Time Course MechInterp: Analyzing the Evolution of Components and Knowledge in Large Language Models

Anonymous ACL submission

Abstract

Understanding how large language models (LLMs) acquire and store factual knowledge is crucial for enhancing their interpretability, reliability, and efficiency. In this work, we analyze the evolution of factual knowledge representation in the OLMo-7B model by tracking the roles of its Attention Heads and Feed Forward Networks (FFNs) over training. We classify these components into four roles—general, entity, relation-answer, and fact-answer specific—and examine their stability and transitions. Our results show that LLMs initially depend on broad, general-purpose components, which later specialize as training progresses. Once the model reliably predicts answers, some components are repurposed, suggesting an adaptive learning process. Notably, answer-specific attention heads display the highest turnover, whereas FFNs remain stable, continually refining stored knowledge. These insights offer a mechanistic view of knowledge formation in LLMs and have implications for model pruning, optimization, and transparency. (A repository link for reproducibility will be provided.)

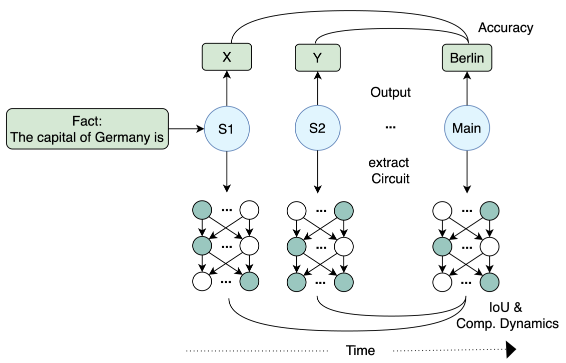

Figure 1: Factual Knowledge Probing. We track how Olmo-7B processes factual knowledge across snapshots by analyzing outputs, extracting information flow circuits, and comparing accuracy. Additionally, we examine component dynamics by tracking their counts, measuring Intersection over Union (IoU) between snapshots and the fully trained main model, and analyzing role switches over time.

1 Introduction

Large Language Models (LLMs) are trained on vast datasets including resources like Wikipedia imbuing them with extensive factual knowledge. As a result, these models can provide informed answers when queried about facts. To uncover the mechanisms that enable such factual responses, mechanistic interpretability (MI) methods (Olah et al., 2020; Elhage et al., 2021) are employed. MI aims to reverse-engineer neural networks by translating their internal processes into human-understandable algorithms and concepts, and has made great progress in explaining how transformer-based LLMs process and store information.

A key approach within MI is Circuit Analysis (Olah et al., 2020; Elhage et al., 2021; Wang et al., 2023), which isolates minimal computational subgraphs—comprising essential components like attention heads and FFNs—that reproduce a model's behavior on a given task. Prior work on factual recall has focused on localizing knowledge within transformer parameters (Meng et al., 2022; Geva et al., 2021, 2022, 2023; Hernandez et al., 2024) and on behavioral analyses that trace the emergence of linguistic and reasoning capabilities during pretraining (Rogers et al., 2020; Liu et al., 2021; Chiang et al., 2020; Chang et al., 2024; Xia et al., 2023; Hu et al., 2023; Biderman et al., 2023a). While these studies have advanced our understanding of factual knowledge from both internal and external perspectives, they have not systematically examined how the components of a factual recall circuit evolve over training.

In this work, we bridge this gap by conducting a time-course mechanistic interpretability study on training snapshots of the Olmo-7B model (Groeneveld et al., 2024). Specifically, we investigate:

- Which components (attention heads, FFNs) contribute to solving factual knowledge tasks?
- How do these circuits for factual knowledge evolve over the course of model training?

To achieve this, we trace information flow routes (Ferrando and Voita, 2024) in Olmo-7B using interpretability tools, analyze component dynamics, and identify which parts of the model (attention heads, layers, FFNs) encode and retrieve factual knowledge across different training snapshots. We make the following contributions:

- **Factual Knowledge Dataset Construction:** A pipeline for constructing a factual knowledge probing dataset that minimizes ambiguity. Using this pipeline, we create a new dataset specifically designed for analyzing factual knowledge in LLMs.
- **Component Attribution for Factual Knowledge:** We analyze which components are responsible for processing factual knowledge at different training stages.
- **Temporal Evolution of Knowledge Representation:** We track how circuits responsible for factual knowledge stabilize or change role over the course of training.

2 Background

2.1 Circuit Analysis

A *circuit* is defined as the minimal computational subgraph that faithfully reproduces a model's performance on a specific task (Olah et al., 2020; Elhage et al., 2021; Wang et al., 2023). Circuits isolate key components—such as attention heads and FFNs—that drive predictions. Various techniques extract these circuits, including activation patching (which selectively corrupts activations to assess performance impact), attribution-based methods (e.g., edge attribution patching (EAP) and its integrated gradients variant, EAP-IG (Hanna et al., 2024; Nanda et al., 2023)), and gradient-based approaches like integrated gradients (Sundararajan et al., 2017). However, these methods often become computationally prohibitive for large models or when evaluating multiple snapshots due to their complexity and memory demands.

2.2 Information Flow Routes

To overcome these limitations, we leverage Information Flow Routes (IFRs) (Ferrando and Voita, 2024). IFRs conceptualize the model as a computational graph and recursively trace pathways from the output token back through the network. At each step, only nodes and edges with contributions exceeding a threshold θ are retained, ensuring that only paths significantly impacting the final prediction are included. The importance of each edge is quantified using a modified ALTI (Aggregation of Layer-Wise Token-to-Token Interactions) score (Ferrando et al., 2022). Compared to traditional circuit-finding methods, IFRs are more scalable, require minimal prompt design, and are well-suited for large models like Olmo-7B across multiple training snapshots. Furthermore, IFRs sidestep challenges posed by self-repair mechanisms in LLMs (McGrath et al., 2023; Rushing and Nanda, 2024), making them a robust tool for circuit analysis.

3 Factual Knowledge Probing over Time

In this section, we describe our approach to probing factual knowledge. We first introduce our dataset (Sec. 3.1), then detail the OLMo-7B training snapshots used (Sec. 3.2), and finally assess snapshot performance via accuracy (Sec. 3.3).

Key Terms. A **fact** is defined as a subject-relation-object triple (e.g., (Canada, has_capital, Ottawa)), where **has_capital** is the **relation** representing the pairing of a country with its capital.

Token Positions. In our experiments, facts appear in sentences such as "Canada has the capital city of Ottawa." We distinguish three sets of subtoken positions: (i) **SUBJECT** for the subject (e.g., "Canada"); (ii) **END** for the subtoken immediately before the answer (e.g., "of" in "has the capital city of"); and (iii) **ANSWER** for the tokens forming the answer, beginning with the token following **END**.

3.1 Dataset

We develop a dataset designed to probe the factual knowledge encoded in the Olmo-7B model. See Table 1. To minimize syntactic ambiguity, we avoid templates that may lead to multiple valid answers; e.g., for the prompt *"The Eiffel Tower is located in"*, both *Paris* and *France* are correct. Similarly, we avoid cases involving regional variations in terminology (e.g., *soccer* vs. *football*) and eliminate instances where the answer is already contained in the subject (e.g., *"The Leaning Tower of Pisa is a landmark in the city of Pisa."*). Our focus is

Location-based Relations (LOC)			
Relation	**Prompt Template**	**# Facts**	**Example Subject**
CITY_IN_COUNTRY	{ } is part of the country of	14	Rio de Janeiro, Buenos Aires
COMPANY_HQ	The headquarters of { } are in the city of	20	Zillow, Bayrischer Rundfunk
COUNTRY_CAPITAL_CITY	{ } has the capital city of	19	Canada, Nigeria
FOOD_FROM_COUNTRY	{ } is from the country of	17	Sushi, Ceviche
OFFICIAL_LANGUAGE	In { }, the official language is	14	France, Egypt
PLAYS_SPORT	{ } plays professionally in the sport of	12	Kobe Bryant, Roger Federer
SIGHTS_IN_CITY	{ } is a landmark in the city of	17	The Eiffel Tower, The Space Needle
Name-based Relations (NAME)			
Relation	**Prompt Template**	**# Facts**	**Example Subject**
BOOKS_WRITTEN	The Book { } was written by the author with the name of	13	The Hunger Games, Life of Pi
COMPANY_CEO	Who is the CEO of { }? Their name is	17	Ubisoft, Pinterest
MOVIE_DIRECTED	The Movie { } was directed by the director with the name of	17	The Godfather, Forrest Gump

Table 1: Overview of the Factual Knowledge dataset, grouped by relation type.

on categorical facts associated with well-defined relation types, specifically **Location-Based Relations** (LOC) and **Name-Based Relations** (NAME). Table 1 provides an overview of these relations, along with the prompt templates, number of facts, and example subjects for each relation type. Although manually curated, our dataset is inspired by existing resources such as **LRE** (Hernandez et al., 2024), **CounterFact** (Meng et al., 2022), and **ParaRel** (Elazar et al., 2021), as well as **Summing Up The Facts** (Chughtai et al., 2024). We extended these resources by integrating relations such as `BOOKS_WRITTEN` and `MOVIE_DIRECTED` using data from Goodreads and IMDb's Top Favorites list. To ensure reliability and eliminate potential confounds in our analysis, we implement a multi-step validation pipeline to rigorously evaluate both the prompts and the facts (see Appendix B).

3.2 Models

We study the evolution of factual knowledge using the **OLMo-7B** model (Groeneveld et al., 2024),[1] a flagship open-source LLM with 32 layers (each with 32 attention heads) pretrained on over 2.5 trillion tokens. During training, checkpoints were saved every 500 steps (2B tokens per interval) from initialization up to `step161000-tokens675B`, and then at 1000-step intervals until the final checkpoint, `step651581-tokens2731B`. For our analysis, we select 40 snapshots spanning from `step5000-tokens20B` to `step200000-tokens838B` (in 5000-step increments), along with the fully trained main model.[2] We denote these snapshots as SX-YB, where X represents the training step (in multiples of 5000) and Y the token count in billions.

[1] `https://huggingface.co/allenai/OLMo-7B-0424-hf/tree/main`

[2] Due to an issue with `step115000`, we use `step115500-tokens462B` instead.

3.3 Accuracy

Since we are interested in the time course of factual knowledge during training, we first establish how the acquisition of knowledge evolves as measured by top-1 and top-10 accuracy on the first token of ANSWER. We group our relations into two groups: NAME (the answer is the name of a person) and LOC (the answer is a location). See Appendix C for per-relation graphs.

Figure 2 shows that LOC relations converge faster than NAME relations: A top-1 accuracy of 0.8 is first reached at S5 for LOC and at S14 for NAME. LOC also has less top-1 volatility than NAME. top-10 accuracy for NAME is also lower than for LOC, but top-10 values are much higher, starting at about S13. This indicates that fairly early on, the correct NAME is in the pool of candidates that the model has identified as relevant and that the remaining problem of knowledge acquisition is then correct ranking. The likely reason for these differencec between NAME and LOC is that there are many more prominent person names than prominent locations in the model's training data, making it more challenging to learn the correct answer for a person than for a location.

4 How do Components Evolve?

We now examine Olmo-7B's internal mechanics during pretraining. Using IFR, we trace the full circuit behind each predicted token to identify the contributing components, i.e., attention heads and FFNs. We classify these components based on their roles in the circuit, distinguishing generalized components that contribute broadly from specialized

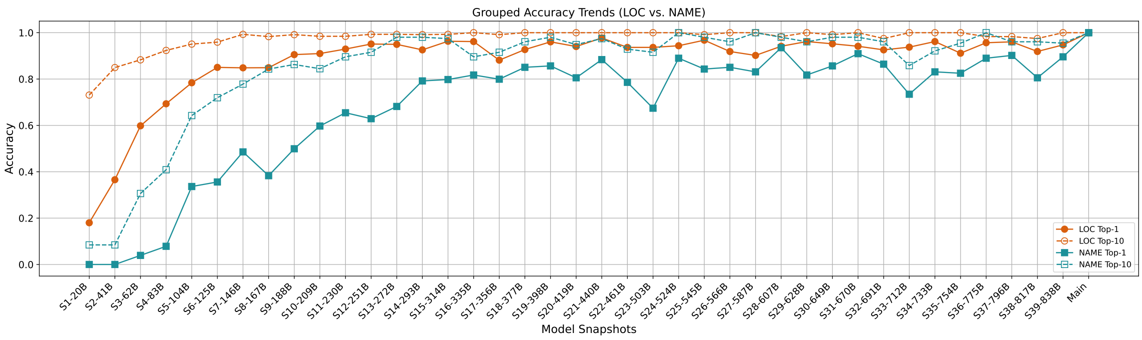

Figure 2: Accuracy of LOC and NAME relations across snapshots.

ones with more focused functions. By tracking how these roles evolve during pretraining, we gain deeper insights into the model's learning dynamics for factual knowledge.

4.1 Model Component Roles

We take a systematic approach to defining model component roles based on their contributions within token circuits. These roles are determined by the types of tokens a component influences and the scope of its contribution. A component may contribute to all tokens or to a subset, to only one fact or multiple facts etc. We now define a structured classification schema that captures the functional behavior of each component.

For each snapshot s, relation r, fact f and subtoken position t, we use IFR to compute the circuit that produced the output subtoken at that t. We set $c_{srft} = 1$ if component c is part of the circuit, 0 otherwise.

We then define $c_{srf}(T) = \frac{1}{|T(f)|} \sum_{t \in T(f)} c_{srft}$, i.e., c_{srf} is the activation of c averaged over the subtoken positions $T(f)$. Given the sentence corresponding to fact f, $T(f)$ is a subset of its subtokens. We now define different roles of a component by defining $T(f)$ differently, e.g., containing only the ANSWER subtokens or all subtokens.

4.1.1 General role

For the general role, we use the subtoken selector $T_g(f)$. $T_g(f)$ is the set of all subtokens of the sentence (except for the final period).

We define the general activation score of a component c for snapshot s as:

$$c_s^g = \frac{\sum_{r \in R} \sum_{f \in r} c_{srf}(T_g)}{\sum_{r \in R} \sum_{f \in r} 1}$$

That is, c_s^g is the activation of c for snapshot s, microaveraged over facts. R is the set of relations.

We classify a component c as having a **general role** for snapshot s if $c_s^g > \theta$, where we set $\theta = 0.1$.

4.1.2 Entity role

For the entity role, we use the subtoken selector $T_e(f)$. $T_e(f)$ is the set of all subtokens of SUBJECT and ANSWER.[3]

We define the entity activation score of a component c for snapshot s as:

$$c_s^e = \frac{\sum_{r \in R} \sum_{f \in r} c_{srf}(T_e)}{\sum_{r \in R} \sum_{f \in r} 1}$$

That is, c_s^e is the subject and answer activation of c for snapshot c, microaveraged over facts.

We classify a component c as having an **entity role** for snapshot s if $c_s^e > \theta$ where $\theta = 0.1$.

4.1.3 Relation-answer specific role

For the relation role, we use the subtoken selector $T_a(f)$. $T_a(f)$ selects the subtokens of the ANSWER. We then define the relation-answer activation score of a component c for snapshot s and relation r as:

$$c_s^r = \frac{\sum_{f \in r} c_{srf}(T_a)}{\sum_{f \in r} 1}$$

That is, c_s^r is the answer activation of c for snapshot s and relation r, averaged over facts.

We classify a component c as having a **relation-answer role** if $c_s^r > \theta$ where $\theta = 0.1$.

4.1.4 Fact-answer specific role

For a fact f belonging to relation r, we set $c_s^f = c_{srf}^f(T_a)$.

We classify a component c as having a **fact-answer role** if $c_s^f > \theta$ where $\theta = 0.1$.

[3] For the SUBJECT, there is no helpful context for the prediction of its first subtoken, e.g., for "France" in "France has the capital ...". We therefore shift the subtokens considered to the right by 1 for SUBJECTS.

4.1.5 Proper Components

Let $\mathcal{J}_g$, $\mathcal{J}_e$, $\mathcal{J}_r$ and $\mathcal{J}_f$ be the sets of components that assume the general, entity, relation-answer, and fact-answer roles, respectively, as defined above.

We define the set of proper entity components $\mathcal{H}_e$ as those components that assume an entity role but not a general role:

$$\mathcal{H}_e = \mathcal{J}_e - \mathcal{J}_g$$

We define the set of proper relation-answer components $\mathcal{H}_r$ as those components that assume a relation-answer role but not an entity or general role:

$$\mathcal{H}_r = \mathcal{J}_r - \mathcal{J}_e - \mathcal{J}_g$$

We define the set of proper fact-answer components $\mathcal{H}_f$ as those components that assume a fact-answer role but not a general, entity, or relation-answer role:

$$\mathcal{H}_f = \mathcal{J}_f - \mathcal{J}_r - \mathcal{J}_e - \mathcal{J}_g$$

We set:

$$\mathcal{H}_g = \mathcal{J}_g$$

because for the general role, there is no change from the original set to the proper set.

Finally, in addition to the sets of components $\mathcal{H}_g$, $\mathcal{H}_e$, $\mathcal{H}_r$, and $\mathcal{H}_f$, we define the set of deactivated components $\mathcal{H}_d$ as the complement of the union of the four other roles g, e, r, f: $\mathcal{H}_d = \mathcal{C}(\mathcal{H}_g \cup \mathcal{H}_e \cup \mathcal{H}_r \cup \mathcal{H}_f)$.

4.2 Analysis of Component Dynamics

After classifying components into five distinct roles (general, entity, relation-answer, answer-specific, deactivated), we analyze how these components evolve during pretraining, quantifying both static and dynamic aspects of the roles. We now describe our methodology including the measures we use.

Consistency and Count Metrics: To quantify stability, we measure the consistency of each role over time by calculating the Jaccard Similarity (Intersection over Union, or IoU) between the set of components with a particular role at a given snapshot and the corresponding set in the final model. For example, for general components, the IoU is defined as:

$$\text{IoU}(\mathcal{H}_g) = \frac{\mathcal{H}_{gs} \cap \mathcal{H}_{gmain}}{\mathcal{H}_{gs} \cup \mathcal{H}_{gmain}},$$

where $\mathcal{H}_{gs}$ represents the set of entity components in the current snapshot, and $\mathcal{H}_{gmain}$ is the corresponding set in the final model.

Role Switch Dynamics: We also track how components change roles over time: whether they activate, deactivate, or switch functions. By computing the accumulated switch counts across selected snapshots (S1, S10, S20, S40, and the main model), we capture the dynamics of these transitions, such as deactivated components reactivating in specialized roles or components switching between different roles.

Markov Chain Modeling of Transitions: To further characterize role dynamics, we model transitions using a Markov chain. The transition probability from state $\mathcal{H}_\alpha$ to state $\mathcal{H}_\beta$ is given by:

$$P(\mathcal{H}_\alpha \to \mathcal{H}_\beta) = \frac{N(\mathcal{H}_\alpha \to \mathcal{H}_\beta)}{\sum_{\gamma \in \{g,e,r,f,d\}} N(\mathcal{H}_\alpha \to \mathcal{H}_\gamma)},$$

where $N(\mathcal{H}_\alpha \to \mathcal{H}_\beta)$ is the number of observed transitions from one state to the following state.

4.3 Temporal Consistency and Role Dynamics of Attention Heads

Our analysis reveals key trends in the evolution of attention heads. Since differences between LOC and NAME relations are marginal (see Appendix D), we combine them for the subsequent analysis.

Using the IoU metric and component counts, we observe that the number of active attention heads increases steadily over the course of training. For instance, the counts for general heads rise from 94 to 233, for relation-answer heads from 8 to 78, and for answer-specific heads from 11 to 99. Overall, the total number of active heads grows from 113 to 423—rising from approximately 11% to 41% of all heads—while nearly 60% remain deactivated. Figure 3 illustrates these dynamics, with the IoU metric confirming that general heads maintain a high consistency with the final model throughout training.

The evolution of attention head roles suggests a hierarchical learning process. Early in training, the model primarily relies on general-purpose heads that generate broad, context-independent representations. As training progresses, specialized heads emerge to support more precise fact retrieval. Notably, answer-specific heads demonstrate the highest turnover, indicating frequent role changes and dynamic reallocation of resources. Furthermore, our observations indicate that tasks involving complex, name-based relations require longer training

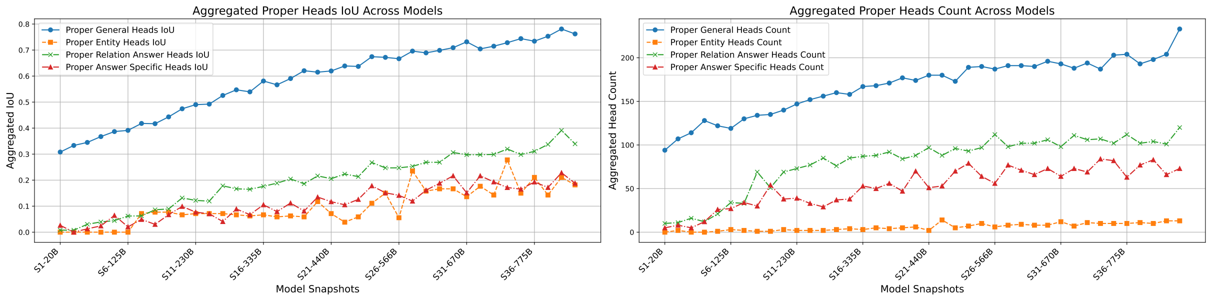

Figure 3: Aggregated Head Count and IoU Across Olmo-7B Snapshots Left: Counts for general, entity, relation-answer, and answer-specific categories over snapshots. Right: IoU values comparing each snapshot to main.

periods and exhibit more frequent role transitions compared to simpler, location-based tasks.

Dynamic Specialization and Generalization of Attention Heads Our analysis reveals that attention heads frequently transition from deactivated to specialized roles—especially to answer-specific roles (see Fig. 4). In contrast, general heads are more stable or shift to relation-answer roles. A heatmap of role transitions (see Fig. 5) shows that early and late layers switch frequently, whereas middle layers (10–18) are more stable. Notably, NAME-based tasks prompt more activations and transitions than LOC-based tasks in these layers, reflecting the greater complexity of name-based relations, suggesting that the increased complexity of NAME tasks demands a higher degree of dynamic reallocation. See Appendix E for details.

Our Markov chain modeling (see Fig. 6) further quantifies these dynamics: specialized heads tend to transition toward more general roles, and once deactivated, they rarely reactivate. Although individual specialized heads often shift into general roles, the overall count of specialized heads increases over time because the rate at which new specialized heads emerge exceeds the rate at which they generalize. In sum, while specialized heads tend to generalize, they are continually replenished—resulting in a net growth in the total number of active heads. This dynamic has significant implications for both model interpretability and pruning.

Overall, the results suggest that while attention heads rapidly establish a stable general foundation, dynamic specialization occurs later to meet the demands of complex factual retrieval. The contrasting behaviors between general and specialized heads highlight the delicate balance between flexibility and stability in model architecture.

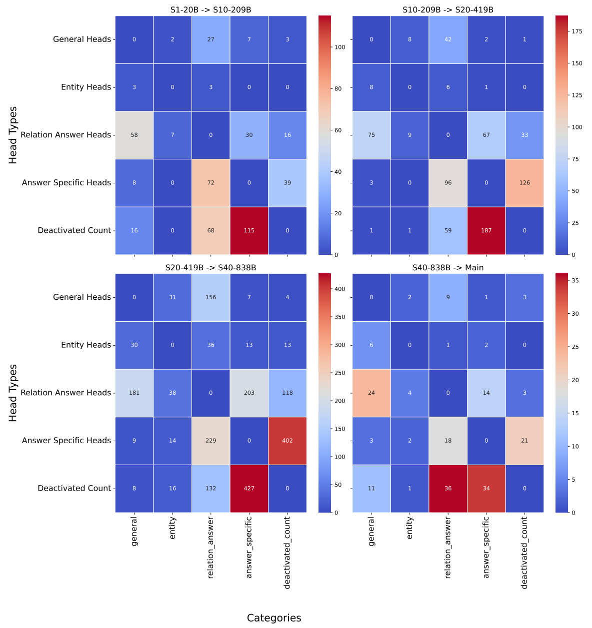

Figure 4: Accumulated Attention Head Switches Across Training Stages. Heatmaps showing the total number of transitions between the four types of heads and the deactivated state at key training snapshots (S1, S10, S20, S40, and Main).

4.4 FFNs over Time

Analogous to our attention head classification, we assign FFNs to four roles (general, entity, relation-answer, and answer-specific), though with a higher activation threshold ($\theta = 0.90$) as suggested in (Ferrando et al., 2022). Unlike the 1024 attention heads, the model uses only 32 FFNs (one per layer), and all actively contribute to answer generation.

Steady Backbone: Consistency and Activation Trends Figure 7 shows that early on, most FFNs serve as general components, with only a few operating in relation-answer or answer-specific roles. Around stages S7–S8, when accuracy exceeds 80%, many general FFNs shift to relation-answer roles. Over time, the role distribution oscillates, as indi-

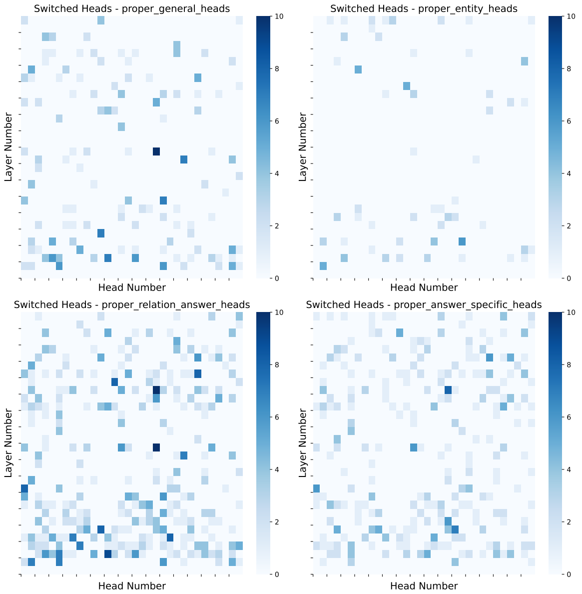

Figure 5: Attention Head Role Transitions. Per-layer heatmaps showing the frequency that a head from one of the four roles general, entity, relation-answer, and answer-specific switches to a different role.

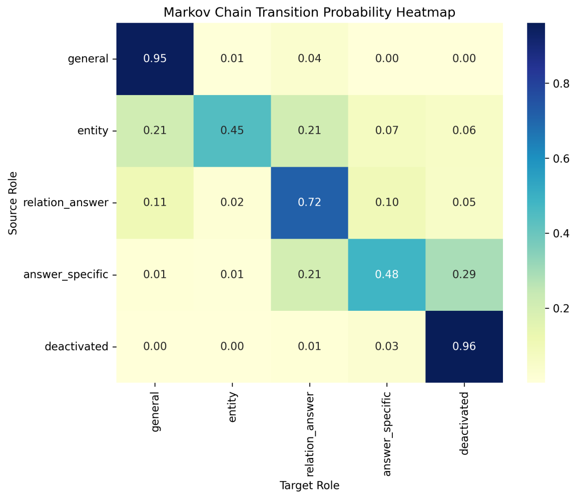

Figure 6: Markov Chain Transition Probability Heatmap. Heatmap showing the transition probabilities between different attention head roles across model snapshots. Each cell represents the probability of a head transitioning from a source role (rows) in snapshot i to a target role (columns) in snapshot $i + 1$.

cated by an IoU of about 0.5.

Oscillatory Dynamics in Role Allocation Although the majority of FFNs remain general, we observe occasional role oscillations. For easier LOC relations, answer-specific FFNs exhibit minimal switching, whereas for the more challenging NAME relations, a small number of FFNs gradually transition into answer-specific roles before reverting to general roles in subsequent pretraining steps. The total switch count and transition probability analyses (see Appendix F) suggest that general FFNs primarily shift among themselves and rarely become permanently specialized.

FFNs as General Processing Components In contrast to the dynamic specialization observed in attention heads, FFNs exhibit notable stability, predominantly refining the representations generated by the attention mechanisms with only minor role transitions. While this consistent performance supports the view of FFNs as a robust backbone for maintaining factual accuracy, it is important to consider that such generality might partly stem from their large size. Essentially, when analyzing a sufficiently large component of any network module, the observed generality could be an artifact of scale.

5 Related Work

This section reviews prior work on mechanistic interpretability, model behavior evolution, and how transformers store and retrieve factual knowledge. While past research has deepened our understanding of fully trained models, less focus has been given to how these mechanisms evolve during training—a gap this work addresses.

5.1 Mechanistic Interpretability

Mechanistic Interpretability aims to reverse-engineer neural networks to uncover circuits driving model behavior. Early work (Elhage et al., 2021; Olah et al., 2020) focused on vision models and has since extended to transformer language models (Meng et al., 2022; Wang et al., 2023; Hanna et al., 2023; Varma et al., 2023; Merullo et al., 2024; Lieberum et al., 2023; Tigges et al., 2023; Mondorf et al., 2024; Tigges et al., 2024). Research has characterized attention heads (Olsson et al., 2022; Chen et al., 2024; Singh et al., 2024; Gould et al., 2024; McDougall et al., 2023; Chughtai et al., 2024; Elhelo and Geva, 2024; Ortu et al., 2024) and FFNs (Geva et al., 2021; Meng et al., 2022; Bricken et al., 2023; Neo et al., 2024; Tian et al., 2024).

5.2 Interpretability Over Time

Behavioral studies have tracked the emergence of linguistic and reasoning capabilities during pre-

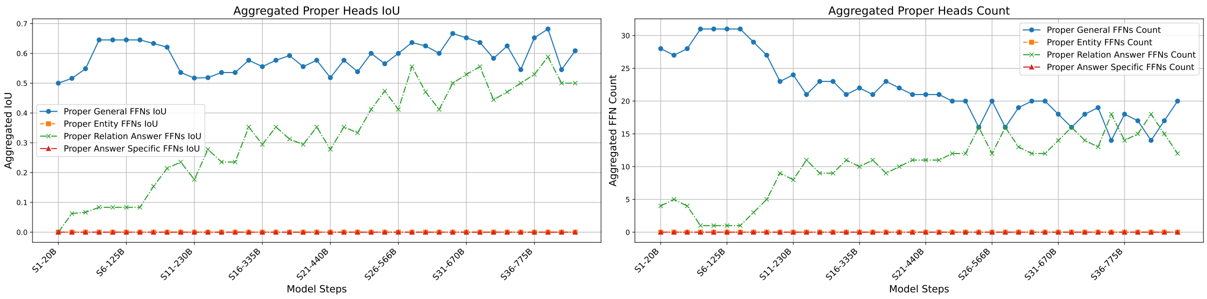

Figure 7: Aggregated FFN Count and IoU Across Olmo-7B Snapshots Left: Counts for general, entity, relation-answer, and answer-specific categories over snapshots. Right: IoU values comparing each snapshot to main.

training (Rogers et al., 2020; Liu et al., 2021; Chiang et al., 2020; Müller-Eberstein et al., 2023; Xia et al., 2023; Chang et al., 2023; Hu et al., 2023; Biderman et al., 2023a). Yet, they offer limited insight into internal circuit evolution. Recent work on smaller models shows that internal circuits can change abruptly even when overall behavior is stable (Nanda et al., 2023; Olsson et al., 2022; Chen et al., 2024) and mechanistic studies have begun tracking circuit evolution (Tigges et al., 2024).

5.3 Mechanisms of Knowledge Storage in Transformers

Studies have shown that transformers store factual knowledge in (subject, relation, attribute) tuples. Causal interventions reveal that early-to-middle FFNs enrich subject representations, while attention heads pass relation information and later layers extract attributes (Meng et al., 2022; Geva et al., 2023). Complementary work demonstrates that these representations can be decoded to recover facts (Hernandez et al., 2024; Chughtai et al., 2024). Other research highlights the balance between in-context and memorized recall (Yu and Ananiadou, 2024; Variengien and Winsor, 2023) and the distributed nature of knowledge retrieval (Haviv et al., 2023; Stoehr et al., 2024; Chuang et al., 2024).

While previous work has focused on static models, we track the evolution of these mechanisms during training, offering a dynamic view of factual knowledge development in LLMs.

6 Discussion & Conclusion

Our study reveals two complementary dynamics in Olmo-7B. Attention heads evolve from stable, general-purpose units into specialized components for complex relational tasks—general heads remain stable, while answer-specific heads exhibit high turnover and irreversible shifts. In contrast, FFNs appear to remain relatively stable, seemingly operating as general processors that refine the representations generated by attention mechanisms. While our observations hint at a complementary dynamic where attention heads adapt to capture task-specific nuances and FFNs offer a consistent foundation for refinement these results should be interpreted with caution.

In summary, our key findings are:

1. **Task Complexity Influences Training Dynamics:** Location-based relations are acquired more rapidly and stably than name-based relations, which require more specialized components.
2. **Hierarchical Learning Process:** Early training is dominated by stable, general attention heads that lay the groundwork for subsequent specialization.
3. **Adaptive vs. Stable Components:** Our analysis indicates that certain attention heads may be repurposed dynamically particularly those associated with answer-specific roles while FFNs tend to exhibit a more stable behavior. These observations hint at a possible complementary dynamic between adaptable attention mechanisms and stable processing components.
4. **Irreversible Specialization:** In later stages, the model stabilizes into a configuration where general heads prevail, and deactivated heads rarely reactivate.

These insights advance our understanding of MI by showing that dynamic specialization in attention heads supported by consistent FFN refinement underpins effective factual knowledge retrieval. Future work may explore neuron-level dynamics, assess redundancy among head roles, and examine scalability in larger models.

7 Limitations

Despite our comprehensive analysis, several limitations remain.

- **Computational Constraints:** Due to resource limitations, we could not extend our analysis to the neuron level, potentially missing finer-grained switching behaviors. Additionally, our role classification relies on fixed activation thresholds, which may introduce bias.
- **Model Checkpoints & Variants:** We analyzed a subset of training snapshots, leaving gaps in tracking role transitions. Further newer versions of the model were released during this study, but incorporating them was infeasible. Comparing with related models like Pythia (Biderman et al., 2023b) could provide additional insights.
- **Dataset Scope & Generalizability:** Our dataset focuses on factual recall in English, covering only location-based and name-based relations. Expanding to other domains, multilingual settings, and ambiguous queries would improve generalizability.
- **Interpretability Framework:** While IFRs efficiently trace knowledge circuits, they may overlook subtle interactions. Future work should compare IFR-based findings with alternative methods like activation patching and causal tracing.
- **Model Adaptability & Downstream Implications:** While attention heads frequently transition roles, FFNs remain stable, but their long-term impact on fine-tuning, pruning, and continual learning is unclear. Investigating their adaptability could enhance optimization strategies.

Future work should address these limitations by incorporating more diverse datasets, additional model variants, and alternative interpretability techniques to deepen our understanding of knowledge formation in LLMs.

References

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2023a. Emergent and predictable memorization in large language models. 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*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023b. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. How do large language models acquire factual knowledge during pretraining? In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. 2023. Characterizing learning curves during language model pre-training: Learning, forgetting, and stability. *CoRR*, abs/2308.15419.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L. Leavitt, and Naomi Saphra. 2024. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

David Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. 2020. Pretrained language model embryology: The birth of ALBERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6813–6828. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International*

Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard H. Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Trans. Assoc. Comput. Linguistics*, 9:1012–1031.

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

Amit Elhelo and Mor Geva. 2024. Inferring functionality of attention heads from their parameters. *CoRR*, abs/2412.11965.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8698–8714. Association for Computational Linguistics.

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, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 17432–17445. Association for Computational Linguistics.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12216–12235. Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 30–45. 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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2024. Successor heads: Recurring, interpretable attention heads in the wild. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Dirk Groeneveld, Iz Beltagy, Evan Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Olmo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15789–15809. Association for Computational Linguistics.

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*.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. *CoRR*, abs/2403.17806.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 248–264. Association for Computational Linguistics.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Michael Y. Hu, Angelica Chen, Naomi Saphra, and Kyunghyun Cho. 2023. Latent state models of training dynamics. *Trans. Mach. Learn. Res.*, 2023.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir

Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *CoRR*, abs/2307.09458.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021. Probing across time: What does roberta know and when? In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 820–842. Association for Computational Linguistics.

Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2023. Copy suppression: Comprehensively understanding an attention head. *CoRR*, abs/2310.04625.

Thomas McGrath, Matthew Rahtz, János Kramár, Vladimir Mikulik, and Shane Legg. 2023. The hydra effect: Emergent self-repair in language model computations. *CoRR*, abs/2307.15771.

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*.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Philipp Mondorf, Sondre Wold, and Barbara Plank. 2024. Circuit compositions: Exploring modular structures in transformer-based language models. *CoRR*, abs/2410.01434.

Max Müller-Eberstein, Rob van der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 13190–13208. Association for Computational Linguistics.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Clement Neo, Shay B. Cohen, and Fazl Barez. 2024. Interpreting context look-ups in transformers: Investigating attention-mlp interactions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 16681–16697. Association for Computational Linguistics.

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.

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. *CoRR*, abs/2209.11895.

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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8420–8436. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8:842–866.

Cody Rushing and Neel Nanda. 2024. Explorations of self-repair in language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. 2024. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Niklas Stoehr, Mitchell Gordon, Chiyuan Zhang, and Owen Lewis. 2024. Localizing paragraph memorization in language models. *CoRR*, abs/2403.19851.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Yuandong Tian, Yiping Wang, Zhenyu Zhang, Beidi Chen, and Simon Shaolei Du. 2024. Joma: Demystifying multilayer transformers via joint dynamics of MLP and attention. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. LLM circuit analyses are consistent across training and scale. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024,* *NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *CoRR*, abs/2310.15154.

Alexandre Variengien and Eric Winsor. 2023. Look before you leap: A universal emergent decomposition of retrieval tasks in language models. *CoRR*, abs/2312.10091.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. 2023. Explaining grokking through circuit efficiency. *CoRR*, abs/2309.02390.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru, Danqi Chen, Luke Zettlemoyer, and Veselin Stoyanov. 2023. Training trajectories of language models across scales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13711–13738. Association for Computational Linguistics.

Zeping Yu and Sophia Ananiadou. 2024. Neuron-level knowledge attribution in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 3267–3280. Association for Computational Linguistics.

A Implementation Details

All datasets are in English. We employed AI assistants to improve the visual appeal and readability of both our data visualizations and certain sections of the text. Our setup involved an NVIDIA RTX A6000 alongside eight NVIDIA HGX A100-80x4-mig GPUs, which were used for inferring OLMo and extracting the circuits detailed in this work. Reproducing our full analysis and experiments takes about 24 hours for OLMo-7B using these eight GPUs.

B Dataset Construction Pipeline

1. **Prompt Template Design and Fact Collection**: For each of the 10 relations, we compiled 10 prompt templates. These prompts were paired with factual examples to serve as inputs for model evaluation.

2. **Template Evaluation and Selection**: We tested all prompt templates with various factual inputs and determined the best-performing one for each relation. The evaluation was based on:

   - The **average probability** of the facts where the **first token** is correct.
   - The **reliability score** of the **second token**, which is calculated as the ratio of valid tokens for the second token (tokens with a probability less than 10%) divided by the total amount of facts.

   Prompts were ranked based on a combined score derived from the average probability of the first token and the reliability of the second token. This scoring ensured that the prompts produced semantically accurate outputs, not merely syntactic completions.

3. **Fact Reliability Validation**: Using the best-performing template for each relation, reliable facts were identified by ensuring that:

   - The **top-1 token** is correct with a probability above 75%.
   - The **second token** has a probability below 10%.

   This approach reduced reliance on syntactic biases and confirmed the semantic validity of the model's predictions.

4. **Final Dataset Generation**: For each relation, the dataset was finalized by pairing the best-performing prompt template with the set of validated, reliable facts.

The resulting dataset includes 160 facts over 10 relations, each with a single best-performing prompt template and a curated collection of reliable facts validated for high accuracy and consistency. Prompts and Facts were validated using the main model to establish a reliable baseline for tracking knowledge evolution.

To ensure the robustness of our dataset, we prioritized semantically meaningful continuations over syntactic ones by evaluating both the first and second token probabilities. A strict scoring framework ensured that the top-1 token accurately reflected the correct answer while minimizing interference from alternative tokens. By combining insights from prior datasets with meticulous manual curation, we created a high-quality resource for probing factual knowledge.

C Accuracy Plots per Relation

To complement the aggregated accuracy results in Section 3.3, we present relation-level accuracy trends for Top-1 and Top-10 metrics across different model snapshots. These plots illustrate that NAME-based relations require significantly more training to achieve high accuracy compared to LOC relations (Figures 8 and 9). Among NAME relations, MOVIE_DIRECTED is the most challenging, requiring approximately S10 to reach high Top-10 accuracy, while COMPANY_CEO and BOOKS_WRITTEN also exhibit slower convergence. In contrast, LOC relations such as PLAYS_SPORT and CITY_IN_COUNTRY are learned much faster.

For Top-1 accuracy, OFFICIAL_LANGUAGE is the first relation to reach 100%, achieving this milestone at S4, whereas in the Top-10 metric, it already attains 100% as early as S1. This suggests that while correct answers are recognized among the top candidates from the beginning, ranking them correctly requires additional training.

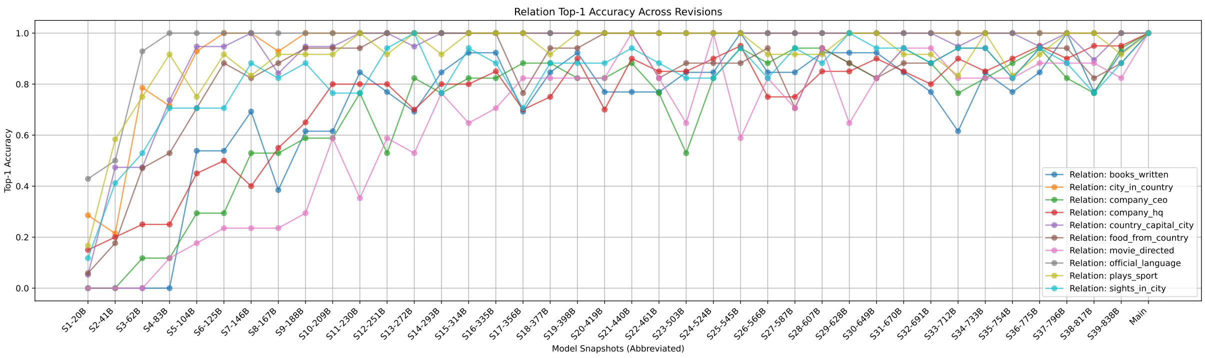

Figure 8: Top-1 accuracy across different revisions of the Olmo model. Snapshots (S_X-YB) represent training checkpoints taken at 5000-step intervals, where Y indicates the number of tokens processed in billions.

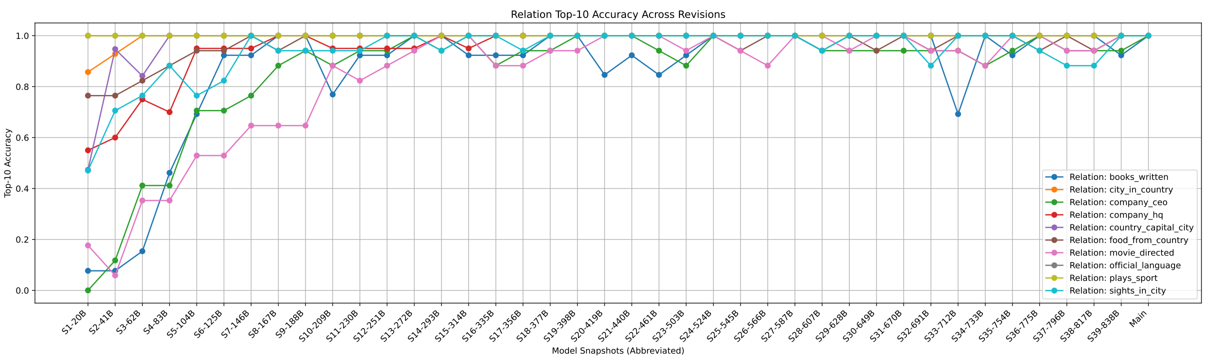

Figure 9: Top-10 accuracy across different revisions of the Olmo model. Snapshots (S_X-YB) represent training checkpoints taken at 5000-step intervals, where Y indicates the number of tokens processed in billions.

D Relation-Level Component Counts and IoU Values

D.1 Attention Heads

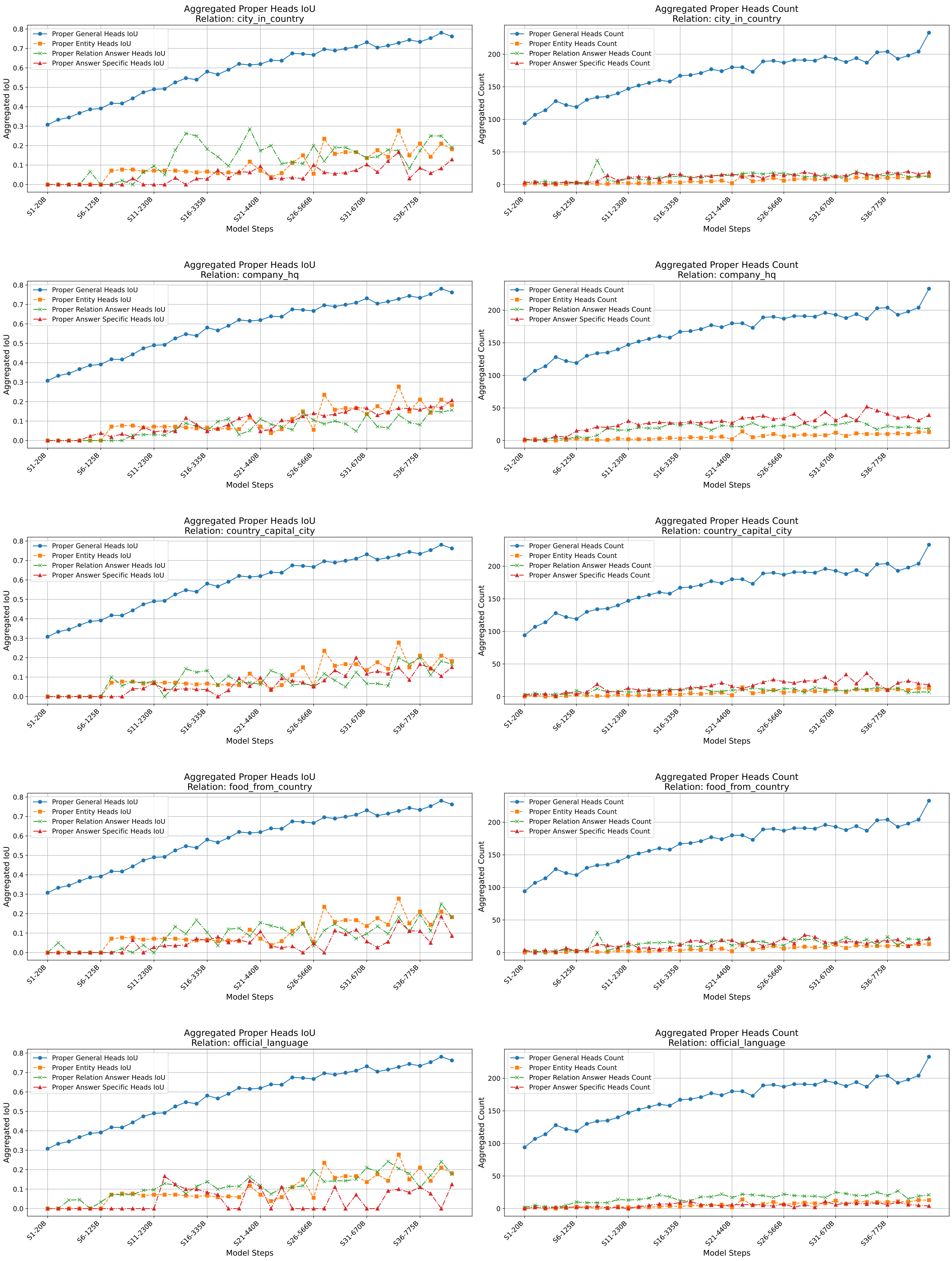

Figure 10: Relation-level head counts and IoU values.

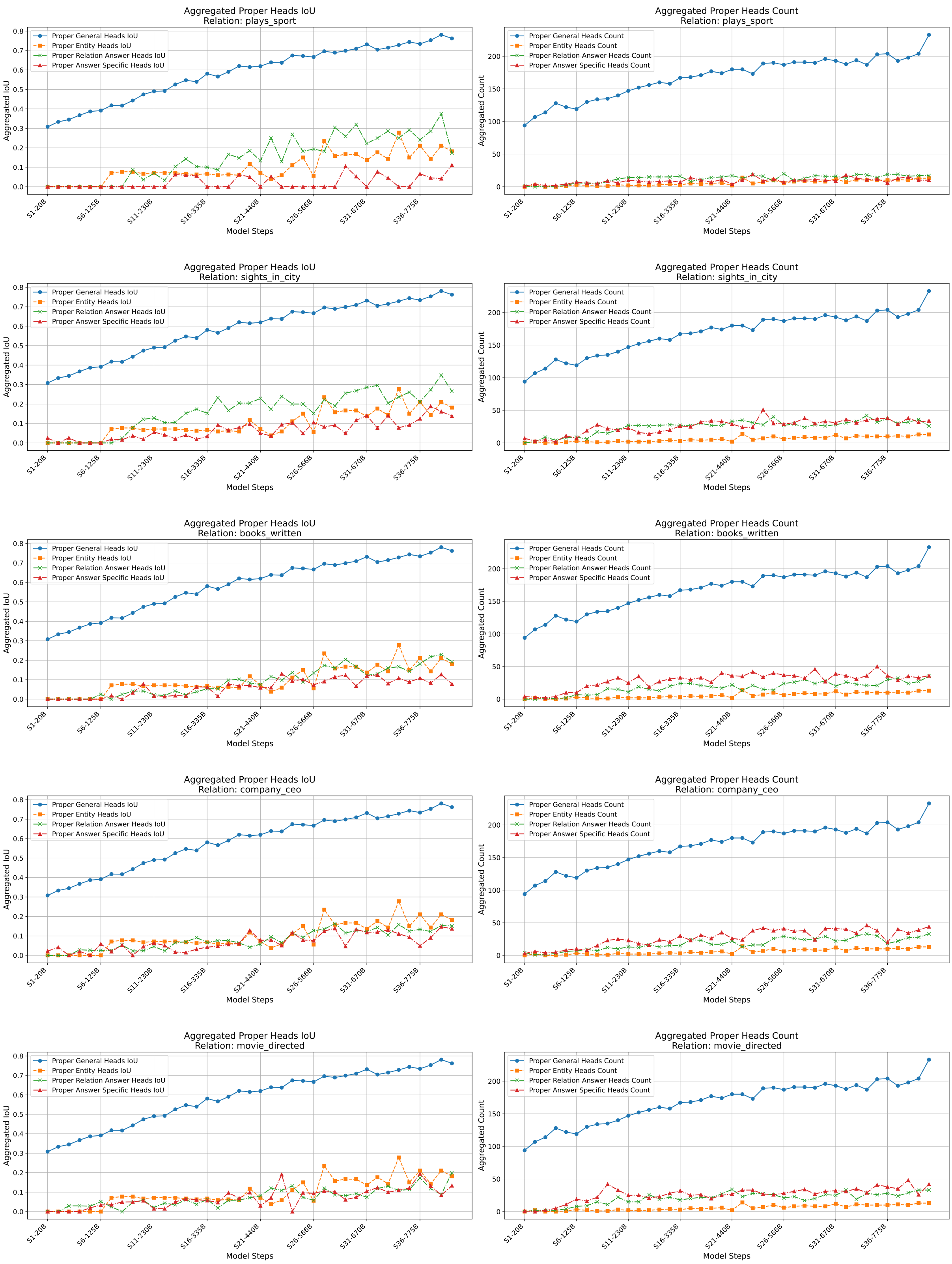

Figure 10: (continued) Relation-level head counts and IoU values.

D.2 Feed Forward Networks

Figure 11: Relation-level FFN counts and IoU values.

Figure 11: (continued) Relation-level FFN counts and IoU values.

E Attention Head Switches

In the following two plots, we observe that, as seen in the aggregated figure, layers 10–18 exhibit fewer switches, while switches occur more frequently in the early (0–10) and late (18–31) layers. However, when examining transitions between relation-answer and answer-specific roles, a clear distinction emerges: NAME-based relations (Fig. 13) show significantly more switches in layers 10–18 compared to LOC-based relations (12). Additionally, NAME-based tasks involve a greater number of distinct attention heads during these transitions. A switch refers to the reallocation of an attention head from one role to another among the four predefined roles.

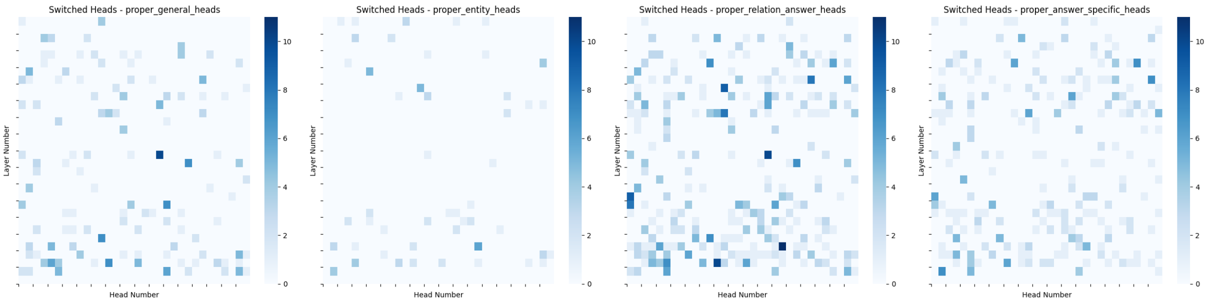

Figure 12: Accumulated head switches for LOC relations, independent of switch type.

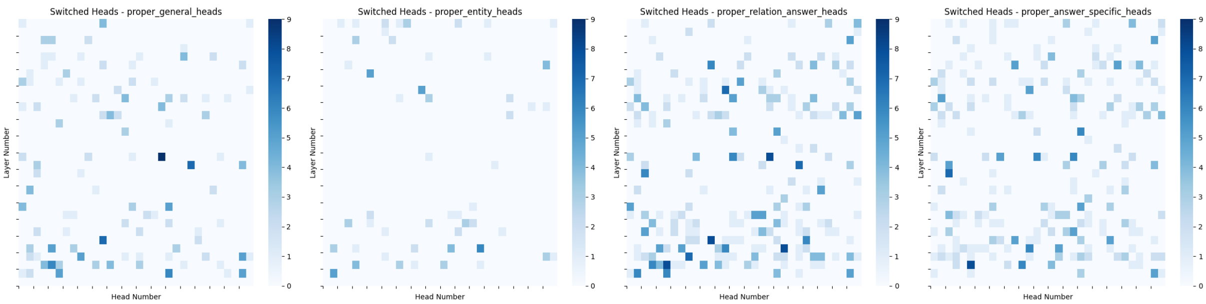

Figure 13: Accumulated head switches for NAME relations, independent of switch type.

F FFN Role Transition Count and Transition Probability

The following three figures present the metrics and methods used to assess component dynamics. As shown in Figure 14, very few switches occur overall, with most transitions happening between general FFNs and relation-answer FFNs. Examining the transition probabilities (see Fig. 15) from our Markov chain analysis, we find that both general and relation-answer FFNs tend to remain in their current roles with high probability; when switches do occur, they are predominantly between these two roles. Additionally, a layer-wise analysis (see Fig. 16) reveals a stark contrast with attention heads: starting from the middle layers onward, FFN role switches are nearly absent, and their roles become firmly established. Entity and answer-specific FFNs exhibit minimal switching across all layers.

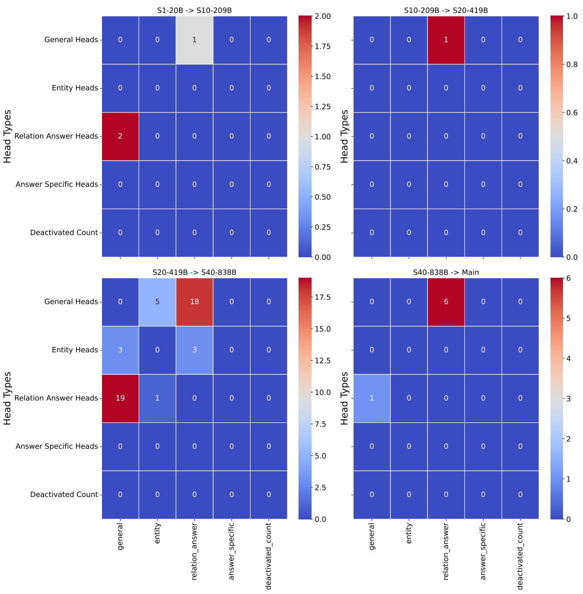

Figure 14: FFN Role Transitions. Heatmaps showing the frequency of role switches among proper general, entity, relation-answer, and answer-specific FFNs across layers.

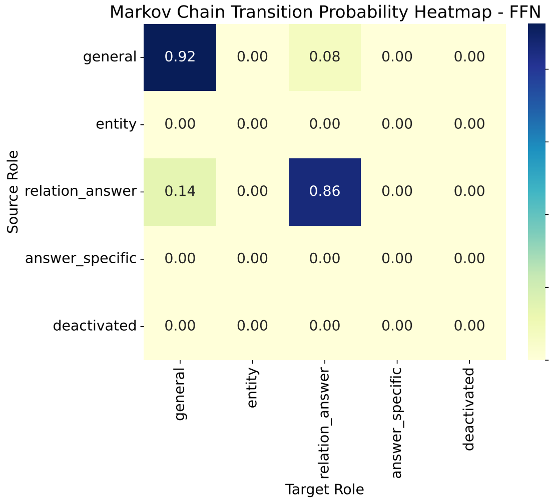

Figure 15: Markov Chain Transition Probability Heatmap Heatmap showing the transition probabilities between different FFNs roles across model snapshots. Each cell represents the probability of a FFN transitioning from a source role (rows) to a target role (columns).

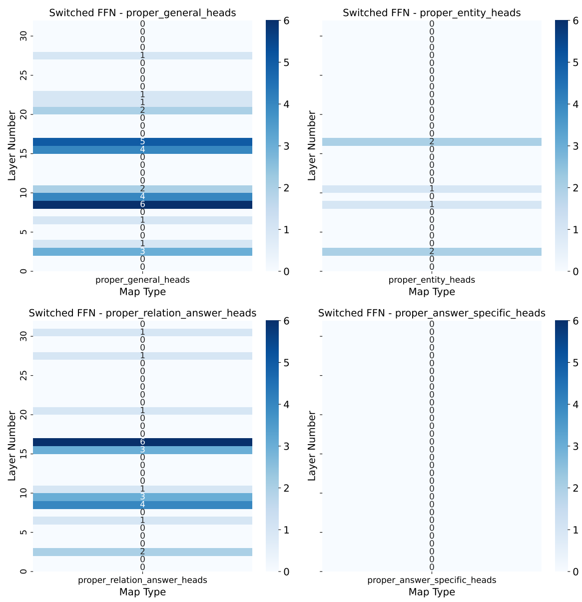

Figure 16: Layer-wise analysis of FFN role switching