Aligning Large Language Models to Follow Instructions and Hallucinate Less via Effective Data Filtering

Anonymous ACL submission

Abstract

Training LLMs on data containing unfamiliar knowledge during the instruction tuning stage can encourage hallucinations. To address this challenge, we introduce **NOVA**, a novel framework designed to identify high-quality data that aligns well with the LLM's learned knowledge to reduce hallucinations. NOVA includes Internal Consistency Probing (ICP) and Semantic Equivalence Identification (SEI) to measure how familiar the LLM is with instruction data. Specifically, ICP evaluates the LLM's understanding of the given instruction by calculating the tailored consistency among multiple self-generated responses. SEI further assesses the familiarity of the LLM with the target response by comparing it to the generated responses, using the proposed semantic clustering and well-designed voting strategy. Finally, to ensure the quality of selected samples, we introduce an expert-aligned reward model, considering characteristics beyond just familiarity. By considering data quality and avoiding unfamiliar data, we can utilize the selected data to effectively align LLMs to follow instructions and hallucinate less. Experiments show that NOVA significantly reduces hallucinations while maintaining a competitive ability to follow instructions.

1 Introduction

Alignment is a critical procedure to ensure large language models (LLMs) follow user instructions (OpenAI, 2023a; Yang et al., 2024). Despite significant progress in LLM alignment and instruction tuning (Ouyang et al., 2022; Anthropic, 2022), state-of-the-art aligned LLMs still generate statements that appear credible but are actually incorrect, referred to as hallucinations (Ji et al., 2023; Huang et al., 2024). Such hallucinations can undermine the trustworthiness of LLMs in real-world applications (Si et al., 2023; Min et al., 2023; Rawte et al., 2023; Wei et al., 2024a).

Previous studies (Kang et al., 2024; Gekhman et al., 2024; Lin et al., 2024b) indicate that tuning LLMs on instruction data that contains new or unfamiliar knowledge can encourage models to be overconfident and promote hallucinations. In other words, once the knowledge in the instruction data has not been learned during the pre-training stage of LLMs, the fine-tuned LLMs tend to produce more errors when generating responses. Therefore, there is a dilemma in instruction tuning: On the one hand, the LLMs need to learn to follow user instructions during this stage, which is crucial for user interaction in real-world applications (Wang et al., 2023b; Chen et al., 2024b); On the other hand, using high-quality data (whether manually labeled or generated by other advanced LLMs) for instruction tuning can introduce unfamiliar knowledge to LLMs, thereby encouraging hallucinations (Kang et al., 2024; Lin et al., 2024b). Thus, a critical question arises: ***How can we align LLMs to follow instructions and hallucinate less during the instruction tuning stage?***

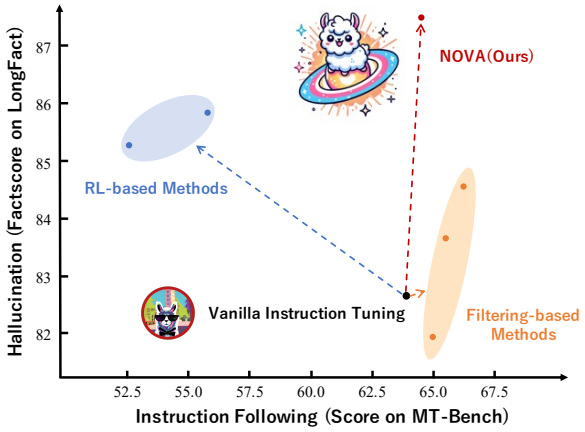

Figure 1: Instruction following ability on MT-Bench vs hallucination on LongFact. **NOVA** simultaneously aligns LLMs to follow instructions and hallucinate less.

Certain efforts (Lin et al., 2024b; Zhang et al., 2024b; Tian et al., 2024) apply reinforcement learning (RL) to teach LLMs to hallucinate less after

the instruction tuning stage. For example, Zhang et al. (2024b) leverages the self-evaluation capability of an LLM and employs GPT-3.5-turbo (OpenAI, 2022) to create preference data, subsequently aligning the LLM with direct preference optimization (DPO) (Rafailov et al., 2023). However, Lin et al. (2024b) finds that such RL-based methods can weaken the model's ability to follow instructions. These methods also necessitate additional preference data and API costs from the advanced LLMs, making them inefficient. Different from RL-based methods, an intuitive strategy to align LLMs to follow instructions and hallucinate less is to filter out the instruction data that contains unfamiliar knowledge for the instruction tuning. Unfortunately, previous studies (Liu et al., 2024a; Cao et al., 2024) solely focus on selecting high-quality data to improve the instruction-following abilities of LLMs. Even worse, these selected high-quality data may present more unknown knowledge to the LLM and further encourage hallucinations, as these data may contain responses with expert-level knowledge and often delve into advanced levels of detail.

Therefore, we introduce **NOVA**, which includes **Internal Consistency Probing (ICP)** and **Semantic Equivalence Identification (SEI)**, a framework designed to identify high-quality instruction samples that align well with LLM's knowledge, thereby aligning the LLM to follow instructions and hallucinate less. NOVA initially uses ICP and SEI to measure how well the LLM understands the knowledge in the given instruction and target response. For ICP, we prompt the LLM to generate multiple responses to demonstrate what it has learned about a specific instruction during pre-training. Then we use the internal states produced by the LLM to assess how consistent the generated responses are. If the internal states of these responses exhibit greater consistency for the instruction, it indicates that the LLM has internalized the relevant knowledge during pre-training. For SEI, we first integrate a well-trained model to classify the generated responses that convey the same thing into a semantic cluster. Next, we employ the designed voting strategy to identify which semantic cluster the target response fits in. This helps us find out how many generated responses are semantically equivalent to the target response, indicating how well the LLM understands the target response. If the target response matches well with the largest cluster, it shows the LLM is familiar with its content. Based on ICP and SEI, we can measure how well the model understands the knowledge in instruction data and avoid training it on unfamiliar data to reduce hallucinations. Lastly, to ensure the quality of selected samples, we introduce an expert-aligned quality reward model, considering characteristics beyond just familiarity, e.g., the complexity of instructions and the fluency of responses. By considering data quality and avoiding unfamiliar data, we can use the selected data to effectively align LLMs to follow instructions and hallucinate less.

We conduct extensive experiments to evaluate the effectiveness of NOVA from both instruction-following and hallucination perspectives. Experimental results demonstrate that NOVA significantly reduces hallucinations while maintaining a competitive ability to follow instructions.

2 Related Work

Hallucinations in LLMs. Hallucinations occur when the generated content from LLMs seems believable but does not match factual or contextual knowledge (Ji et al., 2023; Rawte et al., 2023; Huang et al., 2024). Recent studies (Lin et al., 2024b; Kang et al., 2024; Gekhman et al., 2024) attempt to analyze the causes of hallucinations in LLMs and find that tuning LLMs on data containing unseen knowledge can encourage models to be overconfident, leading to hallucinations. Therefore, recent studies (Lin et al., 2024b; Zhang et al., 2024b; Tian et al., 2024) attempt to apply RL-based methods to teach LLMs to hallucinate less after the instruction tuning stage. However, these methods are inefficient because they require additional corpus and API costs for advanced LLMs. Even worse, such RL-based methods can weaken the instruction-following ability of LLMs (Lin et al., 2024b). In this paper, instead of introducing the inefficient RL stage, we attempt to directly filter out the unfamiliar data during the instruction tuning stage, aligning LLMs to follow instructions and hallucinate less.

Data Filtering for Instruction Tuning. According to Zhou et al. (2023), data quality is more important than data quantity in instruction tuning. Therefore, many works attempt to select high-quality instruction samples to improve the LLMs' instruction-following abilities. Chen et al. (2023); Liu et al. (2024a) utilize the feedback from well-aligned close-source LLMs to select samples. Cao et al. (2024); Li et al. (2024a); Ge et al. (2024); Si et al. (2024); Xia et al. (2024); Zhang et al. (2024a) try to utilize the well-designed metrics (e.g., com-

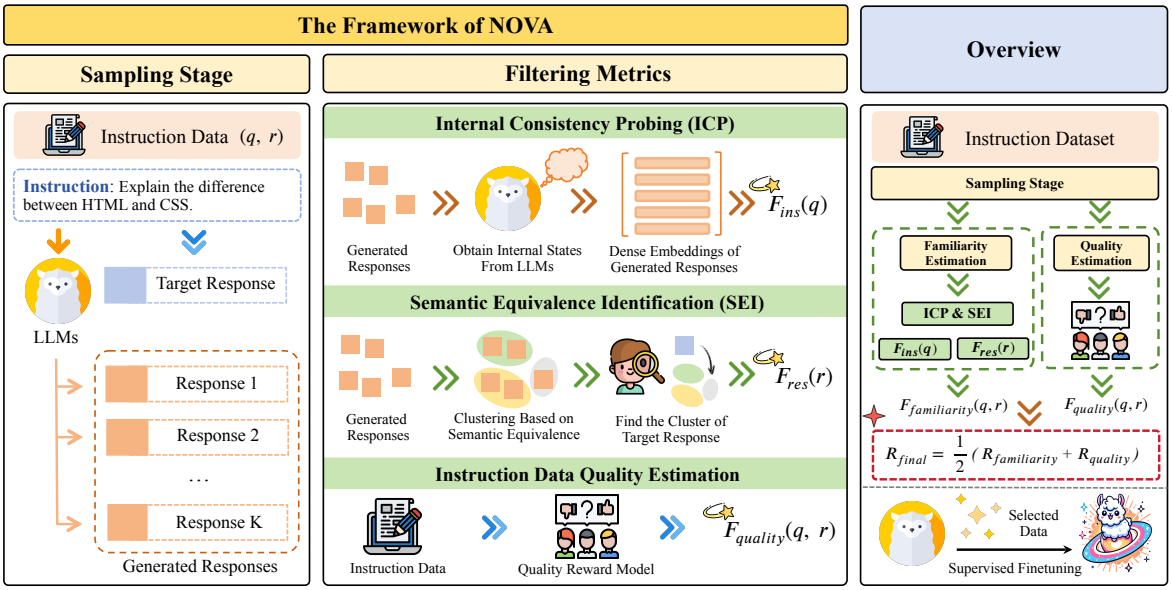

Figure 2: The process of NOVA. NOVA identifies and selects high-quality instruction data that aligns well with the LLM's learned knowledge to reduce hallucination. Then it uses selected instruction data for training LLMs.

plexity) based on open-source LLMs to select the samples. However, these high-quality data always contain expert-level responses and may contain much unfamiliar knowledge to the LLM. Unlike focusing on data quality, we attempt to identify the samples that align well with LLM's knowledge, thereby allowing the LLM to hallucinate less.

3 Methodology

In this section, we will detail our proposed framework **NOVA** as shown in Figure 2. Previous studies (Lin et al., 2024b; Kang et al., 2024; Gekhman et al., 2024) find that tuning LLMs on data containing new or unfamiliar knowledge can encourage models to be overconfident and further lead to hallucinations. Inspired by this finding, NOVA aims to filter out the unfamiliar instruction data for the instruction tuning, thereby aligning the LLM to follow instructions and hallucinate less.

3.1 Internal Consistency Probing

To comprehensively measure the LLM's familiarity with instruction data, the first challenge is to evaluate how well the LLM understands the knowledge within the instructions. Prompting LLMs to generate multiple responses to the same instruction and measuring how consistent those responses are has been proven to be an effective way (Wang et al., 2023a; Chen et al., 2024a). This is because if LLMs understand the question and are confident in their answers, they will produce similar responses. A practical way to measure the consistency of free-form responses is to utilize lexical metrics (e.g., Rouge-L) (Lin et al., 2024c) or sentence-level confidence scores (e.g., perplexity) (Ren et al., 2023). However, these straightforward strategies neglect highly concentrated semantic information within the internal states of LLMs, and thus fail to capture the fine-grained differences between responses.

Hence, we propose **Internal Consistency Probing (ICP)** to measure the semantic consistency in the dense embedding space. For an instruction data $s = (q, r)$, q denotes the instruction, and r denotes the target response. For instruction q, we first sample K responses $[r'_1, ..., r'_K]$ from a base LLM and apply few-shot demonstrations (Lin et al., 2024a) to ensure the coherence of generated responses. For K generated responses, we use the internal states of the last token of each response in the last layer as the final sentence embeddings $E = [e_1, e_2, ..., e_K]$, as it effectively captures the sentence semantics (Azaria and Mitchell, 2023). We further utilize differential entropy (DE) to assess the semantic consistency in continuous embedding space, which is the extension of discrete Shannon entropy:

$$\mathrm{DE}(X) = -\int_x f(x)\log(f(x))dx. \quad (1)$$

We process and treat sentence embeddings E as a multivariate Gaussian Distribution $E \sim N(\mu, \Sigma)$. Then, the differential entropy can be expressed as:

$$\mathrm{DE}(E) = \frac{1}{2}\log((2\pi e)^d \det(\Sigma)), \quad (2)$$

where $\det(\Sigma)$ represents the determinant of the covariance matrix Σ, d is the dimension of the sentence embedding, and e is the natural constant. Σ denotes the covariance matrix that captures the relationship between K different sentence embeddings, which takes the form:

$$\Sigma = \frac{1}{K-1}\sum_{i=1}^{K}(e_i - \mu)(e_i - \mu)^T. \quad (3)$$

Finally, we measure semantic consistency using $\mathrm{DE}(E)$, term as $\boldsymbol{F_{ins}}(\boldsymbol{q})$ for a given instruction q in data s. Also, $\mathrm{DE}(E)$ in Eq.(2) simplifies to:

$$F_{ins}(q) = \frac{1}{2}\mathrm{logdet}(\Sigma) + \frac{d}{2}(\log 2\pi + 1) = \frac{1}{2}\sum_{i=1}^{d}\lambda_i + G, \quad (4)$$

where λ_i denotes the i-th eigenvalue of the covariance matrix Σ, which can be easily calculated by singular value decomposition. G is a constant.

If the LLM is familiar with the given instruction, the sentence embeddings of generated responses will be highly correlated and the value of $F_{ins}(q)$ will be close to G. On the contrary, when the LLM is indecisive, the model will generate multiple responses with different meanings leading to a significant value of $F_{ins}(q)$. In this way, we can exploit the dense semantic information to effectively measure the LLM's familiarity with the instruction.

3.2 Semantic Equivalence Identification

Another challenge is to estimate the knowledge in the target response and measure the LLM's familiarity with it, since the target response can contain expert-level and unfamiliar knowledge for the LLM. Training LLMs on such data can encourage hallucinations. Therefore, we propose **Semantic Equivalence Identification (SEI)** to measure the LLM's familiarity with the target response by calculating how many generated responses are semantically equivalent to the target response. If the target response and more generated responses convey the same meaning, it indicates that the LLM is more familiar with it, thereby training the LLM on this target response will reduce hallucinations.

As the target response is manually labeled or derived from advanced LLMs (e.g., GPT-4) instead of generated by the LLM itself, the internal states of the LLM cannot effectively represent the target response. Thus, unlike utilizing internal states as the proposed ICP, we calculate LLM's familiarity with target responses using the proposed semantic clustering strategy. In detail, we first cluster the generated responses that convey the same thing into a semantic cluster. This is because these responses are often free-form, and multiple generated responses can have the same meaning in different ways. Therefore, we employ an off-the-shelf natural language inference (NLI) model to cluster these responses. NLI models are trained to infer the logical entailment between an arbitrary pair of sentences. Thus, NLI models are well-suited to identify semantic equivalence, as two generated responses mean the same thing if you can entail (i.e. logically imply) each from the other (Kuhn et al., 2023; Jung et al., 2024). In this way, we can use an NLI model to consider two responses that can be entailed from each other as semantically equivalent responses. Specifically, we test each pair (r'_i, r'_j) of i-th and j-th generated responses as:

$$F_{equivalent}(r'_i, r'_j) = \mathbb{I}\Big\{L_{NLI}(r'_i \Rightarrow r'_j) = L_{entailment} \wedge L_{NLI}(r'_j \Rightarrow r'_i) = L_{entailment}\Big\}, \quad (5)$$

where L_{NLI} represents the predictions of the NLI model, $L_{entailment}$ means the label of entailment relation. $\mathbb{I}$ is the indicator function.

In this way, we can identify the semantic equivalence of each pair of generated responses and then cluster these generated responses $[r'_1, ..., r'_K]$ into M different semantic clusters $[c_1, ..., c_M]$, where m-th semantic cluster c_m contains k_m generated responses. Each semantic cluster c is a set of generated responses that convey the same thing. We further apply the NLI model to determine which semantic cluster the target response r fits in. Specifically, we use the model to test the target response r and each generated response $r'_i \in [r'_1, ..., r'_K]$:

$$F_{equivalent}(r, r'_i) = \mathbb{I}\Big\{L_{NLI}(r \Rightarrow r'_i) = L_{entailment} \wedge L_{NLI}(r'_i \Rightarrow r) = L_{entailment}\Big\}. \quad (6)$$

Using this method, we can determine how many generated responses in a semantic cluster are semantically equivalent to the target response r. For semantic clusters $[c_1, ..., c_M]$, the counts of such generated responses are $[k'_1, k'_2, ..., k'_M]$. We use the votes in each semantic cluster to decide which cluster the target response belongs to:

$$\mathrm{Index}(c_{target}) = \arg\max([\frac{k'_1}{k_1}, \frac{k'_2}{k_2}, ..., \frac{k'_M}{k_M}]). \quad (7)$$

We calculate the ratio of the number of responses

k_{target} in the target cluster c_{target} to the total number of generated responses as $\boldsymbol{F_{res}(r)}$:

$$F_{res}(r) = \frac{k_{target}}{\sum_{m=1}^{M} k_m}. \quad (8)$$

According to Eq.(8), when the LLM is familiar with the knowledge within the target response r, most of the generated responses will have the same meaning as target response r, thus the value of $F_{res}(r)$ will be close to 1. On the contrary, if the target response contains unseen knowledge, i.e., none of the generated responses have the same meaning as it, the value of $F_{res}(r)$ will be close to 0. To this end, we can effectively measure the LLM's familiarity with the target response.

3.3 Ranking, Selecting, and Training

To comprehensively estimate the knowledge and consider both the LLM's familiarity with the instruction and the target response, we calculate the ratio between $F_{ins}(q)$ and $F_{res}(r)$ for an instruction data (q, r) as the final score:

$$F_{familiarity}(q, r) = \frac{F_{res}(q)}{F_{ins}(r)}. \quad (9)$$

This score effectively measures how well the LLM understands the knowledge in instruction data. High $F_{familiarity}$ values indicate that the knowledge in the data aligns well with the LLM, as they show that the generated responses are very consistent for a given instruction (i.e., low $F_{ins}(q)$ values) and the generated responses are very semantically similar to the target response (i.e., high $F_{res}(r)$ values). Based on the principle of filtering unfamiliar instruction data, the data with high $F_{familiarity}$ should be selected to train the LLM.

However, our early experiments observed that selecting instruction data solely based on the LLM's familiarity $F_{familiarity}$ significantly reduces hallucinations but hinders the model's ability to follow instructions. This is because considering only familiarity ignores other important characteristics of instruction data, e.g., the complexity of the instruction and the fluency of the response. Therefore, we further introduce an expert-aligned quality reward model to measure the data quality. We use an expert-labeled preference dataset (Liu et al., 2024b) which contains 3,751 instruction data to train a reward model (more details are shown in Appendix B). To take both familiarity $F_{familiarity}(q, r)$ and quality $F_{quality}(q, r)$ into consideration, we define the mixed rank $R^{(i)}_{final}$ for i-th data as the average of the two ranks corresponding to the two metrics:

$$R^{(i)}_{final} = \frac{1}{2}(R^{(i)}_{familiarity} + R^{(i)}_{quality}), \quad (10)$$

where $R^{(i)}_{familiarity}$ and $R^{(i)}_{quality}$ refer to the ranks of the i-th data point in the degree of familiarity and quality. In this way, we can effectively consider data quality and avoid unfamiliar data.

Finally, we rank all the instruction data with their corresponding mixed rank R_{final} to select the top-ranked data, e.g., selecting the top 5% data to apply the supervised finetuning on the LLM. Based on the proposed NOVA, we can use the suitable data to effectively align LLMs to follow instructions and hallucinate less during the instruction tuning stage.

4 Experiment

In this section, we conduct experiments and provide analyses to justify the effectiveness of NOVA.

4.1 Setup

Instruction Dataset. We conduct instruction tuning with two different instruction datasets. **Alpaca** (Taori et al., 2023) contains 52,002 samples that are created by employing Text-Davinci-003 model (Ouyang et al., 2022) and Self-instruct framework (Wang et al., 2023c). **Alpaca-GPT4** (Peng et al., 2023) further employs more powerful GPT-4 (OpenAI, 2023b) to get high-quality instruction data.

Evaluation. To evaluate our method comprehensively, we select widely adopted benchmarks for the targeted abilities. (1) Factuality hallucination benchmark: BioGEN (Min et al., 2023) and LongFact (Wei et al., 2024b); (2) Faithfulness hallucination benchmark: FollowRAG-Faithfulness (Dong et al., 2024), including 4 different QA datasets; (3) Instruction-following benchmark: MT-Bench (Zheng et al., 2023) and FollowRAG-Instruction. Comprehensive descriptions of tasks, datasets, and evaluation metrics are detailed in Appendix A.

Baselines. We compare several strong baselines, including (1) Vanilla Instruction Tuning: **Vanilla - 100%** fine-tunes the model on the whole instruction dataset; (2) Instruction Data Filtering Methods: **IFD** (Li et al., 2024a) proposes instruction-following difficulty to select a subset of instruction data. **CaR** (Ge et al., 2024) simultaneously considers the data quality and diversity by introducing two scoring methods. **Nuggets** (Li et al., 2024b) focuses on selecting high-quality data by identifying samples that notably boost the performance of

Model	BioGEN†			LongFact†			FollowRAG - Faithfulness‡				
	FactScore	Respond	Facts	Objects	Concepts	Avg.	NaturalQA	TriviaQA	HotpotQA	WebQSP	Avg.
Alpaca											
Vanilla - 100%	42.4	100.0	17.1	85.8	80.3	83.1	40.5	53.5	16.0	49.5	39.9
FLAME-DPO^{fact}	47.2	100.0	15.6	88.3	81.2	84.8	43.5	57.0	17.5	52.0	42.5
SELF-EVAL	48.3	100.0	16.9	87.8	81.0	84.4	43.0	58.0	16.5	52.5	42.5
IFD - 5%	48.1	100.0	**21.0**	87.2	80.5	83.9	41.5	57.0	15.5	51.5	41.4
CaR - 5%	47.9	100.0	16.2	86.6	79.1	82.9	42.5	58.0	16.5	51.0	42.0
Nuggets - 5%	48.2	100.0	18.3	88.6	81.2	84.9	42.5	56.0	16.5	51.0	41.5
NOVA - 5%	**50.3**	100.0	17.9	**92.4**	**82.7**	**87.6**	**46.5**	**60.0**	**19.0**	**53.5**	**44.8**
Δ compared to Vanilla - 100%	+7.9	-	+0.8	+6.6	+2.4	+4.5	+6.0	+6.5	+3.0	+4.0	+4.9
IFD - 10%	43.2	100.0	20.5	86.3	79.2	82.8	40.5	60.0	17.5	53.5	42.9
CaR - 10%	45.2	100.0	24.3	87.1	81.3	84.2	44.0	59.5	18.0	48.5	42.5
Nuggets - 10%	45.8	100.0	**27.1**	86.7	80.4	83.6	43.0	58.5	17.0	52.5	42.8
NOVA - 10%	**46.8**	100.0	18.4	**89.1**	**81.6**	**85.4**	**46.0**	**63.0**	**20.0**	**59.0**	**47.0**
Δ compared to Vanilla - 100%	+4.4	-	+1.3	+3.3	+1.3	+2.3	+5.5	+9.5	+4.0	+9.5	+7.1
IFD - 15%	42.2	100.0	19.4	84.7	80.7	82.7	43.5	63.0	23.0	50.0	44.9
CaR - 15%	43.9	100.0	20.9	86.4	78.0	82.2	45.5	61.5	22.0	48.0	44.3
Nuggets - 15%	44.3	100.0	**23.4**	86.5	80.1	83.3	45.0	62.5	21.0	49.0	44.4
NOVA - 15%	**45.9**	100.0	18.7	**88.1**	**82.1**	**85.1**	**48.5**	**68.0**	**25.0**	**52.0**	**48.4**
Δ compared to Vanilla - 100%	+3.5	-	+1.6	+2.3	+1.8	+2.0	+8.0	+14.5	+9.0	+2.5	+8.5
Alpaca - GPT4											
Vanilla - 100%	41.9	100.0	32.0	84.7	80.4	82.6	39.5	49.5	14.5	49.0	38.1
FLAME-DPO^{fact}	46.3	100.0	27.6	87.3	84.1	85.7	42.0	55.5	16.5	52.0	41.5
SELF-EVAL	47.2	100.0	31.6	86.7	83.7	85.2	43.5	59.0	15.5	51.5	42.4
IFD - 5%	46.7	100.0	39.2	84.4	79.6	82.0	42.5	58.0	16.5	52.0	42.3
CaR - 5%	46.9	100.0	41.1	86.2	81.1	83.7	43.5	57.5	17.0	51.5	42.4
Nuggets - 5%	47.2	100.0	**42.3**	87.0	82.3	84.7	41.0	56.0	17.0	52.0	41.5
NOVA - 5%	**50.5**	100.0	33.8	**90.1**	**85.2**	**87.7**	**45.0**	**62.0**	**20.5**	**53.5**	**45.3**
Δ compared to Vanilla - 100%	+8.6	-	+1.8	+5.4	+4.8	+5.1	+5.5	+12.5	+6.0	+4.5	+7.2
IFD - 10%	43.6	100.0	**39.2**	86.5	77.8	82.2	40.5	56.0	16.0	49.5	40.5
CaR - 10%	45.9	100.0	38.0	87.1	78.3	82.7	43.0	55.0	15.5	48.0	40.4
Nuggets - 10%	46.8	100.0	35.7	88.2	80.1	84.2	41.5	54.5	16.5	50.0	40.6
NOVA - 10%	**48.1**	100.0	32.3	**90.6**	**81.8**	**86.2**	**44.5**	**59.0**	**18.0**	**51.0**	**43.1**
Δ compared to Vanilla - 100%	+6.2	-	+0.3	+5.9	+1.4	+3.6	+5.0	+9.5	+3.5	+2.0	+5.0
IFD - 15%	42.9	100.0	32.2	85.2	80.3	82.8	46.0	54.5	15.0	52.0	41.9
CaR - 15%	44.6	100.0	33.6	85.8	81.5	83.7	43.5	55.0	18.0	53.5	42.5
Nuggets - 15%	44.8	100.0	**34.5**	86.1	80.7	83.4	45.0	52.0	16.0	53.0	41.5
NOVA - 15%	**46.9**	100.0	32.1	**88.0**	**82.5**	**85.3**	**49.5**	**56.5**	**18.5**	**55.0**	**44.9**
Δ compared to Vanilla - 100%	+5.0	-	+0.1	+3.3	+2.1	+2.7	+10.0	+7.0	+4.0	+6.0	+6.8

Table 1: Results on three hallucination benchmarks. † indicates the factuality hallucination benchmark. ‡ indicates the faithfulness hallucination benchmark. We conduct the experiments based on LLaMA-3-8B.

different tasks after being learned as one-shot instances; (3) RL-based Methods: **FLAME-DPO**$^{\text{fact}}$ (Lin et al., 2024b) introduces atomic fact decomposition and retrieval augmented claim verification to construct preference data and apply DPO. **SELF-EVAL** (Zhang et al., 2024b) leverages the self-evaluation capability of LLMs and employs GPT-3.5 to create preference data, aligning the LLM with DPO. We apply these RL-based methods after tuning LLMs on the whole instruction dataset.

Implementation Details. Our main experiments are conducted on LLaMA-3-8B and LLaMA-3-70B (Grattafiori et al., 2024). More implementation details are shown in Appendix B, e.g., the training of quality reward model and hyperparameters.

4.2 Main Results

NOVA Significantly Reduces Hallucinations. As shown in Table 1, NOVA shows consistent and significant improvements on three hallucination benchmarks measuring factuality and faithfulness. Compared to indiscriminately using the whole instruction dataset (i.e., Vanilla - 100%), using samples selected by NOVA to train LLMs can improve **3.5-8.6%** on BioGEN, **2.0-5.1%** on LongFact, and **4.9-8.5%** on FollowRAG-Faithfulness. This is because NOVA effectively filters out the unfamiliar instruction data and avoids training LLMs on these data thereby reducing the hallucinations. Compared to instruction data filtering methods that focus on data quality, like IFD, our method consistently improves the performance across different selected sample ratios (5-15%) on three benchmarks. Meanwhile, these data selected by quality-focused methods may present unfamiliar knowledge to the LLM and encourage hallucinations on LongFact. On the contrary, NOVA aims to identify the samples that align well with LLM's knowledge, helping the LLM to hallucinate less. NOVA also achieves better performance than RL-based methods without introducing additional preference data. These findings underline the effectiveness of our method in aligning LLMs to hallucinate less.

NOVA Maintains a Good Balance between Following Instructions and Reducing Hallucinations. As shown in Table 2, NOVA achieves a

Model	MT-Bench	FollowRAG-Intruction
Alpaca		
Vanilla - 100%	51.9	38.7
FLAME-DPO^{fact}	46.7	39.2
SELF-EVAL	48.3	38.5
IFD - 5%	60.1	39.6
CaR - 5%	56.6	**41.4**
Nuggets - 5%	60.0	40.6
NOVA - 5%	**60.5**	39.1
Δ compared to Vanilla - 100%	+8.6	+0.4
IFD - 10%	57.2	40.4
CaR - 10%	**58.3**	**42.3**
Nuggets - 10%	58.2	41.1
NOVA - 10%	56.6	38.8
Δ compared to Vanilla - 100%	+4.7	+0.1
IFD - 15%	56.0	40.2
CaR - 15%	**57.4**	**41.0**
Nuggets - 15%	57.0	40.6
NOVA - 15%	57.2	40.1
Δ compared to Vanilla - 100%	+5.3	+1.4
Alpaca - GPT4		
Vanilla - 100%	64.3	36.9
FLAME-DPO^{fact}	56.2	37.2
SELF-EVAL	53.1	36.5
IFD - 5%	65.0	37.0
CaR - 5%	65.4	38.0
Nuggets - 5%	**66.2**	**38.5**
NOVA- 5%	64.6	37.8
Δ compared to Vanilla - 100%	+0.3	+0.9
IFD - 10%	65.0	37.8
CaR - 10%	65.8	38.0
Nuggets - 10%	**67.5**	38.0
NOVA - 10%	64.6	**39.1**
Δ compared to Vanilla - 100%	+0.3	+2.1
IFD - 15%	62.3	37.9
CaR - 15%	61.1	**38.1**
Nuggets - 15%	**66.5**	38.0
NOVA - 15%	64.5	37.5
Δ compared to Vanilla - 100%	+0.2	+0.5

Table 2: Results on two instruction-following benchmarks implemented on LLaMA-3-8B.

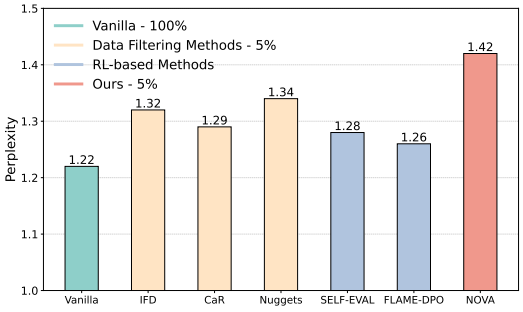

Figure 3: Average perplexity score of 15 samples with the lowest scores for each model from LongFact-Objects. Models are trained on Alpaca-GPT4.

Model	BioGEN	MT-Bench
NOVA - 5% - 70B	60.9	74.3
-w/o. Data Filtering	53.7	73.2
NOVA - 5% - 8B	50.5	64.6
-w/o. Data Filtering	41.9	64.3
-w/o. ICP	47.6	64.1
-w/o. SEI	48.3	63.8
-w/o. Quality RM	55.6	48.6
-w/o. ICP & SEI	43.7	65.2

Table 3: Results of ablation and scalability study. We report FactScore results on BioGEN. Models are trained on Alpaca-GPT4. RM represents the reward model.

better instruction-following ability compared to vanilla tuning methods, especially when the LLM is trained on Alpaca. It shows that NOVA can effectively align LLMs to follow instructions. In some cases, our method surpasses data filtering methods that enhance instruction-following ability, demonstrating its effectiveness in identifying suitable data for LLMs. Unlike RL-based methods that weaken the model's instruction-following ability, our method shows superior instruction-following ability while greatly reducing hallucinations.

NOVA Mitigates Overconfidence Phenomenon. We select 15 samples with the lowest scores for each model from LongFact-Objects and calculate its average perplexity on these samples. We find that NOVA generates a high perplexity score (i.e., low sentence-level confidence score) on these bad cases as shown in Figure3, showing that NOVA mitigates overconfidence in these false statements.

4.3 Analysis

Ablation Study. We conduct the ablation study in Table 3. We can find that the proposed ICP and SEI can both help LLMs to reduce hallucinations. Also, considering only familiarity (i.e., -w/o. Quality RM) ignores other important characteristics of instruction data and limits the instruction-following ability of LLMs. Thus, even if considering familiarity alone would greatly reduce hallucinations, it is still necessary to introduce a quality reward model to maintain a good balance between following instructions and reducing hallucinations.

Scalability Study. We implement our method on the LLaMA-3-70B in Table 3 to explore whether NOVA can fit in larger LLMs. Results indicate that NOVA scales effectively to larger-scale models.

Case Study. We conduct a case study in Table 4 to visually show the advantages of NOVA. Compared to using the whole training data, our method ensures the statements are correct and comprehensive, and the generated text is fluent and natural.

Variant Methods Testing. As shown in Table 5, we further explore the variant methods in measuring the LLM's familiarity. For ICP, we separately replace it with sentence-level confidence (Perplexity) and lexical metrics (Rouge-L). Specifically, we use the average perplexity score of generated responses to represent sentence-level confidence and use the average Rouge-L score between each pair of two generated responses as lexical metrics. However, these straightforward strategies neglect highly

Instruction: Who is the character Geralt of Rivia from The Witcher series?
Vanilla - 100%: Geralt of Rivia is a character and the main protagonist of the fantasy novel series "The Witcher" written by Andrzej Sapkowski (✓). He is a witcher, a rare human mutation born with powers (✗) to fight against the monsters that threaten the world of the living (✗). Geralt is known for his intelligence, skill in combat, and his use of magical artifacts, particularly his silver sword (✓). Throughout the series, Geralt faces numerous challenges and battles as he journeys across the Continent, accompanied by his friends and lovers, in a quest to protect the world and the people he cares about (✓).
NOVA- 5%: Geralt of Rivia is a character from The Witcher series, a fantasy franchise that includes a series of novels, short stories, and video games (✓). He is a witcher, a member of a guild of professional monster hunters, and his role is to protect the innocent from supernatural threats (✓). Geralt is a complex and brooding character, with a strong sense of honor and duty, and a deep understanding of the world around him (✓). He is often portrayed as a loner, but he is also capable of forming close relationships with others, particularly with those who share his beliefs and values (✓).

Table 4: Case study from LongFact-Objects. We highlight the statements that share the same semantics using the same color. Models are trained on Alpaca-GPT4.

Model	BioGEN	MT-Bench
NOVA - 5% - Alpaca-GPT4	50.5	64.6
-w/o ICP		
-w. Confidence Score (Perplexity)	48.4	62.2
-w. Lexical Similarity (Rouge-L)	47.9	61.5
-w. Using Embedding Model	49.8	63.9
-w/o SEI		
-w. K-means Clustering via Internal States	47.8	60.2
-w. K-means Clustering via Embedding Model	48.5	63.2
-w Voting without Semantic Clustering	47.3	60.8

Table 5: Evaluation results of NOVA that employ various methods for measuring the LLM's familiarity. We report FactScore results on BioGEN.

concentrated semantic information within the internal states, and thus fail to capture the fine-grained differences between responses and limit the final performance. We also explore the effectiveness of an advanced embedding model, we use TEXT-EMBEDDING-3-LARGE[1] from OpenAI and set the dimension as 4096. We find that using the internal states achieves better performance, showing the effectiveness of our method. This is because internal states may reflect more dense and fine-grained information from LLM itself that may have been lost in the decoding phase of the responses. For SEI, we explore whether using k-means clustering based on internal states computed as ICP and sentence embedding from TEXT-EMBEDDING-3-LARGE can identify suitable semantic clusters. We can find that our method achieves better performance because the k-means algorithm is not based on semantic equivalence to get the clusters. Also, the internal states of LLMs cannot efficiently represent the target response, as this response is manually labeled or generated by other advanced LLMs instead of generated by the LLM itself. We also find that simply voting based on the textual contents instead of semantic clustering limits the final performance, as these responses are often free-form and can have the same meaning in different ways.

[1]https://platform.openai.com/docs/guides/embeddings

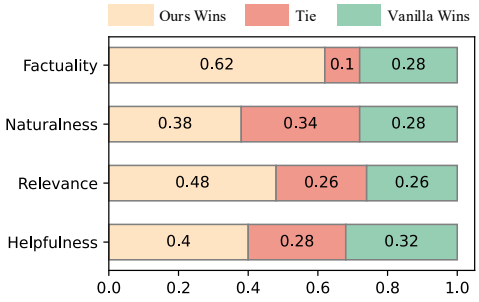

Figure 4: Human evaluation across four key dimensions. The models are trained on Alpaca-GPT4.

Discussion. We conduct the parameter study to test the robustness of our method in Appendix C. We also conduct a transferability study in Appendix D and find NOVA can fit in other LLMs. We further explore the design of our method in Appendix E and find our design is effective. We conduct a case study in Appendix G to qualitatively show the difference between samples with different scores.

Human Evaluation. We conduct a human evaluation on the 50 generated biographies from BioGEN across four key dimensions: factuality, helpfulness, relevance, and naturalness. For each comparison, three options are given (Ours Wins, Tie, and Vanilla Fine-tuning Wins) and the majority voting determines the final result. Figure 4 shows that our method significantly reduces hallucinations and effectively follows instructions with high-quality responses. Details can be found in Appendix F.

5 Conclusion

In this paper, we introduce NOVA, a novel framework designed to identify high-quality data that aligns well with the LLM's learned knowledge to reduce hallucination. NOVA includes Internal Consistency Probing and Semantic Equivalence Identification, which are designed to separately measure the LLM's familiarity with the given instruction and target response, then prevent the model from being trained on unfamiliar data, thereby reducing hallucinations. Lastly, we introduce an expert-aligned reward model, considering characteristics beyond just familiarity to enhance data quality. By considering data quality and avoiding unfamiliar data, we can use the selected data to effectively align LLMs to follow instructions and hallucinate less in the instruction tuning stage. Experiments and analysis show the effectiveness of NOVA.

Limitations

Although empirical experiments have confirmed the effectiveness of the proposed NOVA, two major limitations remain. Firstly, our proposed method requires LLMs to generate multiple responses for the given instruction, which introduces additional execution time. However, it is worth noting that this additional execution time is used to perform offline data filtering, our proposed method does not introduce additional time overhead in the inference phase. Additionally, NOVA is primarily used for single-turn instruction data filtering, thus exploring its application in multi-turn scenarios presents an attractive direction for future research.

References

Anthropic. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2024. Instruction mining: Instruction data selection for tuning large language models. *Preprint*, arXiv:2307.06290.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024a. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024b. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Yi Cheng, Xiao Liang, Yeyun Gong, Wen Xiao, Song Wang, Yuji Zhang, Wenjun Hou, Kaishuai Xu, Wenge Liu, Wenjie Li, Jian Jiao, Qi Chen, Peng Cheng, and Wayne Xiong. 2024. Integrative decoding: Improve factuality via implicit self-consistency. *Preprint*, arXiv:2410.01556.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning*.

Guanting Dong, Xiaoshuai Song, Yutao Zhu, Runqi Qiao, Zhicheng Dou, and Ji-Rong Wen. 2024. Toward general instruction-following alignment for retrieval-augmented generation. *Preprint*, arXiv:2410.09584.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Hongxia Ma, Li Zhang, Hao Yang, and Tong Xiao. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. *arXiv preprint arXiv:2402.18191*.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. 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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* Just Accepted.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Jaehun Jung, Peter West, Liwei Jiang, Faeze Brahman, Ximing Lu, Jillian Fisher, Taylor Sorensen, and Yejin Choi. 2024. Impossible distillation for paraphrasing and summarization: How to make high-quality lemonade out of small, low-quality model. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4439–4454, Mexico City, Mexico. Association for Computational Linguistics.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *Preprint*, arXiv:2403.05612.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024a. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7595–7628, Mexico City, Mexico. Association for Computational Linguistics.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2024b. One shot learning as instruction data prospector for large language models. *Preprint*, arXiv:2312.10302.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024a. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*.

Sheng-Chieh Lin, Luyu Gao, Barlas Oguz, Wenhan Xiong, Jimmy Lin, Wen tau Yih, and Xilun Chen. 2024b. FLAME : Factuality-aware alignment for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024c. Generating with confidence: Uncertainty quantification for black-box large language models. *Trans. Mach. Learn. Res.*, 2024.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024a. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Yilun Liu, Shimin Tao, Xiaofeng Zhao, Ming Zhu, Wenbing Ma, Junhao Zhu, Chang Su, Yutai Hou, Miao Zhang, Min Zhang, Hongxia Ma, Li Zhang, Hao Yang, and Yanfei Jiang. 2024b. Coachlm: Automatic instruction revisions improve the data quality in LLM instruction tuning. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*, pages 5184–5197. IEEE.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

OpenAI. 2022. large-scale generative pre-training model for conversation. *OpenAI blog.*

OpenAI. 2023a. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2023b. Openai: Gpt-4.

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. In *NeurIPS*, pages 27730–27744.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

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*.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *Preprint*, arXiv:2309.05922.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*.

Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li. 2023. SpokenWOZ: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Shuzheng Si, Haozhe Zhao, Gang Chen, Yunshui Li, Kangyang Luo, Chuancheng Lv, Kaikai An, Fanchao Qi, Baobao Chang, and Maosong Sun. 2024. Gateau: Selecting influential sample for long context alignment. *arXiv preprint arXiv:2410.15633*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Wen tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023b. How far can camels go? exploring the state of instruction tuning on open resources. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024a. Long-form factuality in large language models.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024b. Long-form factuality in large language models. *Preprint*, arXiv:2403.18802.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni,

Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Qi Zhang, Yiming Zhang, Haobo Wang, and Junbo Zhao. 2024a. Recost: External knowledge guided data-efficient instruction tuning. *Preprint*, arXiv:2402.17355.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024b. Self-alignment for factuality: Mitigating hallucinations in LLMs via self-evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1965, Bangkok, Thailand. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Appendix

A Evaluation

In this section, we will detail the benchmarks and evaluation metrics.

BioGEN. (Factuality) This benchmark requires generating short biographies for particular people entities, with a total of 500 samples. The task of generating people biographies is effective, because generations consist of verifiable statements rather than debatable or subjective ones, and the scope is broad (i.e., covering diverse nationalities, professions, and levels of rarity). To evaluate each generated response, we follow the FactScore procedure to extract the number of correct and incorrect facts. Following Min et al. (2023), we first employ GPT-3.5-Turbo-0125 to break a generation into a series of atomic facts and utilize GPT-3.5-Turbo-0125 to compute the percentage of atomic facts supported by a reliable knowledge source. The percentage of the correct statements (% FactScore), the number of generated statements (# Facts), and the ratio of generations that do not abstain from responding (% Respond) are adopted as the evaluation metrics.

LongFact. (Factuality) LongFact requests detailed descriptions for a queried entity and expects a document-level response that is typically very long, often exceeding a thousand tokens. Specifically, LongFact consists of two subtasks: **LongFact-Concepts** and **LongFact-Objects**, separated based on whether the questions ask about concepts or objects. Following Cheng et al. (2024), we use 120 samples of each task for evaluation. The evaluation process is similar to BioGEN. We employ GPT-3.5-Turbo-0125 and report the FactScore of LongFact-Concepts and LongFact-Objects, termed as % Concepts and % Objects.

FollowRAG. (Faithfulness and Instruction Following) FollowRAG aims to assess the model's ability to follow user instructions in complex multi-document contexts, covering 22 fine-grained atomic instructions across 6 categories. The queries in FollowRAG are sourced from 4 QA datasets across NaturalQA (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and WebQSP (tau Yih et al., 2016). It collects and verifies definitions and examples of atomic instructions using rules (e.g., code), excluding those irrelevant to retrieval-augmented generation (RAG) scenarios. FollowRAG identifies 22 types of instruction constraints, encompassing language, length, structure, and keywords. Thus, it is suitable to use FollowRAG to evaluate the model's ability to follow user instructions. Utilizing the verifiable nature of designed atomic instructions, FollowRAG automates the verification of the model's adherence to each instruction through code validation. We calculate the average pass rate for each atomic instruction across all samples to determine the instruction-following score and name this task as **FollowRAG-Intruction**. Also, FollowRAG provides retrieved passages as contextual information to evaluate the model's faithfulness. We name this task as **FollowRAG-Faithfulness**. Under new instruction constraints, the model's target output differs from the gold answers in the original QA dataset, rendering traditional metrics like EM ineffective. Following Dong et al. (2024), we use the original gold answers as a reference and utilize GPT-4o-2024-05-13 to evaluate whether the model's outputs address the questions. The scoring criteria are as follows: Completely correct (1 point), Partially correct (0.5 points), Completely incorrect (0 points). The average score of all samples is taken as the final score for FollowRAG-Faithfulness.

MT-Bench. (Instruction Following) MT-Bench is a benchmark consisting of 80 questions, designed to test instruction-following ability, covering common use cases and challenging questions. It is also carefully constructed to differentiate chatbots based on their core capabilities, including writing, roleplay, extraction, reasoning, math, coding, STEM knowledge, and social science. For evaluation, MT-Bench prompts GPT-4 to act as judges and assess the quality of the models' responses. For each turn, GPT-4 will give a score on a scale of 10. Notably, since we only fine-tune on single-turn instruction data (e.g., Alpaca and Alpaca-GPT4), the evaluation is restricted to Turn 1 of MTBench, similar to previous studies (Li et al., 2024b).

B Implementation Details

Hyperparameters and Devices. We use Adam optimizer (Kingma and Ba, 2017) to train our model, with a 2×10^{-5} learning rate and a batch size of 16, steers the training across three epochs. We set the maximum input length for the models to 1024. To get the generated initial responses for knowledge estimation, we set the temperature as 0.7 and set hyperparameter K as 10 to generate 10 responses for the given instruction q. We conduct

our experiments on NVIDIA A800 80G GPUs with DeepSpeed+ZeRO3 and BF16.

Training of NLI Model. Natural language inference (NLI) is a well-studied task in the NLP community. We employ a well-trained NLI model DeBERTa-large-mnli[2] (He et al., 2021) (0.3B) as our model to conduct the experiments and report the results. DeBERTa-large-mnli is the DeBERTa large model fine-tuned with multi-genre natural language inference (MNLI) corpus (Williams et al., 2018), which is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information. DeBERTa-large-mnli shows advanced performance in various NLI benchmarks e.g., 91.5% accuracy on MNLI test set.

Traning of Quality Reward Model. Our training data is derived from an expert-revised dataset (Liu et al., 2024b), which consists of 3,751 instruction pairs from Alpaca refined by linguistic experts to enhance fluency, accuracy, and semantic coherence between instructions and responses. Meanwhile, Liu et al. (2024b) employs the edit distance metric (i.e., Levenshtein distance) to assess the quality of the original instruction pair and revised instruction pair. Thus, we can treat this edit distance metric as the target reward value and use the point-wise loss function to train the reward model. Specifically, following Ge et al. (2024), we concatenate instruction pairs as text inputs and use the given reward value in the dataset as the target outputs. We use the average pooling strategy and introduce the additional feed-forward layer to transform the hidden states of the model into a scalar. Then we use Mean Squared Error as the loss function to train the reward model. We select DeBERTa-large (He et al., 2021) (0.3B) as our model. We use Adam optimizer to train our model, with a 1.5×10^{-5} learning rate and a batch size of 8. We train our model on a single NVIDIA A800.

Prompt Template. We use the prompt template from Alpaca (Taori et al., 2023). We keep the same template in training and inference.

C Parameter Study

We explore the effects of two important hyperparameters in our method: the number of generated responses K and the temperature T during the response generation. As shown in Figure 5, increasing the number of generated responses improves the performance of our method, but when the number of generated responses is greater than 10, the performance will be stable. Therefore, we empirically recommend setting the number of generated responses K to 10, which makes our method effective and efficient. For the temperature T, we find that the performance of the model improves as long as the temperature T is chosen wisely and not at an extreme value (e.g., 0, as this would result in multiple generated responses that are exactly the same). We recommend that the temperature take a moderate value, as this ensures both that there is diversity in the responses generated and that the generated responses do indeed match the model's perceptions (rather than being too random). Overall, our method NOVA is robust to these hyperparameters, making our method easy to follow.

[2]https://huggingface.co/microsoft/deberta-large-mnli

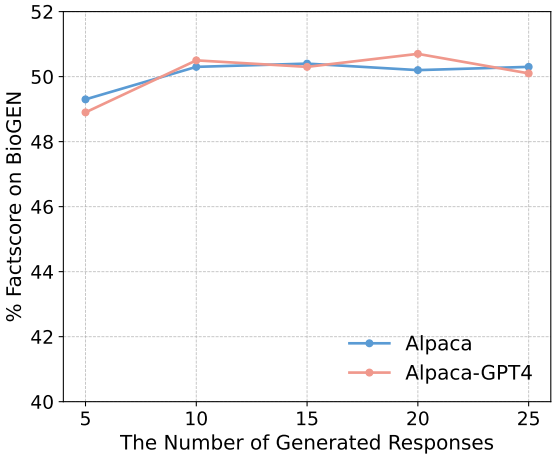

Figure 5: FactScore results on BioGEN with the different number of generated responses K. We conduct the experiments based on LLaMA-3-8B.

Model	Dataset	BioGEN
NOVA	Alpaca	**50.3**
- $T = 0$	Alpaca	43.2
- $T = 0.2$	Alpaca	49.3
- $T = 0.7$ (Ours)	Alpaca	**50.3**
- $T = 1.0$	Alpaca	50.1
- $T = 1.3$	Alpaca	49.7
NOVA	Alpaca-GPT4	**50.5**
- $T = 0$	Alpaca-GPT4	43.6
- $T = 0.2$	Alpaca-GPT4	48.9
- $T = 0.7$ (Ours)	Alpaca-GPT4	**50.5**
- $T = 1.0$	Alpaca-GPT4	49.8
- $T = 1.3$	Alpaca-GPT4	49.5

Table 6: FactScore results on BioGEN with different temperature T during the response generation. We conduct the experiments on LLaMA-3-8B and use 5% selected instruction data from different datasets.

Model	BioGEN†			LongFact†			FollowRAG - Faithfulness‡				
	FactScore	Respond	Facts	Objects	Concepts	Avg.	NaturalQA	TriviaQA	HotpotQA	WebQSP	Avg.
LLaMA-1											
Vanilla - 100%	38.6	100.0	16.6	84.3	78.2	81.3	37.5	50.5	16.0	47.5	37.9
FLAME-DPO^{fact}	41.2	100.0	14.8	86.7	81.2	84.0	41.5	55.0	21.5	52.5	42.6
SELF-EVAL	41.8	100.0	15.7	87.0	80.8	83.9	42.5	56.5	22.5	53.5	43.8
IFD - 5%	40.2	100.0	20.1	83.2	80.4	81.8	38.0	53.5	18.5	49.0	39.8
CaR - 5%	39.6	100.0	18.2	85.9	80.1	83.0	38.0	53.0	19.0	50.5	40.1
Nuggets - 5%	39.3	100.0	19.4	85.1	77.3	81.2	39.5	54.5	20.0	50.0	41.0
NOVA - 5%	**43.6**	100.0	**21.5**	**88.1**	**82.5**	**85.3**	**44.5**	**58.5**	**24.0**	**55.5**	**45.6**
Δ compared to Vanilla - 100%	+5.0	-	+4.9	+3.8	+4.3	+4.1	+7.0	+8.0	+8.0	+8.0	+7.7
IFD - 10%	40.7	100.0	19.2	85.2	80.3	82.8	40.0	54.5	20.0	51.0	41.4
CaR - 10%	40.3	100.0	**21.1**	83.4	79.2	81.3	41.0	52.0	18.0	49.5	40.1
Nuggets - 10%	41.0	100.0	18.8	84.2	78.6	81.4	39.5	53.0	17.5	51.0	40.3
NOVA - 10%	**43.2**	100.0	20.7	**87.6**	**83.2**	**85.4**	**43.5**	**59.5**	**22.5**	**53.0**	**44.6**
Δ compared to Vanilla - 100%	+4.6	-	+4.1	+3.3	+5.0	+4.2	+6.0	+9.0	+6.5	+5.5	+6.7
IFD - 15%	39.2	100.0	18.7	86.1	81.1	83.6	39.5	52.0	17.5	49.5	39.6
CaR - 15%	40.2	100.0	19.3	84.2	80.4	82.3	38.0	51.5	17.0	48.0	38.6
Nuggets - 15%	40.9	100.0	18.1	83.3	80.0	81.7	40.0	52.5	15.5	50.5	39.6
NOVA - 15%	**44.1**	100.0	**19.4**	**89.6**	**83.7**	**86.7**	**42.5**	**56.5**	**23.5**	**54.5**	**44.3**
Δ compared to Vanilla - 100%	+5.5	-	+2.8	+5.3	+5.5	+5.4	+5.0	+6.0	+7.5	+7.0	+6.4
Qwen-2											
Vanilla - 100%	40.3	100.0	17.3	83.4	80.2	81.8	39.5	57.5	18.5	49.0	41.1
FLAME-DPO^{fact}	47.1	100.0	16.9	87.8	82.7	85.3	44.5	58.0	20.5	53.0	44.0
SELF-EVAL	46.8	100.0	14.2	88.2	81.6	84.9	43.5	59.0	21.0	53.0	44.1
IFD - 5%	44.2	100.0	16.5	85.2	81.2	83.2	42.5	56.5	20.5	53.5	43.3
CaR - 5%	45.7	100.0	**18.6**	84.1	81.5	82.8	44.5	55.5	21.0	52.0	43.3
Nuggets - 5%	46.6	100.0	17.8	84.7	81.0	82.9	43.0	57.5	21.5	52.5	43.6
NOVA - 5%	**49.1**	100.0	18.3	**90.2**	**83.2**	**86.7**	**46.0**	**59.6**	**23.5**	**55.5**	**46.1**
Δ compared to Vanilla - 100%	+8.8	-	+1.0	+6.8	+3.0	+4.9	+6.5	+2.1	+5.0	+6.5	+5.0
IFD - 10%	44.5	100.0	17.8	84.2	80.5	82.4	41.5	59.5	19.5	51.0	42.9
CaR - 10%	45.2	100.0	20.3	84.5	79.8	82.2	42.5	60.0	18.5	53.0	43.5
Nuggets - 10%	46.1	100.0	**23.5**	85.2	79.7	82.5	42.0	60.0	20.0	51.5	43.4
NOVA - 10%	**47.5**	**100.0**	18.6	**89.6**	**83.5**	**86.6**	**45.0**	**62.0**	**21.5**	**53.5**	**45.5**
Δ compared to Vanilla - 100%	+7.2	-	+1.3	+6.2	+3.3	+4.7	+5.5	+4.5	+3.0	+4.5	+4.4
IFD - 15%	43.7	100.0	19.2	82.5	79.5	81.0	42.0	61.5	18.5	52.0	43.5
CaR - 15%	44.8	100.0	20.8	81.2	81.3	81.3	43.0	62.5	19.5	53.0	44.5
Nuggets - 15%	45.7	100.0	**21.7**	80.8	80.1	80.5	40.5	62.5	20.0	52.5	43.9
NOVA - 15%	**47.2**	100.0	19.3	**88.8**	**82.9**	**85.9**	**44.5**	**64.5**	**22.0**	**54.0**	**46.3**
Δ compared to Vanilla - 100%	+6.9	-	+2.0	+5.4	+2.7	+4.0	+5.0	+5.0	+3.5	+5.0	+5.2

Table 7: Results on three hallucination benchmarks. † indicates the factuality hallucination benchmark. ‡ indicates the faithfulness hallucination benchmark. We conduct the experiments based on Alpaca dataset.

D Transferability Study

To verify the transferability of the NOVA method, we conducted experiments on different foundation models using the Alpaca instruction dataset shown in Table 7 and Table 8. We select LLaMA (Touvron et al., 2023) and Qwen-2 (Yang et al., 2024) at the 7B size as the new base models. We aim to gain deeper insights into the applicability of the NOVA method across different models, providing a reference for further research and applications. We find that the NOVA method is also applicable to other models, showing strong transferability and robustness to other models and further research. Compared to other baselines, NOVA significantly reduces hallucinations and keeps a strong ability to follow instructions.

E Design Exploration

The Design of NLI Model We further explore the effects of the NLI model on the final performance of NOVA. We first attempt to analyze the effect of the size of the model on the final results. Specifically, we introduce DeBERTa-base-mnli[3], DeBERTa-xlarge-mnli[4] and DeBERTA-xxlarge-mnli[5]. As shown in Table 9, we can find that increasing the size of the NLI model can provide some improvement in the final result, especially when changing the DeBERTa-base-mnli to DeBERTa-large-mnli. However, continuing to increase the model parameters did not have a significant impact on the final performance. Therefore, in order to balance the performance and the inference time of NLI models, we select the DeBERTa-large-mnli to report the final results in our paper. Meanwhile, we further explore whether we use the advanced LLMs (e.g., GPT-4o and GPT-3.5-Turbo) to directly identify the semantic equivalence and get the correct semantic clusters. Specifically, we use the prompt shown in Figure 6 to test the gener-

[3]https://huggingface.co/microsoft/deberta-base-mnli
[4]https://huggingface.co/microsoft/deberta-xlarge-mnli
[5]https://huggingface.co/microsoft/deberta-v2-xxlarge-mnli

Model	MT-Bench	FollowRAG-Intruction
LLaMA-1		
Vanilla - 100%	47.8	37.7
FLAME-DPO^{fact}	40.6	37.5
SELF-EVAL	42.2	38.1
IFD - 5%	48.3	37.8
CaR - 5%	**50.1**	**38.2**
Nuggets - 5%	48.6	38.0
NOVA - 5%	49.8	38.1
Δ compared to Vanilla - 100%	+2.0	+0.4
IFD - 10%	47.9	38.6
CaR - 10%	**49.5**	38.1
Nuggets - 10%	48.4	38.7
NOVA - 10%	49.3	**39.0**
Δ compared to Vanilla - 100%	+1.5	+1.3
IFD - 15%	48.5	38.2
CaR - 15%	**50.3**	37.6
Nuggets - 15%	49.5	**38.6**
NOVA - 15%	48.3	38.0
Δ compared to Vanilla - 100%	+0.5	+0.4
Qwen-2		
Vanilla - 100%	50.2	38.2
FLAME-DPO^{fact}	47.8	38.7
SELF-EVAL	49.5	37.3
IFD - 5%	59.5	39.2
CaR - 5%	**61.2**	39.5
Nuggets - 5%	60.3	**40.2**
NOVA - 5%	60.8	39.7
Δ compared to Vanilla - 100%	+10.6	+1.5
IFD - 10%	59.8	40.1
CaR - 10%	**60.1**	40.5
Nuggets - 10%	58.8	**41.1**
NOVA - 10%	58.4	40.1
Δ compared to Vanilla - 100%	+8.2	+1.9
IFD - 15%	**59.3**	**40.5**
CaR - 15%	57.5	39.8
Nuggets - 15%	58.5	40.3
NOVA - 15%	59.2	40.0
Δ compared to Vanilla - 100%	+9.0	+1.8

Table 8: Results on two instruction-following benchmarks based on Alpaca dataset.

Model	Size	BioGEN
Alpaca		
DeBERTa-base-mnli	0.1B	49.7
DeBERTa-large-mnli	0.3B	50.3
DeBERTa-xlarge-mnli	0.7B	50.1
DeBERTa-xxlarge-mnli	1.3B	**50.5**
GPT-3.5-Turbo-0125	unknown	49.8
GPT-4o-2024-05-13	unknown	50.2
Alpaca- GPT4		
DeBERTa-base-mnli	0.1B	49.4
DeBERTa-large-mnli	0.3B	50.5
DeBERTa-xlarge-mnli	0.7B	**51.2**
DeBERTa-xxlarge-mnli	1.3B	50.3
GPT-3.5-Turbo-0125	unknown	49.2
GPT-4o-2024-05-13	unknown	50.0

Table 9: FactScore results on BioGEN with different models. We conduct experiments on LLaMA-3-8B and use selected 5% data from different datasets.

Model	Accuracy
Our Used Reward Model	**92.0**
GPT-3.5-Turbo-0125	85.0
GPT-4o-2024-05-13	90.0

Table 10: Accuracy of our used reward model and other advanced LLMs on the constructed test set.

ated responses and the target response by querying the advanced LLMs to identify semantic equivalence. We use the same method as SEI, utilizing the outputs of advanced LLMs to derive semantic clusters and calculate the score of $F_{res}(r)$. As shown in Table 9, the direct application of results from advanced LLMs proves effective in identifying semantic equivalence. Nevertheless, using NLI models delivers competitive or superior final performance while avoiding API-related costs. Consequently, employing NLI models to identify semantic equivalence is both efficient and effective, substantiating the efficacy of our designed SEI approach.

The Design of Quality Reward Model We also explore the effectiveness of the quality reward model. We introduce UltraFeedback (Cui et al., 2024) and sample 100 instructions and their corresponding responses as the test set (we find that most of the selected data are in English, but some of the selected instruction types are translation tasks, so a few data contain Chinese responses). Specifically, for each instruction, we randomly select 2 responses and determine the ranking between the responses based on their labeled scores of instruction-following, honesty, truthfulness, and helpfulness. Only if all four scores are higher will the response be considered a high-quality response. Meanwhile, we involve two Ph.D. students to conduct the human evaluation to ensure the correctness of the response ranking of each sample. Afterwards, we take the instructions and the responses as inputs to each model, and let the model determine the ranking between the responses and calculate the accuracy of the model's prediction of the ranking. We compare our used Quality Reward Model with GPT-3.5-Turbo-0125 and GPT-4o-2024-05-13. We use the same prompt for each model as Ge et al. (2024). As shown in Table 10, our reward model achieves better performance, showing the effectiveness of our method. Despite GPT-4o's strong alignment with human preferences in most general tasks, our reward model trained on the expert-revised preference dataset can perform better, highlighting the subtle gap between expert preferences and advanced GPT-4o preferences.

The Design of Obtaining Sentence Embedding. Alpaca-GPT4 For K generated responses, we use the internal states of the last token of each response in the last layer as the final sentence embeddings $E = [e_1, e_2, ..., e_K]$, as it effectively captures the

The Prompt for Identifying the Semantic Equivalence

Please compare the following two sentences and determine whether they are semantically the same. If they are semantically identical, respond with "Identical"; if not, respond with "Different." Consider the meaning, context, and any implicit nuances of the sentences.

Sentence 1: {Sentence 1}
Sentence 2: {Sentence 1}

Provide your judgment below:

Figure 6: The prompt for identifying the semantic equivalence.

Model	BioGEN	MT-Bench
NOVA - 5%	**50.5**	**64.6**
-w. Average Pooling	49.5	64.2
-w. The First Layer	48.9	63.7
-w. The Middle Layer	49.8	64.4
-w. The Last Layer (Ours)	50.5	64.6

Table 11: Evaluation results of NOVA that employ various methods for obtaining sentence embedding. We conduct the experiments based on LLaMA-3-8B and the Alpaca-GPT4 dataset. We report the FactScore results on BioGEN.

Model	BioGEN	MT-Bench
NOVA - LLaMA-3-8B - 5%	**50.3**	**60.5**
-w/o. Few-shot Demonstrations	50.1	59.8
NOVA - LLaMA-1-7B - 5%	**43.6**	**49.8**
-w/o. Few-shot Demonstrations	41.9	49.2

Table 12: The effects of used few-shot demonstrations. We conduct the experiments based on two base models and the Alpaca dataset. We report the FactScore results on BioGEN.

sentence semantics (Azaria and Mitchell, 2023). We further explore the different ways to obtain sentence embedding. Specifically, we first average all the internal states of tokens in the sentence to obtain the sentence embedding (named Average Pooling), which is an intuitive method to get the sentence embedding for decoder-only models. As shown in Table 11, we can find the design of NOVA achieves better performance in both reducing hallucinations and following instructions, showing the effectiveness of our designed SEI. We further explore the internal states from which layer in the LLMs can be used to effectively measure the consistency. Except for the internal states from the last layer, we select both internal states from the first layer and internal states from the middle layer (layer 16 for LLaMA-3-8B), and use the internal states of the last token to represent the sentence embeddings. We can find that using sentence embedding in the shallow layer yields inferior performance compared to using sentence embedding in the deep layers, as the shallow layer may not effectively model the rich semantic information. Overall, extensive experiments show that our design of NOVA is sound and effective.

The Design of Using Few-shot Demonstration. As detailed in Sec. 3.1, we sample K responses $[r'_1, ..., r'_K]$ from a base LLM with few-shot demonstrations (Lin et al., 2024a) to ensure the coherence of generated responses. We use the same demonstrations as Lin et al. (2024a). We further conduct experiments to explore the effects of these used demonstrations. We find that using few-shot demonstrations in the process of generating responses for a given instruction allows the base LLMs to better express what they have learned in the pre-training stage. In turn, this will enable ICP and SEI to better estimate the knowledge contained in the instruction data and thus better identify the high-quality instruction data that aligns well with the LLM's learned knowledge to reduce hallucination and improve instruction-following ability. At the same time, we find that this strategy improves more for base models with poor capabilities (e.g., LLaMA-1-7B), which is due to the fact that a poor base LLM may hold relevant knowledge in response to a query, yet occasionally falters in conveying accurate information (Zhang et al., 2024b).

F Human Evaluation

During the human evaluation, the participants follow the principles in Figure 7 to make the decision. For each comparison, three options are given (Ours Wins, Tie, and Vanilla Fine-tuning Wins) and the majority voting determines the final result. We invite three Ph.D. students to compare the responses

The Principles of Human Evaluation

You are asked to evaluate the biographies generated by different models. You should choose the preferred biography according to the following perspectives independently:

1. **Factuality**: Whether the biography provides relatively more factual statements over the non-factual statements?
2. **Helpfulness**: Whether the biography provides useful information?
3. **Relevance**: Whether the statements contained in the biography relevant to the provided people entity?
4. **Naturalness**: Whether the biography sound natural and fluent?

Finally, please make a decision among 3 opinions, including Win, Tie, and Loss.

Figure 7: The principles of human evaluation.

generated by the models. Before participants begin to make judgments, we describe the principles of our design in detail and ensure that each participant correctly understands the principles. If the final result can not be determined by the majority voting, we will make the discussion among the participants and vote on the result again.

G Case Study for Selected Samples

To evaluate our proposed NOVA qualitatively, we also select some instruction samples from the Alpaca dataset for case studies as shown in Figure 8. Firstly, we can find that simply using $R_{familiarity}$ in Eq. (10) can effectively identify the simple and straightforward instruction samples that align well with LLM's knowledge. On the contrary, the sample ranked last according to $R_{familiarity}$ contains the open-ended instruction and the very subjective target response. Meanwhile, further using R_{final} in Eq. (10) that considers characteristics beyond just familiarity $R_{familiarity}$ by introducing the quality reward model further enhances data quality, e.g., the complexity of instructions. We can also observe that the sample ranked last according to R_{final} also contains the open-ended instruction and the detailed target response, which may introduce familiar knowledge to LLMs and further promote hallucinations.

Case Study for Selected Samples

😁 **Sample ranked first according to $R_{familiarity}$ in Eq. (10):**
Instruction: Find a word that is opposite in meaning to the given word: assured.
Target Response: Doubtful.

☹ **Sample ranked last according to $R_{familiarity}$ in Eq. (10):**
Instruction: Could you pick a set of five songs which will fit our theme: a summer night's party?
Target Response:
1. Billie Eilish – "Bad Guy"
2. Ed Sheeran – "Shape of You"
3. Maroon 5 – "Summer Nights"
4. DNCE – "Cake by the Ocean"
5. Drake – "In My Feelings".

😁 **Sample ranked first according to R_{final} in Eq. (10):**
Instruction: Given a dictionary, create a JSON representation. dict = {"name": "John", "age": 18, "is_adult": True}
Target Response:
{
"name": "John",
"age": 18,
"is_adult": true
}

☹ **Sample ranked last according to R_{final} in Eq. (10):**
Instruction: Describe how to impress a customer.
Target Response: Listen to the customer carefully, acknowledge their needs and desires, make sure to follow up on commitments, and strive to exceed expectations.

Figure 8: Case study for selected samples.