CURLY CITATION: A Token-Efficient Citation Representation for Large Language Model Contexts

Nabil Islam[1*]

[1*]Independent Researcher.

Corresponding author(s). E-mail(s): w3nabil@gmail.com;

Abstract

Citations provide supporting evidence for claims in large language model (LLM) contexts, but existing citation formats can introduce input-token overhead in long documents containing hundreds of references or more. We propose Curly Citation, a compact citation representation that preserves the information required to resolve a cited source while reducing repetitive bibliographic and URL information. It represents each reference in a fixed format using curly braces, an index, a predefined parameter, an endpoint and a pair of backticks. Preliminary benchmarks across multiple tokenisers and associated models show that popular conventional citation formats like IEEE, Harvard, APA and 20 other formats require 31–59 input-tokens to represent a selected citation. Curly Citation, however, reduces the input-token cost representation by approximately 50.00%–68.18% compared to selected existing citation formats. In a document-scale benchmark containing 150 citations, replacing IEEE citations with Curly Citation reduced total document input-token consumption by 11.33%–12.37% across the evaluated tokenisers. These results indicate that Curly Citation may provide a practical approach for lowering citation-related token overhead in LLM document processing.

Keywords: Large Language Models (LLMs), Token Efficiency, Context Efficiency, Citation Compression, Citation Representation, Token-Efficient Citation Format, Curly Citation (Curly)

1 Introduction

Citations provide supporting evidence for claims in large language model (LLM) contexts and allow readers or downstream systems to identify the sources underlying generated information [1]. Removing citations or source references may reduce the input-token requirements of a document, but it also removes information that can assist with source identification, claim grounding, and subsequent verification. Zeng et al. [2] reported a 13.83% improvement in Claim Grounding Rate when references were provided in a real-world LLM benchmark. Consequently, citation information can provide useful context [3], while conventional citation representations also introduce additional token consumption.

The problem is further complicated by the reliability of LLM-generated bibliographic information. Chen et al. [4] have shown that LLM-generated citations can contain field-level citation hallucinations, with bibliographic fields such as author information displaying high error rates across models and citation settings. A citation may therefore identify a valid or accessible source while containing incorrect or unreliable metadata. This creates a gap between the information required to identify and resolve a source and the additional bibliographic information traditionally presented for human readers.

This gap matters most when a document is intended for LLM processing. Many of the fields and/or values required by conventional citation formats are not strictly necessary to resolve a source, a problem examined in detail in Section 2.

To address this challenge, we propose Curly Citation, a compact citation representation designed for LLM document contexts. Curly separates source identification from conventional bibliographic presentation and represents a citation using an index, a predefined parameter, and a parameter value. The parameter identifies the resolver or source registry, while the value provides the information required to locate the underlying source. This representation removes repeated resolver information and unnecessary bibliographic metadata from the primary document representation while retaining a structured mechanism for source resolution.

Curly is not intended to replace established citation standards such as IEEE, APA, MLA, or Chicago in conventional academic publishing. These standards serve human-orientated purposes, including bibliographic identification, attribution, formatting consistency, and publication requirements. Instead, Curly aims to represent source references within human-oriented document contexts where repeated bibliographic and resolver information can contribute high input-token overhead. It treats citation representation as a machine-oriented layer that can coexist with conventional human-readable citation formats rather than replacing them.

2 Background and Related Studies

First, citation information is valuable for grounding and source verification, but its conventional representation may consume a non-trivial portion of the available context

window [1]. Our document-scale benchmark illustrates this effect using a technical report with 150 IEEE citations. Depending on the tokeniser, the complete IEEE representation introduced approximately 8,495—11,809 additional tokens relative to the appropriate citation-free document (Table 3). This demonstrates that the cost of citation is not limited to individual references. It can become a measurable component of the total context size in citation-dense documents.

Second, conventional citation formats frequently include information such as author names, article titles, journals, publishers, publication dates, and page numbers. These fields serve important purposes in human-oriented academic communication, but a source-resolving system may not require all of them to identify the resource. This is not a new observation: the Dublin Core Metadata Initiative's own guidelines for encoding bibliographic citations draw a similar line, noting that citation description exists both "for reading by humans after retrieval of a record" and, separately, "for machine reading" [5], and treats the two as requiring different handling rather than the same fields serving both purposes equally well. Similarly, Crossref's guidance for DOI landing pages requires a full bibliographic citation on the page specifically so a human reader can confirm they have reached the correct item [6], which is a human-verification purpose rather than a machine-resolution one. When the source can be located through a stable identifier or URL, repeatedly representing metadata that can be retrieved from the source itself introduces additional textual and token overhead. Appendix E illustrates this directly: when we asked Sonnet 5 for the abstract of a paper published after its training data, without retrieving information from an external URL, it hesitated even though conventional citation was present. We asked the LLM to infer or verify the information from its existing knowledge, but it explicitly stated that it could not. When given permission to use web fetching, Sonnet 5 retrieved information from the URL without needing any conventional citation. This indicates that bibliography data is not necessarily needed when asked to learn about the work. This does not mean bibliographic metadata is unnecessary, but it may primarily serve human-oriented referencing rather than machine resolution, a distinction the standards above already draw, even if not with LLMs in mind.

Third, a source of unnecessary repetition is the resolver information embedded directly in conventional URLs. For example, a DOI citation may contain a repeated resolver prefix `https://doi.org/`, even though this prefix is identical for every DOI and remains the same. In the `o200k_base` tokeniser used in our benchmark, the resolver prefix `https://doi.org/` requires five tokens, whereas the word `doi` requires one token. Consequently, replacing the repeated resolver prefix with the word saves four tokens per occurrence. Across 100 DOI references, this representation alone can eliminate approximately 400 tokens, before accounting for any additional bibliographic compression. The word `doi` is identical, using the short name instead of the resolver URL can produce substantial reductions in input-token consumption.

Fourth, a DOI URL redirects to the source where the content is present. In most cases, the redirection lands on an abstract page which contains the basic metadata of a work for conventional citations. This is by design rather than incidental: Crossref's own guidance states that DOIs should resolve to a landing page containing metadata about

the item rather than to the content itself, so that a human can confirm they have arrived at the correct item and a machine can retrieve supplementary information [6]. For instance, an arXiv DOI or paper generally redirects to `https://arxiv.org/abs/x.x`, a page which contains the abstract of the paper and the author information when using conventional citation. However, the full paper exists in `https://arxiv.org/html/x.x` in HTML or `https://arxiv.org/pdf/x.x` in PDF format. When an AI system is instructed using a conventional citation format it reaches an abstract page and must then locate the path for html or pdf format for the work. This introduces additional computational cost, and it is exactly the kind of cost a human-verification-first design does not need to account for but a machine-resolution-first one does.

Fifth, conventional citation systems also vary considerably in their syntax and optional fields. Depending on the citation style, information may appear in different orders and some fields may be omitted or included under different circumstances. This is not a minor inconsistency: in the most comprehensive recent survey of the issue, Santos et al. [7] manually analysed 729 articles across 147 journals in 27 subject areas and found 11 different reference styles adopted within the Medicine subject area alone, with nearly half of all sampled journals following a customised "own" reference style rather than a shared standard. This flexibility is appropriate for human-orientated citation standards, where a journal's own conventions can matter more than universal machine parseability, but it introduces exactly the structural variability that machine processing struggles with. Kratochvíl et al. [8] also analysed 1,100 high-impact biomedical journals, and found that 70 (6.5%) use the current American Medical Association or NLM/Vancouver style; 425 (39.2%) use their older versions or a variation; 73 (6.7%) use a standard non-biomedical style; and 432 (39.9%) have their own house style.

Finally, constructing and managing conventional citations often requires reference-management tools such as Zotero, Mendeley, EndNote, or similar applications. Although these tools simplify bibliographic management, they can introduce an additional learning curve for users unfamiliar with academic referencing workflows [9, 10]. A study conducted by Ananda et al. [11] reported that 80% of participants found the Zotero training very helpful for managing references.

To our knowledge, no existing work has proposed a compacted citation format aimed specifically at input-efficiency for LLMs. We do not claim the underlying idea of a short, resolvable identifier is new on its own terms: the W3C's CURIE syntax has used a comparable prefix-plus-reference pattern for compacting URIs since 2007, resolved through a namespace registry rather than written out in full each time [12], and the same general principle underlies persistent identifier schemes used for biomedical data citation [13]. What we believe is new is applying this kind of compaction specifically to citation representation inside an LLM's input context, where the constraint is token cost rather than markup verbosity, and where a construct like Curly's parameter value needs to survive whitespace removal without an XML or RDF processing layer around it. Our work, Curly, attempts to address these challenges by its design, while prioritising maximum readability for machines with minimum input-token consumption and simplicity to construct a citation.

3 Curly Citation

3.1 Design Motivation

Curly is designed around five core components: curly braces, a source endpoint or value, a resolver parameter, a pair of backticks to preserve the url and a citation index. The core components remain the same for every structure and citation representation. The following subsections explain the motivation for retaining, removing, or introducing each component.

3.2 Curly Syntax

Curly provides two syntactic representations, Curly Prefix and Curly Shell, depending on how the citation is intended to be used. In Curly Shell representation, a citation representation remains fully in-text based the end of the line, without requiring a dedicated reference section. However, Curly Prefix is intended for use in a reference section, with the corresponding text line containing an anchor, }N, at the end, similar to a conventional citation. We have explained an additional design goal in Appendix B.

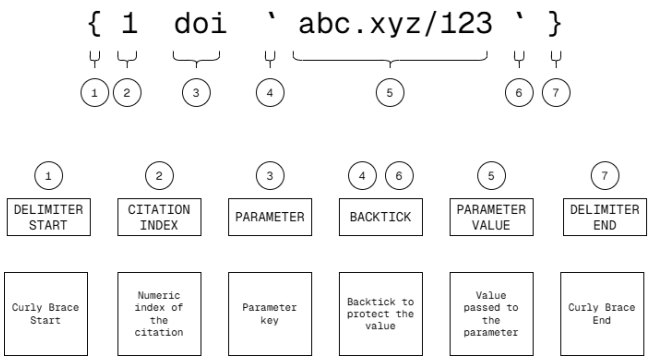


Fig. 1: Curly Shell Architecture

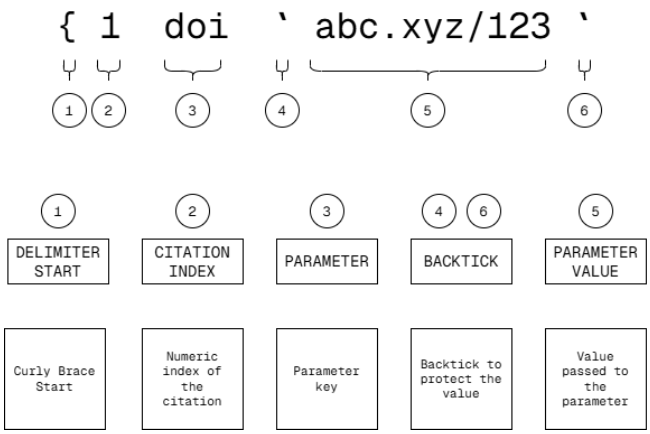


Fig. 2: Curly Prefix Architecture

3.2.1 Formal Grammar

The syntax of Curly Citation is formally defined using Extended Backus-Naur Form (EBNF). Curly provides two representations: Curly Prefix and Curly Shell. Both representations use the same citation components: an integer index, an alphabetic parameter, and a parameter value delimited by backticks. Curly Prefix additionally uses a citation anchor in the text and a corresponding resolver definition in the reference section. Figure 3 illustrates the formal grammar of Curly Citation.

3.2.2 Curly Braces

Curly braces are the primary structural element of Curly Citation. They indicate a machine where a Curly Citation is located.

For Curly Shell, curly brace starts and ends at the same line (Figure 1). In contrast, the Curly Prefix delimiter is not closed, as the anchored text contains the closed delimiter. Appendix C showcases a document level example for Curly Citation.

3.2.3 Index

The index is an integer value used to identify a specific citation. Each index must be unique within the document so that one citation cannot be confused with another. The citation index must be a positive integer and must not use any other value type. However, the index itself is not limited to any maximum integer. Larger integer values remain valid, although their token cost may vary depending on the tokeniser and increase the measured token count mentioned in Figure 4 and 5.

3.2.4 Parameter

In Section 2, we have discussed the resolver and how it increases input-token consumption. To address this problem, we introduced Parameter. A parameter is a short identifier representing the base URL or resolver where the citation lives and/or machine can retrieve data. Parameters should not be case-sensitive, but strictly limited to alphabetic characters. Parameters are limited to predefined and recognised resources. Inventing a parameter does not create a valid resolver or establish a valid base URL. We have briefly discussed the proposed parameter registry in Section 3.3.

3.2.5 Backtick

Backticks are used to separate the Parameter Value from the surrounding Curly syntax. This prevents the value from being interpreted as part of the citation structure and makes the boundary between the endpoint and the Curly components explicit.

3.2.6 Parameter Value

The parameter value identifies the location of the cited resource after the parameter has established the appropriate resolver. Among all the components, Parameter value is the most sensitive and important syntax. For example, `doi` parameter actually expects the suffix and prefix of a DOI. The relationship between Parameter and Parameter Value is supplemental, where both are dependent on each other and misspelling either one may invalidate the citation. Appendix A explains how they both work together to resolve a URL.

3.3 Parameter Registry

A parameter is not simply an abbreviation for a URL. It represents a predefined resolver that determines how the parameter value should be interpreted. Tables 1 and 2 showcase some default parameters, their resolvers, and example values.

The `url` parameter acts as a general fallback for external resources that do not have a dedicated Curly parameter. The value does not require the `https://` scheme. Curly assumes HTTPS when resolving this parameter.

4 Benchmark

We conducted multiple benchmarks to determine whether Curly can reduce the token consumption associated with citation representation and to examine whether the observed reduction remains consistent across different tokenisation schemes. The evaluation was performed at both the citation level and the document level, allowing the behaviour of Curly to be examined under controlled and cumulative citation workloads.

Appendix G lists all the models and/or tokenisers, and Appendix H notes the setup used to conduct the benchmark.

4.1 Curly's Token Consumption Across Tokenisers and Models

4.1.1 Introduction

A token consumption test was conducted to analyse whether the reduction of Curly came from a disguise rather than design. We selected 24 tokeniser/model configurations to observe whether the reduction for Curly resulted from the representation itself rather than tokeniser-specific behaviour.

4.1.2 Data

An actual DOI (`https://doi.org/10.31224/6477`) was used to generate the Curly citation (both prefix and shell) with the citation index set to 1.

4.1.3 Results

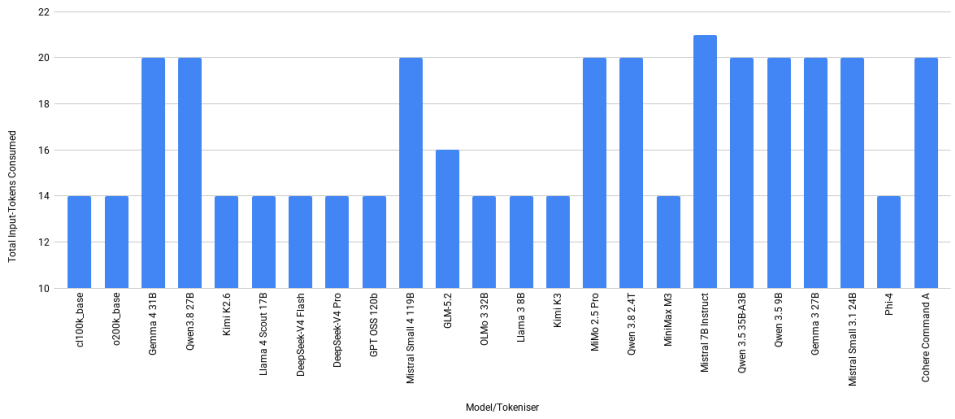


Fig. 4: Curly Prefix's Token Consumption Across Selected Tokenisers and/or Models

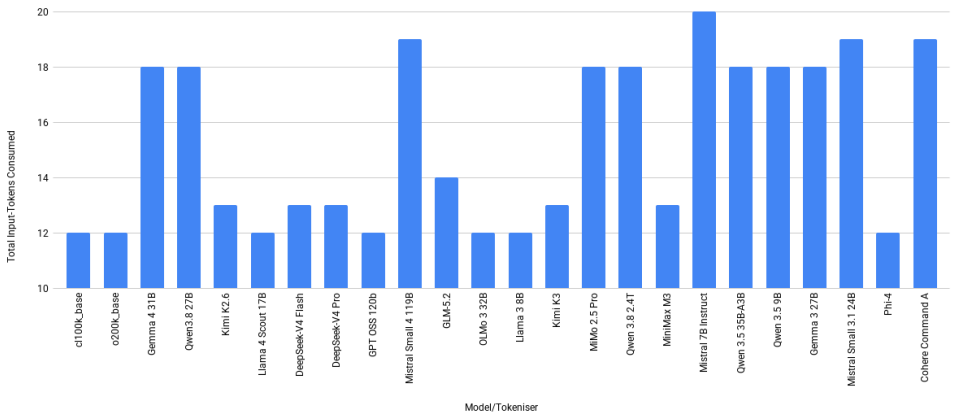


Fig. 5: Curly Shell's Token Consumption Across Selected Tokenisers and/or Models

4.1.4 Discussion

Figure 4 and 5 visualises the truncated result. Appendix F reveals the full result. The Curly Prefix architecture required 14—21 input tokens for all selected tokenisers and models to represent a citation. The highest observed cost occurred for Mistral 7B Instruct v0.3 (21 tokens) and the minimum occurred for Kimi K3 and several other configurations. A Curly Shell architecture required 12—20 tokens across the selected encodings, varied across models, with the highest value observed for the same model where Curly Prefix struggled. Overall, Curly Shell saves a small number of tokens but is less consistent than Curly Prefix.

4.2 Citation Formats' Token Consumption Across Tokenisers and Models

4.2.1 Introduction

To determine which citation formats consume more tokens, we conducted an experiment in which IEEE, Harvard, APA, MLA were present and additionally Curly was added. For this test, we also used the same models and tokenisers.

4.2.2 Data

An actual DOI (`https://doi.org/10.31224/6477`) was used to generate the Curly citation (both prefix and shell) and other formats like IEEE, Harvard, APA, etc., with the citation index set to 1. The tested files are archived at https://github.com/w3nabil/curly-citation/tree/main/dataset/single.

4.2.3 Calculation

The token cost of citation representation depends on both in-text and reference appearance. Calculating only the reference section data may not showcase the actual cost. Even a small in-text citation, such as `[1]`, consumes tokens. Upon discussion, we can say:

$$\text{Total Consumption} = \text{Consumption}_{\text{In-text}} + \text{Consumption}_{\text{Bibliography}} \tag{1}$$

where,

$\text{Consumption}_{\text{In-text}}$: the token cost measured when a citation appears in the main text

$\text{Consumption}_{\text{Bibliography}}$: token cost of citation metadata appearing in the reference section

For multiple citations, the cost should be calculated separately as the length of a citation does not remain same.

4.2.4 Results

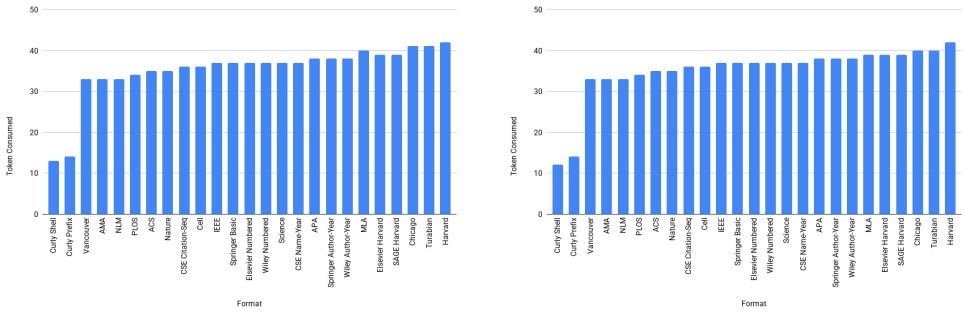


Fig. 6: Comparison of citation token consumption: `Deepseek V4 Flash 0781` (left) and `GPT OSS` (right)

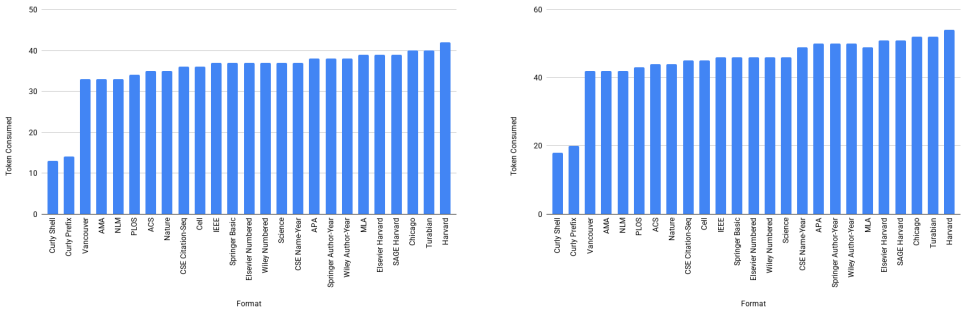


Fig. 7: Comparison of citation token consumption: Kimi K3 (left) and Qwen 3.8 (right)

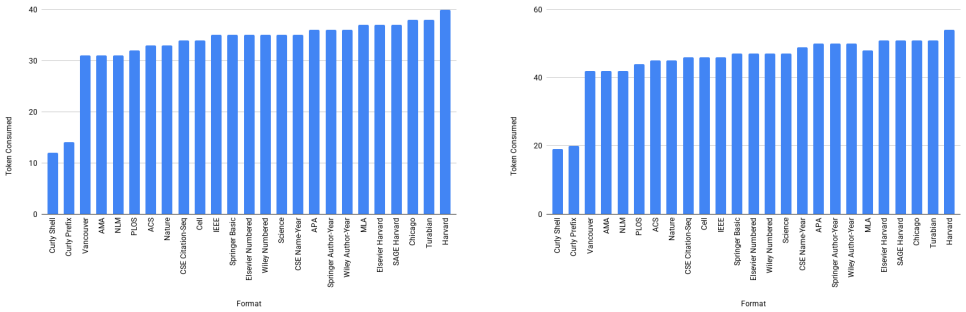


Fig. 8: Comparison of citation token consumption across selected model-native tokenisers: Llama 4 Scout (left) and `Cohere Command A` (right)

4.2.5 Discussion

Figure 8 illustrates the truncated result. Full benchmark is available in Appendix F. Results remained the same without any fluctuations. Curly Prefix has reduced the representation input-token cost from 50% to 68.18%. Curly Shell, however, has reduced 54.76% to 72.73% input-token cost. Results may vary depending on the parameter and value.

4.3 Citation Overhead at Document Scale

4.3.1 Introduction

To evaluate the cumulative effect of citation representation at document scale, we conducted a large benchmark using a document containing 150 citations. The same

document was evaluated in three configurations: the original document containing IEEE citations, the same document with citations represented using Curly Prefix, and a citation-free version used as a baseline for measuring the underlying token consumption of the document itself.

4.3.2 Data

First, we downloaded a technical report. Then using PyMuPDF4LLM Python library, we converted the PDF-formatted paper to Markdown. The paper originally followed the IEEE format, we replaced the IEEE format with Curly Prefix and completely removed the citations to create the NoCitation baseline. After the process, we had three markdown files with the citation representation variation but the paper context remained unchanged. The files are archived at https://github.com/w3nabil/curly-citation/tree/main/dataset/large.

4.3.3 Calculation

Citation-related token overhead can be estimated by comparing the token count of a document with a given citation format against the corresponding document with citations removed. In other words,

$$O_F = T_F - T_{NC} \tag{2}$$

where, O_F denotes the token overhead introduced by the citation format F, T_F denotes the total number of tokens using citation format F, and T_{NC} denotes the total number of tokens in the NoCitation version.
For the citation formats evaluated in this study, Equation 2 becomes,

$$O_{\text{IEEE}} = T_{\text{IEEE}} - T_{NC}, \qquad O_{\text{CURLY}} = T_{\text{CURLY}} - T_{NC} \tag{3}$$

To measure the reduction in citation-related overhead when transitioning from one citation format to another, the following metric is defined:

$$R_{CO_{A \to B}} = \frac{O_A - O_B}{O_A} \times 100, \qquad O_A > O_B \tag{4}$$

where,
$R_{CO_{A \to B}}$ denotes the percentage of citation overhead removed when transitioning from citation format A to citation format B.

For the transition from IEEE to Curly, this becomes,

$$R_{CO_{\text{IEEE} \to \text{CURLY}}} = \frac{O_{\text{IEEE}} - O_{\text{CURLY}}}{O_{\text{IEEE}}} \times 100 \tag{5}$$

To quantify the contribution of citation overhead to the original document, the following metric is defined relative to the original IEEE-formatted document:

$$R_{TT_F} = \frac{O_F}{T_{\text{IEEE}}} \times 100. \tag{6}$$

Here, R_{TT_F} denotes the percentage of the original IEEE document's total tokens attributable to the citation overhead associated with format F. The IEEE-formatted document is used as the fixed baseline because it represents the original paper content.

For the IEEE citation format,

$$R_{TT_{\text{NC}}} = \frac{O_{\text{IEEE}}}{T_{\text{IEEE}}} \times 100 \tag{7}$$

For CURLY, however, the relevant quantity is the total reduction in document tokens relative to the original IEEE version. Therefore,

$$R_{TT_{\text{CURLY}}} = \frac{T_{\text{IEEE}} - T_{\text{CURLY}}}{T_{\text{IEEE}}} \times 100. \tag{8}$$

Thus, $R_{TT_{\text{NC}}}$ represents the proportion of the original IEEE document occupied by citation overhead, whereas $R_{TT_{\text{CURLY}}}$ represents the proportion of the original document's tokens eliminated by replacing IEEE citations with Curly

4.3.4 Result

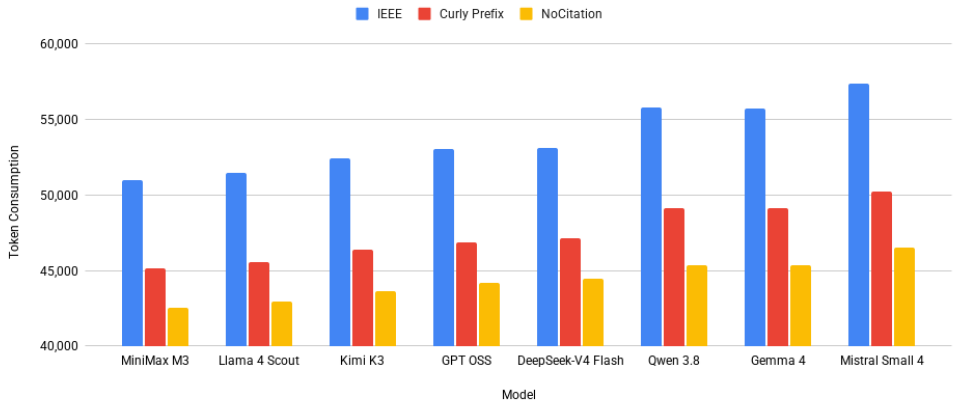


Fig. 9: Token consumption and citation overhead for a 150-citation document under IEEE, Curly Prefix, and no-citation representations.

4.3.5 Discussion

Figure 9 and Table 3 visualises that Curly Prefix consistently reduces total token consumption compared to the IEEE representation across all evaluated tokenisers and/or models. The IEEE representation document required between 51,006 and tokens 63,282, whereas the Curly Prefix representation required between 45,139 and 55,760 tokens as input token. Using the NoCitation document as the baseline, IEEE citations introduced between 8,495 and 11,809 additional tokens, whereas Curly Prefix introduced between 2,568 and 4,287 additional tokens. Consequently, Curly Prefix reduces the citation-related token consumption by approximately 70.10%—61.54%.

The test was extended to observe if Curly Prefix can optimise the document without necessarily reworking the text. Using the IEEE representation as baseline (As the document was written in IEEE format), Curly Prefix reduced total document token consumption by 11.33%—12.37%, where not using any citation format compressed the paper by 16.38%—18.84%. This provides a stronger indication of the practical effect of Curly in citation-heavy documents.

5 Limitations

Although the benchmark results indicate that Curly can substantially reduce citation-related token overhead, several limitations should be acknowledged.

First, Curly relies on a predefined parameter registry. Extending the registry requires adding and maintaining additional parameters and their corresponding resolvers. Parameters that are not registered must currently fall back to the general `url` parameter until a dedicated resolver is introduced.

Second, As of 24 August 2026, Curly has not been adopted as an official citation representation or native citation system by the evaluated language models. Consequently, the potential benefits of Curly may depend on how a model interprets, generates, and handles the proposed syntax. Model-specific training, prompting, or tool support could therefore positively affect the practical behaviour of the format.

Third, Curly does not itself guarantee citation correctness. The format provides a compact representation and resolution mechanism, but the generation, verification, and cross-validation of citations remain dependent on the document-processing pipeline or the system consuming the format.

Fourth, Curly is designed for accessible URLs or source locations and encodes the relevant URL or location of the source file and provides LLM-processing systems with an explicit source path. When citing a book, the mechanism may work differently because an LLM may not be able to resolve the source through Curly alone. Even though a parameter `book` can be introduced in the future, it does not guarantee the same accuracy as when providing the `doi`, `arxiv`, etc. parameters. When citing books with Curly, the `own` parameter requires with a file to be uploaded alongside the document. This represents a common limitation: Curly is less suitable for in-book citations.

Finally, the present evaluation focuses primarily on token consumption. The measured reduction should therefore not be interpreted as a direct or proportional reduction in inference latency, computational cost, or API expenditure. Such effects require separate evaluation under model- and provider-specific conditions.

6 Future Work

Future development will focus on extending Markdown compatibility so that Curly citation representation can be used. While this study helps LLMs reduce surface level citation input-token cost, it does not directly reduce the computational cost of internal LLM processing which may remain largely unchanged. We are developing a structure that would allow LLMs to use Curly directly for internal purposes. Some preliminary ideas are discussed in Appendix D, but future versions will investigate this direction in greater detail.

The parameter registry will also be expanded as additional source types and resolution mechanisms are identified and validated. Future work may also investigate automatic parameter discovery, resolver validation, citation integrity checking, and native integration with LLM-based document processing systems.

A further direction is the evaluation of Curly across larger and more diverse document collections. This would allow the token savings observed in the present citation-level and 150-citation benchmarks to be evaluated across different citation densities, document domains, source types, and tokeniser families.

7 Conclusion

Curly proposes a compact representation for citations designed to reduce the token overhead associated with conventional citation and URL representations in LLM-oriented document processing. By separating the citation index, resolver parameter, and parameter value, Curly preserves the information required to identify and resolve a source within a fixed structure while reducing the amount of repeated structural information contained in conventional citation representations.

The benchmark results indicate that the observed reduction is not restricted to a single tokeniser configuration. In the document-scale experiment, Curly reduced citation-related input-token consumption and total document input-token consumption. However, the results should be interpreted within the scope of the evaluated documents and tokeniser implementations. Broader corpus-level evaluation, model-level experimentation, and native integration remain necessary to determine the practical impact of Curly in production LLM systems.

Acknowledgements. The author declares that this project did not receive any external funding. The author also declares that they have no known competing financial

interests or personal relationships that could have appeared to influence the work reported in this paper.

Declarations

- Funding: Not applicable.
- Conflict of interest/Competing interests: The author declares no competing interests.
- Ethics approval and consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Data availability: The datasets and scripts used in this study are archived at https://github.com/w3nabil/curly-citation.
- Materials availability: Not applicable.
- Code availability: The benchmarking scripts are available at https://github.com/w3nabil/curly-citation/tree/main/test.
- Author contribution: Not applicable (single-author work).

Appendix A How the Curly Parameter and Parameter Value Work Together

The Parameter and Parameter Value together encode the source URL, where they represent the fixed and variable components, respectively. This observation arose from examining websites that store papers or other content at structured URLs. If a website stores papers at a structured URL this means: the stable portion of the URL can be treated as a resolver prefix, where the resolver prefix remains the same but the resource identifier changes depending on the paper. Consider the following example. Suppose we want to identify a paper using a DOI, the paper's URL is `https://doi.org/10.1234/1234`. Now we want to see another paper, the paper's url is `https://doi.org/10.1234/1235`. Observe that a specific URL prefix remains constant across papers which is `https://doi.org/`. The only part that changes is the path after it. This is where the parameter is introduced, where the fixed URL prefix shared by the resources is shortened by a simple and memorisable word. And the part which changes too often but lives inside the main path is called the parameter value. Because the components are compacted in this way, they can also be expanded to reconstruct the resolver. This is the relationship between parameter and parameter value.

Appendix B Why the Curly Parameter Is Limited to Alphabetic Characters and the Citation Index to Integers

Curly was designed so that the absence of whitespace does not prevent the format from being correctly parsed. However, achieving this property requires careful consideration of how each component is represented. During the design process, we identified that each component should have a distinct lexical representation, allowing the format to remain interpretable even when whitespace is removed.

For example, a standard Curly representation is written as {`1 doi ‘10.1234/1234‘`, while its compacted form without spaces becomes {`1doi‘10.1234/1234‘`. In this representation, the citation index can be identified by locating numeric characters that appear outside the backtick-enclosed value. The alphabetic characters represent the parameter name, while the backticks delimit the parameter value and prevent its contents from affecting parameter detection or lookup.

This fixed-type representation is a core feature of Curly. By restricting citation indices to integers and parameters to alphabetic characters, the format remains structurally identifiable even after whitespace removal. As a result, the citation lookup process is not dependent on spacing and can reliably distinguish between the citation index, parameter name, and parameter value.

Appendix C Example of using Curly Citation at your text

Curly Prefix

Introduction

This study explores the input-token overhead for IEEE, APA, Harvard }1. A study conducted by Nabil confirms that citations overhead the input-token by 50% }2. Various other studies confirm that the citation is needed to improve the output quality }3 }4.

References

{1 doi ‘10.1234/1234‘

{2 arxiv ‘2609.1234‘

{3 zenodo ‘12345678‘

{4 own ‘./dataset/test.csv‘

Curly Shell

Introduction

This study explores the input-token overhead for IEEE, APA, Harvard {1 doi '10.1234/1234'}. A study conducted by Nabil confirms that citations overhead the input-token by 50% {2 arxiv '2609.1234'}. Various other studies confirm that the citation is needed to improve the output quality {3 zenodo '12345678'} {4 own './dataset/test.csv'}.

Appendix D Curly4LLM

Curly is heavily optimised for visual end. However, this appendix showcases a possible future version of Curly which works with and processes citations internally within LLM systems. Our preliminary proposal is to represent a Curly-formatted paper (a paper using Curly Citation) in the following format: {`1 doi '10.1234/1234' This text is cited`}. This would allow an LLM to render the cited text in html, markdown or any other format. This mechanism is not developed in the present paper, which focuses solely on citation representation cost. If an LLM stores cited text in this format, it could also convert any visual paper to LLM-internal format which may reduce output and computational token consumption while potentially improving citation or retrieval accuracy (This remains a hypothesis and has not yet been empirically validated).

Appendix E Observing Whether an LLM Can Learn a Paper's Abstract from Conventional Citation Metadata Alone

This appendix supports the claim made in Section 2 that bibliographic metadata included in conventional citation formats (which is out of model's current training data) is not, by itself, sufficient for an LLM to reconstruct or report the content of a cited source. To test this, we presented Sonnet 5 with a passage containing a conventional citation for a specific paper, without granting the model access to any tool that could fetch or retrieve external content, and asked it to report the abstract of the cited work.

The exchange proceeded in two stages.

Stage 1: Initial request, no fetch access and pressed to guess or use existing knowledge. The model was given the citation (author, title, and publication metadata as would appear in an IEEE-style reference) and asked to state the abstract of the cited paper. The model hesitated and did not produce an abstract, despite the conventional citation metadata being fully present in the prompt.

Prompt:

```
Here is a citation:

DeepSeek-AI et al., 'DeepSeek-V4: Towards Highly Efficient Million-Token
    Context Intelligence', arXiv [cs.CL]. 2026. [Online]. Available:
    https://arxiv.org/abs/2606.19348
```

```
Your Task: Briefly explain what's in the abstract without using any tools
    to access the internet. Keep the answer short and compacted. Avoid
    using chatbot artefacts while answering. A short 1-line explanation
    of why the title, author data, and journal did not assist you is
    welcomed.
```

Response:

```
I don't have reliable knowledge of this paper -- my training cutoff is
    January 2026, and a 2026 arXiv paper on "DeepSeek-V4" with
    million-token context could easily fall right around or after that
    boundary, so I can't summarise its abstract from memory without
    risking a fabricated answer. Why the citation details don't help: the
    title tells me the general topic (efficient long-context modeling,
    likely scaling to ~1M tokens), the author block just confirms it's a
    DeepSeek-AI team paper, and "arXiv [cs.CL], 2026" only narrows it to
    a preprint in computational linguistics from this year -- none of
    that substitutes for the actual abstract text, which I haven't seen.
```

Stage 2: Fetch access granted. The model was then given permission to use a web-fetch tool. With this access, the model retrieved information from both the resolved URL and the conventional citation metadata, and was able to report the abstract.

Prompt:

```
Here is a citation:

https://arxiv.org/abs/2606.19348

Your Task: Briefly explain what's in the abstract using any tools that
    may assist you. Keep the answer short and compacted. Avoid using
    chatbot artefacts while answering. A short 1-line explanation of why
    the title, author data, and journal did not assist you is welcomed.
```

Response:

```
The paper introduces DeepSeek-V4 as a preview release with two MoE models
    -- V4-Pro (1.6T total / 49B active parameters) and V4-Flash (284B
    total / 13B active), both supporting one-million-token context. Key
    upgrades: a hybrid attention design mixing Compressed Sparse
    Attention and Heavily Compressed Attention, a new residual-connection
    variant called Manifold-Constrained Hyper-Connections, and the Muon
    optimizer for training stability. Trained on 32T+ tokens with
    post-training refinement. Its top reasoning mode, V4-Pro-Max, claims
    state-of-the-art results among open models, and at 1M-token context
    V4-Pro cuts inference FLOPs to 27% and KV cache to 10% versus
    DeepSeek-V3.2. Why the citation alone didn't help: the title only
    signals the general theme (efficiency, long context) and the author
    list just confirms a large DeepSeek-AI team -- neither reveals the
    specific architecture (CSA/HCA, mHC, Muon) or the concrete efficiency
    numbers that only the fetched abstract provides.
```

Observation. Across the two stages, the model did not treat the conventional bibliographic metadata (for data outside the pre-trained time schedule) as sufficient grounds to report the paper's content, even though a human reader could, in principle, use that same metadata to search for and locate the abstract manually. The model only produced the abstract once given the ability to resolve the source directly. This is consistent with the position taken in Section 2: bibliographic fields that support human-orientated identification do not necessarily give an LLM the information it needs to learn about a cited work, and a resolvable endpoint appears to matter more than descriptive metadata for this purpose.

Caveats. This observation reflects a single exchange with a single model and has not been repeated across multiple models, prompts, or paper types, and it has not been independently reproduced. It is reported here as a motivating illustration for the design rationale in Section 2, not as a controlled experiment, and no quantitative claim in this paper depends on it.

Appendix F Token counts of citation formats across LLM Tokenisers

Results are visualised in Table F.1, F.2, F.3 and F.4.

Appendix G Models and tokeniser Configurations Used in the Evaluation

The evaluation was conducted using 24 imported models and tokenisers, comprising 22 Hugging Face model configurations and two OpenAI tokeniser configurations:

Appendix H Setup for Benchmarking

H.1 Programming Language

1. `Python 3.14.6`

H.2 External Libraries

Library	Version	Reason for Importing
tiktoken	0.13.0	Provides access to the GPT-4 `cl100k_base` and GPT-4o `o200k_base` encodings
transformers	5.15.0	Provides tokeniser implementations for various Hugging Face models
pymupdf4llm	1.28.2	Convert PDF documents to Markdown

The script used for the benchmark is available at https://github.com/w3nabil/curly-citation/tree/main/test

References

[1] Gao, T., Yen, H., Yu, J., Chen, D.: Enabling large language models to generate text with citations. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 6465–6488. Association for Computational Linguistics, Singapore (2023). https://doi.org/10.18653/v1/2023.emnlp-main.398 . https://aclanthology.org/2023.emnlp-main.398/

[2] Zeng, J., Liu, H., Dai, Z., Tang, X., Chen, L., Varshney, S., Zhen, L., Qi, H.: Cite Before You Speak: Enhancing Context-Response Grounding in E-Commerce Conversational LLM-Agents (2025). https://arxiv.org/abs/2503.04830 Accessed 2026-08-13

[3] Asai, A., He, J., Shao, R., Shi, W., Singh, A., Chang, J., Lo, K., Soldaini, L., Feldman, S., D'Arcy, M., Wadden, D., Latzke, M., Sparks, J., Hwang, J.D., Kishore, V., Tian, M., Ji, P., Liu, S., Tong, H., Wu, B., Xiong, Y., Zettlemoyer, L.S., Neubig, G., Weld, D., Downey, D., Yih, W.-t., Koh, P.W., Hajishirzi, H.: Synthesizing scientific literature with retrieval-augmented language models. Nature **650**, 857–863 (2026) https://doi.org/10.1038/s41586-025-10072-4

[4] Chen, Y., Quan, Y., Lin, X., Tang, R.: Where Fake Citations Are Made: Tracing Field-Level Hallucination to Specific Neurons in LLMs (2026). https://arxiv.org/abs/2604.18880 Accessed 2026-08-13

[5] Apps, A.: Guidelines for Encoding Bibliographic Citation Information in Dublin Core™ Metadata (2005). https://www.dublincore.org/specifications/dublin-core/dc-citation-guidelines/ Accessed 2026-08-31

[6] Crossref: Creating a Landing Page. Crossref (2026). https://www.crossref.org/documentation/member-setup/creating-a-landing-page/ Accessed 2026-08-31

[7] Santos, E.A.d., Peroni, S., Mucheroni, M.L.: An analysis of citing and referencing habits across all scholarly disciplines: Approaches and trends in bibliographic referencing and citing practices. Journal of Documentation **79**(7), 196–224 (2023) https://doi.org/10.1108/jd-10-2022-0234

[8] Kratochvíl, J., Abrahámová, H., Fialová, M., Stodůlková, M.: Citation rules through the eyes of biomedical journal editors. Learned Publishing **35** (2021) https://doi.org/10.1002/leap.1425

[9] Zhang, Y.: Comparison of select reference management tools. Medical Reference Services Quarterly **31**(1), 45–60 (2012) https://doi.org/10.1080/02763869.2012.641841

[10] Ivey, C., Crum, J.: Choosing the right citation management tool: Endnote, mendeley, refworks, or zotero. Journal of the Medical Library Association **106**(3) (2018) https://doi.org/10.5195/jmla.2018.468 . Accessed 2026-08-17

[11] Ananda, Y.Y.T., Samsidar, S., Nasrudin, A.R., Sinaga, S.P.H., Ilham, Alamsyah, M.: Zotero's training to improve citation skills in the preparation of students' final project. Jurnal Inovasi Sains dan Teknologi untuk Masyarakat **1**(2), 84–91 (2023) https://doi.org/10.19184/instem.v1i2.478

[12] World Wide Web Consortium: CURIE Syntax 1.0. World Wide Web Consortium (2010). https://www.w3.org/TR/2010/NOTE-curie-20101216/ Accessed 2026-08-31

[13] Wimalaratne, S.M., Juty, N., Kunze, J., Janée, G., McMurry, J.A., Beard, N., Jimenez, R., Grethe, J.S., Hermjakob, H., Martone, M.E., Clark, T.: Uniform resolution of compact identifiers for biomedical data. Scientific Data **5**(1) (2018) https://doi.org/10.1038/sdata.2018.29

```
curly-citation = curly-prefix | curly-shell ;

curly-prefix = prefix-anchor | prefix-reference ;

curly-shell = "{", index, wsp, parameter, wsp, backtick, parameter-value,
    backtick, "}" ;

prefix-reference = "{", index, wsp, parameter, wsp, backtick,
    parameter-value, backtick ;

prefix-anchor = "}", index ;

index = positive-integer ;

positive-integer = nonzero-digit, { digit } ;

parameter = letter, { letter } ;

parameter-value = { value-character } ;

letter = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k"
    | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" |
    "w" | "x" | "y" | "z" ;

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;

nonzero-digit = "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;

wsp = " " | "\t" ;

backtick = "‘" ;

value-character = any-character-except-backtick ;
```

Fig. 3: Formal grammar of Curly Citation in Extended Backus–Naur Form.

Table 1: Curly parameter registry.

Parameter	Meaning	Value Type
`own`	User Uploaded	`URI/FILENAME.EXT`
`doi`	Digital Object Identifier	`DOI`
`pmc`	PubMed Central article	`PMCID`
`pmid`	PubMed record	`PMID`
`iacr`	Cryptology ePrint Archive paper	`PAPER_ID`
`arxiv`	arXiv paper	`PAPER_ID`
`zenodo`	Zenodo record	`RECORD_ID`
`url`	External URL	`DOMAIN/ENDPOINT`

Table 2: Curly parameter resolvers and example endpoints.

Parameter	Resolves To	Example Value
`own`	./	`dataset.csv`
`doi`	https://doi.org/	`10.31224/6477`
`pmc`	https://pmc.ncbi.nlm.nih.gov/articles/	`PMC1234567`
`pmid`	https://pubmed.ncbi.nlm.nih.gov/	`12345678`
`iacr`	https://eprint.iacr.org/	`2026/1234`
`arxiv`	https://arxiv.org/html/	`1234.5678`
`zenodo`	https://zenodo.org/records/	`12345678`
`url`	https://	`example.com/paper/123`

Table 3: Token consumption and citation overhead and various data for a 150-citation document under IEEE, Curly Prefix, and NoCitation representations.

Tokeniser/Model	$\mathbf{T}_{\text{IEEE}}$	$\mathbf{T}_{\text{C}}$	$\mathbf{T}_{\text{NC}}$	$\mathbf{O}_{\text{IEEE}}$	$\mathbf{O}_{\text{C}}$	$\mathbf{R}_{\text{CO}_{\text{I}\to\text{C}}}$	$\mathbf{R}_{\text{TT}_{\text{NC}}}$	$\mathbf{R}_{\text{TT}_{\text{C}}}$
MiniMax M3	51,006	45,139	42,497	8,509	2,642	68.95%	16.68%	11.50%
Llama 4 Scout 17B	51,463	45,536	42,968	8,495	2,568	69.77%	16.51%	11.52%
Kimi K2.6	52,412	46,376	43,649	8,763	2,727	68.88%	16.72%	11.52%
Kimi K3	52,412	46,376	43,649	8,763	2,727	68.88%	16.72%	11.52%
o200k_base	53,049	46,855	44,193	8,856	2,662	69.94%	16.69%	11.68%
GPT OSS 120b	53,049	46,855	44,193	8,856	2,662	69.94%	16.69%	11.68%
Llama 3 8B	53,134	46,881	44,214	8,920	2,667	70.10%	16.79%	11.77%
cl100k_base	53,161	46,906	44,238	8,923	2,668	70.10%	16.78%	11.77%
OLMo 3 32B	53,161	46,906	44,238	8,923	2,668	70.10%	16.78%	11.77%
Phi-4	53,161	46,906	44,238	8,923	2,668	70.10%	16.78%	11.77%
GLM-5.2	53,292	46,986	44,238	9,054	2,748	69.65%	16.99%	11.83%
DeepSeek-V4 Flash	53,134	47,114	44,430	8,704	2,684	69.16%	16.38%	11.33%
DeepSeek-V4 Pro	53,134	47,114	44,430	8,704	2,684	69.16%	16.38%	11.33%
MiMo 2.5 Pro	55,810	48,952	45,314	10,496	3,638	65.34%	18.81%	12.29%
Qwen3.8 27B	55,824	49,132	45,368	10,456	3,764	64.00%	18.73%	11.99%
Qwen 3.8 2.4T	55,824	49,132	45,368	10,456	3,764	64.00%	18.73%	11.99%
Qwen 3.5 35B-A3B	55,824	49,132	45,368	10,456	3,764	64.00%	18.73%	11.99%
Qwen 3.5 9B	55,824	49,132	45,368	10,456	3,764	64.00%	18.73%	11.99%
Gemma 3 27B	55,694	49,141	45,329	10,365	3,812	63.22%	18.61%	11.77%
Cohere Command A	56,035	49,556	45,507	10,528	4,049	61.54%	18.79%	11.56%
Gemma 4 31B	56,568	50,011	46,175	10,393	3,836	63.09%	18.37%	11.59%
Mistral Small 4	57,351	50,258	46,546	10,805	3,712	65.65%	18.84%	12.37%
Mistral Small 3.1	57,351	50,258	46,546	10,805	3,712	65.65%	18.84%	12.37%
Mistral 7B Instruct	63,282	55,760	51,473	11,809	4,287	63.70%	18.66%	11.89%

$\mathbf{R}_{\text{TT}_{\text{C}}} = \mathbf{R}_{\text{TT}_{\text{CURLY}}}$, $\mathbf{R}_{\text{CO}_{\text{I}\to\text{C}}} = \mathbf{R}_{\text{CO}_{\text{I}\to\text{CURLY}}}$, $\mathbf{O}_{\text{C}} = \mathbf{O}_{\text{CURLY}}$ & $\mathbf{T}_{\text{C}} = \mathbf{T}_{\text{CURLY}}$
Different models may employ the same or a similar tokeniser.

Table F.1: Token counts of citation formats across models and tokenisers (1–6).

Citation Format	cl100k base	o200k base	Gemma 4 31B	Qwen3.8 27B	Kimi K2.6	Llama 4 Scout 17B
Curly Shell	12	12	18	18	13	12
Curly Prefix	14	14	20	20	14	14
Vancouver	34	33	40	42	33	31
AMA	34	33	40	42	33	31
NLM	34	33	40	42	33	31
PLOS	35	34	42	43	34	32
ACS	36	35	43	44	35	33
Nature	36	35	43	44	35	33
CSE Citation-Seq	37	36	44	45	36	34
Cell	37	36	44	45	36	34
IEEE	38	37	45	46	37	35
Springer Basic	38	37	45	46	37	35
Elsevier Numbered	38	37	45	46	37	35
Wiley Numbered	38	37	45	46	37	35
Science	38	37	45	46	37	35
CSE Name-Year	39	37	47	49	37	35
APA	40	38	48	50	38	36
Springer Author-Year	40	38	48	50	38	36
Wiley Author-Year	40	38	48	50	38	36
MLA	41	39	48	49	39	37
Elsevier Harvard	41	39	49	51	39	37
SAGE Harvard	41	39	49	51	39	37
Chicago	42	40	51	52	40	38
Turabian	42	40	51	52	40	38
Harvard	44	42	52	54	42	40

Note: Values represent the number of tokens required to encode each citation format under the corresponding model or tokeniser.

Table F.2: Token counts of citation formats across models and tokenisers (7–12).

Citation Format	DeepSeek-V4 Flash 0731	DeepSeek-V4 Pro 0813	GPT OSS 120B	Mistral Small 4 119B	GLM 5.2	OLMo 3 32B
Curly Shell	13	13	12	19	14	12
Curly Prefix	14	14	14	20	16	14
Vancouver	33	33	33	42	36	34
AMA	33	33	33	42	36	34
NLM	33	33	33	42	36	34
PLOS	34	34	34	43	37	35
ACS	35	35	35	44	38	36
Nature	35	35	35	44	38	36
CSE Citation-Seq	36	36	36	45	39	37
Cell	36	36	36	45	39	37
IEEE	37	37	37	46	40	38
Springer Basic	37	37	37	46	40	38
Elsevier Numbered	37	37	37	46	40	38
Wiley Numbered	37	37	37	46	40	38
Science	37	37	37	46	40	38
CSE Name-Year	37	37	37	48	41	39
APA	38	38	38	49	42	40
Springer Author-Year	38	38	38	49	42	40
Wiley Author-Year	38	38	38	49	42	40
MLA	40	40	39	49	43	41
Elsevier Harvard	39	39	39	50	43	41
SAGE Harvard	39	39	39	50	43	41
Chicago	41	41	40	52	44	42
Turabian	41	41	40	52	44	42
Harvard	42	42	42	53	46	44

Note: Values represent the number of tokens required to encode each citation format under the corresponding model or tokeniser.

Table F.3: Token counts of citation formats across models and tokenisers (13–18).

Citation Format	Llama 3 8B	Kimi K3	MiMo 2.5 Pro	Qwen 3.8 2.4T	MiniMax M3	Mistral 7B Instruct v0.3
Curly Shell	12	13	18	18	13	20
Curly Prefix	14	14	20	20	14	21
Vancouver	34	33	42	42	32	46
AMA	34	33	42	42	32	46
NLM	34	33	42	42	32	46
PLOS	35	34	43	43	33	47
ACS	36	35	44	44	34	48
Nature	36	35	44	44	34	50
CSE Citation-Seq	37	36	45	45	35	50
Cell	37	36	45	45	35	50
IEEE	38	37	46	46	36	50
Springer Basic	38	37	46	46	36	51
Elsevier Numbered	38	37	46	46	36	51
Wiley Numbered	38	37	46	46	36	51
Science	38	37	46	46	36	51
CSE Name-Year	39	37	49	49	36	54
APA	40	38	50	50	37	55
Springer Author-Year	40	38	50	50	37	55
Wiley Author-Year	40	38	50	50	37	55
MLA	41	39	49	49	38	55
Elsevier Harvard	41	39	51	51	38	56
SAGE Harvard	41	39	51	51	38	56
Chicago	42	40	52	52	39	58
Turabian	42	40	52	52	39	58
Harvard	44	42	54	54	41	59

Note: Values represent the number of tokens required to encode each citation format under the corresponding model or tokeniser.

Table F.4: Token counts of citation formats across models and tokenisers (19–24).

Citation Format	Qwen 3.5 35B-A3B	Qwen 3.5 9B	Gemma 3 27B	Mistral Small 3.1 24B	Phi-4	Cohere Command A
Curly Shell	18	18	18	19	12	19
Curly Prefix	20	20	20	20	14	20
Vancouver	42	42	40	42	34	42
AMA	42	42	40	42	34	42
NLM	42	42	40	42	34	42
PLOS	43	43	42	43	35	44
ACS	44	44	43	44	36	45
Nature	44	44	43	44	36	45
CSE Citation-Seq	45	45	44	45	37	46
Cell	45	45	44	45	37	46
IEEE	46	46	45	46	38	46
Springer Basic	46	46	45	46	38	47
Elsevier Numbered	46	46	45	46	38	47
Wiley Numbered	46	46	45	46	38	47
Science	46	46	45	46	38	47
CSE Name-Year	49	49	47	48	39	49
APA	50	50	48	49	40	50
Springer Author-Year	50	50	48	49	40	50
Wiley Author-Year	50	50	48	49	40	50
MLA	49	49	48	49	41	48
Elsevier Harvard	51	51	49	50	41	51
SAGE Harvard	51	51	49	50	41	51
Chicago	52	52	51	52	42	51
Turabian	52	52	51	52	42	51
Harvard	54	54	52	53	44	54

Note: Values represent the number of tokens required to encode each citation format under the corresponding model or tokeniser.

Table G.5: Models and tokenisers used in the evaluation.

Source	**Models/tokenisers**
Hugging Face	MiniMax M3 Llama 4 Scout 17B Kimi K2.6 Kimi K3 GPT OSS 120B Llama 3 8B OLMo 3 32B Phi-4 GLM-5.2 DeepSeek-V4 Flash 0731 DeepSeek-V4 Pro 0813 MiMo 2.5 Pro Qwen3.8 27B Qwen 3.8 2.4T Qwen 3.5 35B-A3B Qwen 3.5 9B Gemma 3 27B Cohere Command A Gemma 4 31B Mistral Small 4 119B Mistral Small 3.1 24B Mistral 7B Instruct v0.3
OpenAI	`o200k_base` `cl100k_base`