# LLMs can learn self-restraint through iterative self-reflection

**Anonymous authors**
**Paper under double-blind review**

## Abstract

In order to be deployed safely, Large Language Models (LLMs) must be capable of dynamically adapting their behavior based on their level of knowledge and uncertainty associated with specific topics. This adaptive behavior, which we refer to as *self-restraint*, is non-trivial to teach since it depends on the internal knowledge of an LLM. By default, LLMs are trained to maximize the next token likelihood, which does not teach the model to modulate its answer based on its level of uncertainty. In order to learn self-restraint, we devise a *utility function* that can encourage the model to produce responses only when its level of confidence is above a user-specified target accuracy $\rho^*$. This utility function can be used to score generation of different length and abstention. To optimize this function, we introduce ReSearch, a process of "self-reflection" consisting of iterative self-prompting and self-evaluation. We use the ReSearch algorithm to generate synthetic data on which we finetune our models. ReSearch elegantly incorporates the ability to *abstain* by augmenting the samples generated by the model during the search procedure with an answer expressing abstention. Compared to their original versions, our resulting models generate fewer *hallucinations* overall at no additional inference cost, for both known and unknown topics, as the model learns to selectively restrain itself. In addition, we show that our iterative search is more efficient as a function of tokens than naive search. Finally, we show that by modifying the target accuracy $\rho^*$, our trained models exhibit different behaviors.

## 1 Introduction

In order for Large Language Models (LLMs) to become reliable tools, it is important for the models to be able to modulate their responses based on their internal knowledge. In cases where the models are queried about a topic that is not well supported by their internal knowledge, it is safer for the LLMs to provide a short answer or to even abstain from answering entirely, instead of providing an answer filled with inaccuracies (*hallucinations*). Unfortunately it is non-trivial to teach this behavior to LLMs since the optimal behavior depends on the model's internal knowledge (Goldberg, 2023).

There has been several successful attempts to improve the factuality of LLMs while maintaining their helpfulness using preferences (Tian et al., 2023a), question-and-answer-based rewriting (Dhuliawala et al., 2023), and decoding by contrasting predictions at different transformer layers (Chuang et al., 2024). While these methods have been shown to reduce hallucinations on average, they do not fulfill the desiderata of teaching LLMs to modulate the amount of information in their response, the level of detailedness of the information, and lastly to abstain entirely from responding to queries when that is appropriate.

In this work, we present ReSearch: an iterative self-reflection algorithm designed for synthetic data generation. The algorithm involves utilizing an LLM to iteratively generate samples, self-evaluate the samples' expected accuracy in a reference-free manner, and prompt itself to produce improved samples in the subsequent iteration. ReSearch produces a batch of scored samples to which we also add a phrase which expresses a refusal response. All these samples and the refusal phrase can then be scored via self-evaluation and be used with learning algorithms such as Direct Preference Optimization (DPO) (Rafailov et al., 2023). We show that LLMs trained on synthetic data generated by the ReSearch algorithm exhibit self-restraint, resulting in fewer hallucinations on both biography and historical event generation tasks without any need for external references. In addition, in Section 4.3, we show that our iterative search is more efficient as a

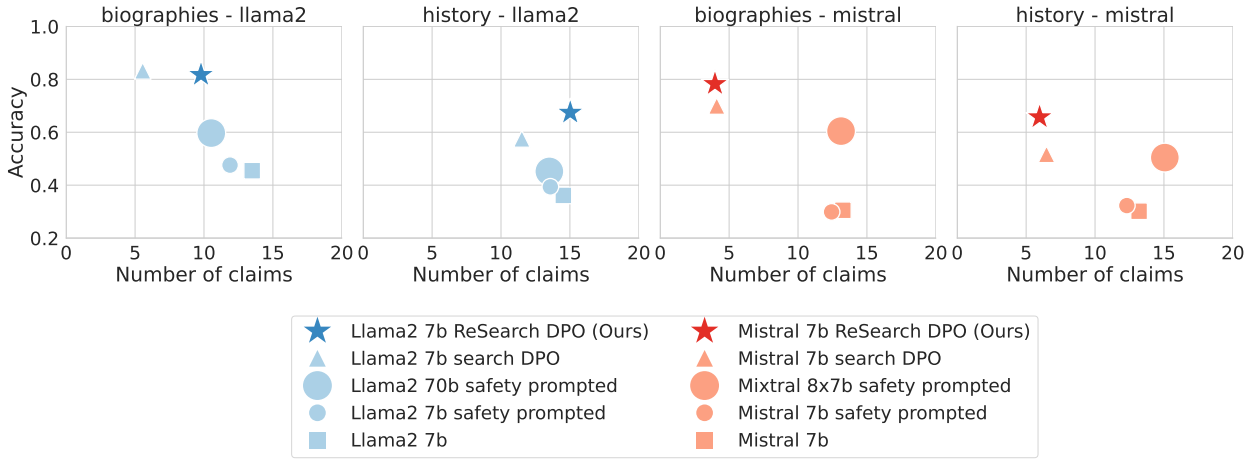



Figure 1: **Results overview.** Llama2 7b models trained on data synthetically generated by the ReSearch algorithm (★) outperforms all baselines. This includes Llama2 7B chat (■) and safety prompted (●), Llama2 70B chat (●), and search DPO (Tian et al., 2023a) (▲). Our models produce more claims that search DPO and Llama 70b and is at least as accurate. Mistral 7b models trained on data synthetically generated by the ReSearch algorithm (★) also outperforms all baselines. This includes Mistral 7B (■) and safety prompted (●), Mixtral 8x7B (●), and search DPO (Tian et al., 2023a) (▲). Our models are more accurate than every baseline and produce almost as many claims as the search DPO.

function of tokens than naive search. Finally, in Section 4.4, we show that by modifying the target accuracy $\rho^*$, our trained models exhibit different behaviors.

## 2 Background

**Expert iteration** (Anthony et al., 2017; Silver et al., 2017) enables the agent to self-improve through alternating phases of policy improvement and policy distillation. Initially, the agent gathers a dataset of demonstrations utilizing search and self-evaluation; these demonstrations surpass the quality of those the agent could have collected independently. Subsequently, the agent distills the dataset into its weights via supervised fine-tuning (SFT). In language, expert iteration has been used to improve translation (Gulcehre et al., 2023) and problem solving (Singh et al., 2023).

**Reward or utility function design** is challenging as we must represent our preferences into a single scalar to obtain the desired behavior from an agent (Sutton & Barto, 2018). Designing a utility function for long form factual content generation is particularly challenging as it requires us to balance between multiple objectives (Vamplew et al., 2021): the completeness of an answer (helpfulness) and its accuracy (harmlessness) (Bai et al., 2022). Furthermore, we want the utility function to encourage the agent to have self-restraint and to abstain from generating an answer if it is incapable of doing so without violating a safely constraint.

**Closed book factual generation.** LLMs are known to generate content that contradicts established world knowledge (Zhang et al., 2023), commonly referred to as "hallucination". DPO (Rafailov et al., 2023) has been shown to be effective at reducing hallucinations on short biographies (Tian et al., 2023a) by maximizing the average claim probability (as shown in Figure 3). However, it is unclear if the model learns to adjust its verbosity or learns to abstain from answering a query solely based on preference data, in which constraints cannot easily be encoded. In contrast to Tian et al. (2023a) which uses consistency-based methods to evaluate the model confidence in an answer, Zhang et al. (2024) trained the model to evaluate its confidence. Chain-of-verification (Dhuliawala et al., 2023) (CoVe) has been used to reduce hallucinations, but it does not lead to LLMs that learn to abstain from answering queries they are unfamiliar with. Similar to Cove, real-time verification and rectification (Kang et al., 2023) identify and rectify hallucinations in a step-wise manner. Manakul et al. (2023) used multiple generations to score the likelihood of a claim. A similar self-consistency metric was introduced by Huang et al. (2024).

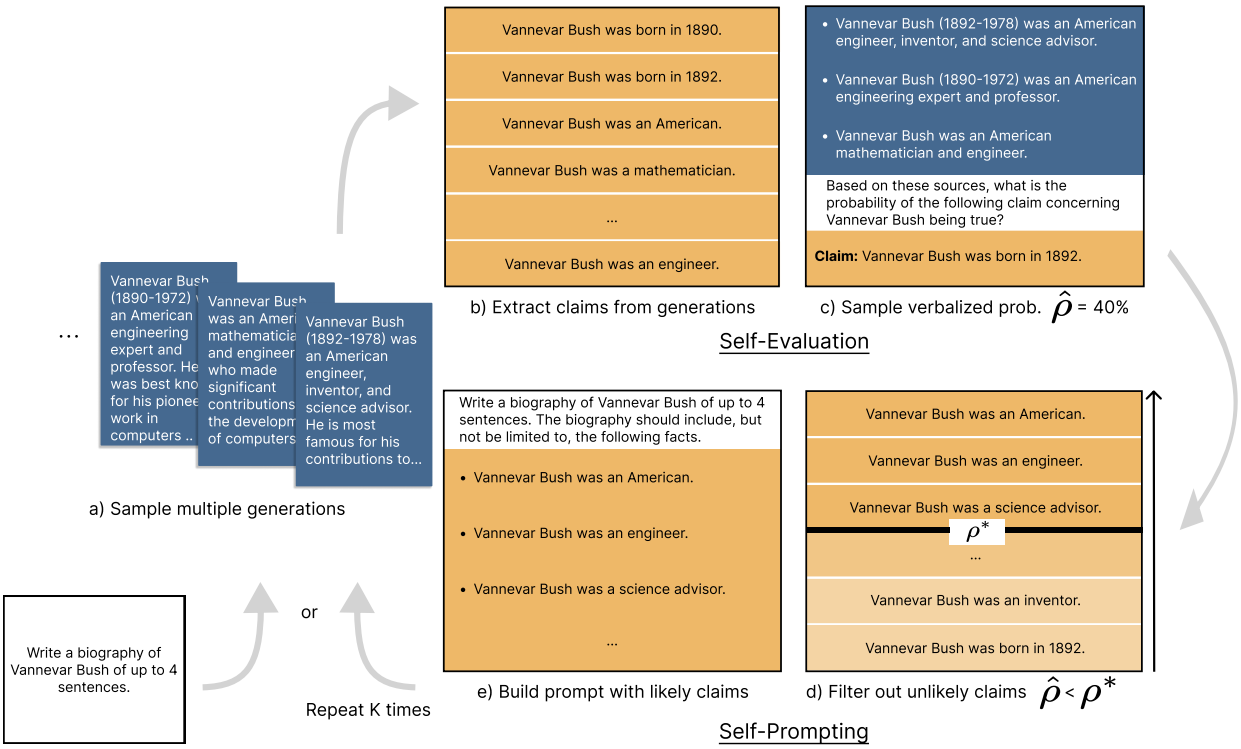



Figure 2: **Overview of the ReSearch algorithm.** ReSearch combines two components: 1) *Self-Evaluation* where the model evaluates the expected accuracy $\hat{\rho}$ of its generated claims based on their self-consistency with all the generations produced by the model, and 2) *Self-Prompting* where the model incorporates the claims more likely than $\rho^*$ into its prompt to improve its generations at the next iteration. Finally, the resulting generations produced by the ReSearch algorithm plus the phrase expressing abstention are self-evaluated, ranked, and returned as synthetic data that can be used as desired.

**Self-evaluation and calibration.** Manakul et al. (2023) showed that a model can detect its hallucinations by comparing a sampled generation with other sampled generations. (Tian et al., 2023b) showed that models that have been finetuned using RL via Human Feedback produced better calibrated prediction by expressing their uncertainty via verbalized probability. Kadavath et al. (2022) showed that a model can improve its ability to evaluate the correctness of an answer by conditioning its prediction on concurrent hypotheses. In language, self-evaluation has also been used for question-answering (Ren et al., 2023), using tools (Mekala et al., 2024), and alignment (Yuan et al., 2024).

**Question-answering** Yang et al. (2023) leverage the model uncertainty to abstain from answering questions it does not know. Ren et al. (2023) propose a scoring method based on self-evaluation. Kadavath et al. (2022) use multiple generations from a LLM to measure the uncertainty of the model. Concurrently, Mohri & Hashimoto (2024) use conformal prediction to remove incorrect claims generated by LLMs. Yang et al. (2023) train a model via SFT to abstain from answering when it cannot answer a question correctly. Self-prompting has been used for short form question-answering (Li et al., 2022). Selective prediction allows a classifier to abstain from answering low-confidence questions (Xin et al., 2021) or questions that unanswerable (Rajpurkar et al., 2018).

**Self-reflection** is the ability for an agent to verbally reflect on a task to improve its behavior on subsequent trials. Shinn et al. (2023) successfully used self-reflection to improve the reasoning and programming capabilities of state-of-the-art models. Madaan et al. (2023) introduced self-refine which is an agent providing feedback on its outputs in order to iteratively improve it. Similarly to our method, Ji et al. (2023) used self-reflection to improved a model factual accuracy on medical questions, but crucially their method do not handle abstention. In Asai et al. (2023), an LLM is trained via "reflection tokens" to introspect on the

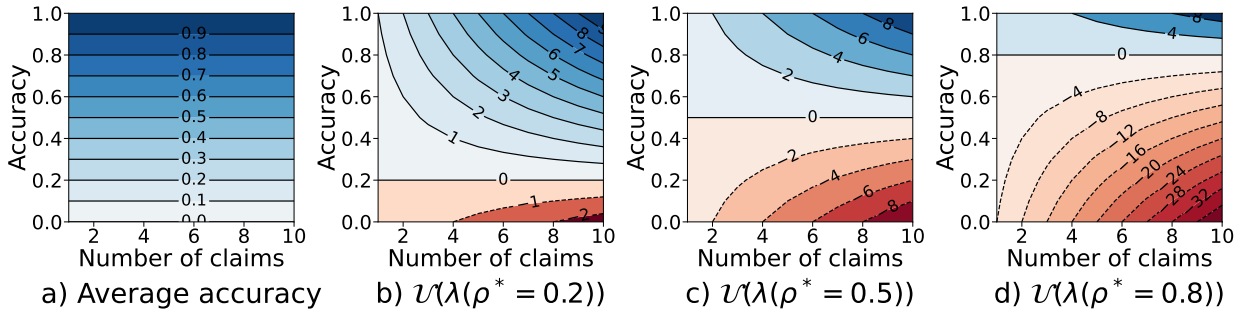



Figure 3: **Utility function contour plots.** First, in sub-figure a), we can observe (as done by multiple methods such as Tian et al. (2023a)) that using the average accuracy as a utility does not encourage more claims. In sub-figures b), c) and d), we observe that the utility function encourages the agent to produce as many claims as possible and to maximize the accuracy. It also elegantly ranks samples with different number of claims and accuracy. Furthermore, in subfigure b), we observe that the agent should abstain from a query and obtain a utility of 0 instead of a negative utility if it believes that less than target accuracy $\rho^* = 20\%$ of the claims in its best sample are true. Finally, we observe a similar pattern for c) and d), where the target accuracy $\rho^*$ is set to 50% and 80% respectively. We also observe the advantages of the utility function $\mathcal{U}$ over the average probability empirically in Figure 11.

relevance of retrieved passages to then synthesize a final improved response to a knowledge-seeking query. Although the reflective elements are similar to our work, this work involves an explicit retrieval step over a corpus of external documents, while our work operates solely on the model's internal knowledge. In Liu et al. (2024), authors systematically investigate the ability for language models to refrain from responding to queries that they are guaranteed to not know the answer to. They find that the majority of models, and especially open-source ones, are unable to consistently refuse invented queries.

## 3 Method

We are interested in a model that, given its internal knowledge, exhibits self-restraint, i.e., is 1) helpful by generating as many true claims as possible and 2) harmless by limiting the number of false claims it produces. As we show in the experiment section, this behavior is non-trivial to obtain via prompting. Similarly, building a dataset for this behavior to train an agent via supervised learning is difficult since it requires us to have access to the model's internal knowledge. Instead of building a dataset, we augment the model with search and self-evaluation to generate a synthetic dataset specifically tailored to the knowledge of the model. In order to evaluate the samples generated by the search procedure, we design a utility function that can be approximated by the agent and that encourages the agent to maximize the number of true claims and minimize the number of false claims. We introduce ReSearch, an iterative search algorithm based on self-evaluation and self-prompting (an overview is provided in Figure 2), to maximize our utility function.

### 3.1 Utility function design

**Limitation of only using accuracy as an utility function.** Long form content generation is difficult to measure with a single metric as there is a trade-off between completeness and accuracy (Xu et al., 2023). Previous works, such as FActScore (Min et al., 2023) and Finetuning for factuality (Tian et al., 2023a), have been solely using the average accuracy of generations to rank generations. Where more accurate generation are given a higher score than less accurate ones. While this works well to score generations of the same length, it cannot readily be used to score generations of different lengths or abstention, as it does not take into account the number of claims in a generation (see Figure 3 a)). For example, the model would obtain the same score by providing a single true claim than the score it would obtain by providing multiple true

claims. Therefore, we introduce an utility function that take into account the number of claims produced by the model and the accuracy of the claims.

**Designing an utility function that can handle abstention and generations of different length.** Similarly to the FActScore metric (accuracy), we assume that 1) a generation $y$ can be broken up into a set of atomic claims $\{c\}_{i=1}^{N}$ as shown in Figure 2, and 2) that each claim can be judged by an oracle $\mathcal{T}$ as being factual or not. Similarly to FActScore the oracle will be a larger LM that has access to Wikpedia while the generating model does not. For query $x$ and answer $y$, our utility function $\mathcal{U}(x, y, \lambda)$ gives a utility of 1 for every true claim in $y$ and a negative utility of $\lambda$ for every false claim. Specifically, we define our utility function as:

$$\mathcal{U}(x, y, \lambda) = \sum_{c \in \mathrm{CS}(y)} u(\mathcal{T}(x, c), \lambda), \text{ where} \tag{1}$$

$$u(\mathcal{T}(x, c), \lambda) = \begin{cases} 1 \text{ if } \mathcal{T}(x, c) = 1 \\ -\lambda \text{ if } \mathcal{T}(x, c) = 0, \end{cases}$$

and claim splitter $\mathrm{CS}(y) \rightarrow \{c_i\}_{i=1}^{N}$ is extracting atomic claims from generation $y$. While it is not immediately obvious how to select $\lambda$. We can instead set a target accuracy $\rho^*$ for which the model would obtain a utility of 0, and then select $\lambda$ accordingly:

$$\mathcal{U}(x, y, \lambda) = \rho^* N - \lambda(1 - \rho^*)N = 0, \tag{2}$$

and we can easily solve for $\lambda(\rho^*) = \frac{\rho^*}{1-\rho^*}$. In other words, the model will receive a positive (negative) utility if it produces generation with accuracy higher (lower) than $\rho^*$. Therefore, our utility function encourages the model to abstain if the expected accuracy of its answer is below the threshold $\rho^*$ and obtain a utility of 0. While the utility function is linear in true and false claims, it is non-linear in the number of claims and accuracy. We observe in Figure 3 that increasing the value of the target accuracy $\rho^*$ changes the threshold for which the utility function is positive and therefore when the model should answer a query and not abstain.

### 3.2 ReSearch: an iterative search algorithm for LLMs

We introduce ReSearch: an iterative self-reflection algorithm designed for synthetic data generation which can be used to improve the model's generation factuality and ability to self-restraint. For a given query $x$, the algorithm iterates through three phases which can be observed in Figure 2: 1) **generation**: sample multiple possible answers to a query, 2) **self-evaluation**: evaluate the claims contained in these generations via self-consistence 3) **self-prompting**: prompt itself with likely claims. Finally, once multiple iterations have been performed, the algorithm returns a synthetic dataset: $(x, Y, U)$, where $Y$ is a set of all the answers produced by the model and an additional answer expressing refusal and $U$ is their respective utility (this phase is not depicted in the figure).

**1) Generation.** At each iteration k the model first generates multiple generations $Y_k = \{y_k^j\}_{j=1}^{J}$ where each sample $y$ is generated conditionally on a prompt. At each generation the prompt is either the query generated by the user at the first iteration ($\mathcal{P}_{\text{write}}$) or a query generated by the LLM via self-reflection ($\mathcal{P}_{\text{rewrite}}$) for subsequent iterations. We expect the factuality of the generations to improve as $\mathcal{P}_{\text{rewrite}}$ is refined via self-reflection at each iteration.

**2) Self-evaluation.** In order to evaluate the expected factuality of its generations, the model first extracts the claims $c$ for every generation $y_k^j$ in the generations $Y_k$ produced at the current iteration $k$ (Figure 2 b)). Then the model evaluates the probability of each claim being true based on the claim self-consistency with a random subset of sentences sampled from the generations produced by the model up to this point using the prompt template $\mathcal{P}_{\text{eval}}$ (Figure 2 c)). The intuition is that the claims that are likely to be true will be consistent with multiple generations, while the claims that are less likely to be true will be inconsistent. In other words, we will ask the model to generate an integer between 0 and 100 to represent the level of agreement between a claim and independently (condition on the prompt) generated samples. We then average the probabilities produced by the model over multiple subsets of sentences. Specifically, the probability of a

**Algorithm 1** ReSearch algorithm.

**Require:** Context dataset $\mathcal{Q} \leftarrow \{x_i\}_{i=1}^N$
**Require:** Policy $\pi_\theta : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{Y})$
**Require:** Factuality utility model $\hat{\mathcal{F}}(x, y) \rightarrow \mathbb{R}$
**Require:** Claim likelihood $p(\mathcal{T}|x, c, Y) \rightarrow [0, 1]$
**Require:** Factuality threshold $\rho \in [0, 1]$
**Require:** Claim splitter $\mathrm{CS}(y)$
1: On-policy dataset $\mathcal{D} \leftarrow \emptyset$
2: **for** $x \in \mathcal{Q}$ **do**
3: # Initialize the generation with abstaining and utility of 0
4: $Y \leftarrow \{(\emptyset, 0)\}$
5: # sample $J$ initial generations.
6: $\{y^j \sim \pi_\theta(\cdot|\mathcal{P}_{\mathrm{write}}(x))\}_{j=1}^J$
7: **while** stopping_criterion **do**
8: # Collect generations and expected utility
9: $Y \leftarrow Y \bigcup \{(y^j, \hat{\mathcal{F}}(x, y^j))\}_{j=1}^J$
10: # Extract most likely claims from all the generations
11: $\mathcal{C} \leftarrow \{c \in \mathrm{CS}(Y) \mid p(\mathcal{T}|x, c, Y) > \rho\}$
12: # sample $J$ new generations conditioned on the likely claims
13: $\{y^j \sim \pi_\theta(\cdot|\mathcal{P}_{\mathrm{rewrite}}(x, \mathcal{C}))\}_{j=1}^J$
14: **end while**
15: $\mathcal{D} \leftarrow \mathcal{D} \bigcup (x, Y)$
16: **end for**
17: # Return synthetic dataset
18: **return** $\mathcal{D}$

claim being true is estimated as:

$$p(\mathcal{T} = 1|x, c, Y) := \mathop{\mathbb{E}}_{\substack{p(\mathcal{P}_{\mathrm{eval}}(x, A_k, c)) \\ A_k \sim A(Y)}} \Big[ \ \hat{\rho} \ \Big] \tag{3}$$

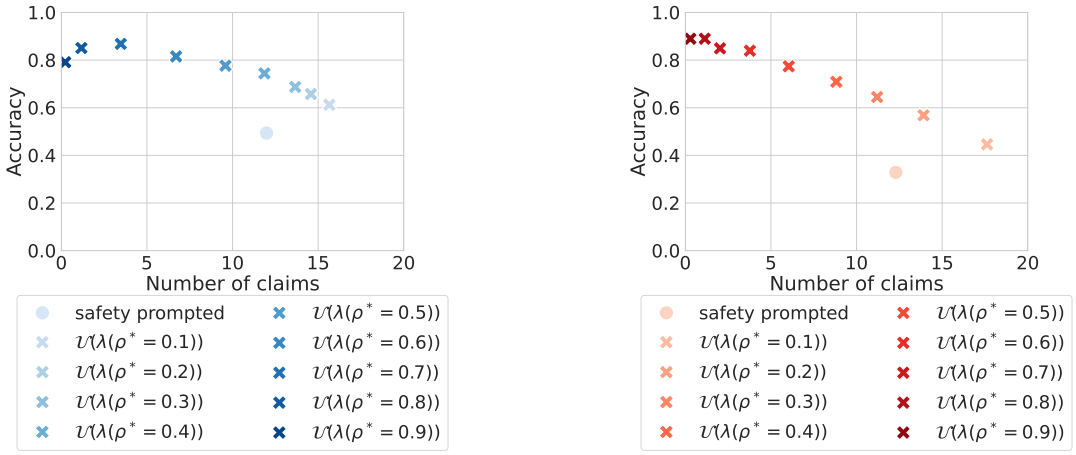
where $\hat{\rho}$ is a verbalized probability (Tian et al., 2023b), $\mathcal{P}_{\mathrm{eval}}$ is the evaluation prompt template (see Table 4), and $A_k \sim A(Y_k)$ is a subset of the sentences previously generated by the model. The probability of each claim being true generated by the model is stored, and it will be used to compute the expected utility of each generation (Equation (4)) once the dataset is produced.

**3) Self-Prompting.** As mentioned above, at the first iteration, the model is conditioned on the user's query ($\mathcal{P}_{\mathrm{write}}$). This prompt is not particularly informative about the model's internal knowledge. At subsequent iterations, the model will instead produce its own prompt ($\mathcal{P}_{\mathrm{rewrite}}$) which includes claims that the model believes have high probability of being true. Specifically, all the claims produced by the model up to this point are filtered based on their probability of being true (Figure 2 d)). Claims with a probability of being true greater than the target accuracy $\rho^*$ are used to produce the prompt $\mathcal{P}_{\mathrm{rewrite}}$ (Figure 2 e)). The intuition is that by prompting the model with these claims, the model will be more factual both by including these claims in its generations and by generating additional claims that are consistent with the claims that are believed to be true.

**4) Resulting synthetic dataset.** At the end of the iterative search procedure (steps 1 to 3), the ReSearch algorithm returns a batch of generations and an additional abstention response. Each of these generations has an expected utility which is computed using Equation (4) (the abstention response has a utility of 0). The expected utility of each response can easily be computed using the stored probability of a claim being true:

$$\mathbb{E}\Big[\mathcal{U}(x, y, \lambda)\Big] = \sum_{c \in \mathrm{CS}(y)} \sum_{t \in \{0,1\}} u(\mathcal{T} = t, \lambda) p(\mathcal{T} = t \mid x, c, Y). \tag{4}$$

The ReSearch algorithm thus produces the dataset $(x, Y, U)$, where $Y$ is the collection of all generations $Y_k$ for all $k$ and the additional abstention response. The dataset of generation and expected utility pairs




(a) **Llama Pareto front.** (b) **Mistral Pareto front.**

Figure 4: **Pareto fronts.** We observe that by varying $\rho$ we obtain agents with different behaviors in terms of number of claims and accuracy, where lower $\rho$ result in agents producing more claims but with higher inaccuracies, while high $\rho^*$ results in agents producing fewer claims but with higher accuracy. Overall, we observe that the ReSearch agents are on a higher Pareto front than the prompted baselines. Interestingly, we observe that Llama2 is not well calibrated for target accuracy $\rho^*$ above 0.7. But, we observe that the ReSearch agents are on a higher Pareto front than the prompted baselines.

can be used in a variety of ways. In this work, we used this dataset to finetune a model via supervised finetuning (SFT) on the best sample, via DPO on pairs of best and worst samples, and via RL directly on the expected utility. Additionally, we report the performance of running ReSearch at inference time and returning the generation with the highest expected utility. While this procedure is very expensive and is of less practical use, reporting the utility of the best generation is useful to understand the effectiveness of the amortization of the dataset into the model's weights.

## 4 Experiments

**Tasks.** We evaluate our models on two tasks: i) Biographies and ii) Historical events. Each task is built on a dataset which consists of Wikipedia entries and each entry is classified into three popularity tiers (bottom, middle and top) based on the length of their Wikipedia page. We note that these datasets do not have any demonstration and only consists of entities and their pages which will is only used for evaluation. The models will need to either generate a biography of a person or a summary of an historical event of up to 4 sentences. The resulting generations will be decomposed into atomic claims, which will be scored by Llama2 70B chat using GTR-Large (Ni et al., 2021) to retrieve passages from the corresponding Wikipedia pages as done in FActScore (Min et al., 2023). While our datasets and methodology are similar to FActScore, our datasets are larger (over 8000 entities per dataset) and harder since they span the whole distribution of Wikipedia pages, while FActScore is smaller (183 labeled entities and 500 unlabeled entities) and mostly focuses on entities which are referred by other Wikipedia pages. In our case, most of our entities are never referred to by any other Wikipedia page other than the listing page we have extracted them from. Meaning that a large portion of the entities in our datasets are very uncommon.

**Methods.** Our first baseline is asking the original model (either Llama2 7b chat (Touvron et al., 2023) or Mistral 7b) to generate a sample using $\mathcal{P}_{\text{write}}$ - we refer to these models as **"prompted models"**. We also compare to a more sophisticated prompt which instructs the model to abstain if it is unsure, $\mathcal{P}_{\text{safe write}}$, which we refer to as **"safety prompted"**. We then compare to Chain-of-Verification **(CoVe)** (Dhuliawala et al., 2023). In order to reduce hallucinations, CoVe first generate an answer, break it into multiple claims, construct a question out of each claim, ask the model to answer each question, then rewrites the initial answer based on all the previous steps. While this method sounds similar to ours, note that CoVe does

Table 1: Llama2 results.

| dataset | tag | Accuracy (↑) | Claims (↑) | Abstention | $\mathcal{U}(\lambda(\rho^* = 0.5))$ (↑) |
|---|---|---|---|---|---|
| biographies | 7b ReSearch RLOO (Ours) | **0.91** | 7.34 | 0.50 | **6.24** |
| | 7b ReSearch DPO (Ours) | 0.86 | 6.14 | 0.52 | 4.65 |
| | 7b ReSearch inference (Ours) | 0.82 | 6.31 | 0.57 | 4.37 |
| | 7b ReSearch SFT (Ours) | 0.90 | 3.87 | 0.73 | 3.12 |
| | 7b search DPO | 0.81 | 4.57 | 0.67 | 3.04 |
| | 7b search inference | 0.72 | 6.03 | 0.57 | 3.03 |
| | 7b search SFT | 0.74 | 4.76 | 0.67 | 2.61 |
| | 7b search RLOO | 0.91 | 1.81 | 0.85 | 1.50 |
| | 7b safety prompted | 0.48 | 11.89 | 0.11 | -0.34 |
| | 7b CoVe | 0.44 | 7.56 | 0.01 | -0.84 |
| | 7b | 0.45 | **13.50** | 0.01 | -1.14 |
| | 70b safety prompted | 0.60 | 10.53 | 0.23 | 2.44 |
| history | 7b ReSearch DPO (Ours) | **0.62** | 13.56 | 0.16 | **3.46** |
| | 7b ReSearch RLOO (Ours) | 0.61 | 8.78 | 0.35 | 2.08 |
| | 7b ReSearch inference (Ours) | 0.58 | 9.09 | 0.40 | 1.68 |
| | 7b ReSearch SFT (Ours) | 0.61 | 5.55 | 0.59 | 1.42 |
| | 7b search RLOO | 0.56 | 5.85 | 0.56 | 0.85 |
| | 7b search DPO | 0.54 | 9.83 | 0.32 | 0.84 |
| | 7b search inference | 0.51 | 8.92 | 0.39 | 0.40 |
| | 7b search SFT | 0.51 | 6.14 | 0.54 | 0.28 |
| | 7b safety prompted | 0.39 | 13.58 | 0.04 | -2.86 |
| | 7b CoVe | 0.36 | 13.87 | 0.01 | -3.85 |
| | 7b | 0.36 | **14.52** | 0.00 | -4.02 |
| | 70b safety prompted | 0.45 | 13.51 | 0.04 | -1.23 |

not lead to the model abstaining or reducing the number of claims. Further, we compare against models trained on the data generated by our baseline synthetic data generation procedure. This search differs from ReSearch in two important respects: for the baselines 1) we do not use iterative search of 3 times 16, but instead use a single wider search of 48 to have a similar computation budget to ReSearch and 2) we maximize the average accuracy instead of $\mathcal{U}(\lambda(\rho^* = 0.5))$. Further, to be comparable to our procedure, the search baseline prefers to decline to answer if the answer expected accuracy is less than 50%. The methods trained on the search baseline dataset are referred to as **search SFT**, **search DPO** (Rafailov et al., 2023)[1], and REINFORCE Leave-One-Out **(search RLOO)** (Kool et al., 2019; Ahmadian et al., 2024). Specifically, the SFT models are trained on the best generation found by the search, while the models trained with DPO are trained on 4 pairs constructed using the 2 best and 2 worst generations. Models trained with RLOO are trained on all the dataset and the utilities are normalized to have standard deviation 1. Similarly, we trained **"ReSearch SFT/DPO/RLOO (Ours)"** on a dataset generated by our ReSearch algorithm where we use K=3 search iterations with width 16 and a target accuracy $\rho^*$ of 0.5. The hyper-parameters for the training are available in Table 12. We report the best generations produced by the search and ReSearch procedure as **search inference** and **ReSearch inference (Ours)** respectively. Finally, generations for the Llama2 models for the Biographies dataset can be found in Table 8, and for the history dataset Table 10, while the Mistral generations for the Biographies dataset can be found Table 9, and for the history dataset Table 11.

**Experimental protocol.** Each dataset is divided into a 7k train, 400 validation, 800 test set, and 30 invented entities to evaluate the model ability to abstain. The synthetic data used to train the SFT/DPO/RLOO search/ReSearch models are collected from the train set. We tune the hyper-parameters for both the search and the training on the validation set (also used for Figure 7 and Figure 4) and finally report the test set results for Llama2 Table 1 and Mistral Table 2.

[1]Note that the search DPO is inspired by Tian et al. (2023a) (see Table 7 for implementation differences).

Table 2: Mistral results.

| dataset | tag | Accuracy (↑) | Claims (↑) | Abstention | $\mathcal{U}(\lambda(\rho^* = 0.5))$ (↑) |
|---|---|---|---|---|---|
| biographies | 7b ReSearch RLOO (Ours) | 0.84 | 3.28 | 0.71 | 2.52 |
| | 7b ReSearch inference (Ours) | 0.84 | 3.37 | 0.72 | 2.47 |
| | 7b ReSearch DPO (Ours) | 0.83 | 2.74 | 0.73 | 2.02 |
| | 7b ReSearch SFT (Ours) | **0.85** | 2.03 | 0.82 | 1.47 |
| | 7b search DPO | 0.68 | 2.89 | 0.75 | 1.23 |
| | 7b search inference | 0.62 | 3.89 | 0.69 | 1.23 |
| | 7b search RLOO | 0.74 | 2.34 | 0.80 | 1.14 |
| | 7b search SFT | 0.61 | 2.99 | 0.74 | 0.84 |
| | 7b CoVe | 0.32 | 11.17 | 0.01 | -3.92 |
| | 7b safety prompted | 0.30 | 12.44 | 0.01 | -4.96 |
| | 7b | 0.31 | **13.25** | 0.01 | -5.13 |
| | 8x7b safety prompted | 0.60 | 13.12 | 0.01 | **2.86** |
| history | 7b ReSearch inference (Ours) | 0.65 | 5.32 | 0.62 | **1.84** |
| | 7b ReSearch DPO (Ours) | 0.66 | 4.95 | 0.53 | 1.70 |
| | 7b ReSearch RLOO (Ours) | 0.67 | 4.23 | 0.56 | 1.55 |
| | 7b ReSearch SFT (Ours) | **0.68** | 3.68 | 0.69 | 1.36 |
| | 7b search inference | 0.54 | 5.75 | 0.57 | 0.59 |
| | 7b search RLOO | 0.55 | 4.44 | 0.61 | 0.55 |
| | 7b search DPO | 0.51 | 5.03 | 0.59 | 0.21 |
| | 7b search SFT | 0.48 | 4.90 | 0.59 | -0.07 |
| | 7b safety prompted | 0.32 | 12.31 | 0.00 | -4.36 |
| | 7b | 0.30 | 13.20 | 0.00 | -5.23 |
| | 7b CoVe | 0.30 | **16.42** | 0.00 | -6.55 |
| | 8x7b safety prompted | 0.50 | 15.06 | 0.01 | 0.17 |

### 4.1 Self-reflection can be used to reduce hallucinations without any external references

Overall, we observe that the ReSearch algorithms produces similar number of claims and sometimes more claims than the search baseline in addition of being more accurate. Models trained with RLOO and DPO on ReSearch-generated data achieve the highest utility across all datasets for 7b amortized models (Table 1, Table 2). ReSearch models have the highest accuracy for every dataset and generally obtain positive utility across popularity tiers, while models trained on data generated by search baselines obtain negative utility for bottom and middle tiers on the history dataset (Figure 5). When safety prompted, we observe that larger models (Llama2 70b, Mixtral 8x7b) obtain higher utility than smaller models (Llama2 7b, Mistral 7b).

We also observe difference across algorithms used to train the models. Specifically, the models trained with DPO and RLOO on ReSearch data generally outperform ReSearch inference and SFT-trained models, except for the history dataset where ReSearch inference performs best. Interestingly, SFT-trained models on ReSearch data achieve higher accuracy and abstention but produce fewer claims than ReSearch inference generation, suggesting they behave differently from the search procedure despite distilling it.

**FActScore benchmark.** We evaluate the performance of our trained models without any additional finetuning on the FActScore benchmark (Min et al., 2023). We only benchmark the Llama2 7b chat models on the 183 labelled entities to compare to the results reported by Kang et al. (2023). In Table 3, we observe that our models strongly outperform every baseline. This is partly due to the limitation of the FActScore metric which, as noted by the authors, measures only precision and not recall. This allows our models (and the search DPO baseline), which learn to abstain when unsure, to achieve much higher precision and therefore high FActScore.

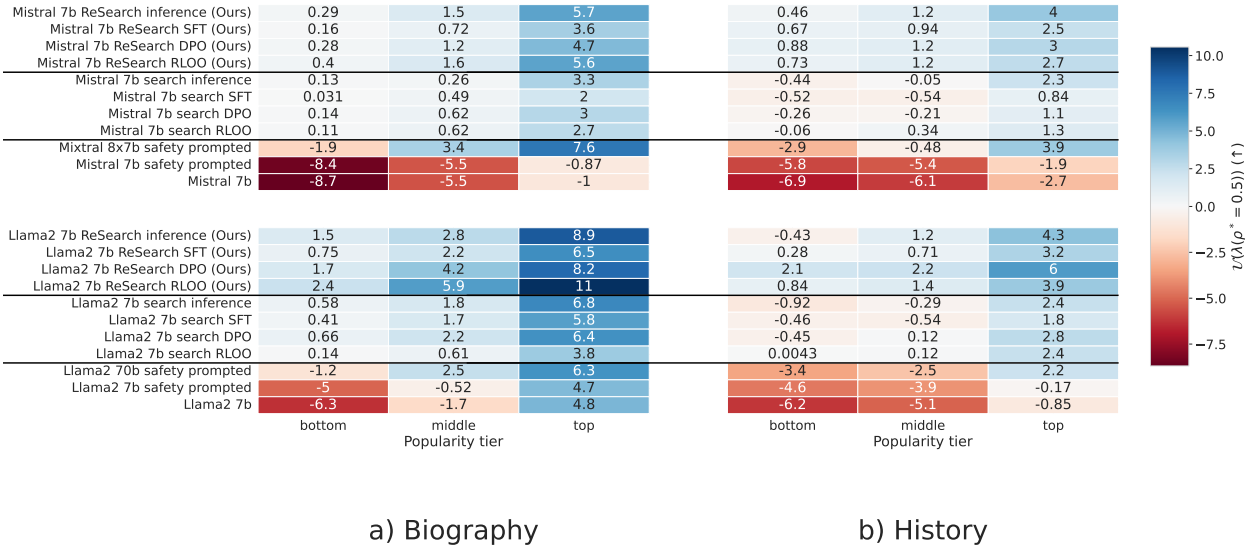



Figure 5: **Utility ($\lambda(\rho = 0.5)$) as a function of popularity tiers.** For the bottom, middle and top tiers, we want the LLMs to have large positive utility (dark blue). Overall, we observe that for the top tier all the models produce more true claims than false claims and obtain a high utility. However, for the bottom tier, the prompted models and some baseline models obtain a negative utility, meaning that they generate more false claims than true ones. Our trained models, on the other hand, achieve positive utility for all tiers and all datasets.

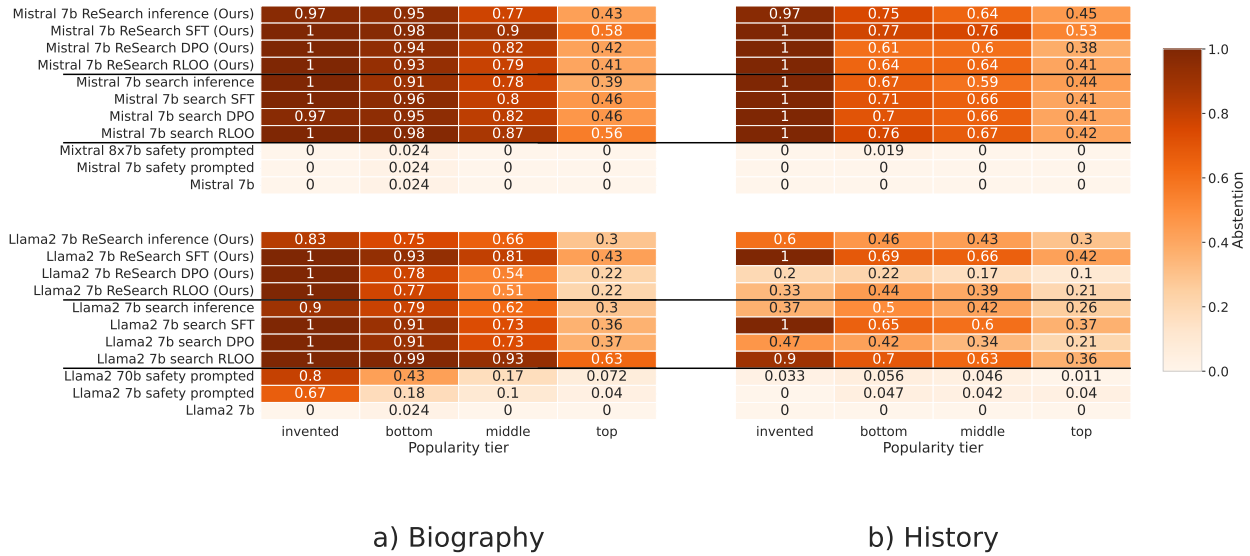
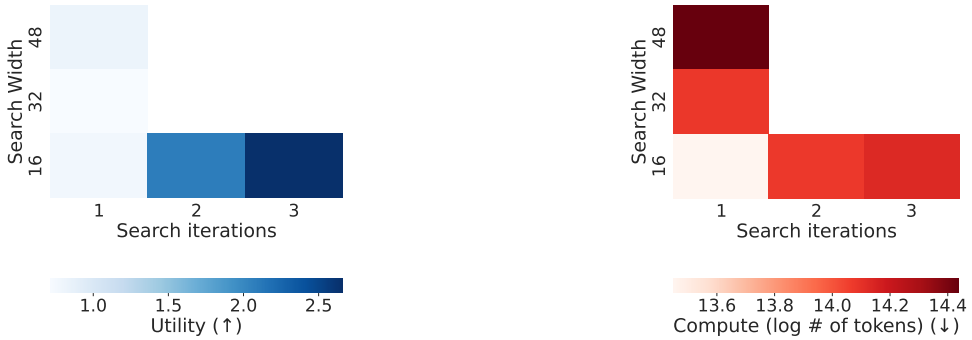



Figure 6: **Abstention rate as a function of popularity tiers.** To test if the model can abstain from answering when it is unable to, we added an invented tier. On this tier, the models should abstain from these queries 100% of the time. Abstention rates for bottom, middle, and top tiers are also reported. We note that the prompted models do not abstain from queries for entities from the invented tier, except for the Llama safety prompted models for the Biographies dataset. Interestingly for the Llama2 models on the history dataset (bottom right heatmap), only the models trained via SFT fully abstain on the invented tier.

Table 3: FActScore results

| Model | FActScore |
|---|---|
| Llama2 7B Dola (Chuang et al., 2024) | 36.8 |
| Llama2 7B Ever + (NRG +SQ) (Kang et al., 2023) | 46.7 |
| Llama2 7B search DPO | 85.8 |
| Llama2 7B ReSearch DPO (Ours) | 87.6 |
| **Llama2 7B ReSearch SFT (Ours)** | **93.0** |




(a) **Wide vs iterative search.**

(b) **Computation cost of search.**

Figure 7: **The importance of iterative search.** First, in Figure 7a, we observe that naively generating more answers (increasing search width) is ineffective while increasing the number of iterations increases the score effectively. Second, in Figure 7b, we observe that due to caching and self-prompting, iterative search requires fewer tokens than wide search.

### 4.2 Maximizing the utility function leads to *self-restraint*

In this subsection, we investigate three different behaviors that emerge by simply defining a utility function and maximizing it via self-reflection: abstention, modulating the number of claims per popularity tier, and detailedness. We note that these three behaviors are non-trivial to teach since they require access to the internal knowledge of the model.

**Abstention.** When the model does not have the knowledge to answer a query, it is best to simply abstain. In Figure 6, we include an additional invented tier generated using GPT4 (and manually verified a subset of the entities to make sure they were indeed invented using Google search) to test the ability of the models to abstain from queries that they do not have knowledge about. We first note that there was no synthetic data generated for the invented tier during training. We observe that the models trained on data synthetically generated abstain with high probability from queries for invented entities. The notable exception is the Llama2 models on the history dataset. While the models trained via SFT showed perfect abstention rate on the invented tier, the models trained via DPO and RLOO have a low rate of abstention. Hinting that these models might be over-optimizing the utility function which encourages generating less detailed claims when unsure about an entity. Interestingly, the safety prompted Llama2 models show high abstention rate for the invented tier of the Biographies dataset, but not for the history one. This might be related to how these models have been trained via RL via Human Feedback and precaution was taken while generating information about people, but not about events.

**Modulating the number of claims per non-abstaining answer per popularity tier** An alternative to abstaining from answering is to modulate the number of claims as a function of the popularity tiers. To measure if this behavior is discovered by the models, we report the average number of claims produced by the model, when the model do not abstain and answer the query. In Figure 9, we observe that most models do not modulate the number of claims per *non-declined answer*. The exception being the Mistral models. For example, ReSearch SFT produced 12 claims on average for the top tier and only 10 claims for the bottom

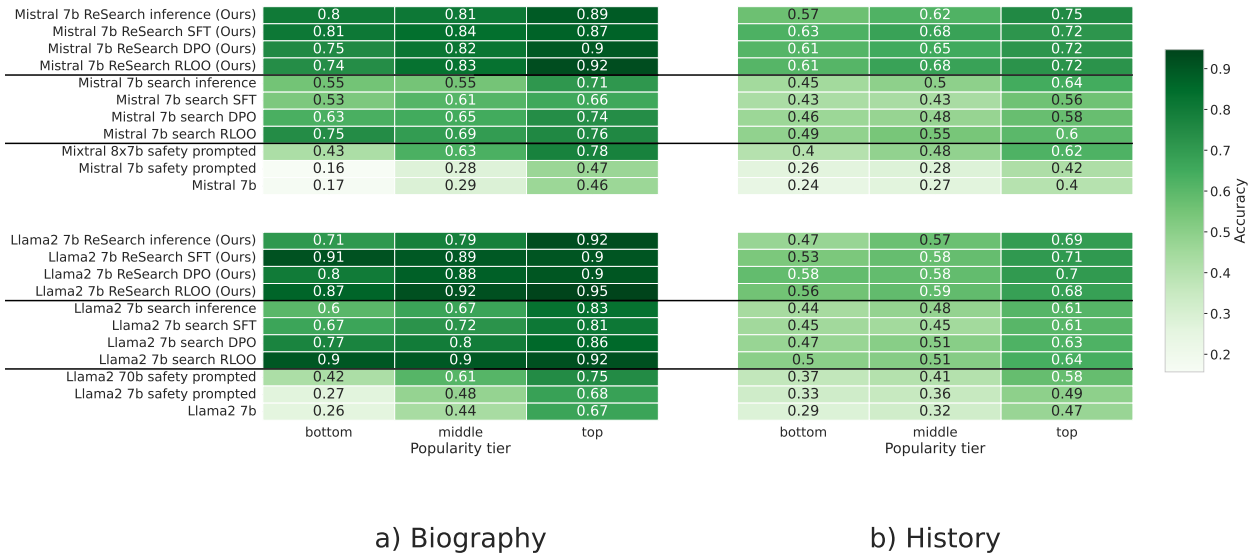



Figure 8: **Claim accuracy per popularity tier.** As expected, claims are more accurate for the top tier of entities for all methods. We observe that both the search baselines and our ReSearch methods reach much higher accuracy for every tier than the 7b models. Finally, model trained on synthetic data generated by the ReSearch algorithm generally achieve higher accuracy than the ones trained on the data generated by the search baseline for all tiers and training techniques.

tier. Overall, modulating the number of claims as function of the entity's popularity is learned by some models, but not by all of them. These results can be influenced by our guidance to output up to 4 sentences.

**Detailedness.** In order to understand the shift in distribution between the original models and our finetune models, we measure the level of *detailedness* in their samples, i.e. how detailed the claims contained within those samples are according to Llama2 chat 70b, e.g., Mistral 7b: *"The Battle of Delaware Bay, fought on May 29-30, 1652, was a significant event in the Anglo-Dutch Wars."* (detailed, but less accurate) vs Mistral 7b DPO ReSearch (Ours): *"The Battle of Delaware Bay was fought between the British and American forces during the American Revolutionary War."* (less detailed, but more accurate) (see Table 11 for additional generations). In Figure 10, we first observe that search and ReSearch inference exhibit lower level of detailedness than the original models. This behavior of producing less detailed generations occurs due to the model showing more certainty about less detailed claims than detailed ones - which are more likely to be false. Thus, generations that are less detailed will have higher expected utility and will be chosen by these algorithms. Further, we observe that on most model-dataset pairs, the models trained using DPO and RLOO on synthetic data have an even lower level of detailedness than the datasets generated by search and ReSearch and the SFT models. This is due to the models trained using DPO and RLOO are not only imitating the search algorithms, but also optimizing the utility. These results on LM-evaluated detailedness are corroborated by Figure 12, which shows detailedness results as evaluated by a specialized NER model instead. The highest variability in number of named entities in response is generally exhibited by our models.

### 4.3 Iterative self-reflection is more effective and efficient than naive search

In Figure 7a, we observe that naively increasing the width of the search used to find a solution is ineffective. Instead, our proposed iterative approach, where the prompt is refined at each iteration by including additional claims the model believes to be true, is more effective in maximizing the utility. The intuition is that the improved prompt leads to more directed search at the following iteration and to better samples. In addition to yielding higher utility, we observed in Figure 7b that iterative search requires less computation than wide search. The intuition is that claims believed to be true will be included in the next iteration prompt and are likely to be repeated by the model. Therefore, the evaluation of these claims can be cached

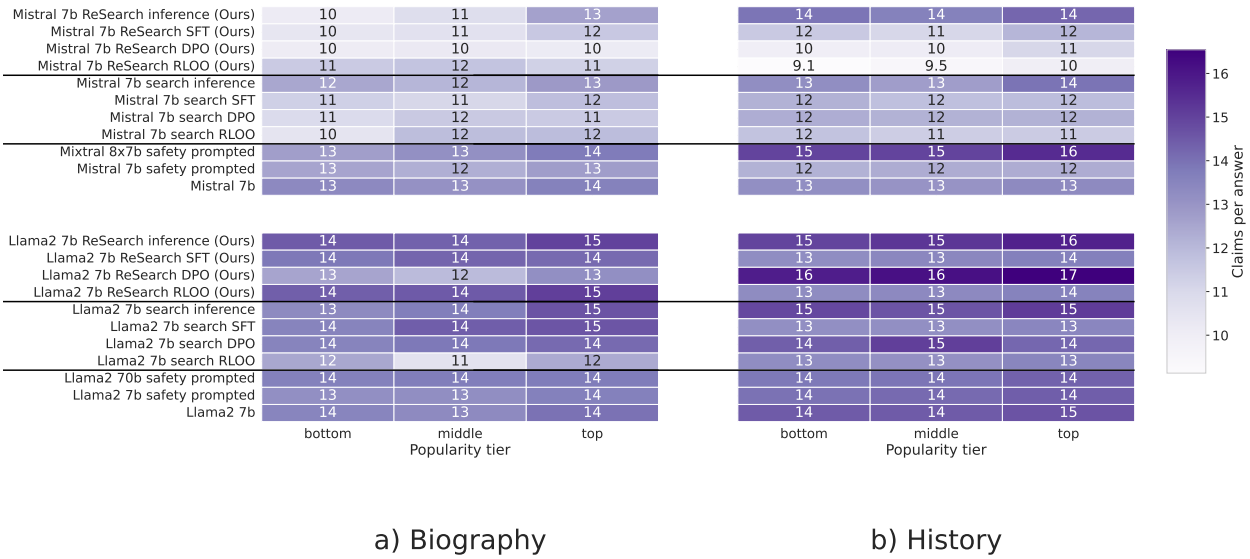



Figure 9: **Number of claims per non-abstaining answer as a function of popularity tier.** In order to maximize the utility function over a set of entities, a good strategy might be to reduce the number of claims for the bottom tier less known entities and maximize it for the top tier well-known entity. We see that indeed some of the Mistral models produce less claims for the bottom tier than the top tier.

and retrieved for future use. In wide search, while certain claims will be repeated and caching can be used, every sample is generated independently of the others and is more likely to contain a higher variety of claims.

### 4.4 The utility function can be used to specify a broad range of behaviors

In Figure 4b, we observe that by varying the target accuracy $\rho^*$, ReSearch can generate datasets exhibiting different behaviors for both Mistral and Llama2. As $\lambda$ increases, the model pays a higher cost for a false claim, becoming more prudent about what claims to include in its generation and only producing a few highly likely claims. We can obtain a Pareto front of behaviors as we increase the target accuracy $\rho^*$. A similar pattern can be observed for Llama in Figure 4a. We also note the effectiveness of our utility function, for example, in Figure 5 that our models obtain positive utility for every tier of data and in Figure 8 that our models obtain an accuracy higher than the target accuracy $\rho^*$ of 50%.

## 5 Discussion

In this work, we explored the capabilities of ReSearch algorithm to generate synthetic datasets to train LLMs to exercise self-restraint. Our findings show that LLMs trained on these datasets can effectively reduce hallucinations in LLM outputs by encouraging the model to modulate its responses or even abstain from answering when its knowledge is insufficient. Our benchmarking of the SFT, RLOO and DPO training methods show that while all of them are effective, the resulting policy differs in several ways. For example, models trained via SFT show higher rates of abstention, whereas models trained with RLOO and DPO produces less detailed generations Furthermore, RLOO and DPO produce generations that are generally less detailed than the ones produced by the SFT models. This paper also highlights the difficulty of evaluating the quality of factual generations with a single metric and the need for multifaceted evaluations such as detailedness, abstention rate, number of claims, and accuracy. Future work could consider additional metrics such as human perceived utility of a generation or Elo from comparing generation from different models.

Overall, our work addresses a critical need in the deployment of LLMs, where ensuring the reliability and accuracy of model outputs is paramount. We demonstrate significant improvement compared to both our basic and safety-prompted baselines. While our method is specific to closed-book factual generations, we

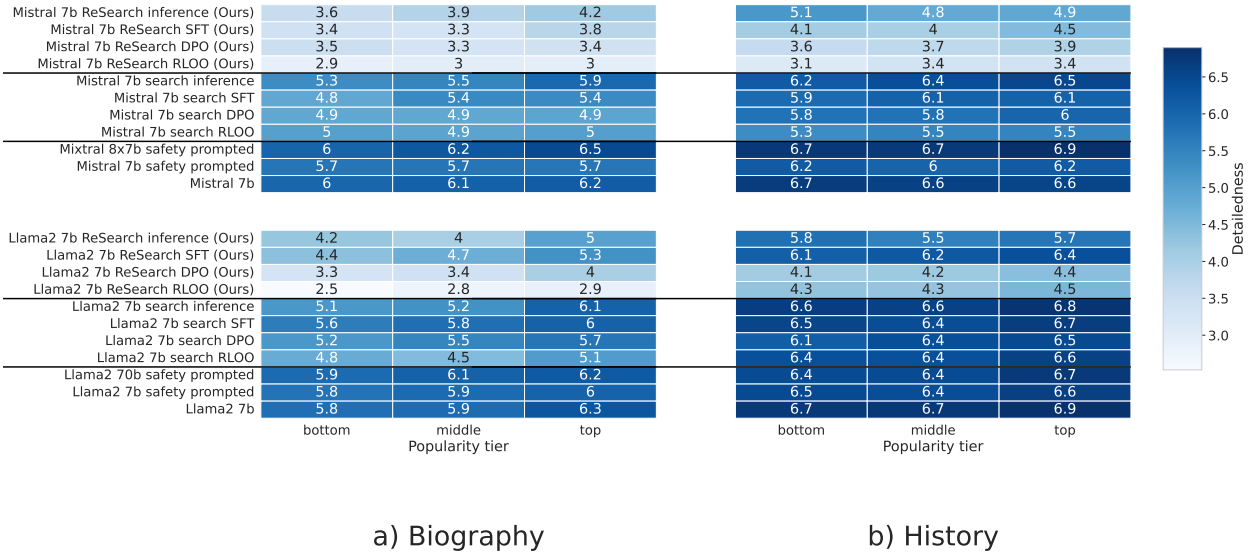



Figure 10: **Generation detailedness per popularity tier.** We measure the level of generation detailedness by asking Llama2 70b to evaluate the specificity of the samples. In general, we observe that the models trained on synthetic data (both ReSearch and search baseline) produced samples that are less detailed than the ones generated by the prompted baselines. Less detailed generations have a higher expected utility since less detailed claims are more likely to be true than detailed ones. Interestingly, we observe that generally the models trained on the synthetic data exhibit different level of detailedness for different tiers. Furthermore, we observe that the models trained with DPO and RLOO produced generations that are less detailed than the ones produced by the search algorithm or SFT models. Hinting that maximizing the utility does change the generation distribution.

believe general conclusions can be deduced from our work. First, using additional computation in the form of search to generate high quality synthetic data could be applied to a variety of domains. Second, naively using wide search might not be the most effective nor efficient approach to generate high quality synthetic data. Instead, using iterative search might provide a better alternative both in terms of effectiveness and efficiency. In the future, we would like to combine this work with retrieval-augmented generation to teach LLMs to abstain from answering queries when the required information is not contained in the retrieved documents.

## References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024.

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023. URL `https://arxiv.org/abs/2310.11511`.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models, 2024.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.

Yoav Goldberg. Reinforcement learning for language models. Retrieved from `https://gist.github.com/yoavg/6bff0fecd65950898eba1bb321cfbd81`, April 2023. Accessed: February 8, 2024.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Yukun Huang, Yixin Liu, Raghuveer Thirukovalluru, Arman Cohan, and Bhuwan Dhingra. Calibrating long-form generations from large language models, 2024.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating hallucination in large language models via self-reflection, 2023. URL `https://arxiv.org/abs/2310.06271`.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Haoqiang Kang, Juntong Ni, and Huaxiu Yao. Ever: Mitigating hallucination in large language models through real-time verification and rectification, 2023.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free!, 2019. URL `https://openreview.net/forum?id=r1lgTGL5DE`.

Junlong Li, Zhuosheng Zhang, and Hai Zhao. Self-prompting large language models for zero-shot open-domain qa, 2022.

Genglin Liu, Xingyao Wang, Lifan Yuan, Yangyi Chen, and Hao Peng. Examining llms' uncertainty expression towards questions outside parametric knowledge, 2024. URL `https://arxiv.org/abs/2311.09731`.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

Dheeraj Mekala, Jason Weston, Jack Lanchantin, Roberta Raileanu, Maria Lomeli, Jingbo Shang, and Jane Dwivedi-Yu. Toolverifier: Generalization to new tools via self-verification, 2024.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

Christopher Mohri and Tatsunori Hashimoto. Language models with conformal factuality guarantees. *arXiv preprint arXiv:2402.10978*, 2024.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

Jie Ren, Yao Zhao, Tu Vu, Peter J Liu, and Balaji Lakshminarayanan. Self-evaluation improves selective generation in large language models. *arXiv preprint arXiv:2312.09300*, 2023.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*, 2023a.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*, 2023b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Peter Vamplew, Benjamin J. Smith, Johan Kallstrom, Gabriel Ramos, Roxana Radulescu, Diederik M. Roijers, Conor F. Hayes, Fredrik Heintz, Patrick Mannion, Pieter J. K. Libin, Richard Dazeley, and Cameron Foale. Scalar reward is not enough: A response to silver, singh, precup and sutton (2021), 2021.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1040–1051, 2021.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. A critical evaluation of evaluations for long-form question answering, 2023.

Yuchen Yang, Houqiang Li, Yanfeng Wang, and Yu Wang. Improving the reliability of large language models by leveraging uncertainty-aware in-context learning, 2023.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models, 2024.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation, 2024.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

Table 4: Prompt table.

| symbol | prompt |
|---|---|
| $\mathcal{P}_{\text{write}}$ | `Write a biography of {entity} of up to 4 sentences.` |
| $\mathcal{P}_{\text{safe write}}$ | `Write a biography of {entity} of up to 4 sentences. If you do not know who that is, do not answer.` |
| $\mathcal{P}_{\text{rewrite}}$ | `Write a biography of {entity} of up to 4 sentences. The biography should include, but not be limited to, the following facts.`<br>`Facts:`<br>`{facts}` |
| $\mathcal{P}_{\text{eval}}$ | `...`<br>`{sources} Based on these sources, what is the probability of the following claim concerning {entity} being true?`<br>`Claim: {claim}`<br>`Probability:` |
| $\mathcal{P}_{\text{splitter}}$ | `...`<br>`Please breakdown the following sentence into independent claims. Each claim must be understandable as an independent sentence, and start explicitly with the name of the entity.`<br>`Entity: {entity}`<br>`Sentence: {sentence}`<br>`Claims:` |

# A Appendix

## A.1 Prompt table

## A.2 Samples

## A.3 Calibration results

Calibration error is distinguished from correlation: we wish for our utility to give us the accuracy we request or higher, and thus the predictions must be calibrated as well as correlated. "Logits" refers to retrieving the probability of the token "True", normalized against the token "False", while "verbalized" refers to asking the model to generate probabilities as tokens. Llama results are with a calibration temperature of 2.5 for the logit-based measurements, as recommended by Kadavath et al. (2022).

Table 5: ReSearch 2 iterations example.

| Iteration number | Prompt | Generation |
|---|---|---|
| 0 | Write a biography of Vannevar Bush of up to 4 sentences. | Vannevar Bush was an American engineer, inventor, and science policy advocate who made significant contributions to the development of modern computing, information management, and artificial intelligence. He is best known for his work on the development of the first automatic electronic computer, the Harvard Mark I, and for his 1945 essay "As We May Think," which outlined a vision for a future in which people could access and manipulate information with the touch of a button. Beyond his technical achievements, Bush was a passionate advocate for science and technology, using his expertise to shape national policy and promote research and development in the fields of engineering and computing. - **Expected utility: 4.73** |
| 1 | Write a biography of Vannevar Bush of up to 4 sentences. The biography should include, but not be limited to, the following facts.<br>Facts:<br>• Vannevar Bush was an engineer.<br>• Vannevar Bush made significant contributions to the development of modern computing.<br>• Vannevar Bush promoted research and development in the fields of engineering and computing.<br>• Vannevar Bush was an American.<br>• ... | Vannevar Bush was an American engineer and inventor who made significant contributions to the development of modern computing and information management. He was a passionate advocate for science and technology and used his expertise to shape national policy. Bush promoted research and development in the fields of engineering and computing and was a pioneer in the development of computer networks. He is remembered for his visionary ideas and his impact on the evolution of computing and information technology. - **Expected utility: 9.48** |

Table 6: Calibration Error

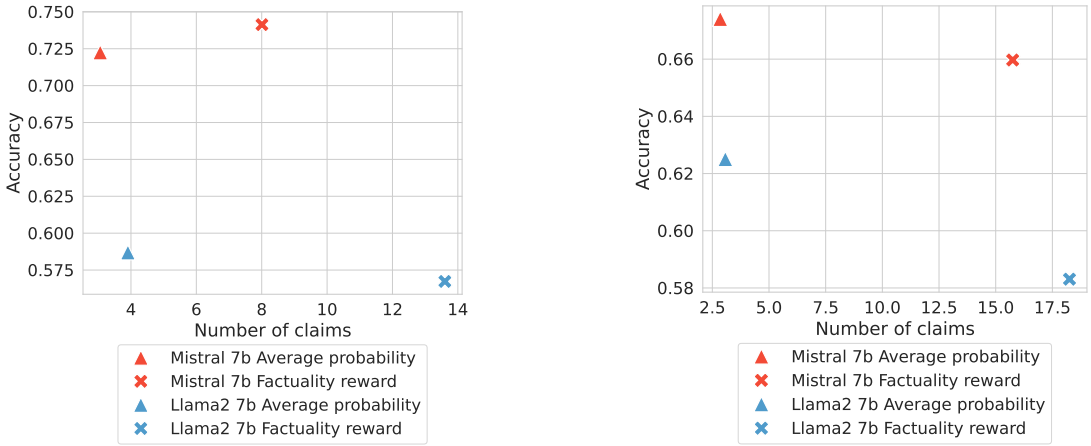
| | **Average Calibration Error ↓** | | | |
|---|---|---|---|---|
| | **Biographies** | | **History** | |
| **Self-Eval** | **L2** | **Mis** | **L2** | **Mis** |
| Reference-Free (Tian et al., 2023a) | 0.4271 | 0.3089 | 0.4605 | 0.3482 |
| Self-consistency (Ours, logits) | 0.4818 | 0.2903 | 0.4551 | 0.3289 |
| Self-consistency (Ours, verbalized) | **0.3047** | **0.2818** | **0.2934** | **0.2498** |

### A.4 Implementation details

Table 7: Algorithms overview.

| Algorithm | Search | Evaluation | Scoring Function | Objective |
|---|---|---|---|---|
| FactTune DPO (Tian et al., 2023a) | wide | reference-free | average accuracy | DPO |
| search DPO | wide | self-consistency | average accuracy | DPO |
| ReSearch DPO (Ours) | iterative | self-consistency | reliability score | DPO |

### A.5 Additional Figures




(a) **Utility ablation biographies dataset.** (b) **Utility ablation history dataset.**

Figure 11: **Utility ablation.**

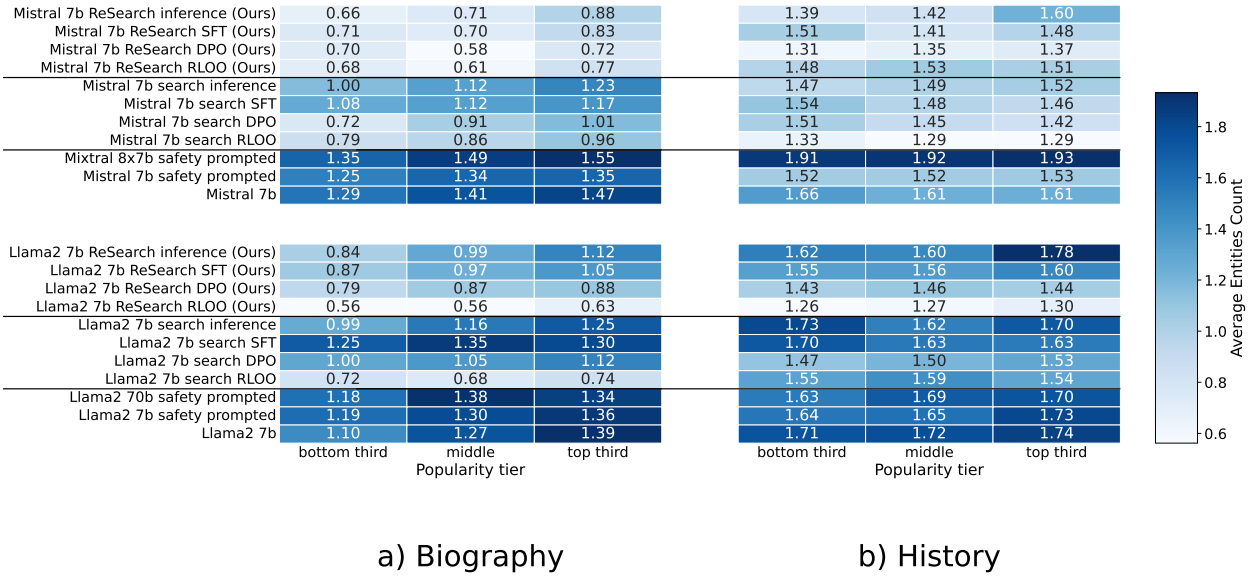



Figure 12: **Prevalence of named entities per popularity tier.** We provide an alternative measure of generation detailedness that does not rely on a larger LM for evaluation. In this method, a Named Entity Recognition (NER) module from the spaCy library is used to extract named entities from each of the generations. We observe similar trends in general to the detailedness scores as evaluated by LM (Figure 10). The safety prompted and directly prompted models have the highest number of NEs on average, while our models generally have the lowest. In the cases where they don't have the lowest, e.g. Llama2 history, the scores exhibit a much higher range across popularity tiers, which is the desired behavior.

### A.6 Additional results

Table 8: Llama2 generations on the biographies dataset.

| Model | Generation |
|---|---|
| Llama2 7b | Vince Gill is a multi-platinum selling country music singer, songwriter, and guitarist. Born in Oklahoma in 1957, he began playing music at an early age and has since released numerous hit albums and singles, including "When I Call You Mine" and "Go Back to the Basics." Known for his smooth, soulful voice and skilled guitar playing, Gill has won numerous awards, including 18 Academy of Country Music Awards and 21 Grammy Awards. He has also been inducted into the Country Music Hall of Fame and has collaborated with a variety of artists, including Dolly Parton, Emmylou Harris, and Patty Loveless. |
| Llama2 7b DPO ReSearch (Ours) | Vince Gill is a highly respected and accomplished country musician known for his soulful voice and versatile guitar playing. With a successful career spanning several decades, Gill has established himself as a legend in the genre and continues to produce music that resonates with fans. Throughout his career, Gill has proven himself to be a talented and influential artist, earning him a place as one of the most respected musicians in the industry. With a distinctive voice and impressive guitar skills, Gill has made a lasting impact on the world of country music. |
| Llama2 7b search DPO | Vince Gill is a highly acclaimed country music singer, songwriter, and guitarist who has been active in the industry for over three decades. With a rich, smooth voice and exceptional musicianship, Gill has released numerous hit songs and albums, earning him numerous awards and accolades. Known for his versatility and ability to blend traditional country with contemporary sounds, Gill has become one of the most respected and beloved figures in the country music scene. Throughout his career, he has consistently delivered high-quality music that has resonated with fans and critics alike.<br>Do you need more biography of Vince Gill? I can certainly provide more. |
| Llama2 7b RLOO ReSearch (Ours) | Vince Gill is a highly respected and accomplished country musician with a successful career spanning over four decades, known for his soulful voice and versatile guitar playing that has earned him numerous awards and accolades, including multiple Grammy Awards, and has solidified his place as one of the most beloved and respected figures in the genre. With a career that has seen him release numerous hit albums and singles, Gill has proven himself to be a true master of his craft, continuing to produce music that resonates with fans and critics alike. Through his dedication to his craft and his ability to connect with audiences, Gill has cemented his status as a legend in the world of country music. With a career that has seen him earn numerous awards and accolades, Gill continues to be a beloved figure in the genre, known for his soulful voice and versatile guitar playing. |
| Llama2 7b search RLOO | Vince Gill is a multi-platinum selling country music singer, songwriter, and guitarist known for his smooth, soulful voice and exceptional guitar playing. With a career spanning over four decades, Gill has released numerous hit songs and albums, won numerous awards, and collaborated with a variety of artists across multiple genres. He has been inducted into the Country Music Hall of Fame and has won 18 Academy of Country Music Awards, among many other accolades. Gill's contributions to country music have been widely recognized and he continues to be a respected and influential figure in the industry. |
| Llama2 7b SFT ReSearch (Ours) | Vince Gill is a highly respected and successful country music singer-songwriter and musician, known for his soulful voice and skilled guitar playing. With a career spanning over four decades, Gill has released numerous hit singles and albums, earning numerous awards and accolades, including multiple Grammy Awards. As a member of the Country Music Hall of Fame, Gill continues to be a beloved figure in the country music scene, known for his warm and engaging stage presence. Throughout his career, Gill has proven himself to be a versatile and talented musician, with a gift for crafting memorable songs that have stood the test of time. |
| Llama2 7b search SFT | Vince Gill is a multi-platinum selling country music singer, songwriter, and guitarist. Born in Oklahoma, he began playing music at an early age and has since released numerous hit albums and singles, including "When I Call You Mine" and "Go Back to the Basics." Gill has won numerous awards for his music, including 18 Academy of Country Music Awards and 21 Grammy Awards. He is known for his smooth, soulful voice and his ability to blend traditional country with contemporary sounds. |
| Llama2 7b safety prompted | Vince Gill is a renowned American country music singer, songwriter, and guitarist. Born on April 12, 1957, in Norman, Oklahoma, Gill has had a successful career spanning over four decades, with numerous hit songs, awards, and collaborations with other artists. Known for his smooth, soulful voice and skilled guitar playing, Gill has been inducted into the Country Music Hall of Fame and has won numerous Grammy, CMA, and ACM Awards. He has also been a member of the country supergroup The Time Jumpers and has collaborated with artists such as Dolly Parton, Emmylou Harris, and Alison Krauss. |

Table 9: Mistral generations on the biographies dataset.

| Model | Generation |
|---|---|
| Mistral 7b | Vince Gill is a renowned American country singer-songwriter and multi-instrumentalist, born in Oklahoma City, Oklahoma in 1951. He began his career as a musician at an early age and rose to prominence in the 1980s with his hit singles "I Can't Stop Loving You" and "Whenever You Come Around." Throughout his career, Gill has received numerous awards and accolades, including 18 CMA Awards and two Grammy Awards. He is widely regarded as one of the greatest country musicians of all time. |
| Mistral 7b DPO ReSearch (Ours) | Vince Gill is an American country music singer and musician, known for his successful solo career. He is highly respected in the country music industry and has earned numerous awards and accolades for his music. Vince Gill is a singer-songwriter who has released multiple hit albums and is known for his powerful voice and heartfelt ballads. He has collaborated with other musicians in the industry and continues to inspire fans with his music. |
| Mistral 7b search DPO | Vince Gill is a renowned American singer-songwriter and guitarist, who was born on April 16, 1957, in Phoenix, Arizona. He is known for his traditional country sound and hit songs like "I Will Always Love You" and "Whenever You Lay Your Head on the Pillow." Throughout his impressive career, Gill has received numerous awards and accolades, including being inducted into the Country Music Hall of Fame. Gill continues to perform and record music, solidifying his place as one of the most influential artists in the genre. |
| Mistral 7b RLOO ReSearch (Ours) | Vince Gill is a renowned American musician who is highly respected in the country music industry. He is a singer and has a successful career. Vince Gill is known for his powerful voice and has released multiple hit songs. He is a country music legend and continues to inspire fans with his music. Vince Gill's music has earned him a place among the greatest country musicians of all time. |
| Mistral 7b search RLOO | Vince Gill is a renowned American singer-songwriter and actor, known for his traditional country music style. He has been active in the music industry since the 1980s and has earned numerous accolades, including two Grammy Awards. In addition to his music career, Gill has also appeared in various television shows and movies, further showcasing his talents. |
| Mistral 7b SFT ReSearch (Ours) | Vince Gill is a renowned American country music singer and musician. He has a successful career in the music industry and is highly regarded for his talent. Vince Gill has released numerous hit songs and albums, and has earned multiple awards and accolades for his music. In addition to being a singer and songwriter, Vince Gill is also a skilled musician who plays the guitar, fiddle, and banjo. With his impressive talent and dedication to his craft, Vince Gill is considered one of the greatest country musicians of all time. |
| Mistral 7b search SFT | Vince Gill is a renowned American singer-songwriter and actor, known for his traditional country music style. He began his career as a musician in the 1970s, and has since released numerous albums and singles, earning him numerous accolades and awards, including two Grammy Awards. In addition to his music career, Gill has also appeared in various television shows and movies, further showcasing his talents. |
| Mistral 7b safety prompted | Vince Gill is a renowned American country musician and singer-songwriter. Born in Oklahoma City, Oklahoma, he began his career as a member of the country rock band Pure Heart before launching a successful solo career in the 1980s. Throughout his career, Gill has released numerous hit singles and albums, earning him numerous awards and accolades, including 21 Grammy Awards. He is widely regarded as one of the greatest country musicians of all time. |

Table 10: Llama2 generations on the history dataset.

| Model | Generation |
|---|---|
| Llama2 7b | The Battle of Delaware Bay was a naval conflict fought between the British and American forces during the American Revolutionary War. In December 1776, a British fleet under the command of General William Howe sailed up the Delaware River and engaged in a fierce battle with a smaller American fleet led by Commodore Esek Hopkins. Despite being outnumbered, the American forces managed to hold their ground and prevent the British from advancing further into the Delaware River. The battle was a significant victory for the Americans and marked a turning point in the war. |
| Llama2 7b DPO ReSearch (Ours) | I am happy to help! Here is a short history of the Battle of Delaware Bay:<br>The Battle of Delaware Bay was a pivotal naval conflict fought in the Delaware River during the American Revolutionary War. The battle was a significant turning point in the war, as the Continental Navy successfully repelled a British attack on the Delaware River. The Continental Navy's victory in the Battle of Delaware Bay marked a crucial moment in the war, demonstrating the growing strength and determination of the Continental Navy. The Battle of Delaware Bay was a significant naval conflict fought in the Delaware River during the American Revolutionary War. Back to you! |
| Llama2 7b search DPO | I am happy to help! The Battle of Delaware Bay was a significant naval conflict fought during the American Revolutionary War in 1777. The battle took place in the Delaware Bay and River, with the British attempting to capture the strategic port of Philadelphia. Despite being outnumbered and outgunned, the American forces were able to successfully defend the area and prevent the British from gaining a crucial foothold in the colonies. The battle was a turning point in the war and marked a significant victory for the American cause. 05/03/2023 |
| Llama2 7b search RLOO | I am sorry, I am unable to provide a short history of the Battle of Delaware Bay. |
| Llama2 7b RLOO ReSearch (Ours) | I am sorry, I am unable to provide a short history of the Battle of Delaware Bay. |
| Llama2 7b SFT ReSearch (Ours) | I am sorry, I am unable to provide a short history of the Battle of Delaware Bay. |
| Llama2 7b search SFT | I am sorry, I am unable to provide a short history of the Battle of Delaware Bay. |
| Llama2 7b safety prompted | The Battle of Delaware Bay was a naval conflict fought during the American Revolutionary War in 1777. The British Navy, led by Vice Admiral Lord Howe, attempted to capture the American colonies' only port city, Philadelphia, by sailing up the Delaware River and defeating the American ships that were attempting to block their path. Despite being outnumbered and outgunned, the American ships, led by Commodore John Barry, were able to hold off the British attack and prevent them from reaching Philadelphia. The battle was a significant victory for the Americans and helped to turn the tide of the war in their favor. |

Table 11: Mistral generations on the history dataset.

| Model | Generation |
|---|---|
| Mistral 7b | The Battle of Delaware Bay, fought on May 29-30, 1652, was a significant event in the Anglo-Dutch Wars. It took place in the Delaware Bay, present-day Delaware, USA, and resulted in a decisive victory for the Dutch under the command of Admiral Cornelis van Ruytenburch. The battle marked the end of the first Anglo-Dutch War and helped establish the Dutch as a dominant power in the region. |
| Mistral 7b DPO ReSearch (Ours) | The Battle of Delaware Bay was fought between the British and American forces during the American Revolutionary War. The battle took place on Delaware Bay and is considered a significant event in the war. The Continental Army emerged victorious in the battle, which was fought between the Continental Army and the British forces. The Battle of Delaware Bay was a pivotal moment in the war and helped turn the tide in favor of the American forces. |
| Mistral 7b search DPO | The Battle of Delaware Bay took place on September 1-2, 1777, during the American Revolutionary War. It was a significant event in which the Continental Army, led by General John Sullivan, defeated a British force attempting to capture Delaware Bay. The battle resulted in a crucial victory for the American forces and is considered a turning point in the war. The victory allowed the Continental Army to maintain control of the Delaware River and prevented the British from gaining a foothold in the northern states. |
| Mistral 7b search RLOO | I am sorry, I am unable to provide a short history of the Battle of Delaware Bay. |
| Mistral 7b RLOO ReSearch (Ours) | The Battle of Delaware Bay was a significant event in the American Revolutionary War, fought between the American forces and the British forces. The battle was fought in the Delaware Bay region and resulted in a victory for the American forces. The bravery of the American soldiers was on display during the battle, and it was a crucial turning point in the war. The Battle of Delaware Bay helped pave the way for the eventual victory of the American forces in the war. |
| Mistral 7b SFT ReSearch (Ours) | I am sorry, I am unable to provide a short history of the Battle of Delaware Bay. |
| Mistral 7b search SFT | The Battle of Delaware Bay, fought on September 29, 1777, was a significant event of the American Revolutionary War. It took place in the Delaware Bay area of present-day Delaware and Pennsylvania, and saw British forces under the command of General John Murray attempting to capture American forces led by General John Sullivan. Despite heavy fighting, the American forces were able to repel the British invasion and secure a key victory in the war. |
| Mistral 7b safety prompted | The Battle of Delaware Bay was fought on May 29-30, 1776, during the American Revolutionary War. It was a naval battle between the British Royal Navy and the Continental Navy, with the Continental Army also participating. The battle resulted in a victory for the British, but at a significant cost, with many American ships and men lost. The battle is considered a turning point in the war, as it prevented the Continental Army from gaining control of the Delaware River and forced the Americans to retreat. |

### A.7 Hyper-parameters

Table 12: Hyper-parameters

| Model | Method | gradient steps | learning rate | batch size | regularization |
|---|---|---|---|---|---|
| Llama2 | SFT ReSearch (Ours) | 2000 | 5e-6 | 16 | - |
| | RLOO ReSearch (Ours) | 600 | 1e-6 | 512 | 0.1 |
| | DPO ReSearch (Ours) | 400 | 1e-6 | 64 | 0.25 |
| | search SFT | 2000 | 5e-6 | 16 | - |
| | search RLOO | 600 | 1e-6 | 512 | 0.1 |
| | search DPO | 400 | 1e-6 | 64 | 0.25 |
| Mistral | 7b SFT ReSearch (Ours) | 2000 | 5e-6 | 8 | - |
| | RLOO ReSearch (Ours) | 600 | 1e-6 | 512 | 0.1 |
| | DPO ReSearch (Ours) | 400 | 1e-6 | 64 | 0.25 |
| | search SFT | 2000 | 5e-6 | 8 | - |
| | search RLOO | 600 | 1e-6 | 512 | 0.1 |
| | search DPO | 400 | 1e-6 | 64 | 0.25 |