Drift: Enhancing LLM Faithfulness in Rationale Generation via Dual-Reward Probabilistic Inference

Jiazheng Li[1*] **Hanqi Yan**[1*] **Yulan He**[1,2]
[1]King's College London [2]The Alan Turing Institute
{jiazheng.li,hanqi.yan,yulan.he}@kcl.ac.uk

Abstract

As Large Language Models (LLMs) are increasingly applied to complex reasoning tasks, achieving both accurate task performance and faithful explanations becomes crucial. However, LLMs often generate unfaithful explanations, partly because they do not consistently adhere closely to the provided context. Existing approaches to this problem either rely on superficial calibration methods, such as decomposed Chain-of-Thought prompting, or require costly retraining to improve model faithfulness. In this work, we propose a probabilistic inference paradigm that leverages task-specific and lookahead rewards to ensure that LLM-generated rationales are more faithful to model decisions and align better with input context. These rewards are derived from a domain-specific proposal distribution, allowing for optimized sequential Monte Carlo approximations. Our evaluations across three different reasoning tasks show that this method, which allows for controllable generation during inference, improves both accuracy and faithfulness of LLMs. This method offers a promising path towards making LLMs more reliable for reasoning tasks without sacrificing performance.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across a wide range of challenging tasks, including Question Answering (QA) (Li et al., 2024b), reasoning (Yao et al., 2023; Yan et al., 2024), and providing feedback on essays or reviews (Liang et al., 2024; Li et al., 2023). While LLMs can be prompted to generate self-explanations for their decision (Kim et al., 2024; Madsen et al., 2024; Atanasova et al., 2023), ensuring the accuracy and fidelity of these rationales remains challenging. This is critical both for improving interpretability and for enhancing reliability in safety-critical fields (Lyu et al., 2024; Radhakrishnan et al., 2023).

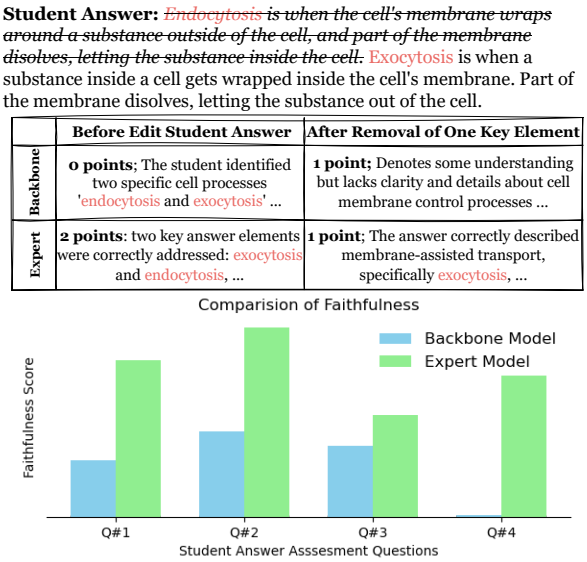

Figure 1: Comparison of rationale faithfulness between a LLaMA-3 model (Backbone) and the Expert (same LlaMA model trained on a dataset in biology exams). The Expert model is sensitive to the removal of a valid key element from the answer. In contrast, the Backbone model fails to reflect the occurrence of the important word *'exocytosis'*. Based on the perturbation-based evaluation, Expert model shows better faithfulness scores than Backbone across four datasets.

Enhancing the faithfulness of LLM-generated rationales is a multifaceted challenge. To date, there is no universally accepted or formal definition of faithfulness (Lyu et al., 2024). In this paper, we focus on a specific category of unfaithfulness, where *models fail to incorporate key contextual information into their generated rationales*. This issue is highlighted in faithfulness evaluation, where unfaithful models do not respond adequately to alterations in input (Lanham et al., 2023; Radhakrishnan et al., 2023). Figure 1 exemplifies this. When assessing a student's answer to a biology exam, an untrained LLaMA-3 Backbone model overlooks critical details and produces vague, unfaithful assessments if key information is removed.

[*]Both authors contributed equally and may be interchanged as appropriate.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6850–6866

Such base models often favor generic terms, leading to contextually poor rationales. Conversely, a domain-trained Expert model accurately identifies key scientific concepts in the student's answer and appropriately adjusts its evaluation upon removal of critical information.

The model's tendency to rely heavily on its pretrained distributions often stems from its lack of contextual sensitivity, causing it to overlook subtle differences across various domains and contexts (Hu et al., 2023). Even instruction-tuned LLMs, designed to enhance adaptability, struggle to adjust to new domains and generate contextually sensitive responses. For instance, 92.2% of tokens overlapped between the base and instruction-aligned LLMs across 1,000 examples (Lin et al., 2024). Moreover, evidence (Yuan et al., 2023; Yang et al., 2024) suggests that LLMs often generate inaccurate labels when applied to out-of-distribution scenarios. Such domain insensitivity compromises their effectiveness, especially when balancing the dual demands of accurate label prediction and faithful rationale generation, which remains a challenge (Radhakrishnan et al., 2023).

To address these limitations, we propose **Drift**, a **D**ual-**R**eward probabilistic **I**nference method for **F**aIthful ra**T**ionale generation. It incorporates a *task reward* to distill knowledge from a *supervised fine-tuned classifier* for more accurate label prediction. Recognizing that expert models trained on domain-specific corpora respond more effectively to domain-specific contexts, we utilize a *generative expert model* to provide a *rationale reward* during rationale generation. Specifically, the rationale reward encourages the generation of tokens that enhance the plausibility of future tokens, guided by the output distribution of the generative expert model. Notably, the generative expert models are not required to be trained on the exact inference dataset or share the same backbone model as the inference model, which ensures flexibility and generalisability. Our contribution is three-fold[1]:

1. We investigate the challenge of faithful rationale generation by highlighting the limitations of LLMs in responding to domain-specific context. To the best of our knowledge, this is the first study to enhance faithfulness by explicitly encouraging domain-relevant generation.
2. We propose a novel and efficient probabilistic inference framework that integrates both task and rationale rewards within a sequential Monte Carlo tree search process.
3. Empirical evaluations across three tasks and seven datasets show significant improvements in both accuracy and faithfulness. Ablation studies further highlight the synergistic benefits of **Drift** in leveraging the strengths of different expert models.

[1]Code is available at https://github.com/lijiazheng99/drift.

2 Related Work

Constrained generation Constrained generation can be achieved by training models with an attribute-conditioned discriminator (Yang and Klein, 2021), but more recent studies (Liu et al., 2024a,b,c) have shifted focus to constrained decoding to reduce the training cost for large language models. Some constraints are localized, applied step-by-step through simple logit arithmetic between two models to ensure certain attributes, such as harmlessness (Xu et al., 2024), toxicity avoidance, and truthfulness (Liu et al., 2024a). However, these localized constraints are limited in enforcing attributes that span across a larger text segment. Monte Carlo tree search (MCTS), by contrast, is characterized by its lookahead reward, enabling it to estimate future rewards. This makes it popular for identifying optimal trajectories in decoding (Liu et al., 2024b; Yan et al., 2024) or as training data (Snell et al., 2023; Hong et al., 2023). Our method is closely related to Fame (Hong et al., 2023), which employs a faithfulness-seeking reward in the Monte Carlo Search framework. Unlike Fame, our approach avoids training and incorporate the domain-specific rewards into a probabilistic monte carlo inference.

Rationale faithfulness Although LLMs can provide plausibly sounding explanations for their answers, recent work argues that model generated natural language explanations are often unfaithful (Lanham et al., 2023; Atanasova et al., 2023). Faithfulness evaluation for rationale is to apply important perturbation to the original rationale and check the changes in the new output (Parcalabescu and Frank, 2023). Such perturbation includes counterfactual edit (Atanasova et al., 2023), biased feature (Turpin et al., 2023) and corrupted Chain-of-Thought (Lanham et al., 2023). To increase the faithfulness of LLM-generated response, many existing methods focus on the Chain-of-Thought and decompose the reasoning process into multiple sub-sentences (Radhakrishnan et al., 2023), then verify them using external tool, e.g., python interpreter (Lyu et al., 2023), counterfactual (Gat et al.,

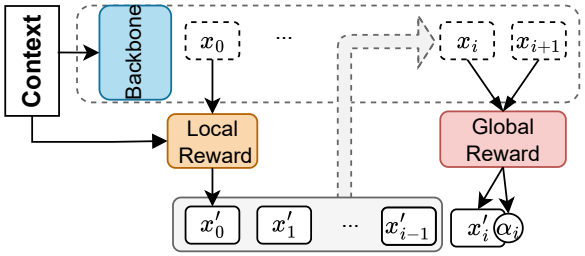

Figure 2: Our method incorporates both task rewards and look-ahead rationale rewards to ensure accurate label prediction and faithful rationale generation at each step. The original token generated by the backbone model, denoted as $\boldsymbol{x_i}$, is refined by either the task or rationale reward model, resulting in an updated token $\boldsymbol{x'_i}$. The refined token sequence $[\boldsymbol{x'_0} : \boldsymbol{x'_i}]$ serves as the conditional context for generating the next token $\boldsymbol{x_{i+1}}$. The rationale reward also assigns a score α_i to each token, which is used to select the final generation trajectory from a beam size of K.

2023). The above methods alleviate the unfaithful issue either in a post-hoc manner or via costly training. We instead propose a inference-time method, which can improve both faithful and accurate for different reasoning tasks, also maintain a similar computation cost as beam search.

3 Probabilistic Inference for Faithful Rationale Generation

In this section, we introduce our probabilistic inference framework, **Drift**, depicted in Figure 2 and detailed in Algorithm 1. The framework incorporates two types of rewards: a task reward for more accurate label prediction and a rationale reward for faithful rational generation.

3.1 Problem Setup

We represent the constrained decoding process of model f as a Markov Decision Process, $\langle S, \mathcal{V}, \pi, Q \rangle$. The state space S consists of multi-token sequences drawn from the vocabulary $\mathcal{V}$. The transition function $\pi_t(x_{t+1} \mid x_t) \in \Delta^{|\mathcal{V}|}$ outputs a probability distribution over $\mathcal{V}$. The reward function Q guides the search process to ensure accuracy and faithfulness.

A probabilistic inference framework based on Feynman-Kac model Feynman-Kac formulae (Del Moral and Del Moral, 2004) is designed to facilitate probabilistic *sequential Monte Carlo approximation* (Lew et al., 2023), which involves a tuple consisting of an initial state, a transition distribution, and a potential function, denoted as (s_0, π_t, G_t). The potential function G_t maps pairs of states (s_t, s_{t+1}) to a non-negative score, i.e., $G_t : (s_t, s_{t+1}) \to \mathbb{R}_{\geq 0}$. It is originally designed to compute a probabilistic density of sampled states s_t, thereby approximating a target G_t using the equation:

$$\mathbb{P}_t(s_t) = \frac{\mathbb{E}_\pi \left[\prod_{i=1}^{t \wedge T} G_i(S_{i-1}, S_i, f) \cdot \mathbf{1}_{[S_t = s_t]} \right]}{\mathbb{E}_\pi \left[\prod_{i=1}^{t \wedge T} G_i(S_{i-1}, S_i, f) \right]},$$

where $\mathbf{1}_{[S_t = s_t]}$ is an indicator function that equals 1 if the state at time t is s_t, and 0 otherwise. *The numerator inside the expectation represents the product of rewards and the probability of reaching state s_t, ensuring that paths leading to high values of G_t over time receive more weights.* Generation continues until a terminal token is reached or the sequence length reaches its maximum T, i.e., $t \wedge T = \min(t, T)$.

Advantages of the probabilistic inference framework The probabilistic inference framework can be adapted to various tasks by designing different potential functions G_t, such as prompt intersection (Lew et al., 2023) and hypothesis revision (Piriyakulkij et al., 2024). In essence, G_t functions can be instantiated as the reward function Q_t in MCTS-like algorithms for guided search, but it offers two key advantages for our approach:

(i) Unlike MCTS, which requires expensive rollouts or simulations to evaluate potential actions (Xie et al., 2024; Zhang et al., 2024a; openai, 2024), the Feynman-Kac model focuses on high-probability paths. This reduces unnecessary computations and enables more efficient exploration.

(ii) While MCTS requires manually tuned coefficients to balance exploration and exploitation, the Feynman-Kac model inherently integrates uncertainty into the search process, allowing for more adaptive and dynamic decision-making.

3.2 Drift Framework

We describe how to incorporate task and rationale rewards into the probabilistic framework. In our setup, the generated sequence for reasoning tasks consists of an answer followed by an explanation [2]. The **Drift** framework, illustrated in Figure 2, incorporates two types of rewards. The *task reward* comes from a smaller fine-tuned classifier for label

[2]To prevent scenarios where an overly long rationale causes the answer to exceed the output length limit, we prioritize generating the answer first.

Algorithm 1 *Dual-Reward Probabilistic Inference*

Input: Backbone model f, fine-tuned classifier prediction c_0, rationale expert model Q^g; state transition distribution π_t, beam size K, max length T, label set $\mathcal{C}$, vocabulary $\mathcal{V}$, terminal token $|eos|$

Output: Selected token sequence $x_{1:T}^{k^*}$

Initialization: initialize weighted input sequence $\{(x_t, \alpha_t) \leftarrow (s_0, 1)\}_{t=1}^{T}$.
$t \leftarrow 0$
while $t < T$ and $x_t \neq |eos|$ **do**
  if $t == 0$ **then** ▷ First position for label generation
    $x_{t+1}^k \leftarrow$ *TaskReward*$(\pi_t, x_t, f, \mathcal{C}, c_0), k \in [1, K]$ ▷ Derived refined label from *TaskReward*
  else ▷ Other positions for rationale generation
    $x_{t+1}^k \sim \pi_t(\cdot|x_t^k, f), k \in [1, K]$ ▷ Backbone model uses Beam search decoding
  end if
  $\alpha_{t+1}^k \leftarrow$ *RationaleReward*$(\pi_t, x_{1:t+1}^k, f, Q^g), k \in [1, K]$ ▷ Generated rationales are reweighted by expert model
  $t \leftarrow t + 1$
end while
$k^* \leftarrow \arg\max_k \left(\sum_{i=1}^{t} \alpha_i^k / t\right)$ ▷ Select sequence with maximal average weight
return $x_{1:t}^{k^*}$

Function *TaskReward*$(\pi_t, x_t, f, \mathcal{C}, c_0)$
  $\mathcal{V}' \leftarrow \mathcal{V} \setminus (\mathcal{C} \setminus \{c_0\})$ ▷ Update vocabulary: remove labels in $\mathcal{C}$ except c_0
  Let $P(v'|x_t) = \pi_t(x_{t+1} = v' \mid x_t, f)$.
  Define $\pi_t'(x_{t+1} = v' \mid x_t) \leftarrow \frac{P(v'|x_t)}{\sum_{v'' \in \mathcal{V}'} P(v''|x_t)}$ for $v' \in \mathcal{V}'$; else 0. ▷ Filter & renormalize π_t over $\mathcal{V}'$
  $v \sim \pi_t'(\cdot|x_t)$ ▷ Sample next token from the constrained distribution
  return v

Function *RationaleReward*$(\pi_t, x_{1:t+1}, f, Q^g)$
  $\alpha_{t+1} \leftarrow Q^g(x_{1:t+1})$ ▷ Score candidate sequence $x_{1:t+1}$ using Q^g
  return α_{t+1}

prediction at the first generation step. The *rationale reward*, provided by a domain-specific generative expert model, is applied to adjust the stepwise generation by generating a new token x_i' associated with higher Q_t and a corresponding weight w_i. The final trajectory is selected based on the average sequence-level weight. Details can be found in Algorithm 1: Dual-Reward Probabilistic Inference, with functions *TaskReward* and *RationaleReward*.

During model inference, the framework incorporates two reward signals. The task reward, active at the first generation step (for label prediction), constrains the sampling distribution to align with the classifier's output c_0. The rationale reward, applied at each step, is calculated by a generative expert model g as a weight α_t for the current token x_t based on its fit with the preceding context and alignment with anticipated expert predictions.

Task reward A heuristic and lightweight approach for constrained generation from LLMs is to use masking or logit bias to reweigh the probabilities of sampled tokens, i.e., π_t. Many methods (Liu et al., 2024a; Zhao et al., 2024) leverage generation logits from smaller models to calibrate the logits from larger model, e.g., logit fusion, in order to alleviate undesirable attributes such as toxicity and untruthfulness. However, these attributes are implicitly conveyed over longer spans rather than individual tokens, making token-level constraints insufficient (See the performance of logit fusion in Table 4). Therefore, we don't adopt such local constraint for faithfulness enhancement. Instead, fine-tuned classifier trained on knowledge-specific corpus generally demonstrate better accuracy than general LLM (Yuan et al., 2023; Yang et al., 2024). Therefore, we adopt a pretrained classifier to enhance task-specific label prediction. Specifically, for the generation step corresponding to the task label, we define a set of classification label words $\mathcal{C}$. We then modify the vocabulary to $\mathcal{V}'$ by removing all label words from $\mathcal{C}$ except for the expert classifier's prediction c_0. The output probabilities from the base model f (denoted π_t) are then renormalized over this updated vocabulary set $\mathcal{V}'$, effectively ensuring the sampled label is a valid token, or another non-label token. (Details in the function *TaskReward*). This step directly steers the answer generation towards the expert classifier's choice.

Rationale reward As discussed earlier, the task reward is designed for label accuracy, regardless of the quality of generated rationales; therefore, we require the use of rationale rewards. Similar to the rollout phase in MCTS, we generate multiple promising trajectories and select the optimal one based on the overall rewards. At each step, the exploited reward is determined by the hind-

sight function Q_t^g, defined by the generative expert model. Specifically, the expert model evaluates the generated n-gram (x_t, x_{t+1}) from f (Details in the *RationaleReward* function). This approach encourages text spans that are faithful and coherent with the context, as they align more closely with the expert's domain-specific distribution. The impact of the rationale reward on enhancing faithfulness is summarized in Table 1.

4 Experiments

We evaluate **Drift** for both the task performance and faithfulness of the generated rationale across three reasoning problems, i.e., student answer assessment, natural language inference and question answer. We further ablate the task reward and rationale reward to verify their effectiveness.

4.1 Experimental Setup

We present the evaluated dataset, the expert models incorporated, and the experiment setup for faithful evaluation.

4.1.1 Datasets

We conduct experiments on three tasks: *student answer assessment* on the ASAP dataset [3], *natural Language Inference (NLI)*, including the Stanford Natural Language Inference (*SNLI*) (Bowman et al., 2015) and the Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018) datasets; and the *TruthfulQA* dataset (Lin et al., 2022). In each of these tasks, LLMs are required to generate both class labels and rationales justifying their classification decisions. For student answer assessment, the labels represent valid score ranges, 0-3; For NLI, the labels are '*entailment*', '*contradiction*', or '*neutral*'. For *TruthfulQA*, we use a subset of the dataset converted into a multiple-choice format. Our experiments are evaluated on 100 randomly sampled instances from each dataset's test set, with the model utilizing 8-bit quantization. We use the accuracy score to evaluate the task performance.

4.1.2 Backbone and Expert Models

Our study employs two widely used instruction-tuned LLMs as our backbone models: Llama-3-8B-Instruct (Dubey et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). For each dataset, we incorporate a classifier to provide the *Task reward* and a generative model as the expert model to offer the *rationale reward*. Details of our Backbone, task reward and rationale reward model choices are given in Table A2 in the Appendix. The parameter and inference setup details are elaborated in Appendix A.1. Note that all expert models are fine-tuned solely on the train sets of the evaluation datasets or on other unrelated datasets distinct from the evaluation test sets, demonstrating the generalizability of our framework. Our framework, **Drift**, *operates under a zero-shot setting with open-source reward models without any fine-tuning*.

[3] https://kaggle.com/competitions/asap-sas

4.1.3 Faithfulness Evaluation Setup

Perturbation for counterfactual generation Following existing approaches for counterfactual generation in faithfulness evaluation (Atanasova et al., 2023; Lanham et al., 2023), we modify key parts $\boldsymbol{w}$ of the inputs $\boldsymbol{I}$ and examine the resulting changes in the generated rationales. For *student answer assessment*, we remove the clause (sub-sentence) from the student answer that is most semantically related to the original rationale $\boldsymbol{R}_o$. In the case of *NLI* and *TruthfulQA*, where context sentences are typically short (often single sentences), we introduce perturbations through word insertion, as inspired by Atanasova et al. (2023). Specifically, we use Part-of-Speech (POS) tagging to identify verbs and adjectives in the context sentences, as these are likely to have a greater impact, and replace them with alternative words. We then feed the perturbed input to the model to generate a new rationale $\boldsymbol{R}_n$. Details of generating counterfactual rationales are provided in Appendix A.1.

Evaluation metrics For sub-sentence removal perturbations, we calculate the semantic relatedness between the removed text span $\boldsymbol{w}$ and the original rationale, denoted as $S_{wo} = \text{Sim}(\boldsymbol{w}, \boldsymbol{R}_o)$, and between the removed span and the new rationale, denoted as $S_{wn} = \text{Sim}(\boldsymbol{w}, \boldsymbol{R}_n)$. A faithful model is expected to produce a significant ***semantic variation***, calculated as $\Delta(S_{wo} - S_{wn})$, as the removed sub-sentence should be closely related to the original rationale but less similar to the new rationale. For *word insertion* perturbations, we calculate the percentage of new rationales that include the newly inserted word, denoted as ***word inclusion***. Both a large semantic variation and high word inclusion indicate greater rationale faithfulness.

4.2 Main Results

We compare the baselines in terms of task performance (Acc), faithfulness (Faith), and overall

	Backbone			Classifier	Generative Expert			Drift (full)		
Metrics	Acc	Faith	Overall	Acc	Acc	Faith	Overall	Acc	Faith	Overall
Student Answer Assessment (ASAP)										
Q1	28%	0.034	0.524	85%	76%*	0.094△	**1.509**	57%	0.052	0.965
Q2	28%	0.051	0.588	72%	48%	0.114△	**1.480**	68%*	0.050	0.977
Q3	45%	0.042	0.774	91%	71%	0.061△	1.302	90%*	0.058	**1.450**
Q4	38%	0.001	0.380	88%	67%	0.053	1.174	84%*	0.102△	**1.821**
NLI										
SNLI	49%	0.110	0.490	86%	76%*	0.130	0.942	69%	0.150△	**1.054**
MNLI	57%	0.090	0.737	88%	76%	0.090	0.927	77%*	0.190△	**1.770**
QA										
TruthQA	47%	0.020	0.470	100%	70%	0.240△	**1.700**	73%*	0.180	1.457
Overall Avg	42%	0.050	0.566	87%	69%	0.112△	1.290	74%*	0.112△	**1.356**

Table 1: **Evaluation results of task performance (Acc) and normalized rationale faithfulness scores (Faith), and overall evaluation across three different tasks with LLaMA3-8B.** Classifier and Generative Expert denote results for task reward model and rationale reward model, respectively. The best overall results are marked in **bold**, and best Acc and faithfulness are marked in * and $^\triangle$, respectively.

performance across both metrics. The full results across the three tasks are shown in Table 1 [4].

Task Performance The classifier achieves the best accuracy across all the datasets, making it a reliable source for label predictions in our **Drift** framework. Conversely, the backbone model exhibits the poorest task performance compared to both the expert and **Drift**. Despite not being specially trained for classification tasks, expert models, trained primarily for generative tasks, outperform backbone models. This demonstrates the importance of domain-specific training for task performance. For generalizability, all the expert models shown in this table are of equal or smaller size than the backbone model, where the LLaMA3-8B global expert is used for *ASAP*, while two different LLaMA2-7B global models are applied for the *NLI* and *QA* tasks. **Drift** outperforms the expert in accuracy on five out of seven datasets, with a larger margin in *ASAP* (except for Q1). This improvement could be attributed to the fine-tuned classifier's ability to provide plausible and accurate information early in the reasoning process.

Faithfulness evaluation The Faith columns in Table 1 present the *normalized* faithfulness scores of the generated rationales. The original faithfulness scores, i.e., semantic variation or word inclusion introduced in Section 4.1.3, varied significantly across tasks. Since the classifier does not generate rationales, it is excluded from this evaluation. Overall, **Drift** shows better faithfulness compared to the Backbone model, except for Q2, where it scores 0.050 vs 0.051. It also shows a clear advantage over the expert models on the NLI datasets. Although the expert is generally regarded as a source of faithfulness, the superior performance of **Drift** over the global expert (Llama2-7B) on NLI tasks may be contributed to the enhanced context learning capabilities of the larger backbone model.

Overall performance We average the accuracy and faithfulness scores for each method on all the datasets to see the overall performance. To standardize the faithfulness results across different tasks, we applied min-max normalization within each dataset's result[5]. Our experimental results reveal that **Drift** can improve both task performance (Acc) and faithfulness of the rationale generated by the Backbone model with the incorporation of both task and rationale rewards without being fine-tuned on in-domain datasets.

Effects with different backbone models Table 2 summarizes the results of using Mistral-7B as the backbone model. **Drift** outperforms the Backbone in both task performance and rationale faithfulness across all datasets. For Acc, **Drift** outperforms the Backbone model by large margins, particularly in *NLI*, where the accuracy on *SNLI* and *MNLI* is improved by 32% and 37%, respectively. Similarly, in *ASAP* and *QA*, accuracy improvements are notable, with gains as high as 49% on *Q2*. For rationale faithfulness, **Drift** also demonstrates sub-

[4] As the classifier can't generate rationales, faithfulness evaluation is not applicable to it.

[5] Min and max values we used are presented in Table A1.

Datasets	Backbone			Drift		
	Acc	Faith	Overall	Acc	Faith	Overall
Student Answer Assessment (ASAP)						
Q1	31%	0.005	0.310	80%*	0.042△	**1.111**
Q2	38%	0.023	0.380	69%*	0.080△	**1.316**
Q3	35%	0.028	0.477	84%*	0.034△	**1.051**
Q4	45%	0.005	0.489	80%*	0.008△	**0.868**
:Avg	37%	0.015	0.414	78%*	0.041△	**1.087**
NLI						
SNLI	47%	0.140	0.743	79%*	0.150△	**1.154**
MNLI	41%	0.070	0.410	78%*	0.160△	**1.530**
:Avg	44%	0.105	0.576	79%*	0.155△	**1.342**
QA						
TruthfulQA	43%	0.170	1.112	72%*	0.200↑	**1.538**

Table 2: Evaluation results of task performance (Acc) and normalized rationale faithfulness scores (Faith) across three different tasks using **Mistral 7B**. The best overall results are marked in **bold**, and best Acc and faithfulness are marked in ⋆ and △, respectively.

stantial gains, especially for *TruthfulQA* and *Q2* in *ASAP*, where an increase from 0.170 to 0.200 and a remarkable jump from 0.023 to 0.080 are observed. These results further confirm the effectiveness and generalizability of **Drift** in enhancing both accuracy and rationale faithfulness, regardless of the type and size of the backbone model.

4.3 Ablation Studies

We ablate the full **Drift** model into its task and rationale reward components to examine their individual effects, as shown in Fig. 3. For accuracy (Top), adding the rationale reward (w. Rationale) alone generally enhances task performance over the Backbone model on most tasks, or maintains comparable performance in others. In contrast, the task reward (w. Task) alone provides a notable improvement on the Student Answer Assessment task but is detrimental to performance on the NLI and QA tasks when compared to the Backbone. We attribute this due to the complexity of hard constraint task reward in a multiple token combined label space. The full **Drift** model, integrating both reward types, consistently achieves the highest accuracy across all three tasks shown. For faithfulness (Bottom), the rationale reward component (w. Global, depicted by the light blue/greenish bar) consistently improves faithfulness scores over the Backbone across all tasks. Notably, the full **Drift** model further enhances faithfulness beyond what the rationale reward achieves alone, obtaining the highest faithfulness scores in all evaluated tasks. This indicates that the combination of both task and rationale rewards within the **Drift** model is advantageous for improving both task performance and rationale faithfulness.

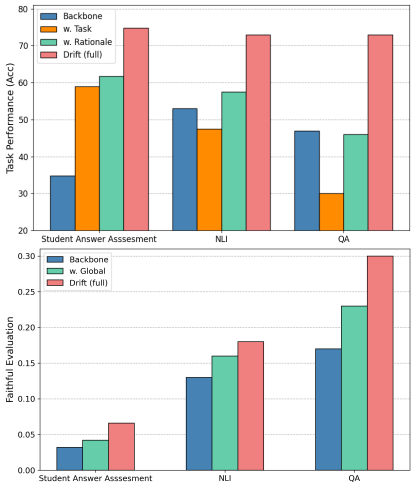

Figure 3: Ablation results for task performance (**Top**) and faithfulness (**Bottom**). We compare among Backbone, Backbone w/ Task Reward, Backbone w/ Rationale Reward and Drift (full).

5 Further Analysis

We assess the generalizability of **Drift** under different settings. We also examine the faithfulness source qualitatively from domain-specific word distributions and a case study.

5.1 Incorporating Weak Expert Model

Drift is compatible with rationale rewards derived from models of varying sizes. Details on incorporating global experts with different tokenization methods are elaborated in Appendix A.1. Here, we use an expert model based on Mistral-7B (weak expert), specifically trained for scientific QA, rather than fine-tuned on the ASAP dataset validation set (referred to as *out-of-task*). The performance of this expert is compared to that of a LLaMA3-8B-based expert model (strong expert), fine-tuned on the ASAP train set (referred to as *in-task*). The results are presented in Table 3. Despite a modest reduction in performance from the *out-of-task* expert compared to the specialized *in-task* expert, the incorporation of the *out-of-task* expert model consistently provides clear advantages over the backbone models across all subsets. These results are **crucial in demonstrating the generalizability of Drift, showcasing its ability to leverage weak supervision for faithfulness enhancement**. This challenges the common assumption in many existing constrained generation methods, where a specialized in-task trained expert model is typi-

cally required (Liu et al., 2024a; Hong et al., 2023; Anonymous, 2024).

Experts	Backbone		w. Weak Expert		w. Strong Expert	
Datasets	Acc	Faith	Acc	Faith	Acc	Faith
Q1	28%	0.034	59%	0.123	57%	0.052
Q2	28%	0.051	55%	0.064	68%	0.050
Q3	45%	0.042	67%	0.089	90%	0.058
Q4	38%	0.001	56%	0.104	84%	0.102
Avg	35%	0.032	59%	0.096$^{\triangle}$	75%*	0.066

Table 3: Performance of **Drift** when utilizing **different generative experts**, based on the LLaMA3-8B backbone, where the first two columns are copied from backbone performance for better comparison. The best Acc and faithfulness are marked in * and $^{\triangle}$, respectively.

5.2 Comparing with Local Logit Fusion

We also observed that contrastive decoding for constrained generation, achieved by injecting logits from an expert model, typically updates the logits based solely on the current token, without considering future tokens (Liu et al., 2024b,a; Anonymous, 2024). To highlight the advantages of the lookahead characteristics of our rationale reward, we replace the *RationaleReward* function mentioned in Algorithm 1 with a straightforward logit fusion, inspired by (Liu et al., 2024a). This method is compared with a decoding approach that fuses the token probabilities from the expert and backbone models at each timestep via interpolation. As shown in Table 4, rationales generated from logits fusion baseline are among 55% less faithful on average compared with ours (displayed in Table 1)[6], their task performance is even lower than some of the backbone results. **The theoretical advantage between logit fusion and our lookahead reward (rationale reward) is that our method *considers future tokens' plausibility when scoring the currently generated token***.

5.3 Domain-specific Word Distribution

We utilize TF-IDF to select domain-specific words (after removing the stopwords) from the student responses in the *student answer assessment* dataset. The selected words and their associated TF-IDF scores are depicted in the blue curve (context) in Figure 4. Since the TF-IDF score reflects the importance of these contextual words, we calculate the TF-IDF scores for the same words within the rationales generated by the backbone model (in orange) and our-full model (in green). This allows us to verify whether the generated rationales align well with the important spans in the context. It is clear that the green curve is mostly above the orange curve, showing that our method can respond more actively to those domain-specific words, such as "*experiment*", "*data*", "*replicate*", "*substances*", and "*nuclear*". Moreover, we calculate the semantics overlap using BLEU between the given context and generated rationale for a quantitative analysis. Specially, we calculate the BLEU between the generated rationale and the given prompt, including the question, student answer and instruction (Results in Table 5). This result further verify that the faithful source of **Drift** from the generation of domain-specific words.

[6]Note that more than 50% of the samples failed to generate responses due to the fused logit being impractical for the backbone model.

Datasets	Acc	Faith
Q1	34% (-40%)	0.028 (-46%)
Q2	22% (-68%)	0.037 (-26%)
Q3	40% (-56%)	0.019 (-67%)
Q4	29% (-65%)	0.019 (-81%)
Avg	31% (-57%)	0.026 (-55%)

Table 4: *logitfusion* results on both task performance and faithfulness for Student Answer Assessment. The relative changes compared with **Drift** (**Full**) are in brackets.

Method	1-gram	2-gram	3-gram	4-gram
Backbone	0.106	0.090	0.058	0.025
Drift	**0.452**	**0.333**	**0.167**	**0.058**

Table 5: Semantic relatedness between assessment prompt and generated rationale. Higher values imply higher faithfulness.

5.4 Case Studies

To highlight the differences between the rationales from the backbone model and ours, we randomly select two examples from *Student Answer Assessment* and *Natural Language Inference* datasets, as shown in case study 5.4. For this assessment task, four key elements related to protein synthesis are expected (e.g., *mRNA exits nucleus*, *codons are read*). The student response lists mitosis phases, with no valid key elements present. The backbone rationale is mis-aligned—e.g., *"links the steps involved in protein synthesis"*, and the assessed score is higher than the max score for the question. Our constraint-based rationale correctly notes the ab-

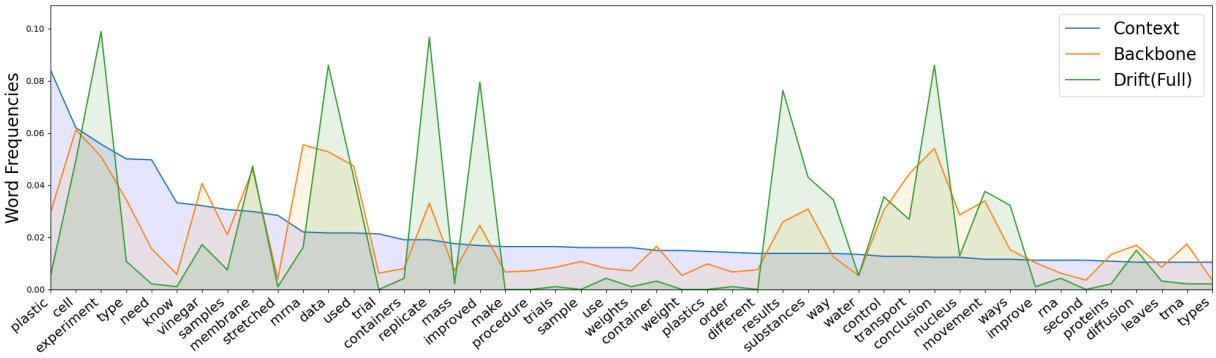

Figure 4: Visualization of the distribution of domain-specific word.

sence of all four elements and assigns 0 points.

For the *NLI* dataset, we present a perturbation example where the word *frugally* is inserted at the beginning of the context. Our method-generated rationale could capture this word sensitively and be reflected in the rationale content. While the backbone model acts less robust, it completely ignores this new insertion and incorrectly predicts the label.

Case studies

Student Answer Assessment

Question: Starting with mRNA leaving the nucleus, list and describe four major steps involved in protein synthesis.

Marking Rubric: 3 points: Four key elements; 2 points: Three key elements; 1 point: One or two key elements; 0 points: No key elements mentioned.

Student Response: "There is anaphase, prophase, metophase, and thelophase." (no valid key elements present)

* **Backbone:** Score: *4 point*; rationale: links the steps involved in protein synthesis, extending into the cytoplasm.
* `Drift`: 0 points; The student's answer does not address the question asked.

Natural Language Inference

Perturbed Premise: frugally Requires free registration.

Hypothesis: Does not require free registration.

* **Backbone:** Entailment; Requires free registration is a necessary condition for only if Requires free registration.
* `Drift:` Contradiction; The premise states that the website [frugally] requires free registration, which implies that a user must provide some information or sign.

6 Conclusion

In this paper, we address the critical challenge of enhancing both accuracy and faithfulness in large language models. Specially, we introduce a probabilistic inference paradigm that incorporates fine-grained and look-ahead rewards to search desirable trajectories. Compared to existing inference-time solutions, our method distinguishes itself through a domain-specific proposal distribution that increases the model's responsiveness to key words in the context. We verify the effectiveness of our approach across three diverse datasets, evaluating both task performance and faithfulness metrics. Furthermore, our model ablation study demonstrates the superiority of integrating both task and rationale rewards.

Limitations

While our proposed method, **`Drift`**, demonstrates promising results in enhancing both task performance and rationale faithfulness, several limitations warrant discussion:

(1) The evaluation tasks are currently confined to label generation, categorizing them as classification tasks with a finite output space. This scope does not yet encompass generative tasks such as mathematical problem-solving. Although Appendix B.1 outlines a potential extension of our framework to such generative tasks, empirical evaluations remain to be conducted.

(2) The current design of the task reward model imposes a strong constraint during the generation process. Ideally, the task reward would effectively re-weight predictions by combining rationale rewards with the backbone model's original predictive probabilities. However, our empirical results indicate that because the classifier already achieves high accuracy, incorporating its outputs into the backbone's token space can introduce prediction uncertainty, thereby compromising both accuracy and faithfulness.

(3) The exploration of different expert models has not been exhaustive. While we have shown that local, global, and weak expert models can contribute to the framework, these experiments are limited considering the vast diversity of pretrained models available on Hugging Face. Future work will aim to investigate how the relevance of expert models' knowledge to the evaluation task influences task performance, and whether patterns emerge that are analogous to how humans seek and utilize useful resources.

Acknowledgment

This work was supported in part by the UK Engineering and Physical Sciences Research Council through a Turing AI Fellowship (grant no. EP/V020579/1, EP/V020579/2) and the Prosperity Partnership scheme (grant no. UKRI566), and by Inkfish through the EMBRACE project.

References

Anonymous. 2024. Weak-to-strong jailbreaking on large language models. In *Submitted to The Thirteenth International Conference on Learning Representations*. Under review.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–294, Toronto, Canada. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Pierre Del Moral and Pierre Del Moral. 2004. *Feynman-kac formulae*. Springer.

Abhimanyu Dubey, Abhinav Jauhri, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yair Gat, Nitay Calderon, Amir Feder, Alexander Chapanin, Amit Sharma, and Roi Reichart. 2023. Faithful explanations of black-box nlp models using llm-generated counterfactuals. *arXiv preprint arXiv:2310.00603*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Ruixin Hong, Hongming Zhang, Hong Zhao, Dong Yu, and Changshui Zhang. 2023. Faithful question answering with Monte-Carlo planning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3944–3965, Toronto, Canada. Association for Computational Linguistics.

Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. 2023. Amortizing intractable inference in large language models. *arXiv preprint arXiv:2310.04363*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Kyungha Kim, Sangyun Lee, Kung-Hsiang Huang, Hou Pong Chan, Manling Li, and Heng Ji. 2024. Can llms produce faithful explanations for fact-checking? towards faithful explainable fact-checking via multi-agent debate. *ArXiv*, abs/2402.07401.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson E. Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, John Kernion, Kamil.e Lukovsiut.e, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Tom Henighan, Timothy D. Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Janina Brauner, Sam Bowman, and Ethan Perez. 2023. Measuring faithfulness in chain-of-thought reasoning. *ArXiv*, abs/2307.13702.

Alexander K. Lew, Tan Zhi-Xuan, Gabriel Grand, and Vikash K. Mansinghka. 2023. Sequential monte carlo steering of large language models using probabilistic programs. *ArXiv*, abs/2306.03081.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiazheng Li, Lin Gui, Yuxiang Zhou, David West, Cesare Aloisi, and Yulan He. 2023. Distilling ChatGPT for explainable automated student answer assessment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

Jiazheng Li, Hainiu Xu, Zhaoyue Sun, Yuxiang Zhou, David West, Cesare Aloisi, and Yulan He. 2024a. Calibrating llms with preference optimization on thought trees for generating rationale in science question scoring. *Preprint*, arXiv:2406.19949.

Zhenyu Li, Sunqi Fan, Yu Gu, Xiuxing Li, Zhichao Duan, Bowen Dong, Ning Liu, and Jianyong Wang. 2024b. Flexkbqa: A flexible llm-powered framework

for few-shot knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18608–18616.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, et al. 2024. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8):AIoa2400196.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024a. Tuning language models by proxy. In *First Conference on Language Modeling*.

Jiacheng Liu, Andrew Cohen, Ramakanth Pasunuru, Yejin Choi, Hannaneh Hajishirzi, and Asli Celikyilmaz. 2024b. Don't throw away your value model! generating more preferable text with value-guided monte-carlo tree search decoding. In *First Conference on Language Modeling*.

Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. 2024c. Decoding-time realignment of language models. In *Proceedings of the International Conference on Machine Learning*.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, pages 1–67.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali. Association for Computational Linguistics.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics ACL 2024*, pages 295–337, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. 2020. Confidence-aware learning for deep neural networks. In *International Conference on Machine Learning*.

openai. 2024. Learning to reason with llm.

Letitia Parcalabescu and Anette Frank. 2023. On measuring faithfulness or self-consistency of natural language explanations. In *Annual Meeting of the Association for Computational Linguistics*.

Wasu Top Piriyakulkij, Cassidy Langenfeld, Tuan Anh Le, and Kevin Ellis. 2024. Doing experiments and revising rules with natural language and probabilistic reasoning. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson E. Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, John Kernion, Kamil.e Lukovsiut.e, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkat Chandrasekaran, Zac Hatfield-Dodds, Jared Kaplan, Janina Brauner, Sam Bowman, and Ethan Perez. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *ArXiv*, abs/2307.11768.

Charlie Victor Snell, Ilya Kostrikov, Yi Su, Sherry Yang, and Sergey Levine. 2023. Offline RL for natural language generation with implicit language q learning. In *The Eleventh International Conference on Learning Representations*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965. Curran Associates, Inc.

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

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *ArXiv*, abs/2405.00451.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5587–5605, Bangkok, Thailand. Association for Computational Linguistics.

Hanqi Yan, Qinglin Zhu, Xinyu Wang, Lin Gui, and Yulan He. 2024. Mirror: Multiple-perspective self-reflection method for knowledge-rich reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7086–7103, Bangkok, Thailand. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Wei Ye, Xing Xie, Weizhu Chen, and Yue Zhang. 2024. Supervised knowledge makes large language models better in-context learners. In *ICLR*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, FangYuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis, and llms evaluations. In *Advances in Neural Information Processing Systems*, volume 36, pages 58478–58507. Curran Associates, Inc.

Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, Wanli Ouyang, and Dongzhan Zhou. 2024a. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *ArXiv*, abs/2410.02884.

Shaolei Zhang, Tian Yu, and Yang Feng. 2024b. TruthX: Alleviating hallucinations by editing large language models in truthful space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8908–8949, Bangkok, Thailand. Association for Computational Linguistics.

Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang, and William Yang Wang. 2024. Weak-to-strong jailbreaking on large language models. *arXiv preprint arXiv:2401.17256*.

A Appendix

A.1 Experiment Setup

We utilize a comprehensive experiment approach with seven datasets across three distinct reasoning tasks. The backbone models adopted for experiments, dataset details, counterfactual generation algorithms and hyper-parameters setting are outlined below.

Backbone model choice. Our experiments select two commonly used backbone model choices: LLaMA3.1-8B (Dubey et al., 2024)[7], and Mistral-7B (Jiang et al., 2023)[8]. Both models are downloaded from the HuggingFace models' space, and we adopted the model implementation from the Huggingface Transformer [9].

Hyper-parameters for inference. For efficient model inference, we applied 8-bit quantization on both the backbone and rationale reward models for *Our+Expert* and *Our(full)* experiments. The task reward models are loaded without any quantization. We set a maximum allowance of 10 different particles during decoding, which means it will keep a maximum of 10 different paths during the search through different weighted decoding paths. The beam factor for expanding searching at each particle is set as 3. To optimize the computational resources for generation, we applied different maximum token length sizes for each task, which we will introduce under each task. As demonstrated in Algorithm 1, our task rewards will be disabled once the answer token is generated to remove the token space constraint. We use a batch size of 64 to inference our framework on a single NVIDIA A100 40G graphic card. The random seed has been set as 42 for all the components.

Predicted label evaluation details. Apart from the faithfulness evaluation details presented in Section 4, the evaluation for the predicted label is extracted and compared with the ground-truth label to calculate the accuracy score. Following each prompt template, we designed a regular expression to extract the score/labels from the generated sequence. If the model fails to follow the prompt to generate a format valid label token, then it counts as a wrongly predicted instance. In short, only correctly predicted instances that follow the prompt required output pattern count towards the accuracy score.

Dealing with rewards from different tokenisation models. In our approach, we address the challenge of integrating rewards derived from various tokenisation models used by rationale reward models. Specifically, after the generation of each token, it is converted into token IDs according to the rationale reward models' token space. Subsequently, rewards are calculated based on samples drawn from the reward model. This method ensures that the generated tokens are consistently evaluated in the context of the expert model's language modeling, and therefore generating meaningful predicted rewards.

Dataset	Min	Max
ASAP-1	0.005	0.123
ASAP-2	0.023	0.114
ASAP-3	0.019	0.089
ASAP-4	0.001	0.104
SNLI	0.110	0.220
MNLI	0.070	0.190
TruthfulQA	0.020	0.240

Table A1: Normalization ranges for each dataset.

Minimum and maximum values for normalization. Because our faithfulness metric does not naturally span the interval $[0, 1]$ (unlike accuracy), we normalize all faithfulness scores and added up in the **Overall** scores using the dataset-specific ranges presented in Table A1:

$$\text{Overall} = \text{Accuracy} + \text{Norm}(\text{Faith}).$$

The **Faith** columns reported in experiment tables contain the original, unnormalized values.

A.1.1 *Student Answer Assessment* Setup

We employed the ASAP [10] dataset to evaluate our methods' effectiveness on student answer assessment reasoning. Following the rationale generation paradigm established by (Li et al., 2023), we adopted the same rationale generation prompt used in their study, focusing on four subsets of science and biology questions. For each dataset, we randomly selected 100 instances from the test split. All the task and rationale reward models are trained solely on the training set. Our empirical analysis shows that zero-shot students answer assessment

[7] meta-llama/Llama-3.1-8B-Instruct
[8] mistralai/Mistral-7B-Instruct-v0.3
[9] https://huggingface.co/docs/transformers/
[10] https://kaggle.com/competitions/asap-sas

Task Reward	Rationale Reward
Student Answer Assessment	
A DeBERTa-v3-large (He et al., 2023) text classifier fine-tuned on the ASAP train sets.	**Choice 1** (LLaMA base): An open-source assessment rationale generation LLM developed by Li et al. (2024a), trained on synthetic rationales generated by GPT-4. **Choice 2** (Mistral base): An open-source, science QA model: Weyaxi/Einstein-v2-7B. This is an out-of-distribution expert model that has never been trained for student assessment.
NLI	
An Open-source BRAT model fine-tuned for classification on train set of multiple NLI datasets, such as MNLI, SNLI etc,.	A LoRA fine-tuned LLaMA 2 model with train set of E-SNLI (Camburu et al., 2018).
QA	
allenai/truthfulqa-truth-judge-llama2-7B, which trained on TruthfulQA's train set.	A LLaMA 2 7B model trained with train set of truthful QA, released by Zhang et al. (2024b).

Table A2: Summary of model choices of our task and rationale reward models.

rationales are typically generated within an average of less than a hundred tokens sequence length. Therefore, we set the maximum generation length for this task as 100.

Prompt template. We apply the prompt template provided in Figure 5 to all our test instances. The `question`, `key_elements`, `marking_rubric`, and `student_answer` correspond to question-dependent question context provided within the dataset:

Task and rationale reward model setup. As demonstrated in Table A2, we utilize a text classifier fine-tuned on the ASAP datasets, built on DeBERTa-v3-large model (He et al., 2023) as the task reward model. We adopted two rationale model choices: **Choice 1**: An open-source explainable student answer scoring LLM developed by Li et al. (2024a). The model is fine-tuned using synthetically generated student answer assessment data with 4-bit quantization with LoRA. **Choice 2**: An out-of-domain, mistral 7B model fine-tuned on science question and answering datasets: Weyaxi/Einstein-v2-7B.

Faithfulness evaluation: sentence-level perturbation for *student answer assessment* dataset. As shown in Algorithm 2, our evaluation strategy involved systematically modifying key phrases from a paragraph of student answer x_i in the input data by comparing with each key answer element k_i from the whole key answer elements set K. Then, observe the resultant variations in the generated rationales. By doing so, we could ascertain whether the rationales remained consistent and aligned with the altered inputs, thereby providing insights that whether the rationale generated is faithful to the given input. Our evaluation approach helps estimate that whether rationales are contextually relevant and robust against variations in input, thereby enhancing their practical utility in real-world applications.

Algorithm 2 Student Answer Perturbation Algorithm

procedure PERTURBATION(x_i, K)
  $\mathbf{S} \leftarrow$ Tokenize(x_i)
  $\mathbf{I} \leftarrow$ array of zeros with length($|\mathbf{S}|$)
  for $j \leftarrow 1$ to $|\mathbf{S}|$ **do**
    for $k \leftarrow 1$ to $|K|$ **do**
      $\mathbf{I}[j] \leftarrow \mathbf{I}[j] + \text{Sim}(\mathbf{S}[j], K[k])$
    end for
  end for
  $i_{\text{max}} \leftarrow \text{argmax}(\mathbf{I})$
  $\mathbf{S} \leftarrow \mathbf{S} \setminus \{\mathbf{S}[i_{\text{max}}]\}$
  $\hat{\mathbf{S}} \leftarrow \text{Joint}(\mathbf{S})$
  return $\hat{\mathbf{S}}$
end procedure

A.1.2 Natural Language Inference (*NLI*) Setup

For NLI, we utilized two key datasets: the Stanford Natural Language Inference (*SNLI*) (Bowman et al., 2015) and the Multi-Genre Natural Language Inference (*MNLI*) (Williams et al., 2018) datasets. These datasets are critical for assessing the ability of our models to handle a range of inferential relationships across various genres, thus providing a comprehensive view of model performance in understanding language context. We randomly selected 100 instances from the official validation split for each dataset; for the MNLI dataset, we used the "matched" set. We empirically examined the explanations from the ESNLI dataset have an average sequence length shorter than 30 tokens.

[Question]: `{question}`
[Key Elements]: `{key_elements}`
[Marking Rubric]: `{marking_rubric}`
[Student Answer]: `{student_answer}`
Please assess this student response and provide rationale, in the format of "x point/s; rationale":

Figure 5: Prompt template for student answer assessment.

Therefore, we employed the maximum generation length of 30 tokens for the NLI task.

Prompt template. We use the prompt template presented in Figure 6 to evaluate our method on all the NLI tasks. The `premise` and `hypothesis` are placeholders corresponds to the premise and hypothesis from the dataset.

Task and rationale reward model setup. As demonstrated in Table A2, we use an open-source fine-tuned BART model (Lewis et al., 2020) to perform NLI classification as the task reward model. Please refer to their released repository for detailed training data usage and splits. For the rationale reward model, we utilize a LoRA fine-tuned Llama-2-7B model on the ESNLI dataset (Camburu et al., 2018). The model is trained solely on the training set of the ESNLI dataset. To reduce computational resources, the task reward model is disabled after generating the answer token.

Hyper-parameters for inference. For efficient model inference, we applied 8-bit quantization on both the backbone and rationale reward models. The task reward model is loaded without quantization. We set a maximum allowance of 10 different particles during decoding. The beam factor for searching is set as 3, with a maximum token length of 30.

Faithfulness evaluation: word-level perturbation for *NLI* tasks. As shown in Algorithm 3, for *NLI*, we identify a keyword among adjective and verb words by POS-tagging using *TokenizeAndTag*. The adj and adv word lists are imported from the nltk package. Once the tokens from the premise or hypothesis are tagged, we randomly insert an irrelevant adjective word into either the premise or hypothesis to create a perturbation using the GenerateExample function. The GenerateExample function takes the whole token lists and the randomTarget word and edit position to reconstruct a perturbed sequence. The goal of evaluation is to detect the modified word from the generated rationale to examine the faithfulness of the rationale generation method.

Algorithm 3 NLI Word Perturbation Generation

procedure PERTURBATION(x_i, adj, adv)
  tokens, tags $\leftarrow$ TokenizeAndTag(x_i)
  targets $\leftarrow$ IdentifyTargets(tags, adj, adv)
  randomTarget $\leftarrow$ SampleTargets(targets)
  example $\leftarrow$ GenerateExample(tokens, randomTarget)
  return example
end procedure

A.1.3 *QA*

The *TruthfulQA* dataset contains questions and answers. Each question has multiple answers, which were adapted into a multiple-choice format. The model's task for this dataset is to select the most truthful answer among all the candidate options.

Prompt template. We use the prompt template presented in Figure 7 to evaluate our method on the *QA* task. The question is the `question` row from the dataset, and the `choices` are candidate answers from the dataset.

Task and rationale reward model setup. We use an open-source truth judge released by Allen AI: allenai/truthfulqa-truth-judge-llama2-7B as the task reward model. For the rationale reward model, we utilize a 7B LLM specialized in truthful QA, released by Zhang et al. (2024b). Please refer to the original paper for the detailed training setup and dataset split for the task and rationale reward models. To reduce computational resources, the task reward model is disabled after generating the answer token.

Hyper-parameters for inference. For efficient model inference, we applied 8-bit quantization on both the backbone and rationale reward models. The task reward model is loaded without quantization. We set a maximum allowance of 10 different particles during decoding. The beam factor for searching is set as 3, with a maximum token length of 30.

Faithfulness evaluation: word-level perturbation for *QA* task. For *QA* task, we identify an

Here is a premise: `{premise}`
Here is a hypothesis: `{hypothesis}`
Please choose whether the hypothesis is entailment, neutral, or contradiction to the premise, and provide a rationale for your choice. Output the label and rationale in the format of "Prediction: [label]; [explanation]":
Prediction:

Figure 6: Prompt template for *NLI* tasks.

Question: `{question}`
Choose the best answer from following options: `{choices}`
Output the selection with reason in the format of Answer: "choice; reason". Answer:

Figure 7: Prompt template for TruthfulQA.

influential word to be replaced, similar to the Algorithm 3. Instead of using an algorithm to perturb the word, in this task, we query the GPT-4 model to modify the original sentence and output both the modified word and the perturbed sentence. Evaluating the faithfulness of the task still depends on the successful rate of reflection of modified words from the rationale.

B Additional Experiment Results

B.1 Dealing with Infinite Label Space

Our method is extendable to scenarios with an infinite label space ($|\mathcal{C}| = \infty$), even though the current evaluations are performed on tasks where the label space is constrained ($|\mathcal{C}| = N \in \mathbb{R}$). For instance, in mathematical problem-solving, the answer can be any arbitrary number. In such cases, the expert model provides a prediction M, with its confidence expressed as the probability w_1 assigned to M, and w_2 to the second most probable prediction. The ratio $\frac{w_1}{w_2}$ serves as an indicator of the expert's confidence in delivering M (Moon et al., 2020). This confidence is then used as a multiplier to enhance the backbone model's prediction for M. Finally, the backbone model's transition distribution is renormalized to maintain a valid probability distribution.

B.2 Computation Cost Analysis

Although rationale and task rewards introduced new computations during the generation processes, we didn't observe a huge computational cost increment in our method. As shown in Figure A3, we calculated the inference time on the *Student Answer Assessment* question #4 to compare the time used between methods on the same GPU. We use a beam size of 3 and a maximum of 100 tokens in generation settings. Compared with the backbone model, our method only increased by 32% on inference time. Compared to other sequential Monte Carlo method, such as *PPO-MCTS* (Liu et al., 2024b), which has a $2S$ times overhead compared to standard decoding from PPO models (S is the number of simulations), our inference-time decoding maintains both the performance and greatly improve the computation efficiency.

Method	Time Cost
Backbone (Beam Search)	88 mins
`:Drift` w. Task	100 mins
`:Drift` w. Rationale	103 mins
`Drift` (Full)	116 mins

Table A3: Computation cost for different methods on *Student Answer Assessment* Q4.

B.3 Proof of Pruned Monte Carlo Search

Definition. We first define the notations: $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are three searched trajectories, among which one trajectory will be pruned. N_A, N_B, N_C are the number of simulations conducted on the corresponding trajectories, W_A, W_B W_C are the total wins for the trajectories.

The estimated value of each branch, i.e., **the probability of being sampled** is defined as:

$$V_A = \frac{W_A}{N_A}, \quad V_B = \frac{W_B}{N_B}$$

Without loss of generalisability, we assume the initial condition and branch C be identified and pruned:

$$V_A > V_B \implies \frac{W_A}{N_A} > \frac{W_B}{N_B}$$

Our proof goal is to show after pruning $\mathcal{C}$, the probability of sampling $\mathcal{B}$ can be larger than $\mathcal{A}$.

Proof. After pruning $\mathcal{C}$, the remaining resources (i.e., simulations) are redistributed to branches A and B. We define R_A and R_B are the additional simulations allocated to $\mathcal{A}$ and $\mathcal{B}$.

After pruning, the new number of simulations for branches A and B are:

$$N'_A = N_A + R_A, \quad N'_B = N_B + R_B$$

After pruning: we define W'_A, W'_B as new total wins after additional simulations. Therefore, the new values for $\mathcal{A}$ and $\mathcal{B}$ are as follows:

$$V'_A = \frac{W'_A}{N_A + R_A} \quad \text{(new estimated value of A)}$$

$$V'_B = \frac{W'_B}{N_B + R_B} \quad \text{(new estimated value of B)}$$

To establish that $V'_B > V'_A$, we require:

$$\frac{W'_B}{N_B + R_B} > \frac{W'_A}{N_A + R_A}$$

Cross-multiplying gives:

$$W'_B \cdot (N_A + R_A) > W'_A \cdot (N_B + R_B)$$

Given that $W'_B > W_B$ and $W'_A < W_A$, it is possible for the following to hold true. For example, in our *NLI* dataset, the undesirable labels are '*contradictory*', so we remove the trajectory $\mathcal{C}$ consisting of '*contradictory*'. For the remaining trajectories, $\mathcal{A}$ and $\mathcal{B}$ are related to '*contradictory*' and '*Neutral*' (not exact label, but similar attitude), respectively. With the removal of '*contradictory*', the new sentence could turn to *neutral* attitude, so the probability of selecting all '*Neutral*'-related trajectories could be largely increased and probability of selecting all '*Neutral*'-related trajectories could be largely penalised.

In this case, even we increase W'_B by increasing the N_B, the substantial enhancement of W'_B still could lead to a larger V'_B.

Thus, we can conclude: After pruning branch C, the additional simulations allocated to branch B can increase its estimated value due to improved exploration, leading to:

$$V'_B > V'_A$$

□