SelfCite: Self-Supervised Alignment for Context Attribution in Large Language Models

Yung-Sung Chuang [1] **Benjamin Cohen-Wang** [1] **Shannon Zejiang Shen** [1] **Zhaofeng Wu** [1] **Hu Xu** [2]
Xi Victoria Lin [2] **James Glass** [1] **Shang-Wen Li** [2] **Wen-tau Yih** [2]

Abstract

We introduce SelfCite, a novel self-supervised approach that aligns LLMs to generate high-quality, fine-grained, sentence-level citations for the statements in their generated responses. Instead of only relying on costly and labor-intensive annotations, SelfCite leverages a reward signal provided by the LLM itself through *context ablation*: If a citation is necessary, removing the cited text from the context should prevent the same response; if sufficient, retaining the cited text alone should preserve the same response. This reward can guide the inference-time best-of-N sampling strategy to improve citation quality significantly, as well as be used in preference optimization to directly fine-tune the models for generating better citations. The effectiveness of SelfCite is demonstrated by increasing citation F1 up to 5.3 points on the LongBench-Cite benchmark across five long-form question answering tasks. The source code is available at https://github.com/facebookresearch/SelfCite.

1. Introduction

Assistants built using large language models (LLMs) have become ubiquitous in helping users gather information and acquire knowledge (OpenAI, 2022; 2023). For instance, when asked about recent news, an assistant can read through dozens of relevant articles—potentially more than a user could comb through themselves—and use these articles as *context* to provide a clear, specific answer to the user's query. While this ability can greatly accelerate information gathering, LLMs often produce hallucinations—content that sounds plausible but is actually fabricated (Ji et al., 2023). Even when provided with accurate context, models may misinterpret the data or include details that are not supported by the context (Shi et al., 2024; Chuang et al., 2024).

Although completely eliminating hallucinations remains difficult, existing approaches have sought to enhance the reliability of LLMs by providing context attributions–commonly referred to as *citations*–which are fine-grained references to relevant evidences from the context, alongside generated responses for user verification (Menick et al., 2022; Slobodkin et al., 2024; Zhang et al., 2024). While they have shown promise in generating citations, an outstanding challenge is their reliance on annotated data either from human (Menick et al., 2022; Slobodkin et al., 2024) or costly proprietary APIs (Zhang et al., 2024) to train models to generate citations. Collecting annotations can be time-consuming or costly, especially with long-context documents.

To address this challenge, we introduce SelfCite, a novel alignment approach designed to autonomously enhance the quality of citations generated by LLMs without the need for any annotations in the alignment process. Drawing inspiration from model interpretability techniques (Lei et al., 2016; Cohen-Wang et al., 2024), SelfCite leverages the inherent capabilities of LLMs to provide feedback through *context ablation*—a process to evaluate the necessity and sufficiency of a citation. If removing the cited text prevents the LLM from assigning high probability to the same response, we can infer that it is *necessary* for the LLM. Conversely, if the response remains highly probable despite removing all context other than the cited text, this indicates that the citation is *sufficient* for the LLM to make the claim. This self-evaluation mechanism enables SelfCite to calculate a reward signal without relying on the annotation processes.

Building on this intuition, we design a reward that can be cheaply computed by the LLM itself, composed by *probability drop* and *probability hold* in context ablation. By integrating this reward function into a best-of-N sampling strategy, SelfCite achieves substantial improvements in citation quality. Furthermore, we employ this reward for preference optimization using SimPO (Meng et al., 2024), which not only maintains these improvements but also eliminates the need for the computationally expensive best-of-N

[1]Massachusetts Institute of Technology, Cambridge, MA 02139, USA [2]Meta FAIR, USA. Correspondence to: Yung-Sung Chuang <yungsung@mit.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

sampling. We outperform the previous state of the art on the LongBench-Cite benchmark (Zhang et al., 2024) by up to 5.3 points in F1 scores, and showing a promising direction to bootstrap the citation quality from LLMs via self-rewarding.

2. Method

In this section, we describe the SelfCite framework. We begin by introducing the task of generating responses with context attributions (2.1), referred to as *citations* for brevity. We then design a reward for providing feedback on citation quality *without* human annotations (2.2) as illustrated in Fig. 1. Finally, we discuss two approaches for utilizing this reward to improve citation quality: best-of-N sampling (2.3) and preference optimization (2.4).

2.1. Problem Formulation

We first formalize the task of generating responses with context attributions and the metrics to self-evaluate context attributions within the SelfCite framework, inspired by previous papers (Zhang et al., 2024; Cohen-Wang et al., 2024) but adapted to our proposed self-supervised reward.

Setup. Consider employing an autoregressive language model (LM) to generate a response to a specific query given a context of relevant information. Specifically, given an LM p_{LM}, let $p_{\text{LM}}(t_i \mid t_1, \ldots, t_{i-1})$ denote its output distribution over the next token t_i based on a sequence of preceding tokens $t_1, \ldots, t_{i-1}$. Next, let C represent the context of relevant information. This context is partitioned into $|C|$ sentences: $c_1, c_2, \ldots, c_{|C|}$. Each sentence c_j is prepended with a unique identifier (e.g., sentence index j) as a way for the model to reference the sentence when generating citations. The context C is followed by a query Q, a question or instruction for the model. A response R is then sampled from the model p_{LM}.

Generating Responses with Context Attributions. In SelfCite, following prior work on generating responses with context attributions (Zhang et al., 2024), each statement r_i in the response R is followed by a citation sequence e_i consisting of the identifiers of sentences from the context C. Thus, the entire response sequence R is $\{r_1, e_1, r_2, e_2, \ldots, r_S, e_S\}$, where S is the total number of generated statements. The citation e_i is intended to reference sentences that support the generation of r_i. Formally, for each response statement r_i, the model outputs a citation sequence $e_i = \{e_i^1, e_i^2, \ldots, e_i^m\}$, where each $e_i^j \in \{1, 2, \ldots, |C|\}$ corresponds to a specific sentence number in the context C, and m sentences are cited. Note that this citation sequence may be empty. The entire response R consisting of statements r_i followed by citations e_i is sampled from the LM p_{LM} as follows:

$$r_i \sim p_{\text{LM}}\left(\cdot \mid c_1, \ldots, c_{|C|}, Q, r_1, e_1, \ldots, r_{i-1}, e_{i-1}\right),$$
$$e_i \sim p_{\text{LM}}\left(\cdot \mid c_1, \ldots, c_{|C|}, Q, r_1, e_1, \ldots, r_{i-1}, e_{i-1}, r_i\right).$$

The objective of optimizing the LM is to ensure that the citation sequence e_i accurately reflects the evidence from the context that supports the generation of r_i. In the SFT setting (Zhang et al., 2024), the probability of a "ground truth" annotated responses and citations $\{\hat{r}_1, \hat{e}_1, ..., \hat{r}_S, \hat{e}_S\}$ will be maximized, given the input C and Q, but it is not trivial to do further alignment with feedback after the SFT data is used up. To achieve this, we introduce SelfCite that can evaluate the quality of these citations based on context ablation as a reward for further preference optimization.

2.2. Self-Supervised Reward via Context Ablation

We measure the quality of a citation sequence e_i by the *changes* in the LM's probability of generating r_i when the cited sentences are either removed from or isolated within the context. To simplify the notation, let all the cited context sentences be $E_i = \{c_{e_i^1}, c_{e_i^2}, \ldots, c_{e_i^m}\}$. We define two key metrics: *necessity score* and *sufficiency score*, and finally combine them into the final reward, as shown in Fig. 1.

Necessity Score: Probability Drop. This metric quantifies the decrease in the probability of generating r_i when the cited sentences in E_i are all removed from the context (denoted as set minus $\setminus$ operator). Formally, it is defined as:

$$\text{Prob-Drop}(e_i) = \log p_{\text{LM}}(r_i \mid C) - \log p_{\text{LM}}\left(r_i \mid C \setminus E_i\right).$$

To keep the equation concise, we ignore Q and $\{r_1, e_1, ..., r_{i-1}, e_{i-1}\}$ in the equation, but they are staying in the context history when computing the probabilities. A larger probability drop indicates that the removal of E_i significantly diminishes the likelihood of generating r_i, thereby validating the necessity of the cited evidence.

Sufficiency Score: Probability Hold. Conversely, this metric measures if the probability of generating r_i is still kept large when *only* the cited sentences are kept in the context, effectively testing the sufficiency of the citation to support the response statement. Formally:

$$\text{Prob-Hold}(e_i) = \log p_{\text{LM}}\left(r_i \mid E_i\right) - \log p_{\text{LM}}(r_i \mid C).$$

A more positive value of probability hold indicates that the cited sentences alone are sufficient to support the generation of r_i, while removing all the other irrelevant context. Please note that the values of probability drop or hold can be either positive or negative. For example, if the citation is not relevant to r_i or even distracting, it is possible for $p(r_i \mid E_i)$ to be lower than $p(r_i \mid C)$.

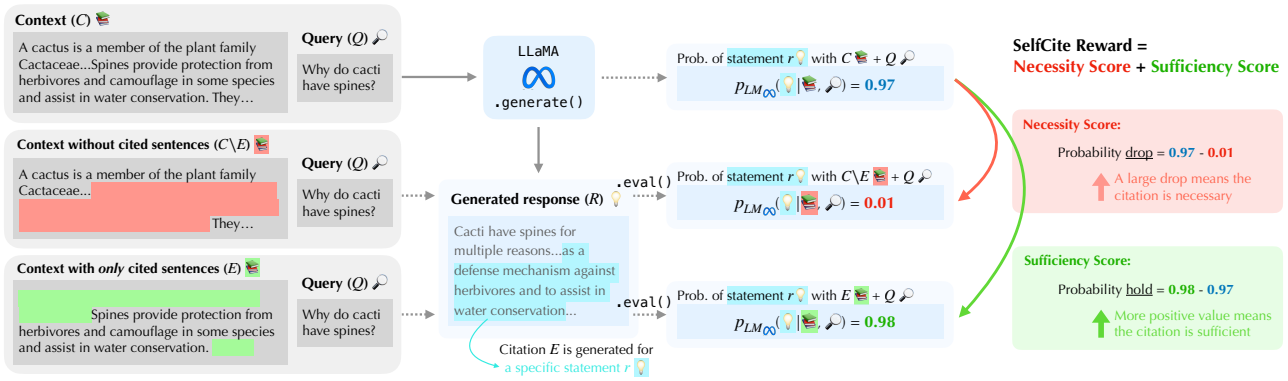

Figure 1. The SelfCite framework calculates rewards based on two metrics: *necessity score* (probability drop) and *sufficiency score* (probability hold). First, the full context is used to generate a response. Then, the framework evaluates the probability of generating the same response after (1) removing the cited sentences from the context and (2) using only the cited sentences in the context. The probability drop and hold are computed from these probability differences, and their sum is used as the final reward.

Final Reward. To comprehensively evaluate the necessity and sufficiency of the generated citations, we add the two metrics together, where the opposing terms cancel out:

$$\begin{aligned}\text{Reward}(e_i) &= \text{Prob-Drop}(e_i) + \text{Prob-Hold}(e_i) \\ &= \log p_{\text{LM}}(r_i|E_i) - \log p_{\text{LM}}(r_i|C \setminus E_i). \quad (1)\end{aligned}$$

The combined reward measures if the citations are both necessary and sufficient for generating the response r_i.

2.3. Best-of-N Sampling

To leverage the self-supervised reward computed via context ablation, we employ a *best-of-N* sampling strategy, which is a common way to test the effectiveness of a reward design (Gao et al., 2023a; Lightman et al., 2024) as a performance oracle without any confounders from training. For convenience, we first generate the full response $R = \{r_1, e_1, \ldots, r_S, e_S\}$ which includes a set of statements (r_i) paired with citations (e_i), and then locate the position of e_i, i.e., where the citation tags `<cite>...</cite>` are generated. Within the citation tags of e_i, we resample N candidate citation sequences $(e_i^{(1)}, \ldots, e_i^{(N)})$, by making the model to continue the generation from $\{C, Q, r_1, e_1, \ldots, r_i\}$, and then select the best citation (e_i^*) that maximizes the combined reward metric, Eq. (1). The corresponding procedure is shown in Algorithm 1. After obtaining all the selected citations $\{e_1^*, \ldots, e_S^*\}$, we replace the original citation sequence e_i with the optimal citation e_i^* for each response statement r_i, while keeping the response statements $\{r_1, \ldots, r_S\}$ unchanged. This process is repeated for each statement in the response R to obtain the final, citation-improved output $R^* = \{r_1, e_1^*, \ldots, r_S, e_S^*\}$. To prevent the model from citing too many sentences, we exclude the candidate e_i if the cited text (E_i) is longer than $L_{\text{max}} = 384$ tokens in total, unless E_i are all from a single long sentence.

Algorithm 1 SelfCite Best-of-N Sampling for Citations

Require: LM p_{LM}, context C, query Q, response R, # of candidates N, length limit L_{max}, $T(\cdot)$ counts # of tokens in a text, $\#(\cdot)$ counts # of sentences in a citation.
for $r_i \in R$ **do**
  $\text{Reward}(k) = -\infty$ for $k = 1, \ldots, N$
  for $k = 1, \ldots, N$ **do**
    $e_i^{(k)} \sim p_{\text{LM}}(\cdot \mid r_i, C, Q)$
    if $T(E_i^{(k)}) <= L_{\text{max}}$ or $\#(e_i^{(k)}) = 1$ **then**
      $\text{Reward}(k)$
      $= \log p_{\text{LM}}(r_i|E_i^{(k)}) - \log p_{\text{LM}}(r_i|C \setminus E_i^{(k)})$
    end if
  end for
  $k^* = \arg\max_k \text{Reward}(k)$
  $e_i^* = e_i^{(k^*)}$
end for
return $R^* = \{r_1, e_1^*, \ldots, r_S, e_S^*\}$

2.4. Preference Optimization

Best-of-N sampling is a straightforward way to obtain better citations, but at the additional inference cost of generating candidates and reranking. Thus, we try to internalize the ability of generating better citations back to the LM itself.

Given documents and queries, we can prompt the LM to generate the responses along with the citations $R = \{r_1, e_1, ..., r_S, e_S\}$. By further applying best-of-N sampling, we can obtain new responses of the same statements but with better citations $R^* = \{r_1, e_1^*, ..., r_S, e_S^*\}$. Such preference data can be used in direct preference optimization (DPO) (Rafailov et al., 2024) to align the model based on the preference between the original outputs and improved outputs. Instead of using DPO, we choose its variant SimPO (Meng et al., 2024) here, as SimPO does not require

a reference model and allows 2× memory saving for 25.6K long-context fine-tuning. Through this self-supervised process, which does not require ground-truth answers or human annotations, the model learns to generate more accurate and contextually grounded citations on its own.

3. Experiments

We evaluate the effectiveness of SelfCite by applying the best-of-N sampling and preference optimization methods to existing models that generate responses with citations.

3.1. Model Details

We use LongCite-8B, the Llama-3.1-8B model (Dubey et al., 2024) fine-tuned on LongCite-45K SFT data (Zhang et al., 2024) as the start point for both best-of-N sampling and preference optimization. We adopt the same text segmentation strategy from Zhang et al. (2024): each document is split into individual sentences using NLTK (Bird, 2006) and Chinese punctuations, and each sentence is prepended with a unique identifier in <C{i}> format. These identifiers serve as the *citation indices*, enabling the model to cite relevant context right after the statements with the format of <statement> {content ...} <cite>[i_1-i_2][i_3-i_4]...</cite></statement>. This format allows the model to cite a single sentence (e.g. $i_1 = i_2$) or a span (e.g. $i_1 < i_2$) efficiently within several tokens. The responses are generated via top-p sampling (Holtzman et al., 2020) with p=0.7 and temperature=0.95. We set p=0.9 and temperature=1.2 when doing best-of-N sampling for the citation strings to increase the diversity. We set N=10 in all the experiments considering the limited diversity in citations.[1]

3.2. Preference Optimization

LongCite-45K. Best-of-N sampling (Section 2.3) requires no training, so no training data is used. For preference optimization with SimPO (Section 2.4), we use 2K document–question pairs from LongCite-45K (Zhang et al., 2024) as the training set but we do not use its ground-truth responses with high-quality citations for SFT. Instead, we generate model responses from the documents and queries, then apply best-of-N to refine citations. We label the original responses as *rejected* and replace their citations with BoN-refined ones to create the *chosen* responses, forming preference pairs to build the dataset for SimPO.

Data Construction and Length Balancing Since best-of-N responses tend to have slightly longer citations, directly fine-tuning on them can lead the model to adopt a shortcut—generating longer citations instead of improving citation quality. To prevent this, we introduce *length balancing*: if an original response has a shorter citation length than the best-of-N response, we insert random citations from nearby sentences. This encourages the model to focus on *where* to cite rather than simply citing *more*. Details are provided in Appendix C, with an ablation study in Section 4.2.

[1] After deduplicating repeated citation candidates, on average there are only 4.8 candidates left per statement in the BoN experiment on LongBench-Cite, with a standard deviation of 3.2.

3.3. Evaluation

Benchmark. We evaluate our approach on **LongBench-Cite** (Zhang et al., 2024), a comprehensive benchmark specifically designed for *long-context QA with citations (LQAC)*. Given a long context C and a query Q, the model must produce a multi-statement answer with each statement cites relevant supporting sentences in C. Unlike chunk-level citation schemes (Gao et al., 2023b) which cites short paragraphs, LongBench-Cite adopts *sentence-level* citations to ensure semantic integrity and finer-grained evidence tracking. LongBench-Cite assesses two main aspects:

- **Citation Quality:** Whether each statement is fully supported by relevant and *only* relevant sentences. GPT-4o measures *citation recall* (extent to which a statement is fully or partially supported by the cited text) and *citation precision* (whether each cited text truly supports the statement). These are combined into a *citation F1* score. Additionally, we track *average citation length* (tokens per citation) to promote fine-grained citations over unnecessarily long passages.
- **Correctness:** How accurately and comprehensively the response answers the query disregarding the citations. This is scored by GPT-4o in a zero-/few-shot fashion based on the query and reference answers.

The benchmark contains five datasets, including single-doc QA *MultiFieldQA-en/zh* (Bai et al., 2023), multi-doc QA *HotpotQA* (Yang et al., 2018) and *DuReader* (He et al., 2018), one summarization dataset *GovReport* (Huang et al., 2021), and *LongBench-Chat* (Bai et al., 2024) which covers diverse real-world queries with long contexts such as document QA, summarization, and coding.

Baselines. SelfCite is compared with these baselines.

- **Prompting**: Zhang et al. (2024) propose the baseline of prompting LLMs with an one-shot example. This can be applied to proprietary models including GPT-4o (OpenAI, 2023), Claude-3-sonnet (Anthropic, 2024), and GLM-4 (GLM et al., 2024), as well as open-source models including GLM-4-9B-chat (GLM et al., 2024), Llama-3.1-{8,70}B-Instruct (Dubey et al., 2024), and Mistral-Large-Instruct (Mistral, 2024).
- **Contributive context attribution**: Contributive context attribution seeks to directly identify the parts of the context that *cause* the model to generate a par-

Table 1. Citation recall (R), citation precision (P), citation F1 (F1), and citation length evaluated on LongBench-Cite benchmark. The best of our results are bolded. The best of previous state of the art are underlined. † indicates the results taken from Zhang et al. (2024).

Model	Longbench-Chat			MultifieldQA			HotpotQA			Dureader			GovReport			Avg.	Citation
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1	Length
Proprietary models																	
GPT-4o†	46.7	53.5	46.7	79.0	87.9	80.6	55.7	62.3	53.4	65.6	74.2	67.4	73.4	90.4	79.8	65.6	220
Claude-3-sonnet†	52.0	67.8	55.1	64.7	85.8	71.3	46.4	65.8	49.9	67.7	89.2	75.5	77.4	**93.9**	84.1	67.2	132
GLM-4†	47.6	53.9	47.1	72.3	80.1	73.6	47.0	50.1	44.4	73.4	82.3	75.0	82.8	93.4	87.1	65.4	169
Open-source models																	
GLM-4-9B-chat†	25.9	20.5	16.7	51.1	60.6	52.0	22.9	28.8	20.1	45.4	48.3	40.9	5.7	8.2	6.3	27.2	96
Llama-3.1-8B-Instruct†	14.1	19.5	12.4	29.8	44.3	31.6	20.2	30.9	20.9	22.0	25.1	17.0	16.2	25.3	16.8	19.7	100
Llama-3.1-70B-Instruct†	25.8	32.0	23.2	53.2	65.2	53.9	29.6	37.3	28.6	38.2	46.0	35.4	53.4	77.5	60.7	40.4	174
Mistral-Large-Instruct†	19.8	23.9	19.0	71.8	80.7	73.8	34.5	40.9	32.1	58.3	67.0	60.1	67.9	79.6	72.5	51.5	132
Contributive context attribution (*with Llama-3.1-8B-Instruct*)																	
ContextCite (32 calls)	56.7	76.8	58.0	76.1	87.2	78.9	40.5	54.7	43.9	58.0	82.4	65.0	67.1	88.8	75.6	64.3	92.7
ContextCite (256 calls)	63.5	**83.1**	64.7	78.8	89.8	81.8	46.5	60.8	49.2	61.7	89.1	70.1	69.1	93.5	78.8	68.9	100.8
Fine-tuned models																	
LongCite-9B†	57.6	78.1	63.6	67.3	91.0	74.8	61.8	78.8	64.8	67.6	89.2	74.4	63.4	76.5	68.2	69.2	91
LongCite-8B†	62.0	79.7	67.4	74.7	93.0	80.8	59.2	72.1	60.3	68.3	85.6	73.1	74.0	86.6	78.5	72.0	85
+ *SimPO w/ NLI Rewards*	64.4	87.1	69.8	70.1	92.4	77.4	58.8	78.1	63.2	69.4	91.1	77.2	83.7	93	87.5	75.0	105.9
Ours: SelfCite																	
LongCite-8B (Our repro.)	67.0	78.1	66.6	74.8	90.7	79.9	60.8	77.9	64.1	67.1	87.2	73.7	81.6	89.3	84.5	73.8	83.5
+ BoN	68.4	81.3	71.2	76.1	92.8	81.2	67.2	81.0	68.8	70.6	90.9	76.9	**87.6**	92.4	**89.3**	77.5	93.4
+ SimPO	68.1	79.5	69.1	75.5	92.6	81.0	**69.4**	82.3	**71.5**	72.7	91.6	78.9	86.4	92.9	89.1	77.9	105.7
+ SimPO then BoN	**73.3**	79.4	**72.8**	76.7	**93.2**	82.2	**69.4**	**83.0**	71.1	**74.2**	**92.2**	**80.3**	86.7	92.7	89.2	**79.1**	94.7
Llama-3.1-8B-Instruct (*fully self-supervised setting*)																	
+ SFT on ContextCite	52.3	70.6	56.5	79.1	90.5	82.0	54.5	72.3	56.3	54.9	79.0	61.6	63.7	84.9	72.3	65.7	83.0
+ BoN	54.8	67.6	58.1	80.4	90.5	83.0	58.3	70.0	57.5	57.6	79.0	63.1	67.2	84.8	74.6	67.3	80.4
+ SimPO	63.3	74.3	64.6	80.2	88.9	82.4	59.7	76.9	61.0	59.0	80.9	65.4	68.5	86.6	76.1	69.9	90.2
+ SimPO then BoN	66.0	82.4	71.1	**81.5**	90.7	**83.2**	61.3	70.0	59.9	62.1	81.4	67.4	68.8	86.2	76.1	71.5	87.4

ticular statement. We consider ContextCite (Cohen-Wang et al., 2024), a contributive context attribution method that performs several random context ablations to model the effect of ablating different parts of the context on a generated statement. We use NLTK to split Llama-3.1-8B-Instruct's responses into statements, and then apply ContextCite with 32 and 256 times of random context ablations to get the citations, with the details described in Appendix B.

- **Fine-tuned models**: LongCite-8B and 9B released by Zhang et al. (2024), trained on LongCite-45K, fine-tuned from Llama-3.1-8B (Dubey et al., 2024) and GLM-4-9B (GLM et al., 2024), respectively. Additionally, we consider a baseline of finetuning LongCite-8B using SimPO with the NLI rewards which resembles Huang et al. (2024a), with the details in Appendix E.

3.4. Main Results

Citation Quality. Table 1 presents our main results. Our best-of-N sampling (BoN) consistently improves both **citation recall** and **citation precision** across tasks, increasing the overall F1 score from 73.8 to 77.5. Using SimPO to internalize BoN's gains—**eliminating the need for costly BoN sampling**—achieves a similar improvement, with an F1 of 77.9. Applying BoN again to the SimPO fine-tuned model further boosts **F1 by 5.3 points to 79.1**, the highest across the datasets, suggesting room for further gains. Our results surpass LongCite-8B/9B at similar citation lengths and outperform proprietary model prompting while producing shorter citations.

To better contextualize the gains of our proposed reward, we additionally implement a variant of SimPO using NLI-based citation precision/recall rewards from Huang et al. (2024a) by using the same training pipeline and initialization as our SimPO, modifying only the reward function (see details in Appendix E). As shown in row of *SimPO w/ NLI Rewards*, this baseline improves LongCite-8B on 3 out of 5 datasets, but is still consistently outperformed by SelfCite. This result highlights that while NLI-based rewards are helpful, our SelfCite reward provides a more accurate signal for optimizing citation quality.

Besides the fine-tuned baselines, we additionally compare our method to ContextCite for reference, a method very different from SelfCite–it does not directly generate citations, it estimates the importance scores of the context sentences after the response is generated (in Appendix B we show how to convert continuous importance scores into citations). Both SelfCite and ContextCite rely on the idea of context ablation, but our approach is significantly better. A key reason is that ContextCite estimates sentence importance from scratch using linear regression, while we rerank ex-

Table 2. Answer correctness when responding with or without citations. † indicates results taken from Zhang et al. (2024). The header contains abbreviations for the same five datasets in Table 1.

Model	Long.	Multi.	Hot.	Dur.	Gov.	Avg
Answering without citations						
LongSFT-8B†	68.6	83.6	69.0	62.3	54.4	67.6
LongSFT-9B†	64.6	83.3	67.5	66.3	46.4	65.6
Llama-3.1-8B-Instruct	66.0	83.7	65.8	62.8	66.1	68.9
Answering with citations						
LongCite-8B (Our repro.)	67.6	86.7	69.3	64.0	60.4	69.6
+ SimPO	67.4	86.7	67.5	66.0	61.3	69.8
Llama-3.1-8B-Instruct	58.4	75.3	67.3	59.3	56.4	63.3
+ SFT on ContextCite	58.8	83.4	65.8	57.8	57.5	64.6
+ SimPO	56.8	80.9	65.3	59.5	60.9	64.7

isting LLM-generated citation candidates, leading to more efficient and accurate citation quality estimation.

Finally, we evaluate the latest released *Claude Citations* API, as shown in Appendix D that SelfCite achieves strong results very close to this commercial-level API, validating the effectiveness of SelfCite.

Fully Self-Supervised Setting. In our main experiment, we start from the Llama-3.1-8B model fine-tuned on the LongCite-45K SFT data, which effectively kick-starts its ability to generate structured citations for best-of-N sampling. The subsequent SimPO alignment stage is entirely self-supervised. We are also curious if it is possible to start from a fully self-supervised SFT model and then apply our self-supervised alignment after that. To begin with, we automatically generate 11K citation SFT data using ContextCite (see Appendix B for details) to replace the LongCite-45K annotations in the training data, as shown in the results at the bottom of Table 1. We can see that SFT on ContextCite can achieve decent initial results (65.7 F1) but still far from LongCite-8B (73.8 F1). BoN helps improving F1 to 67.3. After SimPO training, it achieves 69.9 F1, and additionally applying BoN can boost its F1 by 5.8 to 71.5, significantly closing the gap to LongCite-8B, showing our alignment method not only improve the supervised models, but also enhance the models purely trained from self-supervision.

Answer Correctness. For best-of-N sampling, only the citation parts are modified, so the responses it generates to answer the questions are the same as those of the original LongCite-8B model, maintaining the same correctness. For the SimPO fine-tuned models, we test their answer correctness by the evaluation in Zhang et al. (2024), which contains two settings: answering with/without citations. If answering with citations, the model will be prompted to generate answers with structured citations, making the task more complex, and the citation parts will be removed when evaluating the answer correctness. The results in Table 2 show that the SimPO fine-tuning **does not change the correctness** of the LongCite-8B model much. The correctness is similar to LongSFT-8B/9B (Zhang et al., 2024), which are ablation baselines fine-tuned on LongCite-45k QA pairs but without the citation parts. The same observation still holds when starting from Llama-3.1-8B-Instruct, either SFT with ContextCite data or the further SimPO step do not change the answer correctness significantly. Under the same answer correctness, the additional "citations" can benefit the verifiability of the answers, enabling a user to easily double-check the answer, even in cases where the answers are wrong.

Chunk-level Citation Evaluation. Additionally, we evaluate our methods on the traditional chunk-level citation benchmark **ALCE** (Gao et al., 2023b). However, due to the mismatch of data distributions and different task settings during training (sentence-level) and evaluation (chunk-level), we consider this as a zero-shot evaluation, and the results are shown in Appendix F, due to the limited space.

4. Analysis

4.1. Ablation Study on Rewards

To better understand our final reward design, we explore various reward strategies in the BoN sampling process. Here, all BoN candidates are pre-generated and fixed, the reward is the only factor affecting results. Table 3 presents our ablation results on HotpotQA, while citation lengths are computed across all LongBench-Cite datasets for direct comparison with Table 1. We evaluate four alternative reward designs. *BoN by LM log prob* re-ranks candidates simply by the probability of the citation string, `<cite>[`$i_1 - i_2$`][`$i_3 - i_4$`]...</cite>`, which is similar to beam search but less costly. We observe that this strategy slightly boosts recall while reducing precision, resulting in a minor reduction in F1. *BoN by max citation length* always selects the candidates with the longest citations, i.e. citing the greatest number of sentences. Although it improves recall, it significantly reduces precision from 77.9 to 73.6 and inflates the citation length from 83.5 to 139.8. By contrast, both *BoN by Prob-Drop* and *BoN by Prob-Hold* improve recall without sacrificing precision. Finally, by combining both Prob-Drop and Prob-Hold into our final SelfCite reward, we achieve the best outcome, increasing **both recall and precision and a 4-point improvement in F1**.

We also explored different token-length limits for citations in the bottom of Table 3, as discussed in Section 2.3. By default, we exclude candidates citing more than 384 tokens, unless the citation contains only a single sentence. Lowering the cap to 256 tokens slightly hurts F1, while raising it to 512 tokens has negligible impact. Completely removing length limits inflates citation length to 121.9 tokens and yields worse precision (79.3) but slightly improved recall (67.9). We also notice that the 256 length limit still outperforms the LongCite-8B baseline (66.4 vs 64.1) while having almost

Table 3. Ablation study on HotpotQA citation recall, precision, and F1 (R, P, F1) and citation length for BoN decoding methods.

Decoding Methods	HotpotQA R	HotpotQA P	HotpotQA F1	Citation Length
LongCite-8B (Our repro.)	60.8	77.9	64.1	83.5
+ BoN by LM log prob	62.7	75.5	63.4	74.6
+ BoN by *max citation length*	66.5	73.6	65.1	139.8
+ BoN by Prob-Drop	65.6	78.1	66.6	92.9
+ BoN by Prob-Hold	66.2	78.1	67.0	93.4
+ BoN by SelfCite	67.2	81.0	68.8	93.4
w/ lower length limit (256)	65.8	78.8	66.4	84.5
w/ higher length limit (512)	67.0	82.2	68.5	99.2
w/o length limit (∞)	67.9	79.3	68.1	121.9

Table 4. Ablation study on HotpotQA citation recall, precision, and F1 (R, P, F1) and citation length for finetuned models.

Fine-tuning Methods	HotpotQA R	HotpotQA P	HotpotQA F1	Citation Length
LongCite-8B (Our repro.)	60.8	77.9	64.1	83.5
+ SimPO	69.4	82.3	71.5	105.7
+ SimPO + BoN	72.0	82.7	72.9	126.9
+ SimPO w/ or w/o length balancing				
w/ length balancing	69.4	82.3	71.5	105.7
w/o length balancing	64.4	62.9	60.5	152.9
+ SimPO w/ varying data sizes				
1K examples	62.5	78.9	65.7	90.1
2K examples	69.4	82.3	71.5	105.7
4K examples	68.5	80.4	70.3	134.1
8K examples	64.6	79.5	65.9	158.1
+ SFT on BoN responses	68.8	77.3	68.4	98.7
+ SimPO by denoising perturbed citations				
On original responses	40.5	50.5	41.6	88.8
On BoN responses	42.6	50.7	42.3	79.7

equally long citation length (84.5 vs 83.5), showing that **the improvement of SelfCite correlates less with the citation length**. Overall, using a 384-token limit achieves a good balance for short citation lengths and strong performance.

4.2. Citation Length Balance

As noted in Section 3.2, BoN selects slightly longer citations, making it easy for a model trained directly on BoN-preferred data to adopt the shortcut of generating longer citations without improving quality. To counter this, we apply *length balancing*, injecting random citations into examples where length bias exists to equalize the number of cited sentences. Table 4 (see w/ vs. w/o length balancing) highlights its critical role in length balancing. Without length balancing, the model overextends citations (average length 152.9), leading to lower precision (62.9) and F1 (60.5). In contrast, enabling length balancing maintains high precision (82.3) and recall (69.4), achieving a better F1 of 71.5 while keeping citation length reasonable (105.7). These results confirm that **length balancing prevents shortcut learning, ensuring the model truly learns to cite accurately**.

4.3. Training Size of SimPO

In prior study (Zhou et al., 2023), 1K examples are sufficient to align user preferences effectively. Table 4 presents SimPO results with 1K to 8K examples. 1K examples already bring a moderate improvement, raising F1 from 64.1 to 65.7, with gains in precision and recall. Using 2K examples further boosts F1 to 71.5, while 4K leads to saturated improvement. However, at 8K examples, performance declines, and citation length rises to 158.1. We attribute this to SimPO's off-policy nature, especially because it lacks a reference model to constrain the output distributions to be similar to the collected data. As training steps grow, the model may drift from the collected data, potential overfitting to the biases in preference data. Thus, further fine-tuning may degrade citation quality. To address this, we show initial results from iterative SimPO in Section 4.6.

4.4. SimPO vs. SFT on Best-of-N responses

We also show the effect of applying standard supervised fine-tuning (SFT) on the responses selected by best-of-N sampling, which is a simplified alternative of preference optimization. As the result shown in the last row in Table 4, SFT also improves the F1 score from 64.1 to 68.4, but it still falls behind 71.5 of SimPO. This result confirms that it is necessary to train the model via SimPO with preference data, which enables the model to distinguish between bad and good citations, and thus improve the citation quality.

4.5. Off-policy Denoising Perturbed Citations

We explored a purely *off-policy* alternative approach. Specifically, given a model-generated response, we randomly shift its citation spans to create perturbed variants. SimPO training pairs were then constructed by preferring the *original* citation over the *perturbed* one, encouraging the model to "denoise" citations by restoring their original spans. However, as shown at the bottom of Table 4, this approach *degrades* performance, both when applied to original and best-of-N responses. We attribute this to a mismatch between the training data and the model's natural error distribution—since random shifts do not reflect typical citation errors, they fail to provide useful guidance for improvement.

4.6. Iterative Preference Optimization

It has been discussed that an *on-policy* alignment process can be beneficial to avoid reward exploitation (Bai et al., 2022) and maintains consistency between the generated data and the model's evolving output distribution. We thus experiment with iteratively performing SimPO, similar to the concepts of recent studies (Pang et al., 2024; Yasunaga et al., 2024), to maintain the consistency between the generated data and the model's evolving output distribution.

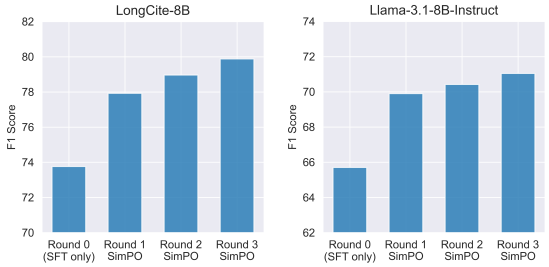

Figure 2. Iteratively applying SimPO for three iterations.

Specifically, after fine-tuning with SimPO, we generate a new dataset via BoN, which is also 2K in size but not overlapped with previous iterations. We continue training the model and repeat the process for three rounds. As shown in Figure 2, while the largest improvement occurs in the first round, improvements continue over three iterations, which further validates the reliability of our reward signal. Iterative SimPO is still not perfect since it remains an off-policy method. Given that our reward can be cheaply computed, we believe that on-policy methods like PPO (Schulman et al., 2017) could further enhance performance. We leave the exploration of such approaches for future work.

4.7. Latency of Best-of-N

Table 5 reports the average per-example latency on LongBench-Cite. As expected, Best-of-N (BoN) introduces additional latency due to the need to generate and rerank multiple citation candidates. In our setup, we use N = 10 candidates, but the sampling time is not 10× longer than direct decoding. This is because we only re-sample short citation spans (typically 5–10 tokens), not the full responses, resulting in relatively lightweight sampling overhead.

However, the increased latency from BoN is not a major concern, because our SelfCite SimPO model also achieves the same performance as BoN in a single pass, without additional latency. For scenarios requiring maximum efficiency, we recommend using the SimPO model directly.

4.8. Qualitative Study

Finally, we examine an example that requires citing multiple context sentences to support a complex response. As shown in Table 6, the response integrates information from sentences 302, 303, and 306. Direct sampling (2) omits sentence 302 while incorrectly including 305. In contrast, the best-of-N candidate (1) correctly includes 302 and excludes 305, achieving a slightly higher reward (0.578 vs. 0.547), demonstrating the effectiveness of our reward design. We also present candidates (3) and (4), which cite more irrelevant sentences and miss key citations, leading to even lower rewards. Additional qualitative examples are provided in Appendix H.

Table 5. Average latency per example on LongBench-Cite (8 × A100 GPUs, batch size 1, model parallel).

Method	Avg Latency (s)
LongCite-8B	24.3
SelfCite BoN Sampling	149.0
SelfCite BoN Reranking	34.0
SelfCite SimPO model	26.2

5. Related Work

Citations for Language Models. Recent work has explored various approaches to teaching language models to generate citations, including fine-tuning with direct human feedback or annotations (Nakano et al., 2021; Menick et al., 2022; Slobodkin et al., 2024), rewards from external NLI models (Huang et al., 2024a;b), and prompting-based methods (Gao et al., 2022; 2023b) to explicitly incorporate relevant retrieved documents. Given the high cost of human annotation, Zhang et al. (2024) introduced **CoF** ("**Co**arse to **F**ine"), an automated multi-stage pipeline that simulates human annotation. This approach leverages proprietary LLMs for query generation, chunk-level retrieval, and sentence-level citation extraction, achieving high citation quality through supervised fine-tuning. However, it depends on larger proprietary models two proprietary APIs—GLM-4 for the LLM and Zhipu Embedding-v2 for retrieval[2]— with carefully designed prompting, effectively distilling the capabilities of these proprietary models into much smaller models in 8B/9B. In contrast, our SelfCite aims at completely eliminating the reliance on annotations for citation, either from human or proprietary APIs. Instead, our method enables a small 8B model to assess citation quality itself using self-supervised reward signal from context ablation, effectively self-improving without external supervision. We additionally provide Table 9 to contrast the key differences between SelfCite and prior papers in Appendix G.

Contributive Context Attribution. Besides being self-supervised, SelfCite also adopts the view that citations should reference the sources from the context that a model actually *uses* when generating a statement–known as *contributive* attribution (Worledge et al., 2023)–rather than any sources that merely *support* the claim. Our reward signal naturally aligns with this attribution framework, as context ablation identifies the sources that *cause* the model to produce a statement. Existing contributive attribution methods for LLMs typically require extensive context ablations or other computationally expensive techniques, such as gradient-based analysis during inference (Cohen-Wang et al., 2024; Qi et al., 2024; Phukan et al., 2024). In contrast,

[2] https://open.bigmodel.cn/pricing

Table 6. An example of differences in the citation from baseline vs BoN. Related information are highlighted in the context/response.

Sent. ID	**Context Sentences** (only showing a paragraph due to limited space)
302 (✓)	In general, consumer advocates believe that any comprehensive federal privacy policy should complement, and not supplant, sector-specific privacy legislation or state-level legislation.
303 (✓)	Finding a global consensus on how to balance open data flows and privacy protection may be key to maintaining trust in the digital environment and advancing international trade.
304 (✗)	One study found that over 120 countries have laws related to personal data protection.
305 (✗)	Divergent national privacy approaches raise the costs of doing business and make it harder for governments to collaborate and share data, whether for scientific research, defense, or law enforcement.
306 (✓)	A system for global interoperability in a least trade-restrictive and nondiscriminatory way between different national systems could help minimize costs and allow entities in different jurisdictions with varying online privacy regimes to share data via cross-border data flows.
Query	Please write a one-page summary of the above government report.
Response (only single statement due to space)	[...] The report concludes by noting that finding a global consensus on how to balance open data flows and privacy protection may be key to maintaining trust in the digital environment and advancing international trade. The report suggests that Congress may consider comprehensive privacy legislation and examine the potential challenges and implications of building a system of interoperability between different national privacy regimes. [...]

BoN Candidates	**Citation Strings** (*green: correct*; *red: wrong*)	**Missing Citations**	**SelfCite Reward**
(1) Best candidate	[302-303][306-306]	–	0.578
(2) Direct sampling	[303-303][305-306]	(302)	0.547
(3) Other candidate	[303-304][308-308][310-311]	(302, 306)	0.461
(4) Other candidate	[303-303][309-309][311-311]	(302, 306)	0.375

SelfCite simply generate the citation tags, and refine citation candidates by preference optimization with reward signals from context ablations, effectively teaching the model to perform contributive context attribution itself.

We also note that there is a distinction between *corroborative* citation—highlighting sources that *support* a claim, as used in benchmarks like LongBench-Cite—and *contributive* attribution, as emphasized in ContextCite. While SelfCite applies a contributive alignment method (via ablations) in the context of a corroborative evaluation framework, we find the two objectives to be at least partially aligned: citations that genuinely influence the generation are often also semantically supportive. Although this alignment is not guaranteed, our empirical results show that enforcing contributive attribution leads to clear improvements on corroborative benchmarks, suggesting that current corroborative methods (e.g., LongCite) still have significant headroom for improvement—even under a slightly mismatched objective.

Self-Supervised Alignment and Reward Modeling. Another relevant area is self- or weakly-supervised approaches for aligning LLMs without human supervision (Kim et al., 2023; Yuan et al., 2024), reducing the need for explicit human feedback (Ouyang et al., 2022), or curating high-quality data for supervised fine-tuning (Zhou et al., 2023). SelfCite shares the same spirit by computing simple probability *differences* under context ablation as rewards, eliminating the need for additional annotation process.

6. Conclusion and Limitations

In this work, we introduced SelfCite, a self-supervised framework that aligns LLMs to generate more accurate, fine-grained citations by directly leveraging their own probabilities for necessity and sufficiency rewards through context ablation. With best-of-N sampling and preference optimization, SelfCite significantly improves citation correctness on the LongBench-Cite benchmark without requiring human annotation, offering a promising self-improving direction towards verifiable and trustworthy LLMs.

SelfCite also has limitations: 1) While achieving good results with SimPO, integrating other preference optimization or reinforcement learning (RL) algorithms, e.g., PPO (Schulman et al., 2017), remains under explored. However, as prior study (Mudgal et al., 2024) shows BoN closely approximates the upper-bound scores of RL, we mainly validate our reward design using BoN, following the established practices (Gao et al., 2023a; Lightman et al., 2024), and further verify it with a training-based method, SimPO. 2) The SelfCite framework assumes the output probabilities of LLMs are available, which may not work for some proprietary LLMs. 3) While we focus on self-supervision in preference optimization for improving the quality of existing citations generated by LLMs, better unsupervised ways to kick-start LLMs' ability in generating structured citations can be further explored.

Impact Statement

This paper introduces SelfCite, a self-supervised framework for improving citation accuracy in large language models (LLMs). Our method enhances the verifiability and trustworthiness of LLM-generated content by aligning citations with relevant supporting evidence in a scalable manner, without relying on costly human annotations. By improving citation quality, SelfCite contributes to the broader goal of reducing misinformation and hallucinations in AI-generated responses. Ensuring that LLMs provide accurate and properly attributed information is particularly crucial in high-stakes domains such as healthcare, law, and journalism, where incorrect or unverified information can have significant real-world consequences. Overall, SelfCite aligns with the broader ethical goal of making machine learning systems more transparent and accountable, reducing the risk of unchecked misinformation while maintaining the efficiency and scalability required for real-world applications.

Acknowledgements

We thank Jiajie Zhang and Yushi Bai for their assistance in providing implementation details of LongCite. Special thanks to Pin-Lun (Byron) Hsu for his invaluable support and guidance with Liger-Kernel. We are also grateful to Tianyu Gao for his timely help in setting up the ALCE benchmark during the rebuttal period. We also appreciate Andrei Barbu, Linlu Qiu, Weijia Shi for their valuable discussions. Yung-Sung was sponsored by the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Anthropic. Anthropic: Introducing claude 3.5 sonnet, 2024. URL https://www.anthropic.com/news/claude-3-5-sonnet.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., Du, Z., Liu, X., Zeng, A., Hou, L., et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023.

Bai, Y., Lv, X., Zhang, J., He, Y., Qi, J., Hou, L., Tang, J., Dong, Y., and Li, J. Longalign: A recipe for long context alignment of large language models. *arXiv preprint arXiv:2401.18058*, 2024.

Bird, S. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pp. 69–72, 2006.

Chuang, Y.-S., Qiu, L., Hsieh, C.-Y., Krishna, R., Kim, Y., and Glass, J. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436, 2024.

Cohen-Wang, B., Shah, H., Georgiev, K., and Madry, A. Contextcite: Attributing model generation to context. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., Zhao, V. Y., Lao, N., Lee, H., Juan, D.-C., et al. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*, 2022.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023a.

Gao, T., Yen, H., Yu, J., and Chen, D. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023b.

GLM, T., Zeng, A., Xu, B., Wang, B., Zhang, C., Yin, D., Rojas, D., Feng, G., Zhao, H., Lai, H., Yu, H., Wang, H., Sun, J., Zhang, J., Cheng, J., Gui, J., Tang, J., Zhang, J., Li, J., Zhao, L., Wu, L., Zhong, L., Liu, M., Huang, M., Zhang, P., Zheng, Q., Lu, R., Duan, S., Zhang, S., Cao, S., Yang, S., Tam, W. L., Zhao, W., Liu, X., Xia, X., Zhang, X., Gu, X., Lv, X., Liu, X., Liu, X., Yang, X., Song, X., Zhang, X., An, Y., Xu, Y., Niu, Y., Yang, Y., Li, Y., Bai, Y., Dong, Y., Qi, Z., Wang, Z., Yang, Z., Du, Z., Hou, Z., and Wang, Z. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024.

He, W., Liu, K., Liu, J., Lyu, Y., Zhao, S., Xiao, X., Liu, Y., Wang, Y., Wu, H., She, Q., et al. Dureader: a chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pp. 37–46, 2018.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Hsu, P.-L., Dai, Y., Kothapalli, V., Song, Q., Tang, S., Zhu, S., Shimizu, S., Sahni, S., Ning, H., and Chen, Y. Liger kernel: Efficient triton kernels for llm training. *arXiv preprint arXiv:2410.10989*, 2024. URL https://arxiv.org/abs/2410.10989.

Huang, C., Wu, Z., Hu, Y., and Wang, W. Training language models to generate text with citations via fine-grained rewards. *arXiv preprint arXiv:2402.04315*, 2024a.

Huang, L., Cao, S., Parulian, N., Ji, H., and Wang, L. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1419–1436, 2021.

Huang, L., Feng, X., Ma, W., Zhao, L., Fan, Y., Zhong, W., Xu, D., Yang, Q., Liu, H., and Qin, B. Advancing large language model attribution through self-improving. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3822–3836, 2024b.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Kim, S., Bae, S., Shin, J., Kang, S., Kwak, D., Yoo, K., and Seo, M. Aligning large language models through synthetic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13677–13700, 2023.

Lei, T., Barzilay, R., and Jaakkola, T. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117, 2016.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Meng, Y., Xia, M., and Chen, D. SimPO: Simple preference optimization with a reference-free reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=3Tzcot1LKb.

Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., et al. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.

Mistral. Mistral large, 2024. URL https://mistral.ai/news/mistral-large/.

Mudgal, S., Lee, J., Ganapathy, H., Li, Y., Wang, T., Huang, Y., Chen, Z., Cheng, H.-T., Collins, M., Strohman, T., et al. Controlled decoding from language models. In *International Conference on Machine Learning*, pp. 36486–36503. PMLR, 2024.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

OpenAI. Introducing chatgpt, November 2022. URL https://openai.com/blog/chatgpt.

OpenAI. Gpt-4 technical report, 2023. URL https://cdn.openai.com/papers/gpt-4.pdf.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., L. Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. *arXiv preprint 2203.02155*, 2022.

Pang, R. Y., Yuan, W., Cho, K., He, H., Sukhbaatar, S., and Weston, J. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*, 2024.

Phukan, A., Somasundaram, S., Saxena, A., Goswami, K., and Srinivasan, B. V. Peering into the mind of language models: An approach for attribution in contextual question answering. *arXiv preprint arXiv:2405.17980*, 2024.

Qi, J., Sarti, G., Fernández, R., and Bisazza, A. Model internals-based answer attribution for trustworthy retrieval-augmented generation. *arXiv preprint arXiv:2406.13663*, 2024.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Shi, W., Han, X., Lewis, M., Tsvetkov, Y., Zettlemoyer, L., and Yih, W.-t. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791, 2024.

Slobodkin, A., Hirsch, E., Cattan, A., Schuster, T., and Dagan, I. Attribute first, then generate: Locally-attributable grounded text generation. *arXiv preprint arXiv:2403.17104*, 2024.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Worledge, T., Shen, J. H., Meister, N., Winston, C., and Guestrin, C. Unifying corroborative and contributive attributions in large language models. *arXiv preprint arXiv:2311.12233*, 2023.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Yasunaga, M., Shamis, L., Zhou, C., Cohen, A., Weston, J., Zettlemoyer, L., and Ghazvininejad, M. Alma: Alignment with minimal annotation. *arXiv preprint arXiv:2412.04305*, 2024.

Yuan, W., Pang, R. Y., Cho, K., Li, X., Sukhbaatar, S., Xu, J., and Weston, J. E. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=0NphYCmgua.

Zhang, J., Bai, Y., Lv, X., Gu, W., Liu, D., Zou, M., Cao, S., Hou, L., Dong, Y., Feng, L., et al. Longcite: Enabling llms to generate fine-grained citations in long-context qa. *arXiv preprint arXiv:2409.02897*, 2024.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., YU, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., and Levy, O. LIMA: Less is more for alignment. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 55006–55021, 2023.

A. Implementation Details

For SimPO fine-tuning, we randomly sample 2K document and question pairs from the LongCite-45k data, generate the best-of-N responses with our Algorithm 1 to obtain the preference data, and train for one epoch. We sample another 100 examples as development set to pick the best learning rate from {1e-7, 3e-7, 5e-7, 7e-7}. We keep other hyperparameters the same as the original SimPO (Meng et al., 2024). We follow the same prompt format used in Zhang et al. (2024)[3] to keep the comparison fair. For the iterative SimPO experiment, in each iteration, we sampled a new, non-overlapping subset of 2K examples to ensure no data repetition across iterations. For self-supervised SFT, we generate 11K citation data unsupervisedly from ContextCite outputs as described in Appendix B, trained with a larger learning rate 7e-6.

We use the SimPO source code [4] built from Huggingface Transformers (Wolf et al., 2020) for the finetuning experiments, as well as Liger-Kernel (Hsu et al., 2024)[5] to enable memory efficient training for long-context examples in LongCite-45K without tensor parallelization. We run all the finetuning experiments on with 8×A100 GPUs of 80 GB memory on a single node. The batch size is set to 1 per GPU due to the long context examples. We set our max context length to 25600 to prevent OOM. For the data examples longer than 25600, we perform truncation, start from truncating the sentences that are the most far away from the sentences cited by the ground truth annotation, so as to keep the impact of truncation to be minimum.

When evaluating the citation length, as well as calculating the token length limit of 384 for excluding long BoN candidates, we follow Zhang et al. (2024) to use GLM4-9B's tokenizer to count tokens.

In the ablation study of off-policy denoising in Section 4.5, the citation examples for denoising are collected by randomly shifting existing citation spans by 3-10 positions in sentence indices.

B. Obtaining Citations from ContextCite

In this section, we first describe how the ContextCite method (Cohen-Wang et al., 2024) estimates continuous attribution scores for each sentence in the context. We then explain a simple heuristic for extracting citations (i.e., selecting a subset of context sources) from these scores.

B.1. ContextCite

Given a language model p_{LM}, a context C, a query Q and a generated response R, ContextCite aims to quantify how each *source* in the context $C = \{c_1, c_2, \ldots, c_{|C|}\}$ contributes to the generated response R (in our case, the sources are sentences). To do so, ContextCite performs several random context ablations. We begin by introducing some notation to describe these ablations. Let $v \in \{0,1\}^{|C|}$ be an ablation vector whose i-th entry toggles whether source c_i is included ($v_i = 1$) or excluded ($v_i = 0$). We write $\textsc{Ablate}(C, v)$ to denote a modified version of the original context C in which sources for which $v_i = 0$ are omitted. ContextCite seeks to understand how the probability of generating the original generated response,

$$f(v) := p_{\text{LM}}(R \mid \textsc{Ablate}(C, v), Q),$$

changes as a function of the ablation vector v.

Attribution via Surrogate Modeling. Directly measuring $f(v)$ for all $2^{|C|}$ ablation vectors is infeasible for large $|C|$. Hence, ContextCite seeks to identify a *surrogate model* $\hat{f}(v)$ that is easy to understand and approximates $f(v)$ well. To simplify this surrogate modeling task, ContextCite applies a logit transform to f, which maps values in $(0, 1)$ to $(-\infty, \infty)$):

$$g(v) := \sigma^{-1}(f(v)) \;=\; \log\Big(\frac{f(v)}{1 - f(v)}\Big).$$

ContextCite then approximates $g(v)$ using a sparse linear function,

$$\hat{g}(v) \;=\; \hat{w}^\top v + \hat{b}.$$

Notice that resulting weights $\hat{w} \in \mathbb{R}^{|C|}$ encode the importance of each source c_i to the probability of generating the original response; they can be interpreted directly as attribution scores (higher scores suggest greater importance).

[3] https://github.com/THUDM/LongCite
[4] https://github.com/princeton-nlp/SimPO
[5] https://github.com/linkedin/Liger-Kernel

Finding a Surrogate Model via LASSO. To learn the parameters $\hat{w}$ and $\hat{b}$ of the surrogate model, ContextCite randomly samples a small number of ablation vectors and measures the corresponding probabilities of generating the original response. It then uses this "training dataset" to fit a sparse linear model with LASSO. Concretely, it learns a surrogate model with the following three steps:

1. Sample n ablation vectors $\{v_i\}_{i=1}^n$ uniformly at random from $\{0,1\}^{|C|}$.
2. For each sample v_i, compute $g(v_i) = \sigma^{-1}(f(v_i))$ by running the LM with only the sources specified by v_i and measuring the (sigmoid) probability of R.
3. Solve a Lasso regression problem to find $\hat{w}$ and $\hat{b}$:

$$\hat{w}, \hat{b} \;=\; \arg\min_{w,\,b} \; \frac{1}{n} \sum_{i=1}^{n} \big(g(v_i) - w^\top v_i - b\big)^2 \;+\; \lambda \|w\|_1,$$

   where λ controls sparsity (larger λ drives more coefficients to zero).

In Cohen-Wang et al. (2024), typical choices of n range from 32 to 256, balancing cost (requires n LM forward passes) and accuracy. If there are multiple statements $\{r_1, r_2, ..., r_{|R|}\}$ in R, the same method can also be applied by focusing only on a subset of tokens in R.

B.2. Heuristic Citation Extraction

In our setting, we would like a discrete list of cited sentences for each generated statement, rather than a score for every sentence. We will now describe how to convert the attribution scores $\hat{w}$ into a discrete subset $C' \subseteq C$ of citations. Let t be a threshold, p be a cumulative probability mass cutoff, and k be a maximum citation limit.

Thresholding and Merging.

1. **Filtering:** Include only those sources c_i whose attribution score $\hat{w}_i \geq t$.
2. **Merging Adjacent Sources:** If multiple *consecutive* sources in the original text each exceed t, merge them into a single "span" S_j. We assign this merged span the maximum score among its constituents:

$$\hat{w}(S_j) = \max_{c_i \,\in\, S_j} \hat{w}_i.$$

   Here, adjacency is defined by the original ordering in C. For instance, if c_2 and c_3 both pass the threshold and appear consecutively, we merge them into a single span S_j.

Softmax Normalization. Let $\{S_j\}$ be the set of spans (or single sources) that survived the threshold. We normalize their scores into a probability distribution:

$$\hat{w}'(S_j) \;=\; \frac{\exp\big(\hat{w}(S_j)\big)}{\sum_i \exp\big(\hat{w}(S_i)\big)},$$

so that $\sum_j \hat{w}'(S_j) = 1$.

Top-p Selection. To avoid including too many low-value sources, we adopt a greedy approach:

$$\text{Add spans in order of descending } \hat{w}'(S_j), \text{ stopping once } \sum_{S_j \in C'} \hat{w}'(S_j) \;\geq\; p.$$

Top-k Filtering. Finally, if $|C'| > k$, we take only the k highest-scoring spans.

We set $t = 1.5$, $p = 0.7$, $k = 4$ in the experiment. When generating supervised fine-tuning (SFT) data, we discard any example for which more than 30% of its statements have no any citations that can survive threshold t. This ensures the dataset emphasizes cases where the LM's response can be tied to explicit context sources. We take the LongCite-45K document and question pairs to generate the responses by Llama-3.1-8B-Instruct itself, and then obtain citations with ContextCite (256 calls), transformed into the statement/citation format of LongCite-45K. Finally, we collect $\sim$ 11K examples used for SFT.

Table 7. Citation recall (R), citation precision (P), citation F1 (F1), and citation length evaluated on LongBench-Cite benchmark. The best results are bolded. † indicates the results taken from Zhang et al. (2024).

Model	Longbench-Chat			MultifieldQA			HotpotQA			Dureader			GovReport			Avg.	Citation
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1	Length
Proprietary models																	
GPT-4o†	46.7	53.5	46.7	**79.0**	87.9	80.6	55.7	62.3	53.4	65.6	74.2	67.4	73.4	90.4	79.8	65.6	220
Claude-3-sonnet†	52.0	67.8	55.1	64.7	85.8	71.3	46.4	65.8	49.9	67.7	89.2	75.5	77.4	93.9	84.1	67.2	132
GLM-4†	47.6	53.9	47.1	72.3	80.1	73.6	47.0	50.1	44.4	73.4	82.3	75.0	82.8	93.4	87.1	65.4	169
Ours: SelfCite																	
LongCite-8B (Our repro.)	67.0	78.1	66.6	74.8	90.7	79.9	60.8	77.9	64.1	67.1	87.2	73.7	81.6	89.3	84.5	73.8	83.5
+ BoN	68.4	81.3	71.2	76.1	92.8	81.2	67.2	81.0	68.8	70.6	90.9	76.9	**87.6**	92.4	**89.3**	77.5	93.4
+ SimPO	68.1	79.5	69.1	75.5	92.6	81.0	**69.4**	82.3	71.5	72.7	91.6	78.9	86.4	92.9	89.1	77.9	105.7
+ SimPO then BoN	**73.3**	79.4	**72.8**	76.7	93.2	82.2	**69.4**	83.0	71.1	74.2	92.2	80.3	86.7	92.7	89.2	79.1	94.7
Topline																	
Claude Citations	61.2	**81.7**	67.8	76.8	**98.4**	**84.9**	61.9	**94.1**	**72.9**	**88.5**	**99.7**	**93.2**	79.4	**99.2**	87.7	**81.3**	88.8

C. Length Balancing

To prevent the model from simply generating longer citations rather than focusing on citation correctness, we apply a *length balancing* procedure to align the total citation length in our two training responses: a *chosen prediction* and a *reject prediction*. First, we find the citation string (e.g., `[435-437]`) enclosed in `<cite>...</cite>` tags for each statement. We then measure each string's total citation "coverage", which means the total number of cited sentences in these intervals.

If a *reject prediction* has a total coverage lower than the corresponding *chosen prediction*, we insert additional citations around nearby sentence indices to match the *chosen* coverage. Conversely, if the *reject* coverage is larger, we randomly remove some of its intervals. We ensure new or inserted citations do not overlap existing intervals and keep them within a small window of 5–10 sentences away from the original citations to maintain realism. Finally, the *reject* and *chosen* will have matched coverage. This approach discourages the model from trivially learning to cite more sentences, instead prompting it to learn *where* and *how* to cite evidence more accurately. Our ablation in Section 4.2 shows that this length balancing technique significantly improves final citation quality.

D. Comparison with *Claude Citations* API

On January 23rd, 2025, Claude announced an API specialized for providing citations along with responses: *Claude Citations*[6]. We also try to evaluate this API on the LongBench-Cite benchmark. Since the implementation details and resource requirements (e.g., training data) of Claude Citations are not publicly available yet, and it relies on a significantly larger and more powerful LLM, Claude-3.5-Sonnet, which potentially has over 100 billions of parameters, we consider it as a topline of the benchmark rather than a baseline.

When evaluating it on Chinese examples from LongBench-Cite, we found that the API does not split Chinese text properly. As a result, it cites large passages when processing Chinese examples, leading to an average citation length of approximately 800 tokens per citation.

To address this issue, we pre-segment the text ourselves using exactly the same method as our approach following LongCite (Zhang et al., 2024), which uses NLTK and Chinese punctuation segmentation. We then run the Claude Citations API, as it supports both non-segmented and pre-segmented document inputs. The evaluation was conducted using the latest version of `claude-3-5-sonnet-20241022`.

As shown in Table 7, Claude Citations achieves an overall F1 score of 81.3, which is higher than all other models we have tested. However, the performance of Claude Citations is not consistent over all datasets. For example, it is worse than SelfCite on LongBench-Chat and GovReport. The main improvement of Claude is from the DuReader dataset, while the results on other datasets are comparable to the results of SelfCite. Given the fact that SelfCite leverages a much smaller 8B model compared to the Claude-3.5-Sonnet model, the result of SelfCite is very impressive, demonstrating its potential to serve as a strong alternative to proprietary solutions.

[6] https://www.anthropic.com/news/introducing-citations-api

E. Baseline: SimPO with NLI Rewards

To provide a stronger fine-tuned baseline, we implement a SimPO variant that adopts NLI-based citation rewards, following the design proposed by Huang et al. (2024a). For fair comparison, we keep our full SelfCite SimPO training pipeline—initializing from LongCite-8B and training on the LongCite-45k dataset—and modify only the reward function as a controlled experiment. This NLI-based reward combines two components:

- **Citation Recall Reward:** This measures whether the full set of cited sentences entails the model-generated statement. It is equivalent to the Citation Recall Reward proposed by Huang et al. (2024a).
- **Citation Precision Reward:** This estimates whether each cited sentence is necessary by ablating one sentence at a time and testing whether the remaining span still entails the statement. If entailment fails after removing a sentence, it indicates that the sentence contributes uniquely to the justification. To reduce latency, we ablate all sentences when the citation contains 5 or fewer; otherwise, we randomly sample 5 for ablation. When there are N ablations, each ablation makes a reward of $\frac{1}{N}$, and finally all ablations sum up to 1.0. It resembles the Citation Precision Reward proposed by Huang et al. (2024a).

We make both rewards positive and capped at 1.0, effectively constructing preference pairs for SimPO. We do not consider the Correctness Recall Reward from Huang et al. (2024a), because the LongCite-45k training set does not contain ground-truth answers. All entailment scores are computed using the public NLI model `google/t5_xxl_true_nli_mixture`[7].

F. Zero-shot Evaluation on Chunk-level Citation Benchmark ALCE

We additionally include the zero-shot evaluation on the chunk-level citation benchmark ALCE (Gao et al., 2023b) and report the results in Table 8. We find that our baseline model, LongCite-8B, although under a zero-shot setting (it is trained on sentence-level citation but test on chunk-level citations), already outperforms the prompting-based approach from Gao et al. (2023b) by a substantial margin in both citation recall and precision. Incorporating NLI-based rewards from Huang et al. (2024a) into our SimPO training yields further improvements. Most notably, our method—SimPO with SelfCite rewards—achieves the best performance among models trained on the same LongCite-45k dataset.

The last row of the table presents the best result reported by Huang et al. (2024a), who fine-tuned their model using supervised data. However, this setting is not directly comparable to ours for several reasons:

1. They optimize directly for the ALCE evaluation metric by using the same NLI evaluator model (`google/t5_xxl_true_nli_mixture`) to provide both training rewards and evaluation scores.
2. Their model is trained on the *in-distribution* QA training sets in ALCE, with exactly the same chunk-level format as the benchmark. In contrast, our SelfCite model is trained on *out-of-distribution* sentence-level citations from LongCite-45k.
3. Their method involves distillation from ChatGPT in the first stage, whereas ours does not rely on external supervision.

Despite this domain and format mismatch, SelfCite demonstrates strong generalization and consistently outperforms both LongCite-8B and the NLI-based SimPO baseline. This highlights the robustness and effectiveness of our approach even in cross-domain, cross-format transfer settings.

G. Comparison with Prior Studies

We further provide a comparison table in Table 9 to contrast the key differences between SelfCite and other prior studies on producing citations from LLMs. Among all methods, SelfCite is the only approach that supports sentence-level citation generation in a single pass, leverages preference optimization, and scales to 128K-token contexts—all without requiring additional supervision. In contrast, prior work such as ALCE (Gao et al., 2023b) and Huang et al. (2024a) use chunk-level citations for shorter context ($\leq$8K) and require prompt-based or supervised NLI signals. ContextCite (Cohen-Wang et al., 2024), while being sentence-level, relies on a computationally expensive (at least 32 inference calls) process for random context ablation and trains a linear model for estimating the importance scores. This comparison underscores the practical advantages and technical contributions of SelfCite.

[7] https://huggingface.co/google/t5_xxl_true_nli_mixture

Table 8. Evaluation on the chunk-level citation benchmark ALCE (Gao et al., 2023b). Our model (SimPO w/ SelfCite) is trained on sentence-level, out-of-distribution LongCite-45k data but still generalizes well to the chunk-level ALCE benchmark.

Model	ASQA			ELI5		
	EM Rec.	**Cite Rec.**	**Cite Prec.**	**Correct**	**Cite Rec.**	**Cite Prec.**
Gao et al. (2023b) (Prompting)						
Llama-2-13B-chat	34.66	37.48	39.62	12.77	17.13	17.05
Llama-3.1-8B-Instruct	42.68	50.64	53.08	13.63	34.66	32.08
Finetuned on LongCite-45k (Out-of-Distribution)						
LongCite-8B	42.11	62.27	57.00	15.37	30.54	29.15
+ SimPO w/ NLI Rewards	41.20	65.65	60.20	15.30	33.06	31.05
+ SimPO w/ SelfCite	**42.57**	**71.68**	**62.05**	**15.17**	**37.09**	**35.62**
Finetuned on ALCE train set (In-Distribution Supervision)						
Huang et al. (2024a)	40.05	77.83	76.33	11.54	60.86	60.23

Table 9. Key differences among prior methods on producing citations from LLMs. CC stands for ContextCite.

Method	Sentence-level citations?	One pass generation?	Preference optimization?	Handle 128K long-context?	External supervision?
ALCE (Gao et al., 2023b)	✗ (chunk-level)	✓	✗ (prompting)	✗ (8K)	2-shot prompting
Huang et al. (2024a)	✗ (chunk-level)	✓	✓	✗ (8K)	NLI + ground truth
CC (Cohen-Wang et al., 2024)	✓	✗ (at least 32 calls)	✗ (not generative)	✓	N/A
LongCite (Zhang et al., 2024)	✓	✓	✗ (SFT only)	✓	SFT data
SelfCite (Ours)	✓	✓	✓	✓	N/A

H. More Qualitative Examples

We further show more qualitative examples in Table 10,11, and 12, to represent the cases where SelfCite is better as well as where the LongCite-8B direct sampling baseline is better. In Table 10, SelfCite BoN avoid the cited irrelevant sentence (42, 47-50) by the baseline, while further including a correct citation (23) that are not found by the baseline. In Table 11, both SelfCite BoN and the baseline cites too many irrelevant sentences (391-393) but SelfCite BoN's citation is slightly better. In Table 12, SelfCite BoN wrongly includes 30 and misses 70, but the baseline is slightly better and only wrongly includes 71.

Table 10. An example of differences in the citation from baseline vs BoN. Related information are highlighted in the context/response.

Sent. ID	**Context Sentences** (only showing cited sentences due to space)
23 (✓)	We explored using a simple and cost-effective procedure to instruction finetune our continually pretrained long models without any human-annotated data.
42 (✗)	Collecting human demonstration and preference labels for LLM alignment is a cumbersome and expensive process (Ouyang et al., 2022; Touvron et al., 2023).
45 (✓)	In this work, we found that a simple and cheap approach which leverages a pre-built large and diverse short-prompt dataset works surprisingly well on long-context benchmarks.
46 (✓)	Specifically, we take the RLHF dataset used in LLAMA 2 CHAT and augment it with synthetic self-instruct (Wang et al., 2022) long data generated by LLAMA 2 CHAT itself, in the hope that the model can learn a diverse set of skills through the large amount of RLHF data and transfer that knowledge to long-context scenarios via self-instruct data.
47-50 (✗)	The data generation process focuses on QA-format tasks: starting from a long document in our pretraining corpus, we select a random chunk and prompt LLAMA 2 CHAT to write question-answer pairs based on information in the text chunk. [...]
Query	What aspects of the LLAMA Long model proposed above have changed relative to the LLAMA-based model? What improvements have been made?
Response (single statement due to space)	[...] 3. Instruction Tuning: The paper proposes a simple and cost-effective procedure to instruction finetune the continually pretrained long models without any human-annotated data. This involves using a pre-built large and diverse short-prompt dataset and augmenting it with synthetic self-instruct long data generated by LLAMA CHAT itself. [...]
Citation Strings (*green: correct*; *red: wrong*)	
Baseline	`[42-42][45-50]`
SelfCite BoN	`[23-23][45-45][46-46]`

Table 11. An example of differences in the citation from baseline vs BoN. Related information are highlighted in the context/response.

Sent. ID	**Context Sentences** (only showing cited sentences due to space)
299 (✗)	Ribosomes link amino acids together in the order specified by the codons of messenger RNA (mRNA) molecules to form polypeptide chains.
300 (✓)	Ribosomes consist of two major components: the small and large ribosomal subunits.
301 (✓)	Each subunit consists of one or more ribosomal RNA (rRNA) molecules and many ribosomal proteins (RPs or r-proteins).
302 (✓)	The ribosomes and associated molecules are also known as the translational apparatus.
383 (✗)	Ribosomes are minute particles consisting of RNA and associated proteins that function to synthesize proteins.
384 (✗)	Proteins are needed for many cellular functions such as repairing damage or directing chemical processes.
385 (✗)	Ribosomes can be found floating within the cytoplasm or attached to the endoplasmic reticulum.
390 (✓)	Each subunit is made of one or more rRNAs and many r-proteins.
391 (✗)	The small subunit (30S in bacteria and archaea, 40S in eukaryotes) has the decoding function, whereas the large subunit (50S in bacteria and archaea, 60S in eukaryotes) catalyzes the formation of peptide bonds, referred to as the peptidyl-transferase activity.
392 (✗)	The bacterial (and archaeal) small subunit contains the 16S rRNA and 21 r-proteins (Escherichia coli), whereas the eukaryotic small subunit contains the 18S rRNA and 32 r-proteins (Saccharomyces cerevisiae; although the numbers vary between species).
393 (✗)	The bacterial large subunit contains the 5S and 23S rRNAs and 34 r-proteins (E. coli), with the eukaryotic large subunit containing the 5S, 5.8S, and 25S / 28S rRNAs and 46 r-proteins (S. cerevisiae; again, the exact numbers vary between species).
Query	The large subunit and small subunit that use two types of RNA are major components that make up what?
Response (single statement due to space)	[...] The ribosome consists of two major components: the small and large ribosomal subunits. Each subunit consists of one or more ribosomal RNA (rRNA) molecules and many ribosomal proteins (RPs or r-proteins). The ribosomes and associated molecules are also known as the translational apparatus. [...]
Citation Strings (*green: correct*; *red: wrong*)	
Baseline	`[299-302][383-385][390-393]`
SelfCite BoN	`[300-302][390-393]`

Table 12. An example of differences in the citation from baseline vs BoN. Related information are highlighted in the context/response.

Sent. ID	**Context Sentences** (only showing cited sentences due to space)
28 (✓)	The Aegis BMD system exists in several variants.
29 (✓)	Listed in order of increasing capability, these include (but are not necessarily limited to) 3.6.X variant, the 4.0.3 variant, the 4.1 variant (also known as the Aegis Baseline [BL] 5.4 variant), the 5.0 CU (Capability Upgrade) variant (also known as the BL 9.1 variant), the 5.1 variant (also known as the BL 9.2 variant), and the 6.X variant (also known as the BL 10 variant).
30 (✗)	Figure 1 summarizes the capabilities of some of these variants (using their designations as of 2016) and correlates them with the phases of the European Phased Adaptive Approach (or EPAA; see discussion below) for European BMD operations.
61 (✓)	The BMD interceptor missiles used by Aegis ships are the Standard Missile-3 (SM-3), the SM-2 Block IV, and the SM-6.
62 (✓)	The SM-3 is designed to intercept ballistic missiles above the atmosphere (i.e., exo-atmospheric intercept), in the midcourse phase of an enemy ballistic missile's flight.
70 (✓)	The SM-2 Block IV is designed to intercept ballistic missiles inside the atmosphere (i.e., endo-atmospheric intercept), during the terminal phase of an enemy ballistic missile's flight.
71 (✗)	It is equipped with a blast fragmentation warhead.
Query	Please write a one-page summary of the above government report.
Response (single statement due to space)	[...] The Aegis BMD system exists in several variants with increasing capability, including the 3.6.X, 4.0.3, 4.1, 5.0 CU, 5.1, and 6.X variants. The system uses the Standard Missile-3 (SM-3), SM-2 Block IV, and SM-6 interceptor missiles. The SM-3 is designed to intercept ballistic missiles above the atmosphere in the midcourse phase, while the SM-2 Block IV and SM-6 are designed to intercept ballistic missiles inside the atmosphere in the terminal phase. [...]
Citation Strings (*green: correct*; *red: wrong*)	
Baseline	`[28-29][61-62][70-71]`
SelfCite BoN	`[28-30][61-62]` (missing: 70)