Unwrapping Circularity: Can Transformers Learn Languages with Circular Schemes?

Anonymous ACL submission

Abstract

The success of Transformer-based language models in NLP has sparked debate about their ability to simulate human language learning. Chomsky contends that these models indiscriminately acquire both natural and "impossible" languages. While recent studies have challenged this claim, the capacity of Transformers to handle unconventional linguistic structures remains underexplored. Inspired by natural and speculative languages with circular structural properties, this study examines the ability of GPT-2 to learn languages featuring circular schemes. We synthesize such circular languages by mapping original sequences onto textual circles and then relinearize them using parametric, mathematically invertible procedures that "unwrap" the circles into linear sequences. We train GPT-2 models on these relinearized corpora and assess the impact of linearization parameters by tracking structural distortion and measuring perplexity. Interestingly, high levels of distortion relative to the original structures do not necessarily correspond to increased perplexity, suggesting that GPT-2 is relatively insensitive to global token order during language acquisition. Instead, preserving local context during linearization plays a more critical role in model learning. Further analysis using surprisal differences reveals that positional shifts pose greater challenges to the model than changes in stride or direction, underscoring the nuanced effects of linearization strategies. These findings offer new insights into the inductive biases of Transformer-based models in acquiring unconventional linguistic structures.

1 Introduction

"As Frodo did so, he now saw fine lines, finer than the finest pen-strokes, running along the ring, outside and inside: lines of fire that seemed to form the letters of a flowing script." Tolkien (1954) certainly made a beautiful and vivid description of what a circularized text sequence may present. Unlike **rendered** circular sequence such as the One Ring inscription, Heptapod B—an alien language imagined in the novella *Stories of Your Life* (Chiang, 2002)—exemplifies an **intrinsic** circular language, where structure are fundamentally circular and constituents graphic. Circular patterns are not exclusive to fictional works. In biology, naturally occurring circular genetic sequences play essential roles in gene regulation, demonstrating that circular structures exist even at the molecular level of genetic information encoding (Vinograd and Lebowitz', 1966; Jeck and Sharpless, 2014; Ebbesen et al., 2017). These examples indicate that circular representation of information, whether naturally evolved or artificially constructed, are viable and meaningful systems, supporting the idea that structurally circular languages need not remain purely speculative.

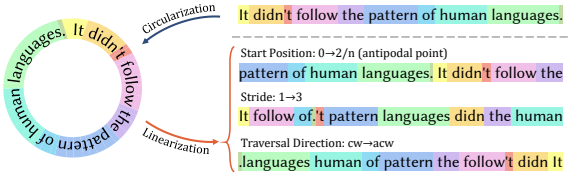

Figure 1: An illustration of circularization and linearization of a text sequence at token level. Since the raw sequence is standard English, the linearized sequences with altered schemes (parametrized by start position, stride and direction) become incomprehensible to human. However, they are equally valid and invertible ways to and "unwrap" a circular sequence.

Recent advances in neural language models based on Transformer architectures (Vaswani et al., 2023) have sparked discussions about whether such models can provide meaningful insights into human language processing and cognition. Notably, Chomsky et al. (2023) argued that, unlike humans, large language models (LLMs) are equally capable of learning both natural and "impossible" lan-

guages, implying that their ability to acquire language does not reflect human cognitive mechanisms (Chomsky, 2014). In response to this claim, Kallini et al. (2024); Yang et al. (2025); Xu et al. (2025) introduced a suite of artificially synthesized (typologically) "impossible" languages and demonstrated that Transformer-based models learn natural languages more effectively than those "impossible" counterparts, directly challenging Chomsky's assertion. This line of research aims to evaluate whether language models share similar limits on language learning with humans, so that the models may potentially provide analogical explanations on human-like intelligence.

Despite existing works considered the positional perspective to a certain degree, many of the proposed "impossible" language construction strategies are bound by linear relations or nondeterministic operations. Thus, it remains an open question **whether Transformers can learn languages with rule-based non-linear—particularly circular—schemes**, as is commonly observed in natural information encoding. This special inquiry inspires us to explore a geometrically guided approach: instead of applying arbitrary positional change or permutations, we design mathematically invertible circularization and linearization transformations. Circular languages generated by such process challenge linear-sequence-targeted Transformer-based language models, yet also avoid collapsing into over randomness or orderless bag-of-words, for it retains certain degree of proximity.

We parameterize a circularization (and inversely linearization) scheme by three key configurations: start position, stride and direction. Then, we present a anatomic analysis on their independent impact. Prior to main experiments, we also quantify the extent of distortions introduced by different schemes. We hypothesize that higher distortions in relinearized circular languages will lead to higher perplexities (indicating more difficult learning) for GPT-2. While the observations confirms that Transformer-based models struggle more to learn languages with circular schemes than natural ones. Contrary to our hypothesis, the results show that **greater distortions did not consistently increase perplexity**. Instead, GPT-2 **preferred linearization schemes preserving inter-token proximity, implying a human-like favor for structurally "natural" sequences**. Another suite of surprisal analyses indicate a conditional **negative correlation between positional learning and distortions** across most circular manipulations. However, the findings also show a **directional sensitivity on precise positional learning**.

This study aims to contribute to the debates about language learnability of LLMs and provides evidence for one of the key factors in making a language difficult for a Transformer-based language model: it is not the linguistic "impossibility" of a language that makes it less unlearnable, but simply due to its randomness in terms of information. In other words, even though LLMs and human both struggle with learning languages with disjoint compositions, their common difficulties does not imply they learn languages in the same way. We hope our study engage the broader debate on the relationship between LLMs and human cognition.

2 Related Work and Background

2.1 Transformer Learning of Unnatural Languages

The development of LLMs raises fundamental questions about what types of sequences transformers can learn, particularly regarding their ability to acquire structured linguistic patterns (Mielke et al., 2019; Liao et al., 2020; Strobl et al., 2024). Sinha et al. (2021) argued that masked language models primarily succeed by capturing high-order word co-occurrence patterns rather than true syntactic structure, as evidenced by their strong downstream performance even when trained on shuffled word-order data. More recent models, such as GPT-4, exhibit remarkable resilience—nearly perfectly reconstructing and interpreting scrambled inputs, often surpassing human capabilities (Cao et al., 2023).

However, recent studies challenge the extent of LLMs' learning capabilities. Kallini et al. (2024) examined GPT-2 small's ability to acquire synthetically constructed "impossible" languages and find that it struggles significantly compared to natural languages. Xu et al. (2025) reinforced this finding from a typological perspective, showing that models trained on typologically implausible languages generalize worse than those trained on natural languages. By contrast, Yang et al. (2025) reported mixed results, noting that some typologically impossible languages exhibit lower perplexity than natural languages. They hypothesize that this may be due to the preservation of constituency structure, a finding consistent with Sankaranarayanan et al. (2025), who showed that LLMs develop distinct

mechanisms for processing hierarchical versus linear grammars.

2.2 Circularity in Nature and Language

Circular structures play essential roles in nature. For example, circular DNA (Vinograd and Lebowitz', 1966) and RNA (Jeck and Sharpless, 2014) form covalently closed loops that regulate gene expression and exhibit distinct biological functions compared to their linear counterparts (Ebbesen et al., 2017). The circularization procedures explored in this study are partly inspired by the biological phenomenon of "circular splicing," a process generating closed-loop RNA molecules (Head, 1987, 1992) that has spurred formal research into circular splicing systems (Pixton, 1995).

In human-created symbolic systems, non-linear structures have historically appeared across diverse cultural and artistic traditions, such as Mayan glyphs (Kettunen and Helmke, 2005) and cuneiform tablets (Anderson and Levoy, 2002), encoding meaning in non-linear arrangements. However, while circular visual motifs are common, genuine circular linguistic structures—where meaning itself is constructed non-linearly—are exceedingly rare. Within linguistics, the term "circular language" typically refers to self-referential semantic phenomena (Leitgeb and Hieke, 2004) rather than structural non-linearity. In this study, we define circular languages as a subset of the constructed languages (Schreyer, 2021) exhibiting explicitly non-linear linguistic structures. Unlike most natural languages, where meaning unfolds sequentially through phonographic symbols, circular languages simultaneously or recursively present information, challenging the assumptions underlying standard NLP models that rely on sequential input processing. We thus extend prior research on "impossible" languages (Kallini et al., 2024; Yang et al., 2025) by introducing a novel class of synthetically constructed languages characterized by circular structures, a concept previously unexplored in this domain.

3 (Re)construction of Circular Languages

We distinguish between rendered circular languages, which appear circular only at the surface level—such as the ring inscription described in Tolkien (1954)—and intrinsic circular languages, where circularity is fundamentally embedded within the grammar and meaning-making processes, exemplified by Heptapod B from Chiang (2002).

A critical distinction lies in the inherent linearity preserved by rendered circular languages derived from natural human languages, as most human scripts are phonographic and thus inherently sequential (Sampson, 2015; Coon, 2020). For example, English is an alphabetic language, which intrinsically encodes sounds in linear sequences. Although, from a computational perspective, the tokenization used by LLMs introduces certain logographic-like behaviors—treating frequent words or subwords as indivisible units —— even constructed or "impossible" languages based on English inevitably retain elements of linearity and local dependencies.

In contrast, intrinsic circular languages do not inherit sequential constraints from spoken forms and thus represent semasiographic writing systems, encoding meaning directly without phonetic mediation (Powell, 2012). Therefore, to effectively simulate an intrinsic circular language, one would arguably need to construct it from inherently semasiographic systems, such as emoji. While exploring this avenue presents an intriguing direction, we leave such investigation to future work.

Since there is no existing natural corpus of circular languages, we synthesize a suite of circular languages with circularization and linearization schemes from a base natural language (e.g., English), as briefly illustrated in Figure 1. We define two granularity levels to specify the transformation: a *circular span* is the linguistic unit over which the circular transformation is applied (e.g., sentences, paragraphs), and a *atomic unit* is the minimal indivisible elements placed around the circle (e.g., tokens, words). In this study, we control circular span at sentence level and the atomic unit at the token-level.

We first implement a circularization function to map original linear sequences into a circular structure, represented as a set of rings of tokens with different "rotations" (or equivalently a ring of tokens with no fixed start point). Having obtained the text circles, we then explore various linearization schemes parametrized by start position p, stride s and direction d, to translate the circular set into linearly aligned sequences. It is noteworthy that linearization and circularization processes are **inverse operations** with specified parameters.

3.1 Definition

Concretely, let $T = (t_0, t_1, \ldots, t_{n-1})$ be a linear sequence of n tokens. Then, a circularization function $\mathcal{C}(T; p, s, d)$ can be defined by first constructing the ordered sequence $(t_{(p+d\cdot s\cdot i) \bmod n})_{i=0}^{n-1}$, which is parametrized by start position $p \in \{0, 1, \ldots, n-1\}$, stride $s \in \mathbb{N}^+$, and traversal direction $d \in \{-1, +1\}$ (-1 = anticlockwise, +1 = clockwise). Since the tokens are arranged on a circle, two sequences that differ only by a rotation are equivalent. Thus, the circularization function is defined as the equivalence class of all rotations of the above sequence:

$$\mathcal{C}(T; p, s, d) = \{(t_{(p+d\cdot s\cdot i) \bmod n})_{i=0}^{n-1} : k = 0, 1, \ldots, n-1\}.$$

Thus, $\mathcal{C}(T; p, s, d)$ is the equivalence class of all rotations derived from T using parameters p, s and d, representing a circular sequence.

We can now define an inverse linearization operation $\mathcal{L}$ that takes a circular sequence and produces a unique linear sequence using the same parameters. Formally, we write:

$$L(\mathcal{C}(T); p, s, d) = (t_{(p+d\cdot s\cdot i) \bmod n})_{i=0}^{n-1}.$$

Having established the definition, we next explore various linearization schemes with typical specifications of those parameters. We also illustrate all tested schemes in Table 1.

3.2 Start Position (p)

As stated in § 3.1, it is noteworthy that p is assigned during the initial linearization process, that is, "start position" here refers to the index of the input linear sequence. Therefore, for an arbitrary circular language without known p, one has to manually define a reference point. A common practice in bio-informatics is selecting biologically meaningful anchor point (e.g., replication origin) (Zhang et al., 2020). In following study, since we build our circular languages on original linear ones, thus we assume a known p. In particular, we find following three values of start position to be most interesting.

1. **Anchor** ($p = 0$) We refer the first token of the original linear sequence as "anchor", namely $p = 0$. Specially, setting $p = 1$, $s = 1$ and $d = +1$, the relinearized sequence becomes:

$$\begin{aligned} L(\mathcal{C}(T); 0, 1, +1) &= (t_{(0+1\cdot 1\cdot i) \bmod n})_{i=0}^{n-1} \\ &= (t_{i \bmod n})_{i=0}^{n-1} \\ &= (t_0, t_1, \ldots, t_{n-1}). \end{aligned}$$

Under this configuration, the linearization process is able to reconstruct the original linear sequence input, hence we use this method our **control** group.

2. **Antipodal Point** ($p = 2/n$) When mapped onto a circle, the diametrically opposite of the anchor is of great interest, for it represents the greatest circle distance with the anchor, hence termed "Antipodal Point". Concretely, we define this specification as:

$$p = \begin{cases} \frac{n}{2}, & \text{if } n \text{ is even,} \\ \left\lfloor \frac{n}{2} \right\rfloor, & \text{if } n \text{ is odd.} \end{cases}.$$

3. **Random Start** (p = random index) For each original linear sequences, the start position is randomly picked. Formally, $p \sim \text{Uniform}\{0, 1, \ldots, n-1\}$.

3.3 Stride (s)

The stride or step size determines the rate or "speed" of traversal around the circle. Higher stride values introduce non-local permutations and might break proximity and syntax dependencies, so it demonstrate a way to test robustness. This intuition also seen in cryptography, a well-known step encoding that resembles this linearization process is refers as Caesar cipher (Luciano and Prichett, 1987).

1. **One Step** ($s = 1$) This is the normal rotation, taking one-token length walk each time. Note that on a circular sequence of length n, if we take a stride of $n + 1$ steps, then we will travel around the full circle, return to the origin and then take one more step. We define all strides that behave equivalently to $s = 1$:

$$\{s \in \mathbb{Z} \mid s \equiv 1 \pmod{n}\} = \{1 + kn \mid k \in \mathbb{Z}\}.$$

2. **Minimal Coprime** (assign s to the minimal non-1 coprime) If the step size and n had a common divisor greater than 1, some indices would be skipped and the linearization process would not cover all tokens. To ensure full coverage—strides of equal step

Parameter	Specification (p, s, d)	Example
Start Position (p)	ANCHOR (0, 1, cw)	↻ It didn ’t follow the pattern of human languages .
	ANTIPODAL POINT (0.5, 1, cw)	pattern of human languages . ↻ It didn ’t follow the
	RANDOM START (rand, 1, cw)	’t follow the pattern of human languages . ↻ It didn
Stride (s)	ONE STEP (0, 1, cw)	↻ It didn ’t follow the pattern of human languages .
	MINIMAL COPRIME (0, mcp, cw)	↻ It follow of . ’t pattern languages didn the human
Direction (d)	CLOCKWISE (0, 1, cw)	↻ It didn ’t follow the pattern of human languages .
	ANTICLOCKWISE (0, 1, acw)	. languages human of pattern the follow ’t didn It ↻
	BIDIRECTIONAL CW (0, 1, bi_cw)	↻ It didn . ’t languages follow human the of pattern
	BIDIRECTIONAL ACW (0, 1, bi_acw)	pattern the of follow human ’t languages didn . It ↻

Table 1: Tested linearization schemes with different parameter specifications and their corresponding with examples. Example sentence are based on the circularization illustrated in Figure 1. Specification names are followed by their detailed parameters. Tokens are colored to improve visual differentiation. ANCHOR, ONE STEP and CLOCKWISE are essentially the same, which outputs the identical sequence as the linear sequence input. An special marker token ↻ is placed right before the first token of the original sequence (i.e., t_0).

size visit all positions on a circle of length n—stride s must be a coprime of n, satisfying $\gcd(s, n) = 1$. Since we already discuss the case of $s = 1$, which also matches the criteria, we exclude it and look for the smallest coprime stride greater than 1. Concretely,

$$s = \min \{s \in \mathbb{N} \mid 2 \leq s < n,\ \gcd(s, n) = 1\}.$$

3.4 Traversal Direction (d)

The direction $d \in \{-1, +1\}$ determines which way one "scan" the circular or linear sequence when taking each stride step. In addition to standard clockwise and anticlockwise, we also introduce two "bidirectional" methods that symmetrically travel from a given central point.

1. CLOCKWISE ($d = +1$) The standard clockwise traversal (i.e., move "foward" around the circle), and we match it with rightward in horizontal linear sequence.

2. ANTICLOCKWISE ($d = -1$) Oppositely to clockwise, traverse the tokens in anticlockwise order (i.e., counterclockwise, backward). Similarly, we also match it with leftward in horizontal linear sequence.

3. BIDIRECTIONAL CLOCKWISE (bi-cw) This method creates a mirror-symmetric traversal centered on the start point. Unlike single-direction traversal, it prioritizes proximity more, gradually expanding outward. Therefore, we term it "bidirectional-clockwise (bi-cw)". To achieve this, we need to slight revise our linearization function to:

$$\mathcal{L}_{\text{bi-cw}}(\mathcal{C}(T)) = (t_{(p+(-1)^i \cdot s \cdot \lceil \frac{i}{2} \rceil) \bmod n})_{i=0}^{n-1}.$$

4. BIDIRECTIONAL ANTICLOCKWISE (bi-acw) This method is identical to BIDIRECTIONAL CLOCKWISE except the first move is made counterclockwise. It can be implemented by changing the power of -1 to $i + 1$:

$$\mathcal{L}_{\text{bi-acw}}(\mathcal{C}(T)) = (t_{(p+(-1)^{i+1} \cdot s \cdot \lceil \frac{i}{2} \rceil) \bmod n})_{i=0}^{n-1}.$$

4 Experiments

The experiments conducted in this study comprise three main parts. First, we evaluate the distortion introduced by each linearization scheme, measured using the normalized cyclic editing distance between the original training sequences and their re-linearized versions. Second, we investigate the influence of three critical linearization parameters by systematically varying their configurations and analyzing their correlations with distortion levels. Finally, we assess the surprisal differences produced by the trained models for a predetermined sentence-initial marker, aiming to determine whether the models effectively learn to retrieve the original sentence-initial token after undergoing cyclic transformations.

Setup For comparability with existing research targeting Transformer-based models' learning capabilities on unnatural languages (Kallini et al., 2024; Yang et al., 2025; Xu et al., 2025), we select the BabyLM dataset strict-small track (Warstadt et al., 2023) as the base for constructing circular languages and utilize GPT-2 Small (Radford, 2018) as the base model. We train GPT-2 Small models using mostly default hyperparameters: the context length is set to 1,024, and the batch size to 512. As

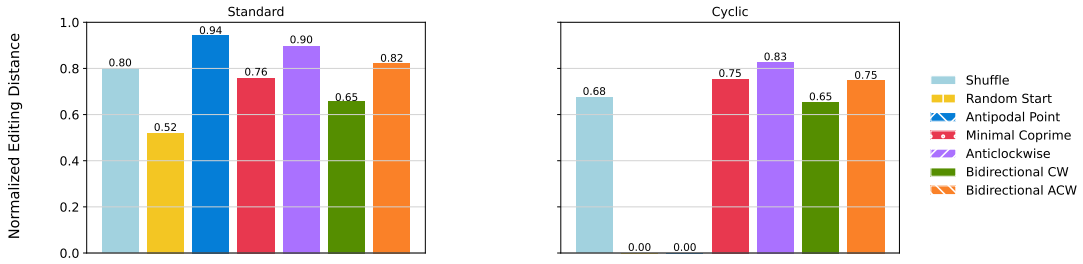

Figure 2: Distortions for linearization schemes averaged over the entire train set sentences, as measured in normalized editing distances. Left plot shows standard editing distances, while the right plot demonstrates cyclic one. "0" indicates prefect alignment. ANCHOR, ONE STEP and CLOCKWISE results are discarded since they share the same parameters that yield relinearized sequences identical to the original.

Yang et al. (2025) demonstrated that GPT-2 begins to overfit after 1,000 steps on the 10M BabyLM training set, we follow their setup and train for 1,200 total steps with 120 warm-up steps. More training details can be found in § 5.

4.1 Preliminary: Measuring Distortion

To assess the sequential distortions caused by the circularization and linearization process, we measure the normalized edit distance (Levenshtein distance) (Marzal and Vidal, 1993; Yujian and Bo, 2007) between pairs of the original linear sequence and the relinearized one. In addition, since rotational distortion is negligible in circular structure, we are interested in **normalized cyclic edit distance** (Charalampopoulos et al., 2024):

$$\mathrm{NCED}(\mathcal{C}(T), T') = \frac{1}{n} \min_{k=0}^{n-1} \mathrm{ED}\Big(\big(t_{(p+d\cdot s\cdot(i+k)) \bmod n}\big)_{i=0}^{n-1}, T' \Big)$$

where ED is the standard edit distance function (provided in Appendix B, as well as a general version for comparing sequences with different lengths). $\big(t_{(p+d\cdot s\cdot(i+k)) \bmod n}\big)_{i=0}^{n-1}$ is the k-rotated version of the circular sequence, and the minimization over k finds the rotation that minimizes the edit distance.

Results Figure 2 shows the distortion measurements. For standard editing distance, RANDOM START introduces the least distortion, and MINIMAL COPRIME is also relatively stable. ANTIPODAL POINT and ANTICLOCKWISE have the highest distortions. For cyclic editing distance, RANDOM START and ANTIPODAL POINT achieve perfect alignment as anticipated, for they both are rotational transforms. MINIMAL COPRIME and BIDIRECTIONAL CLOCKWISE produce nearly identical values with their standard editing distances, while their anticlockwise counterparts yield noticeably lower (-0.07) distortion than measured on the standard editing distances.

4.2 Main Experiment 1: Investigating Impact of Linearization Parameters

We next evaluate how specific linearization parameters affect GPT-2's language acquisition as measured by perplexity (Jelinek et al., 1977), a well-established metric for quantifying model learning progress, following previous practices (Kallini et al., 2024; Yang et al., 2025).

Hypothesis Given equal training steps, models trained on relinearized circular languages with higher sequential distortions (relative to the original linear languages) will yield higher perplexities compared to models trained on languages with lower distortions.

Results The perplexity curves over training steps are grouped by their linearization configurations in Figure 3. For the start position, the control set consistently achieves the lowest perplexity, supporting the idea that the natural order is the easiest for GPT-2 to learn. Interestingly, despite introducing the highest distortion, ANTIPODAL POINT achieves consistently lower perplexities than RANDOM START, contradicting our hypothesis.

Regarding stride, increasing the stride significantly hinders language learning, demonstrated by consistently higher perplexities with the MINIMAL COPRIME stride. For directional changes, surprisingly, ANTICLOCKWISE achieves perplexities similar to CLOCKWISE. In contrast, BIDIRECTIONAL ACW obtains noticeably lower perplexities than its clockwise counterpart, although both bidirectional schemes yield slightly higher perplexities than unidirectional schemes. This aligns with the

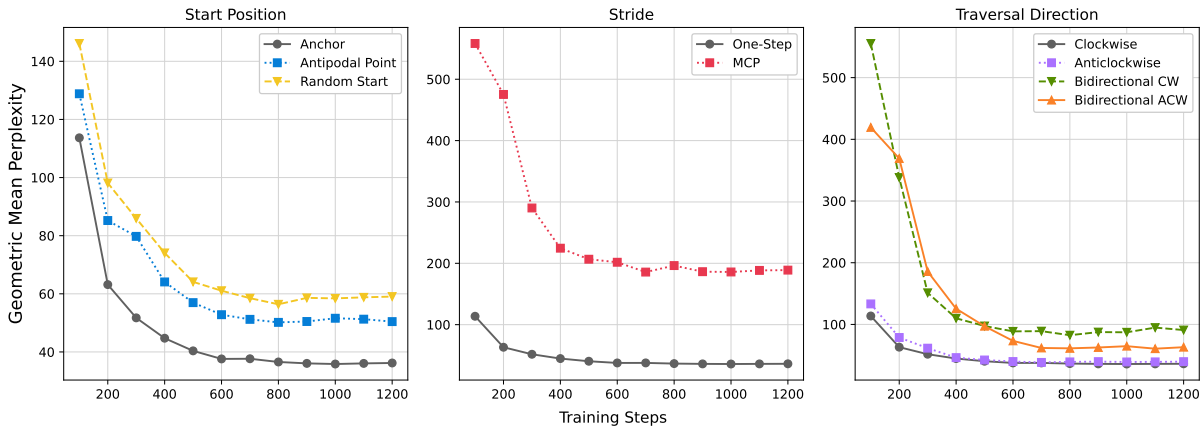

Figure 3: Geometric mean perplexities on a sample of 10,000 test sentences for linearization schemes grouped by parameter configuration. See Figure 6 for an expanded version.

intuition that bidirectional orders impose greater cognitive demands on memory and attention. However, this impact is less severe than the effect caused by changing the stride.

Overall, our hypothesis is not supported. The results suggest that **higher distortions do not necessarily lead to higher perplexities**. Structurally preserved schemes (ANTIPODAL POINT and RANDOM START) achieve comparatively lower perplexities, confirming a **human-like preference for "natural" structures**. More importantly, we observe that schemes that alter the start position and traversal direction, thereby largely preserving inter-token proximity, achieve lower perplexities compared to MINIMAL COPRIME, which significantly disrupts neighboring relations. These findings indicate that **maintaining stable proximity among tokens contributes most significantly to GPT-2's acquisition of circular languages**.

4.3 Main Experiment 2: Evaluating Anchor Retrieval

Sentence-initial tokens hold linguistic importance, frequently carrying syntactic cues and indicating subjects in an SVO language like English (Dorgeloh, 2004). In our context, the first token has geometric significance, serving as the key to reconstructing the original or "natural" token order. As illustrated in Table 1, we intentionally insert a marker token ↻ immediately before the original first token (i.e., t_0).

Surprisal (Goodkind and Bicknell, 2018; Wilcox et al., 2018, 2023), defined as the negative log probability of a token given its preceding context, measures how unexpected a token is according to the language model. We calculate surprisal differences to evaluate whether the trained model learns to recognize the original sentence-initial token. Specifically, rather than using standard surprisal (see Appendix B.4), we rotate the token sequence so that the target token is predicted using the complete sentence context, effectively treating the sentence as circular. The cyclic surprisal of the marker token at position k is thus computed as:

$$S_{\text{cyc}}(\circlearrowright) = -\log_2 p\Big(\circlearrowright \mid t'_{k+2}, t'_{k+3}, \ldots, t'_{n-1}, t'_0, t'_1, \ldots, t'_{k-1}\Big).$$

With this method, regardless of the position of ↻ in T', the model uses the entire sentence (except ↻ and $t'k+1$) as context. Following Kallini et al. (2024), we compare surprisal differences between test sentences and minimal copies where the marker token is removed. We deliberately exclude the token immediately following the marker ($t'k+1$) when calculating the marker's surprisal, ensuring meaningful comparisons between surprisals. This setup specifically assesses learning of **positional structures** rather than grammatical rules.

$$S_{\text{cyc}}(t'_{k+1}) = -\log_2 p\Big(t'_{k+1} \,\Big|\, t'_{k+2}, t'_{k+3}, \ldots, t'_{n-1}, t'_0, t'_1, \ldots, t'_{k-1}\Big),$$

$$\text{Surprisal Difference} = S_{\text{cyc}}(t'_{k+1}) - S_{\text{cyc}}(\circlearrowright).$$

A large surprisal difference indicates the model has learned to expect the marker specifically before the original sentence-initial token, signifying successful recognition of the token that should begin the natural linear sequence.

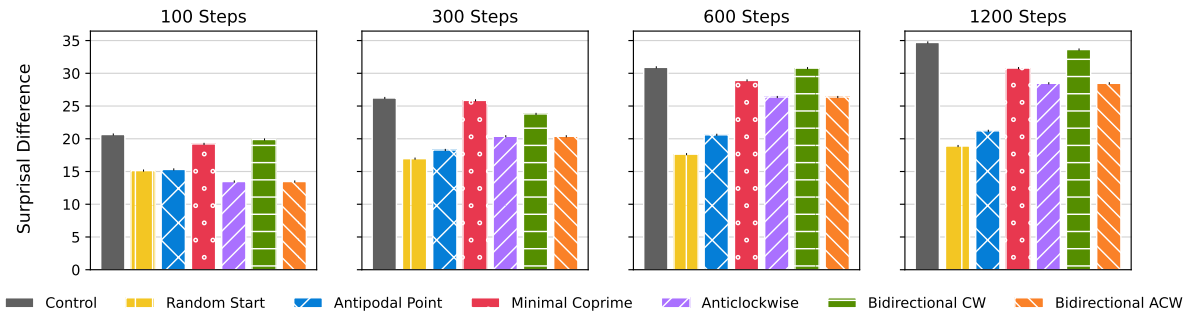

Figure 4: Mean cyclic surprisal differences between the marker token (⟳) and the following token for each linearization scheme over training steps.

Hypothesis Given that distortions directly affect token positions, we anticipate that the mean cyclic surprisal difference across test pairs will be larger for linearization schemes applying less distortions compared to those applying greater distortions.

Results As shown in Figure 4, the control set achieves consistently the highest cyclic surprisal differences, indicating a strong ability to predict the marker in its correct position (before the sentence-initial token). Unsurprisingly, BIDIRECTIONAL CW and MINIMAL COPRIME rank second and third, respectively, since their sentence-initial tokens remain unchanged during circularization. Conversely, ANTICLOCKWISE and BIDIRECTIONAL ACW suffer from reversed text, highlighting a clear **directional sensitivity**.

Interestingly, ANTIPODAL POINT and RANDOM START, despite representing the highest and lowest distortions respectively, yield the lowest surprisal differences. This observation indicates that the models struggle to learn this positional relationship and fail to reliably identify the original sentence-initial token in these relinearized sentences, contradicting our hypothesis. Furthermore, their surprisal differences increase much more slowly over training compared to other methods.

Overall, our hypothesis is partially rejected. While the majority of schemes (five out of seven) show negative correlations between surprisal difference and distortion, ANTIPODAL POINT and RANDOM START are notable exceptions. Transformer models demonstrate surprising resilience in positional learning under moderate distortion levels but fail under extreme conditions. The particularly low surprisal differences for the exceptions further indicate that **shifting the start position challenges the model more severely than changes in stride or traversal direction when recovering the natural linear sequence**.

5 Conclusions

We have shown that languages with circular schemes pose distinct challenges for Transformer-based models. Our findings suggest that the primary difficulty for GPT-2 does not stem from distortions from the natural sequential order, but rather from disruptions to token proximity and the randomness introduced by different linearization strategies. This highlights GPT-2's greater reliance on local contextual cues than on strict sequential ordering. Additional experiments on cyclic surprisal differences reveal that GPT-2's ability to recover natural order depends heavily on positional accuracy and consistency. Most linearization schemes exhibit a negative correlation between surprisal differences and distortion, reinforcing the importance of preserving local token relationships. Both sets of results emphasize that maintaining token proximity and stable local structures is crucial for effective language learning. Interestingly, this may give language models an advantage over humans in acquiring circular languages that preserve inter-token dependencies.

Our exploration of circular language learning underscores that while "impossible" languages are generally more difficult to acquire for both humans and language models, their respective learning biases may diverge. In particular, Transformer models like GPT-2 may be better suited than humans to learn certain classes of these languages—such as those with circular schemes. While our study attempts to isolate the independent effects of linearization parameters, future work should explore how these parameters interact, offering a deeper understanding of model inductive biases in unconventional language learning.

Limitations

To establish comparability with related literature (Kallini et al., 2024; Yang et al., 2025), we construct our circular languages based on English and experiment using GPT-2 Small models. However, we acknowledge that employing non-English languages with distinct linguistic features—such as different writing systems (e.g., Arabic, which is written from right to left) or languages with notable long-distance dependencies (Futrell et al., 2015)—as bases for constructing circular languages may yield different findings. Exploring how linguistic typology influences Transformers' ability to learn circular structures would be a valuable future direction. Additionally, experimenting with models across multiple parameter sizes or architectures could enhance the generalizability of our findings. We intend to expand the scope of our base language models contingent upon resource availability. Finally, as discussed in § 3.1, we anticipate that splicing textual sequences at varying granularity levels, such as at the character or word level, could provide insightful extensions to our analysis.

Ethics Statement

The synthetic circular languages employed in this study are artificial constructs designed solely for evaluating computational language models. Our results and interpretations pertain specifically to model performance and should not be directly extrapolated to human cognitive mechanisms without further cognitive validation. The BabyLM dataset (Warstadt et al., 2023) is a standard, ethically vetted NLP benchmark; nevertheless, we acknowledge the importance of dataset diversity and limitations. All computational experiments were conducted transparently, reporting resource usage to ensure reproducibility and ethical compliance. We encourage responsible interpretation and use of our findings, acknowledging the potential theoretical use of our approaches in adversarial or unintended contexts.

Reproducibility Statement

The GPT-2 models were trained for 1,200 training steps, including 120 warm-up steps. This proportion aligns with previous studies (Kallini et al., 2024; Yang et al., 2025). Our circular languages introduce an additional special token ↻, increasing the vocabulary size to 50,258 (the default GPT-2 vocabulary size is 50,257). Experiments were conducted using two NVIDIA A40 GPUs (48GB). Each pretraining experiment required approximately 9 hours per random seed, resulting in an estimated total training time of 400 hours.

References

S.E. Anderson and M. Levoy. 2002. Unwrapping and visualizing cuneiform tablets. *IEEE Computer Graphics and Applications*, 22(6):82–88.

Qi Cao, Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Unnatural error correction: Gpt-4 can almost perfectly handle unnatural scrambled text. *Preprint*, arXiv:2311.18805.

Panagiotis Charalampopoulos, Solon P Pissis, Jakub Radoszewski, Wojciech Rytter, Tomasz Waleń, and Wiktor Zuba. 2024. Approximate circular pattern matching under edit distance. *arXiv preprint arXiv:2402.14550*.

Ted Chiang. 2002. Story of your life. *Stories of your life and others*, pages 91–145.

Noam Chomsky. 2014. *The minimalist program*. MIT press.

Noam Chomsky, Ian Roberts, and Jeffrey Watumull. 2023. Noam chomsky: The false promise of chatgpt. *The New York Times*.

Jessica Coon. 2020. The linguistics of arrival: Heptapods, field linguistics, and universal grammar. In *Language Invention in Linguistics Pedagogy*. Oxford University Press.

Heidrun Dorgeloh. 2004. Conjunction in sentence and discourse: sentence-initial and and discourse structure. *Journal of Pragmatics*, 36(10):1761–1779. Pragmatics of Discourse.

Karoline K Ebbesen, Thomas B Hansen, and Jørgen Kjems. 2017. Insights into circular rna biology. *RNA biology*, 14(8):1035–1045.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th workshop on cognitive modeling and computational linguistics (CMCL 2018)*, pages 10–18.

Tom Head. 1987. Formal language theory and dna: an analysis of the generative capacity of specific recombinant behaviors. *Bulletin of mathematical biology*, 49:737–759.

Tom Head. 1992. Splicing schemes and dna. *Lindenmayer systems: impacts on theoretical computer science, computer graphics, and developmental biology*, pages 371–383.

William R Jeck and Norman E Sharpless. 2014. Detecting and characterizing circular rnas. *Nature biotechnology*, 32(5):453–461.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Harri Kettunen and Christophe Helmke. 2005. Introduction to maya hieroglyphs. In *Workshop Handbook*, pages 5–10.

Hannes Leitgeb and Alexander Hieke. 2004. Circular languages. *Journal of Logic, Language and Information*, 13:341–371.

Yi Liao, Xin Jiang, and Qun Liu. 2020. Probabilistically masked language model capable of autoregressive generation in arbitrary word order. *Preprint*, arXiv:2004.11579.

Dennis Luciano and Gordon Prichett. 1987. Cryptology: From caesar ciphers to public-key cryptosystems. *The College Mathematics Journal*, 18(1):2–17.

A. Marzal and E. Vidal. 1993. Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):926–932.

Sabrina J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. What kind of language is hard to language-model? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4975–4989, Florence, Italy. Association for Computational Linguistics.

D. Pixton. 1995. Linear and circular splicing systems. In *Proceedings First International Symposium on Intelligence in Neural and Biological Systems. INBS'95*, pages 181–188.

Barry B Powell. 2012. *Writing: Theory and history of the technology of civilization*. John Wiley & Sons.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Geoffrey Sampson. 2015. *Writing systems*. Equinox Publishing Limited.

Aruna Sankaranarayanan, Dylan Hadfield-Menell, and Aaron Mueller. 2025. Disjoint processing mechanisms of hierarchical and linear grammars in large language models. *arXiv preprint arXiv:2501.08618*.

Christine Schreyer. 2021. Constructed languages. *Annual Review of Anthropology*, 50(Volume 50, 2021):327–344.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *Preprint*, arXiv:2104.06644.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. 2024. What formal languages can transformers express? a survey. *Transactions of the Association for Computational Linguistics*, 12:543–561.

John Ronald Reuel Tolkien. 1954. *The fellowship of the ring*. Houghton Mifflin.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Jerome Vinograd and Jacob Lebowitz'. 1966. Physical and topological properties of circular dna. *The Journal of General Physiology*, 49(6):103–125.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan G Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P Levy. 2023. Testing the predictions of surprisal theory in 11 languages. *Transactions of the Association for Computational Linguistics*, 11:1451–1470.

Tianyang Xu, Tatsuki Kuribayashi, Yohei Oseki, Ryan Cotterell, and Alex Warstadt. 2025. Can language models learn typologically implausible languages? *Preprint*, arXiv:2502.12317.

Xiulin Yang, Tatsuya Aoyama, Yuekun Yao, and Ethan Wilcox. 2025. Anything goes? a crosslinguistic study of (im)possible language learning in lms. *Preprint*, arXiv:2502.18795.

Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1091–1095.

Jinyang Zhang, Shuai Chen, Jingwen Yang, and Fangqing Zhao. 2020. Accurate quantification of circular rnas identifies extensive circular isoform switching events. *Nature communications*, 11(1):90.

A An Example of Another Linearization Parameter

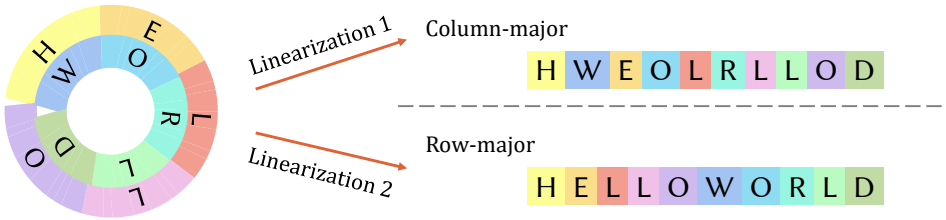

Figure 5: An example of flattening a two-dimension ring layout in row-major or column-major order, which can also be interpreted as different "writing" direction (vertical first or horizontal first).

B Supplementary Equations and Figures

B.1 Normalized Cyclic Editing Distance for Comparing Sequences with Different Lengths

$$\mathrm{NCED}(\mathcal{C}(T), T') = \min_{k=0}^{n-1} \frac{\mathrm{ED}\Big(\big(t_{(p+d\cdot s\cdot(i+k)) \bmod n}\big)_{i=0}^{n-1},\, T'\Big)}{\max\{n, m\}}.$$

B.2 Standard Editing Distance Function

$$\mathrm{ED}(T, T') = \begin{cases} |T'|, & \text{if } T = \varnothing, \\ |T|, & \text{if } T' = \varnothing, \\ \min\Big\{\mathrm{ED}(T[1:], T') + 1,\ \mathrm{ED}(T, T'[1:]) + 1,\ \mathrm{ED}(T[1:], T'[1:]) + \delta(t_0, t'_0)\Big\}, & \text{otherwise,} \end{cases}$$

$$\text{where } \delta(t_0, t'_0) = \begin{cases} 0, & \text{if } t_0 = t'_0, \\ 1, & \text{otherwise.} \end{cases}$$

B.3 Perplexities Results Including NondeterministicShuffle

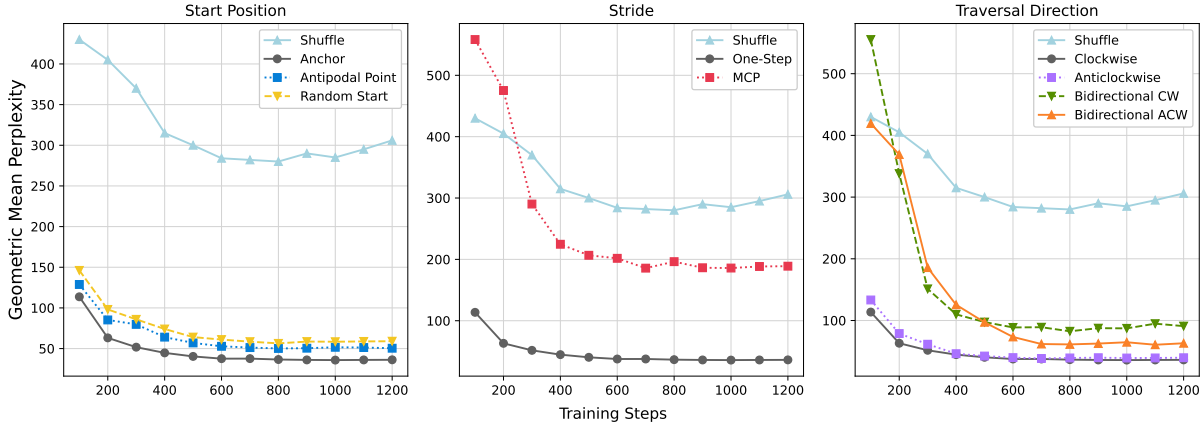

Figure 6: Perplexities on a sample of 10,000 test sentences for each linearized language model over training steps. "Shuffle" results (NondeterministicShuffle) are cited from (Yang et al., 2025).

B.4 Standard Surprisal Function

$$S_{\text{std}}(\circlearrowleft) = -\log_2 p\Big(\circlearrowleft \mid t'_0, t'_1, \ldots, t'_{k-1}\Big).$$