Explanation-aware Soft Ensemble Empowers Large Language Model In-context Learning

Yue Yu[♠*], **Jiaming Shen**[♣], **Tianqi Liu**[♣], **Zhen Qin**[♣], **Jing Nathan Yan**[♢*], **Jialu Liu**[♣], **Chao Zhang**[♠], **Michael Bendersky**[♣]

[♠] Georgia Institute of Technology [♣] Google [♢] Cornell University

Abstract

Large language models (LLMs) have shown remarkable capabilities in various natural language understanding tasks with a few demonstration examples via in-context learning. Common strategies to boost such "in-context" learning ability are to ensemble multiple model decoded results and require the model to generate an explanation along with the prediction. However, these models often treat different class predictions equally and neglect the potential discrepancy between the explanations and predictions. To fully unleash the power of explanations, we propose EASE, an *Explanation-Aware Soft Ensemble* framework to empower in-context learning with LLMs. We design two techniques, explanation-guided ensemble, and soft probability aggregation, to mitigate the effect of unreliable explanations and improve the consistency between explanations and final predictions. Experiments on 10 NLU tasks and 4 varying-size LLMs demonstrate the effectiveness of our framework.

1 Introduction

Recent advancements in Natural Language Processing (NLP) have witnessed the remarkable capabilities of Large Language Models (LLMs) (Brown et al., 2020; Anil et al., 2023; Touvron et al., 2023; OpenAI, 2023). These LLMs can rapidly adapt to new tasks by learning only on a few input-output pairs (*a.k.a.* demonstrations) in context (Wei et al., 2022a). Yet, beyond those demonstrations, a significant facet of human learning revolves around explanations. These explanations[1], typically in the form of a few keywords or sentences, reveal the underlying principles connecting the input and output (Zaidan et al., 2007; Narang et al., 2020). Consequently, integrating free-text explanations into LLM prompting holds great potentials to further enhance in-context learning performance.

Recent studies have examined how to incorporate free-text explanations into LLM in-context learning scheme. For instance, the *Predict-then-Explain* pipeline (Lampinen et al., 2022) proposes to generate the explanation *after* making the prediction. Consequently, the predictions from LLM won't directly benefit from their corresponding explanations. In contrast, the *Explain-then-Predict* pipeline (*a.k.a* "Chain-of-Thought") (Wei et al., 2022b) generates explanations *before* making predictions via greedy sampling. When the LLM-generated explanations are unreliable, predictions from this approach will be largely distracted and defective (Ye and Durrett, 2022; Turpin et al., 2023). To mitigate this issue, (Wang et al., 2023d) improves the "Chain-of-Thought" pipeline by generating multiple predictions with different explanations using temperature sampling and aggregating them via majority voting. However, this approach can be sub-optimal as (1) temperature sampling *increases the inconsistency* between generated explanations and their associated class predictions, and (2) majority voting treats predictions associated with explanations of varying qualities *equally*. How to robustly leverage free-text explanations for LLM in-context learning remains challenging.

In this work, we present an **E**xplanation-**a**ware **S**oft **E**nsemble framework, named EASE, to assist LLM in-context learning with explanations. Our technique integrates explanations into the ensemble procedure and employs soft probability to mitigate discrepancies between explanations and predictions. The key module of the EASE framework hinges upon the idea of weighted ensemble: As shown in Figure 1, instead of treating all predictions equally, we assign a score to each prediction based on the contextual relevance and inherent quality of its associated explanation, which will be used as a weight during the final ensemble stage.

[*] Work done during the internship at Google. E-mail: yueyu@gatech.edu

[1] In this paper, we use the term 'explanations' and 'rationales' interchangeably.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14002–14024

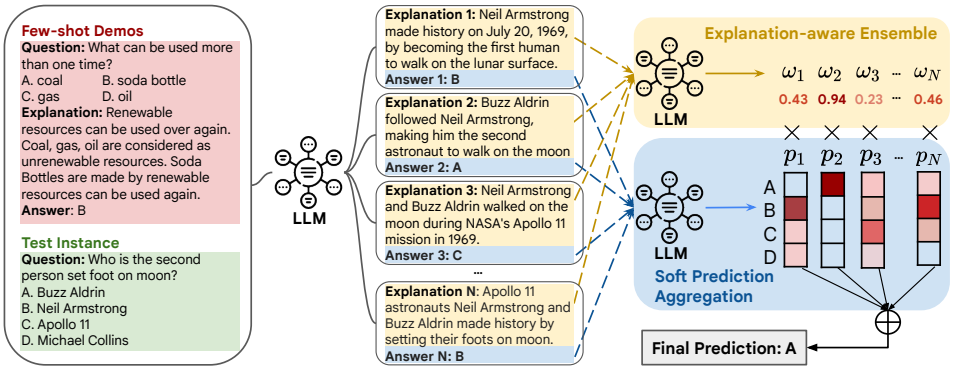


Figure 1: The overview of EASE framework, where we introduce two strategies to better harness explanations for LLM in-context learning.

While several works have studied scoring the LLM-generated text (Wang et al., 2023b; Khalifa et al., 2023; Jiang et al., 2023), these models typically require massive training labels on downstream tasks or intermediate reasoning annotations. To facilitate explanation scoring with under the few-shot setting studied in this work, we realize this explanation-aware ensemble stage with an LLM — after generating explanations and predictions using temperature sampling for each test instance, we prompt the LLM to weight all class predictions based on their associated explanations in an in-context manner. While the LLM offers great promise for the weighting purpose, it is crucial to provide sufficient *supervision signals* as demonstrations to guide the LLM scoring, yet the primary constraint for this step lies in the absence of *negative* explanations from few-shot demonstrations. To construct negative examples, we first use LLM to generate explanations for few-shot demonstrations, then select explanations associated with *incorrect predictions* as the negative samples. In this way, the LLM scorer can be readily applied to perform explanation-aware ensembling without any additional annotation.

Beyond explanation-aware ensembling, EASE incorporates an additional technique named *soft probability aggregation*, which helps to mitigate the *inconsistency* between explanations and predictions, given the sampling process may inevitably infuse noises into the final prediction. Specifically, it employs probabilities across various class-indicative verbalizers in place of the original one-hot predictions. This design, although conceptually simple, can effectively reduce the discrepancies between explanations and predictions and further improve the final predictions accuracy.

Our contributions can be summarized as follows:

- We propose the EASE framework to better facilitate in-context learning for large language models with natural language explanations.
- We begin with an analysis of the limitations in current in-context learning methods, leading to the development of explanation-aware ensemble and soft probability aggregation techniques. These techniques aim to prioritize high-quality explanations and minimize inconsistencies between explanations and predictions, without requiring large amounts of additional annotation.
- We conduct experiments on seven natural language understanding (NLU) datasets spanning between natural language inference (NLI) and question answering (QA), and our method outperforms state-of-the-art approaches using 4 different LLMs as the backbone. Our analysis further justifies the advantage of EASE for dealing with unreliable explanations as well as mitigating inconsistent predictions.

2 Related Work

Two prevalent explanation types exist for interpreting NLP models: (1) *extraction-based explanations* that highlight important segments of the original input (DeYoung et al., 2020; Paranjape et al., 2020; Zhou et al., 2020; Yin and Neubig, 2022) and (2) *free-form explanations* that craft prediction rationales directly using natural language text (Rajani et al., 2019; Sun et al., 2022; Wiegreffe et al., 2021, 2022; Wang et al., 2023a; Ludan et al., 2023; Shen et al., 2023; Zhang et al., 2024). Beyond aiding in model interpretation, recent studies have demonstrated that these explanations can also enhance the few-shot reasoning capabilities of large language models. For example, (Wei et al., 2023b) use task-level explanations to assist the test-time adaptation

for LLMs. (Krishna et al., 2023) leverage the extracted keywords tokens as additional input to assist LLM in-context learning, but requires a white-box LLM. (Wei et al., 2022b; Zelikman et al., 2022) propose to *prepend explanations* before the answers while (Lampinen et al., 2022) suggest adding *post-answer explanations*. Given that these explanations are often derived during the LLM decoding stage and may contain noise, (Wang et al., 2023d, 2022b) advocate for generating multiple candidate explanations with predictions, followed by aggregating these predictions via majority voting. In our study, we focus on *free-form explanations* and explore how to better aggregate these predictions with explanations in a weighted ensemble. Using a bootstrapped LLM, we evaluate each explanation to enhance in-context learning.

Another line of research related to our study is automated explanation quality evaluation (Sun et al., 2022; Wang et al., 2022a; Joshi et al., 2023; Chen et al., 2023a,b). (Ye and Durrett, 2022) utilize lexical features to measure the faithfulness of explanations without considering their semantics. (Chen et al., 2021; Li et al., 2023b) leverage a NLI fine-tuned model to verify the reliability of explanations. (Fu et al., 2023; Liu et al., 2023; Qin et al., 2023; Jiang et al., 2023) also study how to use LLM to build a generic text quality scorers for NLP tasks. These studies rely on ground-truth labels and human annotations, making them less suitable when the labels for test instances are unknown. Besides, several works attempt to build up a scoring model (Wang et al., 2023b; Khalifa et al., 2023) and rely on a large training set for scoring model training, yet such labeled data can be infeasible to obtain under the true few-shot setting. In contrast, our research focuses more on effectively leveraging model-generated explanations to improve LLM in-context learning with a few demonstrations only.

3 Method

In this section, we first give a brief introduction to the problem definition. Then, we present our approach with two designs, namely explanation-aware ensemble and soft probability aggregation, with the goal of leveraging the generated explanations to improve the final prediction performance.

3.1 Problem Definition

We are given an LLM $\mathcal{M}$ parameterized by θ, a set of few-shot demonstrations $\mathcal{D} = \{(x_i, e_i, y_i)\}_{i=1}^{K}$ on a target classification task[2], where K is the number of demonstrations, x_i, y_i are the input and label for the i-th example, and e_i is the ground-truth explanation for x_i. For each example $x \in \mathcal{D}_{\text{test}}$, we use $\mathcal{M}$ and $\mathcal{D}$ to predict its label. Our primary goal is to improve the prediction accuracy on $\mathcal{D}_{\text{test}}$.

3.2 Recap of Self-consistency Pipeline

Here we give an introduction to *self-consistency* (Wang et al., 2023d) for LLM in-context learning. For each test example $x \in \mathcal{D}_{\text{test}}$, it first forms the prompt for few-shot demonstrations as $\mathcal{P} = \{\mathcal{T}, \text{shuffle}(\|_{i=1}^{K}(x_i, e_i, y_i))\}$, where $\mathcal{T}$ is the prompt template, and $\text{shuffle}\left(\|_{i=1}^{K}(x_i, e_i, y_i)\right)$ is a permutation of K demonstrations. Then, it generates N candidate explanations together with predictions (denoted as (e_j, p_j)) via sampling from the LLM with non-zero temperature as

$$(e_j, p_j)_{j=1}^{N} \sim p_\theta\left(e, p \mid \mathcal{P}, x\right). \quad (1)$$

Finally, it aggregates these N candidates into the final prediction via majority voting as $\widetilde{y} = \text{argmax}_y \sum_{j=1}^{N} \mathbb{I}(p_j = y)$. Self-consistency enhances the standard explain-then-predict pipeline by utilizing multiple predictions derived from varied explanations. Despite its strong performance, through our examination, we've pinpointed two primary bottlenecks within the self-consistency pipeline, listed as follows:

- *Explanation-agnostic Ensembling*: Self-consistency uniformly weights all predictions and aggregates them via majority voting. It overlooks the variance in explanation quality, which can be problematic when certain predictions stem from flawed reasoning paths evident in poor-quality explanations (Agarwal et al., 2024).
- *Explanation-Prediction Inconsistency*: During its prediction, self-consistency employs *temperature sampling* to draw samples from the LLM. This sampling step can introduce noise, leading to predictions that are inconsistent with their corresponding explanations (Ye and Durrett, 2022).

The identified limitations necessitate the need to better harvest intermediate explanations for obtaining the final prediction. Towards this goal, we

[2]Our main focus is on classification tasks, but our framework can also be extended to generative tasks (section 4.7).

propose EASE, a framework tailored to tackle the aforementioned challenges. EASE is comprised with two techniques, explanation-aware ensemble and soft probability aggregation, to optimize the LLM's prediction accuracy when deriving final outcomes from multiple candidate explanations.

3.3 Explanation-guided Ensemble

LLMs typically produce multiple explanations along with their predictions through a sampling process. Due to the intrinsic randomness of this sampling, the quality of these predictions can fluctuate. To address the potential pitfalls where erroneous explanations results in inaccurate predictions, we introduce the *explanation-aware ensemble* technique. This method estimates the significance of each class prediction based on its corresponding explanation. Consequently, our explanation-aware ensemble technique ensures that predictions linked with better explanations carry greater weight during the final prediction aggregation phase.

LLM as Few-shot Explanation Scorer To evaluate various explanations, past research either measured the lexical overlap between the explanation and the input (Ye and Durrett, 2022) or employed models fine-tuned on NLI tasks (Chen et al., 2021; Li et al., 2023b). In contrast to them, which either overlook the deep semantics of explanations or require extra human annotations, our explanation scorer is developed based on the LLM $\mathcal{M}$, directly harnessing its inherent reasoning capabilities.

Given the original task input x and one explanation e, we use the verbalizer $v_{\text{pos}}(v_{\text{neg}})$ to represent the class of this explanation being "*positive*" ("*negative*"). A "*positive*" explanation means this explanation can help the model reach correct answer and a "*negative*" explanation means the other way around. Then, we craft a supplementary prompt $\mathcal{T}_{\text{score}}$ = "*Can this explanation be used to help the model answer the question?*" for LLM prompting. With the verbalizers and prompts, we recast the problem of explanation scoring into determining the conditional probability of producing the verbalizer aligned with the positive label v_{pos} as

$$\omega_e = p_\theta\left(y = v_{\text{pos}} \mid \mathcal{T}_{\text{score}}, x, e\right). \tag{2}$$

In this way, the score ω_e is normalized between 0 and 1 and a higher score indicates the explanation with better quality.

Bootstrapped LLM Scorer Although the above approach can already produce scores for each prediction, the score generated with the LLM $\mathcal{M}$ can still be biased and less precise (Wang et al., 2023c), especially under the zero-shot scenario where no demonstrations are provided. To mitigate the bias and generate reliable scores, we propose to provide additional examples to serve as "*positive*" and "*negative*" explanations to facilitate LLM scoring using the original few-shot demonstrations in $\mathcal{D}$.

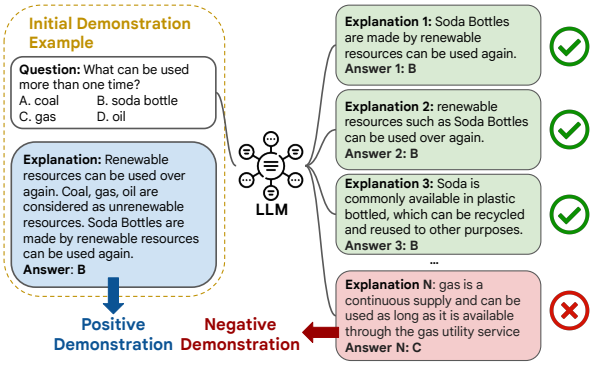


Figure 2: Bootstrapped LLM Scorer.

For each original demonstration instance, it is straightforward to obtain "*positive*" examples from the ground-truth explanation. Obtaining "*negative*" examples, on the other hand, can be more challenging as they are not explicitly provided. To tackle this issue, we exploit the assumption based on the utility of explanations: *an ideal explanation should guide the model towards the accurate prediction of ground-truth labels* (Wiegreffe et al., 2021). Consequently, it's reasonable to classify explanations leading to erroneous predictions as "negative". In practice, for every instance $(x_i, y_i) \in \mathcal{D}$, we randomly select k (8 in this work) exemplars from the training set and then use these as demonstrations and generate a set of candidate pairs $\mathcal{C}_i = \{(e_{ij}, p_{ij})\}_{j=1}^{N}$ via sampling from the LLM. Then, if the explanation-prediction pair (e_{ij}, p_{ij}) from $\mathcal{C}_i$ satisfies $y_i \neq p_{ij}$, we select e_{ij} as the negative explanation set $\mathcal{N}_i$ for x_i as

$$\mathcal{N}_i = \{(e_{ij}, p_{ij}) \in \mathcal{C}_i \mid y_i \neq p_{ij}\}. \tag{3}$$

For every instance, a random "*negative*" explanation is chosen from the candidate set. This produces a *balanced* demonstration set for LLM explanation scoring without requiring extra human annotations.

3.4 Soft Probability Aggregation

In the preceding step, the primary objective is to assign a score to each prediction based on its associated explanation through the LLM $\mathcal{M}$. This process, however, does not account for directly modeling the LLM's output predictions. To bridge this gap, we propose *soft probability aggregation*,

a simple and intuitive approach to resolve the discrepancy between the explanations and predictions — rather than aggregating over the raw predictions, it directly computes the sum of the probabilities associated with each potential label, expressed as

$$\widetilde{y} = \underset{y}{\operatorname{argmax}} \sum_{j=1}^{N} p_\theta \left(y \mid \mathcal{P}, x, e_j \right). \tag{4}$$

The *soft probability aggregation* reduces the noise inherited in LLM sampling-based decoding algorithms, resulting in a more accurate final prediction.

3.5 Summary

By plugging these two techniques together, we obtain the final prediction $\widetilde{y}$ for test instance x as

$$\widetilde{y} = \underset{y}{\operatorname{argmax}} \sum_{j=1}^{N} \omega_{e_j} \times p_\theta \left(y \mid \mathcal{P}, x, e_j \right), \tag{5}$$

where e_j is the explanation generated via Eq. 1, the ω_{e_j} is the weight for e_j using the bootstrapped LLM scorer in Eq. 2, and $p_\theta \left(y \mid \mathcal{P}, x, e_j \right)$ is the soft probability generated using Eq. 4. Overall, calculating the score for explanations and soft probabilities both take an additional $O(N)$ time complexity. Fortunately, these two steps do not require additional training and can be efficiently supported with distributed inference techniques in practice[3]. Other than these techniques, EASE keeps other components intact and can be plugged into most LLMs to empower their in-context learning ability.

4 Experiments

4.1 Experiment Setups

Tasks We evaluate our EASE framework on two types of tasks: natural language inference and question answering including E-SNLI (Camburu et al., 2018), ANLI (Nie et al., 2020), ECQA (Aggarwal et al., 2021), OpenbookQA (Mihaylov et al., 2018) and StrategyQA (Geva et al., 2021). Besides, we conduct additional experiments on XNLI (Conneau et al., 2018), GSM8k (Cobbe et al., 2021) and SVAMP (Patel et al., 2021) in Section 4.7.

Baselines We consider the following approaches: (1) **Standard In-context Learning (ICL)** (Brown et al., 2020), (2) **Predict-then-Explain (PE)** (Lampinen et al., 2022), (3) **Explain-then-Predict (EP)** (Wei et al., 2022b), (4) **Self-consistency** (Wang et al., 2022b, 2023d), (5) **FLamE** (Zhou et al., 2023), (6) **OrderEnsemble** (Lu et al., 2022) and (7) **PromptBoost** (Hou et al., 2023) as baselines. The details are in App. B.

Implementation Details In our main experiments, we use PaLM2-S and PaLM2-L (Anil et al., 2023) as the backbone model. Results on more (open source) backbone models are reported in Section 4.3. For each dataset, we set the size of few-shot examples to 48 following (Zhou et al., 2023; Marasovic et al., 2022), and fit as many instances as possible during inference until reached the maximum length. As the LLM is often sensitive to the selection of few-shot examples (Ye and Durrett, 2023; Liu et al., 2022), for each dataset, we create 5 splits from the original dataset, each containing 300 test examples, and report the average performance over 5 splits. During sampling, we set the default temperate to $t = 0.7$ and sample $N = 9$ candidate explanations for each instance.

4.2 Overall Results

Table 1 exhibits the performance of EASE and baselines on seven datasets using PaLM 2-S and PaLM 2-L as the backbone. From the results, we have the following findings: **First**, leveraging explanations often improves LLM in-context learning, especially when aggregating multiple predictions sampled from the LLM[4]. Conversely, the standard EP pipeline sometimes even hurts the performance of larger models. Existing boosting methods for in-context learning do not help much in improving performance. **Second**, despite its complex design, the latest baseline FLamE often falls short compared to other baselines, which suggests that fine-tuning an additional classifier is particularly important for FLamE and it might be less compatible with the LLM in-context learning framework. **Third**, EASE consistently outperforms all other methods across nearly all datasets and model sizes, providing a reliable way to improve in-context learning. **Finally**, When comparing EASE with its own variants (e.g. w/o BLS and SPA), it's observed that the original EASE consistently holds an advantage, indicating the necessity of both PW and SA components in maximizing performance.

4.3 Results on Open-source Models

In order to demonstrate the generalizability of our EASE framework, as well as promote reproducibil-

[3]More studies are in Appendix E.

[4]We have also tried to incorporate multiple explanations with temperature sampling for PE pipeline, but we find performance drops. This is because the prediction of the PE pipeline can not benefit from generated explanations.

Table 1: The main experiments results, where "BLS" stands for bootstrapped LLM scorer and "SPA" stands for soft probability aggregation. All results have passed the statistically significant test ($p < 0.05$) over baselines.

Backbone	Methods	E-SNLI	ANLI-R1	ANLI-R2	ANLI-R3	ECQA	StrategyQA	OpenbookQA	Average
PaLM 2-S	ICL (Brown et al., 2020)	59.88	54.38	48.10	52.66	59.84	66.69	80.21	60.25
	PE (Lampinen et al., 2022)	71.02	62.59	55.18	57.17	74.39	71.75	79.70	67.40
	EP (Wei et al., 2022b)	64.53	57.40	53.00	53.33	72.11	72.40	81.38	64.88
	Self-consistency (Wang et al., 2023d)	68.68	65.40	56.49	59.00	74.48	76.94	83.47	69.21
	FLamE (Zhou et al., 2023)	67.58	60.36	52.00	50.15	72.80	75.33	80.14	65.48
	OrderEnsemble (Lu et al., 2022)	69.30	63.33	56.33	58.33	73.14	76.35	83.68	68.64
	PromptBoost (Hou et al., 2023)	70.65	64.00	55.63	60.33	73.50	77.30	83.91	69.33
	EASE	**75.01**	66.48	**59.66**	**64.33**	**75.59**	78.23	**84.10**	**71.92 (↑3.91%)**
	EASE w/o BLS	73.84	**66.84**	58.74	62.66	75.17	**78.40**	83.91	71.37
	EASE w/o SPA	69.82	66.77	58.50	62.50	75.42	78.33	83.68	70.73
PaLM 2-L	ICL (Brown et al., 2020)	87.42	79.00	68.33	65.65	81.29	81.13	91.17	79.14
	PE (Lampinen et al., 2022)	88.84	80.55	71.49	68.33	83.13	83.19	92.46	81.14
	EP (Wei et al., 2022b)	84.59	79.03	67.99	67.66	80.51	85.45	89.74	79.28
	Self-consistency (Wang et al., 2023d)	87.34	81.29	73.16	70.16	82.67	87.85	92.88	82.19
	FLamE (Zhou et al., 2023)	83.23	71.85	58.50	56.83	80.26	84.79	93.14	75.51
	OrderEnsemble (Lu et al., 2022)	87.12	80.66	72.66	69.33	82.25	87.55	93.10	81.81
	PromptBoost (Hou et al., 2023)	87.46	80.33	73.00	69.94	83.12	88.15	92.63	82.09
	EASE	**89.42**	**83.69**	**76.16**	**74.00**	**83.65**	**89.90**	**93.93**	**84.40 (↑2.69%)**
	EASE w/o BLS	88.94	82.87	75.60	72.66	83.42	89.34	93.72	83.79
	EASE w/o SPA	88.21	82.59	73.83	71.33	83.42	89.35	93.51	83.18

Table 2: The experiment results on open-source models, where "BLS" stands for bootstrapped LLM scorer and "SPA" stands for soft probability aggregation. All results have passed the significant test ($p < 0.05$) over baselines.

Model (→)	FLAN-UL2 (20B)	Llama-2 (7B)							
Dataset (→)	StrategyQA	E-SNLI	ANLI-R1	ANLI-R2	ANLI-R3	ECQA	StrategyQA	OpenbookQA	Avg.
ICL (Brown et al., 2020)	61.76	51.14	34.58	36.05	27.48	45.48	53.81	47.48	42.29
PE (Lampinen et al., 2022)	73.42	54.25	37.83	37.50	34.37	52.33	56.21	56.48	47.00
EP (Wei et al., 2022b)	75.46	56.90	35.41	39.16	36.04	54.45	57.17	44.35	46.21
Self-consistency (Wang et al., 2023d)	76.01	58.79	40.16	40.16	36.16	55.14	57.12	60.87	49.77
FLamE (Zhou et al., 2023)	72.17	49.32	36.83	35.16	36.50	45.11	57.70	46.23	43.84
OrderEnsemble (Lu et al., 2022)	75.46	58.62	37.50	36.66	36.66	53.42	56.01	44.67	45.93
PromptBoost (Hou et al., 2023)	75.22	58.04	39.17	38.33	35.83	52.57	56.94	46.33	46.74
EASE	**78.70 (↑ 3.55%)**	**60.80**	**44.50**	**41.66**	**41.33**	**60.45**	**59.81**	64.43	**53.28 (↑ 7.05%)**
EASE w/o BLS	77.31	59.54	43.45	41.33	40.33	60.34	59.62	**65.06**	52.81
EASE w/o SPA	77.78	58.50	41.33	40.16	35.33	54.97	57.40	61.71	49.91

Table 3: The study on different scoring approaches. Note that to ensure fair comparison, we do not use soft probability aggregation for our method and baselines.

Dataset (→)	E-SNLI		OpenbookQA		StrategyQA
Model (→)	PaLM 2-S	PaLM 2-L	PaLM 2-S	PaLM 2-L	FLAN-UL2
EASE	**69.82**	**83.68**	**83.68**	93.51	**78.70**
EASE w/ PE Negative	68.90	83.91	83.54	**93.93**	78.06
LLM Zero-shot Scoring (Fu et al., 2023)	66.84	81.77	81.38	88.50	75.15
LLM Pairwise Scoring (Qin et al., 2023)	69.25	82.97	82.97	93.14	76.93
Lexical Scoring (Ye and Durrett, 2022)	67.72	83.54	82.66	93.72	75.34
NLI Scoring (Chen et al., 2021)	64.87	81.89	82.21	91.52	76.11

ity, we extend our investigations to open-source LLMs including FLAN-UL2 (Tay et al., 2023)[5] and Llama-2-7b (Touvron et al., 2023). Both models have publicly accessible weights[6]. Table 2 shows the open-source LLMs generally perform worse than PaLM 2 on the challenging NLU benchmarks, mainly due to having fewer parameters. Despite this, the experiment results still align with our prior findings, demonstrating that our proposed techniques can consistently yield performance enhancements across these open-source LLMs.

[5] https://github.com/google-research/google-research/tree/master/ul2. We only test on StrategyQA since FLAN-UL2 is fine-tuned on labeled data from other datasets, violating the few-shot setting.

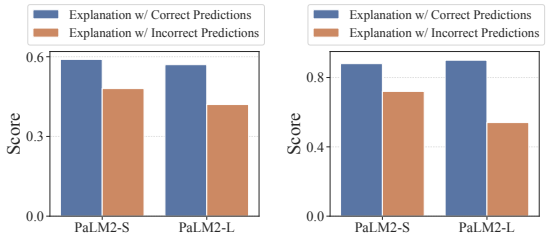
[6] https://huggingface.co/meta-llama/Llama-2-7b.

4.4 Study on Explanation-aware Ensemble

We perform additional experiments to further understand the benefit of the explanation-aware ensemble, and the result is shown in Table 3.

Performance w/ Different Scoring Methods We first compare our LLM-based explanation scorer with a few alternatives including (1) *lexical scoring*, which estimates the reliability of explanations via the lexical gap (Ye and Durrett, 2022), and (2) *NLI Scoring* that uses an NLI model to verify the reliability of explanations. We use MT5-XXL (Xue et al., 2021) fine-tuned on NLI datasets as the scorer. Overall, we observe that our model outperforms these models in most cases, indicating that LLM has a strong capacity for estimating the quality of the explanations with only a few demonstrations. In addition, we observe that pairwise scoring does not perform well for weighting the predictions. This is because it was originally proposed for text ranking, different from our scenarios in terms of input formats and relevance signals.

Performance w/ Different Bootstrapping Strategies To justify the design of leveraging the Explain-then-Predict (EP) pipeline to generate negative



(a) E-SNLI (b) OpenbookQA

Figure 3: Average Score of Bootstrapped LLM Scorer.

Table 4: The study on different probability aggregation approaches. Note that we do not use explanation-aware ensemble for our method and baselines.

Dataset (→)	**E-SNLI**		**OpenbookQA**		**StrategyQA**
Model (→)	PaLM 2-S	PaLM 2-L	PaLM 2-S	PaLM 2-L	FLAN-UL2
Inconsistency Ratio	14.60%	10.06%	13.96%	10.71%	10.00%
EASE	73.84	88.21	83.91	93.72	78.70
w/ argmax	73.20	87.90	83.68	93.51	78.42
Cond. Gen (Li et al., 2023a)	70.77	82.20	78.07	84.38	72.80

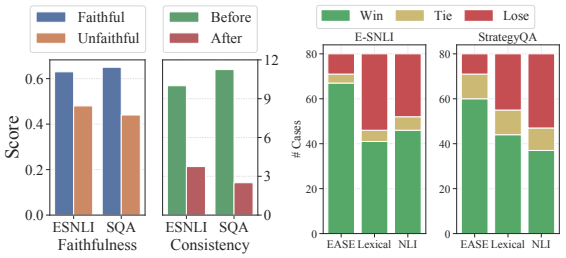
demonstrations, we also consider other ways including removing demonstrations as well as using the Predict-then-Explain (PE) pipeline. Overall, in many cases, using the EP pipeline leads to better results, as we observe that the PE pipeline sometimes causes the *false negative* issue: it will first generate incorrect predictions but followed with reasonable explanations. However, when the model performs reasonably well (e.g. PaLM 2-L on OpenbookQA), then it may make less erroneous prediction during the bootstrapping step, leading to insufficient training signals for EASE to perform well. In addition, no matter whether PE and EP are used, they both largely outperform the baseline where no demonstration is given, necessitating the role of demonstration for explanation-aware ensembling.

EASE Assigns Higher Scores for Reliable Explanations To justify that better scores are assigned to explanations with correct answers, we calculate the average score for explanations associated with correct and incorrect predictions[7]. The results in Fig. 3 show that EASE leads to a higher average score for those explanations with correct answers. We further justify this point in Sec. 4.6 *human study*.

4.5 Study on Soft Probability Aggregation

The premise behind soft probability aggregation is the potential inaccuracy in the prediction token due to temperature sampling variability. To verify this, we calculate the proportion of cases where the prediction token p_i is different than the prediction $p_i \neq \text{argmax}\, p(\cdot|\mathcal{P}, x, e_i)$. As exhibited in Table 4,

[7]To eliminate the effect of the sampling randomness, we calculate the prediction based on the soft probability in Eq. 4.



(a) Faithful & Consistency (b) Pairwise Evaluation

Figure 4: Human Study Results.

we observe that such inconsistency predictions appear in 10% to 15% of the cases, which is not rare in practice. By using the soft score, we observe that it consistently leads to performance boosts. The gain is more evident when the inconsistency issue is more severe — on E-SNLI dataset with PaLM 2-S as the backbone, around 15% examples have inconsistent predictions. Incorporating soft probability aggregation leads to a notable performance gain (from 68.68% to 73.84%). When compared to other methods for prediction correction, such as using hard predictions (*i.e.* $\text{argmax}\, p(\cdot|\mathcal{P}, x, e_i)$) or generation probability conditioned on different verbalizers, EASE achieves better performance. More case studies on using soft probabilities are deferred to Appendix F.2.

4.6 Human Study on Explanations

We conduct additional human studies to further investigate whether the scores generated by LLM are aligned with human preferences. On two datasets, we randomly select 80 instances for evaluation.

Faithfulness and Consistency Evaluation. To provide in-depth analysis, we follow (Ye and Durrett, 2022) to consider two dimensions: (1) *faithfulness*: whether the explanation is grounded in the corresponding input context; (2) *consistency*: whether the explanation entails the prediction. We ask human raters to indicate the 'faithfulness' and 'consistency' of the explanations. The results are shown in Figure 4(a). For *faithfulness*, we observe a higher average score from the LLM scorer to those identified faithful explanations. For *consistency*, we observe that there is a significant drop in the inconsistency ratio when soft probability aggregation is used. These results further justify that two modules in EASE can collectively reduce the effect of unreliable explanations.

Pairwise Evaluation. For each instance, we sample two explanations with *different* predictions as $\{(e_1, p_1), (e_2, p_2)\}$, with one being correct. We

Table 5: Performance on XNLI dataset using four target languages.

	DE	FR	ES	ZH	Avg.
Self-consistency	47.15	44.66	48.33	39.67	44.95
EASE	**51.50**	**48.33**	**50.66**	**43.00**	**48.37 (+7.4%)**
EASE w/o BLS	48.82	47.33	49.66	42.00	46.95
EASE w/o SPA	49.49	42.33	47.00	40.33	44.79

Table 6: Performance on Arithmetic Reasoning datasets.

	GSM8k	SVAMP
Self-consistency	22.50	54.33
EASE (w/ BLS Only, Few Shot)	**24.25**	**56.33**
GRACE (Khalifa et al., 2023) (Supervised)	30.9	55.6

compare our approach and two baselines (NLI model, lexical overlap) with human raters: for each pair of explanations, we first ask four humans to determine which explanation is better and use c_i $(i = 1, 2)$ to denote the number of raters that select e_i as the better one. Then, we use different models to estimate the score for explanations separately, denoted as (s_{e_1}, s_{e_2}). The final judge of "Win-Tie-Lose" is determined to be:

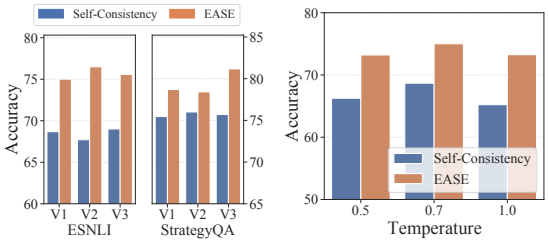
$$r = \begin{cases} \text{win}, & \text{if } (c_1 > c_2 \text{ and } s_{e_1} > s_{e_2}) \textbf{ or } (c_1 < c_2 \text{ and } s_{e_1} < s_{e_2}); \\ \text{tie}, & c_1 = c_2; \\ \text{lose}, & \text{if } (c_1 < c_2 \text{ and } s_{e_1} > s_{e_2}) \textbf{ or } (c_1 > c_2 \text{ and } s_{e_1} < s_{e_2}). \end{cases}$$

The final results are shown in Figure 4(b). The cohen's kappa among human raters are 0.75 (E-SNLI) and 0.64 (StrategyQA), which stands for "*substantial agreement*". Overall, EASE aligns with human preferences the best, indicating its better ability to be the proxy for explanation quality estimation. We display more examples of generated explanations and the scores in Appendix F.1.

4.7 Results on More Challenging Tasks

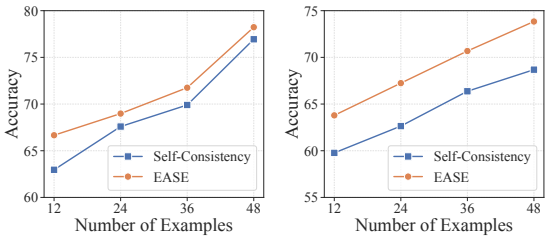
We provide additional experiments to show that EASE can be applied to more challenging tasks where *the initial explanation and predictions are not good enough*. For all experiments, we use LLama2-7b as the backbone LLM.

Cross-Lingual Transfer We extend our framework to multi-lingual NLI tasks, where we provide the demonstrations and explanations in English and inference on target languages including German (DE), French (FR), Spanish (ES), and Chinese (ZH) using the examples in XNLI dataset (Conneau et al., 2018). The performance (accuracy) of our method, as well as our direct baseline (self-consistency), are shown in Table 5. From the table, we observe that our method achieves a 7.4% performance gain when compared to the self-consistency baseline, demonstrating our method can be readily applied to challenging cross-lingual scenarios.



(a) Prompt Format (b) Temperatures (E-SNLI)

Figure 5: Effect of prompt format and temperatures.



(a) E-SNLI (b) StrategyQA

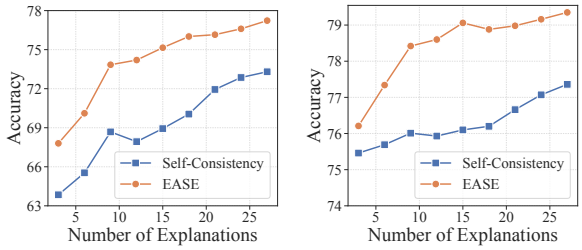
Figure 6: Effect of number of demonstrations.

Arithmetic Reasoning For arithmetic reasoning, one difference is that this task requires the generation of a candidate answer in the form of a numerical value, rather than simply outputting a class or choice. While direct adoption of soft probability aggregation may not be feasible, we can still leverage the bootstrapped LLM scorer to assign weights to different outputs generated by LLMs. To evaluate our approach, we conducted experiments on GSM8k (Cobbe et al., 2021) and SVAMP (Patel et al., 2021). The results of our direct baseline (self-consistency) and our proposed method are summarized in Table 6. From these results, it is evident that incorporating the LLM as an explanation (reasoning) scorer can further enhance the reasoning ability of the LLM.

4.8 Additional Studies

As EASE relies on several key components such as prompts and sampling steps, we study their effect on the final prediction, using PaLM 2-S as the backbone.

Effect of the Sampling Temperatures and Prompt Templates We study the robustness EASE to different prompt templates by choosing three different prompt formats from (Bach et al., 2022) (the details are in Appendix C.3) on two datasets. Overall, from Figure 5(a) we observe that EASE is robust to them as all of the prompt formats lead to performance gains when compared to the strongest baseline self-consistency. Similarly, in Figure 5(b),



(a) E-SNLI (b) StrategyQA

Figure 7: Effect of number of explanations.

Table 7: Verbalizer Study for Bootstrapped LLM Scorer, using PaLM 2-S as the backbone.

Template	V1	V2	V3
E-SNLI	75.01	73.75	74.12
StrategyQA	78.40	78.23	76.75

we observe that EASE also performs better than baseline under all temperature settings, further justify its robustness across different settings.

Effect of the Size of Demonstrations K Figure 6 illustrates the performance with different size of demonstrations. By increasing the number of K, the performance gradually increases, while EASE achieves performance gains under all value of K.

Effect of the Number of Generated Explanations N In Figure 7, we examine the influence of the number of explanations. On both datasets, increasing the explanations generally improves the performance, while EASE achieves better performance than the baselines using only 30% - 40% of the generated explanations, which can reduce the burden of sampling massive explanations while maintaining the performance.

Effect of Verbalizers for Bootstrapped LLM Scorer We investigate the role of verbalizers in representing the "*positive*" and "*negative*" explanations. We consider three set of verbalizers, namely V1:"*Yes*" and "*No*", V2: "*True*" and "*False*", and V3: "*Foo*" and "*Jaa*" using symbolic tuning (Wei et al., 2023a). Using PaLM 2-S as the backbone, the experimental results are shown in Table 7. From the results, we observe that the original "*Yes*" and "*No*" generally perform better. Symbolic tuning does not work as well as other verbalizers with concrete semantics, indicating it may not be strong enough for the explanation scoring task.

5 Conclusion

In this work, we empower LLM's in-context learning ability with natural language explanations. Specifically, we design explanation-aware ensemble to weight multiple predictions using their associated explanations and realize this idea using a bootstrapped LLM scorer. In addition, we leverage a soft probability aggregation scheme to mitigate the issue of inconsistent predictions for ensembling. We conduct extensive experiments on seven datasets from a diverse task set and show our proposed framework can outperform previous state-of-the-art methods using four LLMs as backbones.

Acknowledgements

We would like to thank reviewers and AE from the ACL Rolling Review for the helpful feedback. This work was supported in part by NSF IIS-2008334 and CAREER IIS-2144338.

Limitations

In this work, our primary goal is to identify the existing issues to better leverage explanations to empower in-context learning. While our approach has shown promise, it also comes with increased computational demands, as both explanation-aware ensemble and soft probability aggregation steps require additional computation overhead[8]. Future work could explore designing more powerful prompts to let LLMs directly output the suffix tokens as quality score (Tian et al., 2023). Additionally, our methodology depends on the logits returned in both the explanation-aware ensemble and soft probability aggregation processes, making it less suitable to directly adapt to black-box LLMs (e.g. ChatGPT, OpenAI (2023)). To approximate the soft score, one strategy is to set the temperature to a non-zero value and conduct multiple sampling steps, then use the frequency of the corresponding verbalizers as the proxy of the score.

Besides, the key assumption of EASE is that different explanations are of diverse quality, while those explanation leads to correct predictions tend to be of higher quality. We mainly conduct empirical experiments to support this point, yet there often exists multiple facets to evaluate the quality of free-text explantions (Chen et al., 2023a,b; Sun et al., 2022). More in-depth metrics are needed to

[8]However, in Appendix E, we show that under the same inference time budget, our method still outperforms the best baseline (i.e. self-consistency).

faithfully evaluate the quality of free-text explanations and reveal the true inner workings of EASE.

We also assume that the LLM must possess a baseline of language modeling and reasoning capabilities for effective in-context learning. For tasks where in-context learning is not viable, alternatives such as continuous pretraining and task-specific fine-tuning might be more suitable to enhance the LLM's task-specific knowledge. Besides, our experiments, which included a range of LLMs from Llama-2 to PaLM 2 and cover various NLP tasks, demonstrate that these models generally exhibit some level of reasoning capacity, performing better than random guesses.

Additionally, while EASE augments in-context learning by weighting predictions through explanations, it does not refine the explanation's content. For future works, it is potential to leverage techniques such as self-refinement (Madaan et al., 2023; Ling et al., 2023) and debating (Du et al., 2023) to elevate explanation quality and strengthen the model's reasoning abilities.

Ethics Considerations

We acknowledge the risk that the explanations from large language model may inherit systematic biases from their pretraining data and contain incorrect information. Our approach aims to mitigate this by identifying the most reliable explanations. This process does not eliminate the possibility of bias, but offers a method to reduce its impact, emphasizing the need for future work on improving the way these models are trained and utilized.

References

Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. 2024. Faithfulness vs. plausibility: On the (un) reliability of explanations from large language models. *arXiv preprint arXiv:2402.04614*.

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. 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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9560–9572.

Hanjie Chen, Faeze Brahman, Xiang Ren, Yangfeng Ji, Yejin Choi, and Swabha Swayamdipta. 2023a. REV: Information-theoretic evaluation of free-text rationales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2007–2030, Toronto, Canada.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. 2023b. Do models explain themselves? counterfactual simulatability of natural language explanations. *ArXiv preprint*, abs/2307.08678.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro

Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *ArXiv preprint*, abs/2305.14325.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *ArXiv preprint*, abs/2302.04166.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Bairu Hou, Joe O'Connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. PromptBoosting: Black-box text classification with ten forward passes. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 13309–13324. PMLR.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. 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.

Brihi Joshi, Ziyi Liu, Sahana Ramnath, Aaron Chan, Zhewei Tong, Shaoliang Nie, Qifan Wang, Yejin Choi, and Xiang Ren. 2023. Are machine rationales (not) useful to humans? measuring and improving human utility of free-text rationales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7103–7128, Toronto, Canada.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. GRACE: Discriminator-guided chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15299–15328, Singapore. Association for Computational Linguistics.

Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. 2023. Post hoc explanations of language models can improve language models. *ArXiv preprint*, abs/2305.11426.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023a. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *ArXiv preprint*, abs/2303.16634.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Josh Magnus Ludan, Yixuan Meng, Tai Nguyen, Saurabh Shah, Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2023. Explanation-based finetuning makes models more robust to spurious cues. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4420–4441, Toronto, Canada. Association for Computational Linguistics.

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 *Thirty-seventh Conference on Neural Information Processing Systems*.

Ana Marasovic, Iz Beltagy, Doug Downey, and Matthew Peters. 2022. Few-shot self-rationalization with natural language prompts. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 410–424, Seattle, United States. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *ArXiv preprint*, abs/2004.14546.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1938–1952, Online. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *ArXiv preprint*, abs/2306.17563.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Jiaming Shen, Jialu Liu, Dan Finnie, Negar Rahmati, Mike Bendersky, and Marc Najork. 2023. "why is this misleading?": Detecting news headline hallucinations with explanations. In *Proceedings of the ACM Web Conference 2023*, pages 1662–1672.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Jiao Sun, Swabha Swayamdipta, Jonathan May, and Xuezhe Ma. 2022. Investigating the benefits of free-form rationales. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5867–5882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. 2023. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *ArXiv preprint*, abs/2305.14975.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022a. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Annual Meeting of the Association for Computational Linguistics*.

PeiFeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2023a. PINTO: Faithful language reasoning using prompt-generated rationales. In *The Eleventh International Conference on Learning Representations*.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. 2023b. Making large language models better reasoners with alignment. *arXiv preprint arXiv:2309.02144*.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023c. Large language models are not fair evaluators. *ArXiv preprint*, abs/2305.17926.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022b. Rationale-augmented ensembles in language models. *ArXiv preprint*, abs/2207.00747.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023d. 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*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jerry Wei, Le Hou, Andrew Kyle Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc V Le. 2023a. Symbol tuning improves in-context learning in language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Kangda Wei, Sayan Ghosh, Rakesh Menon, and Shashank Srivastava. 2023b. Leveraging multiple teachers for test-time adaptation of language-guided classifiers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7068–7088, Singapore. Association for Computational Linguistics.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2022. Reframing human-AI collaboration for generating free-text explanations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 632–658, Seattle, United States. Association for Computational Linguistics.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. *Advances in neural information processing systems*, 35:30378–30392.

Xi Ye and Greg Durrett. 2023. Explanation selection using unlabeled data for chain-of-thought prompting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 619–637, Singapore. Association for Computational Linguistics.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267, Rochester, New York. Association for Computational Linguistics.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Haorui Wang, Zhen Qin, Feng Han, Jialu Liu, Simon Baumgartner, Michael Bendersky, and Chao Zhang. 2024. Plad: Preference-based large language model distillation with pseudo-preference pairs.

Wangchunshu Zhou, Jinyi Hu, Hanlin Zhang, Xiaodan Liang, Maosong Sun, Chenyan Xiong, and Jian Tang. 2020. Towards interpretable natural language understanding with explanations as latent variables. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yangqiaoyu Zhou, Yiming Zhang, and Chenhao Tan. 2023. FLamE: Few-shot learning from natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6743–6763, Toronto, Canada. Association for Computational Linguistics.

A Datasets Details

A.1 Datasets Used in Main Experiments

The seven benchmarks in our experiments are all publicly available. The details for these datasets are included as follows:

- **E-SNLI** (Camburu et al., 2018) is an enriched version of the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), augmented with human-annotated natural language explanations for entailment relations;
- **ANLI-R1/R2/R3** (Nie et al., 2020) are a set of three collections of adversarially generated NLI examples curated through a human-in-the-loop process;
- **ECQA** (Aggarwal et al., 2021) is built upon CommonsenseQA benchmark (Talmor et al., 2019) and contains additional human-annotated question explanations;
- **OpenbookQA** (Mihaylov et al., 2018) is a QA dataset that requires comprehensive reasoning from open-book sources. As no ground-truth explanations are given, we use the provided facts as the proxy explanations.
- **StrategyQA** (Geva et al., 2021) focuses on reasoning over complex, multi-hop questions that often require strategic planning.

A.2 Additional Datasets

- **XNLI** (Conneau et al., 2018) is a cross-lingual NLI dataset, designed to evaluate language understanding across different languages. It extends the Stanford NLI (SNLI) dataset by providing translations of the English examples into various languages.
- **GSM8k** (Cobbe et al., 2021) provides a diverse set of questions that require logical reasoning, which focuses on grade-school level mathematical problems.
- **SVAMP** (Patel et al., 2021) is centered around assessing models in solving arithmetic word problems, specifically designed to test the variation in problem phrasing.

A.3 Download Links

Below are the links to downloadable versions of these datasets.

- **E-SNLI**: https://huggingface.co/datasets/esnli;
- **ANLI R1/R2/R3**: https://github.com/facebookresearch/anli;
- **ECQA**: https://github.com/allenai/feb;
- **OpenbookQA**: https://huggingface.co/datasets/openbookqa;
- **StrategyQA**: for StrategyQA we use the question-only set from the link https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/strategyqa;
- **XNLI**: https://huggingface.co/datasets/xnli;
- **GSM8k**: https://huggingface.co/datasets/gsm8k;
- **SVAMP**: https://github.com/arkilpatel/SVAMP/tree/main/data

A.4 Setups

Main Experiment Setups By default, we sample few-shot demonstrations from the train set and sample from the test split for all datasets. For OpenbookQA, as the original dataset only contains 500 test examples, in each split we use 100 examples. For ANLI, as some of the examples contain no explanations, while the explanations for some examples include task-irrelevant information such as '`I think the computer was confused because so many of the words were similar to the description`'. To reduce the effect of such examples, we remove those examples occurs with term '`the system`', '`the computer`', '`the model`', '`the AI`', and manually checked all the few-shot demonstrations to ensure that there is no such information in explanations.

Setups for Multilingual NLI For the multi-lingual NLI task, we simulate a cross-lingual setting where we provide the demonstrations and explanations in English and inference on target languages including German (DE), French (FR), Spanish (ES), and Chinese (ZH) using the examples in XNLI dataset (Conneau et al., 2018). Note that this is one common setting studied in previous multilingual reasoning benchmarks (Shi et al., 2023). We use 16-shot examples from the E-SNLI dataset as demonstrations, and for each target language, we sample 300 instances.

Setups for Arithmetic Reasoning For each dataset, we employed 16 examples from the training set as few-shot demonstrations. We use the prompts provided in existing works (Wang et al., 2023d) to avoid additional prompt engineering.

B Baselines

The detailed information for baselines are described as follows.

- **Standard In-context Learning (ICL)** (Brown et al., 2020): it solely uses the input-label pairs for few-shot learning without using explanations.
- **Predict-then-Explain (PE)** (Lampinen et al., 2022): it provides the explanation after the label for each instance when constructing demonstrations. During the inference stage, it generates the explanation after the prediction.
- **Explain-then-Predict (EP)** (Wei et al., 2022b): it is the standard chain-of-thought pipeline which provides an explanation before the label as demonstrations. During the inference stage, it first generates an explanation, then followed by the prediction. Note that for both PE and EP method, we use greedy sampling to obtain the explanation and prediction.
- **Self-consistency** (Wang et al., 2022b, 2023d): it improves over the standard EP pipeline by aggregating over multiple explanations from LLMs to enhance the result.
- **FLamE** (Zhou et al., 2023) is a recent LLM few-shot learning method that generates multiple label-conditioned explanations and determines the final prediction based on the label that achieves the highest logit after reviewing all explanations for the given instance[9].
- **Order-Ensemble** (Lu et al., 2022) is an ensemble-based method, which generates multiple outputs via shuffling the order of the few-shot demonstrations, and use the majority voting for ensembling.
- **PromptBoost** (Hou et al., 2023) is also an ensemble-based method, which leverages AdaBoost algorithm to assign different weights for predictions from different prompts.

[9]In the original FLamE paper, the RoBERTa is used for final classification. For a fair comparison, we adjusted FLamE to use the in-context LLM as the classifier.

C Prompt Formats

In this section, we list the prompts used in our experiments.

C.1 Prompt Format For In-context Learning

In this step, we list the prompt for generating the explanations and predictions. Many of the prompt formats are adapted from (Bach et al., 2022). Note that the blue text is instance-dependent, while the red text is the model's expected output.

C.1.1 E-SNLI

Listing 1: Prompt Format for E-SNLI dataset, standard in-context learning.

```
In this task, given a premise and
 a hypothesis, your job is to
determine whether the hypothesis
can be inferred from the premise.


# demonstrations (no more than
48)
Based on the premise: [premise],
can we infer the hypothesis:
 [hypothesis] from the premise?
Choose among Yes, Maybe, and No.
Answer: [Answer]

# test examples
Based on the premise: [premise],
can we infer the hypothesis:
 [hypothesis] from the premise?
Choose among Yes, Maybe, and No.
Answer: [Answer]
```

Listing 2: Prompt Format for E-SNLI dataset, using predict-then-explain pipeline.

```
In this task, given a premise and
 a hypothesis, your job is to
determine whether the hypothesis
can be inferred from the premise.


# demonstrations (no more than
48)
Based on the premise: [premise],
can we infer the hypothesis:
 [hypothesis] from the premise?
Choose among Yes, Maybe, and No.
Answer: [Answer]
Explanation: [Explanation]
```

```
# test examples
Based on the premise: [premise],
can we infer the hypothesis:
 [hypothesis] from the premise?
Choose among Yes, Maybe, and No.
Answer: [Answer]
Explanation: [Explanation]
```

Listing 3: Prompt Format for E-SNLI dataset, using explain-then-predict pipeline.

```
In this task, given a premise and
 a hypothesis, your job is to
determine whether the hypothesis
can be inferred from the premise.


# demonstrations (no more than
48)
Based on the premise: [premise],
can we infer the hypothesis:
 [hypothesis] from the premise?
Choose among Yes, Maybe, and No.
Answer: [Answer]
Explanation: [Explanation]

# test examples
Based on the premise: [premise],
can we infer the hypothesis:
 [hypothesis] from the premise?
Choose among Yes, Maybe, and No.
Explanation: [Explanation]
Answer: [Answer]
```

C.1.2 ANLI

Listing 4: Prompt Format for ANLI dataset, standard in-context learning.

```
In this task, given a premise and
 a hypothesis, your job is to
determine whether the hypothesis
can be inferred from the premise.


# demonstrations (no more than
48)
Based on the premise: [premise],
can we infer the hypothesis:
 [premise] from the premise?
Choose among Yes, Maybe, and No.
Answer: [Answer]
```

```
# test examples
Based on the premise: [premise],
can we infer the hypothesis:
 [premise] from the premise?
Choose among Yes, Maybe, and No.
Answer: [Answer]
```

Listing 5: Prompt Format for ANLI dataset, using predict-then-explain pipeline.

```
In this task, given a premise and
 a hypothesis, your job is to
determine whether the hypothesis
can be inferred from the premise.

# demonstrations (no more than
48)
[premise], Based on the previous
passage, is it true that
[hypothesis]? Choose among Yes,
Maybe, and No.
Answer: [Answer]
Explanation: [Explanation]

# test examples
[premise], Based on the previous
passage, is it true that
[hypothesis]? Choose among Yes,
Maybe, and No.
Answer: [Answer]
Explanation: [Explanation]
```

Listing 6: Prompt Format for ANLI dataset, using explain-then-predict pipeline.

```
In this task, given a premise and
 a hypothesis, your job is to
determine whether the hypothesis
can be inferred from the premise.

# demonstrations (no more than
48)
[premise], Based on the previous
passage, is it true that
[hypothesis]? Choose among Yes,
Maybe, and No.
Answer: [Answer]
Explanation: [Explanation]

# test examples
[premise], Based on the previous
passage, is it true that
[hypothesis]? Choose among Yes,
```

```
Maybe, and No.
Explanation: [Explanation]
Answer: [Answer]
```

C.1.3 ECQA & OpenbookQA

As both ECQA & OpenbookQA are multi-choice classification tasks, we use the same prompt formats for them.

Listing 7: Prompt format for multi-choice QA, standard in-context learning.

```
In this task, your job is to
first read the question as well
as the candidate choices. Then,
choose one answer from the
choices for the question.

# demonstrations (no more than
48)
Given the following options, what
 do you think is the correct
answer to the question below?
Question:  [question]
Choices: [choices]
Answer: [Answer]

# test examples
Given the following options, what
 do you think is the correct
answer to the question below?
Question:  [question]
Choices: [choices]
Answer: [Answer]
```

Listing 8: Prompt format for multi-choice QA, using predict-then-explain pipeline.

```
In this task, your job is to
first read the question as well
as the candidate choices. Then,
choose one answer from the
choices for the question.

# demonstrations (no more than
48)
Given the following options, what
 do you think is the correct
answer to the question below?
Question:  [question]
Choices: [choices]
Answer: [Answer]
Explanation: [Explanation]

# test examples
Given the following options, what
 do you think is the correct
answer to the question below?
Question:  [question]
Choices: [choices]
Answer: [Answer]
Explanation: [Explanation]
```

Listing 9: Prompt format for multi-choice QA, using explain-then-predict pipeline.

```
In this task, your job is to
first read the question as well
as the candidate choices. Then,
choose one answer from the
choices for the question.

# demonstrations (no more than
48)
Given the following options, what
 do you think is the correct
answer to the question below?
Question:  [question]
Choices: [choices]
Explanation: [Explanation]
Answer: [Answer]

# test examples
Given the following options, what
 do you think is the correct
answer to the question below?
Question:  [question]
Choices: [choices]
Explanation: [Explanation]
Answer: [Answer]
```

C.1.4 StrategyQA

Listing 10: Prompt format for StrategyQA, standard in-context learning.

```
In this task, given a question,
you need to answer True or False.
# demonstrations (no more than
48)
For the question: '[question]', do
you think it is the True or False
?
Answer: [Answer]

# test examples
```

```
For the question: '[question]', do
you think it is the True or False
?
Answer: [Answer]
```

Listing 11: Prompt format for StrategyQA, using predict-then-explain pipeline.

```
In this task, given a question,
you need to answer True or False.
# demonstrations (no more than
48)
For the question: '[question]', do
you think it is the True or False
?
Answer: [Answer]
Explanation: [Explanation]

# test examples
For the question: '[question]', do
you think it is the True or False
?
Answer: [Answer]
Explanation: [Explanation]
```

Listing 12: Prompt format for StrategyQA, using explain-then-predict pipeline.

```
In this task, given a question,
you need to answer True or False.

# demonstrations (no more than
48)
For the question: '[question]', do
you think it is the True or False
?
Explanation: [Explanation]
Answer: [Answer]

# test examples
For the question: '[question]', do
you think it is the True or False
?
Explanation: [Explanation]
Answer: [Answer]
```

C.2 Prompt Format For Explanation-aware Ensemble

Listing 13: Prompt format for LLM Scoring. Note that we use the probability of the 'Answer' token as the proxy for the quality score.

```
In this task, you will be given
the input for the [task_name]
task, your job is to determine
whether the explanation provided
is a good one for the given input
. Please consider the explanation
's coherence, informativeness,
and consistency with the
prediction to evaluate its
quality.

# demonstrations (no more than
48)
For '[task
input]', can you determine whether
 the explanation is a good one
for the given [task]?
Explanation: [Explanation]
Answer: [Answer] [Yes or No]

# test examples
For '[task
input]', can you determine whether
 the explanation is a good one
for the given [task]?
Explanation: [Explanation]
Answer: [Answer]
```

C.3 Additional Prompt Format Used in Prompt Sensitivity Study

In section 4.8, we have studied the effect of different prompt templates. Here we list them in the following lists.

Listing 14: Prompt Format 2 for E-SNLI dataset

```
In this task, given a premise and
 a hypothesis, your job is to
determine whether the hypothesis
can be inferred from the premise.

# demonstrations (no more than
48)
Based on [premise], does it follow
 that [hypothesis]? Choose among
Yes, Maybe, and No.
Answer: [Answer]
Explanation: [Explanation]

# test examples
Based on [premise], does it follow
 that [hypothesis]? Choose among
Yes, Maybe, and No.
```

```
Explanation: [Explanation]
Answer: [Answer]
```

Listing 15: Prompt Format 3 for E-SNLI dataset

```
In this task, given a premise and
 a hypothesis, your job is to
determine whether the hypothesis
can be inferred from the premise.

# demonstrations (no more than
48)
Based on the premise [premise],
can we conclude the hypothesis
that [hypothesis]? Choose among Yes
, Maybe, and No.
Answer: [Answer]
Explanation: [Explanation]

# test examples
Based on the premise [premise],
can we conclude the hypothesis
that [hypothesis]? Choose among Yes
, Maybe, and No.
Explanation: [Explanation]
Answer: [Answer]
```

Listing 16: Prompt format 2 for StrategyQA, using explain-then-predict pipeline.

```
In this task, given a question,
you need to answer True or False.

# demonstrations (no more than
48)
Answer the question: '[question]',
by True or False.
Explanation: [Explanation]
Answer: [Answer]

# test examples
Answer the question: '[question]',
by True or False.
Explanation: [Explanation]
Answer: [Answer]
```

Listing 17: Prompt format 3 for StrategyQA, using explain-then-predict pipeline.

```
In this task, given a question,
you need to answer True or False.

# demonstrations (no more than
48)
EXAM: Answer by True of False.
Question: '[question]'
Explanation: [Explanation]
Answer: [Answer]

# test examples
EXAM: Answer by True of False.
Question: '[question]'
Explanation: [Explanation]
Answer: [Answer]
```

D Human Evaluation

Here we provide the guidelines for human evaluation

Listing 18: Human Evaluation Guideline for E-SNLI dataset.

```
For this explanation grading task
, given the task input (e.g. the
premise and hypothesis for the
NLI task and the question for the
 QA task), ground-truth answer,
as well as a pair of explanations
 from the LLM, you job is to
determine which explantion will
reach the ground-truth answer for
 that input.
For the E-SNLI dataset, your task
 is to predict if the hypothesis
is entailed/neutral/contradicts
the premise.
```

Listing 19: Human Evaluation Guideline for StrategyQA dataset.

```
For this explanation grading task
, given the task input (e.g. the
premise and hypothesis for the
NLI task and the question for the
 QA task), ground-truth answer,
as well as a pair of explanations
 from the LLM, you job is to
determine which explantion will
reach the ground-truth answer for
 that input.
For the strategyQA dataset, your
task is to answer the question
with 'True' or 'False'.
```

Table 8: Comparison of Self-Consistency and EASE under similar computational costs.

LLM (→)	PaLM2-S			PaLM2-L		
Dataset (→)	E-SNLI	ECQA	StrategyQA	E-SNLI	ECQA	StrategyQA
Self-consistency	68.68	74.48	76.94	87.34	82.67	87.85
Self-consistency (Same Computation)	71.94	75.36	78.00	88.71	**83.80**	88.88
EASE	**75.01**	**75.59**	**78.23**	**89.42**	83.65	**89.90**

E Additional Results on Complexity Comparison

In our method, the additional steps — "soft probabilistic aggregation" and "Explanation Scoring" — involve generating only a single token and its probability score. This process is less time-consuming than the standard explain-then-predict step, which generates both explanations and predictions. Roughly speaking, suppose we generate N explanations for one example during the inference time, requiring $t = N * D$ time for decoding (where D is the time for generating both explanations and predictions). Then for each explanation, suppose the decoding time for generating explanation scores and soft probability are d_1 and d_2, respectively. The additional time for explanation scoring and soft probability aggregation per explanation is $t' = N * (d_1 + d_2)$. As has been discussed above, both d_1 and d_2 are much smaller than D as it only needs to decode for one token. As a result, the additional time $t' << 2t$.

To evaluate the comparative effectiveness of the self-consistency approach when augmented with additional samples to match the overall computational costs of our method. For this purpose, we utilize the PaLM2-S and PaLM2-L models as the backbone Large Language Models. The results are summarized in Table 8, providing a direct comparison between the standard self-consistency method and our EASE approach under similar computational constraints. Overall, we observe that EASE consistently outperforms the self-consistency approach in most scenarios, even when the latter is allocated additional computational resources.

F Additional Studies

F.1 Case study on explanation-aware ensemble

Tables 9 and 10 give an example of Explanation-aware Ensemble process on E-SNLI dataset. Take the Table 9 as an example, where the original prediction using majority voting is "*Entailment*". By leveraging LLM to score each explanation, the LLM is able to reduce the effect of unreliable explanations: for example, explanation 3 copies a part of the premise without providing additional reasoning, thus it receives a low score (0.369). On the contrary, explanation 4 and explanation 6 correctly identifies the key reasoning step, thus receiving higher scores (0.677 and 0.612). After weighted majority voting, the total score of "*Neutral*" class is higher than the "*Entailment*" class, thus we are able to obtain the prediction "*Neutral*", which is the ground-truth answer.

F.2 Case study on soft probability aggregation

Tables 11 and 12 provide examples of Soft Probability Aggregation applied to the ESNLI and StrategyQA Datasets, respectively. For some examples (e.g. Explanation 4 in Table 11 and Explanation 1 in 12), a mismatch between predicted labels and their respective highest probability scores is observed, indicating the potential for inconsistency in LLM predictions. By aggregating the probability score in a soft manner, we are able to rectify the final predictions, justifying the effectiveness of soft probability aggregation.

Table 9: Case study I for explanation-aware ensemble on E-SNLI Dataset.

Premise: A man is working on a computer while two people sit and talk in front. **Hypothesis**: The two people sat and chatted for a bit while the IT guy removed the virus. **Ground-truth Label**: Neutral, **Majority Voting Prediction**: Entail.	
Explanation 1: The two people cannot be sitting and talking in front while the IT guy removes the virus. **Prediction 1**: Contradict	Score 1: 0.468
Explanation 2: There is no proof that there is an IT Person removed the virus. **Prediction 2**: Neutral	Score 2: 0.562
Explanation 3: The two people were sitting and talking. **Prediction 3**: Entail	Score 3: 0.369
Explanation 4: There is no evidence that the man is an IT guy, or that he is removing a virus. **Prediction 4**: Neutral	Score 4: 0.677
Explanation 5: The two people could be talking and the IT guy could be removing the virus. **Prediction 5**: Entail	Score 5: 0.488
Explanation 6: But it's not given that the IT guy is removing a virus. **Prediction 6**: Neutral	Score 6: 0.612
Explanation 7: The man on the computer is the IT guy. **Prediction 7**: Entail	Score 7: 0.468
Explanation 8: The people sitting and talking are not related to the man working on the computer. **Prediction 8**: Contradict	Score 8: 0.447
Explanation 9: The man sitting at the computer is an IT person and is removing the virus. **Prediction 9**: Entail	Score 9: 0.455

Table 10: Case study II for explanation-aware ensemble on StrategyQA Dataset.

Question: Is Hermione Granger eligible for the Order of the British Empire? **Ground-truth Label**: False, **Majority Voting Prediction**: True.	
Explanation 1: The Order of the British Empire is a British order of chivalry rewarding contributions to the arts and sciences, work with charitable and welfare organizations and public service outside the civil service. Hermione Granger is a character in the Harry Potter series by J K. **Prediction**: True	Score 1: 0.426
Explanation 2: The Order of the British Empire is a British Order of Chivalry. Hermione Granger is a fictional character living in a world of magic and wizards. The Order of the British Empire has never been awarded to a fictional character. **Prediction**: False	Score 2: 0.655
Explanation 3: The Order of the British Empire was founded in 1917 by King George V. Hermione Granger is a British citizen To be eligible for the Order of the British Empire, a person must have contributed to the arts, science, medicine, charity, industry or government. **Prediction**: True	Score 3: 0.489
Explanation 4: Hermione Granger is a fictional character from the Harry Potter books, not a British citizen. **Prediction**: False	Score 4: 0.678
Explanation 5: Hermione Granger is a fictional character in the Harry Potter books. The Order of the British Empire is a British award that can only be received by a British subject . **Prediction**: False	Score 5: 0.650
Explanation 6: Hermione Granger is a fictional character in the Harry Potter series Order of the British Empire honors people who have made a major contribution to the arts, science, medicine, charity, or industry. Hermione Granger has contributed to the arts by authoring a book on the history of magic. **Prediction**: True	Score 6: 0.406
Explanation 7: Hermione Granger is a British citizen The Order of the British Empire is an order of chivalry. **Prediction**: True	Score 7: 0.452
Explanation 8: The Order of the British Empire is a knighthood. Hermione Granger is a Hogwarts student. **Prediction**: True	Score 8: 0.406
Explanation 9: Hermione Granger is a Hogwarts student and does not have British citizenship. **Prediction**: False	Score 9: 0.602

Table 11: Case study for Soft Probability Aggregation on ESNLI Dataset. The mismatched prediction is highlighted in red. The number in the list represents the probability score for Entail, Neutral and Contradict, respectively.

Premise: A Seattle firefighter standing in front of his truck. **Hypothesis**: The man is standing in front of the ambulance. **Ground-truth Label**: Contradict, **Majority Voting Prediction**: Neutral.	
Explanation 1: The man is standing in front of the truck and the man is standing in front of the ambulance. **Prediction**: Neutral	Score 1: [0.369, 0.419, 0.175]
Explanation 2: An ambulance is different than a truck. **Prediction**: Contradict	Score 2: [0.001, 0.042, 0.953]
Explanation 3: The Seattle firefighter is standing in front of his truck. The truck is not necessarily an ambulance. **Prediction**: Neutral	Score 3: [0.002, 0.880, 0.105]
Explanation 4: Ambulances are not fire trucks. **Prediction**: Neutral	Score 4: [0.001, 0.372, 0.614]
Explanation 5: The guy is standing in front of a truck. **Prediction**: Entail	Score 5: [0.391, 0.194, 0.378]
Explanation 6: The man is standing in front of the truck. **Prediction**: Entail	Score 6: [0.418, 0.198, 0.364]
Explanation 7: Firetrucks and ambulances are different. **Prediction**: Contradict	Score 7: [0.000, 0.004, 0.995]
Explanation 8: A Seattle firefighter is a man. **Prediction**: Entail	Score 8: [0.928, 0.036, 0.027]
Explanation 9: The firefighter isn't necessarily an ambulance driver. **Prediction**: Neutral	Score 9: [0.001, 0.552, 0.456]

Table 12: Case study for Soft Probability Aggregation on StrategyQA Dataset. The mismatched prediction is highlighted in red. The number in the list represents the probability score for False and True, respectively.

Question: Could every citizen of Samoa send a letter to a unique JPMorgan Chase employee? **Ground-truth Label**: True, **Majority Voting Prediction**: False.	
Explanation 1: Samoa has a population of about 195,000 people JPMorgan Chase has 260,000 employees. **Prediction**: True	Score 1: [0.614, 0.373]
Explanation 2: There are 158,000 citizens of Samoa JPMorgan Chase has 247,000 employees. **Prediction**: False	Score 2: [0.317, 0.673]
Explanation 3: There are roughly 275,000 people in Samoa JPMorgan Chase has roughly 250,000 employees Not all of JPMorgan Chase's employees are eligible to receive mail at work. **Prediction**: False	Score 3: [0.556, 0.443]
Explanation 4: Samoa has a population of 197 751. JPMorgan Chase has about 243 000 employees worldwide. There are 50 states in the United States **Prediction**: False	Score 4: [0.494, 0.494]
Explanation 5: There are around 194,000 people living in Samoa JPMorgan Chase employs around 250,000 people. **Prediction**: False	Score 5: [0.526, 0.464]
Explanation 6: There are 195,308 citizens of Samoa JPMorgan Chase has 257,000 employees. **Prediction**: True	Score 6: [0.180, 0.809]
Explanation 7: Samoa has a population of around 196,000 people JPMorgan Chase has more than 250,000 employees. **Prediction**: False	Score 7: [0.614, 0.372]
Explanation 8: The population of Samoa is about 188 000, J P Morgan Chase employs about 250 000 people **Prediction**: True	Score 8: [0.374, 0.616]
Explanation 9: There are 190,000 citizens of Samoa in 2017 JPMorgan Chase has over 250,000 employees. There are between 10-12 000 letters in an average day. **Prediction**: False	Score 9: [0.465, 0.527]