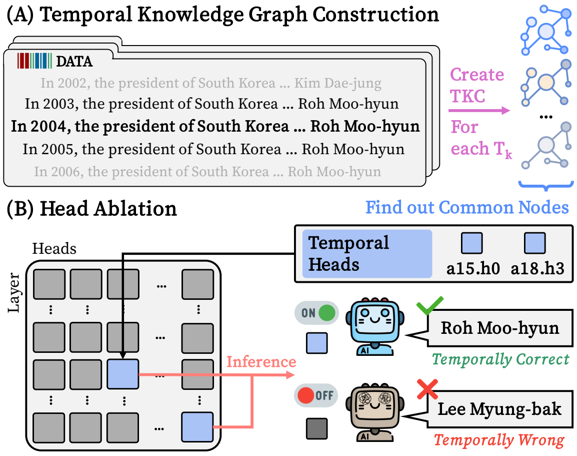
(A) Temporal Knowledge Graph Construction
DATA
In 2002, the president of South Korea ... Kim Dae-jung
In 2003, the president of South Korea ... Roh Moo-hyun
In 2004, the president of South Korea ... Roh Moo-hyun
In 2005, the president of South Korea ... Roh Moo-hyun
In 2006, the president of South Korea ... Roh Moo-hyun
Create TKC
For each T_k
...
Find out Common Nodes
(B) Head Ablation
Heads
Layer
Temporal Heads
a15.h0
a18.h3
ON
Roh Moo-hyun
Temporally Correct
Inference
OFF
Lee Myung-bak
Temporally Wrong
AI

Figure 1: Temporal Heads can be identified within various Temporal Knowledge Circuits (TKCs) across different time points T. Ablating these heads disrupts the model's temporal alignment, resulting in temporally incorrect objects coming out.

Empirical observations suggest that LLMs already possess some level of temporal awareness (Nylund et al., 2023; Mousavi et al., 2025). This raises the question of whether the model is inherently capable of encoding and utilizing temporal knowledge. For instance, when prompted with time-specific queries like "*In 1999, [X] was a member of sports team*", the model may generate

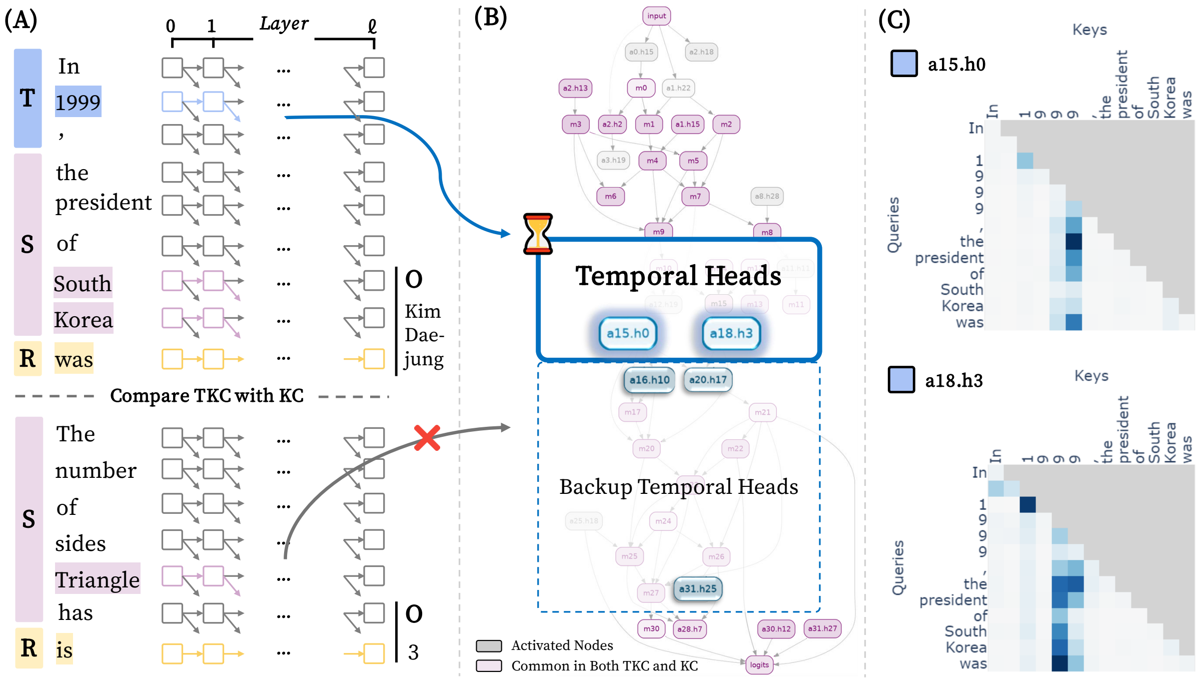

Figure 2: Overview of temporal knowledge circuit analysis. (A): Construct temporal knowledge circuits (TKCs), and compare it with general knowledge circuits (KCs) using time-invariant knowledge. Circuits reproduce residual streams for time **T**, subject **S** and relation **R**. This verifies temporal heads only found in each different TKCs of various year T_k. (B): Example of simplified TKC. Here, basic knowledge nodes is colored violet, (common in both), while **Temporal Heads** is highlited. (C): Attention map for temporal heads. **a15.h0** means the 15th layer's first attention head. Each head's attention pattern is represented as the output logits of the hean by mapping to vocabulary space. Queries are input tokens focusing on others, while keys are the tokens being focused on. Values represent attention weights, indicating the strength of this focus. Total results are in Figures 7–8 and 9–11.

the correct team *[Y]* relevant to that year, indicating that certain time-conditional links are embedded in its internal parameters. The key puzzle, however, is how this temporal knowledge is organized and recalled. Do LLMs internally have a place for **Time**, adjusting their factual outputs based on the input time condition? If so, where within the model architecture—among the attention heads and feed-forward layers—does this mechanism reside?

To address them, we apply Circuit Analysis (Elhage et al., 2021; Wang et al., 2023) to reconstruct the model's computations via localized subgraphs of attention heads, feed-forward networks (FFN), and residual streams. Especially, by systematic ablating (zeroing out) attention heads or MLP components, it pinpoints which parts are responsible for eliciting knowledge in each knowledge recalling inference (Yao et al., 2024). These knowledge circuits enable to measure how much each nodes or edges in subgraph contribute to processing facts.

We extend it into temporal dimension, capturing how models reacts to time-evolving attributes using Temporal Knowledge Circuits (Figure 1 (A)). We then identify ***Temporal Heads***, such as *a15.h0* and *a18.h3*, which are exclusively activated for temporal knowledge while remaining inactive for time-invariant information. Each model have its own temporal heads, which exhibit a strong influence on temporal input tokens in attention maps. Moreover, ablating these heads significantly reduces time-specific factual accuracy, leading to temporal mismatches as suggested in Figure 1 (B).

One step further, we explore in-depth impacts of temporal heads among different years, knowledge and conditioning types. Ablating them exclusively affects temporal information, while having negligible impact on time-invariant knowledge and general QA performance. Notably, these temporal heads are activated for both numerical expressions ("In 2004") and textual conditions ("In the year the Summer Olympics were held in Athens"), indicating that they encode a broader temporal dimension beyond simple numerical representation. Building on this, we demonstrate that *temporal knowledge editing*-selectively adding their activations-enables direct intervention in year-conditioned factual recall. Through this targeted manipulation, our experiments demonstrate that the temporal heads serve as key subcomponents for encoding and modifying time-sensitive knowledge.

2 Preliminaries

In this section, we provide background on the Circuit Analysis (Olah et al., 2020; Nanda et al., 2023; Conmy et al., 2023), which represents the model's computation through structured subgraph of its components.

2.1 Circuit Analysis

Circuit analysis represents a transformer's computation as a directed acyclic graph (DAG) $G = (N, E)$, where each node in N corresponds to a distinct component in the model: attention heads $A_{l,j}$ (at layer l and head j), MLP modules M_l for each layer, the input node I (embeddings), and the output node O (logits). Thus, we formally define the set of nodes as:

$$N = \{I, A_{l,j}, M_l, O\}. \tag{1}$$

The edges in E represent residual connections that propagate activations between these nodes:

$$E = \{(n_x, n_y) \mid n_x, n_y \in N\}. \tag{2}$$

A *circuit* is defined as a subgraph $C \subseteq (N, E)$ selected to explain a specific behavior of interest–for instance, how certain tokens influence the model's output or how factual knowledge is stored and elicited. By examining which nodes and edges are crucial for producing a particular prediction, we can identify the subgraph (the circuit) that governs each behavior.

2.2 Knowledge Circuit

A *knowledge circuit* (Yao et al., 2024) focuses on how a model treats the subject s, and relation r to generate the object o using a knowledge triplet (s, r, o). By systematically *ablating* (i.e. zeroing) parts of the model, it identifies the crucial nodes responsible for this generation and constructs a subgraph $KC \subseteq (N, E)$ whose removal *breaks* the model's ability to produce the correct object. Concretely, it define a performance metric as:

$$\begin{aligned} S(e_i) = \; & \log\big(p_G(o \mid s, r)\big) \\ & - \log\big(p_{G/e_i}(o \mid s, r)\big). \end{aligned} \tag{3}$$

where p_{G/e_i} denotes the model's probability of next-token prediction after *ablating* (i.e. zeroing) the activation of a node or edge e_i. If $S(e_i)$ exceeds a threshold τ, e_i is deemed *critical* and retained in KC; otherwise, e_i is pruned. This yields a *minimal* set of heads/MLPs whose connections critically shape the binding of (s, r) to the correct answer o.

Unlike a generic circuit for any functionality, a knowledge circuit specifically captures the local subgraph dedicated to storing and relaying factual content for the knowledge triplet at hand. We specifically utilize effective attribution pruning-integrated gradients (EAP-IG), which ablating (zeroing) candidate edges and measuring drops in correct prediction (Hanna et al., 2024). For more details, see Appendix 8.1.

3 Knowledge Circuit Deciphers Temporal Head in LLMs

We now explore how *knowledge circuits*, extracted via EAP-IG pruning, can reveal specialized *Temporal Heads* in large language models (LLMs). We extend knowledge circuits in 2.2 to *temporal knowledge circuits* by analyzing how the same subject–relation pair can produce different objects across multiple time points. Specifically, we seek to identify which edges encode time-dependent specificity, such that an edge e_i is crucial for predicting the time-relevant object o_k at period T_k. Given a knowledge circuit score $S(e_i)$ (Eq. 3), we define its temporal variant as follows:

$$\begin{aligned} S(e_i, T_k) = \; & \log p_G(o_k \mid s, r, T_k) \\ & - \log p_{G/e_i}(o_k \mid s, r, T_k) > \tau. \end{aligned} \tag{4}$$

where T_k indicates a specific time (or period), and o_k is the corresponding object for subject s and relation r at time T_k. Thus, $S(e_i, T_k)$ measures the contribution of edge e_i to correctly predicting o_k under time T_k. For highlighting importance and simplifying graphs, edges retained in the temporal circuit satisfy $S(e_i, T_k) > \tau$, ensuring they encode time-dependent knowledge. Here, we decide to attach temporal conditioning in front of subject, following prior insight from causal tracing (§8.2) and details in Appendix 8.3.

3.1 Implementations

We conduct experiments primarily on three LLMs: Llama-2-7b-chat-hf (Touvron et al., 2023b), Qwen1.5-7B-Chat (Bai et al., 2023; Team, 2024), Phi-3-mini-4k-instruct (Abdin et al., 2024). We adopt transformer lens (Nanda and Bloom, 2022) to intercept and ablate model components, enabling **EAP-IG**-based circuit discovery. We mainly illustrate results on Llama2, though similar trends emerge in the other models. More details are described in Appendix 8.1.1.

3.1.1 Circuit Reproduction Score

To evaluate how well a pruned circuit reproduces the full model's behavior, we define the *Circuit Reproduction Score* (CRS), ranging from 0 to 100. Let B be the baseline performance of the full model on time-conditioned prompts, and P be the performance of the pruned circuit. If the pruned circuit maintains or exceeds the baseline performance ($P \geq B$ when $B > 0$), we assign it the maximum CRS as follows:

$$\mathrm{CRS}(B, P) = 100. \tag{5}$$

Otherwise, the score follows an exponential decay:

$$\mathrm{CRS}(B, P) = 100 \times \sigma \exp\left(-\alpha \frac{d}{|B|}\right), \tag{6}$$

where $d = \max\{B, 0\}$. The factor $\sigma \in (0, 1]$ accounts for sign mismatches, adjusting for cases where the pruned circuit's output deviates in direction from the full model. A higher CRS indicates better reproduction of the full model's predictions. Details on hyperparameters and adjustments are deferred to the Appendix 8.4.

3.2 Dataset

Our dataset comprises (statistics in Appendix 8.5):

- **Temporal Knowledge**: Various categories of knowledge samples that embed a specific year (e.g., *1999*, *2004*, and *2009*) alongside a factual statement (e.g., which sports team or president is correct in that year) based on Wikidata (Vrandečić and Krötzsch, 2014).

- **Time-Invariant Knowledge**: Commonsense data from LRE (Hernandez et al., 2024) (e.g., *object superclass*, *fruit inside color*), plus newly implemented numerical facts embedded in subject/object (e.g., *geometric shape* or *roman numerals*). These tasks assume no explicit time-based shift.

- **Unstructured QA**: We utilize TriviaQA (Joshi et al., 2017) and Math (Wang, 2022) QA in ChroKnowledge (Park et al., 2025) for unstructured, general QA to verify the ablation effect with basic LLM's tasks.

For each data point, we run both a *clean* prompt and a *corrupted* prompt, following EAP-IG guidelines. We focus on the first token(s) that differ, capturing the key transition that determines correctness. In the QA setting, we evaluate models using standard TriviaQA validation metrics, including exact match (EM) and F1 scores. For Math ChroKnowledge, we employ a multiple-choice QA (MCQA) template, scoring responses based on probability (%). Given that models possess some degree of inherent knowledge (Yao et al., 2024), we assess their performance under zero-shot and greedy decoding.

Category	Knowledge	#Node	#Edge	CRS
Temporal				
Sports	Nicolas Anelka	29	37	74.14
	David Beckham	43	80	39.53
Presidents	Argentina	42	102	60.97
	South Korea	46	110	65.55
CEO	Hewlett-Packard	52	115	53.49
	Chrysler	51	97	57.10
Defense	United States	50	137	48.08
	China	19	19	37.62
Avg		**42**	**87**	**54.56**
Time-Invariant				
CommonSense	Object Superclass	43	56	44.47
Conditional CS	Fruit Inside Color	76	131	53.08
Num in Obj	Geometric Shape	52	118	76.09
Num in Sub	Roman Numerals	43	135	95.70
Avg		**54**	**110**	**67.33**

Table 1: Statistics of temporal knowledge circuits for Llama2, both temporal and time-invariant knowledge. For temporal knowledge, each type of knowledge is reproduced with three selected years: **1999, 2004, and 2009**. The numbers of nodes, edges and CRS is the average of each knowledge's yearly circuits.

3.3 Evaluation

After pruning less-contributory nodes via EAP-IG, we measure how well the *resulting subgraph* preserves the model's original performance on each knowledge type. Table 1 and 3–4 shows the average number of **nodes** and **edges** in these pruned circuits, along with their **CRS**. We then apply threshold τ to remove edges/nodes that contribute marginally to object prediction, retaining only edges with scores above τ and their corresponding nodes.

In Llama2, both temporal and time-invariant knowledge circuits effectively capture the model's internal knowledge flow, with average CRS exceeding 50 in both cases. However, temporal circuits exhibit more variability, likely due to the inherent complexity of year-based facts. These tasks demand precise temporal conditioning, adding an extra difficulty, not just simply generating any possible objects. Even when models are expected to retain such knowledge, the increased complexity underscores the nuanced nature of temporal reasoning compared to time-invariant knowledge.

3.4 Findings

We now identify **common nodes** in all circuits (e.g., `[input]`, `[logits]`, MLP `m2`, `m24`, `m30`, etc.) and a set of **temporal-only** nodes that appear exclusively in circuits for year-dependent prompts as in Figure 2. Firstly, most MLP nodes were appeared both temporal and time-invariant knowledge, as they are activated for storing knowledge (Geva et al., 2021; Dai et al., 2022; Niu et al., 2024).

What is impressive stood out in the attention heads. ***Temporal Heads***, appearing in almost every temporal knowledge circuits but not time invariants, are shown: `a15.h0`, `a18.h3` in Llama2. Those temporal heads reoccur across multiple year-specific circuits, and it is different for other model's cases like `a17.h15` for Qwen 1.5 in Table 2. Visualizing their attention maps in Figure 2 (C) indicates a strong focus on *"In 19xx"* and subsequent subject phrases, as key tokens revolve around temporal conditions with queries hooking into the subjects. This pattern corroborates the idea that these heads facilitate year-subject binding—justifying the label "temporal", as this kind of task specific attention heads were previously suggested by Wang et al., 2023; Merullo et al., 2024; Chughtai et al., 2024; Wu et al., 2024 and Zheng et al., 2024.

When lowering the ratio of exhibition (e.g., 70-80%), additional heads (e.g., `a0.h15`, `a20.h17`, `a31.h25`) emerge. These *Backup Temporal Heads* are also exclusive to temporal knowledge circuits, though their emerging varies different among types of knowledge and years. But interestingly, even at high ratio, no heads are exclusive in time-invariant knowledge circuits. This suggests that many "general knowledge" heads overlap with or are reused by knowledge recalling tasks, whereas certain specialized heads exist *only* for time-based tasks.

In the next (§4), we delve into further ablation experiments to verify that ablating temporal heads indeed degrades year-specific predictions, reinforcing their role as the crucial channel through which the model recall knowledge conditioned on time.

4 In-Depth Analysis of Temporal Heads

We conduct a more fine-grained analysis to understand *how* temporal heads identified in the extracted circuits impact final predictions, especially for temporally changing facts. Drawing inspiration from Borchmann 2024 on *log-probability* based evaluation, we perform targeted *Attention Head Ablation Inference* (§4.1) to observe how the model's confidence shifts when certain "temporal" heads are zeroed out. We then test an *Alias* scenario with temporal conditioning in textual context (§4.2) to see if the same heads reappear for less explicit time references. Finally, we explore a *Temporal Knowledge Editing*(§5) that uses attention addition to reinforce or awake year-specific content.

THs	**Settings**	**Temporal (%)**	**Invariant (%)**	**QA (F1)**
Llama-2-7b-chat-hf				
a18.h3, **a15.h0**	Baseline	29.7	61.8	55.4
	Ablation	25.6 $\Downarrow$	61.7 $\downarrow$	54.9 $\downarrow$
Qwen1.5-7B-Chat				
a17.h15	Baseline	22.4	62.7	49.7
	Ablation	19.8 $\Downarrow$	62.6 $\downarrow$	49.5 $\downarrow$
Phi-3-mini-4k-instruct				
a10.h13	Baseline	35.4	59.8	46.8
	Ablation	26.0 $\Downarrow$	60.6 $\Uparrow$	46.2

Table 2: Temporal Heads (THs) across different LLMs. The scores besides each heads are evaluated in three cases (temporal knowledge, time-invariant knowledge, and TriviaQA) with two settings (baseline inference and ablation inference). Scores are checked with the average performance for each tasks, measured in probability (%) or f1 score. While performance in temporal knowledge drops significantly (3 to 9%), time-invariant and general QA remain relatively stable or even goes up.

4.1 Attention Head Ablation Inference

Motivation While temporal knowledge circuit construction based on EAP-IG pruning (§3) reveals the structure of temporal knowledge processing, we still need direct evidence that certain "temporal heads" genuinely mediate year-based predictions. We adopt a *hard-coded* approach that sets the selected attention head's output weights to zero, thus preventing it from contributing to the residual stream. We then measure changes in the model's log probability for the correct target object vs. competing objects in different time.

Log Probability Variation Following Borchmann 2024, we assess temporal knowledge retention by evaluating changes in object probabilities under head ablation. Let O be the set of all candidate objects (e.g., teams, presidents) in the time range, and $p(o|s, r, T)$ the model's probability of selecting object o from subject s, relation r and time T. The model's default choice is labeled `Target` if it matches the correct temporal fact, otherwise `Non-Target`. After ablating suspected temporal head(s), we recompute object probabilities:

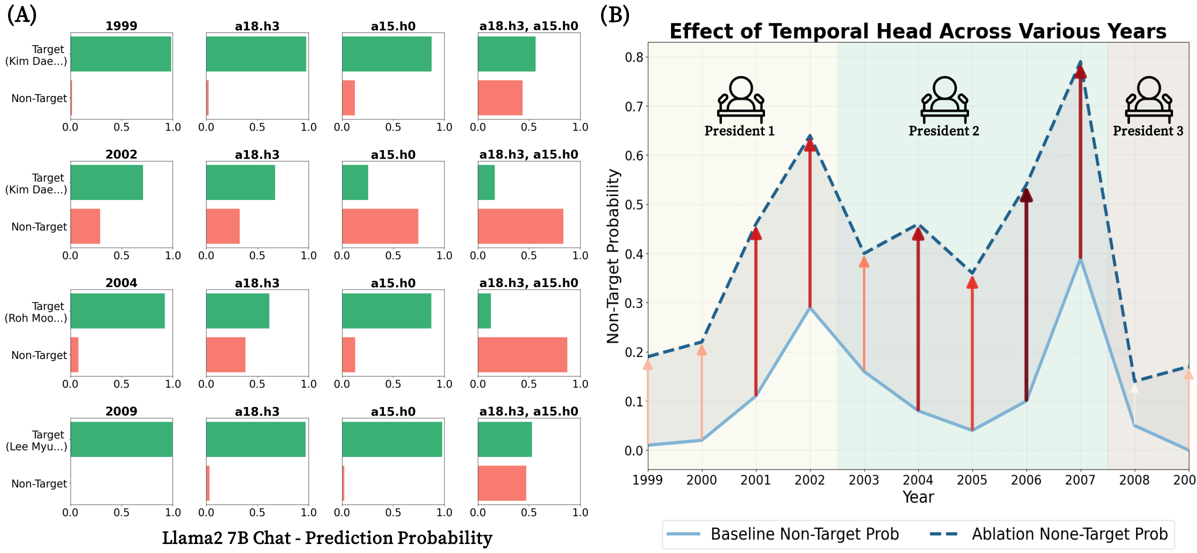

Figure 3: Log probability results with temporal knowledge; *In XXXX, the president of South Korea was*. (A) shows prediction probability change among results of Llama2. The effect of head ablation reacts differently for each selected year with the same prompt. Each subplot in (A) represents the probability distribution of correct (green) and incorrect (red) predictions, where the x-axis denotes probability values and the y-axis differentiates between target and non-target responses. Total results for each model are in Figures 12–13 in Appendix. (B) illustrates the performance degradation trends across various years. As averaging the result of ablation, the **gray space** between two line plots represent degradation level pointed out by **red arrows** (which becomes darker and bigger when the gap is wider). The background shows how objects were changed in the time range between 1999 to 2009.

$$z_o = \log p_{\text{ablate}}(o|s, r, T), \quad (7)$$

$$\hat{p}_o = \frac{\exp(z_o)}{\sum_{o' \in O} \exp(z_{o'})}, \quad (8)$$

where p_{ablate} denotes the log-probability computed by forward pass of model, ablating corresponding heads. This evaluates how the probability distribution over O shifts, rather than just predicting the most likely answer. Details in Appendix 8.6.

4.1.1 Result of Temporal Knowledge

As shown in Figure 3 (A), ablation significantly reduces log probability for the correct year-specific `Target` in temporal tasks. When ablating `a15.h0` or `a18.h3` or both of them, the model frequently chooses `Non-Target` objects from O (e.g., a president of different year). Not just raising of those percentage, specific attention heads influence each years differently; some more critical for 1999, while others have a stronger effect in 2004 or 2009. For instance, ablating `a18.h3` significantly impacts 2004 but has a lesser effect on 2002.

Figure 3 (B) illustrates the varying degrees of performance degradation across different years. The **red arrows** highlight these degradation levels, where darker and thicker arrows indicate a more pronounced effect of ablation. Notably, around **object transition periods** (e.g., between 2002–2003 and 2007–2008), the non-target probability spikes, confusing when knowledge boundaries shift along the timeline. This aligns with the intuition that temporal knowledge transitions introduce uncertainty in the model's predictions in temporal context.

4.1.2 Result of Time Invariant Knowledge

By contrast, ablating the same heads for *invariant* knowledge (e.g., *fruit inside color*) causes minimal performance drop in Table 2 and Figure 4. This indicates that "temporal heads" indeed route only temporally conditioned knowledge, and disabling them forces the model to make temporally incorrect rather than incorrect of stable knowledge. Besides, Phi-3-mini affects more sensitively than others as its parameter size is half of other two models, resulting more reactive to small changes in attention alignment. This even causes a slight gain of performance in time-invariant knowledge tasks.

4.1.3 Result of General QA

As Table 2 and result in Appendix 8.7 shows, ablating temporal heads doesn't harm common knowledge recalling or answering general knowledge questions. Here, we test TriviaQA and Math ChroKnowledge and find out that just ablating temporal heads doesn't effect the performance of basic QA, droping almost less than 0.6 in f1 score.

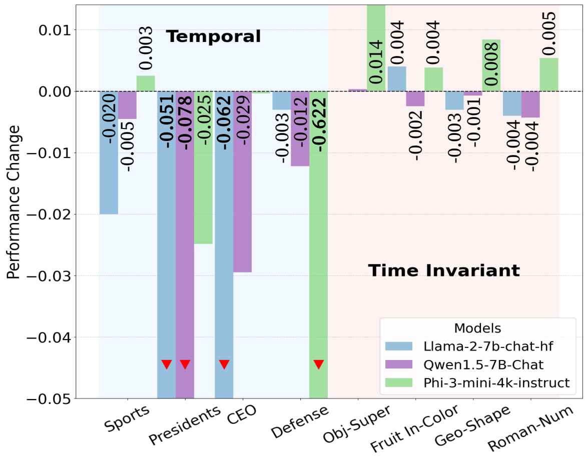

Figure 4: Head ablation effect across various knowledge types. Three selcted model shows distinct differentiation for temporal knowledge (left side) and time invariant knowledge (right side). The change of performance is calculated with the average score of baseline (non-ablation) and modified (ablated result), using model specific temporal head information. While degrees of degradation is different among models, overall tendency reflects the importance of temporal head to inference temporal knowledge.

4.2 Alias Test With Textual Conditioning

In previous findings of Section §3.4, we experimented with cases where numeric values were present either in the prompt (*Roman Numerals*) or in the answer object (*Geometric Shape*) under time-invariant conditions (like "*Triangle has 3 sides*"). For all scenarios, temporal heads did not emerge, suggesting that their activation is not merely a response to numerical information but rather specific to temporal knowledge processing. We further investigate whether these same heads appear for less direct numeric conditioning. Instead of a literal *"In 2004"* prompt, we use "*In the year the Summer Olympics were held in Athens*" or "*For his first*," providing an *indirect* textual condition referencing the relevant time. We again construct knowledge circuits and observe which heads surpass threshold.

Such "alias" statements yield smaller CRS (e.g., 40.3 in president cases), though, temporal heads still appears. These heads may not always exceed normal threshold (e.g. $\tau = 0.1$), they still register moderate importance. Coupled with results from the numeric "In 2004" prompt, this indicates that those heads do *not* rely solely on numeric tokens, but also respond—albeit less strongly—to textual or event-based temporal conditioning. This further validates that they encode a *temporal* dimension, rather than merely responding to arbitrary numbers. Visualized results are in Figure 14 of Appendix.

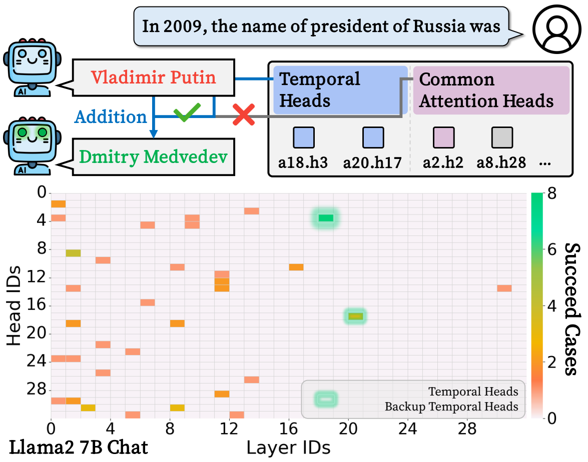

Figure 5: Example Of Temporal Knowledge Editing. From the source prompt, we catch the specific attention value of model's head, for example, **a18.h3**. By simply adding it to target prompt, the model's output is changed into temporally correct answer from temporally wrong answer. The headmap below denotes the number of success in editing for every combination of layers and heads. The most successful case in here is temporal heads **a18.h3** as highlighted, following other heads such as backup temporal heads **a20.h17**.

5 Temporal Knowledge Editing

Lastly, we explore an approach to confirm that injecting or amplifying *temporal head*'s attention value can effectively *"edit"* year-specific knowledge as in Figure 5. Given a `source_prompt` (where the model is confident about a certain year's fact) and a `target_prompt` (where it confuses the same year) based on log probability results, we:

1. **Extract** the value of attention head $\mathbf{a}_{\text{src}}$ from the `source_prompt` at a chosen layer/head (e.g. `a18.h3`).

2. **Average** over total source prompts (e.g., "In 2009, the name of president of South Korea was").

3. **Inject** the modified attention value into the `target_prompt` at the corresponding temporal token position, scaled by a coefficient λ:

Details of adding an attention is in Appendix 8.8.

This modification is applied dynamically using a forward hook mechanism at inference time, preserving the overall model parameters while selectively influencing time-conditioned factual recall. We test it with model wrong answer in a normal condition, varying the injection coefficient across three cases ($\lambda = 1, 3, 6$), following Turner et al., 2023; Rimsky et al., 2024, which emphasized its impact.

Remarkably, the model's completions shift from a temporally incorrect response (*"changed to Vladimir Putin"*) to the correct one (*"Dmitry Medvedev"*), aligning with the known presidency timeline. The heatmap in Figure 5 further supports this by visually representing the effectiveness of temporal knowledge editing across all layers and heads. While certain attention heads can influence the model's response, the **most successful cases** are consistently linked to temporal heads, with `a18.h3` exhibiting the highest success rate. Additionally, backup temporal heads, such as `a20.h17`, also rank among the top-performing heads, reinforcing their critical role in preserving and modifying time-conditioned knowledge. This highlights that temporal factual recall is not arbitrarily distributed but is instead concentrated in specialized subcomponents. Other results are in Figure 15.

This targeted intervention remains minimally invasive, as it does not require global fine-tuning but instead modulates the value of a single specialized head, thereby preserving most of the model's prior knowledge. Taken together, these findings reinforce the hypothesis that LLMs harbor a *temporal subcomponent* within specialized attention heads. By intercepting or amplifying these temporal heads, we can selectively alter time-conditioned responses, strengthening the claim that these heads are integral to the reinforcement of year-based factual knowledge.

6 Related Works

6.1 Temporal Knowledge of LLM

Despite advancements in LLMs, handling *temporal knowledge* remains a key challenge. While prior works focus on factual consistency (Petroni et al., 2019; Kassner and Schütze, 2020) or refining model editing in MLP layer (Mitchell et al.; Meng et al., 2022; Meng et al.), few address how facts evolve over time. Studies on time-aware QA and temporal probing (Chen et al., 2021; Zhang and Choi, 2021; Dhingra et al., 2022; Jang et al., 2022) reveal that LLMs struggle with dynamically shifting facts. Recent approaches attempt explicit temporal alignment (Kim et al., 2024; Zhao et al., 2024; Mousavi et al., 2025; Park et al., 2025), but have focused on external evaluations. Our findings highlight that LLMs encode temporal facts implicitly, relying on manipulable attention heads, underscoring the need for better temporal supervision and disentangled knowledge representations.

6.2 Attention Heads in Language Models

Under mechanistic interpretability (Olah et al., 2020; Vig et al., 2020; Sharkey et al., 2025), researches about attention heads were done by Voita et al., 2019; Wang et al., 2023; McDougall et al., 2024, showing off specific heads that copy key tokens to the output, ensuring consistency in transformers. These *Mover Heads* are a kind of induction heads (Olsson et al., 2022) moving syntactic information (Ferrando and Voita, 2024). Other works were followed as finding out retreval heads (Wu et al., 2024), heads for semantic information for color (Merullo et al., 2024), or subject and relation (Chughtai et al., 2024). Those various kinds of attention heads attend to critical tokens and directly influence the logits by writing their embeddings into the residual stream (Zheng et al., 2024).

Experiments show that ablating those heads significantly disrupts tasks like syntactic induction or semantic information understanding, highlighting their specific roles. A special case, *Backup Heads*, remains inactive under normal conditions but replicates task specific head functionality when primary heads are ablated. This ensures model robustness by maintaining token copying behavior even when key circuit components are disrupted. We treat founded temporal attention heads as a subcategory of semantic heads like subject heads and relation heads (Chughtai et al., 2024) in our experiments.

7 Conclusion

We systematically investigate how LLMs can handle *temporal knowledge*, focusing on time-dependent facts. Through our experiments, we uncovered *Temporal Heads* that selectively mediate the activation of time-variant knowledge. Ablating these heads leads to temporal mismatches while leaving time-invariant knowledge and general QA performance unaffected. Note that these heads are also activated under textual conditioning, and using their value for editing successfully changes the models' responses with minimal intervention.

As a foundational step, our work explores how LLMs can actively manage temporal information rather than merely integrating temporal context. We believe our analysis offers valuable insights into the inner mechanisms of LLMs and can inspire future approaches for *time-aware model alignment* and *precise temporal updates* by selectively targeting *temporal heads*, rather than relying on global retraining.

Limitations

While our approach demonstrates promising results in identifying and analyzing temporal knowledge circuits, we acknowledge some limitations in our current work. First, analysis of unstructured temporal QAs like General ChroKnowledge (Park et al., 2025) were constrained, as the underlying multiple-choice options in those tasks typically do not exhibit temporal dependencies. So we focused more on our temporal knowledge dataset, abundantly describing the effect of ablation in these cases.

On the other side, as EAP-IG didn't support models with Grouped-Query Attention (GQA), which cannot use the `split_qkv_input` option, our main analysis exclude those models like Llama-3-8B-Instruct (Dubey et al., 2024). Still, we checked their results and found that even their CRS is not quite enough and their circuit construction is not detailed, temporal heads are still could be founded: *a18.h15* and *a23.h26*.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Łukasz Borchmann. 2024. In case you missed it: Arc'challenge'is not that challenging. *arXiv preprint arXiv:2412.17758*.

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. 2024. Improving steering vectors by targeting sparse autoencoder features. *arXiv preprint arXiv:2411.02193*.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. *arXiv preprint arXiv:2108.06314*.

Bilal Chughtai, Alan Cooney, and Neel Nanda. 2024. Summing up the facts: Additive mechanisms behind factual recall in llms. *arXiv preprint arXiv:2402.07321*.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Javier Ferrando and Elena Voita. 2024. Information flow routes: Automatically interpreting language models at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17432–17445, Miami, Florida, USA. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495. Association for Computational Linguistics.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. 2023. Realtime qa: what's the answer right now? In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Yujin Kim, Jaehong Yoon, Seonghyeon Ye, Sangmin Bae, Namgyu Ho, Sung Ju Hwang, and Se-Young Yun. 2024. Carpe diem: On the evaluation of world knowledge in lifelong language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2024. Programming refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2024. Copy suppression: Comprehensively understanding a motif in language model attention heads. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 337–363, Miami, Florida, US. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Circuit component reuse across tasks in transformer language models. In *The Twelfth International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR.

Seyed Mahed Mousavi, Simone Alghisi, and Giuseppe Riccardi. 2025. Llms as repositories of factual knowledge: Limitations and solutions. *arXiv preprint arXiv:2501.12774*.

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

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. 2024. What does the knowledge neuron thesis have to do with knowledge? In *The Twelfth International Conference on Learning Representations*.

Kai Nylund, Suchin Gururangan, and Noah A Smith. 2023. Time is encoded in the weights of finetuned language models. *arXiv preprint arXiv:2312.13401*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2024a. Gpt-4o mini, advancing cost-efficient intelligence.

OpenAI. 2024b. Openai o1 system card.

Yein Park, Chanwoong Yoon, Jungwoo Park, Donghyeon Lee, Minbyul Jeong, and Jaewoo Kang. 2025. Chroknowledge: Unveiling chronological knowledge of language models in multiple domains. In *The Thirteenth International Conference on Learning Representations*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Marcel Proust. 1992. *In Search of Lost Time*. Modern Library, New York. Originally published in French as *À la recherche du temps perdu* (1913–1927).

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. 2025. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*.

Qwen Team. 2024. Introducing qwen1.5.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv e-prints*, pages arXiv–2308.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*.

Jianing Wang. 2022. Math-kg: Construction and applications of mathematical knowledge graph. *arXiv preprint arXiv:2205.03772*.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. In *The Eleventh International Conference on Learning Representations*.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024. Knowledge circuits in pretrained transformers. In *Advances in Neural Information Processing Systems*, volume 37, pages 118571–118602. Curran Associates, Inc.

Michael Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387.

Bowen Zhao, Zander Brumbaugh, Yizhong Wang, Hannaneh Hajishirzi, and Noah Smith. 2024. Set the clock: Temporal alignment of pretrained language models. In *Findings of the Association for Computational Linguistics ACL 2024*, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Attention heads of large language models: A survey. *arXiv preprint arXiv:2409.03752*.

8 Appendix

8.1 Effective Attribution Pruning-Integrated Gradients (EAP-IG)

We perform **Effective Attribution Pruning** (EAP) by ablating (zeroing) candidate edges and measuring the drop in correct predictions following Hanna et al., 2024. In tandem, we use **Integrated Gradients** (IG) to capture gradient-based contributions:

$$\mathrm{IG}(\mathbf{z}, \mathbf{z}') = \int_0^1 \frac{\partial}{\partial \mathbf{z}} \mathcal{L}(\mathbf{z}' + \alpha(\mathbf{z} - \mathbf{z}'))\, \mathrm{d}\alpha, \quad (9)$$

where $\mathcal{L}$ is the loss (e.g., negative log-likelihood), and $\mathbf{z}'$ a baseline embedding or activation. Furthermore, not just combining signals to rank each node/edge by its importance, we extend EAP-IG to *time-sensitive* knowledge. We construct **temporal knowledge circuits** by analyzing variations across different years T_k. For a given (s, r) pair:

- **Clean input**: (s, r, o_t) where o_t is correct at T_t.
- **Corrupted inputs**: $(s, r, o_{t'})$ where $o_{t'}$ is the correct object for a different time $T_{t'} \neq T_t$.

Rather than treating $o_{t'}$ as incorrect, we leverage the contrast between different valid temporal associations to isolate time-dependent components. An edge e_i is retained in the temporal circuit if:

$$\begin{aligned} S(e_i, T_k) = {} & \log p_G(o_k \mid s, r, T_k) \\ & - \log p_{G/e_i}(o_k \mid s, r, T_k) > \tau. \end{aligned} \quad (10)$$

This identifies edges that encode temporal specificity rather than general factual associations. By ablating edges across different T_k, we verify if disruptions occur primarily at the corresponding time while preserving outputs for other years. This ensures the extracted circuits genuinely reflect temporal dependencies.

8.1.1 Implementation Details in EAP-IG

In each model's configuration, we set `split_qkv_input` to true in transformer lens (Nanda and Bloom, 2022), ensuring attention heads are disentangled enough for targeted pruning. The `ig_steps` for integrated gradients, we set it as 100. We use `top_n` 5000 settings and the τ for simplified threshold, we use 0.1 as a predefined value for every models to cutting out unimportant edges and nodes. The experiments are all done with one NVIDIA A100 GPUs(80GB), less than 30 minutes per each runs.

8.2 Causal Tracing

Causal Tracing (Vig et al., 2020; Meng et al., 2022) aims to reveal which hidden states in an autoregressive Transformer *cause* correct recall of a fact. Let a fact be (s, r, o) (e.g., $(\texttt{L. Messi}, \texttt{sports\_team}, \texttt{Newell's Old Boys})$), and time T (e.g., `In 1999`). We construct a prompt p (e.g., *"In 1999, Lionel Messi was a member of sports team ..."*) and measure the model's probability of generating o at output:

$$p_{\text{clean}}(o) = G(p), \quad (11)$$

where G is the Transformer. Next, we create a *corrupted* prompt p' (e.g., replacing "Lionel Messi" with a fake name). Denote the model's probability,

$$p_{\text{corr}}(o) = G(p'). \quad (12)$$

Because key information is obfuscated, $p_{\text{corr}}(o)$ typically drops. Finally, in the *corrupted-with-restoration* run, we overwrite certain hidden states in the corrupted run with their clean-run counterparts:

$$p_{\text{restored}}(o) = G_{\text{restore}}\Big(p', \{\mathbf{h}^{(l)}_{\text{clean}}\}\Big), \quad (13)$$

where $\mathbf{h}^{(l)}_{\text{clean}}$ are layer-l hidden states from the clean run. If restoring layer l significantly boosts $p_{\text{restored}}(o)$, those states at layer l are *causally important* for retrieving the fact. Applying this procedure to time-conditioned facts (e.g., specifying "In 1999," "In 2009," etc.) localizes *temporal* knowledge within specific tokens and layers.

8.3 Where Does Temporal Condition Exert Influence on Knowledge Triplets?

We next investigate precisely *where* a temporal cue, such as *"In 1999,"* or *"In 2004,"* exerts its main influence within the triplet (s, r, o). To this end, we adopt a causal-tracing approach (inspired by ROME (Meng et al., 2022)) targeted at isolating *temporal* effects. Specifically, we compare two prompts:

- **Without Temporal Cue:** *"Lionel Messi was a member of sports team ..."*
- **With Temporal Cue:** *"In 1999, Lionel Messi was a member of sports team ..."*

By inserting noise (or other forms of corruption) into specific tokens (often the subject or the year token) and selectively restoring only certain hidden states, we measure how each portion of the input affects final predictions. Our experiments on a Llama2 model highlight that *subject tokens*, when combined with a year, incur the largest impact on retrieving the correct year-specific fact.

8.3.1 Year-Based Causal Tracing of Subject Tokens

Heatmap Illustrations The top 6 plots in Figures 6 depict example heatmaps for *single-layer* restoration (left) vs. *MLP-interval* and *Attention-interval* restoration (center, right). Each subplot visualizes how restoring a given layer (or set of layers) changes the probability of a target answer (e.g., $\mathrm{p}(\text{New})$ or $\mathrm{p}(\text{Barcelona})$). Darker regions indicate larger improvements in the model's correctness after that restoration. We compare:

- **Top row:** Restoration effect on $\mathrm{p}(\text{New})$ or $\mathrm{p}(\text{Barcelona})$ for different single or grouped layers, showing which layers are most responsible for *selecting* a new or correct team.
- **Bottom row:** Similar restoration but for alternative completions (e.g., $\mathrm{p}(2)$ or $\mathrm{p}(\text{Lion})$), revealing how subject or year tokens can shift the model's internal preference.

We observe that certain mid-range layers, especially around 10–20, exhibit strong spikes: when we restore those layers' subject-year hidden states, the model reverts to a correct or plausible answer for the year-specific query.

Time Affects the *Subject* Most As hinted by the heatmaps:

- The *largest gain in correct probability* typically occurs after restoring subject+year hidden states. If corrupted, the model confuses or misaligns the year with the wrong subject, yielding off-target outputs (e.g. a different team or a random hallucination).
- Other tokens (relation or object) produce *smaller* jumps when restored. Although they matter for the final fact, they do not exhibit the same *temporal* sensitivity as the subject domain.

8.3.2 Year-Based Causal Tracing of Relation and Object Tokens

The middle and lower side six plots in Figures 6 replicate the above procedure for *relation tokens* (e.g., "was a member of") and *object tokens* (e.g., a team name). The heatmaps show weaker or narrower restoration effects when the year corruption is placed near those tokens:

- **Relation tokens** only yield modest probability recovery upon restoration, implying that while they shape the factual link, they do not anchor the *time* dimension.
- **Object tokens** affect final correctness but appear less coupled to the year. Overwriting their hidden states helps for precise object naming, yet does not fix *when* an event is said to occur.

8.3.3 Implications for Temporal-Subject Coupling

In line with prior studies (Meng et al., 2022), these findings confirm that the *temporal aspect* is mainly fused into the *subject* representation—the model effectively treats "(Subject in Year)" as a unique entity. Restoring the subject+year region of hidden states yields the greatest improvement, implying that year tokens attach strongly to the subject slot. Conversely, *relation* and *object* tokens are comparatively less sensitive to time cues.

Limitations of Causal Tracing Alone Despite highlighting *which layer* or *token positions* matter, causal tracing alone cannot pinpoint *which heads or MLPs* form the circuit that routes these time signals. For instance, a single layer might have multiple attention heads with different behaviors; or an MLP might selectively process the year dimension but remain obscure at the token-level. As we explore in (§3), adopting a *circuit-level* perspective unveils specific *Temporal Heads* that systematically propagate year-conditioned knowledge throughout the model.

8.4 Details of Circuit Reproduction Score

CRS condenses relative performance differences and sign alignment into a single, intuitive 0–100 metric, offering a streamlined assessment of circuit quality.

8.4.1 Motivation

Existing approaches such as *logit diff* or *MatchNLL* (Conmy et al., 2023; Yao et al., 2024) evaluate circuits by reporting two separate numbers: the **baseline performance** of the original model and the **circuit's performance**. However, this can

Category	Knowledge	#Node	#Edge	CRS
Temporal				
Sports	Nicolas Anelka	27	26	88.81
	David Beckham	42	59	26.50
Presidents	Argentina	38	64	43.99
	South Korea	51	104	53.18
CEO	Hewlett-Packard	31	34	40.36
	Chrysler	26	22	28.14
Defense	United States	8	5	25.60
	China	13	9	25.82
Avg		**30**	**40**	**41.44**
Time-Invariant				
CommonSense	Object Superclass	72	127	42.61
Conditional CS	Fruit Inside Color	43	49	64.83
Num in Obj	Geometric Shape	60	127	62.94
Num in Sub	Roman Numerals	57	108	71.18
Avg		**58**	**103**	**60.39**

Table 3: Statistics of temporal knowledge circuits for Qwen 1.5, both temporal and time-invariant knowledge. For temporal knowledge, each type of knowledge is reproduced with three selected years: **1999, 2004, and 2009**. The numbers of nodes, edges and CRS is the average of each knowledge's yearly circuits. We simplified total circuits with $\tau = 0.1$, same as Llama2.

Category	Knowledge	#Node	#Edge	CRS
Temporal				
Sports	Nicolas Anelka	5	3	64.51
	David Beckham	22	22	42.24
Presidents	Argentina	53	127	91.19
	South Korea	55	142	81.47
CEO	Hewlett-Packard	12	9	35.55
	Chrysler	9	7	73.98
Defense	United States*	3	1	73.03
	China*	2	1	72.85
Avg		**20**	**39**	**66.85**
Time-Invariant				
CommonSense	Object Superclass	73	135	61.49
Conditional CS	Fruit Inside Color	24	44	49.48
Num in Obj	Geometric Shape	16	20	39.98
Num in Sub	Roman Numerals	78	153	74.04
Avg		**48**	**88**	**56.25**

Table 4: Statistics of temporal knowledge circuits for Phi 3 mini, both temporal and time-invariant knowledge. For temporal knowledge, each type of knowledge is reproduced with three selected years: **1999, 2004, and 2009**. The numbers of nodes, edges and CRS is the average of each knowledge's yearly circuits. We simplified total circuits with $\tau = 0.1$, same as Llama2, except knowledge in Defense where at least 30% lower τ is needed. Interestingly, Phi 3 mini shows better CRS of temporal knowledge than time-invariant ones, though their overall simplified nodes and edges are less than same cases of other models.

obscure direct comparisons, especially when values are of different scales or signs. To address this, we introduce the **Circuit Reproduction Score (CRS)**, a unified metric that normalizes these comparisons onto a **0–100 scale**. A score of 0 indicates a circuit that fails to retain meaningful model behavior, while 100 signifies equal or superior performance compared to the original model.

8.4.2 Definition

Let B represent the **baseline performance** of the original model and P the **circuit's performance**. CRS is computed as:

$$CRS(B, P) = 100 \times S(B, P) \times D(B, P), \quad (14)$$

where:

- $S(B, P) \in (0, 1]$ is a sign-based adjustment factor.
- $D(B, P) = \exp(-\alpha R)$ scales based on deviation R.
- α controls the sensitivity to deviations.

The deviation R is defined as:

$$R = \frac{\text{dist}(B, P)}{|B|}, \quad (15)$$

where $\text{dist}(B, P)$ measures how far P deviates from B.

If the circuit's performance meets or exceeds the baseline ($B > 0$ and $P \geq B$), CRS is set to:

$$CRS(B, P) = 100. \quad (16)$$

Handling Positive and Negative Baselines

- If $B > 0$ and $P \geq B$, CRS is 100, indicating that the circuit fully retains or improves upon original performance.
- If $P < B$, the CRS score is exponentially reduced based on the relative performance gap.
- If $B < 0$ (indicating the original model performed poorly), less negative performance is treated as an improvement.
- If B and P differ in sign, CRS applies an intermediate weighting (e.g., 0.6–0.8) to avoid misleadingly high scores.

8.4.3 Implementation

We compute:

$$B = \text{eval\_baseline}(G, \mathcal{D}_{val}, \text{logit\_diff}), \quad (17)$$

$$P = \text{eval\_graph}(G, P, \mathcal{D}_{val}, \text{logit\_diff}). \quad (18)$$

Category	Time Range	# of Cases
***Temporal Knowledge* (*Vrandečić and Krötzsch, 2014*)**		
Sports	1996-2020	81
Presidents	1999-2009	65
CEO	1999-2009	65
Defense	1999-2009	77
Movies	1999-2009	33
GDP	1999-2009	33
Inflations	1999-2009	33
***Time Invariant Knowledge* (*Hernandez et al., 2024*)**		
Object Superclass	-	36
Fruit Inside Color	-	76
Geometric Shape	-	28
Roman Numerals	-	31

Table 5: Statistics of dataset used for circuits.

Dataset	Format	Test	Source
TriviaQA	MCQA	11,313	Joshi et al., 2017
Math ChroKnowledge	MCQA	2,585	Wang, 2022; Park et al., 2025

Table 6: Statistics of dataset used general QA.

These yield average performance values, which are then converted into:

$$CRS = \text{one\_score}(B, P; \alpha, S) \in [0, 100]. \quad (19)$$

The resulting CRS provides a concise and interpretable measure of circuit faithfulness:

- **Both negative:** The circuit's score is capped (e.g., at most 100×0.5).
- **Both positive:** The circuit may reach 100 if it fully retains baseline performance.
- **Mixed sign:** An intermediate factor (e.g., 0.6–0.8) prevents inflated scores if the circuit behaves in an unintended manner.

8.4.4 Hyperparameters

The CRS computation relies on several hyperparameters that modulate its sensitivity to deviations and its handling of different sign scenarios:

- α: Sensitivity to deviation, controlling how sharply CRS decreases as the circuit deviates from the baseline. Default: 1.0.
- sf_{bothpos}: Sign factor when both baseline and circuit performance are positive ($B > 0$, $P > 0$). Default: 1.0.
- sf_{bothneg}: Sign factor when both baseline and circuit performance are negative ($B < 0$, $P < 0$). Default: 0.5.
- $sf_{\text{bneg\_cpos}}$: Sign factor when the baseline is negative but the circuit is positive ($B < 0$, $P > 0$). Default: 0.8.
- $sf_{\text{bpos\_cneg}}$: Sign factor when the baseline is positive but the circuit is negative ($B > 0$, $P < 0$). Default: 0.6.
- ϵ: Small constant for numerical stability, ensuring nonzero denominators and preventing division errors. Default: 10^{-9}.

8.5 Details and Statistics of Dataset

Table 5 and 6 present the statistical details of the knowledge datasets used in our evaluation. For temporal knowledge, we utilize open-sourced WikiData as referenced. These datasets encompass a variety of knowledge categories, each consisting of multiple objects along with their associated time ranges.

8.5.1 Categorization of Knowledge Datasets

Each dataset category represents a specific type of structured knowledge:

Temporal Knowledge. This category contains knowledge that varies over time, requiring temporal awareness for accurate retrieval. The definitions for each subcategory are as follows:

- **Sports**: The teams associated with specific athletes over time.
- **Presidents**: The names of country leaders for given years.
- **CEO**: The chief executive officers of major companies in a given year.
- **Defense**: The national defense budget of different countries.
- **Movies**: The highest-grossing films by country for specific years.
- **GDP**: The annual Gross Domestic Product (GDP) of various countries.
- **Inflation**: The inflation rate of different countries for given years.

Settings	Temporal Knowledge (%)							Average
	Sports	Presidents	CEO	Defense	Movies	GDP	Inflations	
Llama-2-7b-chat-hf - a18,h3, a15.h0								
Baseline	41.9	80.7	27.5	13.5	23.1	10.4	10.8	29.7
Ablation	40.0	75.6	21.3	13.3	9.37	10.7	9.34	25.6
Qwen1.5-7B-Chat - a17.h15								
Baseline	32.4	57.2	19.6	11.5	16.7	9.58	10.0	22.4
Ablation	32.0	49.4	16.6	10.3	10.8	9.50	10.3	19.8
Phi-3-mini-4k-instruct - a10.h13								
Baseline	24.4	72.1	30.8	73.7	21.4	12.2	13.5	35.4
Ablation	24.8	69.6	30.7	11.5	21.6	11.7	11.8	26.0

Table 7: Total results of temporal knowledge across multiple models. Each scores were measured in probability (%) with averaging effect of multiple heads ablation results. The most dropped score for each column is colored red.

Time-Invariant Knowledge. Unlike temporal knowledge, this category consists of facts that do not change over time. The specific subcategories are defined as follows:

- **Object Superclass**: General commonsense knowledge that categorizes objects into broader superclasses.
- **Fruit Inside Color**: Commonsense knowledge conditioned on the phrase "On the inside," focusing on the internal color of fruits.
- **Geometric Shape**: Knowledge where objects are associated with numerical properties, such as shape classifications based on the presence of numbers.
- **Roman Numerals**: Cases where numerical values appear in the subject itself, typically involving Roman numeral representations.

8.5.2 General Question Answering (QA) Datasets

In addition to the structured knowledge datasets, we also utilize benchmark QA datasets for evaluation. The test or validation sets provided by these benchmarks are used in our experiments. All evaluations are conducted under the **Multiple-Choice Question Answering (MCQA)** setting. Statistics are following Table 6.

8.6 Details of Log Probability Check

Our evaluation follows the paradigm outlined in Borchmann 2024, focusing on log probability variation rather than direct answer accuracy. Standard multiple-choice evaluations often overestimate model difficulty by testing answers in isolation rather than in comparative contexts. Instead, we analyze how ablation affects probability distributions across all candidate objects, providing a more granular view of temporal knowledge representation. By using per-object probability tracking, we reveal a more precise representation of how temporal information is encoded and manipulated within the model.

Notations Let M be the transformer model under evaluation, and let O be the set of all candidate objects (e.g., teams, presidents). For a given input, the model assigns a probability $p(o|s, r, T)$ to each object $o \in O$ with given subject s, relation r and time T, representing its likelihood of being the correct answer. The object assigned the highest probability is labeled `Target` if it corresponds to the correct temporal fact, or `Non-Target` otherwise.

Per-Choice Probability Assessment Unlike conventional approaches, which focus solely on the final prediction, we track probability variations across all objects. This ensures that we capture nuanced knowledge shifts caused by ablation, rather than just observing whether the top-ranked answer changes.

Head Ablation and Probability Recalculation To examine the role of temporal attention heads, we zero out selected heads $\hat{H}$ and measure how the model's probability distribution over O shifts. The

Settings	Time Invariant Knowledge (%)					General QA (F1 & %)	
	Obj-Super	Fruit In-Color	Geo-Shape	Roman-Num	Average	TriviaQA	Math
Llama-2-7b-chat-hf - a18,h3, a15.h0							
Baseline	49.7	75.6	68.5	53.5	61.8	55.4	45.4
Ablation	50.2	75.6	68.1	53.0	61.7	54.9	45.3
Qwen1.5-7B-Chat - a17.h15							
Baseline	48.0	72.0	69.4	61.5	62.7	49.7	77.0
Ablation	47.8	72.0	69.3	61.1	62.6	49.5	77.0
Phi-3-mini-4k-instruct - a10.h13							
Baseline	21.8	76.0	68.3	73.2	59.8	46.8	80.8
Ablation	23.2	76.4	69.1	73.7	60.6	46.2	81.2

Table 8: Total results of time invariant knowledge and general QA across multiple models. For TriviaQA, we test the unfiltered, no-context validation set (11.3k). Each scores were measured in probability (%) or f1 score with averaging effect of multiple heads ablation results. Most of cases, the scores remain stable or even goes up such as *Object Superposition*.

recalculated probability after ablation is given by:

$$z_o = \log p_{\text{ablate}}(o|s, r, T), \tag{20}$$

$$\hat{p}_o = \frac{\exp(z_o)}{\sum_{o' \in O} \exp(z_{o'})}, \tag{21}$$

where p_{ablate} denotes the log-probability computed by forward pass of model, with ablation of corresponding heads in $\hat{H}$. Unlike standard evaluation, this method isolates the impact of specific attention heads on temporal knowledge retention.

8.7 Total Result Each Datasets

Table 7–8 indicates total result of time variant, invariant and general QA for all three models. We additionally deal with the case of Movies (which movie is the most popular in each year for each countries), GDP (how much GDP for each year for each countries) and Inflation (the inflation rate of each countries). As colored in red, temporal knowledge drops more drastically than time invariant knowledge or general QA.

8.8 Details of Temporal Knowledge Editing

8.8.1 Attention Value Extraction and Injection

We employ a direct attention value addition method to influence the model's temporal knowledge representation. Though we inspired by activation addition or patching methods like Rimsky et al., 2024; Li et al., 2024; Lee et al., 2024; Chalnev et al., 2024 and especially Turner et al., 2023, which computes an activation difference between positive and negative prompts, our method directly extracts value of attention heads from the `source_prompt` and injects them into the `target_prompt`.

Extracting Value of Attention Head For a given `source_prompt`, we extract the value from a specific attention head (l, h) at the token position corresponding to the temporal entity:

$$\mathbf{a}_{\text{src}} = \text{AttnV}(x_{\text{src}}, l, h), \tag{22}$$

where x_{src} is the tokenized `source_prompt` and $\text{AttnV}(x, l, h)$ returns the attention value at layer l and head h.

To obtain a stable representation across multiple `source_prompts`, we compute the mean value:

$$\mathbf{a}_{\text{src}} = \frac{1}{N} \sum_{i=1}^{N} \text{AttnV}(x_{\text{src}}^{(i)}, l, h), \tag{23}$$

Identifying Temporal Token Position In the `target_prompt`, we locate the last token index of the temporal condition to determine where the AttnV injection should occur.

Attention Value Injection Once the temporal token index t_{subj} is found, we inject the extracted AttnV:

$$\mathbf{a}_{\text{tgt}} = \text{AttnV}(x_{\text{tgt}}, l, h), \tag{24}$$

$$\mathbf{a}_{\text{tgt}}^{\text{new}} = \mathbf{a}_{\text{tgt}} + \lambda \mathbf{a}_{\text{src}}, \tag{25}$$

where x_{tgt} is the tokenized `target_prompt`, λ is the injection coefficient ($\lambda \in \{1, 3, 6\}$), and $\mathbf{a}_{\text{tgt}}^{\text{new}}$ is the modified value. This modification is applied dynamically using a forward hook:

$$\text{Hook}(\mathbf{a}) = \mathbf{a} + \lambda \mathbf{a}_{\text{src}}, \tag{26}$$

where $t = t_{\text{temp}}$ and x_{temp} is the tokenized temporal condition (e.g., "In 2009").

8.8.2 Evaluation Metrics

To assess the impact of attenion value injection, we introduce two evaluation criteria.

First-Token Prediction Shift We measure whether the injected value shifts the model's predicted first token. Given the target prompt x_{tgt}, we compare the probability of the correct response w^* before and after injection:

$$P(w^*|x_{\text{tgt}}) < P(w^*|x_{\text{tgt}}^{\text{new}}), \qquad (27)$$

where $P(w^*|x_{\text{tgt}})$ is the original probability of generating the correct token and $P(w^*|x_{\text{tgt}}^{\text{new}})$ is the probability after attention value injection.

This probability shift is measured using log-probabilities from the model's output distribution.

Full-Text Response Validation To further verify the efficacy of our method, we check whether the model's full generated response contains the expected factual entity. Specifically, we count the number of experiments where the correct answer appears in the model's output (e.g., "Dmitry Medvedev" for the name of president of Russia in 2009).

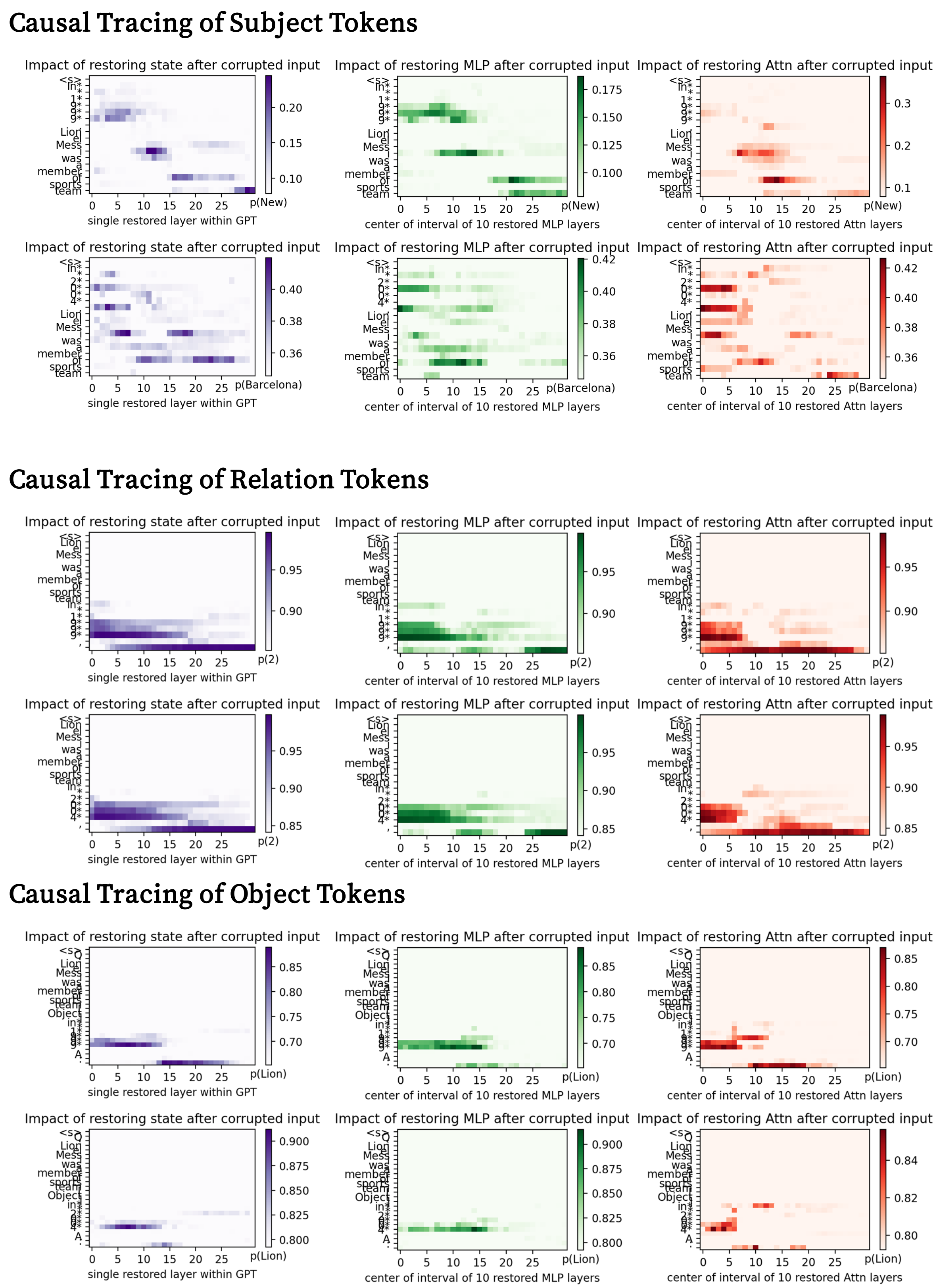

Figure 6: Results of Causal Tracing for all position(subject, relation, object), six plots for each cases from the top to middle and bottom. The restoring part is set to each temporal conditioning, in two different age: 1999 and 2004. (Illustrative) Causal tracing heatmaps showing how restoring different layers (x-axis) after temporal corruption affects p(New) or p(Barcelona). For the object position, we set a simulated *[Object]* for the place holder. Each figure's left column represents single-layer restoration; the center and right columns reflect MLP vs. attention intervals. Restoring subject+year at mid layers yields pronounced differences (dark regions). On the other hand, restoring relation+year or object+year yields trivial differences as their range is overlap significantly.

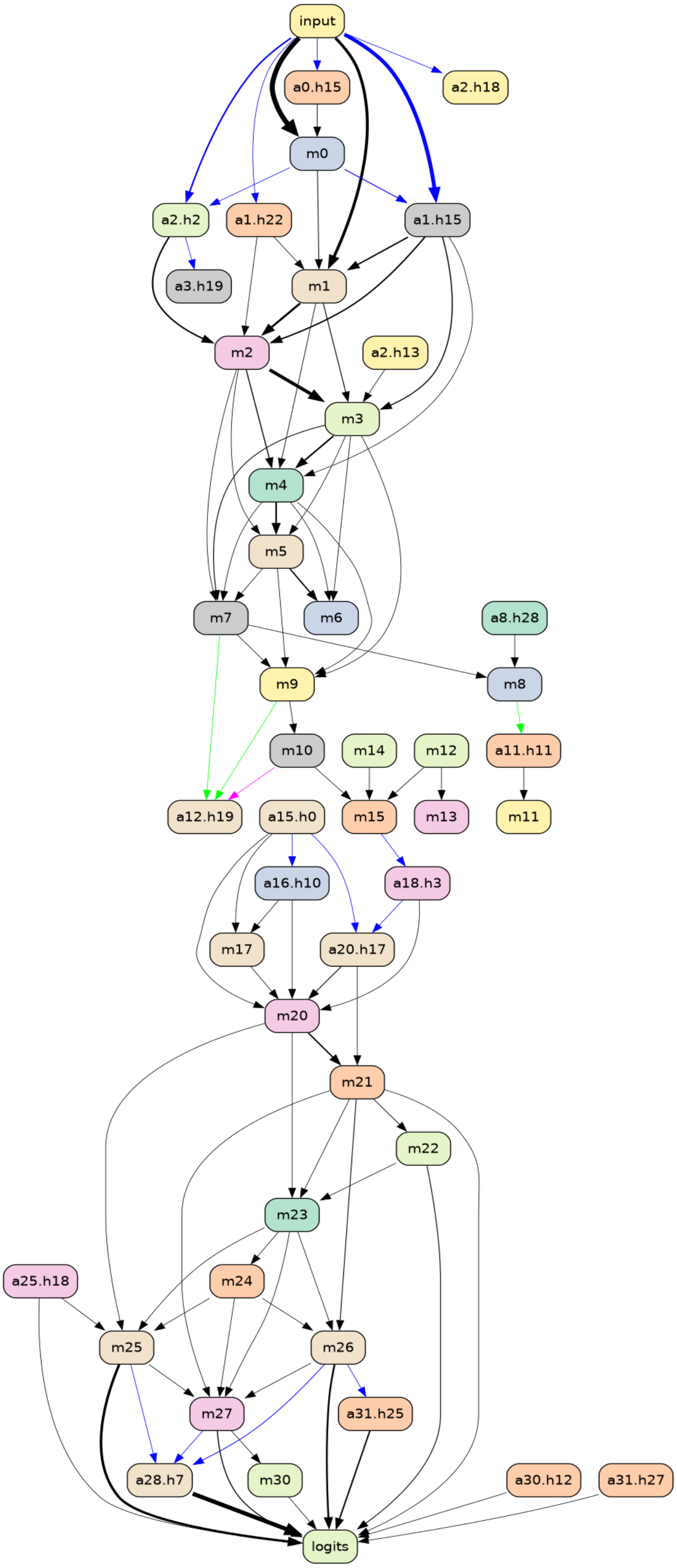

Figure 7: Temporal knowledge circuit of Llama2. It is simplified version of total circuit by its importance of each nodes using $\tau = 0.1$ as threshold.

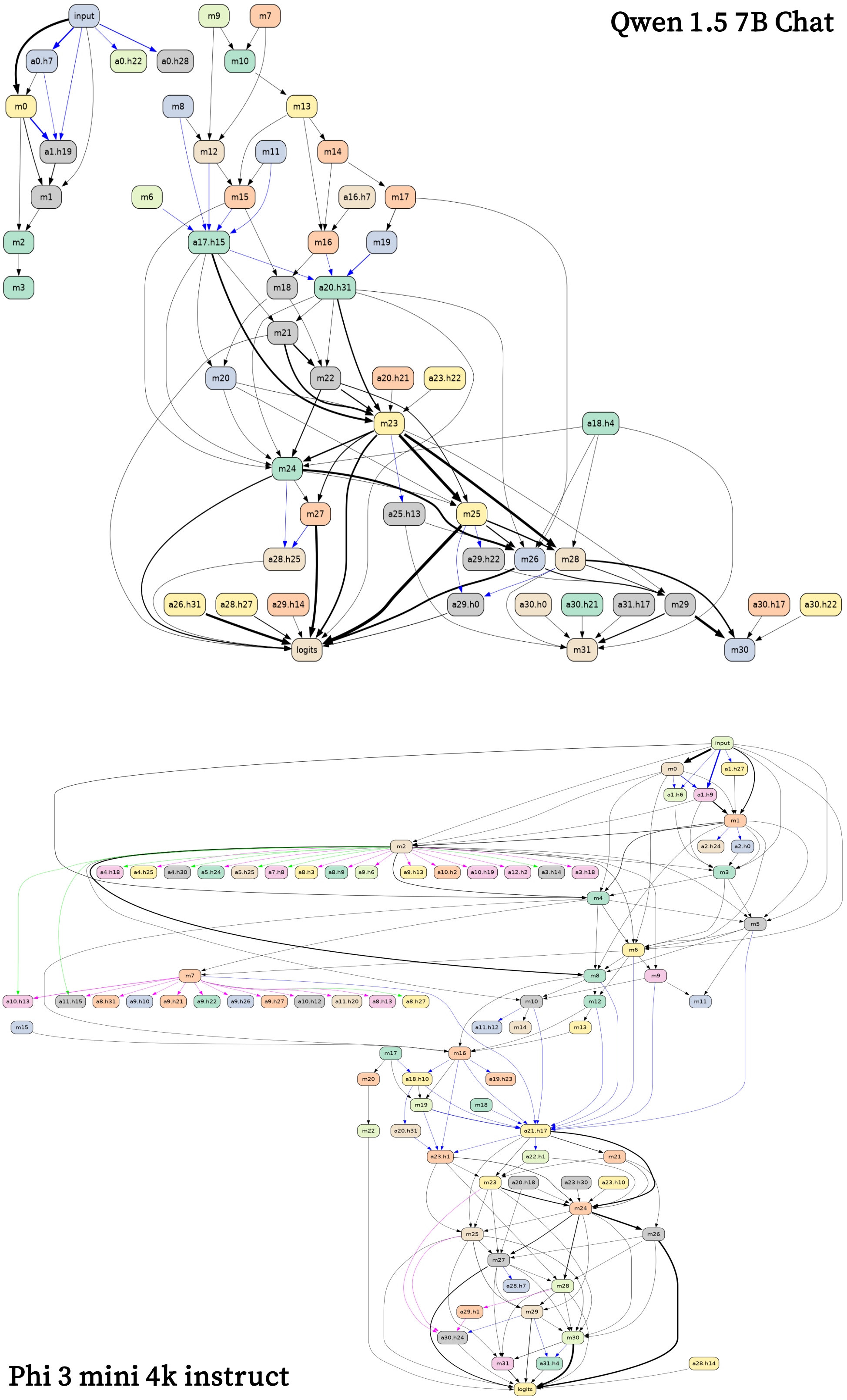

Figure 8: Temporal knowledge circuit of Qwen 1.5 and Phi 3 mini. Those are simplified version of total circuit according to each nodes and edges' importance of using same $\tau = 0.1$ as threshold.

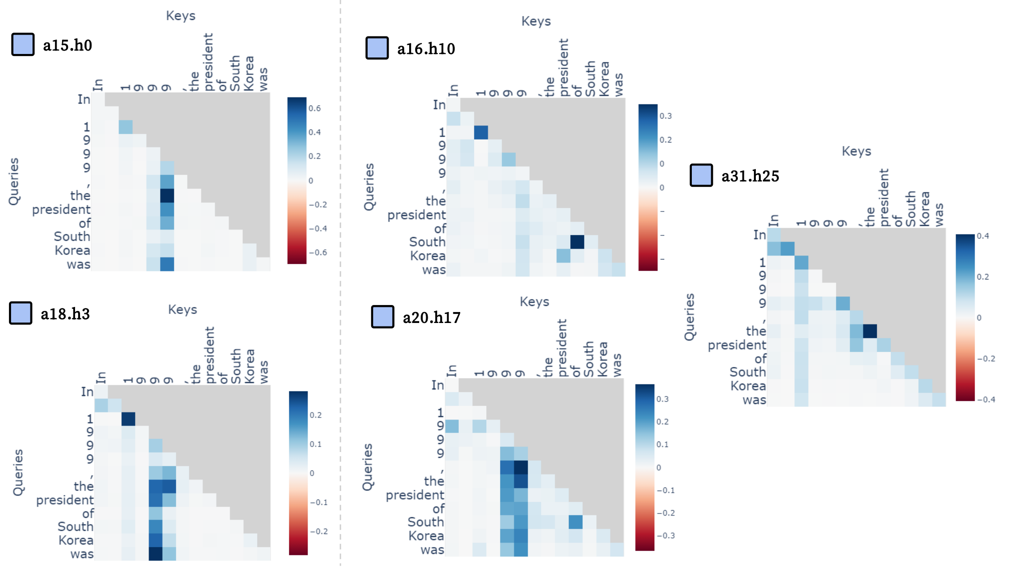

Figure 9: Total map of attention with Llama2-7b-chat-hf, for each temporal heads and backup temporal heads. The left side of border line is the attention map of **Temporal Heads**, and the other side is the result of **Backup Temporal Heads**.

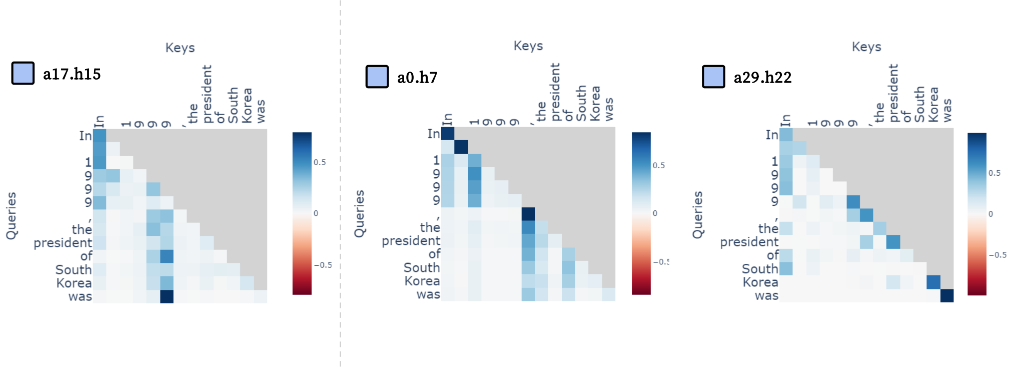

Figure 10: Total map of attention with Qwen1.5-7B-Chat, for each temporal heads and backup temporal heads. The left side of border line is the attention map of **Temporal Heads**, and the other side is the result of **Backup Temporal Heads**.

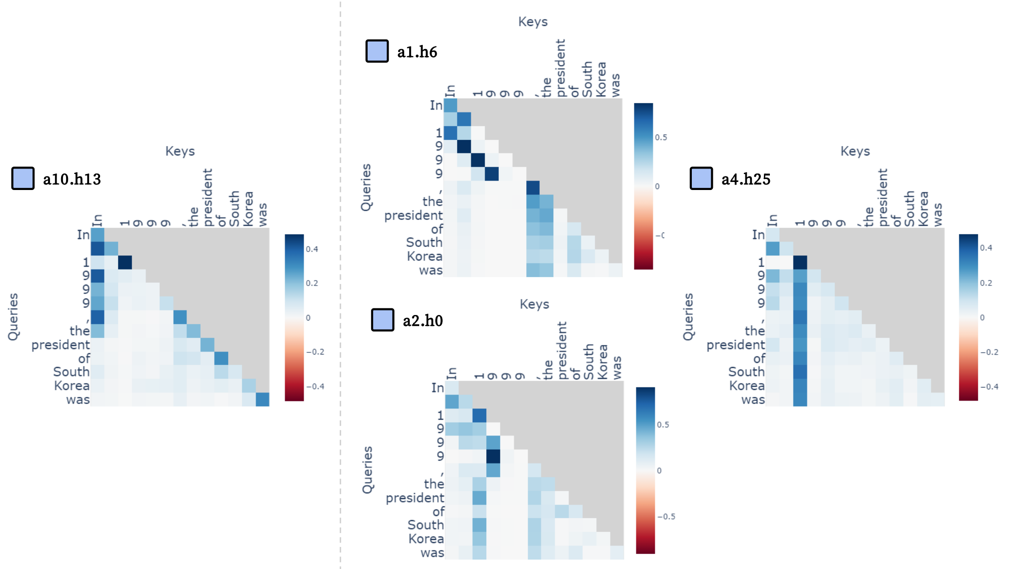

Figure 11: Total map of attention with Phi-3-mini-4k-instruct, for each temporal heads and backup temporal heads. The left side of border line is the attention map of **Temporal Heads**, and the other side is the result of **Backup Temporal Heads**.

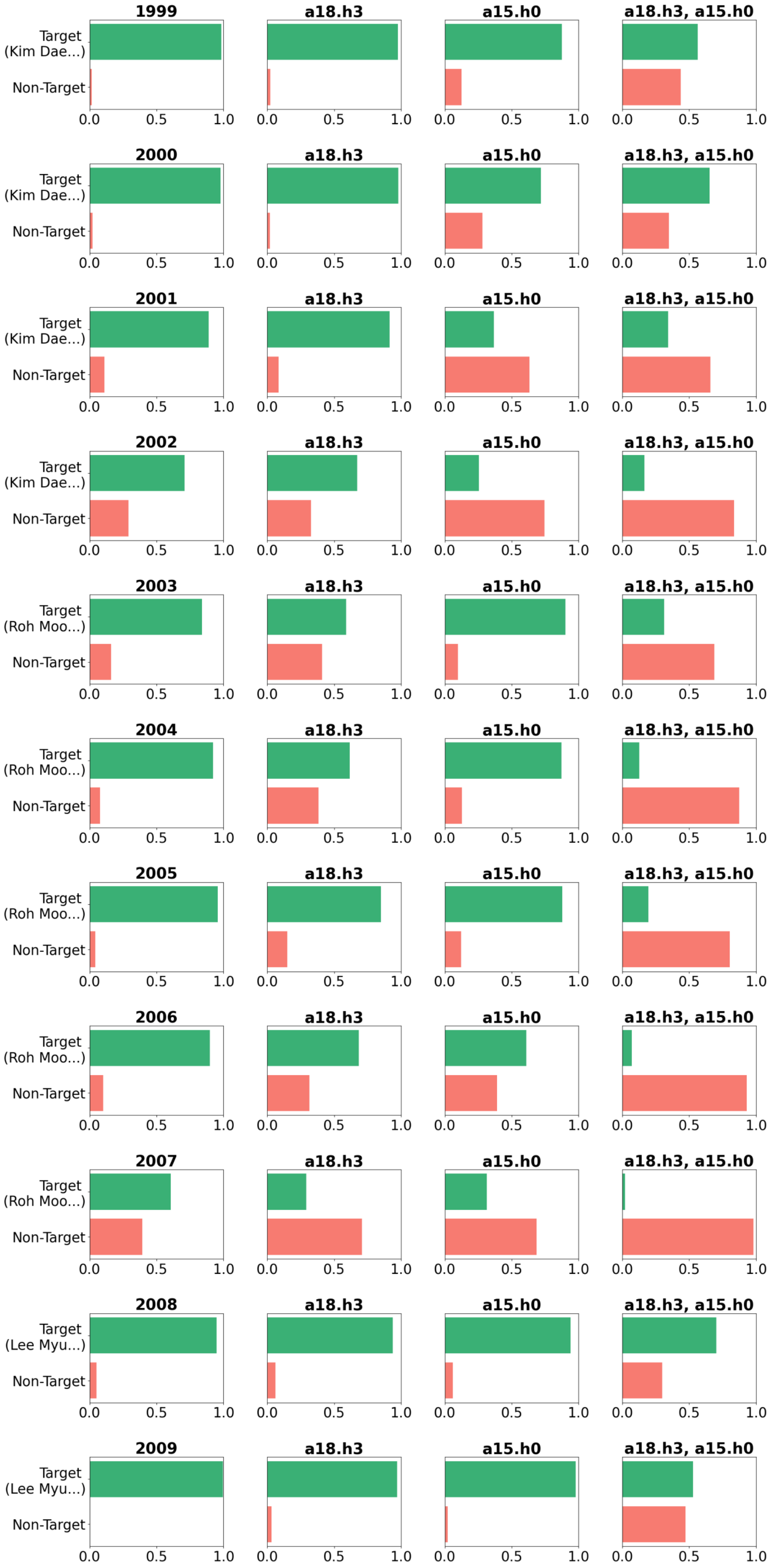

Figure 12: Total results of Llama2-7b-chat-hf, head ablation inference with log probability.

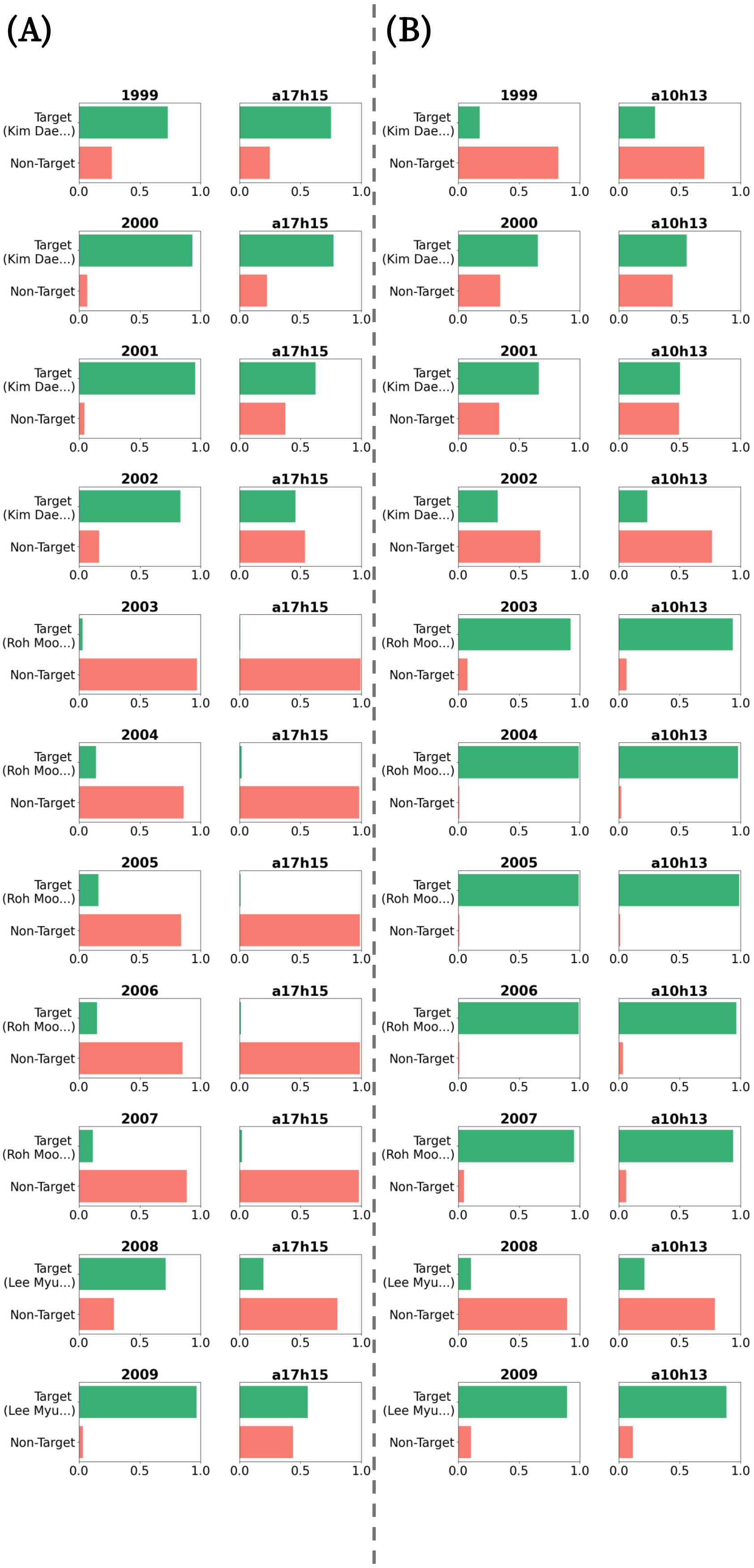

Figure 13: Total results of Qwen1.5-7B-Chat and Phi-3-mini-4k-instruct, head ablation inference with log probability. (A) denotes the result of Qwen 1.5 and (B) represents the result of Phi 3 mini.

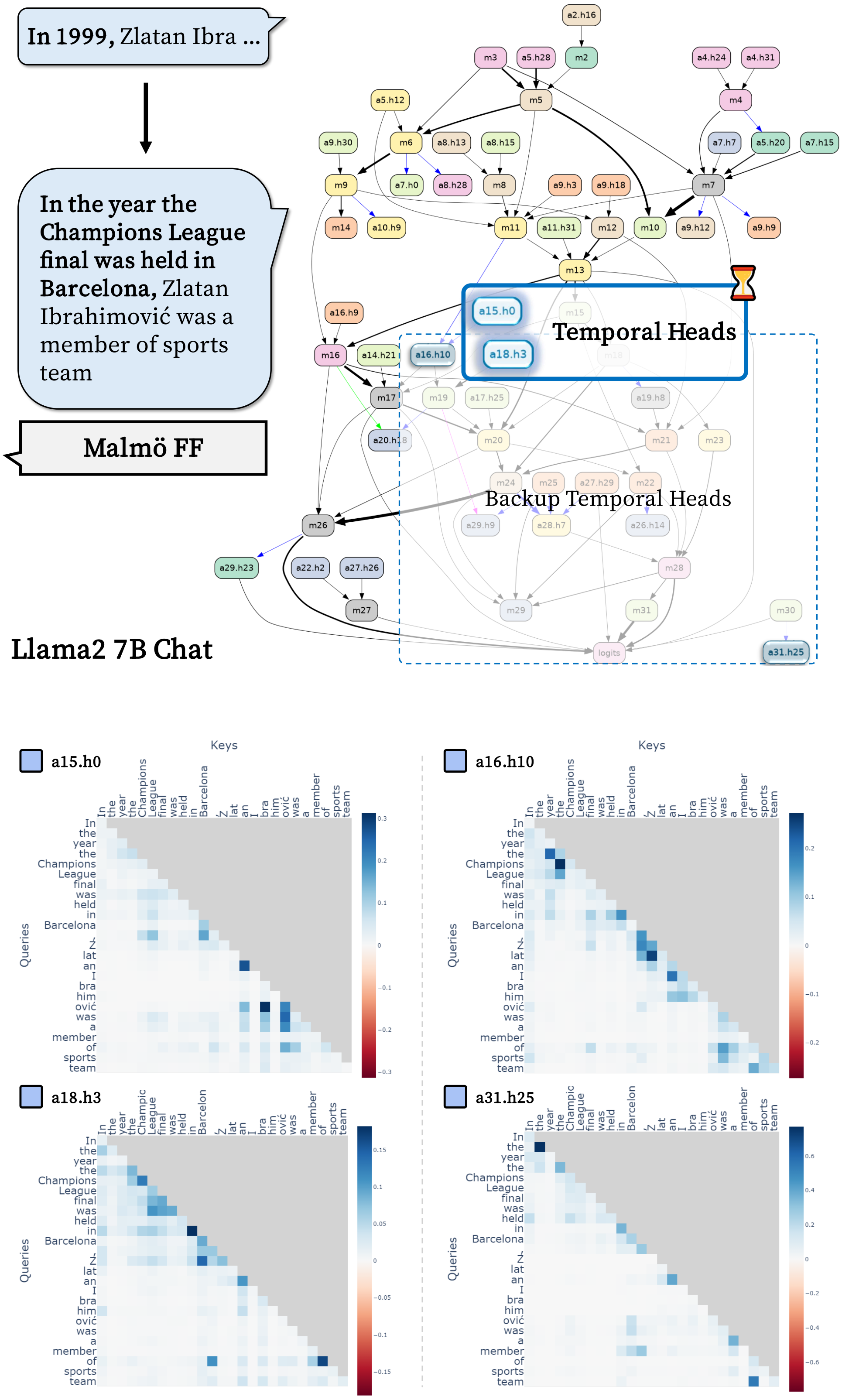

Figure 14: Temporal knowledge circuit from textual temporal conditioned prompt. Here, we change the temporal condition *"In 1999"* into *"In the year the Champions League final was held in Barcelona"*, which model already correctly recalls the answer *Malmö FF*. The temporal knowledge circuit successfully catches temporal conditioning even with alias based on event based textual conditioning, with correctly showing off temporal knowledge heads and some backup temporal heads. Figure of downside is the attention maps for each temporal heads and backup temporal heads. Each of those figures highlight various tokens in conditioning part of prompt.

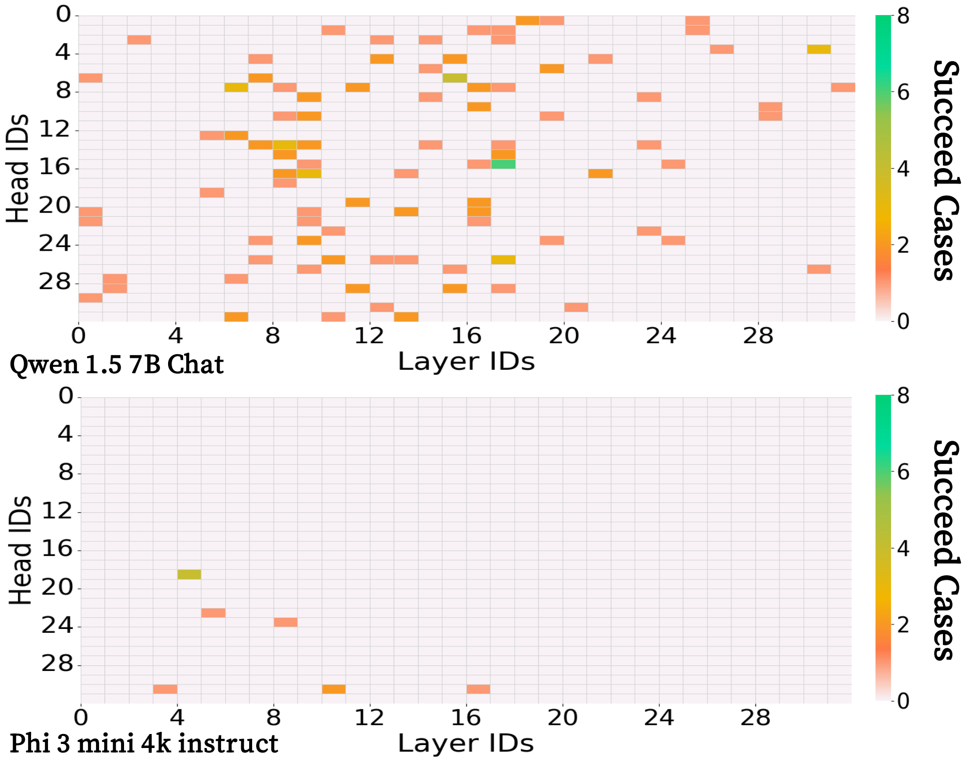

Figure 15: Result Of temporal knowledge editing in Qwen 1.5 7B Chat and Phi 3 mini 4k instruct. From the source prompt, we catch the attention value of each model's temporal head, **a17.h15** and **a10.h13**. The model's output is changed into temporally correct answer from temporally wrong answer. The headmap below denotes the number of success in editing for every combination of layers and heads. Though the most successful case of editing is the temporal head **a17.h15** in Qwen 1.5 7B Chat, Phi 3 mini 4k instruct shows that adding attention had minimal impact, and temporal heads failed to enable effective editing. This suggests that the model, constrained by its small parameter size (3.8B), requires a more sophisticated vector steering mechanism rather than relying on a single attention head value modification.