Promote, Suppress, Iterate
How Language Models Answer One-to-Many Factual Queries

Tianyi Lorena Yan

Robin Jia

USC University of Southern California

The Complex Side of Answering 1-to-N Factual Queries

`List three cities from` **`France`**`: 1.` **`Paris`** `2.` **`Marseille`** `3.` **`Lyon`**

📋 Two subtasks:

- **Knowledge recall**: Given the query, retrieve relevant answers
- **Repetition avoidance**: Should not generate answers that have been generated

The Complex Side of Answering 1-to-N Factual Queries

`List three cities from` **`France`**`: 1.` **`Paris`** `2.` **`Marseille`** `3.` **`Lyon`**

📋 Two subtasks:

- **Knowledge recall**: Given the query, retrieve relevant answers
- **Repetition avoidance**: Should not generate answers that have been generated

🧐 RQs:

❑ **Overall Mechanism**: How do LMs get different answers at different steps?
❑ **Source**: How are knowledge recall and repetition avoidance implemented?

List three cities from France: 1. Paris 2. Marseille 3. Lyon

Experiment Settings

- Datasets:

Dataset	Example Prompt Template
Country-Cities	List three cities from <country>
Artist-Songs	Name three songs sung by <artist>
Actor-Movies	State three movie titles starring <actor>

Experiment Settings

- Datasets:

Dataset	Example Prompt Template
Country-Cities	List three cities from <country>
Artist-Songs	Name three songs sung by <artist>
Actor-Movies	State three movie titles starring <actor>

- Models: `Llama-3-8B-Instruct` & `Mistral-7B-Instruct-v0.2`

Experiment Settings

- Datasets:

Dataset	Example Prompt Template
Country-Cities	List three cities from <country>
Artist-Songs	Name three songs sung by <artist>
Actor-Movies	State three movie titles starring <actor>

- Models: `Llama-3-8B-Instruct` & `Mistral-7B-Instruct-v0.2`

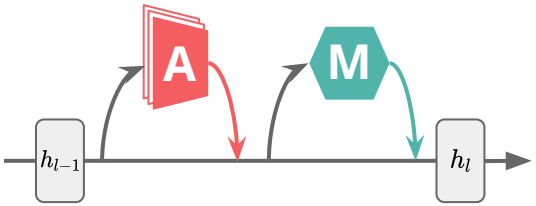
- Three templates for each model and dataset. Focus on correct cases for analyses
  When analyzing models' behaviors at answer step ***i***, use all tokens before ***i***-th answer as input.

(In the slides, all results are macro-averaged across all datasets, models, and templates. Refer to our paper for more details.)

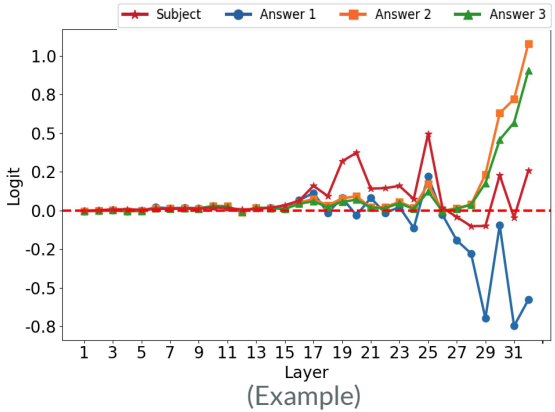
List three cities from France: 1. Paris 2. Marseille 3. Lyon

Overall Mechanism: Decode attention & MLP Outputs

- Unembed update from **Attention** and **MLP** at the last token position across layers



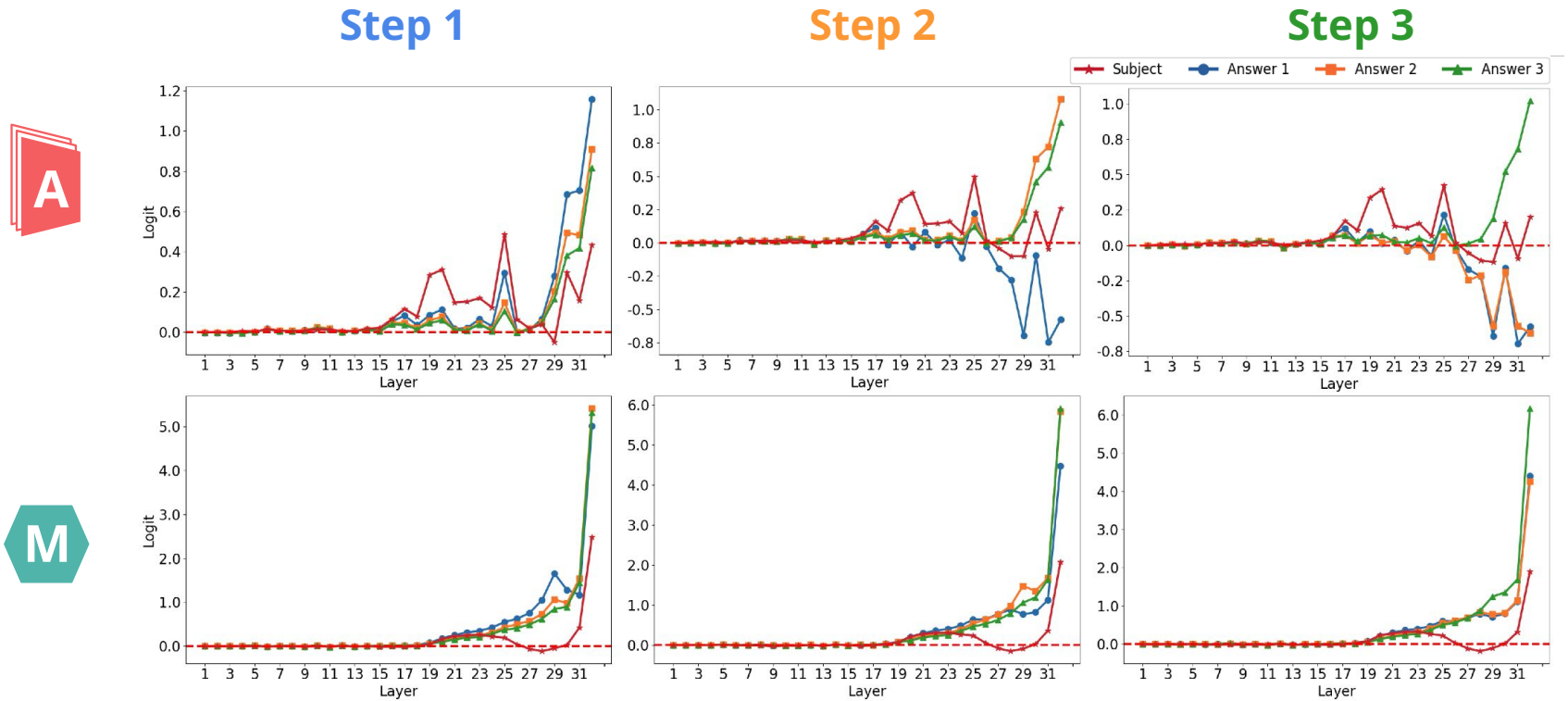
- Visualize logit of the first token of answers predicted across answer steps and the subject
  - 📈 Positive: Promotion
  - 📉 Negative: Suppression



(Example)

List three cities from France: 1. Paris 2. Marseille 3. Lyon

Overall Mechanism: Promote Then Suppress



(1) Attention copies subject (2) MLP promotes all answers (3) Previous answers suppressed

🧐 RQs

✅ **Overall Mechanism**: How do LMs get different answers at different steps?

`Promote all answers then suppress previous generated ones`

❑ **Source**: How are knowledge recall and repetition avoidance implemented?

🧐 RQs

✅ **Overall Mechanism**: How do LMs get different answers at different steps?

`Promote all answers then suppress previous generated ones`

❑ **Source**: How are knowledge recall and repetition avoidance implemented?

- Causal tracing: **Subject** and **previous answers** are crucial to LMs' outputs.

`List three cities from France: 1. Paris 2. Marseille 3. Lyon`

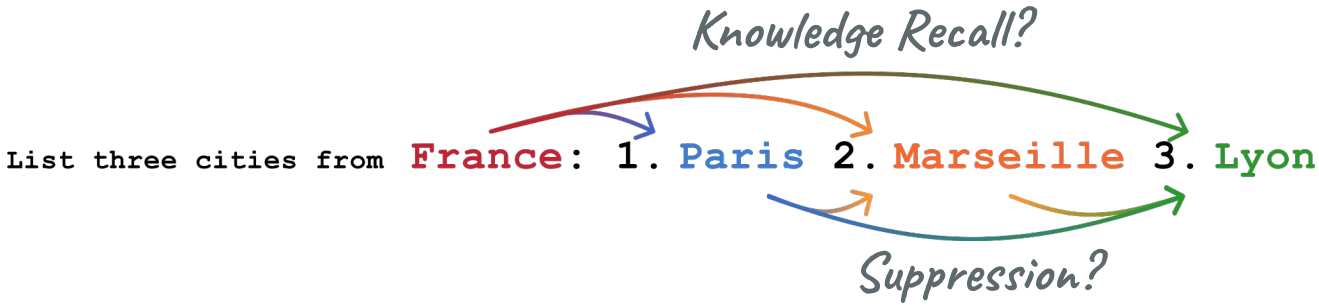
🧐 RQs

✅ **Overall Mechanism**: How do LMs get different answers at different steps?

`Promote all answers then suppress previous generated ones`

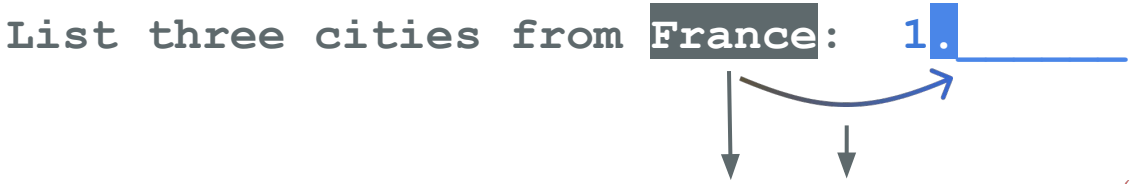
❑ **Source**: How are knowledge recall and repetition avoidance implemented?

- Causal tracing: **Subject** and **previous answers** are crucial to LMs' outputs.
- How do attention and MLPs use these tokens to facilitate:



Attention: Token Lens

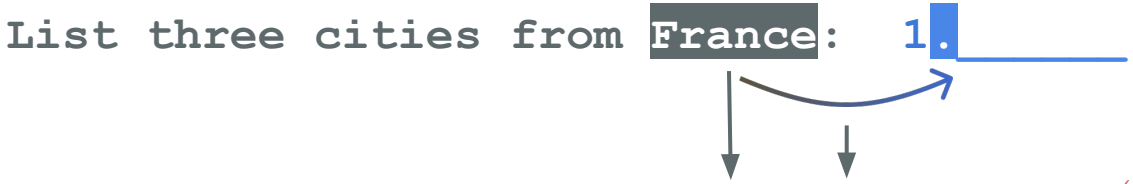
Unembed the attention output of the target tokens



- Layer l attention head i ($a^{(li)}$) = v · p → $a^{(li)}$'s contribution from the country to the last token

Attention: Token Lens

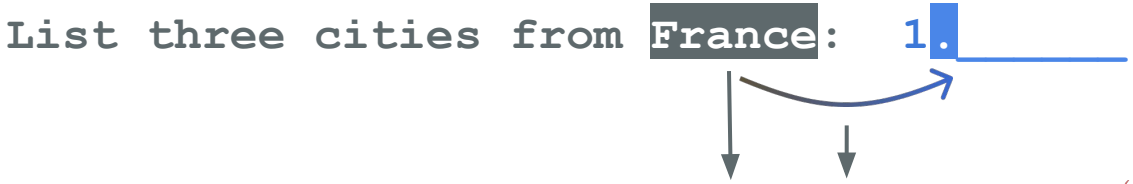
Unembed the attention output of the target tokens



- Layer l attention head i ($a^{(li)}$) = v · p → $a^{(li)}$'s contribution from the country to the last token
- Layer l attention output $a^{l} = W_o^{(l)} \cdot \text{Concat}(a^{(l_1)}, \ldots, a^{(l_n)})$

Attention: Token Lens

Unembed the attention output of the target tokens

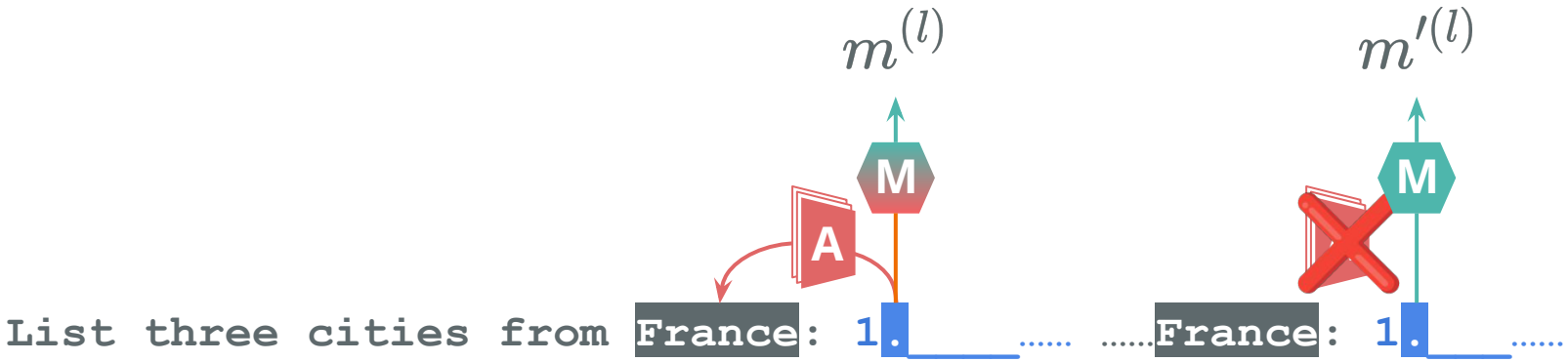


- Layer l attention head i ($a^{(li)}$) = v · p → $a^{(li)}$'s contribution from the country to the last token
- Layer l attention output $a^l = W_o^{(l)} \cdot \text{Concat}(a^{(l_1)}, \ldots, a^{(l_n)})$
- Unembed a^l and examine logits: 📈Positive for promotion. 📉Negative for suppression.

MLPs: Attention Knockout

MLPs: Attention Knockout

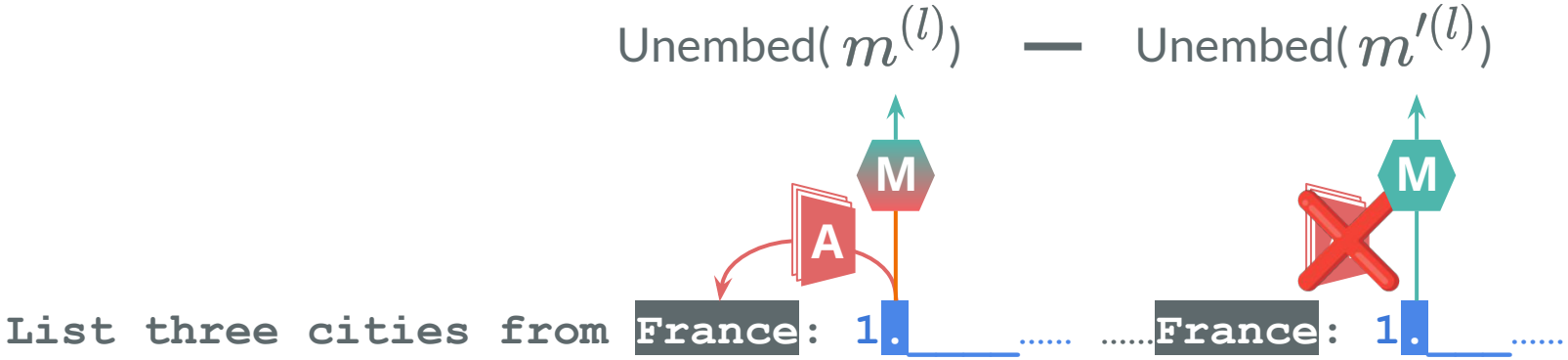
To measure how MLPs utilize target token information for answer promotion/suppression:

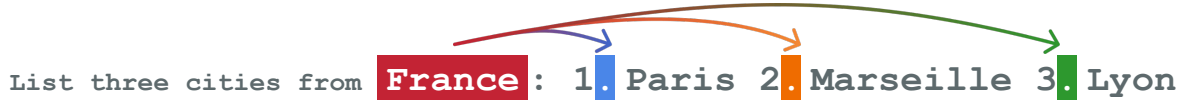


MLPs: Attention Knockout

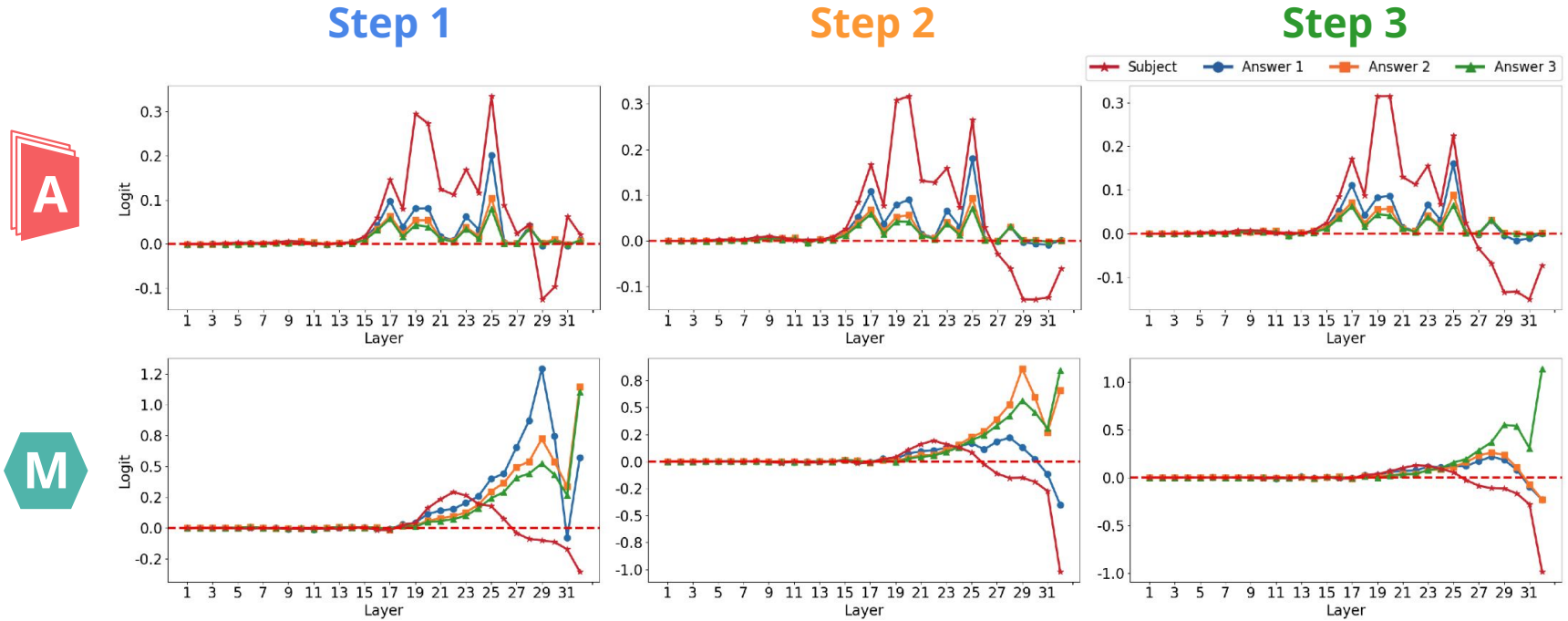
To measure how MLPs utilize target token information for answer promotion/suppression:

- Zero out Attention from the last token to target tokens
- Visualize difference in MLP output logits w/ vs. w/o attention knockout across layers



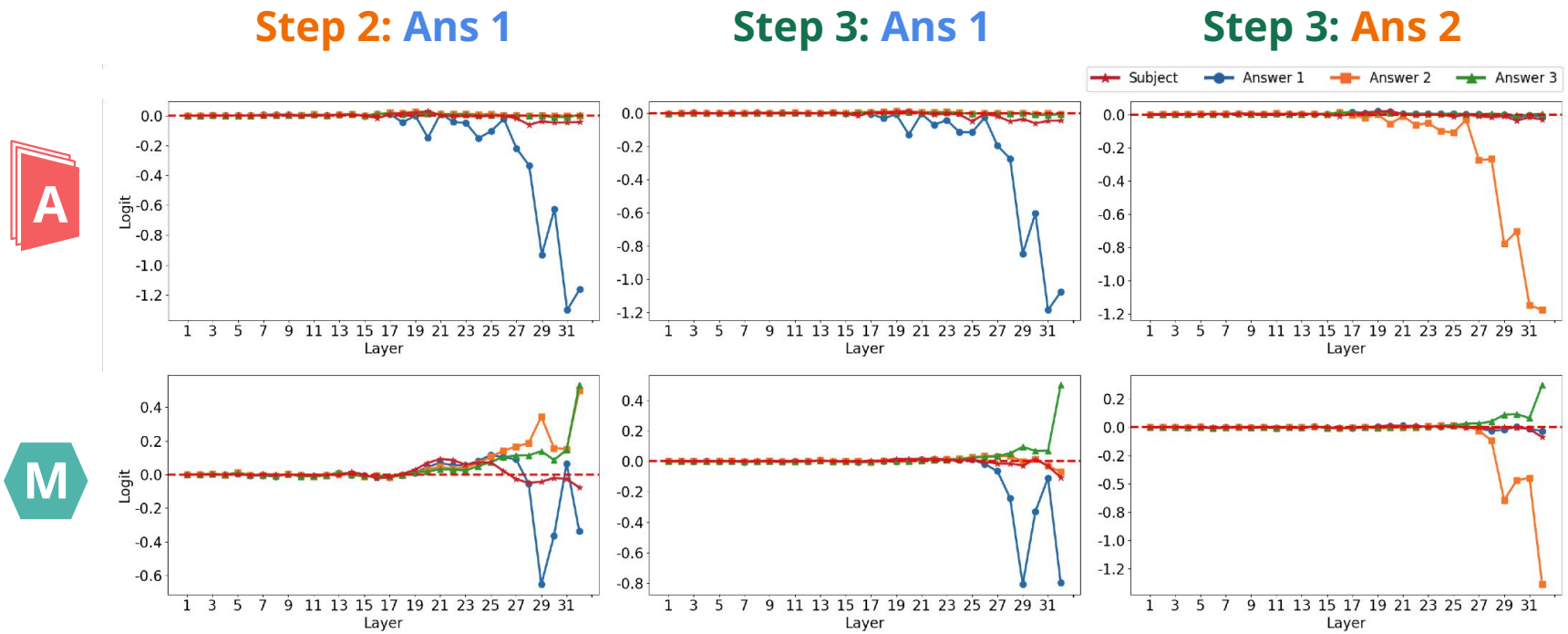


Subject for Knowledge Recall



- Attention propagates subject. MLPs promote answers.
- Attention suppresses subject and MLPs amplify subject suppression at late layers.

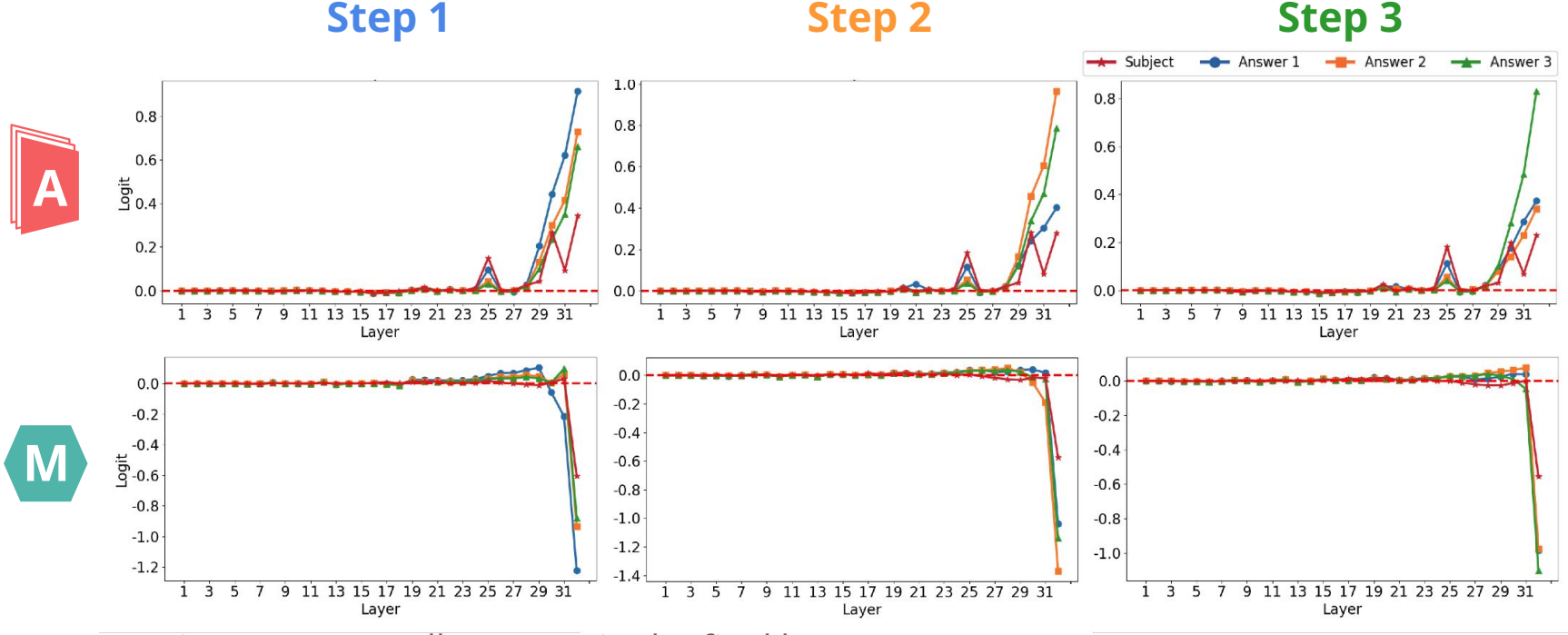
Previous Answer for Knowledge Recall + Suppression



- Attention suppresses repetition. MLP amplifies suppression.
- MLPs use previous answers to promote new answers.

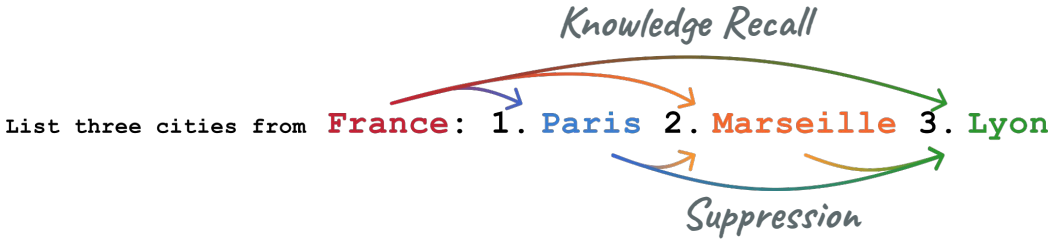
List three cities from France : 1. Paris 2. Marseille 3. Lyon

Last Token: Aggregate Knowledge Recall & Suppression



- Attention promotes all answers in the final layers.
- MLPs compensate answer promotions when the direct attention to the last token is absent.

Summary



✅ **Overall Mechanism**: How do LMs get different answers at different steps?

`Promote all answers then suppress previously generated ones`

✅ **Source**: How are knowledge recall and repetition avoidance implemented?

- Knowledge Recall:

  `Attention propagates subject information.`

  `MLPs use both the subject and previous answers to promote new answers.`

- Repetition Avoidance:

  `Attention attends to and suppresses previous answers.`

  `MLPs amplifies suppression.`

Cited Work

Kevin Meng, David Bau, Alex Andonian, Yonatan Belinkov. Locating and Editing Factual Associations in GPT. NeurIPS 2022.

QR Codes/Contacts

Paper

Code

tianyi.yan@columbia.edu / @LorenaYannnnn