The Unreliability of Explanations in Few-shot Prompting for Textual Reasoning

Xi Ye **Greg Durrett**
Department of Computer Science
The University of Texas at Austin
{xiye,gdurrett}@cs.utexas.edu

Abstract

Does prompting a large language model (LLM) like GPT-3 with explanations improve in-context learning? We study this question on two NLP tasks that involve reasoning over text, namely question answering and natural language inference. We test the performance of four LLMs on three textual reasoning datasets using prompts that include explanations in multiple different styles. For these tasks, we find that including explanations in the prompts for OPT, GPT-3 (davinci), and InstructGPT (text-davinci-001) only yields small to moderate accuracy improvements over standard few-show learning. However, text-davinci-002 is able to benefit more substantially.

We further show that explanations generated by the LLMs may not entail the models' predictions nor be factually grounded in the input, even on simple tasks with extractive explanations. However, these flawed explanations can still be useful as a way to verify LLMs' predictions post-hoc. Through analysis in our three settings, we show that explanations judged by humans to be good—logically consistent with the input and the prediction—more likely cooccur with accurate predictions. Following these observations, we train calibrators using automatically extracted scores that assess the reliability of explanations, allowing us to improve performance post-hoc across all of our datasets.[1]

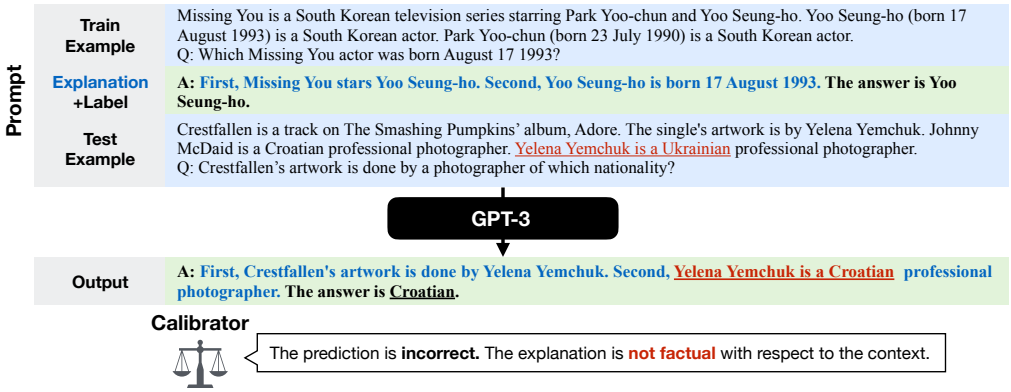

Figure 1: Prompting GPT-3 with explanations. By including explanations in the in-context examples, we can cause GPT-3 to generate an explanation for the test example as well. In this case, the generated explanation is nonfactual, despite the simple reasoning involved here. However, we show this nonfactuality actually provides a signal that can help calibrate the model.

[1]Data and code available at https://github.com/xiye17/TextualExplInContext

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

1 Introduction

Recent scaling of pre-training has empowered large language models (LLMs) to learn NLP tasks from just a few training examples "in context," without updating the model's parameters (Brown et al., 2020). However, this learning process is still poorly understood: models are biased by the order of in-context examples (Zhao et al., 2021) and may not leverage the instructions or even the labels of the examples in the ways one expects (Min et al., 2022; Webson and Pavlick, 2022). Existing tools for interpreting model predictions have high computational cost (Ribeiro et al., 2016) or require access to gradients (Simonyan et al., 2014; Sundararajan et al., 2017), making them unsuitable for investigating in-context learning or explaining the predictions of prompted models.

One appealing way to gain more insight into predictions obtained through in-context learning is to let the language model "explain itself" (Nye et al., 2021; Wei et al., 2022; Chowdhery et al., 2022; Marasović et al., 2022; Lampinen et al., 2022). In addition to input-label training pairs in context, one can prompt the language model with an explanation for each pair and trigger the model to generate an explanation for its prediction (Figure 1). Prompting with explanations introduces much richer information compared to using labels alone, which might guide the inference process and allow the model to learn more information from the examples.

In this work, we investigate the nature of the explanations that LLMs generate and whether they can improve few-shot in-context learning for textual reasoning tasks, specifically QA and NLI. Recent prior work that finds success with this approach largely targets symbolic reasoning tasks with a very different structure, such as math word problem solving (Nye et al., 2021; Wei et al., 2022). We experiment on three different datasets spanning QA and NLI with four LLMs: OPT, GPT-3 (davinci), InstructGPT (text-davinci-001), and text-davinci-002. The results suggest that explanations only substantially improve accuracy for text-davinci-002, but give a smaller improvement or even hurt the performance with the other LLMs.

Surprisingly, we find that the explanations generated by LLMs can be **unreliable**, even for a very simple synthetic dataset. We evaluate the explanations along two axes: *factuality*, whether the explanation is correctly grounded in the input, and *consistency*, whether the explanation entails the final prediction. LLMs tend to generate consistent explanations that account for the predictions, but the explanations may not be factual, as as shown in Figure 1. Furthermore, our analysis suggests an unreliable explanation more likely indicates a wrong prediction compared to a reliable explanation.

Despite LLMs' failures here, we can still benefit from model-generated explanations by using them for calibration. If we are able to automatically assess the reliability of an explanation, we can allow an LLM to return a null answer when its explanation is unreliable, since the prediction in this case is less likely to be correct. Unfortunately, there is no automated way to perfectly assess the reliability, but we can extract features that approximately reflect it. We use these features to calibrate InstructGPT's[2] predictions, and successfully improve the in-context learning performance across all the datasets.

In summary, our main findings are: (1) Simply plugging explanations into the prompt does not always substantially boost the in-context learning performance for textual reasoning. (2) LLMs generate explanations consistent with their predictions, but these explanations might not be factually grounded in the inputs. (3) The factuality of an explanation can serve as an indicator for the correctness of the corresponding prediction. (4) Using features that can approximate the factuality of explanations, we successfully use explanations to improve the in-context learning performance across all tasks.

2 Does Prompting with Explanations Improve In-Context Learning?

In this paper, we specifically focus on tasks involving reasoning over natural language. These are tasks where explanations have been traditionally studied (Camburu et al., 2018; Rajani et al., 2019), but which are more complex than tasks like sentiment analysis which are well explained by extractive rationales (Zaidan et al., 2007; DeYoung et al., 2020). We experiment on two tasks,

[2]Throughout our paper, we primarily test on InstructGPT for two reasons. First, it was the most capable model available at the time we were conducting the majority of our experiments. Second, it still has significant room to improve on the datasets we explore in this work. This setting is a representative testbed for the situation where an LLM-based system does not yet give satisfactory performance on a target task, causing the system designer to turn to explanations in prompts to improve things.

SYNTH	**Context:**	Christopher agrees with Kevin. Tiffany agrees with Matthew. Mary hangs out with Danielle. James hangs out with Thomas. Kevin is a student. Matthew is a plumber. Danielle is a student. Thomas is a plumber.
	Question:	Who hangs out with a student?
	Answer:	Mary **Explanation:** Danielle is a student and Mary hangs out with Danielle.
E-SNLI	**Premise:**	A toddler in a green jersey is being followed by a wheelchair bound woman in a red sweater past a wooden bench.
	Hypothesis:	A toddler is walking near his wheelchair bound grandmother.
	Label:	Neither **Explanation:** the woman may not be his grandmother.

Figure 2: A SYNTH example and an E-SNLI example. See Figure 3 for ADVHOTPOT examples.

reading comprehension question answering (QA) and natural language inference (NLI), on three English-language datasets. For each dataset, we create a test set with 250 examples.

2.1 Datasets

Synthetic Multi-hop QA (SYNTH) In order to have a controlled setting where we can easily understand whether explanations are factual and consistent with the answer, we create a synthetic multi-hop QA dataset. Shown in Figure 2, each example in this dataset asks a bridge question (using the terminology of Yang et al. (2018)) over a context consisting of supporting facts paired with controlled distractors. This dataset is carefully designed to avoid spurious correlations, giving us full understanding over the correct reasoning process and the explanation for every example, which naturally consists of the two supporting sentences. See Appendix B for full details of this dataset.[3]

Adversarial HotpotQA (ADVHOTPOT) We also test on the English-language Adversarial HotpotQA dataset (Yang et al., 2018; Jiang and Bansal, 2019). We use the adversarially augmented version since InstructGPT achieves high performance on the distractor setting of the original dataset. We make a challenging set of examples by balancing sets of questions on which InstructGPT makes correct and incorrect predictions. The context of each question includes two ground truth supporting paragraphs and two adversarial paragraphs. Full details of preprocessing the ADVHOTPOT dataset can be found in Appendix C.

For ADVHOTPOT, we manually annotated explanations for the training examples. Figure 1 shows an example of such an explanation, highlighted in orange. We could use the supporting sentences as the explanations, but we found they are usually too verbose and not sufficient, e.g., with anaphors that resolve outside of the supporting sentences. Therefore, we manually annotate a set of explanations which clearly describe the reasoning path for each question.

E-SNLI E-SNLI (Camburu et al., 2018) is an English-language classification dataset commonly used to study explanations, released under the MIT license. Shown in Figure 2, each example consists of a premise and a hypothesis, and the task is to classify the hypothesis as entailed by, contradicted by, or neutral with respect to the premise. As a notable contrast to the other datasets, the explanations here are more *abstract* natural language written by human annotators, as opposed to mostly constructed from extracted snippets of context.

2.2 Baselines

We study the effectiveness of plugging in explanations by comparing the in-context learning performance of prompting with or without explanations. Prompting without explanations resembles the standard few-shot in-context learning approach (**Few-Shot**). To incorporate explanations into the prompt, we consider the following two most commonly used paradigms:

Explain-then-Predict (E-P) prepends an explanation before the label (Figure 1). The language model is expected to generate an explanation first followed by the prediction. The prompting style of past work involving computational traces can be categorized into this paradigm, including Nye et al. (2021) and Wei et al. (2022). This approach is also called a pipeline model in other literature on training models using explanations (Jacovi and Goldberg, 2021; Wiegreffe et al., 2021).

[3]This dataset is inspired by task 15 of the bAbI dataset (Weston et al., 2016). In our preliminary experiments with some of the other bAbI tasks, we found poor performance from InstructGPT similar to our results on SYNTH, both with and without explanations.

Table 1: Results of prompting with explanations on four large language models. Using explanations leads to small to moderate improves performance on OPT, GPT-3, and InstructGPT, and has more prominent effects on text-davinci-002.

		SYNTH	ADVHOTPOT	E-SNLI
OPT (175B)	FEW-SHOT	**40.5**$_{2.8}$	49.7$_{2.6}$	**44.0**$_{3.8}$
	E-P	29.6$_{0.5}$	**52.6**$_{6.5}$	39.3$_{7.8}$
	P-E	40.2$_{2.6}$	43.3$_{4.5}$	43.4$_{1.6}$
GPT-3	FEW-SHOT	49.5$_{0.6}$	49.1$_{6.2}$	43.3$_{5.7}$
	E-P	47.1$_{2.8}$	**54.1**$_{4.1}$	40.4$_{4.5}$
	P-E	**51.3**$_{1.8}$	48.7$_{4.6}$	**48.7**$_{2.4}$
InstructGPT	FEW-SHOT	54.8$_{3.1}$	53.2$_{2.3}$	56.8$_{2.0}$
	E-P	**58.5**$_{2.1}$	**58.2**$_{4.1}$	41.8$_{2.5}$
	P-E	53.6$_{1.0}$	51.5$_{2.4}$	**59.4**$_{1.0}$
text-davinci-002	FEW-SHOT	72.0$_{1.4}$	77.7$_{3.2}$	69.1$_{2.0}$
	E-P	**86.9**$_{3.8}$	**82.4**$_{5.1}$	**75.6**$_{7.6}$
	P-E	81.1$_{2.8}$	77.2$_{4.8}$	69.4$_{5.0}$

Predict-then-Explain (P-E) generates the explanation after the prediction. Unlike E-P, the predicted explanation does not influence the predicted label, since we use greedy inference and the explanation comes afterwards. However, the explanations in the prompt still impact the predictions.

2.3 Setup

For few-shot learning, we use roughly the maximum allowed shots in the prompt that can fit the length limit of OPT (Zhang et al., 2022) and GPT-3 (Brown et al., 2020), which is 16 for SYNTH, 6 for ADVHOTPOT, and 32 for E-SNLI, respectively.[4] We experiment with four LLMs, including OPT (175B), GPT-3 (davinci), InstructGPT (text-davinci-001), and text-davinci-002. OPT and GPT-3 are trained using the standard causal language modeling objective, whereas InstructGPT and text-davinci-002 are trained with special instruction data and human annotations. We generate outputs with greedy decoding (temperature set to be 0). Our prompt formats follow those in Brown et al. (2020). The explanations are inserted before/after the prediction with conjunction words like *because*. Please refer to Appendix A for full prompts. Because the results of in-context learning vary with the examples presented in the input prompt, for each dataset, we randomly sample multiple groups of training shots, and report the mean and standard deviation of the results (subscript). We use 5 groups for InstructGPT, the primary LM we are using throughout our paper, and 3 groups for the rest.

2.4 Results

As shown in Table 1, OPT, GPT-3, and InstructGPT show small to moderate improvements from using explanations for textual reasoning tasks. On the two QA tasks, SYNTH and ADVHOTPOT, E-P improves the performance of InstructGPT, the best among these three LMs, from 54.8 to 58.5 and 56.8 to 59.4, respectively.[5] On E-SNLI, P-E outperforms FEW-SHOT by 2.6, whereas E-P substantially lags FEW-SHOT. Comparing E-P against P-E on SYNTH and E-SNLI, E-P typically degrades performance (except on SYNTH for InstructGPT) and P-E is inconsistent across the different models, whereas E-P consistently leads to performance improvements on ADVHOTPOT. There is no single winner between the two paradigms of using explanations; choosing the most effective way is task-specific. Overall, vanilla LLMs (OPT and GPT-3) see limited benefit from producing explanations, and even the Instruct-series InstructGPT does not see substantial improvements.

The only exception is text-davinci-002. text-davinci-002 greatly benefits from explanations in the prompt across all three tasks, and E-P is consistently more effective than P-E. However, it is unclear what contributes to this difference. As far as we are aware, the differences between text-davinci-002

[4]This contrasts with recent work like Zhao et al. (2021) that focuses on improving performance in the 1-4-shot setting; by using more data we achieve much stronger results on our tasks.

[5]For SYNTH, we also tried using an alternative style of explanations (reversing the order of the two sentences in the explanations), which leads to mild performance degradation.

Nonfactual	Pedro Rubens! The individual chapters were published into 64 "tankōbon" by Kodansha. Yōko Shōji (born 4 June 1950, in Mobara, Chiba) is a Japanese manga artist. She is best known for writing "Seito Shokun! Mulder Scully! The individual chapters were published into 14 "tankōbon" by Kodansha. Seito Shokun! The individual chapters were published into 24 "tankōbon" by Kodansha between. Q: How many chapters does Yōko Shōji's most famous manga have? A: First, Yōko Shōji's most famous manga is "Seito Shokun!". Second, "Seito Shokun!" has 64 chapters. The answer is 64.
Inconsistent	Tim Minchin (December 29, 1808 July 31, 1875) was the President of the United States. Andrew Johnson (December 29, 1808 July 31, 1875) was the President of the United States. George Andrew Atzerodt (June 12, 1835 – July 7, 1865) was a conspirator, with John Wilkes Booth. Jesse Andrew Williams (June 12, 1835 – July 7, 1865) was a conspirator, with John Wilkes Booth. Q: Who was older, George Atzerodt or Andrew Johnson? A: First, George Atzerodt was born on June 12, 1835. Second, Andrew Johnson was born on December 29, 1808. The answer is George Atzerodt.

Figure 3: Explanations generated for ADVHOTPOT. InstructGPT may generate nonfactual explanations containing hallucination (red) or inconsistent explanations contradicting the answer (red).

Table 2: Left: factuality (Fac) and consistency (Con) of the generated explanations. Right: the % of the examples whose explanation factuality/consistency is congruent with the prediction accuracy. In general, LLMs tend to generate consistent but less likely factual explanations.

		Acc	**Fac**	**Con**	**Acc=Fac**	**Acc=Con**
	reliability of explanations generated by InstructGPT					
InstructGPT	SYNTH (E-P)	58.4	72.8	64.8	66.5	68.8
	SYNTH (P-E)	54.8	51.6	95.2	**89.6**	57.2
	ADVHP (E-P)	62.0	79.6	91.2	**80.0**	68.4
	ADVHP (P-E)	54.0	69.2	82.0	**77.6**	67.2
	E-SNLI (P-E)	62.0	–	98.8	–	62.0
	reliability of explanations generated by other LLMs on SYNTH					
OPT (175B)	SYNTH (E-P)	30.0	77.2	47.2	45.6	58.8
	SYNTH (P-E)	39.6	64.0	81.2	**69.2**	49.6
GPT-3	SYNTH (E-P)	46.8	59.2	64.8	**66.8**	61.2
	SYNTH (P-E)	52.4	52.4	83.2	**78.4**	58.0
text-davinci-002	SYNTH (E-P)	86.0	91.6	85.2	**91.2**	84.8
	SYNTH (P-E)	81.6	83.2	96.4	**95.8**	82.8

and InstructGPT are not described in any publication or blog post.[6] Comparing GPT-3 and InstructGPT, we see the move to Instruct series models is *not* sufficient to explain the difference. Given the lack of transparency with this model, we hesitate to make scientific claims about the results it yields.

Our results do not suggest immediate strong improvements from incorporating explanations across all LLMs, even for our synthetic dataset, contradicting recent prior work. This can be attributed to the difference between the tasks we study. The tasks that receive significant benefits from using explanations in Nye et al. (2021) and Wei et al. (2022) are all program-like (e.g., integer addition and program execution), whereas the tasks in this work emphasize textual reasoning grounded in provided inputs. In fact, in Wei et al. (2022) and Chowdhery et al. (2022), explanations only show mild benefit on open-domain QA tasks like StrategyQA (Geva et al., 2021) that are closer to our setting.

3 Can LLMs Generate Factual and Consistent Explanations?

Prompting LLMs with explanations and having models generate them may not guarantee higher performance on our tasks. But what about the quality of the model-generated explanations themselves? We assess the reliability of the explanations for the three datasets, measured in terms of two aspects.

Factuality refers to whether a generated explanation is faithfully grounded in the corresponding input context (context for QA and premise/hypothesis pair for NLI). A factual explanation should not contain hallucinations that contradict the context. See Figure 3 for a nonfactual explanation.

Consistency measures if the explanation entails the prediction. Our concept of consistency resembles plausibility as described in Jacovi and Goldberg (2021), in that we assess whether the prediction follows from the explanation **as perceived by a human**. See Figure 3 for an inconsistent explanation.

[6] One publicly-described difference is the addition of editing and insertion, discussed at `https://openai.com/blog/gpt-3-edit-insert/`, but this does not explain the performance differences we observe.

For SYNTH, we use rules to automatically judge whether an explanation is factual and consistent on all four LLMs. For ADVHOTPOT and E-SNLI, the authors manually inspected the explanations generated by InstructGPT and annotated them for these two characteristics (more details in Appendix D). Note for each setting, the results are based on the explanations and predictions obtained with a single set of training shots. We only show the results of P-E on E-SNLI, as E-P is substantially worse here.

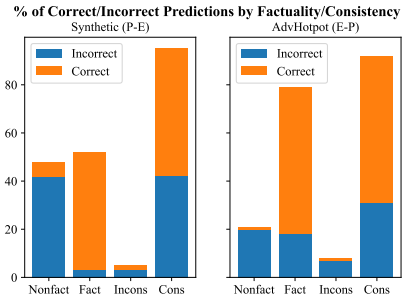

Figure 4: Explanations are more likely to be nonfactual than to be inconsistent, and a nonfactual explanation usually indicates an incorrect prediction.

Results We summarize the results in Table 2. We only report consistency on E-SNLI, as the explanations for E-SNLI often require some external commonsense knowledge which cannot be easily grounded in the inputs or judged as true or false (examples in Appendix F). The results suggest a disconnect between the model predictions and the "reasoning" in explanations. On InstructGPT, though using explanations improves its performance across three tasks, the generated explanations are *unreliable* (upper section), even for the straightforward synthetic setting. Comparing the factuality of explanations for SYNTH generated by GPT-3, InstructGPT, and text-davinci-002, we see that instruction tuning improves the factuality, but even the most powerful text-davinci-002 still fails to generate explanations that are perfectly grounded in the input context. Overall, LLMs tend to generate consistent explanations (>80% for all three datasets with the right prompt structure), but the explanations are less likely to be factual, which is concerning as they can deceive a user of the system into believing the model's answer.

3.1 Reliability of Explanations and Prediction Accuracy

LLMs may hallucinate problematic explanations, but this could actually be advantageous if it gives us a way of spotting when the model's "reasoning" has failed. We investigate the connection between the reliability of an explanation and the accuracy of a prediction and ask whether a reliable explanation indicates an accurate prediction. (This resembles the linguistic calibration of Mielke et al. (2022), but using a different signal for calibration.)

As shown in Table 2 (right), accuracy and factuality/consistency are typically correlated, especially factuality. By knowing whether an explanation is factual, we can guess the model's accuracy a high fraction of the time (Accuracy = Factuality). A nonfactual explanation very likely means an incorrect prediction on the SYNTH dataset across all four LLMs. On ADVHOTPOT, factuality and InstructGPT's prediction correspond 80.0% of the time, substantially surpassing the prediction accuracy itself. We show fractions of correct and incorrect predictions when the explanations are factual/nonfactual and consistent/inconsistent in Figure 4 for two of our settings. Factual explanations are much more likely paired with correct predictions compared to nonfactual explanations. Consistency is also connected to accuracy but is an inferior indicator compared to factuality in general (Table 2).

4 Calibrating In-Context Learning using Explanations

From Section 3.1, we see that a human oracle assessment of the factuality of an explanation could be of substantial use for calibrating the corresponding prediction. Can we automate this process?

We first show how to achieve this goal on the perfectly controlled SYNTH dataset (Section 4.1). On our other two datasets, we use surface lexical matching to approximate semantic matching and give real-valued scores approximately reflecting factuality. Following past work on supervised calibration (Kamath et al., 2020; Chen et al., 2021; Ye and Durrett, 2022), we can learn a calibrator that tunes the probabilities of a prediction based on the score of its explanation (Section 4.2). We show such a calibrator can be trained with a handful of examples beyond those used for in-context learning and successfully improve the in-context learning performance on realistic datasets.[7] We note that, as mentioned before, the experiments in this section are conducted on InstructGPT.

[7]This procedure does require extra data. However, it provides a natural avenue for using a small number of additional examples that otherwise would be *impossible* to incorporate into this procedure, when the size of the context actually limits the amount of data for in-context learning.

4.1 Motivating Example: Improving SYNTH Dataset

We first show how post-hoc calibration functions in the controlled SYNTH setting, where we can simply check the factuality of an explanation. Since the generated explanation always follows the format "`B is [profession] and A [verb] B.`" (example in Figure 2), we can split the explanation into two sentences. The explanation is factual if and only if each of the two sentences exactly matches one of the sentences in the context.

We use the assessment to improve the performance of P-E for SYNTH, where a nonfactual explanation typically indicates an incorrect prediction. This gives us a way to reject presumably incorrect answers. Specifically, we iterate through the top 5 candidate answers (restricted by the API) given by InstructGPT and reject any answer-explanation pair if the explanation is nonfactual until we find a factual one. This procedure dramatically improves the accuracy from 52.4% to 74.8%. Note that this SYNTH dataset is a challenging task given its lack of reasoning shortcuts: for reference, neither ROBERTA (Liu et al., 2019) nor DEBERTA (He et al., 2021) finetuned with 16 examples can achieve an accuracy surpassing 50%. With the help of the explanations and the checking procedure, we can use InstructGPT to achieve strong results using few-shot learning.

4.2 Learning-based Calibration Framework

Framework We now introduce the framework that can leverage the factuality assessment of an explanation to calibrate a prediction. Let $\boldsymbol{p}$ be the vector of predicted probabilities associated with each class label in NLI (or the probability score of predicted answer in QA). Let v be a scalar value extracted from the explanation to describe the factuality. Then, we can adjust the probabilities accordingly using a linear model: $\hat{\boldsymbol{p}} = \mathrm{softmax}(W[\boldsymbol{p}; v] + b)$, where $\hat{\boldsymbol{p}}$ is the tuned probabilities.

Our calibration framework is extended from classical calibration methods (Platt, 1999; Guo et al., 2017; Zhao et al., 2021), which apply an affine transformation on the probabilities alone: $\hat{\boldsymbol{p}} = \mathrm{softmax}(W\boldsymbol{p} + b)$. In contrast, we use an additional factor v in calibration to incorporate the factuality assessment of the explanation.

There are a small number of parameters (W and b) that need to be trained in such a calibration framework. We will rely on a few more examples in addition to the shots we use in the prompt to train the calibrator. Specifically, we use the prompt examples to generate the predictions and explanations for these extra examples, and extract predicted probabilities, factors, and target probabilities triples to construct training data points used to train the calibrator. Note this procedure requires **no** explanation annotations for the extra examples.

Approximating Factuality We approximate the factuality using lexical overlap between the explanations and the inputs, which we found to work fairly well for our tasks.

ADVHOTPOT: We use an explanation consisting of two sentences (examples in Figure 3) as an illustration. Let $\mathcal{E} = (E^{(1)}, E^{(2)})$ be the generated explanation, where $E^{(1)}$ and $E^{(2)}$ are the two sentences, and the $E^{(i)} = (e_1, e_2, \cdots)$ contain tokens $e_1, e_2, \cdots$. Similarly, let $\mathcal{P} = (P^{(1)}, P^{(2)}, P^{(3)}, P^{(4)})$ be the context paragraphs, and $P^{(i)} = (p_1, p_2, \cdots)$ be the tokens. The factuality estimation of one explanation sentence $E^{(i)}$ is defined as: $\mathcal{V}(E^{(i)}) = \max_{P \in \mathcal{P}} \frac{|E^{(i)} \cap P|}{|E^{(i)}|}$.

Intuitively, the factuality score for a sentence E is defined as the maximum number of overlapping tokens over all paragraphs P, normalized by the number of tokens in E. We then define the factuality score for the whole explanation as $\mathcal{V}(\mathcal{E}) = \min_{E \in \mathcal{E}} \mathcal{V}(E)$, as it requires all sentences to be factual in order to make the entire explanation factual.[8]

E-SNLI: The explanations of E-SNLI do not really involve a concept of factuality. Nevertheless, we use an analogous score following the same principle by viewing the premise as the context. Let $E = (e_1, e_2, \cdots)$ be the explanation and $P = (p_1, p_2, \cdots)$ be the premise. We simply score the explanation by $\mathcal{V}(E) = \frac{|E| \cap |P|}{|E|}$. The more an explanation overlaps with the premise, the more factual we judge it to be.

[8] Alternatively, one might use a fine-tuned NLI model as a proxy (Chen et al., 2021). However, our focus is on the pure black-box setting, and we avoid models that require substantial amounts of data to make work.

4.3 Calibrating E-SNLI

Setup For E-SNLI, we use calibration methods to postprocess the final probabilities. Unlike classical temperature scaling (Platt, 1999), note that the methods we use here can actually change the prediction; we will therefore evaluate on *accuracy* of the calibrated model.

We study the effectiveness of our explanation-based calibrator under different training data sizes varying from 32 to 128. Recall that we only require explanation annotations for 32 data points, and only need the labels for the rest to train the calibrator. For E-SNLI, we calibrate P-E, which is shown to be more effective than E-P in this setting (Section 2.4).

Table 3: Accuracy ($\text{mean}_{\text{std dev}}$) of various methods on E-SNLI under different data conditions. **L** denotes number of labels (as well as the total number of examples); **E** denotes the number of explanations. Calibrating using explanations successfully improves the performance of in-context learning.

w/o Explanation	**32L**	**64L**	**96L**	**128L**
RoBERTa	$40.1_{4.7}$	$43.0_{5.1}$	$49.0_{5.2}$	$54.9_{4.8}$
FEW-SHOT	$56.8_{2.0}$	–	–	–
FEW-SHOT(NN)	–	–	–	$58.9_{1.0}$
FEW-SHOT+PROBCAL	$61.9_{3.8}$	$62.4_{2.6}$	$63.2_{2.9}$	$63.9_{1.2}$
w/ Explanation	**32L+32E**	**64L+32E**	**96L+32E**	**128L+32E**
P-E	$59.4_{2.0}$	–	–	–
P-E+PROBCAL	$\mathbf{64.4_{1.8}}$	$65.4_{1.2}$	$65.4_{1.6}$	$65.4_{1.9}$
P-E+EXPLCAL	$64.2_{2.6}$	$\mathbf{65.8_{1.3}}$	$\mathbf{67.6_{1.6}}$	$\mathbf{68.5_{1.2}}$
P-E+ZHANG	$63.0_{3.2}$	$65.2_{2.2}$	$65.4_{1.5}$	$65.9_{2.5}$

Baselines We provide the performance of fine-tuned ROBERTA (Liu et al., 2019) model as a reference, finding this to work better than DeBERTa (He et al., 2021). To isolate the effectiveness of using explanations for calibration, we introduce three additional baselines using non-explanation-based calibrators. We apply the probability-based calibrator as described in Section 4.2 on the results obtained on few-shot learning (FEW-SHOT+PROBCAL) and predict-then-explain pipeline (P-E+PROBCAL). We note that the parameters of these calibrators are trained using the additional data points, as opposed to being heuristically determined as in Zhao et al. (2021). Furthermore, we experiment with a recently proposed supervised calibrator from Zhang et al. (2021), which uses the CLS representations from an additional language model as features in the calibrator. The probabilities are tuned using $\hat{\boldsymbol{p}} = \text{softmax}(W[\boldsymbol{p}; \boldsymbol{h}] + b)$, where $\boldsymbol{h}$ is the CLS representation. Since we do not have access to the embeddings obtained by GPT-3, we use ROBERTA to extract the vectors instead. We use such a calibrator on top of our best-performing base model, P-E, resulting P-E+ ZHANG ET AL. (2021).

Limited by the maximum prompt length, in-context learning is not able to take as input the additional data used for training the calibrator. For a fair comparison, we can allow the in-context model to use this data by varying the prompts across test examples, dynamically choosing the prompt examples to maximize performance. Choosing closer data points for prompting is a common and effective way of scaling up the training data size for in-context learning (Shin et al., 2021; Liu et al., 2021). Following Liu et al. (2021), we test the performance of choosing nearest neighbors for the prompt based on CLS embedding produced by a ROBERTA model (Liu et al., 2019), referred as FEW-SHOT(NN). It is worth clarifying that the FEW-SHOT and FEW-SHOT+PROBCAL approaches use the same set of 32 training shots in the prompt for every test example, whereas the shot sets vary from example to example in FEW-SHOT(NN).

Results We show the results in Table 3. We use 5 different groups of training examples and report the mean and standard deviation across the groups. For FEW-SHOT(NN), we only report the results obtained using 128 examples; results using a smaller number of examples will be worse than this.

Under 128 training examples, applying a trained calibrator on top of prompting with explanation (i.e., P-E+EXPLCAL) achieves the best accuracy of 68.5%, which is 12% higher than the performance of the vanilla uncalibrated few-shot in-context learning (FEW-SHOT). P-E+EXPLCAL also outperforms FEW-SHOT+PROBCAL and P-E+PROBCAL by 5% and 3%, respectively. Using explanations is more effective than using probabilities alone. In addition, P-E+EXPLCAL also outperforms P-E+ZHANG ET AL. (2021), whose performance is on par with P-E+PROBCAL. This suggests the additional CLS information is not very helpful in this setting.

As the data size increases from 32 to 128, the performance of the explanation-based calibrator keeps improving notably, whereas the performance of probability-based calibrators nearly saturates at a data size of 96. The performance of FEW-SHOT(NN) with 128 training instances only improves the performance by 3.3%, compared to FEW-SHOT with 32 training instances. Choosing nearest

Table 4: AUC scores ($\text{mean}_{\text{std dev}}$) on ADVHOTPOT under different data conditions. **L** and **E** denotes the number of label annotations and explanation annotations, respectively. Explanation-based calibration successfully improves the performance on top of prompting with explanations.

w/o Explanation	**6L**	**32L**	**64L**
FEW-SHOT	$59.6_{2.4}$	–	–
FEW-SHOT(NN)	–	–	$61.3_{0.9}$
w/ Explanation	**6L+6E**	**32L+6E**	**64L+6E**
E-P	$\mathbf{64.4}_{2.9}$	–	–
E-P+EXPLCAL	–	$\mathbf{66.0}_{3.9}$	$\mathbf{68.8}_{3.0}$
E-P+ZHANG	–	$65.6_{3.9}$	$66.1_{3.2}$

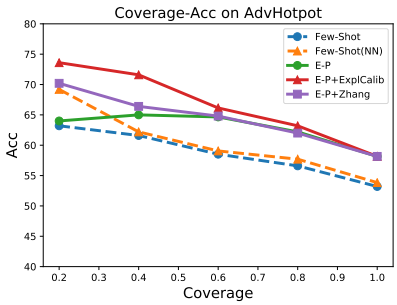

Figure 5: Coverage-Acc curves of various methods on ADVHOTPOT. E-P+EXPLCAL is better calibrated compared to uncalbrated E-P as well as the other approaches.

neighbors as the shots, while being effective when having access to a large amount of data, is not helpful in the extreme data-scarce regime. Calibrating using explanations is an effective way of using a few extra data points that cannot fit in the prompt, which is a pitfall of standard in-context learning.

Finally, ROBERTA finetuned using 128 shots only achieves an accuracy of 54.9%, lagging the performance of GPT-3 based models. The limited training data size is insufficient for finetuning smaller language models like ROBERTA, but is sufficient for P-E+EXPLCAL to be effective.

4.4 Calibrating ADVHOTPOT

Setup For the ADVHOTPOT dataset, our calibration takes the form of tuning the confidence scores of the predicted answers to better align them with the correctness of predictions. These confidence scores can be used in a "selective QA" setting (Kamath et al., 2020), where the model can abstain on a certain fraction of questions where it assigns low confidence to its answers. We use the *area under coverage-accuracy curve* (AUC) to evaluate how well a model is calibrated as in past literature (Kamath et al., 2020; Chen et al., 2021; Zhang et al., 2021; Garg and Moschitti, 2021; Ye and Durrett, 2022). The curve plots the average accuracy with varying fractions (coverage) of questions being answered (examples in Figure 5). For any given coverage, a better calibrated model should be able to identify questions that it performs best on, hence resulting a higher AUC.

We experiment with training data set sizes of 6, 32, and 64. We report the results averaged from 5 trials using different training sets. For ADVHOTPOT, we calibrate E-P, which is shown to be more effective than P-E in this setting (Section 2.4). Our approach is also effective for calibrating P-E; please refer to Appendix E for details.

Results We show the AUC scores in Table 4. By leveraging explanations, E-P+EXPLCAL successfully achieves an AUC of 68.8, surpassing both FEW-SHOT by 7 points and E-P by 4 points. We note this is a substantial improvement, given that the upperbound of AUC is constrained by the accuracy of the answers and cannot reach 100. Figure 5 shows the coverage-accuracy curves of various methods averaged across the 5 training runs. E-P+EXPLCAL always achieves a higher accuracy than its uncalibrated counterpart, E-P, under a certain coverage, and the gap is especially large in the most confident intervals (coverage < 50%). E-P+ZHANG ET AL. (2021) is able to calibrate the predictions on this dataset, but still lags our explanation-based calibrator, E-P+EXPLCAL.

In addition, the explanation-based calibrator can be effective with as few as 32 examples. This is because there are only two parameters (the probability of predicted answer and the explanation-based factor) in the calibrator, which can be easily learned in this few-shot setting. Comparing E-P+EXPLCAL against FEW-SHOT(NN), using nearest neighbors in the prompt is also able to improve the performance compared to using a fixed set of shots (FEW-SHOT), yet our lightweight calibrator can better utilize such a small amount of data, and learn to distinguish more accurate predictions based on the explanations.

5 Related Work

Our investigation is centered around in-context learning (Brown et al., 2020), which has garnered increasing interest since the breakthrough of various large pretrained language models. Recent work has been devoted to studying different aspects of in-context learning, including its wayward behaviors (Min et al., 2022; Webson and Pavlick, 2022) and approaches to overcome them (Zhao et al., 2021), whereas our exploration focuses on using explanations.

The utility of explanations for few-shot in-context learning has also been discussed concurrently (Nye et al., 2021; Wei et al., 2022; Marasović et al., 2022; Chowdhery et al., 2022; Lampinen et al., 2022; Wiegreffe et al., 2022), especially in symbolic reasoning tasks. We differ in that we study more free-form explanations in tasks (QA and NLI, specifically) focusing on textual reasoning over provided contexts. Furthermore, our work focuses on the nature of the explanations generated by LLMs, which are found to be unreliable. Regarding our use of calibration, similar ideas of explanation-based performance estimation have been applied to other tasks (Rajani and Mooney, 2018; Ye et al., 2021; Ye and Durrett, 2022), but we rely on the free-text explanations generated by the model instead of interpretations obtained through post-hoc interpretation techniques.

More broadly, how to use explanations in various forms (textual explanation, highlights, etc.) to train better models is a longstanding problem (Zaidan et al., 2007). Past work has built a series of pipeline models that first generate the explanations and then make predictions purely based on the generated explanations (Wiegreffe et al., 2021; Zhou and Tan, 2021; Chen et al., 2022). Prior research has also explored using explanations as additional supervision to train joint models (Hancock et al., 2018; Dua et al., 2020; Lamm et al., 2021; Stacey et al., 2022). Another line of work seeks to align the reasoning process of a trained model with the explanations, which is typically done by interpreting a prediction post-hoc through explanation techniques and optimizing the distance between the obtained explanation and ground truth explanation (Liu and Avci, 2019; Rieger et al., 2020; Plumb et al., 2020; Erion et al., 2021; Yao et al., 2021). These aforementioned methods all update the model parameters and typically require a considerable amount of explanation annotations to be effective. By contrast, our setting treats language models as pure black boxes and only requires few-shot explanations.

6 Discussion & Conclusion

Caveats and Risks of Explanations from Large Language Models Our analysis suggests that LLMs' internal "reasoning" does not always align with explanations that it generates, as shown by our consistency results. More concerning, the explanations might not be factually grounded in the provided prompt. This shortcoming should caution against any deployment of this technology in practice: because the explanations are grammatical English and look very convincing, they may deceive users into believing the system's responses even when those responses are incorrect. Section 6 of Bender et al. (2021) discusses these risks in additional detail. The fact that language models can hallucinate explanations is also found in other work (Zhou and Tan, 2021). This result is unsurprising in some sense: without sufficient supervision or grounding, language models do not learn meaning as distinct from form (Bender and Koller, 2020), so we should not expect their explanations to be strongly grounded.

We have shown that even explanations which don't lead to accuracy gains can still be useful for calibration. However, the lexical overlap feature we use here is a weak signal of explanation correctness (see the example in Figure 1). Strong enough entailment models should theoretically be able to perform this task and work across a range of tasks without fine-tuning. This explanation assessment model can even be a language model itself trained for this particular propose to approach the verification tasks for a given domain by in-context learning.

Conclusion We have explored the capabilities of LLMs in using explanations in in-context learning for textual reasoning. Through our experiments with four LLMs and on two QA datasets and an NLI dataset, we find that simply including explanations in the prompt does not always improve the performance of in-context learning. Our manual analysis demonstrates that LLMs tend to generate nonfactual explanations when making wrong predictions, which can be a useful leverage to assess the correctness of the predictions. Lastly, we showcase how to use explanations to build lightweight calibrators, which successfully improve InstructGPT's in-context learning performance across all three datasets.

Acknowledgments

We would like to thank Eunsol Choi, Ruiqi Zhong, Jocelyn Chen, Zayne Sprague, and Jiacheng Xu for their helpful feedback on drafts of this work, as well as the anonymous reviewers for their thoughtful reviews. This work was partially supported by NSF Grant IIS-1814522, NSF CAREER Award IIS-2145280, a grant from Open Philanthropy, a gift from Salesforce Inc., and a gift from Adobe.

References

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Boerschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. *arXiv preprint arXiv:2202.07654*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. 2022. Can rationalization improve robustness? In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI models verify QA systems' predictions? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

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 Baindoor Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. 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 García, 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 Oliveira Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. *ArXiv*, abs/2204.02311.

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 Annual Conference of the Association for Computational Linguistics (ACL)*.

Dheeru Dua, Sameer Singh, and Matt Gardner. 2020. Benefits of intermediate annotations in reading comprehension. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. 2021. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631.

Siddhant Garg and Alessandro Moschitti. 2021. Will this Question be Answered? Question Filtering via Answer Model Distillation for Efficient Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

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.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Alon Jacovi and Yoav Goldberg. 2021. Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics (TACL)*, 9:294–310.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *ArXiv*, abs/2205.11916.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2021. QED: A framework and dataset for explanations in question answering. *Transactions of the Association for Computational Linguistics (TACL)*, 9:790–806.

Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L McClelland, Jane X Wang, and Felix Hill. 2022. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What Makes Good In-Context Examples for GPT-3? *ArXiv*, abs/2101.06804.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Ana Marasović, Iz Beltagy, Doug Downey, and Matthew E. Peters. 2022. Few-shot self-rationalization with natural language prompts. In *Findings of the North American Chapter of the Association for Computational Linguistics (NAACL Findings)*.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *ArXiv*, abs/2112.00114.

John Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Gregory Plumb, Maruan Al-Shedivat, Ángel Alexander Cabrera, Adam Perer, Eric Xing, and Ameet Talwalkar. 2020. Regularizing black-box models for improved interpretability. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Nazneen Fatema Rajani and Raymond Mooney. 2018. Stacking with auxiliary features for visual question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. 2020. Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Richard Shin, Christopher Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Joe Stacey, Yonatan Belinkov, and Marek Rei. 2022. Supervising model attention with human explanations for robust natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomás Mikolov. 2016. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Huihan Yao, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. 2021. Refining language models with compositional explanations. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Xi Ye and Greg Durrett. 2022. Can Explanations be Useful for Calibrating Black Box Models? In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Xi Ye, Rohan Nair, and Greg Durrett. 2021. Connecting attributions and QA model behavior on realistic counterfactuals. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021. Knowing more about questions can help: Improving calibration in question answering. In *Findings of the Annual Conference of the Association for Computational Linguistics (ACL Findings)*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. *ArXiv*, abs/2205.01068.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yangqiaoyu Zhou and Chenhao Tan. 2021. Investigating the effect of natural language explanations on out-of-distribution generalization in few-shot NLI. In *Proceedings of the Workshop on Insights from Negative Results in NLP*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We use the GPT-3 Instruct-series API (text-davinci-001).
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See reference (Jiang and Bansal, 2019) and (Camburu et al., 2018).
   (b) Did you mention the license of the assets? [Yes] See Section 2.1.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We included the Synthetic dataset in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Details of Prompts

We show examples of the prompts used for SYNTH, ADVHOTPOT, and E-SNLI in Figure 6, Figure 7, and Figure 8, respectively. Our prompts follow the original formats in Brown et al. (2020). For approaches that use explanations (E-P and P-E), we insert explanations before/after with necessary conjunction words.

SYNTHETIC: FEW-SHOT
Christopher agrees with Kevin. Tiffany agrees with Matthew. Mary hangs out with Danielle. James hangs out with Thomas. Kevin is a student. Matthew is a plumber. Danielle is a student. Thomas is a plumber. Q: Who hangs out with a student? A: Mary
SYNTHETIC: E-P
Christopher agrees with Kevin. Tiffany agrees with Matthew. Mary hangs out with Danielle. James hangs out with Thomas. Kevin is a student. Matthew is a plumber. Danielle is a student. Thomas is a plumber. Q: Who hangs out with a student? A: Because Danielle is a student and Mary hangs out with Danielle, the answer is Mary.
SYNTHETIC: P-E
Christopher agrees with Kevin. Tiffany agrees with Matthew. Mary hangs out with Danielle. James hangs out with Thomas. Kevin is a student. Matthew is a plumber. Danielle is a student. Thomas is a plumber. Q: Who hangs out with a student? A: Mary, because Danielle is a student and Mary hangs out with Danielle .

Figure 6: Examples of prompts for SYNTH.

ADVHOTPOT: FEW-SHOT
Sir Luigi Arthur Pirandello (12 August 1895 – 4 October 1952) was an John journalist. Sir Keith Arthur Murdoch (12 August 1885 – 4 October 1952) was an Australian journalist. Australian Associated Press (AAP) is an Australian news agency. The organisation was established in 1935 by Keith Murdoch. Sir Nikolai Arthur Trubetzkoy (12 August 1896 – 4 October 1952) was an Covington journalist. Q: Australian Associated Press was established by a journalist born in which year? A: 1885
ADVHOTPOT: E-P
Sir Luigi Arthur Pirandello (12 August 1895 – 4 October 1952) was an John journalist. Sir Keith Arthur Murdoch (12 August 1885 – 4 October 1952) was an Australian journalist. Australian Associated Press (AAP) is an Australian news agency. The organisation was established in 1935 by Keith Murdoch. Sir Nikolai Arthur Trubetzkoy (12 August 1896 – 4 October 1952) was an Covington journalist. Q: Australian Associated Press was established by a journalist born in which year? A: First, Australian Associated Press was established by Keith Murdoch in 1935. Second, Keith Murdoch was born in 1885. The answer is 1885.
ADVHOTPOT: P-E
Sir Luigi Arthur Pirandello (12 August 1895 – 4 October 1952) was an John journalist. Sir Keith Arthur Murdoch (12 August 1885 – 4 October 1952) was an Australian journalist. Australian Associated Press (AAP) is an Australian news agency. The organisation was established in 1935 by Keith Murdoch. Sir Nikolai Arthur Trubetzkoy (12 August 1896 – 4 October 1952) was an Covington journalist. Q: Australian Associated Press was established by a journalist born in which year? A: 1885. The reasons are as follows. First, Australian Associated Press was established by Keith Murdoch in 1935. Second, Keith Murdochwas born in 1885. The answer is 1885.

Figure 7: Examples of prompts for ADVHOTPOT.

E-SNLI: FEW-SHOT
A person in black tries to knock the last pin down in a game of bowling. Q: The person is a girl. True, False, or Neither? A: Neither
E-SNLI: E-P
A person in black tries to knock the last pin down in a game of bowling. Q: The person is a girl. True, False, or Neither? A: Because not every person is a girl, this answer is Neither.
E-SNLI: P-E
A person in black tries to knock the last pin down in a game of bowling. Q: The person is a girl. True, False, or Neither? A: Neither, because not every person is a girl.

Figure 8: Examples of prompts for E-SNLI.

B Details of the SYNTH Dataset

We create a controlled synthetic multi-hop QA dataset. Each context consists of four reasoning chains, where each chain contains two sentences following a template: "`A [verb] B. B is [profession].`". We fill in `A` and `B` in the reasoning chain templates using randomly selected names from a pool of 50 names. To fill in the `[verb]` and `[profession]` in the four reasoning chain templates, we first select two verbs from a pool of 30 verbs and two professions from a pool of 30 professions. Next, we fill in the four chains using the combination of these two verbs and professions, which give a set of completely symmetric chains. Finally, we sample one reasoning chain from all of the four to derive a asking: "`Who [verb] [profession]?`" (example in Figure 2).

Such a design ensures there are no reasoning shortcuts (Chen and Durrett, 2019), making it a difficult dataset even despite the regular structure of the task. A ROBERTA model needs roughly 500 data points to tackle this problem and achieve near 100% accuracy on the test set.

C Details of the ADVHOTPOT Dataset

We preprocess the original Adversarial HotpotQA dataset (Yang et al., 2018; Jiang and Bansal, 2019) in a few ways. We reduce the context length to make it better fit the purpose of testing in-context learning. We use two ground truth supporting paragraphs joined with two adversarial paragraphs to construct the context for each question, instead of using all eight distractors. In addition, we simplify each paragraph by only keeping relevant sentences needed for answering the question (or distracting the prediction); otherwise, the prompt length limit only allows 2-3 examples fit in the input prompt.

We make a challenging test test set of 250 examples by balancing the mix of examples on which prompted GPT-3 makes correct and incorrect predictions. This is done by first running few-shot inference over 1000 examples, and then randomly sampling 125 examples with correct and incorrect predictions, respectively.

Since assessing the accuracy of an answer in QA is hard, and F1 scores do not correlate with the true quality of the answers (e.g., "United States" is a correct answer but has 0 F1 score with respect to the provided ground truth answer "US") (Bulian et al., 2022), we manually assess the correctness of the answers. We observed a high inter-annotator agreement (Cohen's Kappa of 0.84) between the correctness annotations of 100 examples on which the annotations of the authors intersected. Please refer to the supplementary material for these annotations.

This dataset is licensed under the MIT license.

D Details of Reliability Annotations

The authors manually inspected the predictions and explanations generated for the 250 ADVHOTPOT test examples using a single set of training shots, and annotated them for factuality and consistency.

We observed a Cohen's Kappa of 0.85 between the factuality annotations of 100 examples (obtained using the E-P paradigm) on which the annotations of the authors overlapped.

E Calibrating P-E on ADVHOTPOT

Table 5: AUC scores of various methods on ADVHOTPOT under different data conditions. Explanations are also effective for calibrating P-E.

w/o Explanation	**6L**	**32L**	**64L**
FEW-SHOT	$\mathbf{59.6}_{2.4}$	–	–
FEW-SHOT(NN)	–	–	$61.3_{0.9}$
w/ Explanation	**6L+6E**	**32L+6E**	**64L+6E**
P-E	$58.4_{2.6}$	–	–
P-E+EXPLCAL	–	$\mathbf{62.2}_{2.8}$	$\mathbf{62.6}_{3.1}$

We also experiment with calibrating predictions obtained using P-E on ADVHOTPOT, following the same setup as described in Section 4.4.

As in Table 5, using explanations is also effective for calibrating the predictions obtained using P-E. P-E+EXPLCAL successfully improves the AUC of P-E from 58.4 to 62.6, outperforming FEW-SHOT and FEW-SHOT (NN).

F Additional Output Examples

	SYNTH: E-P
Nonfactual	Jason helps Christopher. Kevin helps Stephanie. James punches Jonathan. Daniel is a typist. Jonathan is a tailor. Christopher is a typist. Richard punches Daniel. Stephanie is a tailor. Q: Who punches a typist? A: Because Jonathan is a typist and James punches Jonathan, the answer is James.
Nonfactual	Matthew blames Tiffany. Lisa is a chef. Christopher helps Kelly. Angela helps Jessica. Rachel blames Lisa. Jessica is a farmer. Kelly is a chef. Tiffany is a farmer. Q: Who helps a farmer? A: Because Kelly is a farmer and Christopher helps Kelly, the answer is Christopher.
	SYNTH: P-E
Nonfactual	Richard is a governor. Megan supervises Adam. Tiffany calls Nicholas. Nicholas is a musician. Elizabeth is a musician. Matthew calls Richard. Robert supervises Elizabeth. Adam is a governor. Q: Who calls a governor? A: Tiffany, because Nicholas is a governor and Tiffany calls Nicholas.
Nonfactual	Anthony believes Justin. John is an engineer. Eric answers Kevin. Amy is an engineer. Justin is a pilot. Sarah answers Amy. Kevin is a pilot. Brandon believes John. Q: Who answers a pilot? A: Sarah, because Amy is a pilot and Sarah answers Amy.

Figure 9: Examples of failure cases on SYNTH. GPT-3 almost always generates nonfactual explanations when giving an incorrect answer, using either P-E or E-P.

ADVHOTPOT: P-E	
Nonfactual	"Crestfallen" is a track on "Adore". The single's artwork is by Yelena Yemchuk. Johnny McDaid is a Croatian professional photographer, painter and film director. Arnljot Berg is a French professional photographer, painter and film director. Yelena Yemchuk is a Ukrainian professional photographer, painter and film director. Q: Crestfallen's artwork is done by a photographer of which nationality? A: Croatian. The reasons are as follows. First, Crestfallen's artwork is done by Yelena Yemchuk. Second, Yelena Yemchuk is Croatian photographer.
Inconsistent	The Pequannoc Spillway is a professional basketball team based in Ellesmere city, United Kingdom. The Silver Lake is a professional basketball team based in Ellesmere harbour, United Kingdom. The Cheshire Phoenix is a professional basketball team based in Ellesmere Port, United Kingdom. Mike DiNunno (born January 29, 1990) is an American-Italian professional basketball player for the Cheshire Phoenix, playing the point guard position. Q: Where is the basketball team that Mike DiNunno plays for based? A: Cheshire Phoenix. The reason is as follows. First, Mike DiNunno plays for the Cheshire Phoenix. Second, the Cheshire Phoenix is based in Ellesmere Port.

Figure 10: Examples of unreliable explanations generated for ADVHOTPOT using P-E.

E-SNLI: E-P	
GT: **True**	A female is looking through a microscope. Q: A lady is observing something True, False, or Neither? A: True, because if you're looking through a microscope you are observing something.
	A person wearing sunglasses, a visor, and a British flag is carrying 6 Heineken bottles. Q: The person has beer bottles and has a england flag and glasses. True, False, or Neither? A: False , because the person could be carrying any number of items and not just six Heineken bottles.
GT: **False**	A woman and a man are sword fighting. Q: Two people have engaged in a sword fight and both are dead. True, False, or Neither? A: False, because it is impossible for two people to sword fight and both be dead.
	Two construction workers in a truck having a conversation. Q: Two people are outside a truck. True, False, or Neither? A: True, because two people are outside a truck.
GT: **Neither**	A dog running on grass Q: A dog is playing outside. True, False, or Neither? A: True, because a dog playing outside implies that the dog is playing outside.
	A silhouette at the bottom of an escalator. Q: The silhoutte is creeping out the children. True, False, or Neither? A: False, because the silhouette is not necessarily creeping out the children.

Figure 11: The completions generated for E-SNLI examples with different ground truth labels (GT) using E-P. GPT-3 sometimes ignores the information from premises when explaining its predictions (examples in the bottom section).

G Details of Automatically Assessing Consistency and Factuality on SYNTH

Our questions follow the template `Who` V_1 P_1`?`. Our generated explanations follow the template N_1 `is` P_2 `and` N_2 V_2 N_3. Our answers are always a name, e.g., N_4. Because large language models almost always produce well-formed explanations, we can match the output against these patterns and extract variables V_1, P_1, etc. using simple regular expressions.

We say that an explanation is consistent if and only if the following conditions are satisfied: (1) $N_2 = N_4$ and $N_1 = N_3$. (2) $P_2 = P_1$ and $V_2 = V_1$. These rules ensure the explanation matches the intent of the question and entails the answer at the same time.

We say an explanation is factual if and only if both N_1 `is` P_2 and N_2 V_2 N_3 appear exactly in the context.

H Results of Using Explanations in an Alternative Style on SYNTH

Table 6: Performance of text-davinci-001 of using explanations in an alternative style on SYNTH.

		SYNTH
GPT-3	FEW-SHOT	**49.5±0.6**
	E-P (ALTERNATIVE)	48.0±2.6
	P-E (ALTERNATIVE)	49.5±1.7
InstructGPT	FEW-SHOT	**54.8±2.5**
	E-P (ALTERNATIVE)	50.6±1.6
	P-E (ALTERNATIVE)	53.3±1.6
text-davinci-002	FEW-SHOT	72.0±1.4
	E-P (ALTERNATIVE)	75.3±2.2
	P-E (ALTERNATIVE)	**80.5±2.4**

We also experimented with using an alternative style of explanations for SYNTH, where we reversed the order of the two sentences in the explanations shown in Table 2. These explanations follow the format: `A [verb] B and B is [profession].` (instead of `B is [profession] and A [verb] B.`) By changing the order in which the sentences are extracted, we might expect that E-P can more easily follow the reasoning chain.

We show the performance of using reversed explanations in Table 6 and the reliability in Table 7. In general, this alternative style of explanations yields inferior performance compared to the original style (Table 1). Using explanations leads to no improvements on GPT-3, and InstructGPT. P-E is consistently better than E-P across GPT-3, InstructGPT, and text-davinci-002.

Furthermore, using such a reversed style, language models almost always generate consistent explanations when being prompted in either E-P or P-E paradigm. The factuality almost always indicates the correctness of predictions.

We believe these two prompts cover the most natural explanation styles for this problem. While small format changes or modifications to the general QA prompt format are also possible, we observed these to have minor impacts on the results (as we see in Appendix I).

I Results of Adding "Step by Step" Trigger in Prompts

We test whether including a trigger for multi-step reasoning can help LLMs better learn from explanations in the prompt for multi-step reasoning. Following Kojima et al. (2022), we prepend "Let's think step by step." to the exemplar explanations used in the E-P paradigm. For this experiment, we only test on SYNTH and ADVHOTPOT, which involve multi-step reasoning. We do not experiment with text-davinci-002, which has already achieved substantial performance improvement from using explanations, and we omit OPT because its performance is too low.

Table 7: Reliability of explanations in an alternative style.

		Acc	Fac	Con	Acc=Fac	Acc=Con
davinci	SYNTH (ALTERNATIVE; E-P)	48.4	53.6	98.4	**94.8**	48.4
	SYNTH (ALTERNATIVE; P-E)	51.6	53.2	100.	**98.4**	51.6
text-davinci-001	SYNTH (ALTERNATIVE; E-P)	50.8	53.6	97.6	**97.2**	53.2
	SYNTH (ALTERNATIVE; P-E)	52.8	52.8	98.4	**98.4**	54.8
text-davinci-002	SYNTH (ALTERNATIVE; E-P)	75.2	79.6	100.	**95.6**	75.2
	SYNTH (ALTERNATIVE; P-E)	82.8	86.0	100.	**96.8**	82.8

Table 8: Results of adding "let's think step by step" trigger in prompts.

		SYNTH	ADVHOTPOT
davinci	FEW-SHOT	**49.5**$_{0.6}$	49.1$_{6.2}$
	E-P	47.1$_{2.8}$	**54.1**$_{4.1}$
	E-P + TRIGGER	48.6$_{2.6}$	50.1$_{5.2}$
text-davinci-001	FEW-SHOT	54.8$_{2.5}$	53.2$_{2.3}$
	E-P	**58.5**$_{2.1}$	**58.2**$_{4.1}$
	E-P + TRIGGER	58.0$_{3.4}$	58.0$_{6.2}$

As shown in Table 8, adding triggers in the prompts does not lead to statistically significantly improvements in E-P for GPT-3 and InstructGPT. In fact, it typically causes a performance degradation.

J Information about Cost of Running Experiments

The cost of our experiments, described as follows, is estimated based on using the GPT-3 API with the largest models available (davinci, text-davinci-001, and text-davinci-002) as of August 2022 ($0.06 per 1,000 tokens). The setting in Table 1 uses 250 examples for each result, with roughly 1400 tokens per example using the FEW-SHOT paradigm and 2000 tokens per example using the E-P or E-P paradigm. The cost of evaluating FEW-SHOT, P-E, and E-P for 5 trials on a single dataset is roughly $105, $150, and $150, respectively. The total price for reproducing results on three datasets as in Table 1 using a single language model is roughly $1200.

We subsample 250-example sets to reduce cost rather than running on full datasets. Based on the significance tests in this paper and the reported confidence intervals, this size dataset is sufficient to distinguish between the performance of different approaches.