Evaluating Design Choices in Verifiable Generation with Open-source Models

Shuyang Cao and **Lu Wang**
University of Michigan
Ann Arbor, MI
{caoshuy, wangluxy}@umich.edu

Abstract

Verifiable generation is introduced to improve the transparency and trustworthiness of outputs produced by large language models (LLMs). Recent studies observe that open-source models struggle to include accurate citations to supporting documents in their generation with in-context learning, in contrast to the strong performance demonstrated by proprietary models. Our work aims to reveal the critical design choices that can benefit open-source models, including generation pipelines, fine-tuning methods, and inference-time compute techniques. We consider three generation pipelines, producing the outputs directly or decomposing the generation into subtasks. These generation pipelines are fine-tuned using supervised fine-tuning and preference-based optimization including further fine-tuning with rejection sampling data and direct preference optimization (DPO). The construction of preference data with varying content and citation diversity is also investigated. Additionally, we examine the benefit of an additional reranking step. With four open-source models, our experiments show that directly generating the outputs achieves the best performance. Compared to other fine-tuning methods, DPO that computes training signals from contrastive pairs consistently yields better performance, and it reaches the peak performance when the contrastive pairs are constructed with sufficient content diversity. We also find that reranking can further boost the performance of verifiable generation systems, but the marginal improvement might not justify the additional cost.

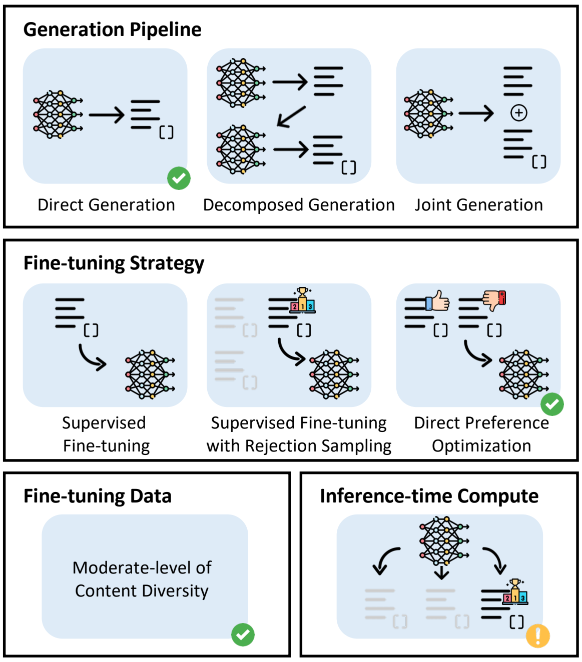

Figure 1: Illustration of our findings. To effectively employ medium-size open-source LLMs for verifiable generation, we suggest using the direct generation pipeline fine-tuned with DPO on samples that are sufficiently diverse in content. Though reranking over-generated samples during inference time can further increase output quality, the gain is limited.

1 Introduction

Verifiable generation, a generation paradigm where large language models (LLMs) are required to produce outputs along with citations to supporting documents, has gained increased attention for its potential to enhance user trust in the model responses (Liu et al., 2023; Huang and Chang, 2024). By allowing users to verify the generated content against cited sources, this approach not only enhances reliability but also facilitates access to additional relevant information. The paradigm has been incorporated into online services like Google and Bing Chat that are powered by proprietary models such as Gemini (Team et al., 2024) and GPT-4o (OpenAI et al., 2024).

Nevertheless, prior studies have demonstrated that *open-source* LLMs struggle to generate high-quality citations compared to proprietary models (Gao et al., 2023b), limiting their practical application. To address this gap, recent research has explored methods such as gathering citation-rich

Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025), pages 412–431
May 3, 2025 ©2025 Association for Computational Linguistics

data (Cao and Wang, 2024) and incorporating human preference data for fine-tuning (Huang et al., 2024). However, the scope of these investigations remains narrow, as they cover only a limited number of backbone LLMs, fine-tuning methods, and approaches to verifiable generation.

Our study systematically investigates the design considerations for verifiable generation using *medium-size open-source* LLMs. Specifically, we examine three crucial components: the structure of generation pipelines, the selection of fine-tuning strategies, and the construction of preference data.

Various approaches exist for generating outputs with citations. The simplest method is *direct generation*, where a model produces both content and citations in a single step. Alternatively, the task can be *decomposed* into two sequential steps handled by two separate models: content generation followed by citation generation. To further enhance citation quality, we introduce a hybrid *joint* pipeline, where the model first generates a response without citations, then revises it by incorporating citations within the same inference run.

Fine-tuning plays a crucial role in enhancing verifiable generation capabilities, especially for medium-size open-source LLMs. Starting with supervised fine-tuning using existing data, we collect preference data and perform further supervised fine-tuning on the most preferred samples (Nakano et al., 2022). Alternatively, we use direct preference optimization (DPO) on pairs of preferred and rejected samples (Rafailov et al., 2023). Both methods rely on preference data, collection of which is important to effectiveness of fine-tuning. Therefore, we construct preference data of various diversity in content and citations and study its impacts. We further explore the benefits of inference-time compute (Snell et al., 2024) by adding a scoring and reranking step upon over-generated model outputs.

We conduct experiments on SciFi, a citation-rich dataset (Cao and Wang, 2024), and ALCE, a question-answering dataset with retrieved documents for benchmarking verifiable generation (Gao et al., 2023b). The backbone models include Llama-3.1 (Grattafiori et al., 2024), Mistral-Nemo (AI, 2024), Qwen-2.5 (Team, 2024), and Phi-3.5 (Abdin et al., 2024). Models are fine-tuned on SciFi and tested on ALCE as an out-of-domain dataset. Our findings, as illustrated in Figure 1, indicate that:

1. Direct generation of outputs with citations outperforms pipelines that decompose the task into content generation and citation generation;
2. Fine-tuning on preference data of moderate content diversity with DPO yields the best-performing model and consistently improves the citation quality measured by the entailment level between the citation text and cited sources;
3. Reranking over-generated outputs consistently improves the fine-tuned generation pipelines, while the improvement is marginal for the top fine-tuned models.

2 Related Work

Verifiable Generation. Early exploration of large language models (LLMs) for verifiable generation trains LLMs to learn citation generation behaviors (Nakano et al., 2022). Recent advancements in LLM pre-training, instruction-tuning, and alignment have enabled prompting with human instructions to generate outputs with citations directly (Gao et al., 2023b), although the generated citations might not always be accurate. The intricacies of verifiable generation inspire a modular approach, where dedicated modules are employed for generating content and identifying supporting documents, respectively (Gao et al., 2023a). While more sophisticated systems can incorporate additional processes such as verification and regeneration to enhance citation quality (Sun et al., 2024), our work focuses on studying pipelines that generate final outputs either directly or in two steps, which is *orthogonal to the design of more complex systems* and can serve as the generation module for those systems.

Most existing verifiable generation systems rely on the citation generation capability of powerful backbone LLMs activated with instructions (Liu et al., 2023). For less capable models, fine-tuning with human-annotated (Menick et al., 2022) or web-sourced data (Cao and Wang, 2024) is essential to achieve comparable performance. Huang et al. (2024) propose warming up open-source LLMs with samples distilled from large proprietary models and using evaluation metrics to guide the construction of training samples for reinforcement learning. Our experiments similarly utilize preference data labeled with automatic metrics, though we verify the effectiveness of additional training data for various verifiable generation pipelines, with the training data constructed using different strategies under the same labeling budget.

Preference-based Optimization. Early work has aligned LLMs with human preference by training reward models using pairwise preference data and employing reinforcement learning (Ouyang et al., 2022). To circumvent the computational expenses associated with reward models in the learning algorithm (Schulman et al., 2017), Zhao et al. (2023) consider directly learning with contrastive loss on pairwise preference data. Rafailov et al. (2023) further introduce direct preference optimization (DPO), based on a mapping between reward functions and optimal policies, to align LLMs with human preference without reward models.

3 Verifiable Generation

In this section, we first introduce the candidate pipelines for verifiable generation (§3.1). Following the introduction of these pipelines, we discuss the strategies for fine-tuning models to enhance their performance and the methods for collecting training samples (§3.2). Lastly, we investigate the techniques that leverage inference-time compute (§3.3).

Task Formulation. We adhere to the task formulation outlined by (Gao et al., 2023b). Specifically, a system is given a query q and a set of candidate cited sources $\mathcal{D} = \{d_1, \ldots, d_M\}$, where M denotes the total number of candidate cited sources. Each cited source d_i can be either a text passage or an entire document, depending on the dataset. To process the lengthy aggregation of $\mathcal{D}$, we provide each system with summarized versions of the documents. We leave the exploration of long-context processing techniques to future research, as using summarized documents achieves comparable performance to enabling truncation or more sophisticated methods such as interactive lookup of full documents (Gao et al., 2023b).

Typically, verifiable generation systems indicate citations in their outputs with square brackets that enclose indices of cited sources (e.g., [1]). We denote the system output as $y = [y_1, \ldots, y_L]$ and define this output format by treating y_i as a tuple comprising a text token and a set of indices $\mathcal{C}_i = \{c_{i,1}, \ldots\}$, which point to the supporting documents. L represents the total number of text tokens. For instance, a generated span *"British Empire [3]"* corresponds to the tuples (`"British"`, `{}`) and (`"Empire"`, `{3}`).

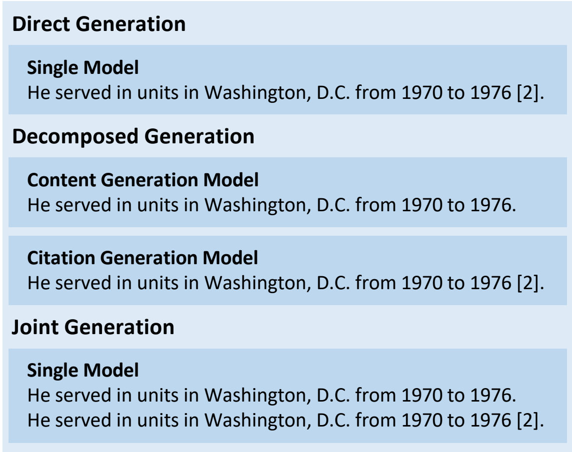

Figure 2: The generation pipelines examined in this study. Decomposed generation employs two separate models for content generation and citation generation. In contrast, both direct generation and joint generation utilize single models. While joint generation also decomposes verifiable generation, it performs the subtasks in a single pass.

3.1 Generation Pipelines

A generation pipeline outlines the process for deriving the final output y, as illustrated in Figure 2. We abstract each pipeline using formulations, with detailed templates and instructions provided in Appendix C.5.

Direct Generation. Direct generation treats the composition of responses with citations as an inherent ability of LLMs and leverages this capability to generate the final output in a single stage. Formally, $y = f(q, \mathcal{D})$, where f is an LLM. Additionally, f is supplied with instructions, which are omitted in the formulation for simplicity in this paper.

Decomposed Generation. Decomposed generation separates verifiable generation into two distinct steps—content generation and citation generation—employing a different model for each step. This separation enables dedicate optimization for each step. During content generation, an intermediate output without citation, denoted as $\bar{y}$, is produced as $\bar{y} = f_1(q, \mathcal{D})$, where $\mathcal{C}_i = \varnothing, \ \forall \bar{y}_i$. The intermediate output is then processed by a separate LLM specialized in citation generation to obtain the final output: $y = f_2(q, \bar{y}, \mathcal{D})$. Decomposed generation can be viewed as a post-hoc attribution method. Unlike traditional post-hoc attribution

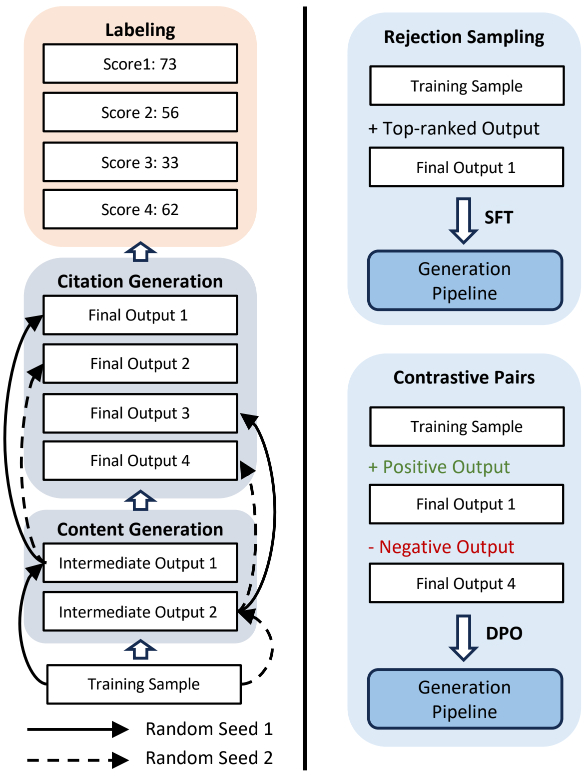

Figure 3: Left: The preference data construction process. Right: The studied preference-based optimization methods. We show an example of using two distinct random seeds for content generation and two distinct random seeds for citation generation, creating four final outputs in total.

methods that rely on pairwise similarity measures (e.g., entailment scores (Huo et al., 2023; Chen et al., 2023)), decomposed generation takes a generative approach and eliminates the need to iterate over all candidate cited sources individually.

Joint Generation. In decomposed generation, the content generation LLM is not explicitly required to establish connections between source documents and the intermediate output. This limitation can result in less grounded outputs and constrain the performance of the citation generation module. We propose a hybrid approach that combines direct and decomposed generation, where both intermediate and final outputs are generated sequentially in a single pass: $[\bar{y}; y] = f(q, \mathcal{D})$. $[\cdot; \cdot]$ denotes the concatenation of two sequences. By maintaining awareness of the requirements for the final output, the LLM can enforce stronger groundedness for $\bar{y}$ while employing different skills to generate both outputs.

3.2 Fine-tuning Strategies

For each generation pipeline, we first conduct supervised fine-tuning on the training set of the experimented dataset. The reference output y is provided by the dataset, and we obtain $\bar{y}$ by removing all citation notations from y. During fine-tuning, the loss is computed across all output tokens for each model. Based on the supervised fine-tuned models, we collect preference data to further enhance them with preference-based optimization methods.

Preference Data Sampling. To collect preference data, the common practice involves sampling outputs from supervised fine-tuned models and annotating them using human efforts or automatic evaluators (Stiennon et al., 2020; Lee et al., 2024). For cost-effective data collection, it is critical to produce and select outputs that are more beneficial for model enhancement to be annotated. To this end, our paper investigates the effect of using data with varying degrees of diversity in content and citations. Specifically, for each training sample, we generate outputs using the supervised fine-tuned decomposed generation pipeline, where multiple intermediate outputs are sampled from the content generation module using different random seeds. Subsequently, different citations are inserted into each intermediate output by the citation generation module, also using different random seeds. For fair comparisons, the number of final sampled outputs across preference datasets created with different random seed combinations is kept constant, simulating a *fixed annotation budget*. Finally, each sampled output is assigned a content quality score, a citation quality score, and a combined overall quality score using the automatic evaluation metrics detailed in §4.

Preference-based Optimization. Given the labeled preference dataset, we consider continuing fine-tuning each generation pipeline with sampled outputs that have the best quality score, which resembles fine-tuning with data created by rejection sampling (Nakano et al., 2022).

For direct generation, we fine-tune the model using y^o,the sampled output with the highest overall quality score. For decomposed generation, we separately fine-tune the content and citation generation models. The content generation model is trained on $\bar{y}^{con}$, which is the sampled output with the highest content quality score after removing citations. The citation generation model is trained on y^{cit}, which represents the sampled output with the highest citation quality score. The training approach differs for the joint generation pipeline. Instead of computing the loss across all output tokens as in direct and decomposed generation, we employ a selective loss computation strategy. When training with $[\bar{y}^{con}; y^{con}]$ to enhance content generation, we minimize the loss only for tokens in $\bar{y}^{con}$ while ignoring the loss for tokens in y^{con}. Similarly, when improving citation generation with $[\bar{y}^{cit}; y^{cit}]$, we compute the loss only for tokens in y^{cit} while ignoring those in $\bar{y}^{cit}$.

Beyond fine-tuning with top-ranked outputs alone, we explore learning from contrastive pairs using direct preference optimization (DPO) (Rafailov et al., 2023). Given pairs of positive and negative samples constructed from sampled outputs, DPO increases the difference between the generation probabilities of pairs of positive and negative samples, promoting the generation of positive samples while discouraging negative ones. To ensure stable model optimization, DPO additionally uses generation probabilities from a reference model as baselines.[1]

For paired sampled outputs, we determine positive and negative samples by comparing their quality scores. Direct generation uses overall quality scores for comparisons, while decomposed and joint generation use content and citation quality scores for their respective optimization tasks. Similar to fine-tuning with rejection sampling data, for joint generation, we ignore the loss over tokens that are irrelevant to the task being optimized. To maintain a reasonable computational cost, each sampled output is included in only one pair, ensuring that all sampled outputs are covered while keeping the size of the fine-tuning samples manageable. Compared to rejection sampling, where models learn to imitate the most preferred output, DPO teaches models to differentiate between negative and positive outputs, aiming to avoid the generation of negative outputs.

3.3 Inference-time Compute

In addition to training-time techniques, we evaluate the effectiveness of scoring and reranking during inference. Specifically, an LLM-based scorer f_{eval} assesses a candidate output y' and produces two scores: $r_{y',a}$ and $r_{y',c}$. These scores, ranging from 1 to 5 on a Likert scale, measure the quality of the answers and citations, respectively. The scoring process can be formally expressed as $[r_{y',a}, r_{y',c}] = f_{eval}(y', q, \mathcal{D})$. To train the scorer, we partition our preference data's content quality and citation quality scores into 5 equally-sized bins. Each data point receives a Likert score based on its bin assignment.

During test time, we generate multiple outputs from each pipeline using different random seeds. The scorer then reranks these outputs to select the one that maximizes the sum of quality scores, expressed as: $y = \arg\max_{y' \in \mathcal{Y}}(r_{y',a} + r_{y',c})$, where $\mathcal{Y}$ represents the set of generated outputs for reranking.

4 Experiment Setups

Datasets. We conduct experiments on SciFi, a citation-rich dataset featuring subsentence-level citations sourced from Wikipedia (Cao and Wang, 2024). The training and test sets consist of $4{,}000$ and $1{,}000$ samples, respectively. For preference data collection, we sample model outputs on the training set of SciFi.

To evaluate generalizability, we further test each generation pipeline on the ALCE dataset (Gao et al., 2023b). ALCE comprises three subsets of knowledge-intensive question-answering samples, each paired with retrieved text passages that serve as candidate cited sources. We select the ASQA and ELI5 subsets, which feature questions with natural language responses. These subsets contain 948 and $1{,}000$ samples, respectively.

Evaluation Metrics. We evaluate citation quality by assessing the entailment level between each output statement and its corresponding cited source, in line with previous research (Rashkin et al., 2023). To decompose each model output into independent statements, we prompt Llama-3.1-8b (Grattafiori et al., 2024) with in-context examples. The cited documents, indicated by square brackets enclosing their indices, are then assigned to the output statements based on the heuristic rules outlined in prior work (Cao and Wang, 2024). Finally, we use an off-the-shelf NLI model (Honovich et al., 2022) to estimate the entailment level between output statements and their corresponding cited sources. Details of the evaluation metrics are provided in Appendix A.

The evaluation of content quality differs across datasets. For SciFi, we calculate the precision of statements by averaging the scores of the generated

[1] The supervised fine-tuned models serve as reference models in this paper.

statements entailing the reference, and the recall of statements by averaging scores of the reference statements entailing the generated output. The overall content quality is then determined by calculating the F1 score based on the precision and recall. For ALCE, we follow Gao et al. (2023b) and compute the recall of answer words and statements as the measure of content quality.

Additionally, we consider combining the two quality metrics into a single metric for SciFi. Specifically, when calculating the precision of the generated statements in the content quality metric, we adjust the entailment level between each output statement and the reference by multiplying it with the entailment level between the output statement and its corresponding cited source.

Model Setups and Comparisons. We conduct experiments with four open-source LLMs containing around 10B parameters: Llama-3.1-8B (Grattafiori et al., 2024), Mistral-Nemo (12B) (AI, 2024), Phi-3.5-mini (4B) (Abdin et al., 2024), and Qwen-2.5-7B (Team, 2024). For all models, we take their variants that have been aligned with human feedback.

For preference-based optimization, we consistently sample 8 outputs per training instance across all configurations for data collection, yielding 32,000 samples in total. Four configurations are considered for allocating the sampling budget. In each configuration, we generate 1, 2, 4, or 8 outputs during the citation generation step, corresponding to 8, 4, 2, or 1 intermediate outputs from the content generation step, respectively. Due to the high computational cost, we experiment with these configurations using only Llama-3.1-8B and apply the best-performing configuration to other LLMs.

In addition to the generation pipelines described in §3, we include an in-context learning (ICL) setup that performs direct generation by prompting the backbone LLMs with instructions and two demonstrations.

Training Details. We adopt LoRA (Hu et al., 2021) for model fine-tuning. The LoRA adapters are applied to all linear projection layers of each backbone LLM. We set the LoRA rank to 32 and use an α of 64. All systems are fine-tuned with supervised learning for 3 epochs on SciFi and are further fine-tuned with rejection sampling or DPO for 1 epoch. We use an effective batch size of 16 and a learning rate of 10^{-5}. For computing infrastructure, we use 4 A40 GPU, each with 48GB of memory during model training. During inference, we use a single A40 GPU. The average training time of each system is 10 hours for supervised fine-tuning, and 10 hours for further fine-tuning with preference-based optimization.

Pipeline	Content	Citation	Combined
Llama-3.1-8B			
Direct	**21.80**	**71.82**	**18.56**
Decomposed	21.77	41.61	15.13
Joint	21.07	64.59	16.60
Mistral-Nemo			
Direct	**23.08**	**72.02**	**19.25**
Decomposed	22.86	60.06	18.20
Joint	22.75	61.49	17.81
Qwen-2.5-7B			
Direct	21.04	**57.69**	**15.13**
Decomposed	**21.64**	42.55	14.91
Joint	19.22	44.61	14.18
Phi-3.5-Mini			
Direct	16.59	**43.27**	12.07
Decomposed	**17.00**	37.04	11.32
Joint	16.93	41.61	**12.12**

Table 1: Performance of different generation pipelines on SciFi. Results of the best-performing fine-tuning methods are reported. For each metric, the best result for each backbone LLMs is **bolded**.

5 Results

5.1 Main Results

We first compare the performance of different generation pipelines, as shown in Table 1. **Direct generation achieves better or comparable combined quality compared to the other pipelines** across all four backbone LLMs. Despite dedicate fine-tuning for each subtask, decomposed generation consistently produces citations of the lowest quality, as the content generation stage lacks awareness of the citation task's groundedness requirements. While joint optimization of content and citation generation enhances citation quality, this approach remains less effective than direct generation. We believe that direct generation benefits from its closer alignment with the pre-training text formats, as LLM pre-training increasingly emphasizes output verifiability, which is also evidenced by the performance improvements observed in newer generation models compared to older ones (results of Llama-2-7B and Llama-3-8B are in Table 6 of Appendix B).

Figure 4 presents the results for various fine-tuning strategies employed on different generation pipelines. **Systems fine-tuned with DPO consistently outperform others** across different back-

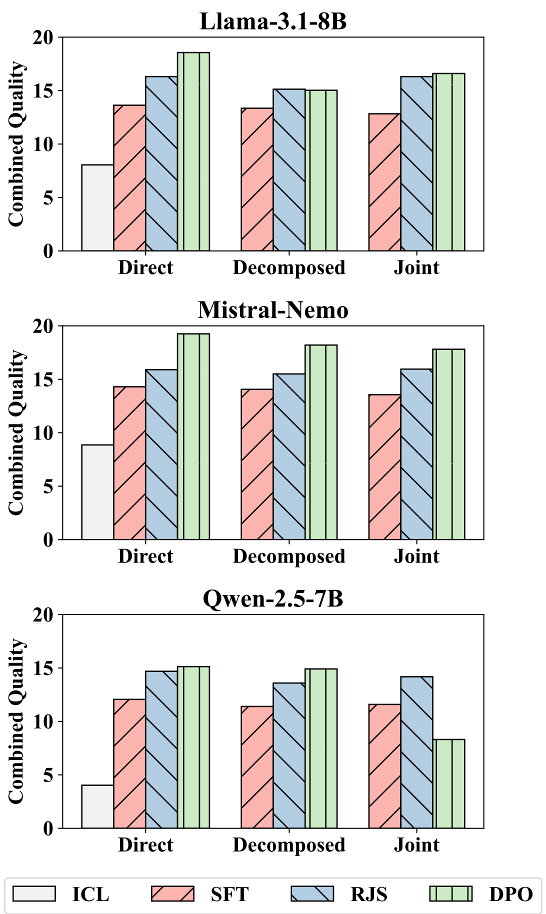

Figure 4: Performance of generation pipelines fine-tuned with different methods on SCIFI. ICL: in-context learning; SFT: vanilla supervised fine-tuning; RJS: supervised with rejection sampling data. Detailed results are in Appendix B.

Configuration	Content	Citation	Combined
Direct Generation + RJS			
(Gen × 1) × (Cite × 8)	21.71	44.07	15.74
(Gen × 2) × (Cite × 4)	21.84	44.73	15.97
(Gen × 4) × (Cite × 2)	21.69	45.25	15.85
(Gen × 8) × (Cite × 1)	**22.23**	**45.90**	**16.31**
Direct Generation + DPO			
(Gen × 1) × (Cite × 8)	16.09	63.99	13.30
(Gen × 2) × (Cite × 4)	21.08	**76.16**	18.17
(Gen × 4) × (Cite × 2)	**21.80**	71.82	**18.56**
(Gen × 8) × (Cite × 1)	20.96	50.65	12.55
Decomposed Generation + RJS			
(Gen × 1) × (Cite × 8)	16.67	45.16	12.88
(Gen × 2) × (Cite × 4)	18.22	47.14	14.17
(Gen × 4) × (Cite × 2)	18.69	**48.87**	14.53
(Gen × 8) × (Cite × 1)	**21.77**	41.61	**15.13**
Decomposed Generation + DPO			
(Gen × 1) × (Cite × 8)	19.77	40.94	14.89
(Gen × 2) × (Cite × 4)	19.12	52.79	**15.03**
(Gen × 4) × (Cite × 2)	13.99	**59.53**	11.19
(Gen × 8) × (Cite × 1)	**20.29**	49.10	13.94
Joint Generation + RJS			
(Gen × 1) × (Cite × 8)	21.33	43.67	15.50
(Gen × 2) × (Cite × 4)	21.83	44.62	15.88
(Gen × 4) × (Cite × 2)	21.46	45.46	15.76
(Gen × 8) × (Cite × 1)	**22.23**	**45.49**	**16.31**
Joint Generation + DPO			
(Gen × 1) × (Cite × 8)	20.69	62.51	16.40
(Gen × 2) × (Cite × 4)	**21.07**	**64.59**	**16.60**
(Gen × 4) × (Cite × 2)	19.53	56.48	14.74
(Gen × 8) × (Cite × 1)	18.08	19.93	5.77

Table 2: Performance of generation pipelines on SCIFI with different configurations for obtaining sampled outputs. All the systems are based on Llama-3.1-8B. For each generation pipeline and fine-tuning method, the best data configuration is **bolded**. For both optimization methods, using more than 1 intermediate output to generate final outputs with citations leads to better citation quality. The best configuration for each optimization method is applied to other backbone models in the main experiments.

bone LLMs and generation pipelines, with two exceptions: decomposed generation with Llama-3.1 and joint generation with Qwen-2.5. Unlike supervised fine-tuning with rejection sampling data that only learns from the best sampled outputs, DPO leverage contrastive pairs of sampled outputs, which effectively guides LLMs towards the desired behaviors by training LLMs to distinguish between higher and lower quality outputs. Notably, all fine-tuning methods significantly outperform in-context learning, highlighting the effectiveness of fine-tuning for open-source models.

5.2 Analysis of Preference Data Configurations

Different configurations for collecting preference data within the sampling budget are compared in Table 2. The notation "(Gen × 4) × (Cite × 2)" indicates that the content generation model produces 4 intermediate outputs, and the citation generation model creates 2 outputs with citations for each intermediate output, resulting in 8 total final outputs. Our analysis reveals that maintaining **sufficient content diversity** among these sampled outputs is crucial. Configurations that allocate the entire budget to generating outputs with different citations do not achieve better citation quality compared to other configurations that allocate more budget for content diversity. For instance, after fine-tuning direct generation with DPO using 8 outputs comprising different citations and the same content, the system performs 17% worse than using outputs based on two distinct intermediate outputs.

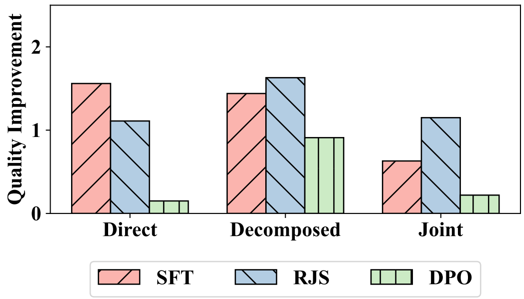

Figure 5: Changes of combined quality after applying over-generation and reranking to Llama-3.1-8B pipelines on SciFi. For each test sample, four outputs are generated and reranked. All systems benefit from inference-time compute, though the improvement is not as significant as fine-tuning.

Fine-tuning	ASQA Cont.	ASQA Cit.	ELI5 Cont.	ELI5 Cit.
ICL	**42.14**	19.78	**14.57**	16.98
SFT	35.84	35.63	11.89	20.89
RS	36.65	49.91	12.06	31.23
DPO	39.43	**62.00**	13.86	**51.26**

Table 3: Performance of direction generation that is based on Llama-3.1-8B and fine-tuned on SciFi and tested on the ASQA and ELI5 subsets of ALCE. Systems optimized with DPO again achieves the best citation quality, and the trend of improvement in citation quality over the in-context learning baseline is similar to the one on SciFi. However, compared to in-context learning, the content quality would drop.

5.3 Effectiveness of Inference-Time Compute

We apply the over-generation and reranking technique on top of verifiable generation systems that are based on Llama-3.1-8B. During over-generation, we sample from each system with 4 different random seeds. For decomposed generation, we use the same random seed for the content generation model and the citation generation model. As shown in Figure 5, the scoring and reranking technique can **consistently enhance the quality of the final output for all systems**. Compared to systems fine-tuned with other methods, systems fine-tuned with DPO observe smaller improvement after reranking. Considering the cost of over-generating outputs and training the reranking model, employing inference-time compute methods **might not be cost-effective for the top models**.

5.4 Generalizability

Finally, we evaluate the generaliablity of direct generation that are based on Llama-3.1-8B. The strong citation quality of systems fine-tuned with DPO well generalizes to test samples that do not come from the dataset used for model training. Overall, the trend in citation quality remains consistent with the results on SciFi, suggesting that **the citation capability acquired through fine-tuning are robust across datasets**. However, fine-tuning on out-of-domain data can lead to a decline in content quality when applied to in-domain data, as observed on both ASQA and ELI5. We believe this is due to the variation of focus of output content across different domains.

6 Conclusions

We conduct an analysis of design choices in the development of verifiable generation systems, including generation pipelines and optimization methods. Three generation pipelines are investigated: direct generation that outputs responses with citations in one pass; decomposed generation that connects a content generator with a citation generator to produce outputs in two steps; joint generation that combines the aforementioned pipelines. We conduct supervised fine-tuning for these generation pipelines and additionally apply preference-based optimization including further supervised fine-tuning with rejection sampling data and direct preference optimization (DPO). Moreover, we examine the effect of content and citation diversity on fine-tuned model performance. Besides training-time techniques, we also study an inference-time technique—over-generation and reranking. Our experiments find that (1) direct generation yields the best overall quality; (2) DPO is the best fine-tuning method; (3) maintaining sufficient content diversity is crucial for preference-based optimization; (3) reranking of over-generated samples can benefit all verifiable generation systems but cost-effectiveness might be low; (4) LLMs' ability to cite supporting sources is robust across datasets. We hope our findings can guide further development of verifiable generation systems with open-source LLMs.

Acknowledgments

This work is supported in part by the National Science Foundation through grant IIS-2046016. Shuyang Cao is supported by a Bloomberg Data

Science Ph.D. Fellowship. We thank ARR reviewers for their feedback.

7 Limitations and Potential Risks

Limitations. Our work conducts a wide range of experiments, but there remain design choices that are not investigated, due to the complexity of verifiable generation systems. For example, the process of handling the pool of candidate cited sources could benefit from more sophisticated strategies, which might include multi-turn processing of cited sources or the construction of dense representations.

The datasets employed in our experiments provide a fixed set of candidate sources with well-formatted content. However, in real-world scenarios, candidate sources are dynamically retrieved from online search engines. The use of online search engines can introduce a greater diversity of candidate sources, resulting in domain and style shifts that could impact model behavior and task performance unpredictably.

Potential Risks. Echoing the limitations mentioned, our results are based on a pool of trustworthy sources, such as Wikipedia articles. However, when verifiable generation systems are deployed in practical settings, they may encounter sources with varying degrees of reliability. This creates a risk of propagating misinformation if the system inadvertently relies on less credible sources. Furthermore, dynamically retrieved data could include biased or malicious content, potentially leading to harmful consequences. Therefore, our study reveals best practices of verifiable generation systems in controlled conditions, the robustness of them in uncontrolled environments requires further investigation. Developers should equip their systems with additional content filters to ensure healthy outputs.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Mistral AI. 2024. Mistral nemo.

Shuyang Cao and Lu Wang. 2024. Verifiable generation with subsentence-level fine-grained citations. *Preprint*, arXiv:2406.06125.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023. Complex claim verification with evidence retrieved in the wild. *Preprint*, arXiv:2305.11859.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste

Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich

Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Chengyu Huang, Zeqiu Wu, Yushi Hu, and Wenya Wang. 2024. Training language models to generate text with citations via fine-grained rewards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2926–2949, Bangkok, Thailand. Association for Computational Linguistics.

Jie Huang and Kevin Chen-Chuan Chang. 2024. Citation: A key to building responsible and accountable large language models. *Preprint*, arXiv:2307.02185.

Siqing Huo, Negar Arabzadeh, and Charles Clarke. 2023. Retrieving supporting evidence for generative question answering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP '23. ACM.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. WiCE: Real-world entailment for claims in Wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583, Singapore. Association for Computational Linguistics.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *Preprint*, arXiv:2309.00267.

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes. *Preprint*, arXiv:2203.11147.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Hao Sun, Hengyi Cai, Bo Wang, Yingyan Hou, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2024. Towards verifiable text generation with evolving memory and self-reflection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8211–8227, Miami, Florida, USA. Association for Computational Linguistics.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan

Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhra-

jit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu,

John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *Preprint*, arXiv:2305.10425.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Evaluation Metrics

Citation Quality. Given an output statement s_i and its corresponding cited document d_{s_i}, we use a T5-based NLI model[2] to calculate the score of how d_{s_i} support s_i as the citation quality measure. We take the probability of the NLI model predicting "entail" as the score. As the length of d_{s_i} might exceed the maximum input length of the NLI model and the NLI model is trained with shorter sequences, following (Kamoi et al., 2023), we split the document into chunks of 256 tokens $\{d_{s_i}^1, \ldots, d_{s_i}^M\}$ and take the maximum entailment score between s_i and chunks of d_{s_i} as the entailment score between s_i and d_{s_i}:

$$u_{cit}(s_i) = \max_{1 \leq m \leq M} ent(s_i, d_{s_i}^m) \tag{1}$$

where $u_{cit}(s_i)$ denotes citation quality score of s_i. The citation quality score of a system output is then computed by averaging $u_{cit}(s_i)$ for all statements in the output.

Content Quality. We calculate the precision of system generated statements as $\frac{1}{N}\sum_i ent(s_i, \hat{y})$, where $\hat{y}$ is the reference output and N is the total number of statements in the system output. Similarly, the recall of reference statement is calculated as $\frac{1}{\hat{N}} ent(\hat{s}_i, y)$, where y is the system output, $\hat{s}_i$ is a reference statement, and $\hat{N}$ is the total number of statements in the reference output. We take the harmonic mean of the precision and recall as the content quality of a system output. The entailment is calculated between a statement and a full text output following (Gao et al., 2023b).

Combined Quality. The combined quality is similar to the content quality, except that we change the precision calculation to $\frac{1}{N}\sum_i ent(s_i, \hat{y}) \times u_{cit}(s_i)$.

Citation Mapping. To determine the cited document for each statement given the raw system output, we use the assignment rule as in (Cao and Wang, 2024). After decomposing the system output into individual statements, each statement is mapped back to a segment in the original system output by prompting a Llama-3.1-8B model with in-context examples adapted from (Min et al., 2023; Kamoi et al., 2023). For an output statement, the generated citation that is closest to the end of its corresponding segment is taken as its cited source.

B Additional Results

Fine-tuning Strategies. In Table 4 and 5, we provide detailed results of generation pipelines paired with different fine-tuning strategies. Using DPO achieves the best performance across different pipelines.

[2] https://huggingface.co/google/t5_xxl_truenli_mixture

Pipeline	Fine-tuning	Content	Citation	Combined
Llama-3.1-8B				
Direct	ICL	16.58	32.63	8.05
	SFT	19.60	36.99	13.63
	RJS	**22.23**	45.90	16.31
	DPO	21.80	**71.82**	**18.56**
Decomposed	SFT	19.71	35.22	13.35
	RJS	**21.77**	41.61	**15.13**
	DPO	19.12	**52.79**	15.03
Joint	SFT	19.21	35.53	12.83
	RJS	**22.23**	45.49	16.31
	DPO	21.07	**64.59**	**16.60**
Mistral-Nemo (12B)				
Direct	ICL	19.37	31.64	8.86
	SFT	21.05	36.55	14.30
	RJS	21.46	47.45	15.90
	DPO	**23.08**	**72.02**	**19.25**
Decomposed	SFT	20.92	36.15	14.06
	RJS	22.02	43.25	15.50
	DPO	**22.86**	**60.06**	**18.20**
Joint	SFT	20.48	35.28	13.56
	RJS	21.66	46.05	15.95
	DPO	**22.75**	**61.49**	**17.81**
Qwen-2.5-7B				
Direct	ICL	15.68	17.78	4.03
	SFT	17.24	35.64	12.06
	RJS	19.65	45.81	14.69
	DPO	**21.04**	**57.69**	**15.13**
Decomposed	SFT	17.20	33.86	11.40
	RJS	19.34	41.25	13.59
	DPO	**21.64**	**42.55**	**14.91**
Joint	SFT	16.82	35.59	11.59
	RJS	19.22	**44.61**	**14.18**
	DPO	**20.58**	32.94	8.32

Table 4: Performance of generation pipelines fine-tuned with different methods on SciFi. ICL: in-context learning; SFT: vanilla supervised fine-tuning; RJS: supervised with rejection sampling data. For each metric and pipeline, the best fine-tuning method is **bolded**.

Older Models. We report results based on different Llama models in Table 6. The latest Llama model obtains significantly better performance than its older generations, suggesting the increased emphasis of verifiability during model pre-training and alignment. We also observe a decrease in the effectiveness of joint generation, which might be due to the increase number of pre-training samples that contain citations.

Pipeline	Fine-tuning	Content	Citation	Combined
Phi-3.5-Mini (4B)				
Direct	ICL	5.43	2.90	0.83
	SFT	14.82	33.39	9.81
	RJS	16.59	43.27	12.07
	DPO	**18.48**	**49.70**	13.02
Decomposed	SFT	14.60	32.28	9.60
	RJS	**17.00**	37.04	11.32
	DPO	16.52	**41.96**	**11.83**
Joint	SFT	14.50	31.30	9.07
	RJS	16.93	41.61	12.12
	DPO	**17.97**	**45.32**	**13.58**

Table 5: Continuation of Table 4.

Pipeline	Content	Citation	Combined
Llama-2-7B			
Direct	**13.98**	23.48	6.79
Decomposed	13.23	30.17	9.68
Joint	13.87	**36.71**	**10.49**
Llama-3-8B			
Direct	**17.58**	41.82	13.21
Decomposed	16.51	37.65	12.39
Joint	17.04	**43.56**	**13.42**
Llama-3.1-8B			
Direct	**21.80**	**71.82**	**18.56**
Decomposed	21.77	41.61	15.13
Joint	21.07	64.59	16.60

Table 6: Performance of different generation pipelines on SciFi, based on Llama models of various generations. For each metric, the best result for each backbone LLMs is **bolded**.

C Implementations

C.1 Datasets

We obtain the SciFi dataset[3] and the ALCE dataset[4] from their authors' official releases. They are with CC-By-4.0 and MIT licenses, respectively.

C.2 Models

All the backbone LLMs are retrieved from the Huggingface Hub:

- Llama-3.1-7B: `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`
- Mistral-Nemo: `https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407`
- Phi-3.5-Mini: `https://huggingface.co/microsoft/Phi-3.5-mini-instruct`

[3] `https://shuyangcao.github.io/projects/subsentence_citation/`
[4] `https://github.com/princeton-nlp/ALCE`

- Qwen-2.5-7B: `https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`

C.3 Training

We use LLaMA-Factory (Zheng et al., 2024) for the implementations of model trainers including the DPO optimization algorithm.

C.4 Usage of AIi Assistant

We use Copilot for implementation of experiment code and analysis code. ChatGPT is used for refining the grammar and fixing typo during writing.

C.5 Prompt Templates

The instructions and prompts we use for each generation pipeline are shown in Table 7–10.

Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant) and cite them properly. You are provided summaries of the search results, rather than the original search results. Use an unbiased and journalistic tone. Always cite after the completion of each individual fact in the answer. Facts might be completed in the middle of a sentence.

Question: `{query}`

Document [1] (Title: `{document1_title}`)
`{document1_text}`

...

Document [N] (Title: `{documentN_title}`)
`{documentN_text}`

Answer: `{output_with_citation}`

Table 7: Instruction and prompt for intrinsic generation.

Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant). You are provided summaries of the search results, rather than the original search results. Use an unbiased and journalistic tone.

Question: `{query}`

Document [1] (Title: `{document1_title}`)
`{document1_text}`

...

Document [N] (Title: `{documentN_title}`)
`{documentN_text}`

Answer: `{content_generation_output}`

Table 8: Instruction and prompt for content generation in modular generation.

Instruction: Support facts in the given statement by citing the provided search results (some of which might be irrelevant). You are provided summaries of the search results, rather than the original search results. Cite after the completion of each individual fact in the answer. Facts might be completed in the middle of a sentence.
 	
Question: `{query}`
 	
Document [1] (Title: `{document1_title}`)
`{document1_text}`
 	
...
 	
Document [N] (Title: `{documentN_title}`)
`{documentN_text}`
 	
Statement: `{content_generation_output}`
 	
Statement with Citations: `{output_with_citation}`

Table 9: Instruction and prompt for citation generation in modular generation.

Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant) and cite them properly. You are provided summaries of the search results, rather than the original search results. Use an unbiased and journalistic tone. Always cite after the completion of each individual fact in the answer. Facts might be completed in the middle of a sentence.
 	
Question: `{query}`
 	
Document [1] (Title: `{document1_title}`)
`{document1_text}`
 	
...
 	
Document [N] (Title: `{documentN_title}`)
`{documentN_text}`
 	
Answer: `{output_without_citation}` | Answer with Citations: `{output_with_citation}`

Table 10: Instruction and prompt for intrinsic-modular generation.