Utilize the Flow before Stepping into the Same River Twice: Certainty Represented Knowledge Flow for Refusal-Aware Instruction Tuning

Runchuan Zhu[1,2*], **Zhipeng Ma**[3*], **Jiang Wu**[1*†], **Junyuan Gao**[1,4], **Jiaqi Wang**[1], **Dahua Lin**[1], **Conghui He**[1‡]

[1]Shanghai Artificial Intelligence Laboratory
[2]Peking University
[3]Southwest Jiaotong University
[4]University of Chinese Academy of Sciences
{zhurunchuan, wujiang, gaojuanyuan, wangjiaqi, lindahua, heconghui}@pjlab.org.cn,
mazhipeng1024@my.swjtu.edu.cn

Abstract

Refusal-Aware Instruction Tuning (RAIT) enables Large Language Models (LLMs) to refuse to answer unknown questions. By modifying responses of unknown questions in the training data to refusal responses such as "I don't know", RAIT enhances the reliability of LLMs and reduces their hallucination. Generally, RAIT modifies training samples based on the correctness of the initial LLM's response. However, this crude approach can cause LLMs to excessively refuse answering questions they could have correctly answered, the problem we call over-refusal. In this paper, we explore two primary causes of over-refusal: ***Static conflict*** occurs when similar samples within the LLM's feature space receive differing supervision signals (original vs. modified "I don't know"). ***Dynamic conflict*** arises as the LLM's evolving knowledge during SFT enables it to answer previously unanswerable questions, but the now-answerable training samples still retain the original "I don't know" supervision signals from the initial LLM state, leading to inconsistencies. These conflicts cause the trained LLM to misclassify known questions as unknown, resulting in over-refusal. To address this issue, we introduce Certainty Represented Knowledge Flow for Refusal-Aware Instructions Tuning (CRaFT). CRaFT centers on two main contributions: First, we additionally incorporate response certainty to selectively filter and modify data, reducing static conflicts. Second, we implement preliminary rehearsal training to characterize changes in the LLM's knowledge state, which helps mitigate dynamic conflicts during the fine-tuning process. We conducted extensive experiments on open-ended question answering and multiple-choice question task. Experiment results show that CRaFT can improve LLM's overall performance during the RAIT process.

Code & Data — https://github.com/opendatalab/CRaFT
Extended version — https://arxiv.org/abs/2410.06913

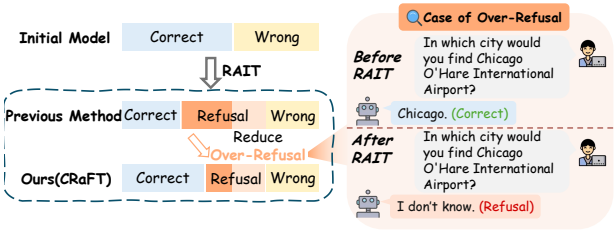

Figure 1: Previous RAIT methods resulted in significant over-refusal, while our CRaFT mitigates this issue, enhancing the LLM's reliability and helpfulness.

1 Introduction

Recently, Large Language Model (LLM) technology has made significant progress, becoming an important milestone towards AGI (Achiam et al. 2023; Dubey et al. 2024; Touvron et al. 2023; Yang et al. 2024). However, current LLMs often output fabricated information, which is referred to as hallucinations (Ji et al. 2023). This phenomenon severely limits the usefulness and reliability of LLMs. An important reason for the occurrence of hallucinations is that when exposed to questions beyond their internal knowledge (i.e., unknown questions), LLMs may forcefully generate responses (Kang et al. 2024). Ideally, the reliable LLM should actively refuse to answer questions it doesn't know to avoid incorrect responses (Wen et al. 2024; Li et al. 2024b). Recent studies have shown (Yang et al. 2023; Zhang et al. 2024; Xu et al. 2024a; Cheng et al. 2024; Xu et al. 2024a; Bai et al. 2023; Cheng et al. 2024) that Refusal-Aware Instruction Tuning (RAIT) can enable LLMs to refuse answering questions beyond their knowledge.

The RAIT process can be described as follows: the initial LLM answers all questions in the train set D_{src}. Based on response accuracy, samples are split into two groups. Correct responses are labeled as vanilla samples D_{van}, with unchanged answers, while incorrect responses are replaced with "I don't know," forming the IdK samples D_{idk}. The combined RAIT data $D_{\text{rait}} = D_{\text{van}} \cup D_{\text{idk}}$ is used to fine-tune the LLM, improving its ability to refuse unknown questions.

*These authors contributed equally.
†Project lead.
‡Corresponding author (heconghui@pjlab.org.cn).

This original RAIT method is referred to as Cor-RAIT.

However, (Cheng et al. 2024) shows that Cor-RAIT causes the fine-tuned LLM to refuse some questions that could have been answered correctly. Experiments reveal a significant accuracy drop after Cor-RAIT: on the TriviaQA dataset (Joshi et al. 2017), accuracy falls from 45.05% to 28.57%, and on the Natural Questions dataset (Kwiatkowski et al. 2019), it drops from 24.65% to 15.93%. We refer to this phenomenon as **over-refusal**, as shown in Figure 1.

In addressing the over-refusal brought by Cor-RAIT, we identified two primary causes as shown in Figure 2. **(1) Static Conflict**: In the LLM representation space, two closely located samples might be assigned to D_{van} and D_{idk} under the Cor-RAIT framework. As illustrated by t-SNE in Figure 3(a), significant intersections exist between D_{van} and D_{idk}, complicating their differentiation. These similar samples provide conflict supervision during training, impairing the LLM's ability to distinguish between known and unknown questions, resulting in over-refusal. **(2) Dynamic Conflict**: This arises from overlooking the dynamic shifts in LLM's knowledge state during training. Research (Ren et al. 2024; Ren and Sutherland 2024; Xu et al. 2024b) shows that the knowledge state of LLMs changes during Supervised Fine-Tuning (SFT), with questions potentially shifting from unknown to known and vice versa. This phenomenon is reminiscent of Heraclitus' saying, "*no man ever steps in the same river twice*." However, current methods use static RAIT data reflecting the initial LLM's knowledge state throughout SFT, which ignores these changes. This oversight leads to conflicts between the RAIT data and the LLM's evolving knowledge, resulting inefficient training and over-refusal.

To address the two problems above, we propose Certainty Represented Knowledge Flow for Refusal-Aware Instructions Construction (CRaFT). Our approach consists of two stages. **Stage 1: Querying the Knowledge State and Flow of the LLM.** First, we probe the initial LLM's knowledge state. Unlike Cor-RAIT, we incorporate response certainty alongside correctness, effectively alleviating the *static* conflict between the supervision signals in D_{van} and D_{idk}. To capture the LLM's dynamic knowledge changes during training, we introduce a rehearsal training mechanism. This fine-tunes the LLM with data samples that align closely with its internal knowledge, without introducing new knowledge (Ren et al. 2024; Kang et al. 2024). This approach allows us to observe the LLM's natural knowledge adjustments. The differences between the fine-tuned and initial LLMs reveal the knowledge flow during training, helping to identify and resolve *dynamic* conflicts. **Stage 2: Refusal-aware instructions construction and tuning**. By considering both the static knowledge state and dynamic knowledge flow, we filter out vanilla and IdK samples from RAIT data, reducing conflicts. We then fine-tune the initial LLM with the refined data, improving overall performance.

In conducting our experimental analysis, we sought a well-founded metric within current research. Existing methods have notable limitations, either proposing multiple metrics that are hard to optimize simultaneously or relying on inherently flawed metrics, as demonstrated by our counterexamples. Consequently, we examined these shortcomings and introduced a singular and comprehensive metric: **Truthful Helpfulness Score(THS)**.

Overall, our main contributions are as follows:

- We conducted the in-depth analysis of static and dynamic conflicts in existing correctness-based RAIT data, revealing that they cause the trained LLMs' mis-classification of known and unknown questions, leading to the issue of over-refusal in current RAIT methods.
- To address static and dynamic conflicts, we introduce CRaFT: it reduces static conflicts by incorporating certainty alongside correctness during RAIT data construction, and mitigates dynamic conflicts through rehearsal training to capture knowledge flow trends. Extensive experiments demonstrate that CRaFT alleviates over-refusal and improves overall LLM performance.
- We analyze the shortcomings of existing refusal-aware metrics and introduce the Truthful Helpfulness Score (THS), which balances reliability and helpfulness for a comprehensive evaluation of LLM performance.

2 Related Work

2.1 Mitigating Hallucinations of LLMs

Researchers have developed various methods to mitigate LLM hallucinations, including data augmentation (Neeman et al. 2022), improved decoding strategies (Holtzman et al. 2019; Chuang et al. 2023), external knowledge integration (Karpukhin et al. 2020), knowledge editing (Zhang, Yu, and Feng 2024; Li et al. 2024a), and honesty alignment (Zhang et al. 2024; Xu et al. 2024a; Bai et al. 2024). Unlike traditional correction methods, honesty alignment encourages models to say "I don't know" for unknown questions.

2.2 Refusal-Aware Instruction Tuning

RAIT is the supervised technique that improves LLMs' responses by training LLMs to directly respond with "I don't know" to unknown questions. R-Tuning (Zhang et al. 2024) identifies these questions by having the LLM answer each once and verifying response accuracy. In (Yang et al. 2023), the LLM answers the same question multiple times, with the target answer adjusted based on the correctness ratio. (Wan et al. 2024) uses a knowledge-based verification mechanism to ensure consistency with trusted external sources, enhancing refusal accuracy and preventing misinformation.

3 Over-Refusal: Analysis and Insights

3.1 Refusal-Aware Instruction Tuning

Given the initial LLM $\mathcal{M}_0$ and the instruction dataset D_{src} of question-answer pairs $x = (q, a)$, we modify D_{src} to construct D_{rait}, consisting of pairs (q, a_{rait}). D_{rait} is then used for SFT on $\mathcal{M}_0$, resulting in a new LLM capable of declining unknown questions, a process called Refusal-Aware Instruction Tuning (RAIT). Existing studies (Zhang et al. 2024; Yang et al. 2023; Cheng et al. 2024) use $\mathcal{M}_0$ to infer and assess the correctness of questions in D_{src}, denoted as μ. As shown in Figure 4(a), the correctness threshold τ_μ is first defined. If $\mu < \tau_\mu$, the answer is changed to "I don't know"

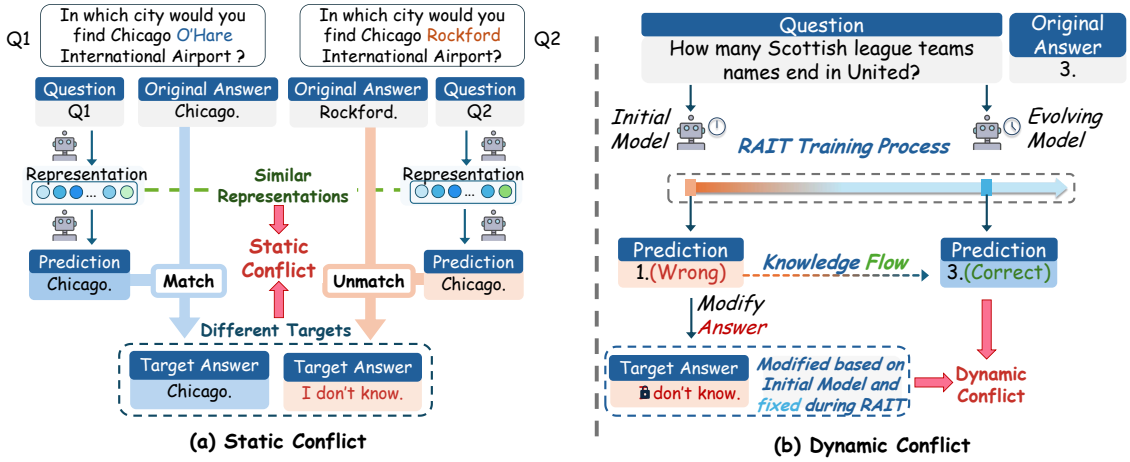

Figure 2: Two causes of over-refusal: (a) *Static conflict* means the similar samples in the LLM's feature space being assigned different labels (original vs. modified "I don't know"). (b) *Dynamic conflict* arises since the LLM's knowledge state evolves during SFT, turning initally unknown questions to knowns, while the target answer remains IdK. These conflicts cause the trained LLM to misclassify known questions as unknown, resulting in over-refusal.

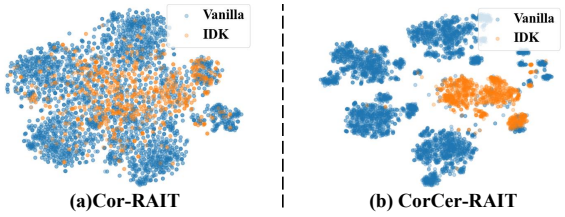

Figure 3: t-SNE visualization of the LLM feature space

and assigned to the IdK subset D_{idk}. If $\mu \geq \tau_{\mu}$, the original answer remains, and the pair is assigned to the vanilla subset D_{van}. The resulting RAIT data is $D_{\text{rait}} = D_{\text{van}} \cup D_{\text{idk}}$, and this correctness-based RAIT is called Cor-RAIT.

However, LLMs exhibited significant over-refusal after Cor-RAIT, as shown in Figure 1. Subsequent sections of this chapter will analyze the causes and offer practical insights.

3.2 Static Conflicts in Cor-RAIT

During the Cor-RAIT process, LLMs learn to reject unknown samples by supervisions from D_{idk}. Our insight is that if the Cor-RAIT dataset contains vanilla and IdK samples that are closely positioned in the LLM's representation space, the trained LLM may mistakenly classify similar vanilla samples as IdK samples, causing over-refusals. To verify this, we analyzed the sample distributions of D_{van} and D_{idk}. We extract latent representation of each question from the last hidden layer of the LLM. Then, t-SNE is adopted to visualize sample representations. Figure 3(a) displays the distributions of samples from the test split of MMLU dataset (Hendrycks et al. 2020) in the LLaMA-3-8B-instruct (Dubey et al. 2024) representation space, where IdK and vanilla samples have significant intersections. [1]

Furthermore, we introduce the **C**onflict **R**ate for **S**imilar **S**amples (CRSS) to quantitatively assess conflicts in supervision signals among similar samples in the RAIT dataset. For each sample x_i in D_{idk}, we compute the cosine similarity between its question representation r_i and the question representation r_j of each sample x_j in D_{van}. We identify and record the highest similarity value obtained. If this value exceeds the predefined similarity threshold τ_{sim}, we record the occurrence. The CRSS is then calculated as:

$$\text{CRSS} = \frac{\sum_{x_i \in D_{\text{idk}}} \mathbf{1}\left(\max_{x_j \in D_{\text{van}}} \cos(r_i, r_j) > \tau_{\text{sim}}\right)}{|D_{\text{idk}}|}$$

Therefore, the higher CRSS indicates more conflicting similar sample pairs, potentially leading to over-refusal. We computed the CRSS for Cor-RAIT, as shown in Figure 5. The results show that at $\tau_{\text{sim}} = 0.97$, CRSS reaches significant levels across various LLM and dataset combinations, supporting earlier t-SNE findings[2].

The above analysis reveals that Cor-RAIT generates numerous similar sample pairs between D_{van} and D_{idk}, resulting in conflicting supervision signals which leads to over-refusal. We term this **static conflict** to distinguish it from another conflict type discussed later.

3.3 Certainty Mitigates the Static Conflicts

We conducted a theoretical analysis [3] establishing a weak (non-differentiable) link between the LLM's feature and the response correctness μ for the specific question q. This weak correlation causes highly similar samples being categorized into D_{van} and D_{idk} respectively. To mitigate this, we propose incorporating a robust indicator variable aligned with *correctness* to select and construct the RAIT data. This variable should ensure that similar samples share comparable values,

[1] More experiments involving additional datasets and LLMs are provided in Appendix A.1.

[2] We show more results and details in Appendix A.4.

[3] detailed proof in Appendix A.2.

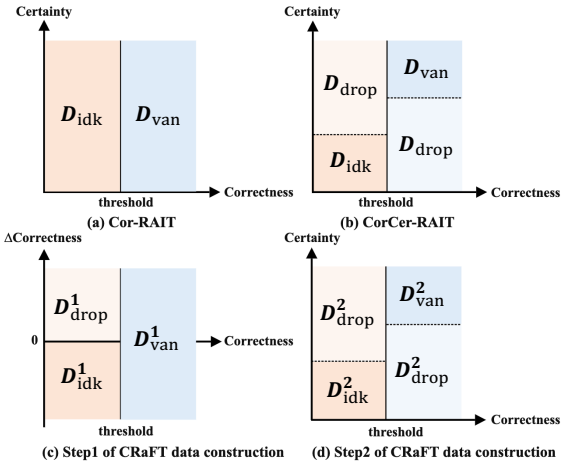

Figure 4: RAIT Data Construction of Cor-RAIT, CorCer-RAIT and CRaFT. Cor-RAIT partitions data based on accuracy μ and a threshold τ_μ. For CorCer-RAIT, D_{van} is derived from samples with accuracy exceeding the threshold and the highest certainty, while D_{idk} consists of samples with accuracy below the threshold and the lowest certainty. CRaFT employs a two-stage process: in the first stage, data with $\Delta\mu > 0$ is excluded through knowledge queries; the second stage follows the same procedure as CorCer-RAIT.

reducing the above misclassification. We suggest adopting the *certainty* (Jiang et al. 2023) of the LLM's response as the indicator variable. Our theoretical analysis shows that certainty meets the above requirements.[4]

To incorporate correctness and certainty into the RAIT data selection, we developed the CorCer-RAIT framework, as shown in Figure 4(b). We visualized the sample distribution in D_{van} and D_{idk} using t-SNE in the LLM representation space, as shown in Figure 3(b), which shows a significant decrease in the overlap between D_{van} and D_{idk} compared to the Cor-RAIT in Figure 3(a). Furthermore, we calculated the CRSS for both methods, as shown in Figure 5, highlighting substantial reductions in CorCer-RAIT over Cor-RAIT. Therefore, the joint use of correctness and certainty effectively alleviates the *static* conflict between the supervision signals in D_{van} and D_{idk}.

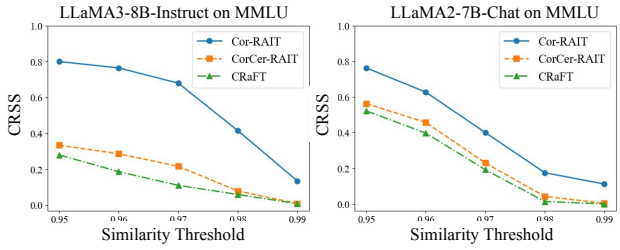

Figure 5: CRSS of Different RAIT Samples.

[4]Section 4.2 discusses various methods for representing LLM response certainty. Appendix A.3 uses entropy as a measure, but our findings extend to other methods as well.

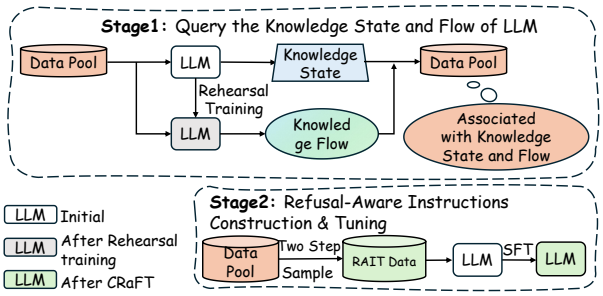

Figure 6: The Framework of CRaFT: Stage 1 queries knowledge state and flow, while Stage 2 constructs RAI data and tunes.

3.4 Knowledge Flow and Dynamic Conflict

Research (Ren et al. 2024; Gekhman et al. 2024; Ren and Sutherland 2024) reveals that the knowledge state of LLMs evolves during the SFT process. The phenomenon, which we refer to as "knowledge flow", can cause previously incorrectly answered questions to become correct ones during SFT. Despite this dynamic evolutions, the target answer of the training data remains static during the RAIt, which reflects the knowledge state of the initial LLM but ignores subsequent changes. We term it as **dynamic conflict**, which significantly contributes to the over-refusal in Cor-RAIT.

We select the data with the highest correctness and certainty for SFT, a process we refer to as *rehearsal training*[5]. *Rehearsal training* is designed to capture the LLM's natural knowledge flow during SFT. Experiments on the MMLU dataset (Hendrycks et al. 2020) and LLaMA3-8B-Instruct demonstrated that the correctness of **69%** of samples, initially below 0.5, improved, thereby validating the aforementioned analysis on dynamic conflict. Additional experimental results are provided in Appendix A.5.

4 Methodology

4.1 Overview

Based on Section 3, we propose the Certainty Represented Knowledge Flow for Refusal-Aware Instructions Construction (CRaFT) to solve the over-refusal problem. CRaFT contains two stages, as shown in Figure 6.

Stage 1: Query the Knowledge State and Flow of LLM The output of stage one is the knowledge state and flow indicators of the model. First, we perform a knowledge state query to obtain the correctness and certainty of the model's responses to the samples in the source dataset. Next, we conduct rehearsal training on the model, resulting in the perturbed version. By comparing the knowledge states before and after perturbation, we derive the indicators of knowledge flow during the supervised fine-tuning process.

Stage 2: Refusal-Aware Instructions Construction and Tuning Using the knowledge state and flow from Stage 1, we select suitable samples from D_{src} to construct the RAIT data, which is is used to fine-tune the initial model.

[5]Details are provided in Appendix B.2.

4.2 Query the Knowledge State and Flow of LLM

Knowledge State Query The input for knowledge state query consists of the LLM ($\mathcal{M}_0$ or $\widetilde{\mathcal{M}}$) and D_{src}. The output is the LLM's correctness and certainty for each sample in D_{src}, represented as $\{\mu_0 = Cor(\mathcal{M}, x_i), \sigma_0 = Cer(\mathcal{M}, x_i) | x_i \in D_{\text{src}}\}$, which indicate the LLM's knowledge state. Our research focuses on the Multiple-Choice Question Answering (MCQA) and Open-ended Questions Answer (OEQA) tasks, which correspond to different methods of knowledge state query.

In the MCQA task, for a given model $\mathcal{M}$ and question q, the possible answers are included in $O = \{A, B, C, D\}$. We obtain the token probability of $\hat{a}$, denoted as $p(\hat{a}|q, \mathcal{M})$, where $\hat{a} \in O$. We use the probability of the target answer token to represent correctness. Certainty is calculated through negative entropy. The corresponding formulas are:

$$Cor(\mathcal{M}, x_i) = p(x_i.a | x_i.q, \mathcal{M}),$$
$$Cer(\mathcal{M}, x_i) = -\sum_{\hat{a} \in O} p(\hat{a}|x_i.q, \mathcal{M}) \log(p(\hat{a}|x_i.q, \mathcal{M}))$$

In the OEQA task, following (Yang et al. 2023; Cheng et al. 2024), given a sample x_i, the LLM $\mathcal{M}$ performs inference on $x_i.q$ and generates responses N times (with $N = 10$). The generated responses $\{\hat{a}_0, \ldots, \hat{a}_{N-1}\}$ are denoted as $\hat{A}_i$. The generation process is carried out with a temperature of 1.0 and sampling enabled (`do_sample=True`).

$$Cor(\mathcal{M}, x_i) = \frac{1}{N} \sum_{\hat{a}_j \in \hat{A}_i} \mathbf{1}(\hat{a}_j = x_i.a)$$
$$Cer(\mathcal{M}, x_i) = \frac{1}{N(N-1)} \sum_{\hat{a}_j, \hat{a}_k \in \hat{A}_i, j \neq k} \cos(E(\hat{a}_j), E(\hat{a}_k))$$

Correctness is obtained through exact match across the N responses, calculating the proportion of accurate answers. Certainty is evaluated using a pretrained SentenceTransformer model [6] to encode each response $\hat{a}_j$ into embedding $E(\hat{a}_j)$, and the average similarity is computed between these embeddings (excluding diagonal elements). The correctness values range from $[0, 1]$. In MCQA task, certainty ranges from $[-log|O|, 0]$, and for OEQA, from $[0, 1]$. More details about knowledge state query are in Appendix B.1.

Rehearsal Training and Knowledge Flow During rehearsal training, we select high-certainty and high-correctness samples from D_{src} to fine-tuning $\mathcal{M}_0$. $\widetilde{\mathcal{M}}$ is obtained after fine-tuning. In the same way, we assess the perturbed LLM's knowledge state by performing another knowledge state query, yielding correctness and certainty for each QA pair in D_{src}: $\{\tilde{\mu} = Cor(\widetilde{\mathcal{M}}, x_i), \tilde{\sigma} = Cer(\widetilde{\mathcal{M}}, x_i) | x_i \in D_{\text{src}}\}$. The knowledge flow from the original $\mathcal{M}_0$ to the perturbed $\widetilde{\mathcal{M}}$ is quantified as:

$$\Delta\mu = Cor(\widetilde{\mathcal{M}}) - Cer(\mathcal{M}_0)$$

$$\Delta\sigma = Cor(\widetilde{\mathcal{M}}) - Cer(\mathcal{M}_0)$$

Rehearsal training sample selection prioritizes those with the highest correctness and certainty. This insight is supported by (Ren et al. 2024; Kang et al. 2024; Gekhman et al. 2024), which indicates that LLMs primarily refine and activate existing knowledge rather than acquire new knowledge during SFT. We align the rehearsal training with the LLM's internal knowledge state, ensuring a more natural and effective knowledge flow during the SFT process.

[6] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Algorithm 1: RAIT Data Construction Process

Input: $D_{\text{src}} = \{x_0, x_1, ..., x_N\}$, τ_μ, N_{van}, N_{idk}
Output: $D_{rait} \subseteq D_{\text{src}}$

$D^1_{\text{van}} = \{x_i \mid x_i \in D_{\text{src}}, x_i.\mu \geq \tau_\mu\}$
$D^1_{\text{idk}} = \{x_j \mid x_j \in D_{\text{src}}, x_j.\mu < \tau_\mu \text{ and } x_j.\Delta\mu < 0\}$
$D^1_{\text{van}} = \text{sort}(D^1_{\text{van}}, \text{key} = \sigma, \text{order=descend})$
$D^1_{\text{idk}} = \text{sort}(D^1_{\text{idk}}, \text{key} = \sigma, \text{order=ascend})$
$D^2_{\text{van}} = \text{TopK}(D^1_{\text{van}}, N_{\text{van}})$
$D^2_{\text{idk}} = \text{TopK}(D^1_{\text{idk}}, N_{\text{idk}})$
for x_i in D^2_{van} **do**
    $x_i.a_{\text{rait}} = x_i.a$
end for
for x_j in D^2_{idk} **do**
    $x_j.a_{\text{rait}}$ = "I don't know"
end for
$D_{\text{rait}} = D^2_{\text{van}} \cup D^2_{\text{idk}}$
return D_{rait}

4.3 Refusal-Aware Instructions Constuction and Tuning

Unlike Cor-RAIT, which selects RAIT samples solely based on correctness, our approach leverages four parameters μ, σ, $\Delta\mu$, and $\Delta\sigma$ to characterize both the knowledge state and flow of $\mathcal{M}_0$. The challenge lies in making informed sample selections across these four dimensions. We propose a two-step heuristic method outlined in Algorithm 1.

Step 1 As shown in Figure 4(c), we first filter the training sample D_{src} on the μ and $\Delta\mu$ plane. Setting a correctness threshold τ_μ, we define the vanilla candidate set $D^1_{\text{van}} = \{x_i | x_i.\mu \geq \tau_\mu\}$. For IdK candidates, unlike Cor-RAIT, we select $D^1_{\text{idk}} = \{x_j | x_j.\mu < \tau_\mu \; and \; x_j.\Delta\mu < 0\}$. Samples in $D^1_{\text{drop}} = \{x_k | x_k.\mu < \tau_\mu \; and \; x_k.\Delta\mu \geq 0\}$ are discarded because their correctness is actively increasing during SFT, shifting from unknown to known, which could lead to dynamic conflicts.

Step 2 As shown in Figure 4(d), we sort both D^1_{van} and D^1_{idk} by certainty σ. From D^1_{van}, we select the top N_{van} samples as final vanilla samples D^2_{van}, and the bottom N_{idk} samples as IdK candidates of D^2_{idk}, whose answers are then modified to "I don't know". The samples in D^2_{drop} are discarded. The final RAIT data $D_{\text{rait}} = D^2_{\text{van}} \cup D^2_{\text{idk}}$.

5 Experimental Setup

5.1 Dataset

We evaluate two tasks: knowledge-oriented Multiple Choice Questions Answering (MCQA) and Open-ended Questions Answering (OEQA). For MCQA, the MMLU (Hendrycks et al. 2020) test split serves as the training set, MMLU val as the In-Domain (ID) test set, and ARC-c (Clark et al. 2018)

test split as the Out-Of-Domain (OOD) test set. For OEQA, the TriviaQA (Joshi et al. 2017) train split is used for training, TriviaQA dev for the ID test set, and NQ (Kwiatkowski et al. 2019) dev for the OOD test set. More details are in Appendix D.1.

5.2 Metric

In post RAIT evaluation of LLMs, each test sample is classified as correct, incorrect, or refused. We calculate accuracy (P_c), error (P_w), and refusal rates (P_r) to assess performance, highlighting the key question: How to identify the better-performing model?

Shortcomings of existing refusal-aware metrics We conducted the in-depth analysis of existing refusal-aware metrics, identifying several design shortcomings (see Appendix C.1). We highlighted these shortcomings through constructed examples, as shown in Table 1.

Metric	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$
$P_c \uparrow$	0.3	0.3	0.5	1
$P_w \downarrow$	0.2	0.15	0	0
P_r	0.5	0.55	0.5	0
S_{honesty} (Yang et al. 2023) ↑	(0.8)	(0.794)	(1)	(1)
TRUTHFUL (Cheng et al. 2024) ↑	(0.8)	(0.75)	1	1
rely (Xu et al. 2024a) ↑	(0.55)	(0.548)	0.75	1
R-Acc (Feng et al. 2024) ↑	(0.8)	(0.778)	1	1
ER (Feng et al. 2024) ↑	(0.3)	(0.25)	0.5	1
A-Acc (Feng et al. 2024) ↑	(0.8)	(0.75)	1	1
A-F1 (Feng et al. 2024) ↑	(0.8)	(0.762)	1	1
AP (Zhang et al. 2024) ↑	—	—	(1)	(1)
THS (ours) ↑	0.1	0.15	0.5	1

Table 1: Comparison of refusal-aware metrics: The performance of constructed LLMs is $\mathcal{M}_1 < \mathcal{M}_2 < \mathcal{M}_3 < \mathcal{M}_4$. However, existing metrics exhibit significant issues, as indicated by the numbers in (parentheses).

We constructed an initial model $\mathcal{M}_0$ and four refined models $\mathcal{M}_1$ to $\mathcal{M}_4$, showing progressive improvement: $\mathcal{M}_1 < \mathcal{M}_2 < \mathcal{M}_3 < \mathcal{M}_4$. Details on these models are in Appendix C.2. However, existing metrics have notable flaws: $S_{\textbf{honesty}}$ (Yang et al. 2023) ranks $\mathcal{M}_1$ higher than $\mathcal{M}_2$ and treats $\mathcal{M}_3$ the same as $\mathcal{M}_4$; **TRUTHFUL** (Cheng et al. 2024) favors $\mathcal{M}_1$ over $\mathcal{M}_2$; and **R-Acc**, **ER**, **A-Acc**, and **A-F1** (Feng et al. 2024) also rank $\mathcal{M}_1$ higher than $\mathcal{M}_2$. Additionally, **AP** (Zhang et al. 2024) fails to distinguish between $\mathcal{M}_3$ and $\mathcal{M}_4$.

Our Metric: Truthful Helpfulness Score (THS) Due to the shortcomings of existing metrics, we propose the Truthful Helpfulness Score (THS). We first establish a Cartesian coordinate system with P_c and P_w as axes, where point E_1 represents the coordinates of the initial LLM, and point E_2 represents the coordinates of the refined. When E_2 falls below OE_1, a larger area of triangle $\triangle OE_1E_2$ indicates a stronger model. If E_2 is above OE_1, it suggests a decline in the model's performance. Based on this, we define THS as the ratio of the cross product of OE_1 and OE_2 to the maximum cross product value:

$$\text{THS} = (\overrightarrow{OE_2} \times \overrightarrow{OE_1}) / (\overrightarrow{OA} \times \overrightarrow{OE_1})$$

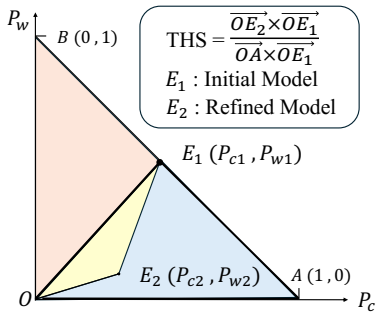

Figure 7: Truthful Helpfulness Score (THS).

The results in Table 1 clearly demonstrate the effectiveness of THS. For a more detailed analysis of THS's effectiveness, please refer to Appendix C.3.

5.3 Baselines

To verify CRaFT's effectiveness, we compared it with mainstream methods: **Init-Basic**: Uses the initial LLM with common question-answering prompts. **Init-Refuse**: Adds instructions like "If you don't know, respond with 'I don't know.'". **Van-Tuning**: Randomly selects $N_{\text{van}} + N_{\text{idk}}$ samples from D_{src} for instruct-tuning without modification. **Cor-RAIT**: Implements the method from (Zhang et al. 2024), filtering and modifying RAIT data based on response correctness. Detailed prompts for each baseline are in Appendix D.2.

5.4 Implementation Details

In the experiments, we used LLaMA2-7B-Chat (Touvron et al. 2023) and LLaMA3-8B-Instruct (Dubey et al. 2024) as the initial LLM $\mathcal{M}_0$. For the MCQA task, we selected 5000 samples from MMLU, and for the OEQA task, we used 10,000 samples from TriviaQA as training data. In all RAIT settings, except Van-Tuning, the ratio of vanilla to IdK samples was 1:4. We applied 5-shot and 3-shot knowledge state queries for the MCQA and OEQA tasks, respectively. Details on knowledge state and flow queries are in Appendix B. For Instruct Tuning, we used XTuner [7] with 3 epochs and a maximum context length of 2048. In MCQA, we applied LoRA (Hu et al. 2021) with settings $r = 64$, $\alpha = 16$, dropout=0.1, and a learning rate of 2e-4; for OEQA, full parameter training was used. More details on training are in Appendix D.3. We used 0-shot and greedy decoding for evaluation, with further details in Appendix D.4. OpenCompass [8] was employed for knowledge state queries and evaluations. All experiments were run on NVIDIA A100-80GB GPUs.

6 Experimental Results and Analyses

6.1 Overall Performance

The experimental results on the OEQA and MCQA tasks are presented in Table 2. Under the ID setting for both

[7] https://github.com/InternLM/xtuner

[8] https://github.com/open-compass/opencompass

LLMs	QA Type		MCQA						OEQA					
	Dataset		MMLU (ID)			ARC-c (OOD)			TriviaQA (ID)			NQ (OOD)		
	Metric		P_c	$P_w \downarrow$	THS↑	P_c	$P_w \downarrow$	THS↑	P_c	$P_w \downarrow$	THS↑	P_c	$P_w \downarrow$	THS↑
LLaMA2-7B Chat	Baselines	Init-Basic	45.6	52.8	00.0	53.9	46.0	00.0	54.0	46.0	00.0	29.3	70.7	00.0
		Init-Refuse	36.4	38.9	03.9	44.4	35.7	02.6	37.0	21.7	11.5	20.8	38.6	04.8
		Van-Tuning	46.9	53.1	01.2	54.5	45.5	01.2	48.6	44.5	-03.7	18.3	50.2	-02.5
		Cor-RAIT	44.5	39.6	11.3	55.8	38.1	11.1	41.3	18.3	19.7	16.2	27.6	04.7
	Ours	CRaFT	43.9	36.4	12.5	54.7	35.9	12.6	38.5	**12.9**	**23.3**	15.8	22.4	**06.5**
	Ablations	w/o Flow	39.7	**31.0**	**13.0**	51.4	**32.3**	13.5	45.2	20.5	21.1	21.2	38.8	05.2
		w/o Cer	38.4	32.1	11.5	52.5	32.9	**13.9**	38.5	15.7	20.1	14.6	**22.1**	05.4
LLaMA3-8B Instruct	Baselines	Init-Basic	66.8	33.1	00.0	80.6	19.5	00.0	66.8	33.2	00.0	40.3	59.7	00.0
		Init-Refuse	50.0	17.0	15.6	65.3	14.4	05.6	53.9	20.8	12.0	31.1	38.6	05.0
		Van-Tuning	69.5	30.5	08.0	80.3	19.7	-01.3	55.0	38.1	-21.8	21.0	48.5	-11.7
		Cor-RAIT	63.9	21.6	20.4	79.4	16.2	12.2	45.4	13.2	18.8	17.2	25.6	-00.1
	Ours	CRaFT	53.3	**09.6**	**34.0**	74.1	**12.7**	**21.4**	43.5	**10.9**	**21.5**	**19.0**	27.5	**00.4**
	Ablations	w/o Flow	57.5	15.3	27.2	75.8	14.9	13.9	49.1	18.0	12.8	22.3	41.6	-05.8
		w/o Cer	62.1	18.4	25.0	78.2	17.3	06.5	43.0	11.2	20.5	15.8	**23.5**	-00.1

Table 2: Performance comparisons on MMLU, ARC-c, TriviaQA and NQ. The best performance is highlighted in **boldface**.

types of tasks, our method outperformed four baseline models on THS, achieving the best results. Specifically, under the ID setting for OEQA, compared to the current best RAIT baseline, CRaFT improved the THS on LLaMA2-7B-Chat and LLaMA3-8B-Instruct by 3.56 and 2.72, respectively. Similarly, under the ID setting for MCQA, CRaFT improved the THS by 1.14 and 13.57, respectively. This indicates that CRaFT can significantly improve the model's rejection capability under the ID setting. Under the OOD setting, CRaFT improved the THS on the MCQA task by 1.5 and 9.2, respectively, compared to Cor-RAIT. On the OEQA's LLaMA2-7B-Chat, it improved by 1.76 compared to the most competitive method, Init-Refuse. Overall, CRaFT demonstrated excellent competitiveness in model generalization. Furthermore, we found that on the MCQA task, compared to other baselines, Cor-RAIT showed significant improvements under both ID and OOD settings. However, on the OEQA task, Cor-RAIT performed worse than Init-Refuse under the OOD setting. This reveals the limitations of the instruction fine-tuning method. It's worth mentioning that Van-Tuning generally had a negative impact on the improvement of overall capability, implying that the instruction fine-tuning approach of forcing the model to answer can undermine the model's inherent rejection capability. Therefore, although CRaFT surpassed Cor-RAIT under all tasks and settings, the improvement was limited under the OOD setting for OEQA due to training paradigm.

6.2 Ablation Experiments

In order to resolve the static and dynamic conflicts that lead to over-refusal, we extend Cor-RAIT to construct RAIT data using the information of correctness, certainty, and knowledge flow. We conduct sufficient ablation experiments to deeply investigate the impact of the above three factors on RAIT data selection. Compared to Cor-RAIT, the method only introducing response certainty which named as "w/o Flow" achieved significant gains on the THS in the MCQA and OEQA tasks. This indicates that eliminating static conflicts can effectively mitigate the over-refusal of LLMs and this improvement is generalizable. "w/o Cer" only uses response correctness and knowledge flow. Similarly, experimental results show that introducing knowledge flow to filter dynamic conflicts can also maintain the factuality of the model while improving its rejection capability. Finally, CRaFT considers both static and dynamic conflicts, further enhancing performance improvement.

7 Conclusion

In this paper, we identify over-refusal in correctness-based RAIT methods, caused by static and dynamic conflicts in RAIT data. To address this, we propose CRaFT: it mitigates static conflicts by incorporating response certainty during data construction and overcomes dynamic conflicts through rehearsal training to capture knowledge flow trends in LLMs. Extensive experiments on MCQA and OEQA tasks show CRaFT outperforms existing baselines, validating its effectiveness. Future work includes enhancing CRaFT with RL-based strategies and adapting it for more complex tasks, such as reasoning and multi-turn dialogue.

Acknowledgments

This research was supported by Shanghai Artificial Intelligence Laboratory.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.;

Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bai, F.; Zhang, H.; Tao, T.; Wu, Z.; Wang, Y.; and Xu, B. 2023. PiCor: Multi-Task Deep Reinforcement Learning with Policy Correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6): 6728–6736.

Bai, F.; Zhao, R.; Zhang, H.; Cui, S.; Wen, Y.; Yang, Y.; Xu, B.; and Han, L. 2024. Efficient Preference-based Reinforcement Learning via Aligned Experience Estimation. *arXiv preprint arXiv:2405.18688*.

Cheng, Q.; Sun, T.; Liu, X.; Zhang, W.; Yin, Z.; Li, S.; Li, L.; Chen, K.; and Qiu, X. 2024. Can AI Assistants Know What They Don't Know? *arXiv preprint arXiv:2401.13275*.

Chuang, Y.-S.; Xie, Y.; Luo, H.; Kim, Y.; Glass, J.; and He, P. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Feng, S.; Shi, W.; Wang, Y.; Ding, W.; Balachandran, V.; and Tsvetkov, Y. 2024. Don't Hallucinate, Abstain: Identifying LLM Knowledge Gaps via Multi-LLM Collaboration. arXiv:2402.00367.

Gekhman, Z.; Yona, G.; Aharoni, R.; Eyal, M.; Feder, A.; Reichart, R.; and Herzig, J. 2024. Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? *arXiv preprint arXiv:2405.05904*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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

Jiang, X.; Zhuang, D.; Zhang, X.; Chen, H.; Luo, J.; and Gao, X. 2023. Uncertainty quantification via spatial-temporal tweedie model for zero-inflated and long-tail travel demand prediction. In *CIKM*, 3983–3987.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kang, K.; Wallace, E.; Tomlin, C.; Kumar, A.; and Levine, S. 2024. Unfamiliar finetuning examples control how language models hallucinate. *arXiv preprint arXiv:2403.05612*.

Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Li, K.; Patel, O.; Viégas, F.; Pfister, H.; and Wattenberg, M. 2024a. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Li, S.; Yang, C.; Wu, T.; Shi, C.; Zhang, Y.; Zhu, X.; Cheng, Z.; Cai, D.; Yu, M.; Liu, L.; et al. 2024b. A Survey on the Honesty of Large Language Models. *arXiv preprint arXiv:2409.18786*.

Neeman, E.; Aharoni, R.; Honovich, O.; Choshen, L.; Szpektor, I.; and Abend, O. 2022. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering. *arXiv preprint arXiv:2211.05655*.

Ren, M.; Cao, B.; Lin, H.; Cao, L.; Han, X.; Zeng, K.; Wan, G.; Cai, X.; and Sun, L. 2024. Learning or self-aligning? rethinking instruction fine-tuning. *arXiv preprint arXiv:2402.18243*.

Ren, Y.; and Sutherland, D. J. 2024. Learning Dynamics of LLM Finetuning. *arXiv preprint arXiv:2407.10490*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wan, F.; Huang, X.; Cui, L.; Quan, X.; Bi, W.; and Shi, S. 2024. Mitigating Hallucinations of Large Language Models via Knowledge Consistent Alignment. *arXiv preprint arXiv:2401.10768*.

Wen, B.; Yao, J.; Feng, S.; Xu, C.; Tsvetkov, Y.; Howe, B.; and Wang, L. L. 2024. Know Your Limits: A Survey of Abstention in Large Language Models. *arXiv preprint arXiv:2407.18418*.

Xu, H.; Zhu, Z.; Ma, D.; Zhang, S.; Fan, S.; Chen, L.; and Yu, K. 2024a. Rejection Improves Reliability: Training LLMs to Refuse Unknown Questions Using RL from Knowledge Feedback. *arXiv preprint arXiv:2403.18349*.

Xu, Y.; Zhang, R.; Jiang, X.; Feng, Y.; Xiao, Y.; Ma, X.; Zhu, R.; Chu, X.; Zhao, J.; and Wang, Y. 2024b. Parenting: Optimizing knowledge selection of retrieval-augmented language models with parameter decoupling and tailored tuning. *arXiv preprint arXiv:2410.10360*.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yang, Y.; Chern, E.; Qiu, X.; Neubig, G.; and Liu, P. 2023. Alignment for honesty. *arXiv preprint arXiv:2312.07000*.

Zhang, H.; Diao, S.; Lin, Y.; Fung, Y.; Lian, Q.; Wang, X.; Chen, Y.; Ji, H.; and Zhang, T. 2024. R-Tuning: Instructing Large Language Models to Say 'I Don't Know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 7106–7132.

Zhang, S.; Yu, T.; and Feng, Y. 2024. Truthx: Alleviating hallucinations by editing large language models in truthful space. *arXiv preprint arXiv:2402.17811*.