# Evaluating Humanlike Memory Effects in Transformers Using Item Recognition Tasks

**Christian Clark and William Schuler**
Department of Linguistics
The Ohio State University
{clark.3664, schuler.77}@osu.edu

## Abstract

Recent studies examining cued recall in Transformers have observed that these language models remember information from the beginning or end of a passage more easily than information in the middle, a pattern which is evocative of serial position effects (primacy and recency) observed in human memory. However, while these effects have been documented in humans across a range of memory tasks (e.g., serial recall, free recall, item recognition), it is less clear whether they generalize beyond cued recall in Transformers.

We address this limitation of previous work by performing novel behavioral evaluations on Transformers using a simple item recognition paradigm, which we compare against evaluations using cued recall. We find that Transformers show weak or absent recency effects in item recognition, a pattern which differs from human behavior and from Transformers' own behavior in cued recall. A subsequent experiment examines the role of Transformers' architectural biases in producing serial position effects in item recognition and cued recall.

## 1 Introduction

An essential capacity of language models (LMs) based on the Transformer architecture (Vaswani et al., 2017) is memory retrieval: referencing words from the previous context to determine future words. Memory retrieval, accomplished by Transfomers' attention mechanism, enables these models to resolve linguistic dependencies (Michel et al., 2019; Clark et al., 2019), which is necessary for generating grammatical text; memory retrieval is also critical for practical tasks like summarization and question answering for which Transformers are widely used (Dong et al., 2023).

Given the central role of memory retrieval in Transformers' performance, it is natural to ask whether these models' memory is influenced by the same biases that affect human memory. A particular case that has received recent attention involves *serial position effects*. Given a list of items to memorize (e.g., words), humans tend to remember initial items and final items better than items in the middle; these two patterns are respectively referred to as *primacy* and *recency* effects. Recency effects in humans are often attributed to factors like decay or interference (Peterson and Peterson, 1959; Bjork and Whitten, 1974), while primacy is often attributed to factors like rehearsal of initial memory items (Rundus, 1971).

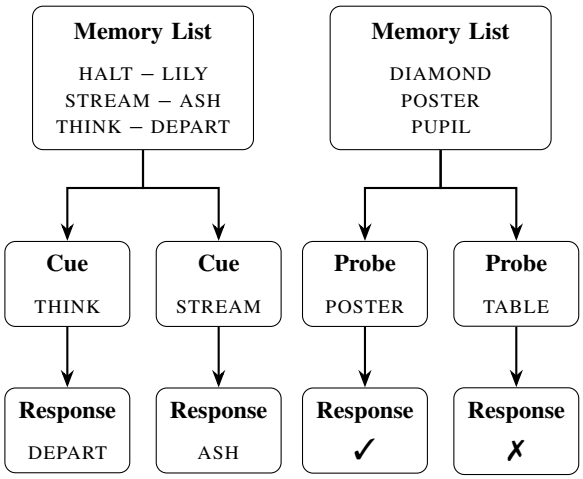
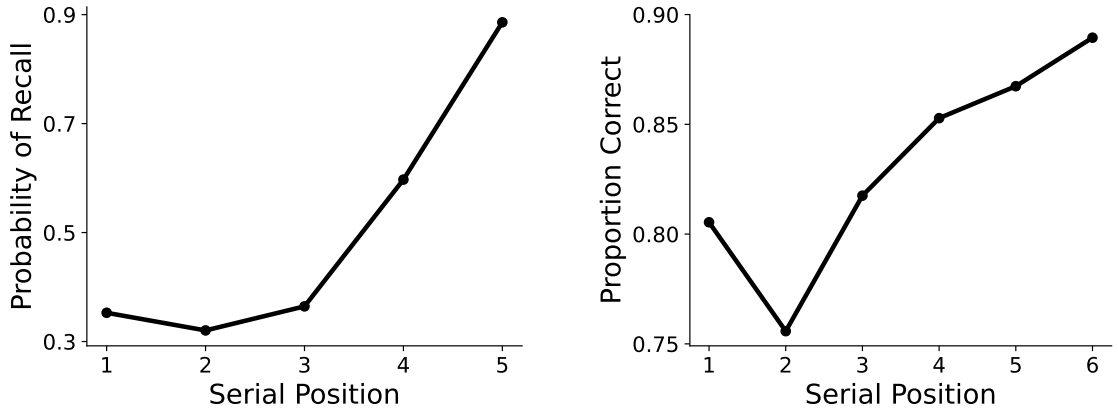



Figure 1: Illustration of cued recall (left) and item recognition (right) tasks used to study human memory. The cued recall example is borrowed from Murdock (1963b).

Recent work has reported evidence of primacy and recency effects in Transformers as well. This is an interesting result because there is little intuitive reason to expect that factors like decay, interference, or rehearsal would play into Transformer memory in the same way they play into human memory. While it seems possible that Transformers develop primacy and recency effects as a result of learning from human-authored text that reflects these biases (Janik, 2023), it has also been argued that primacy and recency effects directly re-




(a) Cued Recall

(b) Item Recognition

Figure 2: Examples of serial position effects documented in previous studies of human memory. Figure (a), adapted from Murdock (1963a, Fig. 2), is from a paired-associate cued recall task. Figure (b), adapted from Oberauer (2003, Fig. 3), is from an item recognition task.

sult from elements of the Transformer architecture like causal masking and position encoding (Wu et al., 2025; Wang et al., 2025).

In the effort to relate serial position effects in Transformers and humans, one overlooked detail involves the choice of evaluation task. Serial position effects in Transformers have often been reported from tasks involving **cued recall**, in which a model is exposed to a set of cue–target pairs, a particular cue is presented a second time, and the model determines the corresponding target. For example, Liu et al. (2024) test for serial position effects using a JSON key-value retrieval task and a multi-document question answering task; both of these follow the same fundamental form of locating a piece of information based on a retrieval cue.

Cued recall tasks are also sometimes used to study human memory, as illustrated in Figure 1 (left). Murdock (1963a) reports the serial position curves from one such evaluation, results from which are reproduced in Figure 2a. While a strong recency effect is visible, these results do not show much evidence of a primacy effect in humans, unlike results from other common memory tasks (e.g., serial recall, free recall, or item recognition). In general, relatively few studies have used cued recall paradigms to evaluate serial position effects in humans; a survey in Oberauer et al. (2018) of serial position effects across memory paradigms does not mention cued recall. The scarcity of experimental data makes it difficult to compare primacy and recency effects in LMs and humans based on cued recall tasks alone.

The present work therefore introduces a novel Transformer evaluation task which is inspired by **item recognition** studies with humans. As shown in Figure 1 (right), a typical item recognition task for humans requires the subject to memorize a list of single words. A probe word is then presented, and the subject responds "yes" or "no" depending on whether the word was previously in the list.

Primacy and recency effects have been observed in both humans' reaction times and their accuracy rates on item recognition tasks (Monsell, 1978; McElree and Dosher, 1989; Oberauer, 2003). For example, Figure 2b reproduces results from an accuracy-based evaluation in Oberauer (2003), showing a first-item primacy effect followed by a more extended recency effect.

We compare Transformers' serial position effects in this new task to their effects in a standard cued recall evaluation. In behavioral experiments using Pythia (Biderman et al., 2023) LMs, we replicate the widely observed U-shaped curve for cued recall, but observe a different pattern for item recognition, with strong primacy effects and weak recency effects.

To understand whether Transformers' architectural biases influence cued recall and item recognition in similar ways, we perform a further experiment in which Transformers are trained from scratch on controlled datasets. We find evidence that architectural bias plays a similar role in the two tasks, although its effects on item recognition are somewhat weaker. Evaluations on ablated models show that positional encodings and causal masking

may have a more nuanced connection to serial position effects than has been described in other work (Wu et al., 2025; Wang et al., 2025).

The code used for the experiments in this paper is available at https://github.com/christian-clark/item-recognition.

## 2 Related Work

Primacy and recency effects in Transformers have been widely documented in natural language processing (NLP) tasks that follow a "needle-in-a-haystack" format, such as key-value retrieval and multi-document question answering (e.g., Kamradt, 2023; Liu et al., 2024; Wang et al., 2025). Most of these studies use English LMs, but similar effects have been observed crosslinguistically (Menschikov et al., 2025). Serial position effects have also been reported in other NLP tasks with less resemblance to cued recall, such as classification and summarization (e.g., Wang et al., 2023; Guo and Vosoughi, 2025), although the effects often deviate from the U-shaped serial position curves seen in cued recall tasks. Our work uses simplified tasks rather than these NLP evaluations in order to more directly interface with research on human memory.

Recent work has looked at potential causes behind serial position effects (e.g., causal masking and positional encoding) through formal analyses (Wu et al., 2025; Wang et al., 2025) and/or experiments on trained-from-scratch models (Wu et al., 2025; Salvatore et al., 2025). Our Experiment 3, which also evaluates trained-from-scratch models, expands this work to the study of item recognition tasks.

Two other relevant studies are Armeni et al. (2022) and Janik (2023). Armeni et al. (2022) tests for verbatim memory in Transformers using a prompting technique resembling serial recall evaluations of humans. However, because the prompts allow the model to condition on the correct initial list items when processing later items (i.e., teacher forcing), the behavior they elicit does not straightforwardly map to human recall. Janik (2023) shows strong primacy and recency effects using a simple cued recall paradigm that we adopt for our Experiment 1.

## 3 Experiment 1: Cued Recall

Our first experiment uses a cued recall paradigm as a replication of previous memory studies showing U-shaped serial position curves in Transformers.

> *Please try to remember these sentences. Britt lives in Santiago. Ethan lives in Portland. ...Harry lives in Nashville. Now, after you received all this information, try to concentrate, drink a cup of coffee, go for a walk. Then please complete the following sentence, based on what you read earlier. Ethan lives in* Portland

Figure 3: Example prompt for the cued recall task (Experiment 1).

### 3.1 Methods

We adopt the prompting technique of Janik (2023), an example of which is shown in Figure 3. Each prompt begins with brief instructions, followed by a list of sentences containing pairs of names and cities, e.g., *Ethan lives in Portland*. After an intervening passage, one of the name–city sentences is repeated, and the surprisal of its city word is recorded. This task involves cued recall because the name (e.g., *Ethan*) is used as a retrieval cue to identify the corresponding city (*Portland*). We restrict our evaluation to name–city pairs for this experiment, but Janik (2023) shows that similar results obtain with other kinds of cue–target pairs.

The names in the main sentences are a subset of the Natural Language Toolkit names corpus (Bird et al., 2009), which consists of English first names. The full set of first names was filtered down to names that are a single token in vocabulary of the Pythia and GPT-2 models; the size of this filtered subset was 1248 names. The cities in the main sentences were selected from the top 1000 world cities by population according to a world cities database.[1] These were similarly filtered down to single-token cities, resulting in a final set of 100 cities.

1000 sets of name–city pairs were randomly generated, half of which contained 20 pairs and the other half of which contained 50 pairs. Names and cities never overlapped across pairs. These were used to generate prompts following the form of Figure 3, in which name–city pairs are embedded in sentences and a single pair is repeated at the end.

Our evaluation measure, which we term *surprisal reduction*, measures the degree to which the surprisal (negative log probability) of the target city (*Portland* in Figure 3) is reduced at the end of the prompt, relative to its surprisal when it first occurs in the prompt. The surprisal reduction of target

[1] https://simplemaps.com/data/world-cities

word $w_i$ is defined as follows:

$$\text{SurpReduc}(w_i) = \frac{\mathsf{S}_{\text{initial}}(w_i) - \mathsf{S}_{\text{final}}(w_i)}{\mathsf{S}_{\text{initial}}(w_i)}, \quad (1)$$

where $\mathsf{S}_{\text{initial}}(w_i)$ is the target word's surprisal when it first appears in the prompt and $\mathsf{S}_{\text{final}}(w_i)$ is the same word's surprisal at the end of the prompt. A surprisal reduction near 1 means that the LM strongly expects the target city $w_i$ after being cued with its corresponding name; on the other hand, a surprisal reduction near 0 means that the LM shows no sign of remembering the target city. Surprisal reduction is closely related to the repeat surprisal metric used by Armeni et al. (2022).[2]

The LMs used in this experiment were from the Pythia (Biderman et al., 2023) family of models, which includes models with 14M, 31M, 70M, 160M, 410M, 1B, 1.6B, 6.7B, and 12B parameters.[3]

### 3.2 Results

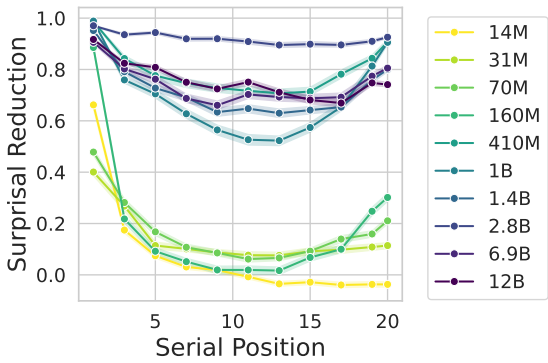
Figure 4 shows the average surprisal reduction by serial position across the Pythia models. These curves are plotted separately for list lengths (i.e., number of name–city sentences) of 20 or 50. Across both list lengths, we generally replicate the U-shaped curve noted in Janik (2023) and other previous studies on Transformers' serial position effects in cued recall. The main exceptions are the smallest Pythia models (14M, 31M, and 70M), which show strong primacy effects but weakened or absent recency effects; this is consistent with trends observed by Janik (2023). See Appendix A for a replication with GPT-2 LMs (Radford et al., 2019) which shows similar trends.

## 4 Experiment 2: Item Recognition

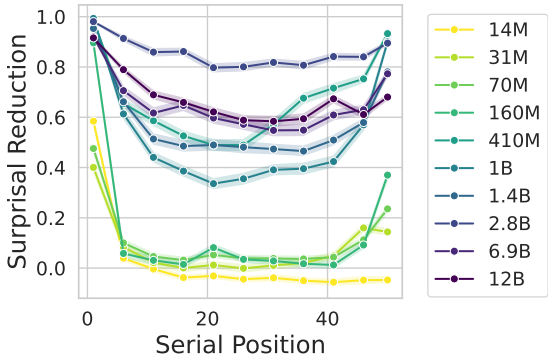
Our second experiment tests whether Transformer LMs display similar primacy and recency effects in a behavioral evaluation inspired by item recognition studies in humans (Fig. 1, right). Unlike cued recall, item recognition is a task for which primacy and recency effects in humans are well attested (Madigan, 1971; McElree and Dosher, 1989; Oberauer, 2003).

[2] $\text{SurpReduc}(w_i) = 1 - \text{RepeatSurprisal}(w_i)$

[3] We use the Pythia variants trained on deduplicated data, except for the 14M and 31M sizes, for which "deduped" variants are not currently available.




(a) List length 20




(b) List length 50

Figure 4: Pythia cued recall results.

### 4.1 Methods

Figure 5a shows an example prompt used to test item recognition. The prompt consists of two sentences, the first of which introduces a list of names, and the second of which repeats one of the names in a context in which any of the listed names would be appropriate. We assume that lower surprisal associated with the repeated name reflects stronger item recognition in LMs, just as higher response accuracy and faster reaction times reflect stronger item recognition in humans.

The same set of single-token names was used as in Experiment 1. Names in the prompt's first sentence were separated by commas and spaces; to avoid distinguishing the last name in the list (*Lisa* in Fig. 5a), we omitted the conjunction *and* that would typically precede it. Similarly to Experiment 1, a set of 1000 name lists was generated to fill in the prompt, 500 of which contained 20 names and the other 500 of which contained 50 names.

To measure recognition of the final word, we compare its surprisal at the end of the main prompt (e.g., Fig. 5a) to the surprisal of the same word following a baseline prompt (Fig. 5b), which pre-

*This morning, Elena, Clint, . . . , Adrian, Lisa were the people I encountered. In the afternoon, I again saw* *Adrian*

(a) Main prompt

*In the afternoon, I again saw* *Adrian*

(b) Baseline prompt

Figure 5: Example prompt for item recognition (a) and prompt used to calculate baseline surprisal (b) in Experiment 2.

serves some of the context but without the initial name list.

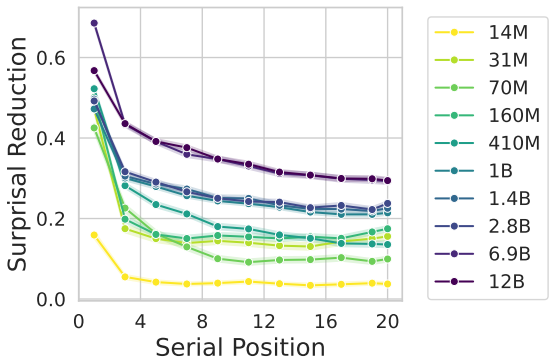
Our evaluation metric for this task also accounts for the fact that an LM must divide its probability mass over the possible names at the end of the main prompt. For instance, an LM with strong item recognition would not only assign high probability (low surprisal) to *Adrian* at the end of the prompt in Figure 5a, but would also assign high probability to other names in the memory list (*Elena*, *Clint*, *Lisa*, etc.). For a list of $N$ names, we assume an LM with optimal item recognition (and no positional bias) would assign each name a uniform probability of $1/N$, and thus a surprisal of $\mathsf{S}_{\text{unif}}(w_i) = \ln N$. Our revised definition of surprisal reduction for item recognition includes this term in the denominator:

$$\text{SurpReduc}(w_i) = \frac{\mathsf{S}_{\text{baseline}}(w_i) - \mathsf{S}_{\text{main}}(w_i)}{\mathsf{S}_{\text{baseline}}(w_i) - \mathsf{S}_{\text{unif}}(w_i)}. \quad (2)$$

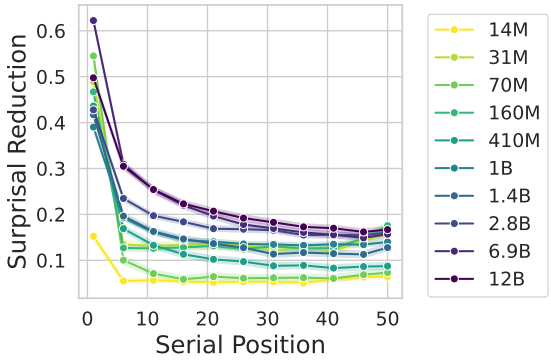
Here, $\mathsf{S}_{\text{main}}(w_i)$ and $\mathsf{S}_{\text{baseline}}(w_i)$ are the surprisal of $w_i$ at the end of the main prompt and the baseline prompt respectively. Including $\mathsf{S}_{\text{unif}}$ in the denominator means that a surprisal reduction near 1 occurs when $\mathsf{S}_{\text{main}}$ is near $\mathsf{S}_{\text{unif}}(w_i)$, the "best-case" surprisal.

### 4.2 Results

Figure 6 shows the serial position curves for item recognition across the Pythia models. Unsurprisingly, larger models tend to show a stronger surprisal reduction. Primacy effects are consistently present; these are particularly strong at the first list items but carry over into later positions in most models. Evidence of recency effects, on the other hand, is scant. A slight increase in surprisal reduction occurs for final words in certain models (particularly Pythia 160M), but it is much weaker than the primacy effects.




(a) List length 20




(b) List length 50

Figure 6: Pythia item recognition results.

These serial position curves differ from the cued recall results in Experiment 1 (Figure 4). The curves in Experiment 1 were mostly symmetrical, while the curves from item recognition show dominant primacy effects. The difference across experiments suggests that serial position effects in Transformers are sensitive to the choice of evaluation task. Figure 6 also contrasts markedly with results from item recognition evaluations in humans, which show both primacy and recency effects, the latter of which are usually stronger (Monsell, 1978; McElree and Dosher, 1989; Oberauer, 2003). Appendix A presents results from GPT-2 models on the same evaluations, which again show similar trends to Pythia models.

To verify that the behavior observed in this experiment was not driven by the specific choice of prompt (Figure 5) and list items, evaluations were performed using differently worded prompts, and using lists of cities instead of names. The results from these evaluations showed similar trends to Figure 6; see Appendix B for further details.

## 5 Experiment 3: The Role of Architectural Bias

The previous experiments use off-the-shelf Pythia and GPT-2 LMs whose serial position effects may reflect a mixture of innate architectural biases in Transformers and patterns learned from their training corpora. Our third experiment uses controlled evaluations to assess the degree to which serial position effects in cued recall and item recognition arise from the Transformer architecture itself. Specifically, it examines the learning trajectories of Transformers that are trained to predict patterns resembling simplified forms of the cued recall and item recognition prompts used in Experiments 1 and 2. We test whether Transformers take longer to learn patterns involving repetition of a list-medial item than patterns involving repetition of list-initial or list-final items. We also test ablated models without positional encodings or causal masking to assess how these elements of the architecture influence observed serial position effects in item recognition and cued recall.

### 5.1 Methods

A set of Transformer LMs was trained from scratch using artificial corpora. The sentences in each artificial corpus followed one of two patterns. The first pattern, meant to resemble the cued recall paradigm in Experiment 1, took the form

$$x_1\ y_1\ x_2\ y_2\ \ldots\ x_n\ y_n\ x_i\ y_i. \tag{3}$$

The second pattern, meant to resemble the item recognition paradigm in Experiment 2, took the form

$$x_1\ x_2\ \ldots\ x_n\ \mathsf{SEP}\ x_i. \tag{4}$$

Here, $n$ is the list length and $i \in \{1, \ldots, n\}$ is the index of the repeated pair or item. The SEP token in (4) separates the memory list from the repeated item.[4]

The $x$ and $y$ items in the memory lists were chosen by randomly permuting two disjoint sets of tokens $X$ and $Y$, with $|X| = |Y| = n$. In keeping with Experiments 1 and 2, we tested $n \in \{20, 50\}$.

For corpora with $n = 20$, we tested $i \in \{1, 2, 10, 11, 19, 20\}$, and for corpora with $n = 50$, we tested $i \in \{1, 2, 25, 26, 49, 50\}$. These indices were selected to examine how easily the model learns to repeat memory items at the beginning, middle, and end of the list, with two indices at each location to check whether observed behavior was reasonably consistent between neighboring indices.[5] For each $i$ value, 10 training corpora containing 1 million sentences were randomly generated and used to train 10 randomly initialized Transformers.

The Transformers trained in this experiment were two-layer models based on the Pythia architecture. These models included 2 layers with 4 attention heads and a hidden dimension of 64, for a total parameter count of 2.2M. This small size was chosen to allow for training a large set of models on the available budget. Models' vocabulary was restricted to $X \cup Y$ (for models trained on the cued recall–style corpora) or $X \cup \{\mathsf{SEP}\}$ (for models trained on the item recognition–style corpora). Like standard Pythia models, the trained-from-scratch Transformers used rotary embeddings (Su et al., 2024) by default.

Because the tokens in the memory list are selected randomly, their losses do not provide a meaningful training signal. We therefore modified the training objective to only consider the loss of the final token, which is equivalent to its surprisal. The surprisal reduction of token $w_i$ at training step $t$ was calculated by comparing its loss at $t$ to its loss in an untrained model (i.e., $t = 0$):

$$\mathrm{SurpReduc}(w_i, t) = \frac{\mathsf{S}_{\mathrm{untrained}}(w_i) - \mathsf{S}_t(w_i)}{\mathsf{S}_{\mathrm{untrained}}(w_i)}, \tag{5}$$

where $\mathsf{S}_{\mathrm{untrained}}(w_i)$ is the loss from the untrained model and $\mathsf{S}_t(w_i)$ is the loss at step $t$.[6]

### 5.2 Ablations

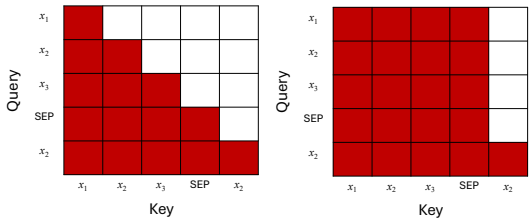
Along with testing the learning dynamics of models following the standard Pythia design, we tried ablating two elements in the Transformer architecture that have been argued to influence serial position effects (Wu et al., 2025; Wang et al., 2025):

- **Rotary embeddings (RoPE).** The rotary embeddings used in Pythia LMs perform a rotation to query and key embeddings proportional to their position in the input sequence,

[4] SEP was added to (4) after seeing that models trained on the item recognition pattern without an intervening SEP converged extremely quickly when $i = n$, because of the ease of predicting a second $x_n$ directly from $x_n$ without having to attend to other tokens. This issue did not affect models trained on the cued recall pattern because $x_i$ already separates $y_i$ from $y_n$.

[5] Other values for $i$ were not tested due to limitations in compute budget.

[6] Including $\mathsf{S}_{\mathrm{unif}}(w_i)$ in the denominator, as in Equation (2), was not necessary here because there is only one correct candidate for the final token.




(a) Causal Mask (b) Prefix Mask

Figure 7: Illustration of the standard Transformer causal mask and a prefix mask. The prefix mask allows all items up to SEP to attend to each other, potentially reducing the primacy bias arising from causal masking.

which causes positions closer together to have more similar embeddings. This was argued to introduce decay into Transformers, supporting a recency effect. We trained ablated models that lacked these positional encodings.

- **Causal masking.** The causal mask in autoregressive Transformer models blocks a token at index $i$ from attending to tokens at index $i + 1$ and beyond. This was argued to contribute to primacy effects because it amplifies the influence of early positions. We ablate the causal mask by replacing it with a prefix mask that allows all items in the memory list to attend to each other, removing the early-token advantage (Fig. 7).

Models with ablated rotary embeddings ($-$RoPE) and ablated causal masks ($-$Causal) were trained following the methods in §5.1.

### 5.3 Results

The results from this experiment are presented in Figure 8. The top two rows show the learning dynamics from models trained on cued recall–style corpora with memory lists of length $n = 20$ (Fig. 8a) and 50 (Fig. 8b). The bottom two rows similarly show the learning dynamics from models trained on the item recognition–style corpora with memory lists of length 20 (Fig. 8c) and 50 (Fig. 8d). Within each row, the three plots show results from standard Pythia-based models (left), models with ablated positional encodings (center), and models with ablated causal masks (right).

Each plot shows the time course of the surprisal reduction as a function of the repeated index $i$. For a particular repeated index, this time course is represented as a series of vertically stacked markers with varying opacity; these show the surprisal reduction at evenly spaced time steps over the course of training. Lines between markers trace out serial position curves by showing how the surprisal reduction at a given time step varies by repeated index.

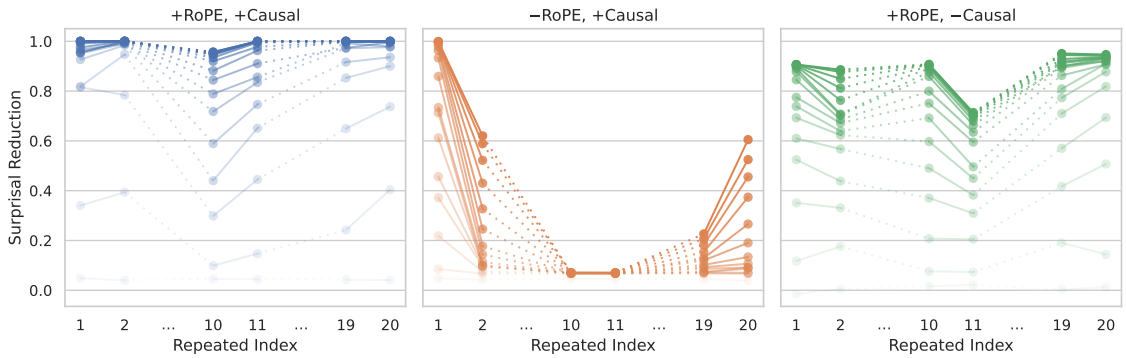
The serial position curves in Figure 8a (left) and 8b (left) show that Transformers trained on the cued recall–style corpora take longer to learn to repeat list-medial items, a result which is consistent with lost-in-the-middle trends observed from most models in Experiment 1 (Fig. 4). Primacy and recency effects are especially strong in the evaluation with $n = 50$ (Fig. 8b, left).

The serial position curves from models trained on item recognition–style corpora—Figure 8c (left) and 8d (left)—show somewhat weaker trends, with serial position effects most clearly visible when $n = 50$ (Fig. 8d, left).

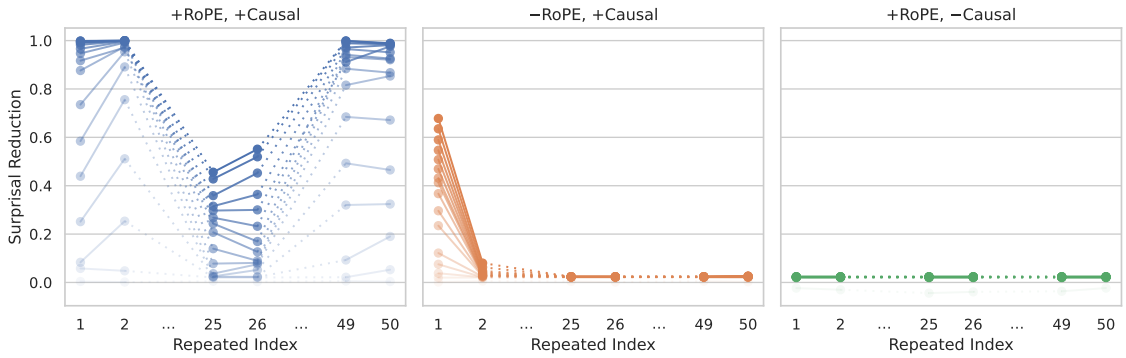
These results provide evidence that the Transformer architecture itself contributes to serial position effects in both cued recall and item recognition. But the effects are somewhat attenuated in item recognition, indicating that their strength may depend on the choice of evaluation task.

It is not entirely clear why the serial position curves from item recognition (Fig. 8c, left, and 8d, left) follow a different shape from what is seen in Experiment 2 (Fig. 6). Three possibilities are that (1) the serial position curves from Experiment 2 reflects biases in the Pythia training data that steer the models away from recency effects; (2) the shape of the curve varies by model scale, with weaker recency effects in Pythia-scale models than in the small Transformers tested in Experiment 3; or (3) the shape of the curve is sensitive to the differences between the specific item recognition tasks used in Experiments 2 and 3. In future work we plan to test these possible explanations with additional controlled experiments.

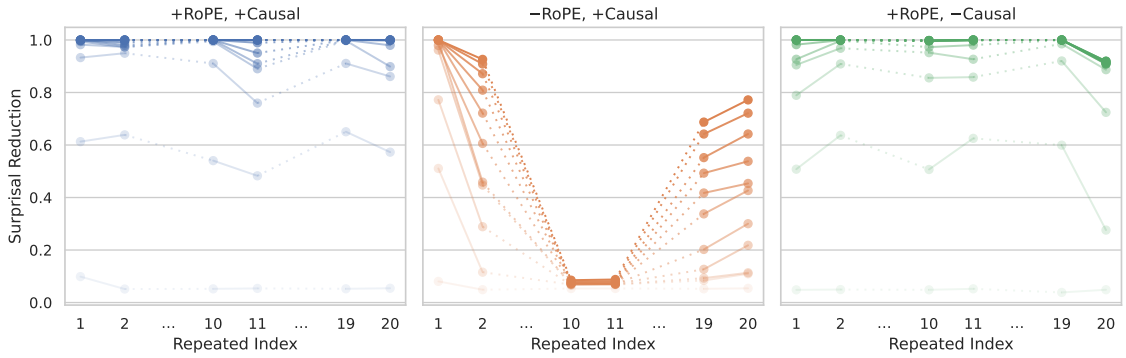
The results from ablated models, in the center and right plots of Figure 8, show interesting evidence regarding the links between positional encoding, causal masking, and serial position effects. As noted in §5.2, prior work by Wu et al. (2025) and Wang et al. (2025) attributes primacy effects to causal masking and recency effects to positional encodings. However, we see cases of primacy effects during training in models without causal masking (Fig. 8d, right), and even stronger cases of recency effects in models without positional encodings (e.g., Fig. 8b, middle). These findings may indicate that causal masking and positional encoding do not fully account for serial position effects,



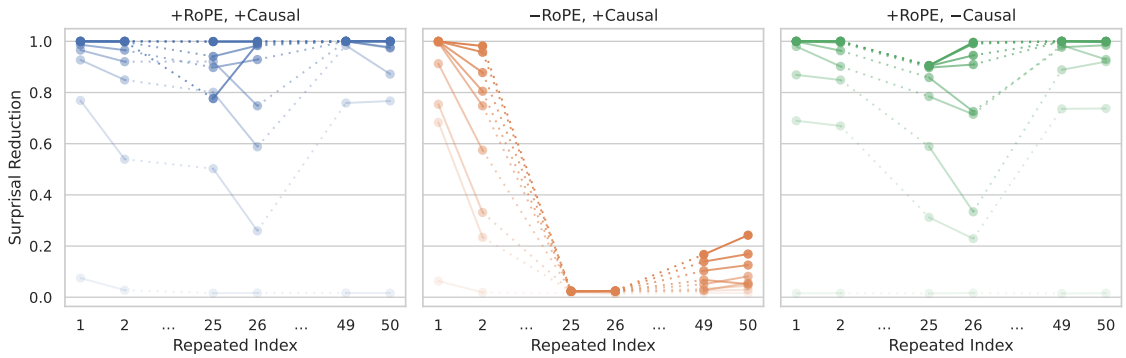

(a) Cued Recall, List Length 20




(b) Cued Recall, List Length 50




(c) Item Recognition, List Length 20




(d) Item Recognition, List Length 50

Figure 8: Progression of surprisal reduction as a function of the serial position of the repeated index. Higher-opacity markers correspond to later steps in training.

or at a minimum that their contributions to these effects are not entirely separable.

## 6 Conclusion

A range of studies have pointed out primacy and recency effects in Transformer memory when these models are used for cued recall tasks. We introduce an alternative task based on item recognition to test whether this pattern holds in a paradigm in which primacy and recency effects are more clearly attested in humans (Monsell, 1978; McElree and Dosher, 1989; Oberauer, 2003). In behavioral evaluations of Pythia models, we observe a lack of recency effects, contrasting with results from human memory studies. Evaluations on smaller Transformers trained on synthetic data show some evidence for primacy and recency effects in item recognition, but a shallower serial position curve compared to cued recall.

The contrast between results from cued recall and item recognition indicates that Transformers' serial position effects are sensitive to the choice of memory task. Although this variability in Transformers leads to a mismatch with humans in some cases, it is worth noting that serial position effects in humans are also task-sensitive. For example, the relative strength of humans' primacy and recency effects differs between forward and backward serial tasks (Madigan, 1971). A fruitful direction for future work may be to apply a wider range of memory evaluations to Transformers—and other LM architectures—to more fully understand in what cases the LMs show similar behavioral biases to humans.

## Acknowledgments

We thank the Ohio State Clippers discussion group, members of the Laboratoire de Sciences Cognitive et Psycholinguistique at ENS Paris, and the anonymous reviewers for their feedback on this work.

## References

Kristijan Armeni, Christopher Honey, and Tal Linzen. 2022. Characterizing verbatim short-term memory in neural language models. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 405–424, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 2397–2430.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Robert A Bjork and William B Whitten. 1974. Recency-sensitive retrieval processes in long-term free recall. *Cognitive Psychology*, 6(2):173–189.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of berts attention. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pages 276–286.

Zican Dong, Tianyi Tang, Junyi Li, and Wayne Xin Zhao. 2023. A survey on long text modeling with transformers. *arXiv preprint arXiv:2302.14502*.

Xiaobo Guo and Soroush Vosoughi. 2025. Serial position effects of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 927–953, Vienna, Austria. Association for Computational Linguistics.

Romuald A Janik. 2023. Aspects of human memory and large language models. *arXiv preprint arXiv:2311.03839*.

Gregory Kamradt. 2023. Needle in a haystack – pressure testing LLMs. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`. GitHub repository.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Stephen A Madigan. 1971. Modality and recall order interactions in short-term memory for serial order. *Journal of Experimental Psychology*, 87(2):294.

Brian McElree and Barbara A Dosher. 1989. Serial position and set size in short-term memory: the time course of recognition. *Journal of experimental psychology: General*, 118(4):346.

Mikhail Menschikov, Alexander Kharitonov, Maiia Kotyga, Vadim Porvatov, Anna Zhukovskaya, David Kagramanyan, Egor Shvetsov, and Evgeny Burnaev. 2025. Beyond early-token bias: Model-specific and language-specific position effects in multilingual llms. *Preprint*, arXiv:2505.16134.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Stephen Monsell. 1978. Recency, immediate recognition memory, and reaction time. *Cognitive Psychology*, 10(4):465–501.

Bennet B Murdock. 1963a. Short-term memory and paired-associate learning. *Journal of Verbal Learning and Verbal Behavior*, 2(4):320–328.

Bennet B Murdock. 1963b. Short-term retention of single paired associates. *Journal of Experimental Psychology*, 65(5):433.

Klaus Oberauer. 2003. Understanding serial position curves in short-term recognition and recall. *Journal of Memory and Language*, 49(4):469–483.

Klaus Oberauer, Stephan Lewandowsky, Edward Awh, Gordon DA Brown, Andrew Conway, Nelson Cowan, Christopher Donkin, Simon Farrell, Graham J Hitch, Mark J Hurlstone, Wei Ji Ma, Candice C Morey, Derek Evan Nee, Judith Schweppe, Evie Vergauwe, and Geoff Ward. 2018. Benchmarks for models of short-term and working memory. *Psychological bulletin*, 144(9):885.

Lloyd Peterson and Margaret Jean Peterson. 1959. Short-term retention of individual verbal items. *Journal of experimental psychology*, 58(3):193.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *ArXiv*.

Dewey Rundus. 1971. Analysis of rehearsal processes in free recall. *Journal of experimental psychology*, 89(1):63.

Nikolaus Salvatore, Hao Wang, and Qiong Zhang. 2025. Lost in the middle: An emergent property from information retrieval demands in llms. *Preprint*, arXiv:2510.10276.

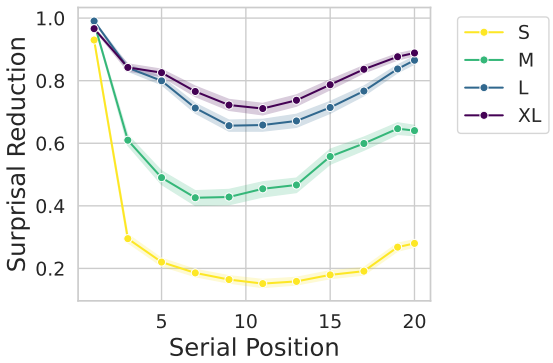
Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 6000–6010.

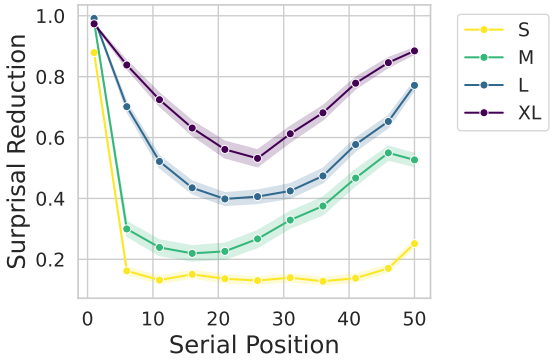
Yiwei Wang, Yujun Cai, Muhao Chen, Yuxuan Liang, and Bryan Hooi. 2023. Primacy effect of ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 108–115, Singapore. Association for Computational Linguistics.

Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M. Kakade, Hao Peng, and Heng Ji. 2025. Eliminating position bias of language models: A mechanistic approach. *Preprint*, arXiv:2407.01100.

Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. 2025. On the emergence of position bias in transformers. *Preprint*, arXiv:2502.01951.

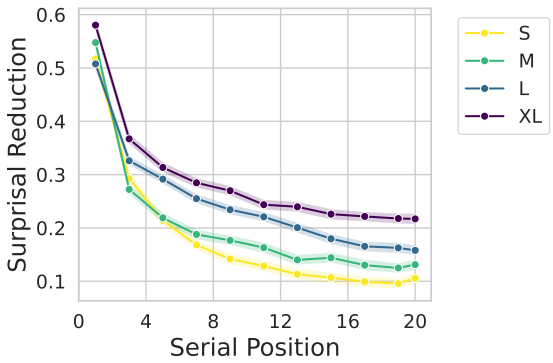



(a) List length 20

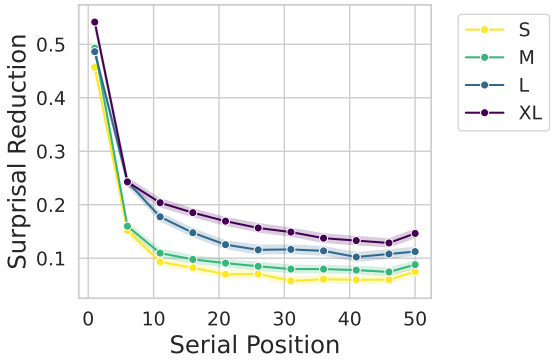



(b) List length 50

Figure 9: GPT-2 cued recall results.




(a) List length 20




(b) List length 50

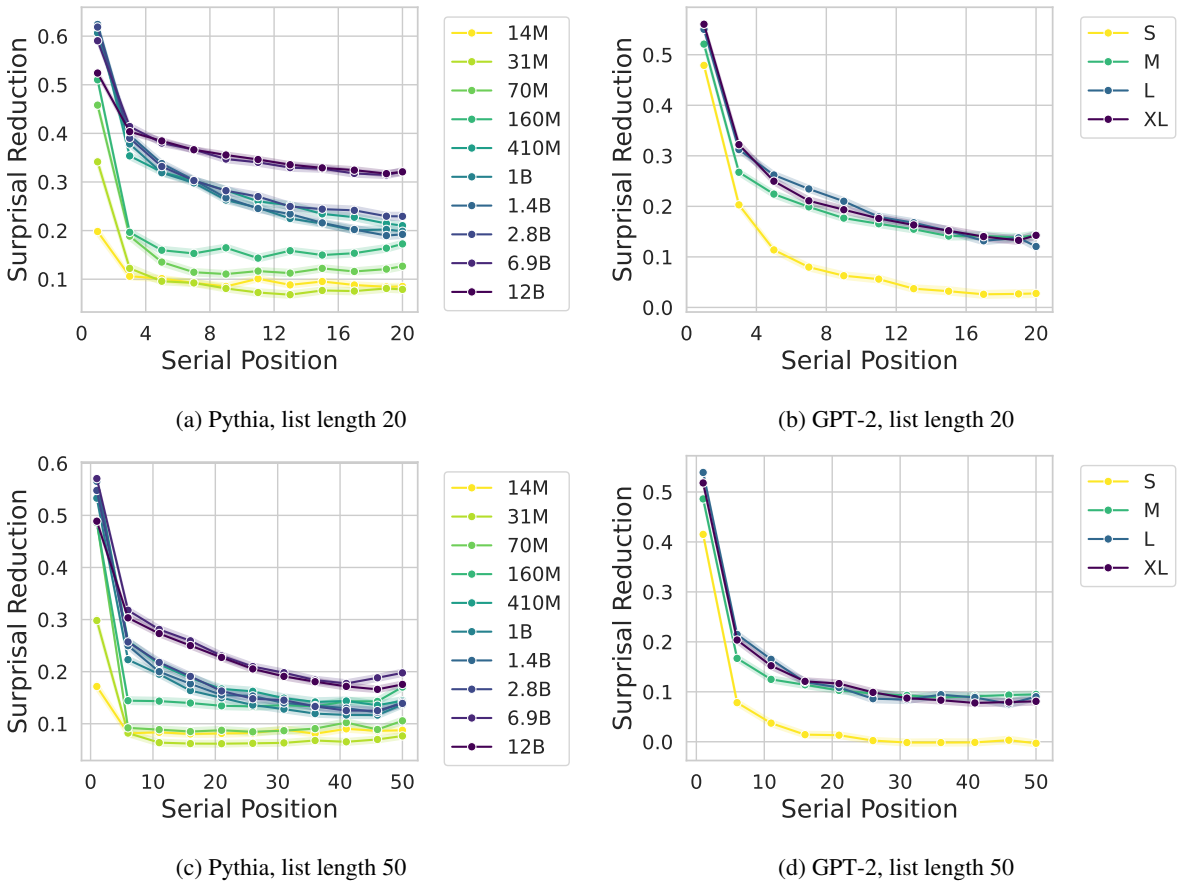
Figure 10: GPT-2 item recognition results.

## A Cued Recall and Item Recognition Results from GPT-2

Figures 9 and 10 show evaluation results on GPT-2 models (Radford et al., 2019) using the cued recall task from Experiment 1 and item recognition task from Experiment 2, respectively.

## B Additional Item Recognition Evaluations

Figures 11, 12, and 13 present results from three alternative item recognition evaluations on the Pythia and GPT-2 LMs. Figure 11 uses a prompt that clarifies that the speaker met the people all at once in a group, to avoid the potential implication of chronological ordering in the original prompt (Figure 5). Figure 12 uses an instruction-based prompt similar to the Experiment 1 prompt (Figure 3). Figure 13 uses cities rather than names as list items.



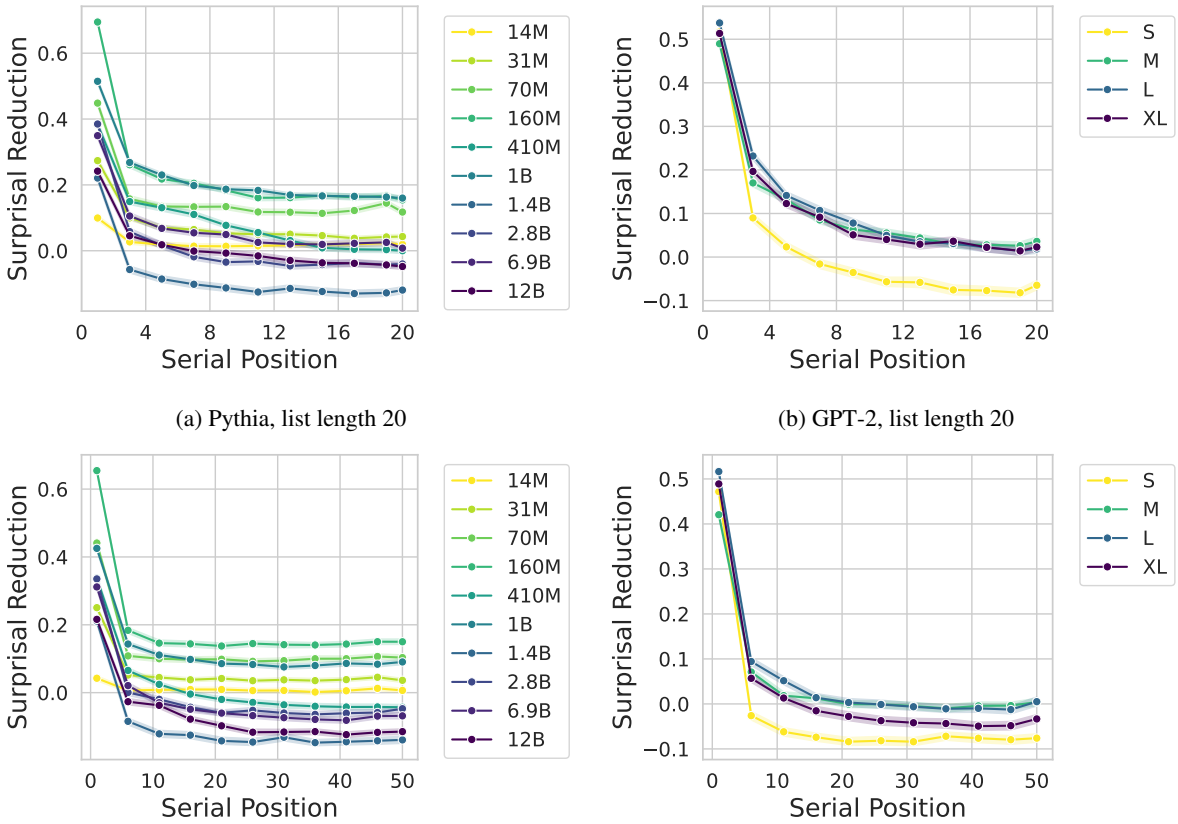

(a) Pythia, list length 20

(b) GPT-2, list length 20

(c) Pythia, list length 50

(d) GPT-2, list length 50

Figure 11: Pythia and GPT-2 item recognition results with an alternative prompt. Example of main prompt: *In the morning, I had a meeting with a group of people including Elena, Clint, . . . , Adrian, Lisa. In the afternoon, I again encountered Adrian*. Baseline prompt: *In the afternoon, I again encountered Adrian*.



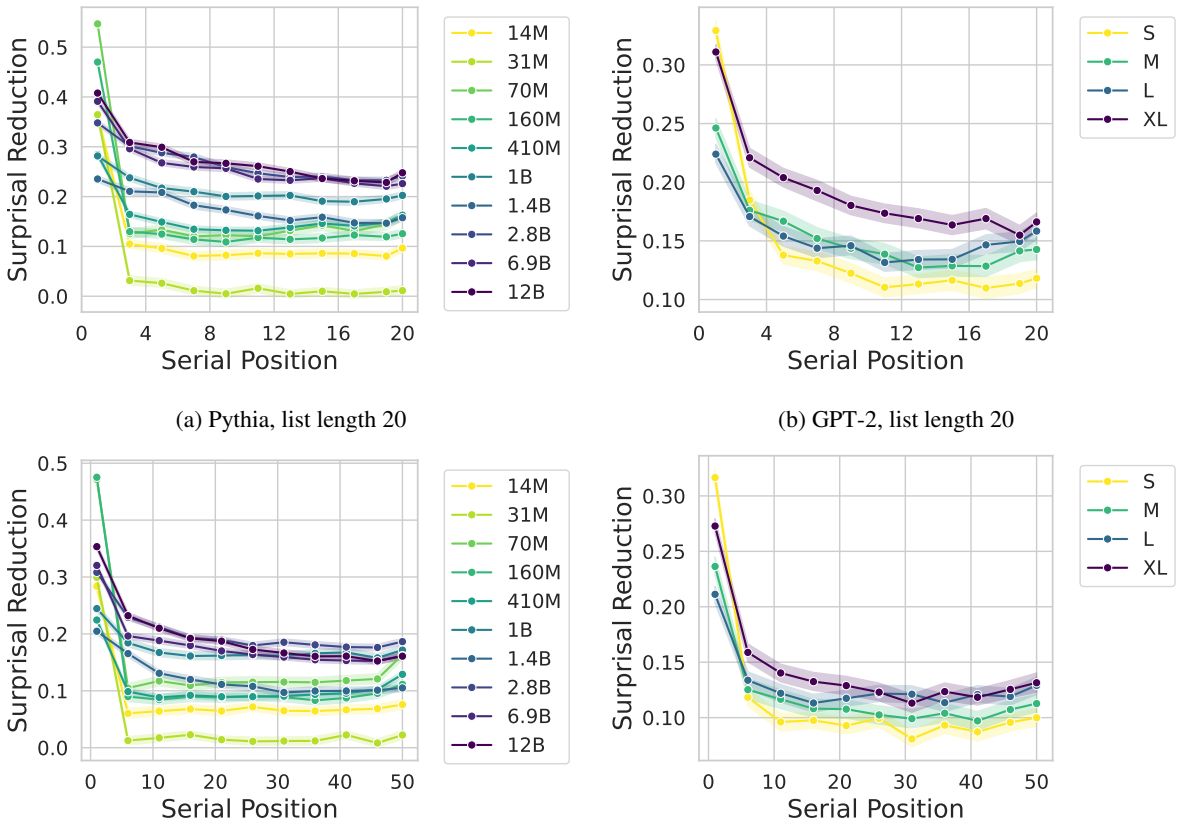

(a) Pythia, list length 20

(b) GPT-2, list length 20

(c) Pythia, list length 50

(d) GPT-2, list length 50

Figure 12: Pythia and GPT-2 item recognition results with an alternative prompt. Example of main prompt: *Please try to remember the following names: Elena, Clint, . . . , Adrian, Lisa. Now, after you received all this information, try to concentrate, drink a cup of coffee, go for a walk. Then please complete the following sentence using one of the names you read earlier. One of the names is Adrian*. Baseline prompt: *One of the names is Adrian*.




(a) Pythia, list length 20

(b) GPT-2, list length 20

(c) Pythia, list length 50

(d) GPT-2, list length 50

Figure 13: Pythia and GPT-2 item recognition results with an alternative prompt. Example of main prompt: *The cities I have traveled to include Edmonton, Prague, . . . , Barcelona, Miami. Next year I will again be visiting Barcelona*. Baseline prompt: *Next year I will again be visiting Barcelona*.