Learning Together to Perform Better: Teaching Small-Scale LLMs to Collaborate via Preferential Rationale Tuning

Sohan Patnaik*, **Milan Aggarwal***, **Sumit Bhatia, Balaji Krishnamurthy**
Media and Data Science Research (MDSR) Lab, Adobe
Correspondence: soha@adobe.com, milaggar@adobe.com

Abstract

LLMs such as GPT-4 have shown a remarkable ability to solve complex questions by generating step-by-step rationales. Prior works have utilized this capability to improve smaller and cheaper LMs (say, with 7B parameters). However, various practical constraints, such as copyright and legal issues, owing to lack of transparency in the pre-training data of large (often closed) models, prevent their use in commercial settings. Little focus has been given to improving the *innate* reasoning ability of smaller models without distilling information from larger LLMs. To address this, we propose COLLATE, a trainable framework that tunes a (small) LLM to generate those outputs from a pool of diverse rationales that selectively improves the downstream task. COLLATE enforces multiple instances of the *same* LLM to exhibit distinct behavior and employs them to generate rationales to obtain diverse outputs. The LLM is then tuned via preference optimization to choose the candidate rationale which maximizes the likelihood of ground-truth answer. COLLATE outperforms several trainable and prompting baselines on 5 datasets across 3 domains - maths problem solving, natural language inference, and commonsense reasoning. We show the efficacy of COLLATE on LLMs from different model families across varying parameter scales (1B to 8B) and demonstrate the benefit of multiple rationale providers guided by the end task through ablations. Code is released here.

1 Introduction

Large Language Models (LLMs) are ubiquitously used to solve various Natural Language Processing (NLP) tasks (Brown et al., 2020). Due to cost and latency constraints, small-scale LLMs are more suitable for deployment in applications. Given a task-specific dataset consisting of input-output pairs, the LLM is trained on this data through Supervised Fine-Tuning (SFT) to perform the task.

Additionally, prior methods have shown that soliciting the LLM to first elucidate a rationale describing the steps required to derive the answer yields better performance compared to generating the final response directly (Wei et al., 2022a; Kojima et al., 2022). Such a capability is notably exhibited by massive-scale LLMs such as GPT-4 (Achiam et al., 2023), PaLM-540B (Chowdhery et al., 2022) etc. which solve complex questions by generating step-by-step rationales (Wei et al., 2022b). Hence, several works improve smaller LLMs through larger LLMs by using their generated rationales (Hsieh et al., 2023; Tunstall et al., 2023).

However, lack of transparency in the pre-training data of larger (often closed) LMs prevents their use in commercial settings due to legal constraints. Models like GPT-4 restrict using their outputs to train other models for commercial use. Hence, the responses of such models can only be used through prompting. Further, long-term use of such LLMs and their APIs can be costly, compute-intensive, and unreliable from a maintenance perspective. Limited attention has been given to developing reasoning ability of small-scale efficient LLMs without relying on any other external LLM. Although Chain-of-Thought (CoT) dataset (Kim et al., 2023) (comprising of instruction-rationale-response triples) was introduced to teach small LMs to generate rationales via Instruction Fine-Tuning (IFT), it does not contain specific rationale annotations needed for each task. Likewise, prior works such as SPIN (Chen et al., 2024) proposes to tune LLM via Direct Preference Optimisation (DPO - please refer appendix C) (Rafailov et al., 2023) by selecting ground-truth (GT) response as the chosen output over LLM-generated answer. However, lack of availability of annotated rationales for an end-task limits their applicability. Other methods (Yuan et al., 2024) rely on the scale of large LMs to generate diverse responses and rate them via LLM-as-a-judge paradigm (Zheng et al.,

---

*Equal contribution.

15493

2023) for preference tuning. Such approaches do not generalise well to small-scale LLMs as demonstrated in the experiments section.

Given these constraints, we ask an important research question - *"Can we teach a small-scale LLM to interact with itself without relying on any other LLM to refine the ability of generating rationales which improves a given end-task?"* We hypothesize that LLMs acquire adequate knowledge during pre-training that is required to generate reasonable rationales. However, the output distribution of the rationales generated by the LLM has to be aligned in a manner that is guided by the final task. Driven by this intuition, we generate diverse rationales for each task sample and optimise the LLM to selectively choose the rationale which enhances the chances of generating correct answer. To achieve this, we propose COLLATE (Tea**C**hing to **COLL**aborate via Preferenti**A**l Ra**T**ional**E** Tuning), a framework that leverages multiple *Rationale Provider* LLMs to obtain the set of diverse rationales. The rationale providers are obtained by creating multiple instances of the same LLM such that they exhibit distinct behavior. Given a task instruction, the rationale providers are each employed to generate rationale to create a pool of rationales. Each output in the obtained set of rationales is then assigned a *usefulness score* – estimated as the likelihood of generating ground-truth answer for the given task conditioned on rationale in input. The usefulness score is used to rank and select the rationales to tune the LLM through DPO. We pick only those samples for DPO where best rationale enhances ground-truth answer likelihood compared to not using any rationale.

We evaluate the effectiveness of COLLATE based on the utility of the generated rationales to improve performance of small-scale LLMs on five datasets across three diverse task domains: 1) Maths Word Problem Solving - GSM8K (Cobbe et al., 2021), 2) Natural Language Inference - PIQA (Bisk et al., 2020) and WinoGrande (Sakaguchi et al., 2020), and 3) Commonsense Reasoning - CSQA (Talmor et al., 2019) and HellaSwag (Zellers et al., 2019). We observe that COLLATE outperforms several baselines (§ 4.1) by gains of up to 7% without dependence on other larger LLMs (unlike prior methods) for generating diverse rationales and rating their quality. Further, we show the efficacy of COLLATE on improving a variety of LLMs across varying parameter-scales of 1B to 8B (§ 4.2). We also conduct **human stud-**

ies which show that the rationales from COLLATE are reliable, helpful and **diverse** (§ 4.3). Further, we conduct extensive ablations to validate - **1)** employing rationale providers for diverse rationales as opposed to sampling-based decoding; and **2)** using end-task guided likelihood-based rationale selection and sample filtration for DPO (Appendix 4.4).

2 Related Work

Prompt-based Reasoning Generation: It has been shown that generating intermediate reasoning chains improves the performance of large-scale LLMs (Wei et al., 2022a; Kojima et al., 2022). Chain-of-Thought (CoT) prompting (Wei et al., 2022b) is the first such technique which performs this by showing exemplar demonstrations of step-by-step reasoning in the prompt. Consequently, some other methods have focused on improving the diversity of the exemplar demonstrations in the prompt (Zhang et al., 2023; Diao et al., 2024; Li and Qiu, 2023a,b). Tree-of-Thought (ToT) (Yao et al., 2023) was introduced as a generalisation of CoT where different reasoning paths are organised in the form of a tree such that the LLM can look-ahead or backtrack in tree to yield the optimal reasoning chains. Further, many self-correction methods were proposed where the LLM identifies and rectifies its own mistakes (Madaan et al., 2023; Wang et al., 2023a). However, it was shown that a single LLM struggles to improve its rationales through just prompting-based mechanisms in the absence of external feedback (Huang et al., 2024; Valmeekam et al., 2023; Stechly et al., 2023).

Rationale Enhancement via Preference Training: Some works optimise reasoning chains by obtaining a set of rationales and applying Direct Preference Optimisation (DPO) (Rafailov et al., 2023). SPIN (Chen et al., 2024) proposes to choose the final ground-truth (GT) answer over the LLM generated final response for DPO training. Likewise, Lai et al. (2024) shows that DPO can be applied at each step of the reasoning chain which requires extensive annotations. Lack of availability of GT rationales on a new task limit their applicability. On the other hand, we employ different instances of the same LLM to generate diverse rationales for selection during DPO. Zhang et al. (2024) proposed an alternative way to obtain multiple rationales by extracting reasoning chains sampled at each step of ToT tree search. Very recently, Wang et al. (2024) proposed sampling-based decoding to
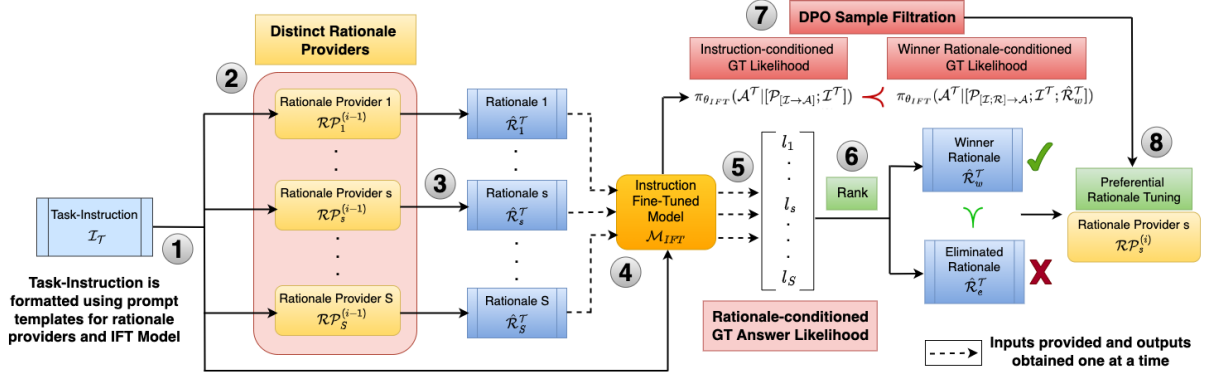
Figure 1: Overview of architecture of COLLATE. The task-instruction ($\mathcal{I}^{\mathcal{T}}$) is pre-processed (step 1) and given as input to distinct rationale providers - $\mathcal{RP}_s^{(i-1)}$ (step 2) to obtain set of diverse rationales - $\{\hat{\mathcal{R}}_s^{\mathcal{T}}\}$ (step 3). IFT version of the LLM - $\mathcal{M}_{\mathcal{IFT}}$ is used to estimate likelihood (l_s) of generating ground-truth (GT) answer conditioned on task-instruction and each rationale (steps 4 and 5) which is used to rank rationales to identify winning ($\hat{\mathcal{R}}_w^{\mathcal{T}}$) and eliminated ($\hat{\mathcal{R}}_e^{\mathcal{T}}$) rationales (step 6). If the winning-rationale enhances likelihood of obtaining GT answer compared to using just the instruction (step 7), the sample is used to perform DPO to obtain rationale provider $\mathcal{RP}_s^{(i)}$ (step 8).

obtain diverse rationales for math problems such that rationales which contain GT answer are considered as preferred outputs. Such an approach cannot be extended to non-math domains where just the presence of answer is not indicative of rationale quality. Yuan et al. (2024) showed that very-large LMs (Llama-2 70B) can be used to rank outputs through prompting. Contrastingly, COLLATE rates rationale quality by leveraging LLM's likelihood of GT answer conditioned on the rationale.

Response Synthesis through Multi-LLM Interaction: Several works have explored ensembling multiple LLMs to improve overall response quality (Jiang et al., 2023b; Yu et al., 2024; Juneja et al., 2023; Ulmer et al., 2024; Lu et al., 2024b). Zephyr (Tunstall et al., 2023) distills responses generated by Falcon (Penedo et al., 2023), Llama (Touvron et al., 2023) etc. into Mistral-7B (Jiang et al., 2023a) by rating them using GPT-4 (Achiam et al., 2023) to perform DPO. Likewise, Hsieh et al. (2023) distills rationales from PaLM-540B model (Chowdhery et al., 2022) into T5 (Raffel et al., 2023) through SFT. Few approaches personalise the feedback from larger teacher LLM based on weaknesses of smaller student LLM (Wang and Li, 2023; Saha et al., 2023; Jiang et al., 2023c). Kang et al. (2023) aim at improving LLMs in a task-oriented manner by retrieving knowledge using rationales obtained from a larger LLM. Orca (Mukherjee et al., 2023) argues that IFT over GPT-generated outputs makes smaller LLM imitate just the style (Gudibande et al., 2024) and proposes to train smaller LLM on explanation traces of GPT-

4. Lack of transparency in training data of larger LLMs hinders their commercial use due to legal implications. Other methods mimic the way humans conduct discussions (Mousavi et al., 2023) by facilitating prompt-based deliberation (Yin et al., 2023). We show that such methods work well only with large LMs but performs poorly with smaller LMs.

3 Methodology

Overview of our Approach: Figure 1 provides an outline of the proposed COLLATE framework to generate better rationales for improving end-task performance *without* relying on distilling information from any external LLM. To achieve this, we first train the instruction fine-tuned (IFT) version of the base LLM - $\mathcal{M}_{IFT}$ (§ 3.1). We then create multiple rationale providers by cloning S instances of $\mathcal{M}_{IFT}$ and optimise them on separate data splits to ensure they exhibit distinct behaviour (§ 3.2). The rationale providers are tuned through preference optimisation (DPO) to selectively generate those outputs from a pool of diverse rationales which help in improving the end-task (§ 3.3). The set of diverse rationales is obtained by generating reasoning chains from the *Rationale-Provider* ($\mathcal{RP}$) LLMs (steps 1-3 in fig.1). Once the candidate set of rationales is obtained, a usefulness score is assigned to each rationale based on end-task performance. The score is estimated as the likelihood of generating the ground-truth (GT) answer by $\mathcal{M}_{IFT}$ conditioned on the rationale in input (steps 4-5 in fig.1). The score is used to select the winning and eliminated outputs for DPO (steps 6-8 in fig.1). We

feed only high-quality samples for DPO based on whether the winning rationale enhances likelihood of generating GT compared not using any rationale (step 7 in fig.1). We now describe the individual components of COLLATE in detail.

3.1 Multi-Mode Instruction Fine-Tuning

Conventional Instruction Fine-Tuning (IFT) aims at enabling the LLM $\mathcal{M}$ to follow a given instruction to generate the final response accordingly. However, we require the instruction-tuned model $\mathcal{M}_{IFT}$ for two additional objectives - 1) generate rationale required to derive the final answer given the instruction as input; 2) generate/estimate the likelihood of producing an answer given the instruction and rationale as input. To enable $\mathcal{M}_{IFT}$ to operate in different modes, we format the samples using different prompts (please refer appendix B) during training to indicate the model about the mode in which it needs to generate the output. We leverage a dataset $\mathcal{D}_{IFT}^{rationale}$ which comprises of samples containing rationales in addition to instruction-answer pairs to enable the two additional modes discussed above. Formally, given an instruction $\mathcal{I}$, rationale $\mathcal{R}$ and final answer $\mathcal{A}$, we perform IFT in three modes using cross-entropy loss through teacher forcing (Vaswani et al., 2017):

$$\mathcal{L}_{\mathcal{I}\to\mathcal{A}} = -log\,p(\mathcal{A}_t|\,[\mathcal{P}_{\mathcal{I}\to\mathcal{A}};\mathcal{I};\mathcal{A}_{<t}],\theta_{IFT}) \quad (1)$$

$$\mathcal{L}_{\mathcal{I}\to\mathcal{R}} = -log\,p(\mathcal{R}_t|\,[\mathcal{P}_{\mathcal{I}\to\mathcal{R}};\mathcal{I};\mathcal{R}_{<t}],\theta_{IFT}) \quad (2)$$

$$\mathcal{L}_{[\mathcal{I};\mathcal{R}]\to\mathcal{A}} = -log\,p(\mathcal{A}_t|\,[\mathcal{P}_{[\mathcal{I};\mathcal{R}]\to\mathcal{A}};\mathcal{I};\mathcal{R};\mathcal{A}_{<t}],\theta_{IFT}) \quad (3)$$

where, p represents probability, $\mathcal{A}_t$ and $\mathcal{R}_t$ depict the tth token in ground-truth answer and rationale respectively, $< t$ indicates tokens before t^{th} index, $\mathcal{P}_m$ and $\mathcal{L}_m$ are the prompt format and loss function respectively for the mth mode; $[;]$ represents the formatting operation to prepare LLM input after arranging the instruction, answer and/or rationale into mode-specific prompt $\mathcal{P}_m$, and θ_{IFT} is the set of trainable LLM parameters. For samples in IFT data which do not contain rationales, only loss $\mathcal{L}_{\mathcal{I}\to\mathcal{A}}$ is applied. Once $\mathcal{M}_{IFT}$ is trained, it is used to obtain distinct rationale provider LLMs and rank rationales generated by them for task-guided preferential rationale tuning in subsections to follow.

3.2 Distinct Rationale Providers

To improve rationale generation ability on a new task, COLLATE generates diverse rationales to obtain a candidate set of outputs that can be used to optimise the LLM via DPO. To obtain diverse rationales, we train multiple Rationale Provider (RP) LLMs - $\mathcal{RP}_1, \mathcal{RP}_2, ..., \mathcal{RP}_S$ such that they can be leveraged to obtain a diverse set of outputs. The rationale providers are obtained by cloning S instances of $\mathcal{M}_{IFT}$ and tuning them for rationale generation on different data splits so that they exhibit different behavior. Specifically, we consider a dataset containing rationale annotations - $\mathcal{D}^{rationale}$ and divide it into S equal splits randomly. We denote s^{th} data split as $\mathcal{D}_s^{rationale} = \{d_1^s, ..., d_{N/S}^s\}$, where $1 \le s \le S$, N is the number of samples in $\mathcal{D}^{rationale}$, and d_j^s denotes j^{th} sample in split s comprising of instruction-rationale-response triples - $\{\mathcal{I}_j^s, \mathcal{R}_j^s, \mathcal{A}_j^s\}$. We assign the sth data split to rationale-provider $\mathcal{RP}_s$ and prompt it to generate the rationale as:

$$\hat{\mathcal{R}}_j^s = \mathcal{RP}_s([\mathcal{P}_{\mathcal{I}\to\mathcal{R}};\mathcal{I}_j^s]\,|\,\theta_{RP_s}) \quad (4)$$

The rationale provider $\mathcal{RP}_s$ is then fine-tuned through DPO by choosing the ground-truth rationale $\mathcal{R}_j^s$ as the preferred option over the generated rationale $\hat{\mathcal{R}}_j^s$. Formally, $\mathcal{RP}_s$ is optimised using the following loss function $\mathcal{L}_{\mathcal{RP}_s}$:

$$\mathcal{L}_{\mathcal{RP}_s} = -log\sigma[\beta(log\frac{\pi_{\mathcal{RP}_s}(\mathcal{R}_j^s)}{\pi_{\mathcal{M}_{IFT}}(\mathcal{R}_j^s)} - log\frac{\pi_{\mathcal{RP}_s}(\hat{\mathcal{R}}_j^s)}{\pi_{\mathcal{M}_{IFT}}(\hat{\mathcal{R}}_j^s)})] \quad (5)$$

where, $\pi_{\mathcal{Z}}(y)$ represents the likelihood of generating tokens in $y \in \{\mathcal{R}_j^s, \hat{\mathcal{R}}_j^s\}$ by model $\mathcal{Z} \in \{\mathcal{RP}_s, \mathcal{M}_{IFT}\}$, σ is sigmoid activation, $\beta (= 0.1)$ is a coefficient to control deviation from reference model $\mathcal{M}_{IFT}$. This yields a set of S distinct rationale providers which are used to generate diverse rationales for a given task-specific instruction. The generated rationales are used to optimise the rationale providers further in a task-guided manner as described in next sub-section. We refer to s^{th} rationale provider obtained till this stage as $\mathcal{RP}_s^{(0)}$ as a convention for subsequent discussions.

3.3 Task-Based Preferential Rationale Tuning

For a given end-task $\mathcal{T}$, we denote the corresponding dataset as $\mathcal{D}^{\mathcal{T}}$ comprising of instruction-answer pairs of the form $(\mathcal{I}^{\mathcal{T}}, \mathcal{A}^{\mathcal{T}})$. However, the supervised dataset for the given task does not contain rationale annotations. To mitigate this, we leverage distinct rationale providers to construct a set of diverse rationales. Quality of a rationale is determined based on its usefulness to enhance end-task performance i.e. likelihood of generating the ground-truth answer $\mathcal{A}^{\mathcal{T}}$. Each rationale provider

is then optimised iteratively to prefer generating better rationale candidate as opposed to other options which result in lower likelihood of generating the ground-truth answer. The rationale providers are tuned for numerous iterations by conducting multiple passes over $\mathcal{D}^{\mathcal{T}}$.

In particular, consider the ith iteration ($i \in \{1, 2\}$) such that the task-instruction $\mathcal{I}^{\mathcal{T}}$ is given as input to each rationale provider - $\mathcal{RP}_s^{(i-1)}$ (obtained till last $(i-1)^{th}$ iteration) separately to generate the corresponding rationale as depicted in the following equation (steps 1-3 in fig.1):

$$\hat{\mathcal{R}}_s^{\mathcal{T}} = \mathcal{RP}_s^{(i-1)}([\mathcal{P}_{\mathcal{I}\to\mathcal{R}}; \mathcal{I}^{\mathcal{T}}] \mid \theta_{RP_s^{(i-1)}}) \quad (6)$$

We obtain a set of generated rationales - $\{\hat{\mathcal{R}}_1^{\mathcal{T}}, \hat{\mathcal{R}}_2^{\mathcal{T}}, ..., \hat{\mathcal{R}}_S^{\mathcal{T}}\}$. Each rationale in the set is assigned a usefulness score l_s ($1 \leq s \leq S$) by estimating the likelihood of generating the ground-truth (GT) answer $\mathcal{A}^{\mathcal{T}}$ by the instruction fine-tuned model $\mathcal{M}_{IFT}$ conditioned on rationale and task-instruction as input - $\pi_{\theta_{IFT}}(\mathcal{A}^{\mathcal{T}} \mid [\mathcal{P}_{[\mathcal{I};\mathcal{R}]\to\mathcal{A}}; \mathcal{I}^{\mathcal{T}}; \hat{\mathcal{R}}_s^{\mathcal{T}}])$ (steps 4 and 5 in fig.1). The rationales $\hat{\mathcal{R}}_s^{\mathcal{T}}$ ($1 \leq s \leq S$) are ranked based on their usefulness score l_s such that the first and last elements in the ranked list are selected as winner ($\hat{\mathcal{R}}_w^{\mathcal{T}}$) and eliminated ($\hat{\mathcal{R}}_e^{\mathcal{T}}$) rationales respectively (step 6 in fig. 1):

$$w = \underset{1 \leq s \leq S}{\arg\max}\{l_s\} \; ; \; e = \underset{1 \leq s \leq S}{\arg\min}\{l_s\} \quad (7)$$

To maintain the distinctness in the behaviour of different rationale providers, we tune them on different splits of $\mathcal{D}^{\mathcal{T}}$. The task-specific dataset $\mathcal{D}^{\mathcal{T}}$ is divided into S equal splits randomly such that split $\mathcal{D}_s^{\mathcal{T}}$ ($1 \leq s \leq S$) is assigned to optimise rationale provider $\mathcal{RP}_s^{(i)}$. Without loss of generality, consider $(\mathcal{I}^{\mathcal{T}}, \mathcal{A}^{\mathcal{T}}) \in \mathcal{D}_s^{\mathcal{T}}$. $\hat{\mathcal{R}}_w^{\mathcal{T}}$ and $\hat{\mathcal{R}}_e^{\mathcal{T}}$ are used to tune s^{th} rationale provider $\mathcal{RP}_s^{(i)}$ in current iteration i via preference optimisation (step 8 in fig. 1) using the loss $\mathcal{L}_{\mathcal{RP}_s^{(i)}}^{\mathcal{T}}$:

$$\mathcal{L}_{\mathcal{RP}_s^{(i)}}^{\mathcal{T}} = -log\sigma[\beta(log\frac{\pi_{\mathcal{RP}_s^{(i)}}(\hat{\mathcal{R}}_w^{\mathcal{T}})}{\pi_{\mathcal{RP}_s^{(j)}}(\hat{\mathcal{R}}_w^{\mathcal{T}})} - log\frac{\pi_{\mathcal{RP}_s^{(i)}}(\hat{\mathcal{R}}_e^{\mathcal{T}})}{\pi_{\mathcal{RP}_s^{(j)}}(\hat{\mathcal{R}}_e^{\mathcal{T}})})] \quad (8)$$

where, $\beta = 0.1$ and $j = (i-1)$. It can be noted that the previous iteration $(i-1)$ version of the rationale provider - $\mathcal{RP}_s^{(i-1)}$ is used as the reference model for the DPO training to obtain $\mathcal{RP}_s^{(i)}$.

Likelihood-based Sample Filtration: To ensure that high-quality samples are used to tune rationale providers, we apply a filtration criteria to remove the samples where the winning rationale does not enhance the likelihood of generating the GT. Formally, we retain the samples $(\mathcal{I}^{\mathcal{T}}, \mathcal{A}^{\mathcal{T}})$ which qualify the following criteria (step 7 in fig.1):

$$\pi_{\theta_{IFT}}(\mathcal{A}^{\mathcal{T}} \mid [\mathcal{P}_{[\mathcal{I};\mathcal{R}]\to\mathcal{A}}; \mathcal{I}^{\mathcal{T}}; \hat{\mathcal{R}}_w^{\mathcal{T}}]) > \pi_{\theta_{IFT}}(\mathcal{A}^{\mathcal{T}} \mid [\mathcal{P}_{[\mathcal{I}\to\mathcal{A}]}; \mathcal{I}^{\mathcal{T}}]) \quad (9)$$

Equation 9 compares the likelihood of generating the ground-truth by $\mathcal{M}_{IFT}$ for the task instruction in the absence and presence of the winning rationale in the input. The sample is used for DPO training if the winning rationale enhances the likelihood compared to not using any rationale.

4 Experiments and Evaluation

Datasets: We evaluate COLLATE on three diverse reasoning task domains using five datasets - **1) Maths Problem Solving** on GSM8K (Cobbe et al., 2021); **2) Natural Language Inference (NLI)** using PIQA (Bisk et al., 2020) and WinoGrande (Sakaguchi et al., 2020); and **3) Commonsense Reasoning** through CSQA (Talmor et al., 2019) and HellaSwag (Zellers et al., 2019). **GSM8K** comprises of maths word problems requiring understanding of problem and sequence of calculations. **PIQA** requires understanding of physical relation between objects and comprises of samples with a goal text coupled with two candidate texts out of which only one can lead to the goal. **WinoGrande** is a very challenging co-reference resolution task, comprising of a statement with two parts such that the latter half refers to some entity in the first part. **CSQA** tests model's ability to answer MCQ questions by picking correct choice using commonsense knowledge. **HellaSwag** evaluates ability to predict continuation of a context by choosing most plausible ending. Number of samples in train/test splits of each dataset is GSM8k - 7.5k/1.3k, PIQA - 16k/3k, WinoGrande - 40k/1.7k, CSQA - 9.7k/1.2k and HellaSwag - 39.9k/10k. Please refer to the appendix D for examples from each dataset.

Implementation Details: In all training runs, we use a batch size (BS) of 16 on 8 80GB A100 GPUs (BS of 2/GPU), a learning rate of 1e-5, bfloat16 precision with cosine annealing (Loshchilov and Hutter, 2017) using AdamW optimizer (Loshchilov and Hutter, 2019). We leverage DeepSpeed Zero 2 with sharding of optimizer states and gradients across GPUs and enable gradient check-pointing.

Method	Maths	NLI		Comonsense	
	GSM8K	WinoGrande	PIQA	HellaSwag	CSQA
Prompt-based Rationale Refinement					
Chain-of-Thought (Wei et al., 2022b)	13.55	28.19	28.81	55.89	36.27
CoT Self-Consistency (Wang et al., 2023b)	14.56	33.32	31.04	51.23	37.17
Tree-of-Thought (Yao et al., 2023)	17.31	30.10	31.61	63.33	39.49
Exchange-of-Thought (Yin et al., 2023)	26.80	33.29	38.17	72.13	55.31
Task-specific Supervised Fine-Tuning					
open-llama-7b-v2 SFT (Geng and Liu, 2023)	30.76	48.81	50.94	94.79	73.82
Distilling Step-by-Step (Hsieh et al., 2023)	38.41	50.69	50.65	96.01	75.02
Training-driven Rationale Enhancement					
Self-Rewarding LMs (Yuan et al., 2024)	39.13	41.44	47.41	93.26	71.77
SPIN (Chen et al., 2024)	48.41	50.58	51.94	98.14	78.76
COLLATE (ours)	**53.48**	**53.49**	**59.57**	**99.21**	**80.13**

Table 1: Comparison of COLLATE with multiple categories of baselines on five datasets over three task domains - Maths Problem Solving, Natural Language Inference (NLI) and Commonsense Reasoning. COLLATE performs better than baselines. We use common LLM backbone i.e. open-llama-v2-7b for all methods in this table.

1) Multi-Mode Instruction Fine-Tuning (IFT) - We experiment with different LLM backbones $\mathcal{M}$ (open-llama-v2-7b, OLMo-1B, Phi3-3.8B, Qwen1.5-4B and LLaMA3-8B) to obtain $\mathcal{M}_{IFT}$. We use a random subset of 140k samples from CoT-Collection (Kim et al., 2023) as $\mathcal{D}_{IFT}^{rationale}$ to enable two additional IFT modes - $\mathcal{I} \rightarrow \mathcal{R}$ and $[\mathcal{I}; \mathcal{R}] \rightarrow \mathcal{A}$. We use 40k samples of Dolly-HHRLHF (MosaicML, 2023) and Open Assistant datasets combined to create samples for $\mathcal{I} \rightarrow \mathcal{A}$ mode. The IFT training is performed for 2 epochs.

2) Distinct Rationale Providers - We tune $S = 2$ clones of $\mathcal{M}_{IFT}$ using subset of 195k samples in CoT-Collection ($\mathcal{D}^{rationale}$) to obtain rationale providers. DPO training is performed for 5 epochs.

3) Task-guided Preferential Rationale Tuning - We carry out 2 iterations of task-guided DPO with 10 epochs in each iteration. The rationale providers are evaluated on a val-split of CoT comprising of 8k samples to identify the rationale provider to be used for evaluation as described next.

Evaluation Procedure: We evaluate COLLATE and baselines by conducting rationale-conditioned Supervised Fine-Tuning (SFT) of the base LLM $\mathcal{M}$ (for 3 epochs) on task-specific dataset comprising of pairs of input instruction and final answer. The rationale generated by a method is given as additional input along with task instruction during this SFT. The accuracy achieved is indicative of usefulness of rationales. We use a single rationale provider while assessing rationales generated by COLLATE. The best rationale provider is selected based on likelihood estimation on CoT val-split samples where we measure probability of generating ground-truth answer by the IFT model $\mathcal{M}_{IFT}$ conditioned on the rationale. The rationale provider that yields maximum value (averaged across samples) is used for evaluation. Figure 2 in Appendix J shows correlation plot between task test-accuracy and likelihood indicating that rationale provider with best likelihood is a suitable choice.

4.1 Comparison of COLLATE with Baselines

We present a detailed comparative analysis of COLLATE with three categories of baselines (Table 1) - **(i) Prompting** techniques which enable the LLM to either generate reasoning chains in different ways such as Chain-of-Thought (Wei et al., 2022b), Tree-of-Thought (Yao et al., 2023), etc. or facilitate exchange of feedback between multiple LLMs - Exchange-of-Thought (Yin et al., 2023); **(ii) Task-specific Supervised Fine-Tuning (SFT)** of LLM without any rationales in the input and with rationales generated by the IFT version of the LLM $\mathcal{M}_{IFT}$ - Distilling Step-by-Step (Hsieh et al., 2023); and **(iii) Training-driven Rationale Enhancement** where the LLM is tuned to improve rationales iteratively through DPO - SPIN (Chen et al., 2024) and Self-rewarding Language Models (Yuan et al., 2024)). We employ a common LLM backbone (open-llama-v2-7b) for COLLATE as well as baselines for a fair comparison.

Table 1 summarises the results where it can be seen that COLLATE outperforms all the baselines

Task-Instruction and Rationales Generated by Best Baseline (SPIN) and Rationale Providers - $\mathcal{RP}_1$ and $\mathcal{RP}_2$ in COLLATE
Instruction: Betty is saving money for a new wallet which costs $100. Betty has only half of the money she needs. Her parents decided to give her $15 for that purpose, and her grandparents twice as much as her parents. How much more money does Betty need to buy the wallet?
Baseline: Betty has $50. Her parents give her $15, and her grandparents give her twice that, which is $100. So, she gets an additional $115. Now she has $50 + $115 = $165. Therefore, she needs $65 more to buy the wallet. ✗
Rationale Provider 1: Betty needs $100 for a new wallet. She currently has half of that amount, i.e. $100/2=$50. Her parents contribute $15 towards her goal, and her grandparents contribute twice as much as her parents, which is $30. Betty currently has $50 + $15 + $30 = $95. We need to subtract her current amount from the price of the wallet, which is $100 - $95 = $5. ✓
Rationale Provider 2: Betty needs $100 to buy the wallet. She has $50. Her parents give her $15, and her grandparents give her twice as much, which is $30. The amount she needs is $55. ✗

Table 2: Qualitative comparison of rationales generated by best baseline (SPIN) and rationale providers in COLLATE on GSM8K. COLLATE generates better rationales by virtue of employing distinct rationale providers.

uniformly across the three task domains. Specifically, it performs better than the best baseline (SPIN) by $\sim 5\%$ on GSM8K indicating the helpfulness of rationales from COLLATE for solving maths problems. Further, COLLATE achieves significantly better results than SPIN for NLI with an improvement of $\sim 7.5\%$ on PIQA and $\sim 3\%$ on WinoGrande where most baselines achieve close to random-selection accuracy. Finally, for commonsense reasoning, even though the accuracy of SPIN (98.14%) is already close to 100 on HellaSwag, COLLATE still performs better by achieving 50% of remaining possible improvement. In particular, COLLATE performs better than SPIN by 1.37% on CSQA and 1.07% on HellaSwag.

Additionally, COLLATE gives a significant performance boost to the base LLM compared to both, carrying out SFT over task input-output pairs without rationales, and using the rationales generated by the IFT model as additional input during SFT (Distilling Step-by-Step - DSS). Compared to 'DSS', there is an increase of $\sim 15\%$ on GSM8K, 3.2% on HellaSwag, $\sim 5\%$ on CSQA, $\sim 3\%$ on WinoGrande, and $\sim 9\%$ on PIQA . Moreover, the baseline 'Self-Rewarding Language Models' which leverages larger LLM (Llama-70B) to both generate and rate diverse rationales does not generalise with 7B LM. It performs worse than 'DSS' on most tasks. Finally, it is noted that prompting-based methods give significantly lower performance at the scale of 7B LLM. We also perform few-shot prompt-based evaluation (instead of SFT) of rationales from COLLATE in appendix E where we observe a similar trend compared to baselines.

Table 2 compares the quality of rationales generated by rationale providers with best baseline (SPIN). COLLATE generates better rationales by virtue of employing distinct rationale providers. Please refer to appendix A for more examples.

4.2 Does COLLATE Enhance Small LLMs of Varying Scales and Model Families?

We verify if COLLATE improves performance of multiple small-scale LLMs of varying scales ranging from **1B to 8B** parameters belonging to different model families. We compare the accuracy of task-specific SFT version of $\mathcal{M}$ trained using rationales generated by COLLATE vs. the rationales from IFT model $\mathcal{M}_{IFT}$. Specifically, we experiment with **1) OLMo-1B** (Groeneveld et al., 2024), **2) Phi3-3.8B** (Abdin et al., 2024), **3) Qwen1.5-4B** (Bai et al., 2023), **4) open-llama-v2-7B** (Geng and Liu, 2023), and **5) LLaMA3-8B** (Dubey et al., 2024). Table 3 summarises the results where it can be seen that COLLATE provides significant performance increase on all the tasks uniformly over parameter-scales and LLM families. Across parameters scale, there is a gain of $4-8\%$ on GSM8K, $2.5-6\%$ on WinoGrande, $4-9\%$ on PIQA, $\sim 1.5-5\%$ on HellaSwag and $5-7\%$ on CSQA. Likewise across tasks, there is a boost of $4-7\%$ at 1B scale ; $3-7\%$ increase at $\sim$4B scale; and $\sim 1.5-9\%$ gain for 7-8B parameters LLMs.

4.3 Human Study of COLLATE's Rationales

We conducted a human study to evaluate the rationales generated by the rationale providers (details in Appendix K). We asked human evaluators to judge if 1) final rationale obtained from COLLATE is correct and helpful, and 2) whether the rationale generated by one of the rationale providers is better than the other and if yes, select the better one. Based on the answer to Question 2, we also estimate if the better rationale adjudged by human evaluator aligns with likelihood-based rationale selection. We found that for 1) 72% cases, COLLATE's rationales are meaningful and helpful; 2) for 78.67% cases, one rationale is better than

Model	Parameter Scale	Maths GSM8K	NLI WinoGrande	PIQA	Commonsense HellaSwag	CSQA
OLMo (Groeneveld et al., 2024)	1B	11.89	31.31	25.98	68.02	48.19
w/ COLLATE (ours)	1B	**16.02**	**36.01**	**29.94**	**73.14**	**55.17**
Phi3 (Abdin et al., 2024)	3.8B	21.08	42.04	37.72	76.09	57.46
w/ COLLATE (ours)	3.8B	**28.29**	**45.28**	**42.35**	**81.67**	**64.14**
Qwen1.5 (Bai et al., 2023)	4B	19.17	39.66	34.57	80.03	56.21
w/ COLLATE (ours)	4B	**26.03**	**43.35**	**39.22**	**83.31**	**61.89**
Open-LLaMA-v2 (Geng and Liu, 2023)	7B	38.41	50.69	50.65	96.01	75.02
w/ COLLATE (ours)	7B	**53.48**	**53.49**	**59.57**	**99.21**	**80.13**
LLaMA3 (Dubey et al., 2024)	8B	60.68	48.21	49.35	90.35	80.39
w/ COLLATE (ours)	8B	**69.09**	**54.41**	**58.70**	**92.86**	**87.42**

Table 3: Evaluation of COLLATE on LLMs of varying scale of parameters (1B to 8B) and different LLMs (OLMo, Phi3, Qwen1.5, Open-llama-v2, Llama3). COLLATE yields significant gains across scales and models on all tasks.

the other indicating that rationales from the two rationale providers are distinct and diverse; and 3) in 84% cases, better rationale judged by humans aligns with likelihood-based rationale selection.

Qualitative Error Analysis: We discuss the errors made by COLLATE to understand the scope of further improvements. Consider the following **sample** from GSM8K - *"Elsa started the day with 40 marbles. At breakfast, she lost 3 marbles. At lunchtime, she gave her best friend Susie 5 marbles. In afternoon, Elsa's mom bought her a new bag with 12 marbles. Susie came back and gave Elsa twice as many marbles as she received at lunch. How many marbles did Elsa end the day with?"*. The **correct logic** would be - *"Elsa had 40 (marbles) - 3 - 5 = 32 marbles. Elsa then gained 12 marbles + 32 = 44 marbles. At dinner Elsa received 2 * 5 = 10 marbles from Susie. In total, Elsa had 44 marbles + 10 = 54"*. The rationale generated by the **first rationale provider** ($\mathcal{RP}_1$) is - *"Elsa started the day with 40 - 3 = 37 marbles. She gave Susie 5 + 12 = 17 marbles. Susie then gave Elsa 17 * 2 = 34 marbles. In total, Elsa ended the day with 37 + 17 + 34 = 88 marbles"*. The rationale generated by the **second rationale provider** ($\mathcal{RP}_2$) is - *"Elsa started the day with 40 marbles. At breakfast, she lost 3 marbles while playing (40 - 3). At lunchtime, she gave her best friend Susie 5 marbles (40 - 3 + 5). In the afternoon, Elsa's mom bought her a new bag with 12 marbles (40 - 3 + 5 + 12). Susie came back and gave Elsa twice as many marbles as she received at lunch (40 - 3 + 5 + 12 + 2*5)"*. This example demonstrates that the rationale generated by $\mathcal{RP}_2$ is almost correct except that while incorporating the fact that Elsa gave 5 marbles, it

uses a '+' sign instead of '-'. This indicates that the rationale from a rationale provider might only be slightly incorrect and minor refinement can make it correct. Thus, rationale refinement can be explored in future work. Further, employing more rationale providers can increase the chances of generating the correct rationale which can be used for DPO.

Additional Experiments: We treat number of rationale providers (RPs) as a hyper-parameter and show that employing 3 RPs improve accuracy further (Appendix G). We observe that rationales from COLLATE are effective through perplexity estimation (Appendix I). Appendix H discusses that COLLATE's computation load is shifted to train time and inference requires generating a single rationale.

4.4 Impact of Design Choices for COLLATE

We examine the effectiveness of following components (using open-llama-v2-7b backbone) - 1) Distinct Rationale Providers, 2) Task-guided Likelihood-based Rationale Selection, and 3) Sample Filtration for DPO. Table 4 shows the results where it can be seen that obtaining diverse rationales from distinct rationale providers gives significantly better performance than sampling-based decoding using a single model (**COLLATE vs. row 1**). To analyse the importance of using likelihood of generating the GT answer to rate a rationale, we leverage LLM-as-a-judge paradigm (using IFT version of backbone LLM) as an alternative where the LLM is prompted to rate rationales (**row 2**). Notable drop in performance than COLLATE indicates that prompt-based rating does not work at scale of small LMs. Likewise, filtration of samples for task-guided DPO conditioned on whether

Ablation ID	Distinct Rationale Providers	Likelihood-based Rationale Selection	DPO Sample Filtration	Maths GSM8K	NLI WinoGrande	PIQA	Comonsense HellaSwag	CSQA
1	No	Yes	Yes	38.22	40.61	45.17	94.15	72.45
2	Yes	No	Yes	47.16	44.62	53.17	95.27	76.39
3	Yes	Yes	No	42.11	42.75	48.25	94.36	73.18
4	Yes	No	No	40.26	41.73	48.52	92.11	72.13
5	No	No	No	39.13	41.44	47.41	93.26	71.77
COLLATE	Yes	Yes	Yes	**53.48**	**53.49**	**59.57**	**99.21**	**80.13**

Table 4: Ablation study showing importance of COLLATE's components - **A) Distinct Rationale Providers:** Sampling-based decoding is used for diverse outputs as an alternative; **B) Likelihood-based rationale selection**: LLM-as-a-judge is used to rate rationales; **C) Sample filtration**: Entire train set is used for DPO w/o filtration.

Model	Maths GSM8K	NLI WinoGrande	PIQA	Comonsense HellaSwag	CSQA
GPT-4o (3-shot)	77.84	88.47	83.38	95.49	92.97
LLaMA3-8B (Dubey et al., 2024)	60.68 (-17.16)	48.21 (-40.26)	49.35 (-34.03)	90.35 (-5.14)	80.39 (-12.58)
w/ COLLATE (ours)	**69.09** (-8.75)	**54.41** (-34.06)	**58.70** (-24.68)	**92.86** (-2.63)	**87.42** (-5.55)

Table 5: Comparison of 3-shot evaluation of GPT-4o with LLaMA3-8B with and without COLLATE. We see a significant reduction in the performance gap between GPT-4o and LLaMa3-8B when used with COLLATE.

best rationale enhances likelihood of generating GT (than not using any rationale) is critical for gains achieved by COLLATE (**vs. row 3**). Omitting both likelihood-based rationale selection and sample filtration (**row 4**) as well as all the three components (**row 5**) gives significantly lower performance.

4.5 Bridging Accuracy Gap between Smaller Models and GPT-4o using COLLATE

We evaluate the performance of GPT-4o for different tasks (using 3-shot prompting). We then compare the accuracy gap of Llama3-8B from GPT-4o and report reduction in this gap using rationales from COLLATE (with Llama3-8B) in Table 5. For GSM8K, the performance gap of Llama3-8B with Gpt-4o reduces from 17.16% to 8.75% which is a significant relative reduction of 49%. Likewise, it reduces from 40.26% to 34.06% (relative reduction of 15%) on WinoGrande, 34% to 24% (relative reduction of 27.48%) on PIQA, 5% to 2.63% on HellaSwag, and from 12.58% to 5.55% on CSQA.

5 Conclusion

Step-by-step rationales generated by large LMs have been commonly used to improve smaller LMs. Such large-scale (often closed) LLMs cannot be used to train other models in commercial setting owing to legal constraints. Little focus has been given to enhancing rationale generation ability of small LMs. We propose COLLATE which optimises an LLM to selectively generate better ratio-

nales from a pool of diverse candidates produced by distinct instances of the same LLM. Rationale candidates are ranked for preference tuning based on end-task-guided likelihood score. COLLATE outperforms multiple baselines on diverse tasks without relying on larger LMs to generate and rate rationales. It provides significant accuracy gains for LMs of different parameter-scales and families.

6 Limitations

To obtain the distinct rationale providers, currently we randomly divide the training set into as many equal splits as the number of distinct rationale providers. However, it could be explored if the samples in the train dataset could be allocated to each rationale provider in a way such that each rationale provider gains expertise in certain task domain(s) compared to the other rationale providers. This way, the different rationale providers would automatically become specialized experts for different types of problems and tasks. This can be used to obtain insights into which type of experts are required to improve rationales for different task domains. Additionally, the number of distinct rationale providers to be used a hyper-parameter. We experiment with two rationale providers in the main paper and also show that employing 3 rationale providers improve the accuracy gains further (in Appendix G, Table 10). However, it could be explored if the optimal number of rationale providers could be determined automatically for a given task.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Shizhe Diao, Pengcheng Wang, LIN Yong, Rui Pan, Xiang Liu, and Tong Zhang. 2024. Active prompting with chain-of-thought for large language models.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Arnav Gudibande, Eric Wallace, Charlie Victor Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2024. The false promise of imitating proprietary language models. In *The Twelfth International Conference on Learning Representations*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

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. 2023a. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023c. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3134–3154, Singapore. Association for Computational Linguistics.

Gurusha Juneja, Subhabrata Dutta, Soumen Chakrabarti, Sunny Manchanda, and Tanmoy Chakraborty. 2023. Small language models fine-tuned to coordinate larger language models improve complex reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3691, Singapore. Association for Computational Linguistics.

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. 2023. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.14045*.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*.

Xiaonan Li and Xipeng Qiu. 2023a. Finding supporting examples for in-context learning. *arXiv preprint arXiv:2302.13539*, 3.

Xiaonan Li and Xipeng Qiu. 2023b. Mot: Pre-thinking and recalling enable chatgpt to self-improve with memory-of-thoughts. *arXiv preprint arXiv:2305.05181*, 16.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024a. Routing to the expert: Efficient reward-guided ensemble of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1964–1974, Mexico City, Mexico. Association for Computational Linguistics.

Xiaoding Lu, Adian Liusie, Vyas Raina, Yuwen Zhang, and William Beauchamp. 2024b. Blending is all you need: Cheaper, better alternative to trillion-parameters llm. *arXiv preprint arXiv:2401.02994*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.

MosaicML. 2023. Dolly-hhrlhf dataset.

Sajad Mousavi, Ricardo Luna Gutiérrez, Desik Rengarajan, Vineet Gundecha, Ashwin Ramesh Babu, Avisek Naug, Antonio Guillen, and Soumyendu Sarkar. 2023. N-critics: Self-refinement of large language models with ensemble of critics. *arXiv preprint arXiv:2310.18679*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza

Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *Preprint*, arXiv:2306.01116.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Swarnadeep Saha, Peter Hase, and Mohit Bansal. 2023. Can language models teach? teacher explanations improve student performance via personalization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

Chenglei Si, Weijia Shi, Chen Zhao, Luke Zettlemoyer, and Jordan Boyd-Graber. 2023. Getting MoRE out of mixture of language model reasoning experts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8234–8249, Singapore. Association for Computational Linguistics.

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. *arXiv preprint arXiv:2310.12397*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Moly-

bog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Dennis Ulmer, Elman Mansimov, Kaixiang Lin, Justin Sun, Xibin Gao, and Yi Zhang. 2024. Bootstrapping llm-based task-oriented dialogue agents via self-talk. *arXiv preprint arXiv:2401.05033*.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Danqing Wang and Lei Li. 2023. Learning from mistakes via cooperative study assistant for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10667–10685, Singapore. Association for Computational Linguistics.

Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. 2023a. Boosting language models reasoning with chain-of-knowledge prompting. *arXiv preprint arXiv:2306.06427*.

Tianduo Wang, Shichen Li, and Wei Lu. 2024. Self-training with direct preference optimization improves chain-of-thought reasoning. *Preprint*, arXiv:2407.18248.

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

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. 2023. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15135–15153, Singapore. Association for Computational Linguistics.

Yue Yu, Jiaming Shen, Tianqi Liu, Zhen Qin, Jing Nathan Yan, Jialu Liu, Chao Zhang, and Michael Bendersky. 2024. Explanation-aware soft ensemble empowers large language model in-context learning.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *Preprint*, arXiv:2401.10020.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. 2024. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *arXiv preprint arXiv:2406.09136*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Qualitative Analysis and Comparison of Rationales

We discussed the qualitative comparison of rationales generated by COLLATE with the best baseline in the section - 'Comparison of COLLATE with Baselines' (Table 2) in the main paper for one dataset from each task domain. Table 6 shows more examples comparing the quality of rationales across all the datasets for different task domains.

B Prompt Templates for Multi-Mode Instruction Fine-Tuning

As discussed in the Methodology section in the main paper, the base model $\mathcal{M}$ is instruction fine-tuned to enable the LLM to operate in three modes - (i) generate the final answer given the instruction as input ($\mathcal{I} \to \mathcal{A}$); (ii) generate the rationale given the instruction as input ($\mathcal{I} \to \mathcal{R}$); and (iii) generate the answer conditioned on the instruction and rationale as input ($[\mathcal{I}; \mathcal{R}] \to \mathcal{A}$). The inputs to the LLM are formatted using corresponding prompts ($\mathcal{P}_{\mathcal{I} \to \mathcal{A}}$; $\mathcal{P}_{\mathcal{I} \to \mathcal{R}}$; $\mathcal{P}_{[\mathcal{I}; \mathcal{R}] \to \mathcal{A}}$) for each of these modes so that the LLM can generate an appropriate output accordingly. The textual instruction for each prompt template is specified as follows:

1. $\mathcal{P}_{\mathcal{I} \to \mathcal{A}}$ = "You are an AI assistant 'M'. Provide a response to the given instruction denoted by Task Description.

   [TASK DESCRIPTION STARTS]
   ⟨Task Description⟩: In this task, you will be given an 'Instruction'. Generate the correct answer for the given instruction.
   'Instruction' - ⟨instruction⟩
   [TASK DESCRIPTION ENDS]

   For the given ⟨Task Description⟩, give your response. [M RESPONSE BEGINS]: "

2. $\mathcal{P}_{\mathcal{I} \to \mathcal{R}}$ = "You are an AI assistant 'M'. Provide a response to the given instruction denoted by Task Description.

   [TASK DESCRIPTION STARTS]
   ⟨Task Description⟩: In this task, you will be given an 'Instruction'. Generate descriptive reasoning on how to derive the correct answer for the instruction such that the descriptive reasoning will be useful to another AI assistant to generate the correct answer.
   'Instruction' - ⟨instruction⟩
   [TASK DESCRIPTION ENDS]

   For the given ⟨Task Description⟩, give your response. [M RESPONSE BEGINS]: "

Dataset	Task-Instruction and Rationales Generated by Best Baseline (SPIN) and Rationale Providers - $\mathcal{RP}_1$ and $\mathcal{RP}_2$ in COLLATE
GSM8K	**Instruction:** I have 10 liters of orange drink that are two-thirds water and I wish to add it to 15 liters of pineapple drink that is three-fifths water. But as I pour it, I spill one liter of the orange drink. How much water is in the remaining 24 liters? **Baseline:** The orange drink is two-thirds water, so 10 liters would have 6.5 liters of water. The pineapple drink is three-fifths water, so 15 liters would have 9 liters of water. Since 1 liter of the orange drink is spilled, only 5.5 liters of water from the orange drink remains. Adding it to the 9 liters from the pineapple drink gives a total of 14.5 liters of water in the remaining 24 liters. ✗ **Rationale Provider 1:** The orange drink is two-thirds water. When 1 liter is spilled, the remaining 9 liters of orange drink contains 2/3x9=6 liters of water. The pineapple drink is three-fifths water, so 15 liters contains 9 liters of water. The total water content in 24 liters is 6+9=15 liters of water. ✓ **Rationale Provider 2:** Since the orange drink is two-thirds water, spilling 1 liter reduces the water content by 1 liter, leaving 2/3 x 9 liters of water. The pineapple drink has 9 liters of water. Therefore, the total amount of water should be 9+9=18 liters of water in the remaining 24 liters. ✗
WinoGrande	**Instruction:** Terry tried to bake the eggplant in the toaster oven but the _____ was too big. A. eggplant, B. toaster **Baseline:** The eggplant didn't fit properly in the toaster. Therefore, the correct answer seems to be "toaster". ✗ **Rationale Provider 1:** The issue described is that the eggplant didn't fit in the toaster. Therefore, the correct answer is "eggplant" because it was too big for the toaster.✓ **Rationale Provider 2:** Since the toaster is the appliance being used, it might have been too large to handle for baking the eggplant. So, the correct answer could be "toaster". ✗
PIQA	**Instruction:** To make a bedroom temperature colder without a fan or air conditioner, sol1: hang up a thick, and dark blanket over the windows and then ensure their is proper air flow in the room. sol2: wave a large blanket up and down to generate a gust of wind to cool the room down when it gets too hot. **Baseline:** Waving a large blanket up and down would create a gust of wind, which might help cool the room down quickly. Therefore, sol 2 is correct. ✗ **Rationale Provider 1:** Generating wind by waving a blanket can circulate air and help cool the room, especially when it gets too hot. Therefore, sol 2 is likely the more effective method for cooling the room. ✗ **Rationale Provider 2:** To cool down a room, blocking out sunlight and reducing heat entry is essential. Hanging a thick, dark blanket over the windows prevents heat from entering, which helps keep the room cooler. Therefore, sol 1 the correct answer. ✓
HellaSwag	**Instruction:** Then he takes a small stone from the flowing river and smashes it on another stone. He starts to crush the small stone to smaller pieces. He _____. A. cuts the center stone in half and blow it on to make it bigger. B. grind it hard to make the pieces smaller, C. eventually brings it back into view and adds it to the smaller ones to make a small triangular shaped piece, D. starts to party with them and throw the pieces by hand while they celebrate. **Baseline:** ✗ Once the small stone is crushed, we need to start to party with them and throw the pieces by hand while they celebrate. This is an unexpected, yet possible action, making option D correct. ✗ **Rationale Provider 1:** After crushing the stone into smaller pieces, one should grind it hard to make the pieces smaller. So, option B is the correct answer. ✓ **Rationale Provider 2:** After crushing the stone into smaller pieces, it makes sense to cut the center stone in half and blow it on to make it bigger. So option A seems correct. ✗
CSQA	**Instruction:** If I was getting drunk, and people couldn't understand me, what might I be having? A. a seizure, B. slurred speech, C. death, D. forgetfulness, E. pass out **Baseline:** Getting drunk could lead to serious consequences like "death" if things get out of hand. If people can't understand you, it could be a sign that something is seriously wrong, making option C correct. ✗ **Rationale Provider 1:** EWhen you're getting drunk, the most common reason people can't understand you is "slurred speech." Alcohol often impairs speech clarity, making option B the correct answer. ✓ **Rationale Provider 2:** If people can't understand you while you're getting drunk, it might be because you're experiencing "forgetfulness." Alcohol can affect memory, so you might be forgetting words or how to speak clearly. Therefore, option D seems correct. ✗

Table 6: Qualitative comparison of rationales generated by best baseline (SPIN) and rationale providers in COLLATE on all datasets from each task domain. COLLATE generates better rationales by virtue of employing distinct rationale providers.

3. $\mathcal{P}_{[\mathcal{I};\mathcal{R}]\to\mathcal{A}}$ = "You are an AI assistant 'M'. Provide a response to the given instruction denoted by Task Description.

[TASK DESCRIPTION STARTS]
⟨Task Description⟩: In this task, you will be given an 'Instruction' and a rationale denoted by 'Rationale'. Analyse the rationale and come up with the correct answer for the given instruction.
'Instruction' - ⟨instruction⟩
'Rationale' - ⟨rationale⟩
[TASK DESCRIPTION ENDS]

For the given ⟨Task Description⟩, give your response. [M RESPONSE BEGINS]: "

In the above prompt templates, ⟨instruction⟩ is a placeholder for the actual task instruction $\mathcal{I}^{\mathcal{T}}$ and ⟨rationale⟩ is a placeholder for the rationale text.

C Direct Preference Optimisation (DPO)

Direct Preference Optimisation (DPO) (Rafailov et al., 2023) was introduced as an alternative to Reinforcement Learning using Human Feedback (RLHF) (Ouyang et al., 2022) technique to alleviate the need of training a reward model. RLHF depends on training a reward model to assign a score to the outputs generated by an LLM to fine-tune

Dataset	Task-Instruction and Ground-Truth Answers for Each Dataset used to evaluate COLLATE
GSM8K	**Instruction:** Betty is saving money for a new wallet which costs $100. Betty has only half of the money she needs. Her parents decided to give her $15 for that purpose, and her grandparents twice as much as her parents. How much more money does Betty need to buy the wallet? **Ground-Truth Response:** 5
WinoGrande	**Instruction:** Terry tried to bake the eggplant in the toaster oven but the _____ was too big. A. eggplant, B. toaster **Ground-Truth Response:** A. eggplant
PIQA	**Instruction:** How to dry flowers? sol1: Find a dark, moist area with good circulation, such as an attic or unused closet. With unflavored dental floss, secure the bottom of the flowers' stems to a hanger so that they hang upside down to dry. Leave flowers for two to three weeks until completely dry, sol2: Find a dark, dry area with good circulation, such as an attic or unused closet. With unflavored dental floss, secure the bottom of the flowers' stems to a hanger so that they hang upside down to dry. Leave flowers for two to three weeks until completely dry. **Ground-Truth Response:** sol2
HellaSwag	**Instruction:** Then he takes a small stone from the flowing river and smashes it on another stone. He starts to crush the small stone to smaller pieces. He _____. A. cuts the center stone in half and blow it on to make it bigger. B. grind it hard to make the pieces smaller, C. eventually brings it back into view and adds it to the smaller ones to make a small triangular shaped piece, D. starts to party with them and throw the pieces by hand while they celebrate. **Ground-Truth Response:** B
CSQA	**Instruction:** When learning about the world and different cultures, what is important if you are committed to eliminating preconceived notions. A. newness, B. loss of innocence, C. enlightenment, D. open mind, E. smartness **Ground-Truth Response:** D. open mind

Table 7: Examples of instructions from different datasets belonging to diverse task domains used in the experiments - (i) Maths Problem Solving (GSM8K), (ii) Natural Language Inference (WinoGrande and PIQA), and (iii) Commonsense Reasoning (HellaSwag and CSQA).

the LLM through reinforcement learning to align it with human preferences. On the other hand, DPO transforms the loss over the reward-function to a loss over the LLM policy such that the reward is optimised implicitly by optimising the loss over the policy. It does so by leveraging human preference data which compares two possible outputs generated by an LLM such that the better output is considered as the winner candidate - y_w while the inferior output is considered as the loser candidate - y_l. Given a static dataset of the form $\mathcal{D} = \{x, y_w, y_l\}$, where x is the input, the loss is modeled as -

$$\mathcal{L}_R = -log[\sigma(r(x, y_w) - r(x, y_l))] \quad (10)$$

$$r(x, y) = \beta log(\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}) \quad (11)$$

where, $\pi_{\mathcal{Z}}(y|x)$ is the likelihood of generating y given x as input to the model $\mathcal{Z} \in \{\mathcal{M}_{ref}, \mathcal{M}_\theta\}$, $\mathcal{M}_{ref}$ is usually taken to be the instruction fine-tuned model in the case of an LLM to prevent the LLM policy from deviating too much from the initial policy, $\mathcal{M}_\theta$ represents the LLM policy being optimised through DPO, σ is the sigmoid activation, and β is a coefficient that controls the amount of deviation from the reference model. In summary, the algorithm optimises the LLM to learn to prefer generating certain outputs over other candidates without requiring an explicit reward model. Please refer to the original publication (Rafailov et al., 2023) for an elaborate discussion of the details.

D Dataset Samples

Details of datasets were discussed in the 'Experiments and Evaluation' section in the main paper. Table 7 in the appendix shows samples of instructions for each dataset from all task domains - (i) Maths Problem Solving (GSM8K), (ii) Natural Language Inference (WinoGrande and PIQA), and (iii) Commonsense Reasoning (HellaSwag and CSQA).

E Few-shot evaluation of COLLATE

The few-shot evaluation results in Table 8 demonstrate that COLLATE significantly improves performance across tasks. On GSM8K, the base model rose from 52.16 to 54.49 (+2.33), and the instruct model from 76.12 to 81.26 (+5.14). For NLI, COLLATE boosted WinoGrande by +1.94 (base) and +2.71 (instruct), and PIQA by +3.86 (base) and +3.63 (instruct). In commonsense reasoning, HellaSwag improved by +2.95 (base) and +3.94 (instruct), while CSQA saw smaller gains of +3.08 (base) and +1.45 (instruct). These results highlight COLLATE's consistent effectiveness in enhancing few-shot performance across diverse benchmarks.

F Performance of Rationale Providers across Iterations

Table 9 shows the performance analysis of rationale provider (selected for task-specific SFT-based evaluation) across different iterations of DPO training using open-llama-v2-7b backbone. It can be

Model	Maths GSM8K	NLI WinoGrande	PIQA	Comonsense HellaSwag	CSQA
Llama-3-8B-Base	52.16	79.43	73.16	69.21	60.07
Llama-3-8B-Base w COLLATE	54.49	81.37	77.02	72.16	63.15
Llama-3-8B-Instruct	76.12	78.51	72.06	76.25	57.69
Llama-3-8B-Instruct w COLLATE	81.26	81.22	75.69	80.19	59.14

Table 8: Few-shot evaluation of LLaMa-3-8B models (Base and Instruct) on Maths (GSM8K), NLI (WinoGrande, PIQA), and Commonsense reasoning (HellaSwag, CSQA). Results show that COLLATE outperforms corresponding baseline, with notable gains in both NLI and Commonsense tasks.

DPO Iteration	Maths GSM8K	NLI WinoGrande	PIQA	Comonsense HellaSwag	CSQA
Iteration 0	47.95	50.86	49.84	98.93	78.74
Iteration 1	53.41	51.65	55.33	99.21	80.11
Iteration 2	**53.48**	**53.49**	**59.57**	**99.21**	**80.13**

Table 9: Performance analysis of rationale provider (selected for task-specific SFT-based evaluation) across different iterations of DPO training using open-llama-v2-7b backbone. It can be seen that performance increases with iterations for all the tasks.

seen that performance increases with iterations for all the tasks.

G More Rationale Providers (RPs) improve accuracy further

Table 10 shows that using 3 RPs improves performance over the configuration with 2 RPs across GSM8K and PIQA. Specifically, on GSM8K, COLLATE with 2 RPs achieves an accuracy of 53.58, while using 3 RPs boosts this to 55.24, a gain of 1.66%. Similarly, for PIQA, the accuracy improves from 59.57 (with 2 RPs) to 61.73 (with 3 RPs), reflecting a 2.16% increase. These results demonstrate that increasing the number of RPs further enhances the accuracy, showing that COLLATE benefits from the diversity and richness of additional rationales. This improvement over baselines is further amplified when using 3 RPs, highlighting the efficacy of COLLATE.

H Computational Complexity - Average number of Rationale tokens

Table 11 highlights the average number of rationale tokens generated during inference for COLLATE and baseline methods. COLLATE maintains comparable inference-time efficiency, with no significant increase in rationale tokens compared to baselines. For example, COLLATE generates an average of 33.3 tokens per rationale, similar to

Method	GSM8K	PIQA
COLLATE w 2 RPs	53.58	59.57
COLLATE w 3 RPs	55.24	61.73

Table 10: Impact of Using Multiple Rationale Providers (RPs) in COLLATE. This table compares the performance of COLLATE with 2 and 3 rationale providers (RPs) on the GSM8K and PIQA benchmarks, demonstrating that using 3 RPs leads to further improvements in accuracy for the open-llama-v2-7B backbone.

baselines like CoT (24.5 tokens) and SPIN (29.8 tokens). EoT, being a prompt-based method employs multiple LLMs during inference, thereby shooting the average number of rationale tokens to 83.4

The efficiency of COLLATE is due to its design, which shifts the computational overhead to the training phase—a one-time process. During training, multiple rationale providers (RPs) are used to generate diverse rationales for DPO, ensuring high-quality rationale generation. However, at inference time, COLLATE employs only a single RP to generate a single rationale, minimizing additional computation. This contrasts with prompt-based baselines, which require generating multiple rationales during inference, leading to higher computational costs. By reducing inference-time overhead, COLLATE is particularly well-suited for practical applications where computational re-

Model	Maths	NLI		Comonsense	
	GSM8K	WinoGrande	PIQA	HellaSwag	CSQA
CoT	35.1	18.5	17.7	28.4	22.9
EoT	121.5	66.1	69.4	75.9	84.1
Distilling Step-by-Step	36.1	19.0	18.4	28.1	23.1
SPIN	43.2	22.7	22.2	33.3	27.5
COLLATE	48.1	25.7	24.9	37.1	30.8

Table 11: Comparison of Average Number of Rationale Tokens Generated During Inference. COLLATE demonstrates no significant increase in inference-time overhead compared to baselines, as its computational cost is primarily shifted to the training phase where diverse rationales are generated using multiple rationale providers (RPs) for DPO. In contrast, prompt-based baselines incur higher inference costs as they require generating multiple rationales during inference.

Model	Maths	NLI		Comonsense	
	GSM8K	WinoGrande	PIQA	HellaSwag	CSQA
SFT w/o rationales	12.93	3.39	5.86	8.4	9.77
SFT w EoT rationales	12.61	3.28	5.59	7.92	9.04
SFT w Distilling Step-by-Step	11.47	3.15	5.36	7.33	8.48
SFT w SPIN rationales	10.28	3.10	5.27	7.17	8.09
SFT w COLLATE rationales	8.03	2.53	4.79	6.26	7.72

Table 12: Perplexity Comparison of COLLATE and Baseline Methods for generating ground-truth (GT) answers conditioned on rationales in the input, as evaluated by task-specific SFT models (Llama-3-8B). COLLATE achieves the lowest perplexity across all tasks, demonstrating its effectiveness in providing rationales that enhance the model's certainty in generating correct answers.

sources and latency are critical. This efficiency, coupled with its robust performance, makes COLLATE a better alternative for deploying models in real-world settings.

I Additional Evaluation of Rationales from COLLATE

As an additional evaluation metric, we estimate the perplexity of generating ground-truth (GT) answers conditioned on rationales in the input for test samples. This assessment was conducted using task-specific SFT models (open-llama-v2-7B), and Table 12 compares COLLATE against the best-performing baseline from each category.

The results indicate that COLLATE achieves the lowest perplexity across all tasks, highlighting its ability to produce rationales that better guide the model towards correct answers. For instance, on GSM8K, COLLATE achieves a perplexity of 8.03, compared to 10.28 for the best baseline (SPIN). Similarly, on CSQA and HellaSwag, COLLATE records perplexities of 7.72 and 6.26 respectively, outperforming SPIN with scores of 8.09 and 7.17.

This improvement underscores the effectiveness of COLLATE rationales in reducing model uncertainty, making it superior to other methods. The results reaffirm COLLATE's ability to provide rationales that not only enhance model performance but also ensure higher confidence in the answers, as evidenced by consistently lower perplexity scores.

J Selection of Rationale Provider for Evaluation

We use a single rationale provider while assessing rationales generated by COLLATE. The best rationale provider is selected based on likelihood estimation on CoT val-split samples. We measure probability of generating ground-truth answer by the IFT model $\mathcal{M}_{IFT}$ conditioned on the rationale. The rationale provider that yields maximum value (averaged across samples) is used for evaluation. Figure 2 shows correlation between test-accuracy and likelihood indicating that rationale provider with best likelihood is a suitable choice.
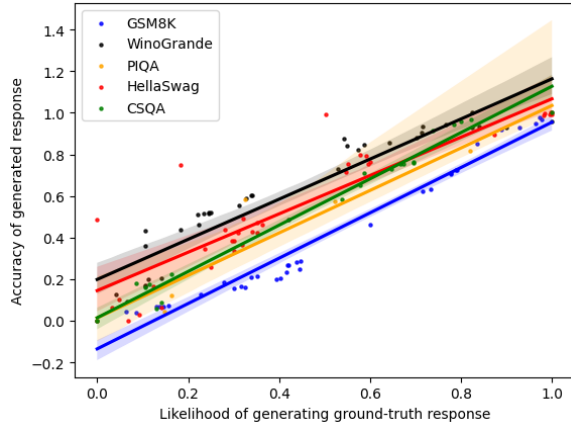
Figure 2: Normalized plot between task test accuracy vs. likelihood of generating GT on CoT val-split. Positive correlation indicates that rationale provider selected based on better likelihood is suitable choice to generate rationales for task-specific SFT-based evaluation.

K Human Study for Rationale Evaluation

We conducted a human study to evaluate the effectiveness of rationales obtained using the proposed COLLATE framework. The following steps describe creation of data for human evaluation:

Dataset Creation for Human Evaluation

1. We collected a total of 75 samples based on random selection by taking an equal number of samples for each task i.e. 15 samples randomly from the test sets of each of the 5 task datasets.

2. For each sample, we obtain the rationales $R1_g$, $R2_g$ from the two LLM rationale providers.

Once the above rationales are obtained, we employed three paid human annotators and presented them with the instruction in each sample along with different rationales obtained above. The human evaluators are asked to judge the quality of different rationales based on the following questions and guidelines:

Questions and Guidelines

1. Question 1: Is the final rationale obtained from COLLATE useful for answering the question correctly? The rationale is useful if it is correct and provides the correct explanation on how the answer for the instruction in the sample should be derived. Provide a label out of 0 or 1 such that 0 means that the final rationale is totally wrong; and 1 means that the final rationale is totally correct.

2. Question 2: Compare the two rationales obtained using the two rationale providers - $R1_g$ and $R2_g$. Provide a label of 0 or 1 where 0 means that none of the rationales is better than the other and 1 means that one rationale is better than the other.

3. Question 3: In Question 2, in case one rationale is better than the other (between the rationales obtained from the two rationale providers), select the better rationale.

Definition of Metrics Estimated from Human Labels

Different rationales were presented to human evaluators in jumbled order to avoid biases while comparing rationales. Based on the judgement labels provided by the human evaluators for 3 questions above for the 75 samples, we estimate the following metrics:

1. Final Rationale Alignment – % proportion of samples which were assigned label 1 i.e. totally correct.

2. Diversity b/w two Rationales - % proportion of samples where the two rationales $R1_g$ and $R2_g$ obtained from two rationale providers are different i.e. cases where one of the two rationales is better than the other (label 1). This metric is estimated to verify if the variants truly generate distinct rationales.

3. Better Rationale Alignment with Likelihood based Selection: We consider samples where label 1 is provided to Question 2 i.e. one of the generated rationales is judged better than the other generated rationale (comparing $R1_g$ and $R2_g$). We estimate the metric as % proportion cases from these samples where better rationale determined using likelihood-based utility score matches the better rationale from human judgement.

Human Study Results and Discussion

We compute the above metrics using the 75 samples used for human evaluation. We report the average of metrics obtained for the three human evaluators in Table 13. We discuss following observations from the results in Table 13:

1. From Table 13, we can observe that the final rationale alignment is 72.0% which means that final rationale obtained from COLLATE is reliable and aligns with human preferences.

Metric Name	Value (in %)
Final Rationale Alignment	72.0
Diversity b/w two Rationales	78.67
Better Rationale Alignment with Likelihood-based Selection	84.0

Table 13: Human study results summarizing values of different metrics evaluated using human labels. It is observed that for good proportion of cases, final rationale obtained from COLLATE aligns with human preferences, the rationales obtained from two variants are diverse and better rationale judged by humans matches with winner rationale selected using likelihood based utility score.

2. Rationales from the two rationale providers are diverse: It is observed that for 78.67% cases, one rationale obtained was judged to be better than the other generated rationale. This means that employing two variants of same LLM is useful to obtain distinct and diverse rationales which are useful to improve quality of preference data for DPO.

3. Likelihood based rationale selection aligns with human preferences: For 84.0% cases, better generated rationale determined based on human preferences matches the better rationale based on likelihood-based utility score. This shows that our choice of using likelihood of final GT answer for selecting winner rationale aligns with human preferences and is suitable to obtain the preference data.

Inter-Annotator Agreement: We also report the inter-annotator agreement by estimating the fleiss' kappa coefficient which is commonly used to measure agreement between three annotators. For the human study, the fleiss-kappa coefficient for inter-annotator agreement came out to be: 0.5829.

Following is mapping of fleiss-kappa coefficient value ranges with interpretation:

0 – 0.2: Slight agreement
0.21 - 0.4: Fair agreement
0.41 - 0.6: Moderate agreement
0.61 - 0.8: Substantial agreement
0.81 - 1.0: Almost Perfect agreement

Based on the coefficient obtained for different metrics and the above scale, it can be seen that human labels have moderate to good agreement.

L   Additional Related Work

Differently, some works use mixture-of-experts where the instruction is routed to suitable LLM expert either at the query-level (Lu et al., 2024a), in latent space (Jiang et al., 2024) or at the output-layer (Si et al., 2023).

M   Justification for Splitting Data Randomly to Train Rationale Providers

We intentionally divided the dataset into random splits for training and obtaining distinct rationale providers because of the consideration that if a rationale provider is trained on samples from a specific task, then it can become an expert in that task only and might not be able to generalize and generate useful rationale candidates for other tasks. However, obtaining diverse, good-quality and competing rationale candidates is critical to obtain a good rationales set for improving the rationale providers for a given task through preference tuning. Hence, by design, we divide the dataset randomly so as to prevent the rationale providers become very specific to a particular domain or task. However, we do agree that the effect of using domain-specialized rationale providers could be explored as future work.

N   Comparison of COLLATE with SPIN on 3 additional datasets

To further demonstrate the robustness and generalizability of our proposed approach COLLATE, we conducted evaluations on three additional, challenging datasets: **AIME** (a more recent and difficult benchmark for mathematical reasoning), **CROW** (common sense reasoning), and **HotpotQA** (long chain multi-hop reasoning). Evaluation on these datasets provide further empirical evidence for COLLATE's performance improvements over the strongest baseline, SPIN. These datasets complement the already diverse set of benchmarks used in the main paper, extending the coverage across: **1) Mathematical Reasoning:** AIME focuses on complex math word problems. **2) Commonsense Reasoning:** CROW requires applying background knowledge to infer correct outcomes. **3) Multi-hop QA:** HotpotQA evaluates long chain reasoning and contextual grounding.

The results are summarized in Table 14, where

Table 14: Performance comparison of COLLATE with SPIN on additional benchmarks.

Method	AIME	CROW	HotpotQA
SPIN	16.02	68.29	62.39
COLLATE (ours)	**19.72**	**72.18**	**64.11**

we observe that COLLATE consistently outperforms SPIN across all three benchmarks, with an absolute improvement of **3.7%** on AIME, **4%** on CROW, and **1.7%** on HotpotQA. These results strengthen our claim that COLLATE is a general-purpose reasoning enhancement approach that scales well across diverse and difficult reasoning tasks.