

# NILE 🏰: Internal Consistency Alignment in Large Language Models

Minda Hu<sup>1</sup>, Qiyuan Zhang<sup>2</sup>, Yufei Wang<sup>3</sup>, Bowei He<sup>2</sup>, Hongru Wang<sup>4</sup>  
Jingyan Zhou<sup>1</sup>, Liangyou Li<sup>3</sup>, Yasheng Wang<sup>3</sup>, Chen Ma<sup>2</sup>, Irwin King<sup>1</sup>

<sup>1</sup>The Chinese University of Hong Kong <sup>2</sup>City University of Hong Kong

<sup>3</sup>Huawei Noah’s Ark Lab <sup>4</sup>University of Edinburgh  
{mindahu21, king}@cse.cuhk.edu.hk

## Abstract

Recent advances show that the world knowledge in the Instruction Fine-Tuning (IFT) dataset, which is incompatible with LLMs’ internal knowledge, can greatly hurt the IFT performance. However, the effective integration and balancing of the internal knowledge of LLMs, acquired during pre-training, with existing IFT datasets remains a largely underexplored area of research. To address this gap, this work introduces NILE, a novel framework to optimize the effectiveness of IFT by adjusting IFT datasets through carefully aligning the world and internal knowledge. NILE employs a three-stage pipeline to effectively quantify and adjust consistency with the internal knowledge of target LLMs. Our analysis provides compelling evidence that balancing such consistency with pre-trained internal knowledge is pivotal for unleashing LLM potential, and confirms that NILE can systematically contribute to these substantial performance improvements. Experimental results demonstrate that NILE-aligned IFT datasets sharply boost LLM performance across multiple LLM ability evaluation datasets, achieving up to 66.6% gain on Arena-Hard and 68.5% on Alpaca-Eval V2.

## 1 Introduction

Instruction Fine-Tuning (IFT), which fine-tunes Large Language Models (LLMs) on instruction-response pairs, has been proven to be an effective and crucial method to enhance the capabilities and controllability of LLMs (Touvron et al., 2023; Dubey et al., 2024; Zhang et al., 2023; Chen et al., 2024; Wang et al., 2025). Most IFT approaches predominantly focus on the quantity and diversity of datasets, based on the assumption that a greater size of instruction-response pairs would lead to better performance (Honovich et al., 2023; Wang et al., 2023b; Taori et al.; Chiang et al., 2023; Sun et al., 2024). These approaches narrowly emphasize data quantity while overlooking IFT’s core purpose: un-

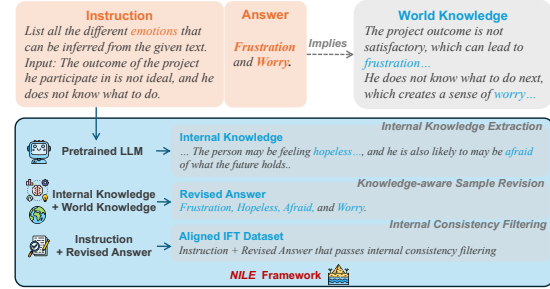


Figure 1: Demonstration of LLM internal knowledge and world knowledge from IFT datasets.

locking the latent capabilities of pre-trained LLMs. They do not adequately consider underlying correlations between IFT datasets and LLMs, which is crucial to the efficacy of IFT (Ren et al., 2024).

A key factor influencing IFT performance is the level of *internal consistency*, i.e., the consistency between the world knowledge in IFT datasets and the internal knowledge embedded within LLM parameters (Ren et al., 2024). When trained on totally unfamiliar data, i.e., data with low internal consistency, LLMs may only capture superficial correlations in instruction-response pairs, such as text styles, and tend to make “blind guesses” when faced with new queries (Kang et al., 2024). Nonetheless, Ren et al. (2024) shows that merely maximizing internal consistency does not necessarily lead to optimal IFT performance. These works suggest that examining and curating the internal consistency of IFT datasets for the target pre-trained LLM is a promising direction for effective training. However, how to revise and balance the internal consistency level remains under-explored.

In this work, we propose a novel framework **NILE** (INTERNAL CONSISTENCY ALIGNMENT). NILE bridges the aforementioned research gap by flexibly improving existing IFT datasets in terms of internal consistency for the target pretrained LLM. Specifically, NILE addresses the problem through the following three steps: 1) *Internal Knowledge Extraction*.

As a prerequisite, accurately extracting internal knowledge is crucial. We adopt in-context learning techniques with high-quality customized examples. 2) *Knowledge-aware Sample Revision*. To fully utilize existing data, we designed a revision step to improve the existing data with LLMs’ internal knowledge, resulting in a data sample with higher consistency. 3) *Internal Consistency Filtering*. Lastly, we developed a novel metric to measure the consistency level between the data sample and the LLM. By doing so, we can flexibly adjust the level of internal consistency of existing IFT data with any target pre-trained LLM to achieve optimized IFT performance.

It is important to highlight that our method does not rely on any additional forms of supervision (i.e., human experts). To conclude, our contributions can be summarized as follows:

- We propose NILE, a novel framework to adjust and select better IFT datasets considering the consistency between internal parameter knowledge in LLMs and world knowledge in IFT datasets, as shown in Figure 1. To the best of our knowledge, we are among the first to leverage the concept of internal consistency for IFT data selection and generation.<sup>1</sup>
- Through comprehensive ablation studies and empirical analysis, we demonstrate that balancing consistency between IFT datasets and LLMs’ internal knowledge is crucial for unlocking model capabilities. Our results provide strong evidence that each component of NILE contributes to performance gains.
- Our extensive experiments across multiple benchmarks show that NILE-optimized datasets enable substantial improvements in LLM performance, achieving up to 66.6% gains on Arena-Hard and 68.5% on Alpaca-Eval V2. These results demonstrate that NILE’s balanced integration of world and internal knowledge enhances LLMs’ ability to generalize to novel tasks and domains.

## 2 Related Works

### 2.1 Data Synthesis in Instruction Tuning

Earlier research on instruction tuning has primarily focused on developing large, high-quality datasets

<sup>1</sup>Corresponding NILE-revised IFT datasets can be found in <https://huggingface.co/datasets/mindahu/NILE-IFT-Dataset>.

curated by human experts (Wei et al., 2022; Wang et al., 2022). However, this process is often time-consuming and labor-intensive. Thus, several studies have explored the use of more advanced models or self-critique prompting methods (Wang et al., 2023a, 2024; Zhang et al., 2024; Pi et al., 2024b,a) to generate instruction-tuning datasets automatically. For example, Self-Instruct (Wang et al., 2023b) leverages GPT-3 to expand asks to many diverse domains in an in-context learning manner while several recent studies directly use the latest SOTA model to generate the response or reflect on current samples (Mukherjee et al., 2023), such as WizardLM (Xu et al., 2023) and Reflection-tuning (Li et al., 2024a). In addition to focusing on the quality side, another area of work aims to create more diverse and larger instruction-tuning datasets. For example, UltraChat (Ding et al., 2023) defines specific scopes and systematically generates a wide range of instructions within each area. In contrast, Magpie (Xu et al., 2024) only feeds the left-side templates up to the position reserved for user messages as input to generate more diverse user queries.

For complex reasoning tasks such as coding and mathematics, many efforts have been made to integrate human priors into data synthesis (Zhou et al., 2025). KPDDS (Huang et al., 2025) leverages key points and exemplar practices to synthesize mathematical reasoning-focused IFT datasets. Additionally, Case2Code (Shao et al., 2024) introduces observations of input-output examples and program behaviors to infer underlying code implementations.

### 2.2 Data Selection in Instruction Tuning

Data selection (or revision) has been widely studied in large language model instruction tuning, considering the importance of data quality in model training (Li et al., 2024b; Cao et al., 2023; Li et al., 2024d; Zhou et al., 2024; Liu et al., 2024; Li et al., 2024a). Most previous studies fall into two categories: 1) relying on more powerful models or human experts to select better data (Zhou et al., 2024; Liu et al., 2024); 2) calculating the perplexity gains considering generated samples and original samples (Li et al., 2024a). While both methods improve downstream performance, they face significant limitations, such as the high cost of human labeling. More importantly, such studies (Chen et al.; Li et al.; Sun et al., 2024) can not provide fundamental explanations regarding the key factors

that define better instruction-tuning datasets. In contrast to these approaches, our work aligns the internal knowledge of LLMs with external world knowledge derived from IFT datasets, resulting in improved datasets that offer better explainability and transparency.

### 3 Method

Figure 2 demonstrates our framework NILE for increasing knowledge affinity between LLMs’ internal knowledge and instruction-tuning datasets. It can be divided into three parts: (1) *Internal Knowledge Extraction (IKE)*, (2) *Knowledge-aware Sample Revision (KSR)*, and (3) *Internal Consistency Filtering (ICF)*. The core contribution of our framework lies in our deliberate focus on internal consistency, which enables the process to function effectively. IKE accesses the memory of pretrained LLMs to sample their internal knowledge. KSR revises existing dataset samples by automatically infusing the sampled internal knowledge. ICF introduces a novel internal consistency measurement to filter out low-quality revisions from the second phase. In the following subsections, we introduce the above three components in detail. Implementation details of IKE, KSR, and ICF are listed in Appendix A.1.

#### 3.1 Internal Knowledge Extraction

This stage aims to effectively sample the internal knowledge from the target pre-trained LLM  $\mathcal{M}$  for instructions in the original IFT dataset  $\mathcal{D}^o = \{(q_i^o, a_i^o)\}_{i=1}^n$ , where  $q_i^o$  is the concatenated query sequence of **instruction** <sub>$i$</sub>  <sup>$o$</sup>  and **input** <sub>$i$</sub>  <sup>$o$</sup> , and  $a_i^o$  is the answer. Formally, we aim to sample the internal knowledge  $ik_i$  corresponding to  $q_i^o$  from  $\mathcal{M}$  through in-context learning. Instead of using a fixed set of examples, we use following three-step strategy to provide the most relevant examples to better exert the internal knowledge from  $\mathcal{M}$ .

1. **Demonstration set construction:** we first randomly sample a subset of queries  $\{q_j^d\}_{j=1}^m$  from an IFT dataset. Then, as shown in Table 1, a strong LLM (GPT-4 utilized in the experiments) is prompted to generate the corresponding knowledge snippet  $ik_j^d$  for each  $q_j^d$ , resulting in a demonstration database index  $\mathcal{F}^{demo} = \{(q_j^d, ik_j^d)\}_{j=1}^m$ . Details are provided in Appendix A.1.3.
2. **Example selection:** for each query  $q_i^o \in \mathcal{D}^o$ ,

we select  $k$  few-shot examples  $f^{\mathcal{R}}(q_i^o) = \{(q_{i_t}^d, ik_{i_t}^d)\}_{t=1}^k$  from  $\mathcal{F}^{demo}$ , where  $(i_t)_{t=1}^k$  denotes the indices of top- $k$  example pairs ranked by the query semantic similarity between  $\{q_j^d\}_{j=1}^m$  and  $q_i^o$  from retriever  $\mathcal{R}$ .  $\mathcal{R}$  is implemented by information retrieval algorithms such as BM25.

3. **Internal knowledge generation:** we formulate the prompt shown in Table 2 to  $\mathcal{M}$  with few-shot examples  $f^{\mathcal{R}}(q_i^o)$  and the original instruction  $q_i^o$ . By this means, it can effectively exert internal knowledge  $ik_i$  from the target LLM  $\mathcal{M}$ .

Generate a list of related knowledge about the following Instruction and input up to 500 words. Do not directly output the answer, but focus on the related knowledge required for answering the Input.

Instruction: "{instruction <sub>$j$</sub>  <sup>$d$</sup> }"  
Input: "{input <sub>$j$</sub>  <sup>$d$</sup> }"

Table 1: Prompt for demonstration set construction.

$\{(q_{i_t}^d, ik_{i_t}^d)\}_{t=1}^k$

Instruction:  
{instruction <sub>$i$</sub>  <sup>$o$</sup> , input <sub>$i$</sub>  <sup>$o$</sup> }

Related Knowledge:

Table 2: Prompt for knowledge extraction. Sample few-shot demonstration prompt is listed in A.1.3.

By following this approach, we can effectively extract the internal knowledge of unaligned LLMs relevant to the original instructions, leveraging the power of few-shot demonstration learning.

#### 3.2 Knowledge-aware Sample Revision

After obtaining a relatively accurate sampling  $ik_i$  of the target LLM’s internal knowledge (analyzed in Section 4.8), for each original instruction  $q_i^o$ , we design a prompt for the revisor LLM agent  $\mathcal{A}_r$  to infuse  $ik_i$  into the current instruction and get the revised answer  $a_i^{ik}$ . The prompt for KSR is displayed in Table 3.

This step aims to enhance affinity between the target model  $\mathcal{M}$ ’s internal knowledge  $ik_i$  and the original answer  $a_i^o$  from  $\mathcal{D}^o$  with world knowledge, resulting an improved answer  $a_i^{ik}$ .

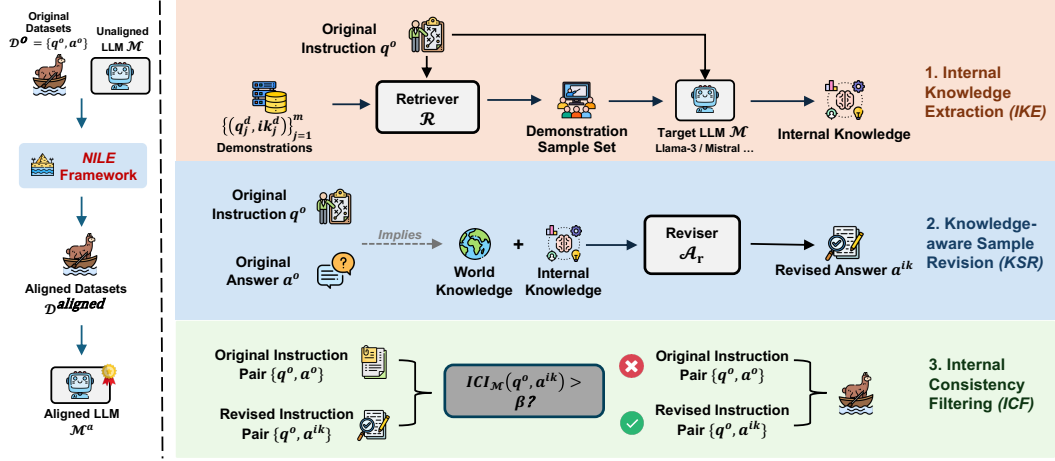


Figure 2: Overview of our NILE framework. NILE consists of three main steps: *Internal Knowledge Extraction (IKE)*, *Knowledge-aware Sample Revision (KSR)*, and *Internal Consistency Filtering (ICF)*.

Provide a better response based on " $\{a_i^o\}$ " to comply with given instruction, input, and related knowledge.

Instruction:  $\{\text{instruction}_i^o\}$   
Input:  $\{\text{input}_i^o\}$   
Related Knowledge:  $\{\text{ik}_i\}$

Please directly output the improved response.

Table 3: Prompt for Knowledge-aware Sample Revision.

### 3.3 Internal Consistency Filtering

In this stage, we evaluate the effectiveness of KSR by comparing the quality of the revised answer  $a_i^{ik}$  with the original answer  $a_i^o$ . Drawing inspiration from IFD and PMI (Li et al., 2023), we introduce a novel metric called **INTERNAL CONSISTENCY INDEX (ICI)** to quantify how well one answer promotes knowledge associations in the pretrained LLM  $\mathcal{M}$ .

During the instruction alignment process, the loss of a sample pair  $(q, a)$  is computed using the sequence probability of  $a$  conditioned on  $q$ :

$$P_{\mathcal{M}}(a | q) = \frac{1}{N} \sum_{i=1}^N \log P_{\mathcal{M}}(w_i^a | q, w_1^a, w_2^a, \dots, w_{i-1}^a), \quad (1)$$

where  $w_i$  is the tokens in  $a$  and  $N$  is the sequence length of  $a$ . This probability measures the familiarity of  $\mathcal{M}$  with answer  $a$  given the context  $q$ . It can also reflect the strength of the encoded association between  $a$  and  $q$  in the LLM’s representations, which is empirically supported by Kang et al.. Building upon this idea, we formulate ICI as

follows:

$$\text{ICI}_{\mathcal{M}}(q, a^{ik}) = \frac{P_{\mathcal{M}}(a^{ik} | q, ik)}{P_{\mathcal{M}}(a^{ik} | q)}, \quad (2)$$

where  $P_{\mathcal{M}}(a^{ik} | q)$  measures the associations between revised responses  $a^{ik}$  and instructions  $q$  alone, while  $P_{\mathcal{M}}(a^{ik} | q, ik)$  captures the overall association strength between  $a^{ik}$  and the combination of  $q$  and its corresponding extracted internal knowledge  $ik$ . To isolate the influence of  $ik$  on the revised answer  $a^{ik}$ , we minimize the influence of  $q$  in the ICI formulation by dividing  $P_{\mathcal{M}}(a^{ik} | q, ik)$  with  $P_{\mathcal{M}}(a^{ik} | q)$ .

For samples with higher ICI, the model more effectively integrates and leverages the explicitly provided internal knowledge when generating the revised answer, suggesting a stronger alignment between the revised answer and the model’s internal knowledge. Conversely, for samples with lower ICI, providing internal knowledge may not benefit or could even hinder the generation of the revised answer, indicating that the revised answer does not have a strong association with what the model has learned internally, as suggested by Ren et al. (2024). Therefore, we employ a filtering mechanism ICF to filter out these redundant low ICI samples to an aligned dataset  $\mathcal{D}^{\text{aligned}}$  for fine-tuning an aligned LLM  $\mathcal{M}^a$  from  $\mathcal{M}$ . To control dataset size in the experiment and ensure stable improvement, we revert to the original samples  $(q, a^o)$  when the ICI values of  $(q, a^{ik})$  are lower than the threshold  $\beta$ :

$$\mathcal{D}^{\text{aligned}} = \{q_{1..n}^o, a_{1..n}^{\text{aligned}}\},$$

$$\text{where } a_i^{\text{aligned}} = \begin{cases} a_i^{ik}, & \text{if } \text{ICI}_{\mathcal{M}}(q_i^o, a_i^{ik}) > \beta \\ a_i^o, & \text{otherwise} \end{cases} \quad (3)$$



Here we use  $\beta$  to control the degree of internal consistency in ICF.

## 4 Experiments

For the main experiment, we use open source models like MISTRAL-7B-V0.3 (Jiang et al., 2023) and META-LLAMA-3.1-8B (Dubey et al., 2024) on two public datasets Alpaca (Taori et al.) and OpenOrca (Mukherjee et al., 2023) to examine NILE framework’s robustness extensively. In addition, we conduct an ablation study to evaluate the efficacy of our design choices in the pipeline. More experiment details, ablation study, inference overhead, and case studies can be found in A.1.

### 4.1 IFT Datasets

**Alpaca** The Alpaca dataset contains 52,000 instruction-following data generated using the techniques in the Self-Instruct (Wang et al., 2023c). It starts with a limited (e.g., 175 in our study) seed set of manually written tasks that are used to guide the overall generation. Then language models are utilized and prompted to augment these instructions and create corresponding instruction-answer instances. In our experiments, we use all the samples in a newer version of Alpaca<sup>2</sup> dataset, which includes instruction-following instances generated using GPT-4 (Peng et al., 2023).

**Orca** OpenOrca is a large-scale dataset built upon the Flan 2022 Collection (Mukherjee et al., 2023; Longpre et al., 2023). In the Orca dataset, query-response pairs are augmented with detailed responses from GPT-4 that explain the reasoning process of the teacher as it generates the response. In contrast with vanilla instruction tuning methods like Alpaca providing little opportunity for mimicking the “thought” process, this dataset provides additional signals for learning to elicit such explanations. For experiments, we use the officially released dataset<sup>3</sup>, and randomly select 50,000 sample pairs from a pool of 1 million samples.

### 4.2 Evaluation

We briefly introduce evaluation methods used in our experiments as follows.

<sup>2</sup><https://huggingface.co/datasets/vicgalle/alpaca-gpt4>

<sup>3</sup><https://huggingface.co/datasets/Open-Orca/1million-gpt-4>

**Arena-Hard (A.-H.)** Arena-Hard-Auto<sup>4</sup> is a popular open-ended evaluation tool for instruction-tuned LLMs (Li et al., 2024c). It contains 500 challenging user queries. GPT-4-Turbo is prompted as a judge to compare the models’ responses against a baseline model. Notably, Arena-Hard keeps a high correlation and separability to Chatbot Arena (Chiang et al., 2024).

**Alpaca-Eval V2 (A.-E. V2)** Alpaca-Eval V2<sup>5</sup> is an automatic evaluation system for instruction-following language models (Dubois et al., 2024). It builds upon the original AlpacaEval system, which benchmarked against OpenAI’s Davinci-003. AlpacaEval V2 instead uses GPT-4-Turbo, signaling the new state-of-the-art model since the original system’s creation.

A key innovation in Alpaca-Eval V2 is the introduction of **Length-Controlled Win Rates (LCWR)**. It increases the correlation with ChatBot Arena to 0.98, significantly decreasing length gameability in comparison with the original **Win Rate (WR)**. In presenting experimental results, we display reports both metrics in the format: **LCWR / WR**. This provides a more comprehensive picture of model performance, with **LCWR** serving as the primary metric while still allowing comparison to the original **WR** scores.

**MTBench (MTB.)** MT-Bench comprises 80 multi-turn questions spanning eight distinct knowledge domains. The models are required to respond to an initial question and subsequently provide a second response to a follow-up question. GPT-4 assesses each model’s responses on a scale from 1 to 10, and the overall score is determined by the mean over the two turns across all questions. We evaluate using the Fastchat implementation<sup>6</sup>.

**BBH** Big Bench Hard<sup>7</sup> (BBH) is a suite of 23 challenging BIG-Bench tasks (Suzgun et al., 2023; Srivastava et al., 2022). These tasks are chosen because prior language models showed performance below the average human-raters. Since many tasks in BBH require multi-step reasoning, CoT prompting is added to better depict the LLMs’ capacities on these complex tasks that are challenging even

<sup>4</sup><https://github.com/lmarena/arena-hard-auto>

<sup>5</sup>[https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval)

<sup>6</sup>[https://github.com/lm-sys/FastChat/blob/main/fastchat/llm\\_judge](https://github.com/lm-sys/FastChat/blob/main/fastchat/llm_judge)

<sup>7</sup>[https://github.com/EleutherAI/lm-evaluation-harness/tree/main/lm\\_eval/tasks/bbh](https://github.com/EleutherAI/lm-evaluation-harness/tree/main/lm_eval/tasks/bbh)

for humans.

### 4.3 Implementation details

For our experiments, we fine-tune the pretrained but unaligned models, MISTRAL-7B-v0.3 and META-LLAMA-3.1-8B. For selecting retriever  $\mathcal{R}$  in IKE, we find that BM25 is more effective than a strong neural retriever such as contriver (Lei et al., 2023) in retrieving higher-quality demonstrations, which is evaluated and validated in Appendix A.1.2. To maintain a better state of internal consistency, we set  $\beta$  in Eq. 3 to the 1st percentile of the ICI distribution for Alpaca and to the 2nd percentile for OpenOrca to rule out a small amount of low ICI samples. Based on our manual random screening of 100 sample points respectively in Alpaca and OpenOrca datasets, we found the selected values in ICF to be a reasonable balance - lower thresholds retain too many misaligned knowledge samples that could directly impair performance, while higher thresholds discard too many consistent samples.

### 4.4 Baselines

**Vanilla** Vanilla setting refers to using the original, unmodified IFT datasets for fine-tuning LLMs such as MISTRAL and LLAMA-3. This serves as a baseline to compare the effectiveness of dataset revision techniques.

**SR** Sample Revision (SR) marks the baseline for revising the instruction-answer pairs without leveraging any internal knowledge from the target LLM  $\mathcal{M}$ . This lets SR solely infuse knowledge from the revisor agent  $\mathcal{A}_r$  into IFT datasets. Details of SR can be found in A.1.5.

**NILE** NILE represents our complete proposed method. In the experiments, Alpaca and Orca datasets undergo a step-by-step revision process through the pipeline of IKE, KSR, and ICF introduced in Section 3.

To maintain consistency and a fair comparison with the Vanilla setting, the implementation of NILE and SR baseline rewrites only the responses  $a^o$ , leaving the rest of the dataset unchanged.

### 4.5 Results on Orca Dataset

Table 4 shows the performance of our NILE framework and all baselines on model MISTRAL-7B-v0.3 and META-LLAMA-3.1-8B in OpenOrca dataset. As we can see, Orca dataset brings unbalanced improvements on different LLMs, with

LLAMA-3 having less improvements on **Arena-Hard** and **Alpaca-Eval V2 LCWR** and more on **MTBench** and **BBH** than MISTRAL, which reflects different underlying characteristics and potentially distinct internal knowledge in these two models.

Compared with ORCA VANILLA, ORCA + NILE brings substantial improvements on all benchmarks in both LLMs. It increases **Arena-Hard** score by 1.4 points (26.4% relative improvement) in MISTRAL and 2.4 points (66.6%) in LLAMA-3. NILE also significantly enhances **Alpaca-Eval V2 LCWR** from 12.73 to 21.63 in MISTRAL and 10.84 to 13.70 in LLAMA-3, achieving 68.5% and 26.4% relative improvements respectively.

In addition, it is noteworthy that **NILE** also brings considerable boosts on **BBH** benchmark by 4.64 in MISTRAL and by 1.05 in LLAMA-3. BBH tasks mainly focus on tasks requiring complex reasoning and expert knowledge, and performance lift of ORCA + NILE compared to ORCA VANILLA indicates the fact that alignment dataset revised by NILE encroaches fewer LLMs’ innate capability of multi-step complex reasoning since instructions in OpenOrca dataset itself is barely involved with multi-step complex reasoning, and yet ORCA + NILE helps unleashing the reasoning ability of the LLMs, as shown in the result of **BBH**. The universal improvements in these four well-tested benchmarks provide strong support for NILE’s effectiveness in improving LLMs’ general capacity.

Compared to ORCA + NILE, ORCA + SR infuses only the internal knowledge of the GPT-4 revisor model without utilizing extracted knowledge from MISTRAL and LLAMA-3 or the ICF phase. The experiment involving ORCA + SR is designed to investigate the contribution that introducing LLMs’ own internal knowledge makes in the NILE framework. ORCA + NILE largely surpasses ORCA + SR by 3.4 and 5.0 points on **Alpaca-Eval V2 LCWR** and **BBH** in MISTRAL model, 1.3 and 1.8 points on **Alpaca-Eval V2 LCWR** and **Arena-Hard** in LLAMA-3. This indicates that internal knowledge extracted from LLMs is crucial for bringing more performance uplift in LLM’s general capability.

### 4.6 Results on Alpaca

Compared with Orca dataset, LLMs finetuned with Alpaca dataset are generally weaker than ones with Orca, which highlights the sheer quality differences between the two datasets. Despite these differences, ALPACA + NILE still brings significant improve-

Method	Arena-Hard $\uparrow$	Alpaca-Eval V2 $\uparrow$	MTBench $\uparrow$	BBH $\uparrow$
MISTRAL-7B-V0.3				
ALPACA VANILLA	3.00	<u>11.73 / 7.39</u>	<u>6.37</u>	34.46
ALPACA + SR	<u>4.20</u>	<u>11.50 / 6.52</u>	6.28	<u>38.40</u>
ALPACA + NILE	<b>6.20</b>	<b>15.39 / 9.70</b>	<b>6.56</b>	<b>38.52</b>
ORCA VANILLA	5.30	12.84 / 9.54	5.34	<u>46.37</u>
ORCA + SR	<u>5.70</u>	<u>18.19 / 15.24</u>	<u>6.13</u>	46.01
ORCA + NILE	<b>6.70</b>	<b>21.63 / 17.25</b>	<b>6.73</b>	<b>51.01</b>
META-LLAMA-3.1-8B				
ALPACA VANILLA	2.10	7.58 / 5.53	6.31	58.64
ALPACA + SR	<u>3.30</u>	<u>9.08 / 6.84</u>	<u>6.39</u>	<u>59.91</u>
ALPACA + NILE	<b>4.80</b>	<b>10.69 / 10.43</b>	<b>6.90</b>	<b>61.40</b>
ORCA VANILLA	3.60	10.84 / 7.52	7.01	63.02
ORCA + SR	<u>4.20</u>	<u>12.36 / 10.46</u>	<u>7.18</u>	<u>63.77</u>
ORCA + NILE	<b>6.00</b>	<b>13.70 / 12.11</b>	<b>7.48</b>	<b>64.05</b>

Table 4: Main experiment results on Alpaca and OpenOrca datasets. The highest values are **bolded**, and the second highest is underlined.

ments over ALPACA VANILLA in all metrics, coming close to or even surpassing ORCA VANILLA in most of the benchmarks except **BBH**. It achieves a performance uplift by 3.7 and 4.1 points on **Alpaca-Eval V2 LCWR** and **BBH** in MISTRAL. Moreover, ALPACA + NILE raises **Alpaca-Eval V2 LCWR** and **Arena-Hard** by 3.1 and 2.7 in LLAMA-3.

Measured against ALPACA + SR, ALPACA + NILE still maintains major advantages. It enhances **Arena-Hard** and **Alpaca-Eval V2** by 2.0 and 3.9 in MISTRAL model, 1.5 and 1.6 in LLAMA-3. These results further illustrate the necessity of extracting internal knowledge in NILE.

#### 4.7 Experiment Results on More LLMs

We conduct additional experiments of NILE on pretrained models of varying sizes, such as META-LLAMA-3.2-3B<sup>8</sup>, as well as on different model families, including QWEN2.5-7B<sup>9</sup> and QWEN2.5-14B<sup>10</sup>, using the **MTBench** and **Alpaca-Eval V2** benchmarks. The consistent and significant relative improvements (up to 85.7% on **Alpaca-Eval V2 LCWR**) observed in Table 5 demonstrate that NILE consistently delivers meaningful improvements across diverse LLM configurations.

#### 4.8 Ablation Study

##### Effects of Different Internal Knowledge Sources

We closely examine the effect of introducing LLMs’ internal knowledge into NILE by switching the original internal knowledge source from MISTRAL

Method	MTBench $\uparrow$	Alpaca-Eval V2 $\uparrow$
META-LLAMA-3.2-3B		
ALPACA VANILLA	5.52	6.17 / 3.54
ALPACA + SR	<u>5.65</u>	<u>6.18 / 4.43</u>
ALPACA + NILE	<b>5.94</b>	<b>6.61 / 5.10</b>
ORCA VANILLA	2.20	5.61 / 4.41
ORCA + SR	<u>3.06</u>	<u>6.18 / 5.51</u>
ORCA + NILE	<b>4.63</b>	<b>10.77 / 8.46</b>
QWEN2.5-7B		
ALPACA VANILLA	<u>7.13</u>	13.84 / 7.83
ALPACA + SR	6.78	<u>15.40 / 8.90</u>
ALPACA + NILE	<b>8.13</b>	<b>17.42 / 12.24</b>
ORCA VANILLA	6.60	<u>19.19 / 13.42</u>
ORCA + SR	<u>7.05</u>	<u>18.45 / 14.34</u>
ORCA + NILE	<b>7.31</b>	<b>20.55 / 16.58</b>
QWEN2.5-14B		
ALPACA VANILLA	7.33	15.37 / 8.16
ALPACA + SR	<u>7.73</u>	<u>21.56 / 12.59</u>
ALPACA + NILE	<b>8.06</b>	<b>28.55 / 17.45</b>
ORCA VANILLA	7.68	19.99 / 17.44
ORCA + SR	<u>7.90</u>	<u>24.40 / 18.85</u>
ORCA + NILE	<b>8.21</b>	<b>32.12 / 29.82</b>

Table 5: Experiment results of more LLMs on Alpaca and OpenOrca datasets. The highest values are **bolded**, and the second highest is underlined. Complete results on more benchmarks are placed in Table 17.

to that from LLAMA-3 in KSR (extracted by FIXED DEMONSTRATION (FD) prompting described in Appendix A.4.2). Table 8 shows the comprehensive advantage of using LLAMA-3’s internal knowledge over using MISTRAL’s. Switching from MISTRAL to LLAMA-3 increases **Arena-Hard** by 1.6 and 1.2 points in LLAMA-3 model on the Alpaca and Orca dataset. It is also interesting to see that using internal knowledge from MISTRAL has a huge negative

<sup>8</sup><https://huggingface.co/meta-llama/Llama-3.2-3B>

<sup>9</sup><https://huggingface.co/Qwen/Qwen2.5-7B>

<sup>10</sup><https://huggingface.co/Qwen/Qwen2.5-14B>

Method	A.-H. $\uparrow$	A.-E. V2 $\uparrow$	MTB. $\uparrow$	BBH $\uparrow$
A.+K. w. FD	<b>4.80</b>	10.75 / 9.38	6.67	60.73
A.+K. w. FS 1 IKE	<u>4.50</u>	<b>11.20</b> / <u>9.75</u>	<u>6.72</u>	59.25
A.+K. w. FS 2 IKE	<u>4.50</u>	<u>10.82</u> / <b>10.56</b>	<b>6.76</b>	<b>61.40</b>
O.+K. w. FD	<u>5.20</u>	<b>13.67</b> / 11.21	<b>7.51</b>	64.03
O.+K. w. FS 1 IKE	4.90	12.46 / 10.99	7.40	63.89
O.+K. w. FS 2 IKE	<b>5.50</b>	<u>13.00</u> / <b>11.50</b>	<u>7.43</u>	<b>64.29</b>

Table 6: Effects of IKE with different fewshot numbers (FS) in LLAMA-3. The highest values are **bolded**, and the second highest is underlined. For brevity, ALPACA + KSR and ORCA + KSR are abbreviated as A.+K. and O.+K., respectively.

Method	A.-H. $\uparrow$	A.-E. V2 $\uparrow$	MTB. $\uparrow$	BBH $\uparrow$
A.+N. wo. ICF	<u>4.50</u>	<b>10.82</b> / <b>10.56</b>	6.76	61.40
A.+N. w. ICF (LOW)	<b>4.80</b>	<u>10.69</u> / <u>10.43</u>	<b>6.90</b>	61.40
A.+N. w. ICF (MED.)	4.30	<u>9.99</u> / 9.81	6.65	<u>61.56</u>
A.+N. w. ICF (HIGH)	<u>4.50</u>	9.92 / 9.70	<u>6.79</u>	<b>61.71</b>
O.+N. wo. ICF	<u>5.50</u>	13.00 / 11.50	<u>7.43</u>	<b>64.29</b>
O.+N. w. ICF (LOW)	<b>6.00</b>	<b>13.70</b> / <b>12.11</b>	<b>7.48</b>	64.05
O.+N. w. ICF (MED.)	5.00	<u>13.27</u> / <u>11.59</u>	<u>7.43</u>	64.09
O.+N. w. ICF (HIGH)	4.80	13.19 / 11.49	7.30	63.95

Table 7: Effects of ICF using different  $\beta$  parameters in LLAMA-3. The highest values are **bolded**, and the second highest is underlined. ALPACA + NILE and ORCA + NILE are abbreviated as A.+N. and O.+N. for simplicity.

impact on LLAMA-3 on the **BBH** task requiring expert knowledge and complex reasoning, further highlighting the importance of such consistency. This suggests that maintaining general consistency between world knowledge from datasets and LLM internal knowledge is of necessity in effective IFT.

Method	A.-H. $\uparrow$	A.-E. V2 $\uparrow$	MTB. $\uparrow$	BBH $\uparrow$
A.+K. (MISTRAL)	4.00	9.14 / 7.29	6.64	57.67
A.+K. (LLAMA)	<b>4.80</b>	<b>10.75</b> / <b>9.38</b>	<b>6.67</b>	<b>60.73</b>
O.+K. (MISTRAL)	5.10	12.50 / 10.25	5.93	22.32
O.+K. (LLAMA)	<b>5.20</b>	<b>13.67</b> / <b>11.21</b>	<b>7.51</b>	<b>64.03</b>

Table 8: Effects of KSR in LLAMA-3 finetuned with internal knowledge from different LLMs. The highest values are **bolded**. Here ALPACA + KSR and ORCA + KSR are abbreviated as A.+K. and O.+K. for brevity.

**Effects of IKE Fewshot Number** Table 6 examines how different few-shot numbers of demonstration learning in IKE affect LLM performance. Here we evaluate three variants: 1) w. FD, which extracts LLM’s internal knowledge with a fixed 2-shot demonstration described in A.4.2; 2) w. FS 1 IKE, which retrieves the top 1 most similar samples with BM25 as demonstrations; and 3) w. FS 2 IKE, which retrieves the top 2 most similar samples with BM25 as demonstrations; Though w. FS 2 IKE leads to degradation in some benchmarks, such as **Arena-Hard** for ALPACA and **Alpaca-Eval** for ORCA, it still achieves overall improvements with **BBH** for ALPACA increasing by 0.7 and **Arena-**

**Hard** for ORCA increasing by 0.3. The results show that IKE is necessary for unaligned LLMs to more effectively extract internal knowledge, while fixed prompting reaches subpar performance.

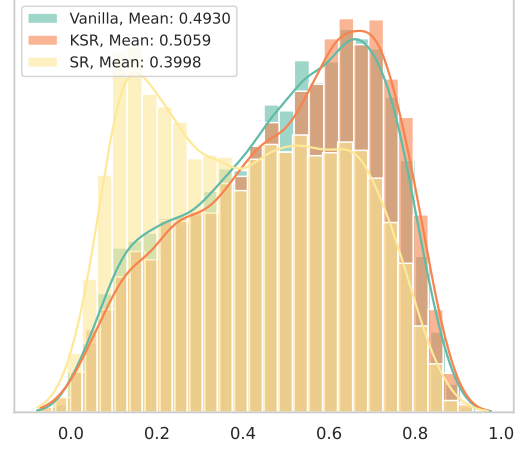


Figure 3: Distribution plot of sentence embedding similarity score in ALPACA dataset for MISTRAL model.

**Effects of KSR** We evaluated KSR’s effectiveness in improving internal consistency between world knowledge from instructions and the model’s internal knowledge. Our experiments assessed the degree to which responses incorporated internal knowledge across various models. We compared the models’ vanilla, KSR-generated, and SR-generated responses for 10K randomly sampled instructions by calculating sentence similarity scores. As shown in Figure 3, outputs generated by KSR exhibit a similarity score distribution significantly closer to 1 compared to SR and the vanilla baseline, with Chi-squared test p-values below 0.01. Additional results in A.4.3 further validate these findings. These results strongly support the effectiveness of KSR in enhancing internal consistency by integrating world and internal knowledge.

**Effects of ICF** Table 7 looks into the effect of ICF.  $\beta$  is set to 1-st percentile in ALPACA + NILE w. ICF (LOW) and to 2-nd percentile in ORCA + NILE w. ICF (LOW). We set  $\beta$  to 5-th and 10-th percentile for NILE w. ICF (MEDIUM) and NILE w. ICF (HIGH). The results empirically prove that striking a balance between consistent and inconsistent knowledge in the IFT dataset is necessary for NILE to achieve ideal performance. We find the general advantage of ALPACA + NILE w. ICF (LOW) over ALPACA + NILE w. ICF (MEDIUM) and ALPACA + NILE w. ICF (HIGH) and discarding ICF (ALPACA + NILE wo. ICF), indicating that a surplus of overly consistent or in-



consistent samples in IFT datasets both hurt LLM’s performance, and it is crucial to find the middle ground in these samples. This experiment further verifies our design choices of the ICF phase.

## 5 Conclusion

We present NILE, an innovative framework designed to enhance training efficacy of IFT datasets by aligning them with LLMs’ internal knowledge. Our extensive experiments demonstrate substantial improvements across various benchmarks, highlighting the crucial role of maintaining consistency between models’ internal knowledge and external knowledge in datasets. Each component of the NILE framework has been validated, reinforcing its importance in achieving better alignment. NILE offers promising directions for boosting the capabilities of LLMs and unlocking their full potential.

## Limitations

While NILE can already obtain satisfactory performance, future works should expand NILE’s training by utilizing the complete OpenOrca dataset rather than the current 50,000-sample subset (5% of the dataset), due to limited time and computational resources. To ensure a fair comparison of experiments in our study, we maintain a consistent dataset size by reverting to the original answer rather than discarding samples during the ICF phase. In future work, we aim to explore more advanced data selection techniques for the ICF process. Additionally, future research should examine NILE’s capability for iterative instruction refinement, as the current implementation uses only a single revision pass. These expansions could further enhance NILE’s instruction-following capabilities.

## Ethics Statement

We conducted this study strictly under the guidance of community ethical principles. The utilized IFT datasets are reported to be safe and free of content that may contain discrimination, personally identifiable information, or any other undesirable behaviors. We meticulously curate our instructions to the LLMs to ensure that the tasks are limited to knowledge generation and knowledge-relevant revisions, thereby avoiding content that may pose ethical concerns.

## Acknowledgement

The work described in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 2410072, RGC R1015-23). As the first author, I would like to express my heartfelt gratitude to my family, co-authors, and advisor, Prof. Irwin King, for their unwavering support and invaluable guidance throughout this work.

I would also like to dedicate this acknowledgment to the memory of 方大同 (*Khalil Fong*), an extraordinarily talented and influential R&B singer-songwriter. During the countless days and nights spent on this research and my PhD journey, his music and life philosophy provided me with comfort and strength to overcome every obstacle. His passing on February 21, 2025, was a profound loss for Chinese pop music and for me personally. I believe his vision and influence will continue to accompany us, transcending time and mortality, into the distant future. 成長是永遠，離別是空懸。在千尋之外，我依然存在。 (*Growth is everlasting, and parting is always in suspense. Beyond searches, I still exist.*)

## References

- Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2023. [Instruction mining: Instruction data selection for tuning large language models](#). *arXiv preprint arXiv:2307.06290*.
- Xiuxi Chen, Hongzhi Wen, Sreyashi Nag, Chen Luo, Qingyu Yin, Ruirui Li, Zheng Li, and Wei Wang. 2024. [IterAlign: Iterative constitutional alignment of large language models](#). 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 1423–1433, Mexico City, Mexico. Association for Computational Linguistics.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Forty-first International Conference on Machine Learning*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E

- Gonzalez, et al. 2024. [Chatbot arena: An open platform for evaluating llms by human preference](#). *ArXiv preprint*, abs/2403.04132.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. [Enhancing chat language models by scaling high-quality instructional conversations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. [The llama 3 herd of models](#). *ArXiv preprint*, abs/2407.21783.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. [Length-controlled alpaca-eval: A simple way to debias automatic evaluators](#). *ArXiv preprint*, abs/2404.04475.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.
- Yiming Huang, Xiao Liu, Yeyun Gong, Zhibin Gou, Yelong Shen, Nan Duan, and Weizhu Chen. 2025. Key-point-driven data synthesis with its enhancement on mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24176–24184.
- AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de las Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. 2023. [Mistral 7b \(2023\)](#). *ArXiv preprint*, abs/2310.06825.
- Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. [Unfamiliar finetuning examples control how language models hallucinate](#). *ArXiv preprint*, abs/2403.05612.
- Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. [Unsupervised dense retrieval with relevance-aware contrastive pre-training](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.
- Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. 2024a. [Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 16189–16211. Association for Computational Linguistics.
- Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023. [From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning](#). *ArXiv preprint*, abs/2308.12032.
- Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. 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.
- Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024c. [From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline](#). *ArXiv preprint*, abs/2406.11939.
- Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*.
- Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2024d. [One-shot learning as instruction data prospector for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4586–4601. Association for Computational Linguistics.
- Yilun Liu, Shimin Tao, Xiaofeng Zhao, Ming Zhu, Wenbing Ma, Junhao Zhu, Chang Su, Yutai Hou, Miao Zhang, Min Zhang, et al. 2024. Coachlm: Automatic instruction revisions improve the data quality in llm instruction tuning. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 5184–5197. IEEE.
- Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.
- Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. [Orca: Progressive learning from complex explanation traces of gpt-4](#). *ArXiv preprint*, abs/2306.02707.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. [Instruction tuning with gpt-4](#). *ArXiv preprint*, abs/2304.03277.
- Renjie Pi, Jianshu Zhang, Tianyang Han, Jipeng Zhang, Rui Pan, and Tong Zhang. 2024a. Personalized visual instruction tuning. *arXiv preprint arXiv:2410.07113*.

- Renjie Pi, Jianshu Zhang, Jipeng Zhang, Rui Pan, Zhekai Chen, and Tong Zhang. 2024b. Image textualization: An automatic framework for creating accurate and detailed image descriptions. *arXiv preprint arXiv:2406.07502*.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.
- Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Wan Guanglu, Xunliang Cai, and Le Sun. 2024. [Learning or self-aligning? rethinking instruction fine-tuning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6090–6105, Bangkok, Thailand. Association for Computational Linguistics.
- Yunfan Shao, Linyang Li, Yichuan Ma, Peiji Li, Demin Song, Qinyuan Cheng, Shimin Li, Xiaonan Li, Pengyu Wang, Qipeng Guo, et al. 2024. Case2code: Scalable synthetic data for code generation. *arXiv preprint arXiv:2407.12504*.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. [Beyond the imitation game: Quantifying and extrapolating the capabilities of language models](#). *ArXiv preprint*, abs/2206.04615.
- Zhiqing Sun, Yikang Shen, Qinzhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2024. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. [Challenging BIG-bench tasks and whether chain-of-thought can solve them](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: a strong, replicable instruction-following model; 2023. *URL* <https://crfm.stanford.edu/2023/03/13/alpaca.html>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *ArXiv preprint*, abs/2307.09288.
- Hongru Wang, Deng Cai, Wanjun Zhong, Shijue Huang, Jeff Z. Pan, Zeming Liu, and Kam-Fai Wong. 2025. [Self-reasoning language models: Unfold hidden reasoning chains with few reasoning catalyst](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5578–5596, Vienna, Austria. Association for Computational Linguistics.
- Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. 2023a. [Cue-CoT: Chain-of-thought prompting for responding to in-depth dialogue questions with LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12047–12064, Singapore. Association for Computational Linguistics.
- Rui Wang, Hongru Wang, Fei Mi, Boyang Xue, Yi Chen, Kam-Fai Wong, and Ruifeng Xu. 2024. [Enhancing large language models against inductive instructions with dual-critique prompting](#). 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 5345–5363, Mexico City, Mexico. Association for Computational Linguistics.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. 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.
- 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.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krma Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujana Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. [Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. [Finetuned language models are zero-shot learners](#). In *The Tenth*



*International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022.* OpenReview.net.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. [Wizardlm: Empowering large language models to follow complex instructions.](#)

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. [Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing.](#)

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. [Instruction tuning for large language models: A survey.](#) *ArXiv preprint*, abs/2308.10792.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Yipeng Zhang, Haitao Mi, and Helen Meng. 2024. [Self-tuning: Instructing llms to effectively acquire new knowledge through self-teaching.](#)

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Xuanhe Zhou, Junxuan He, Wei Zhou, Haodong Chen, Zirui Tang, Haoyu Zhao, Xin Tong, Guoliang Li, Youmin Chen, Jun Zhou, et al. 2025. A survey of llm  $\times$  data. *arXiv preprint arXiv:2505.18458*.

## A Appendix

### A.1 Implementation Details of NILE

For all experiments in this work, we use the Python 3.10.14 environment and vLLM 0.5.5 library<sup>11</sup> for LLM local inference of both MISTRAL and LLAMA-3. For vLLM inference hyperparameters, we set the random seed to 42, max\_tokens to 1024, temperature to 0.7, top\_k to 50, top\_p to 0.7, and repetition\_penalty to 1. We run all experiments on a server with an Intel Xeon Silver 4309Y CPU and 8 Nvidia RTX A6000 GPU having 48GB GDDR6 VRAM, and we utilize official checkpoints MISTRAL-7B-v0.3<sup>12</sup> for MISTRAL and META-LLAMA-3.1-8B<sup>13</sup> for LLAMA-3. For LLM instruction fine-tuning in this work, we choose llama-recipes<sup>14</sup> for LLAMA-3 and

alignment-handbook<sup>15</sup> for MISTRAL. For LLAMA-3 fine-tuning, we set context\_length to 2048, gradient\_accumulation\_step to 32, learning rate to 2e-5, and training batch size to 4. As for MISTRAL fine-tuning, we set context\_length to 2048, gradient\_accumulation\_step to 32, learning rate to 2e-5, training batch size to 4, lr\_scheduler\_type to "cosine", num\_train\_epochs to 3, and warmup\_ratio to 0.1. Fine-tuning for both LLAMA-3 and MISTRAL is done within 5 hours using 8 A6000 GPU.

#### A.1.1 Internal Knowledge Extraction (IKE)

For demonstration sample, we randomly sample  $m = 5,000$  instruction pairs  $q_i^d = \{\text{instruction}_i^d, \text{input}_i^d\}$  from Alpaca dataset<sup>16</sup>, since instructions in it are simple and straightforward, which is suitable for LLM demonstration learning. We leverage GPT-4-TURBO-2024-04-09 through Openai API for generating demonstrations given  $q_i^d$  shown in Table 1. For GPT-4 endpoints, we use openai 1.42.0 python library and set n to 1, temperature to 0.7, and max\_tokens to 1,024. We stick to the regulations from the OpenAI company when accessing its API. For retriever  $\mathcal{R}$  in IKE, we choose the BM25 and Contriver implementation<sup>17</sup> from the Pyserini 0.38.0 library.

#### A.1.2 Design Choice: BM25 vs Contriver

The performance gain of the NILE choosing BM25 over Contriver in IKE is shown in Table 9.

#### A.1.3 IKE Sample Demonstration

Table 10 illustrates the sample 2-shot demonstration set from IKE, and Table 11 shows two samples from the demonstration database  $\mathcal{F}^{demo}$  in IKE.

#### A.1.4 Knowledge-aware Sample Revision (KSR)

For KSR, we also use GPT-4-TURBO-2024-04-09 endpoint as revisor agent  $\mathcal{A}_r$ . We use openai 1.42.0 python library with n set to 1, temperature to 0.7, and max\_tokens to 1,024. We run KSR on 52,000 samples from the Alpaca dataset and 50,000 samples from the OpenOrca dataset. Case studies of KSR can be found in Table 19. These results display the KSR’s capability to infuse internal knowledge information into original answers through revision.

<sup>11</sup><https://github.com/vllm-project/vllm>

<sup>12</sup><https://huggingface.co/mistralai/Mistral-7B-v0.3>

<sup>13</sup><https://huggingface.co/meta-llama/Llama-3.1-8B>

<sup>14</sup><https://github.com/meta-llama/llama-recipes>

<sup>15</sup><https://github.com/huggingface/alignment-handbook>

<sup>16</sup><https://huggingface.co/datasets/vicgalle/alpaca-gpt4>

<sup>17</sup><https://github.com/castorini/pyserini>



Method	Arena-Hard $\uparrow$	Alpaca-Eval V2 $\uparrow$	MTBench $\uparrow$	BBH $\uparrow$
ORCA + NILE W. CONTRIVER	4.70	<b>14.63 / 12.17</b>	7.29	64.00
ORCA + NILE W. BM25	<b>5.50</b>	13.00 / 11.50	<b>7.43</b>	<b>64.29</b>

Table 9: Comparison between choosing BM25 and Contriver. The highest values are **bolded**.

**Instruction**  $instruction^o$ : *Recommend 3 books that could get someone into a new field of study.*

**Input**  $input^o$ : None

**2-shot Demonstration**  $f^{demo}(q^o) = \{(q_1^{\mathcal{R}}, ik_1^{\mathcal{R}}), (q_2^{\mathcal{R}}, ik_2^{\mathcal{R}})\}$ :

$q_1^{\mathcal{R}}$ : Provide 3 pieces of advice for someone who is struggling to stay motivated in their studies.

$ik_1^{\mathcal{R}}$ : 1. Understanding Motivation: It is essential to comprehend the concept of motivation, including intrinsic and extrinsic motivation. Intrinsic motivation refers to doing something because it is inherently interesting or enjoyable, while extrinsic motivation involves doing something because it leads to a separable outcome. Understanding these types of motivation can help identify what drives an individual’s desire to study and how to bolster it.

2. Goal Setting: Knowledge of effective goal-setting strategies is crucial. Setting specific, measurable, achievable, relevant, and time-bound (SMART) goals can help create a clear roadmap for studying. This process can help break down the overwhelming feeling into manageable tasks, making it easier to stay motivated.

.....

$q_2^{\mathcal{R}}$ : Please recommend a movie for someone who likes animated films.

$ik_2^{\mathcal{R}}$ : 1. Understanding Animated Films: Animated films, also known as cartoons, use drawn or computer-generated imagery to create the illusion of movement. They can be in 2D, 3D, stop-motion or other animation techniques. Knowledge of the different styles and techniques of animation is crucial to recommending films that align with specific preferences.

2. Popular Animation Studios: Some of the most prominent animation studios include Disney, Pixar, DreamWorks, Studio Ghibli, and Illumination. Each studio has its own unique style and storytelling approach. Familiarity with the filmographies of these studios can help in making informed recommendations.

3. Genres within Animation: Animated films encompass a wide range of genres, from adventure and fantasy to drama and comedy. Some animated films are targeted towards children, while others might appeal to a broader age range including adults. Understanding the various genres and their target audiences can guide the recommendation process.

4. Cultural and Thematic Elements: Animated films often incorporate diverse cultural stories and themes. Films might explore complex themes like identity, family, and morality, or might be more light-hearted and comedic. Awareness of these elements can help match a film to a viewer’s personal interests and values.

.....

Table 10: Sample 2-shot demonstration in IKE.

### A.1.5 Sample Revision (SR)

Unlike KSR, SAMPLE REVISION (SR) does not revise for each instruction pair  $(q^o, a^o)$ . Therefore,  $\mathcal{A}_r$  in SR only uses external knowledge such as world knowledge from  $(q^o, a^o)$  and its own parameter knowledge, being completely isolated from internal knowledge  $ik$  of  $\mathcal{M}$ . Table 12 shows the

detailed prompt of the revisor  $\mathcal{A}_r$  in SR.

### A.1.6 Statistics of Inference Overhead

In the IKE step of NILE, internal knowledge is efficiently extracted from sample datasets within 6 hours using vLLM on an 8-A6000 GPU server. To further show how much inference overhead is intro-

$q_1^d$  : What would be the best type of exercise for a person who has arthritis?

$ik_1^d$  : 1. Understanding Arthritis: Arthritis is a chronic condition characterized by inflammation in the joints, which can cause pain, stiffness, and reduced mobility. There are different types of arthritis, such as osteoarthritis and rheumatoid arthritis, each with varying symptoms and treatment approaches.

2. Impact of Exercise on Arthritis: Exercise is generally considered beneficial for individuals with arthritis. It can help to reduce joint pain, increase flexibility, improve muscle strength, enhance endurance, and promote overall physical function.

3. Types of Exercise Suitable for Arthritis:

- Low-impact Aerobic Activities: These exercises are gentle on the joints and include walking, swimming, and cycling. They help in cardiovascular conditioning without putting excessive stress on the joints.
- Resistance Training: Using light weights or resistance bands can help strengthen the muscles around the joints, providing better support and reducing the burden on the joints.
- Flexibility Exercises: Activities such as stretching and yoga can improve joint flexibility and range of motion, helping to alleviate stiffness.

.....

$q_2^d$  : Calculate the atomic mass for lithium.

$ik_2^d$  : 1. Understanding Atomic Mass: Atomic mass, also known as atomic weight, is defined as the weighted average mass of atoms of an element based on the abundance of each isotope of the element in nature. It is usually measured in atomic mass units (amu).

2. Isotopes: Isotopes are atoms of the same element that have the same number of protons but different numbers of neutrons. This results in different mass numbers for each isotope. The atomic mass of an element is calculated by taking into account the masses and relative abundances of all its naturally occurring isotopes.

3. Lithium Isotopes: Lithium has two stable isotopes, lithium-6 ( ${}^6\text{Li}$ ) and lithium-7 ( ${}^7\text{Li}$ ). These isotopes differ in their neutron count, affecting their individual atomic masses. Lithium-6 has 3 neutrons, while lithium-7 has 4 neutrons.

4. Natural Abundance: The natural abundance of an isotope refers to the percentage of that isotope found naturally in a sample of the element. For lithium, lithium-7 is more abundant than lithium-6. The exact percentages of natural abundance can vary slightly depending on the source, but generally, lithium-7 accounts for about 92.5% while lithium-6 is about 7.5%.

.....

Table 11: Samples from demonstration database  $\mathcal{F}^{demo}$  in IKE.

Provide a better response based on " $\{a^o\}$ " to comply with given instruction, input, and related knowledge.

Instruction:  $\{\text{instruction}^o\}$   
Input:  $\{\text{input}^o\}$

Please directly output the improved response.

Table 12: Prompt for Sample Revision.

duced, we measured the average token usage per sample for this step, which is detailed in Table 13.

Model / Dataset	LLAMA	MISTRAL
OPENORCA	684.8	801.1
ALPACA	620.6	544.5

Table 13: Average generated tokens per sample using vLLM during IKE across different datasets and models.

The Knowledge-aware Sample Revision (KSR) step further optimizes efficiency, with GPT-4 achieving modest token usage per sample, also

shown in Table 14. This results in exceptionally low operational costs, making our approach scalable, cost-effective, and practical for real-world applications.

Model / Dataset	LLAMA	MISTRAL
OPENORCA	\$0.015/182.0	\$0.016/182.1
ALPACA	\$0.013/193.7	\$0.011/165.3

Table 14: Price (in USD) and average generated tokens per sample during KSR across different datasets and models.

## A.2 Evaluating NILE’s Effectiveness on External Knowledge-Intensive Tasks

To briefly examine some of NILE’s potential issues and limitations, we conducted an additional experiment on the SQuADv2 (Rajpurkar et al., 2018) validation set using our sampled Alpaca-GPT4 dataset under the same settings outlined in our paper. The SQuADv2 validation set was chosen because it contains 119,000 test samples of reading comprehension, where large language models (LLMs) must answer questions based on external knowledge provided in corresponding supporting passages. As such, it serves as a suitable and rigorous benchmark for evaluating an LLM’s ability to comprehend and utilize external knowledge effectively. The results in Table 16 demonstrate that NILE can positively influence this capability.

## A.3 Examining NILE’s Performance on Multitask Accuracy

In order to further test the extensiveness of NILE’s improvement on LLMs, we have conducted additional experiments on MMLU using two models (META-LLAMA-3.1-8B and MISTRAL-7B-V0.3) and two IFT datasets (Alpaca and Orca). All experiments adhered to the official default configuration from the lm-evaluation-harness implementation<sup>18</sup>. The results presented in Table 15 demonstrate that NILE consistently achieves significant performance improvements in highly complex QA settings like MMLU. Interestingly, we observed a noticeable dip in accuracy with the SR baseline across both datasets and models. This result further underscores the necessity of incorporating internal

knowledge within the NILE framework to enhance alignment in IFT datasets.

Method	Accuracy (%) ↑
MISTRAL-7B-V0.3	
ALPACA VANILLA	57.21
ALPACA + SR	56.25
ALPACA + NILE	<b>57.56</b>
ORCA VANILLA	<b>56.92</b>
ORCA + SR	54.54
ORCA + NILE	56.91
META-LLAMA-3.1-8B	
ALPACA VANILLA	62.51
ALPACA + SR	62.41
ALPACA + NILE	<b>63.93</b>
ORCA VANILLA	62.68
ORCA + SR	62.19
ORCA + NILE	<b>63.09</b>

Table 15: Experiment results of NILE on MMLU benchmark. The highest values are **bolded**.

Method	EM ↑	F1 ↑
MISTRAL-7B-V0.3		
ALPACA VANILLA	4.91	14.13
ALPACA + NILE	<b>5.61</b>	<b>14.51</b>
META-LLAMA-3.1-8B		
ALPACA VANILLA	1.41	9.35
ALPACA + NILE	<b>4.51</b>	<b>11.65</b>

Table 16: Experiment results of NILE on SQuADv2 dataset. The highest values are **bolded**.

## A.4 Experiment Details

### A.4.1 Benchmarks

We use the officially recommended settings from all benchmarks for evaluation. For Alpaca-Eval V2, we use "alpaca\_eval\_cot\_gpt4\_turbo\_fn" as annotators, and we set max\_new\_tokens to 1024, temperature to 1.0, top\_p to 1.0, and batch\_size to 128. For Arena-Hard, we set the temperature to 0.0, max\_tokens to 1024, judge\_model to gpt-4-1106-preview, baseline\_model to gpt-4-0314, and num\_choices to 1. For the Arena-Hard and Alpaca-Eval V2 benchmark, we keep the same alpaca-style system prompt as the fine-tuning stage during evaluation. As for BBH and MTBench, we use the default settings in the official source code.

<sup>18</sup>[https://github.com/EleutherAI/lm-evaluation-harness/tree/main/lm\\_eval/tasks/mmlu](https://github.com/EleutherAI/lm-evaluation-harness/tree/main/lm_eval/tasks/mmlu)

Method	Arena-Hard $\uparrow$	Alpaca-Eval V2 $\uparrow$	MTBench $\uparrow$	BBH $\uparrow$
META-LLAMA-3.2-3B				
ALPACA VANILLA	3.50	6.17 / 3.54	5.52	<u>43.63</u>
ALPACA + SR	<u>3.00</u>	<u>6.18 / 4.43</u>	<u>5.65</u>	43.02
ALPACA + NILE	<b>4.60</b>	<b>6.61 / 5.10</b>	<b>5.94</b>	<b>43.74</b>
ORCA VANILLA	3.60	5.61 / 4.41	2.20	<u>43.36</u>
ORCA + SR	<u>4.00</u>	<u>6.18 / 5.51</u>	<u>3.06</u>	41.87
ORCA + NILE	<b>4.20</b>	<b>10.77 / 8.46</b>	<b>4.63</b>	<b>45.69</b>
QWEN2.5-7B				
ALPACA VANILLA	11.90	13.84 / 7.83	<u>7.13</u>	<b>45.95</b>
ALPACA + SR	<u>14.40</u>	<u>15.40 / 8.90</u>	6.78	<u>45.94</u>
ALPACA + NILE	<b>18.40</b>	<b>17.42 / 12.24</b>	<b>8.13</b>	45.72
ORCA VANILLA	<u>14.90</u>	<u>19.19 / 13.42</u>	6.60	46.89
ORCA + SR	13.60	18.45 / <u>14.34</u>	<u>7.05</u>	<b>49.32</b>
ORCA + NILE	<b>17.10</b>	<b>20.55 / 16.58</b>	<b>7.31</b>	<u>48.61</u>
QWEN2.5-14B				
ALPACA VANILLA	12.80	15.37 / 8.16	7.33	48.01
ALPACA + SR	<u>20.00</u>	<u>21.56 / 12.59</u>	<u>7.73</u>	48.69
ALPACA + NILE	<b>22.60</b>	<b>28.55 / 17.45</b>	<b>8.06</b>	<b>49.01</b>
ORCA VANILLA	24.00	19.99 / 17.44	7.68	48.49
ORCA + SR	<u>26.60</u>	<u>24.40 / 18.85</u>	<u>7.90</u>	<u>47.66</u>
ORCA + NILE	<b>31.90</b>	<b>32.12 / 29.82</b>	<b>8.21</b>	<b>49.96</b>

Table 17: Complete experiment results of more LLMs on Alpaca and OpenOrca datasets for **Arena-Hard**, **Alpaca-Eval V2 LCWR**, **MTBench**, and **BBH** benchmarks. The highest values are **bolded**, and the second highest is underlined.

#### A.4.2 Fixed Demonstration (FD)

Table 18 provides the prompt of the Fixed Demonstration (FD) used for extracting LLM internal knowledge in the experiments. The FD employs a fixed set of 2-shot demonstrations, serving as a baseline for IKE without incorporating demonstration learning.

#### A.4.3 Effects of KSR

To validate KSR’s effectiveness in enhancing internal consistency between world knowledge from instructions and the model’s internal knowledge, we conducted experiments measuring the similarity between extracted internal knowledge and baseline knowledge across different models and datasets. In LLAMA-3 and MISTRAL, we used the instructions from the Alpaca and Orca as prompts to evaluate the models’ internal knowledge. Then, we obtained the models’ vanilla output for these instructions, the output adjusted using KSR, and the output using SR. We randomly sampled 10,000

instructions to calculate the sentence similarity between these three outputs and the internal knowledge. As demonstrated in Figure 4, the similarity score distribution of the outputs adjusted by KSR is significantly closer to 1 compared to the other two methods, with Chi-squared test p-values lower than 0.01.

These results provide strong evidence supporting the effectiveness of the KSR approach in increasing the internal consistency from instructions by integrating relevant world and internal knowledge. The case study of KSR is listed in Table 19.



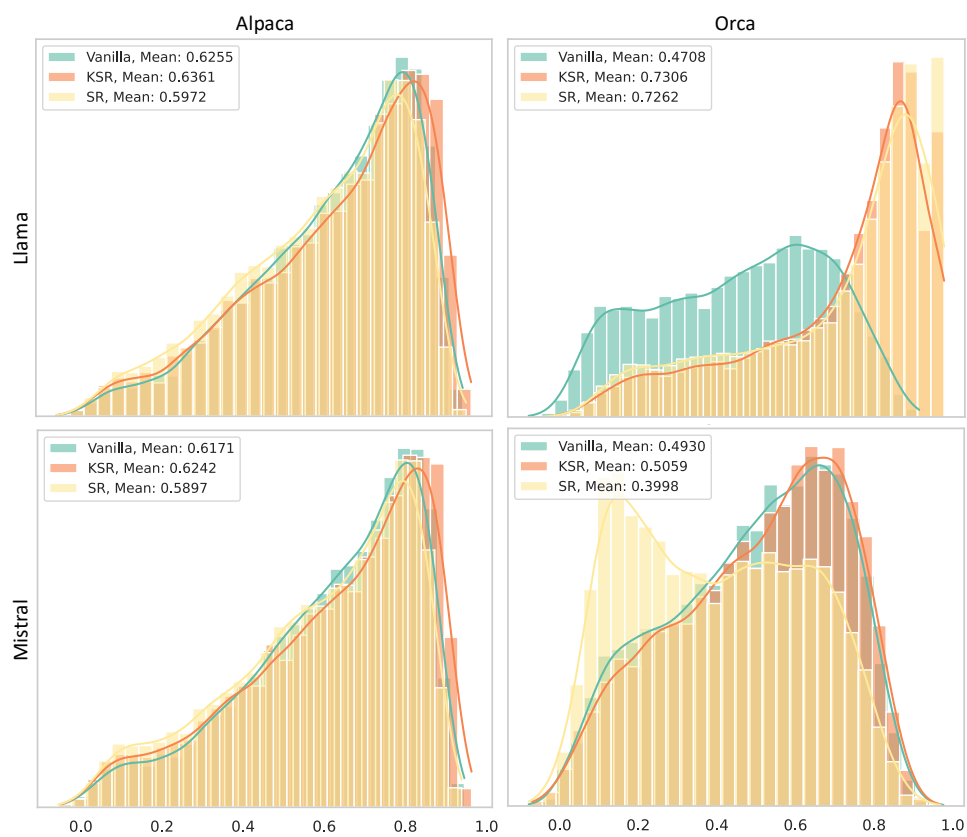


Figure 4: Distribution of sentence embedding similarity across different LLMs and IFT datasets.

Instruction:

"Give three tips for staying healthy."

Related Knowledge:

\* Importance of health: Maintaining good health is crucial for overall well-being and quality of life. \* Factors affecting health: A person's health can be influenced by various factors such as diet, exercise, sleep, stress, and genetics. \* Prevention is key: Preventing illnesses and maintaining good health requires adopting healthy habits and making lifestyle changes. \* Healthy habits: Developing healthy habits such as regular exercise, balanced diet, and adequate sleep can help prevent chronic diseases. \* Lifestyle changes: Making lifestyle changes such as quitting smoking, reducing sugar intake, and managing stress can also contribute to good health. \* Importance of self-care: Taking care of one's physical, emotional, and mental health is essential for overall well-being. \* Access to healthcare: Having access to quality healthcare and medical facilities is also important for staying healthy. \* Healthy behaviors: Engaging in healthy behaviors such as regular check-ups, vaccinations, and screenings can help prevent illnesses and detect health issues early. \* Health literacy: Having knowledge and understanding of health-related information is important for making informed decisions about one's health.

Instruction:

"What are the three primary colors?"

Related Knowledge:

\* Primary colors are colors that cannot be created by mixing other colors together. \* The three primary colors are: 1. Red 2. Blue 3. Yellow \* Primary colors are the base colors used to create all other colors. \* By mixing different combinations of primary colors, you can create secondary colors, tertiary colors, and a wide range of shades and hues. \* Primary colors are often used in art, design, and painting to create bold and vibrant colors. \* The primary colors are also used in color theory to understand how colors interact with each other and how they can be used to create contrast, harmony, and balance. \* In addition to art and design, primary colors are also used in science, technology, engineering, and mathematics (STEM) fields, such as physics, chemistry, and biology, to describe and analyze the properties of light and color. \* The primary colors are a fundamental concept in many cultures and have been used in art and design for centuries, with examples found in ancient civilizations such as Egypt, Greece, and China.

Instruction:

{**instruction**<sup>o</sup>, **input**<sup>o</sup>}

Related Knowledge:

Table 18: Prompt for Fixed Demonstration (FD).

Instruction	Response	Internal Knowledge
<p><i>Compose a new email subject line for a following email.</i></p> <p><b>Input:</b> The email body contains a request to renew a subscription.</p>	<p><b>Vanilla:</b> "Reminder: Renew Your Subscription Today!"</p> <p><b>Revised:</b> "Action Required: Time to Renew Your Subscription!"</p>	<p>Email subject line: The subject line of an email is the first thing a recipient sees and can determine whether they open the email or not. It should be clear, concise, and relevant to the email content.* Subscription renewal: A subscription renewal is a request to continue receiving a service or product after the initial subscription period has ended. It is important to provide clear instructions and information about the renewal process.* Clear instructions: Providing clear instructions in the email body can help ensure that the recipient understands the request and knows what to do next. It is important to use simple language and avoid technical jargon.* Information about renewal process: Including information about the renewal process, such as the cost, payment options, and deadlines, can help the recipient make an informed decision and take the necessary steps to renew their subscription.* <b>Call-to-action: Including a clear call-to-action in the email body can help ensure that the recipient knows what to do next and takes the desired action. It is important to use a strong and direct call-to-action that is easy to understand and takes the recipient to the next step.</b></p>
<p><i>Write a sentence containing a hyperbolic metaphor.</i></p> <p><b>Input:</b> None</p>	<p><b>Vanilla:</b> He was so hungry, he could eat an entire elephant.</p> <p><b>Revised:</b> He was so ravenous, <b>he could devour a mountain of hamburgers as tall as Everest.</b></p>	<p>Hyperbole is a figure of speech that uses exaggeration for emphasis or effect.* <b>Hyperbolic metaphors are metaphors that use hyperbole to create a vivid and dramatic image.</b>* For example, "The crowd was so large that it stretched from one end of the stadium to the other."* This metaphor uses hyperbole to create a dramatic image of the crowd's size.* Hyperbolic metaphors can be used to create a strong emotional response or to make a point more effectively.* <b>They can also be used to add humor or to create a sense of exaggeration or absurdity.</b></p>
<p><i>Create a 3-5 sentence definition for the term "data science".</i></p> <p><b>Input:</b> None</p>	<p><b>Vanilla:</b> Data science is an interdisciplinary field that combines techniques from statistics, mathematics, computer science, and information science to extract insights and knowledge from data. This field deals with both structured and unstructured data, using algorithms, predictive models, and machine learning to make informed decisions. Data science has broad applications, from businesses and marketing to healthcare and scientific research, where it is used to answer questions, find patterns, and make strategic decisions.</p> <p><b>Revised:</b> Data science is an interdisciplinary field that uses methods from statistics, mathematics, computer science, and domain-specific knowledge to extract insights and information from both structured and unstructured data. It employs techniques such as machine learning, predictive modeling, and data mining to analyze data sets, identify patterns, and make predictions. <b>The ultimate goal of data science is to help organizations make informed decisions, optimize processes, and solve complex problems by leveraging vast amounts of data. This field is critical across various industries, including healthcare, finance, and technology, where data-driven insights are essential for strategic decision-making and operational improvements.</b></p>	<p>Data science is an interdisciplinary field that combines techniques from statistics, computer science, and domain knowledge to extract insights from data.* It involves the use of advanced analytics and machine learning algorithms to analyze large datasets and identify patterns, trends, and correlations.* Data scientists use various tools and techniques such as data mining, predictive modeling, and natural language processing to uncover insights and make predictions about future events.* <b>The goal of data science is to use data to solve complex problems, make better decisions, and drive business value.* Data science has become increasingly important in recent years due to the rapid growth of data and the need for organizations to make sense of it.* Data scientists play a critical role in many industries, including healthcare, finance, retail, and technology, where they help organizations gain insights from data to improve operations, optimize processes, and make better decisions.</b></p>

Table 19: Case study of KSR. Related KSR revisions and internal knowledge are marked in red.