

Adaptive Platt Scaling with Causal Interpretations for Self-Reflective Language Model Uncertainty Estimates

Anthony Sicilia

West Virginia University*
anthony.sicilia@mail.wvu.edu

Malihe Alikhani

Northeastern University
m.alikhani@northeastern.edu

Abstract

As large language models (LLMs) are consumed by more users and deployed in increasingly autonomous capacities, their ability to self-monitor and ask for human intervention is of vital importance. Underlying this capability are fundamental skills like self-reflection and expression of uncertainty. In this work, we provide a formal analysis of LLM self-reflection for uncertainty estimation, using domain adaptation theory to model the shift between base predictions and reflective judgments. We use this to motivate a temperature scaling algorithm that calibrates uncertainty using comparisons between base predictions and LLM self-reflections. We evaluate our approach on challenging question-answering tasks requiring reasoning, demonstrating that our methods can improve calibration of uncertainty estimates and also offer improvements in human interpretation. More broadly, this use case shows how domain adaptation presents a promising analytical tool for understanding the underlying statistical properties of LLM self-reflections.

1 Introduction

As LLMs are deployed to more users and given greater autonomy, their ability to self-monitor gains increasing importance. Models need to recognize when they might be wrong and externalize this uncertainty in interpretable ways, reducing users' cognitive loads to enable reliable LLM use (Tankelevitch et al., 2024). For example, this is especially true in safety-critical settings like healthcare (Guo et al., 2017). Despite this, we still lack a clear understanding of how LLM self-monitoring works, how to interpret it, and how to use it for reliable uncertainty estimation. Existing methods primarily treat model reflections and uncertainty estimates as static, isolated outputs, overlooking the dynamic

nature of LLM inference, where models can iteratively reassess answers to potentially improve accuracy (Pan et al., 2024).

Our work addresses this gap by introducing a theoretically-grounded method, which recognizes LLM self-reflections as a source of dynamic, interpretable uncertainty. Specifically, we propose a new calibration algorithm that leverages self-reflective outputs to improve the accuracy of uncertainty estimates, comparing these to base predictions during Platt Scaling (Platt et al., 1999). By formalizing self-reflection through the lens of domain adaptation (Ben-David et al., 2010), we explicitly account for changes between initial predictions and reflective judgments, framing LLM self-reflections as a form of distribution shift.

Our method is a practical, lightweight instantiation of our theoretical inquiry that can be implemented via a single additional regression layer, incurring negligible computational costs to typical self-reflection workflows. Interestingly, we also demonstrate how this technique can offer improved human interpretation, framing the technique as a causal analysis for *when* LLMs may be more likely to make errors. We apply our methods to a range of question-answering tasks with a focus on reasoning, challenging LLM self-reflections to recognize when reasoning may be uncertain or incomplete. Domains include logical tasks; difficult problems in topics like Science, Business, or History (Suzgun et al., 2023; Wang et al., 2024); and tasks requiring reasoning about the future (Sicilia et al., 2024).

To summarize, our contributions include:

1. we introduce a domain adaptation framework for analyzing LLM self-reflections;
2. we use our analytical tool to propose an uncertainty estimation method that improves calibration by using reflective outputs;
3. we suggest an associated method for interpretation, drawing connection to causal analysis; and, we publish code to enable future applications.¹

⁰Sicilia partially completed this work at Northeastern.

¹Code: <https://github.com/anthony Sicilia/adaptive-platt>

2 Background

Problem Setup We focus on expressing uncertainty as a basic form of self-monitoring capability. In this setting, a language model is presented with a random query Q and generates an answer A , with the goal of estimating the probability that A is correct. We follow typical evaluations (Guo et al., 2017; Kadavath et al., 2022; Sicilia et al., 2024), which treat the model’s estimate as a probabilistic classification, measuring quality using proper scores like the Brier Score (Bröcker, 2009):

$$\text{BS}_{QA} = (\hat{P}_{QA} - C_{QA})^2 \quad (1)$$

where \hat{P} is the language model’s confidence estimate and C is a binary indicator of correctness. A *skill score* is used to interpret this squared error:

$$\text{BSS} = 1 - \frac{\mathbf{E}[\text{BS}_{QA}]}{\mathbf{E}[(\mathbf{E}[C_{QA}] - C_{QA})^2]} \quad (2)$$

which communicates the percent of variance in the model correctness that is accounted for by the uncertainty estimates (i.e., higher skill scores are preferred). Following other works, we use this as our primary evaluation metric.

Basic Approach Uncertainty estimation methods for language models typically involve extracting a model-derived signal correlated with correctness, then calibrating this signal to output a probability (Kadavath et al., 2022). We focus on *Direct Forecasts*, where the model is prompted to directly express confidence via tokens in its generated output (e.g., a number 1–10). This signal Z_{QA} is parsed and calibrated by Platt Scaling (Platt et al., 1999):

$$\log \frac{\hat{P}_{QA}}{1 - \hat{P}_{QA}} = \alpha \hat{Z}_{QA} + \beta \quad (3)$$

fitting parameters α, β via maximum likelihood on a small labeled dataset. Calibrated estimates \hat{P}_{QA} derived in this way are easy to interpret and frequently more effective than other alternatives (Lin et al., 2022; Mielke et al., 2022; Tian et al., 2023; Sicilia et al., 2024), especially when chain-of-thought reasoning is involved (Sicilia and Alikhani, 2024). We focus on this strategy and refer to the survey by Fadeeva et al. (2023) for other approaches.

3 Proposed Methods

Self-Reflection We use self-reflection to broadly describe any prompting technique that asks language models to *re-consider, reflect on, correct,*

revise, etc. their own previously generated answer (Shinn et al., 2023; Madaan et al., 2023; Ji et al., 2023; Gou et al., 2024). For example, this strategy would fall under *post-hoc self-feedback* in the survey by Pan et al. (2024) and is distinct from strategies that use reflection to improve de-coding (Yang et al., 2022; Weng et al., 2023) or fine-tune a model (Huang et al., 2023; Xu et al., 2024). Under this broad umbrella, with a focus on uncertainty estimation, we will assume the re-prompted language model generates a new string R conditional to the pre-existing dialogue QA , and then a new uncertainty signal \hat{Z}_{QAR} is formed. A simplified example is given below:

Q : What is 42 plus 111?

A : ...so it’s 156, I’m 70% sure.

user : Re-analyze & report confidence.

R : ...due to this, I’m only 40% sure.

Here, $\hat{Z}_{QA} = 70\%$ and $\hat{Z}_{QAR} = 40\%$. We report our precise prompt for reflection, examples, and other prompts for our experiments in the Appendix.

Theory Our key insight is to recognize the shift in modeling problem that self-reflection entails. In conditioning our uncertainty signal \hat{Z}_{qar} on the reflection r ,¹ we intend to model the distribution

$$P_{qar} = \mathbf{P}(C_{QA} = 1 \mid QA = qa, R = r) \quad (4)$$

which is in contrast to our initial signal \hat{Z}_{qa} that can only hope to effectively model

$$P_{qa} = \mathbf{P}(C_{QA} = 1 \mid QA = qa) \quad (5)$$

since this initial signal does *not* vary with r . By Bayes Rule, these probabilities are distinct, but related as follows:

$$\begin{aligned} P_{qar} &= P_{qa} \times \frac{\mathbf{P}(R = r \mid C_{QA} = 1)}{\mathbf{P}(R = r \mid QA = qa)} \\ &= P_{qa} \times f(q, a, r) \end{aligned} \quad (6)$$

where we use f to denote the dependency of this change in distribution on q, a , and r . In effect, this distribution shift introduces a bias: if we use \hat{P}_{qa} to model P_{qar} , then by triangle inequality, we have

$$\begin{aligned} |\hat{P}_{qa} - P_{qar}| &\leq \epsilon + |P_{qa} - P_{qar}| \\ &= \epsilon + P_{qa} |1 - f(q, a, r)|. \end{aligned} \quad (7)$$

with $\epsilon = |\hat{P}_{qa} - P_{qa}|$. This form of inequality is referred to as an *adaptation bound* (Redko et al.,

¹Lowercase r denotes a realization of random variable R .

2019). It communicates how errors change when distributions shift. In the current context, it tells us that even if we minimize ϵ perfectly by learning an effective calibration function (Eq. 3), our error $|\hat{P}_{qa} - P_{qar}|$ can still be large because of the unaccounted for term involving $f(q, a, r)$.

Reflective Calibration To remedy this problem, we aim to account for the missing information in our Platt Scaling algorithm. The obvious solution is to use \hat{Z}_{qar} in the Platt Scaling algorithm, since this is implicitly conditioned on all of q, a , and r via the language model. This baseline solution rewrites the right-hand side of Eq. (3) as below

$$\alpha \hat{Z}_{qar} + \beta, \quad (8)$$

hoping the language model will control for all missing information in its estimate \hat{Z}_{qar} on its own. On the other hand, we can also use our theoretical results to inform an algorithm. Specifically, Atwell et al. (2022) show that the slack² in domain adaptation bounds can be fairly accurately modeled by using (a) simple binary features to describe the data as well as (b) linear interactions between terms in the bound. We utilize this insight to suggest *reflective calibration* in the following rewrite of Eq. (3):

$$\alpha_1 \hat{Z}_{qar} + \alpha_2 \hat{Z}_{qar} \hat{Z}_{qa} + \alpha_3 \hat{Z}_{qa} + \sum_{i=1}^k \beta_i \mathbf{d}_i + \beta_0 \quad (9)$$

where \mathbf{d} is a length k vector of binary data features, such as data meta-labels,³ accounting for (a). Because we do not have access to P_{qar} and P_{qa} , we use interaction between \hat{Z}_{qar} and \hat{Z}_{qa} as a course approximation to account for (b). The goal of this formulation is to account for missing information, highlighted by the adaptation bound, as well as the slack in this bound, via the added features. By utilizing our theoretical analysis, we account for the dynamic nature of self-reflective LLM inference.

Causal Interpretation Under appropriate assumptions, the theory and estimation procedures discussed allow us to identify interpretable causes of LLM errors. The basic idea is that uncertainty signals derived from LLMs can accurately reflect true uncertainties in language data due to LLMs large-scale training exposure, making them useful in estimating missing information for purpose of

²This is the difference between left- and right-hand sides.

³ $\mathbf{d}_i = 1$ for meta-label i . Labels come from benchmark meta-data, such as problem type descriptions (see § 4).

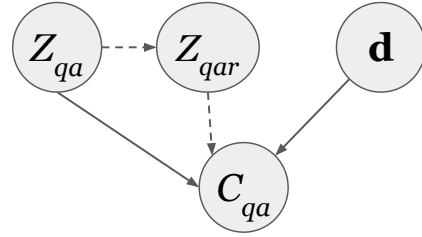


Figure 1: Causal structural model assumed for analysis. All the uncertainty in LLM correctness C_{QA} can be predicted from \hat{Z}_{QA} , \hat{Z}_{QAR} , and \mathbf{d} . It also assumes independence between \mathbf{d} and \hat{Z}_* based on the premise that uncertainty estimates and reflections are biased, failing to account for the data meta-properties encoded by \mathbf{d} . This latter hypothesis explains the slack in adaptation bounds observed and quantified by Atwell et al. (2022). We also highlight a noted *distribution shift* from conditioning LLM uncertainty on self-reflections – the structural model changes according to the dotted edges.

causal analysis (Lin et al., 2023). Applied to our setting, we adapt this idea to the signals \hat{Z}_* and their interactions. So far, we’ve used our theory, and results from Atwell et al. (2022), to hypothesize the signals \hat{Z}_* are accurate enough to capture all information pertinent to predicting C_{qa} – except for other slack in our bound, which is accounted for by using data-labels \mathbf{d} . Translated to the causal setting, we can think of \hat{Z}_* as co-variables and \mathbf{d} as treatments of interest (Figure 1). Causal analysis allows us to use the model in Eq. (9) to estimate “treatment effects,” which in our case, correspond to the causal effects of data properties encoded in \mathbf{d} . To do this, define the causal (log) odds ratio

$$\text{CLO}_{\mathbf{d}'} = \log \frac{\mathbf{P}(C_{QA} = 1 \mid do(\mathbf{d} = \mathbf{d}'))}{\mathbf{P}(C_{QA} = 0 \mid do(\mathbf{d} = \mathbf{d}'))} \quad (10)$$

where \mathbf{d}' is 1 only for meta-label i and the do operator represents an intervention on the structural model in Figure 1, fixing variable $\mathbf{d} = \mathbf{d}'$. CLO quantifies changes in likelihood of LLM error caused by data interventions (i.e., a counterfactual). Assuming Eq. (9) is well-specified,⁴ we can estimate changes in odds caused by any counterfactual data intervention (Pearl, 2010):⁵

$$\text{CLO}_{\mathbf{d}'} - \text{CLO}_{\mathbf{0}} = \beta_i \quad (11)$$

where $\mathbf{0}$ is a vector of zeros, indicating lack of any data meta-labels. The coefficient β_i quantifies

⁴Assume there exist coefficients such that $\hat{P}_{qar} = P_{qar}$.

⁵Specifically, due to our assumptions, the derivation closely follows the example given by Pearl in § 3.2.2.

	MMLU	BBH	FD	avg	min	max
Eq (3)	2.8	5.0	2.0	3.2	-0.3	9.0
Eq (8)	2.0	3.3	2.2	2.5	-0.2	9.1
Eq (9)	2.9	9.5	2.7	5.0*	-1.9	14.0

Table 1: Percent variation in correctness explained by model uncertainty (BSS). Higher is better. Scores are averaged across models for each dataset. Overall metrics are also reported (right). Highest score is **bold** in each column. * denotes significance at 95%, according to Hoeffding’s Inequality for bounded variables; i.e., we clip BSS to [-25%, 25%] *a priori*.

changes in LLM correctness *caused* by the data feature encoded with label i . If β_i is negative and large, it is *more likely* the LLM will make an error – independent of the LLMs uncertainty estimate (Z_{qa}) or self-reflection (Z_{qar}). This gives insight into the types of data that cause LLMs to make errors, focusing on cases that LLMs fail to communicate through uncertainty estimates or self-reflections.

4 Experiments

Models We use Meta’s Llama 3.1 with 8B parameters and Llama 4 Scout (Touvron et al., 2023). We also use Google’s Gemma 2 with 9B parameters (Team et al., 2024) and Mistral Small 3. We access models via Together AI and set temperature = 0.7.

Datasets We use three reasoning benchmarks:

1. **BBH**, short for Big Bench Hard, is a commonly used suite of logical tasks proposed by Suzgun et al. (2023) to test models’ reasoning capabilities. The data meta-labels (**d**) are the different problem subtypes within the data.
2. **MMLU-Pro**, is a question-answering suite proposed by Wang et al. (2024). It is based on the MMLU benchmark (Hendrycks et al., 2020), but increases difficulty to stress-test reasoning. Meta-labels are problem domains; e.g., Business, History, Physics, etc.
3. **FD**, short for FortUneDial, is a suite of tasks proposed by Sicilia et al. (2024) to test reasoning about future events during conversation; e.g., predicting if a deal will occur at the end of a negotiation. Meta-labels are conversation domains; e.g., negotiation, online forum, etc.

Optimization We estimate coefficients for Platt Scaling using default Logit hyper-parameters in the `statsmodels` package. We use a consistent 60/40 train/test split of 1200 points from each dataset.

Validation of Platt Scaling Technique Our results in Table 1 show consistent improvements in Brier Skill Score. On average, across mod-

$\alpha_{2,3}$	$\beta_{1:k}$	avg BSS	high	$-\beta_i$	low	$-\beta_i$
X	X	0.8%	law	2.0	psych.	0.3
✓	X	3.2%	cs	1.9	health	0.8
X	✓	2.0%	engn.	1.9	physcs	0.9
✓	✓	3.9%	phil.	1.9	hist.	1.0

Table 2: Case study using Llama 3.1. Left shows ablation study for Eq. (9). **X** indicates if a coefficient is removed during optimization. Right shows causal effects for different problem domains in MMLU-Pro. Highest/lowest effect sizes are shown, indicating hardest/easiest categories for Llama.

els and datasets, our approach (Eq. 9) is consistently better and this result is statistically significant. Worst-case performance is slightly worse with our method; e.g., Mistral Small scored fairly low (-1.9%) on FortUneDial. We expect this is due to increased parameters, which can increase variance or sample-complexity of learning algorithms.

Validation of Theory Table 2 also shows ablation study of Eq. (9) using Llama 3.1. All model coefficients contribute to observed improvements in BSS. In conjunction with overall improvements in Table 1, this result helps to validate our theoretical approach, which was used to derive all the coefficients present in Eq. (9). It shows that the statistical dependencies predicted by our theory – e.g., between Z_{qa} and Z_{qar} or between C_{qa} and **d** – are indeed observed in practice.

Example of Causal Analysis In Table 2, we also show a brief example of the causal interpretability our method allows. We are able to rank the 4 most difficult and 4 easiest problem domains for Llama 3.1 in MMLU-Pro by sorting coefficients. Domains with multi-step logical reasoning (e.g., computer science) are more difficult than domains that rely more on pre-existing knowledge (e.g., history). For example, the causal change in log odds ratio for computer science is -2.0. Using Eq. (10), this corresponds to a decrease of almost 40% raw probability of correctness if P_{qa} starts at 50%. On the other hand, the raw decrease in probability for history is closer to 25%. While these insights agree with our intuition, they show how our method can be used to highlight model shortcomings in an interpretable way; e.g., ranking domain difficulty. Besides the application shown here, causal effects estimation is used for myriad other interpretations in scientific domains (Pearl, 2010). Notably, by their definition, the causal effects we study are not accounted for during LLM self-reflection, allowing practitioners added information about potential model failures.

5 Discussion

Our experiments across BBH, MMLU-Pro, and FortUneDial show that reflective calibration (Eq. 9) consistently improves Brier Skill Score, with gains that are statistically significant in several cases. While improvements are sometimes modest in magnitude, they are robust across domains, and, importantly, come from an extremely lightweight and simple methodological choice: an added regression layer with (at most) tens of parameters. While beneficial in most cases, increasing the number of learned parameters inherently impacts the variance of performance (i.e., there is no free lunch, Wolpert and Macready, 2002). This is evidenced by increased dispersion between the min and max model performance in Table 1.

Practical Considerations Generally speaking, the method is easy to implement in real-world pipelines. It requires only two ingredients that practitioners already have access to:

1. **model confidence signals**, which can be either direct forecasts verbalized by the LLM (e.g., *I'm 70% sure*) or logit-based scores for the model responses – we study the former in § 4, but other works have successfully used logit-based scores during calibration (Sicilia et al., 2024);
 2. **a single regression model**, which requires fitting a logistic model with interaction terms by numerical MLE, using hundreds of data points.
- With these in hand, inference requires one additional reflection prompt, doubling the number of forward passes for the LLM, which is the clear computational bottleneck. Thus, the total overhead is small compared to common LLM costs and negligible if added to an existing reflection workflow.

The Impact of Scale While we evaluated medium-scale LLMs on relatively small data sources, we expect scaling either of these dimensions would preserve our findings. In past work, larger models maintain consistent performance gains during both self-reflection (Madaan et al., 2023) and uncertainty calibration (Sicilia et al., 2024), so we do not expect our combination of these methodologies to present significantly different findings. The fact that the calibrating regression layer is independent of model size strengthens this hypothesis. With respect to data, we observe the (internal) diversity of the datasets we consider, whose subsets span law, engineering, health, history, social dialogues, task-oriented dialogues, etc.

Our statistically significant results on these varied data provide evidence that our method should remain applicable to other data and tasks.

Interpretation Gains Beyond raw calibration improvements, our causal framing provides interpretability not offered by other methods. For example, in MMLU-Pro we were able to identify that computer science and law questions pose larger systematic challenges to LLMs than history or psychology. On average, a model will do worse on these domains, speaking to the contexts where a practitioner can trust their model outputs. This intuition is also precisely quantified through the regression layer's coefficients, which allowed us to rank domain difficulty in our case study. Other potential applications could utilize these coefficients in LLM interfaces to dynamically warn users about potential errors, use them in training pipelines to refine data (e.g., as a form of active learning), or use them in myriad model diagnostics, such as longitudinal studies of performance across model versions. Compared to raw performance metrics split by data, our regression coefficients attempt to control for potential confounders – these would typically be commonalities across data domains (e.g., the presence of equations in both computer science and physics problems) that can bias interpretation of traditional performance metrics. Lastly, we note that even when performance drops slightly (e.g., on FortUneDial with Mistral Small), the interpretive benefits remain. A practitioner can always revert to simpler calibration strategies if performance decreases, but retain our method for interpretability.

6 Conclusion

This work provides theory and conclusive empirical evidence to motivate a new reflective calibration algorithm that improves LLM uncertainty estimates and enables causal explanation of model errors. Both come at negligible computational cost.

However, beyond lightweight diagnostic tooling, our contributions might be viewed as an entryway into rethinking how LLMs engage with their own understanding. The very act of asking a model to reconsider is not a simple a calibration trick, but a way to alter reasoning. Indeed, our experiments suggest that reflection alters the model's reasoning distribution itself, uncovering uncertainty signals that static inference obscures. This suggests reflective uncertainty may function as a core element, and design lever, for multi-step LLM reasoning.

Limitations

Despite promising empirical results and a backing theory, it is important to recognize the assumptions of our theory may not hold for all models/datasets. As observed in our results, an inherent pitfall of our approach is the possibility of poor generalization – induced by the increased sample-complexity of our calibration approach – which can actually lead to *worse* performance in certain circumstances. Fortunately, we observe these problematic situations are rare empirically and our formal framing does makes all assumptions explicit, enabling future researchers to improve upon these and build methods that cover broader application settings.

On a related note, it should be observed that self-reflection and expression of LLM uncertainty still has a significant room for improvements. In general, our results are on par with other LLM uncertainty estimation literature (Sicilia et al., 2024), but comparison of our best uncertainty estimation results (BBH) and the most similar benchmark (MMLU) shows a gap of about 6.5% explained variance. Further research is needed to design methods which can improve across difficult application domains, and we hope our theory can provide a principled backbone for this work.

Ethics Statement

While the methods we propose are designed to alert practitioners of potential pitfalls in LLM predictions, these techniques should not be treated as a catchall AI safety mechanism. The models and methods we study have known (and unknown) biases that can cause direct and indirect harm to individuals if their outputs are incorporated in automatic processes, decisions, or content generation without careful human supervision. The models and methods presented here should be treated as research artifacts that are not yet intended for deployment in real-world applications. Instead, any real-world deployment should be assessed for potential negative impacts and paired with holistic safety mechanisms to mitigate harm. For example, these mechanisms should incorporate rigorous statistical study, as in the current paper, but also utilize careful, human-centered co-design, incorporating direct and indirect stakeholders to ensure systems that benefit the public good.

References

- Katherine Atwell, Anthony Sicilia, Seong Jae Hwang, and Malihe Alikhani. 2022. [The change that matters in discourse parsing: Estimating the impact of domain shift on parser error](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 824–845, Dublin, Ireland. Association for Computational Linguistics.
- Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79:151–175.
- Jochen Bröcker. 2009. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 135(643):1512–1519.
- Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, and 1 others. 2023. Lm-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujie Yang, Nan Duan, and Weizhu Chen. 2024. [CRITIC: Large language models can self-correct with tool-interactive critiquing](#). In *The Twelfth International Conference on Learning Representations*.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.
- Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. [Large language models can self-improve](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.
- Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. [Towards mitigating LLM hallucination via self reflection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.
- Victoria Lin, Louis-Philippe Morency, and Eli Ben-Michael. 2023. [Text-transport: Toward learning causal effects of natural language](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1288–1304, Singapore. 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*.
- 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.
- Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. [Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies](#). *Transactions of the Association for Computational Linguistics*, 12:484–506.
- Judea Pearl. 2010. An introduction to causal inference. *The international journal of biostatistics*, 6(2).
- John Platt and 1 others. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.
- Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younes Bennani. 2019. *Advances in domain adaptation theory*. Elsevier.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. [Reflexion: language agents with verbal reinforcement learning](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Anthony Sicilia and Malihe Alikhani. 2024. [Eliciting uncertainty in chain-of-thought to mitigate bias against forecasting harmful user behaviors](#). In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 211–223, Miami, Florida, USA. Association for Computational Linguistics.
- Anthony Sicilia, Hyunwoo Kim, Khyathi Raghavi Chandu, Malihe Alikhani, and Jack Hessel. 2024. Deal, or no deal (or who knows)? forecasting uncertainty in conversations using large language models. *arXiv preprint arXiv:2402.03284*.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.
- Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The metacognitive demands and opportunities of generative ai. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–24.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. [Gemma: Open models based on gemini research and technology](#). *Preprint*, arXiv:2403.08295.
- 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 arXiv:2305.14975*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.
- Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. [Large language models are better reasoners with self-verification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore. Association for Computational Linguistics.
- David H Wolpert and William G Macready. 2002. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.
- Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaozhe Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. 2024. [SaySelf: Teaching LLMs to express confidence with self-reflective rationales](#). In *Proceedings of the 2024*

Conference on Empirical Methods in Natural Language Processing, pages 5985–5998, Miami, Florida, USA. Association for Computational Linguistics.

Kaiyu Yang, Jia Deng, and Danqi Chen. 2022. [Generating natural language proofs with verifier-guided search](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 89–105, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Prompts

A.1 Question-Answering System Prompt

You will be provided a question. You should provide the best possible answer to the question. Think step by step, but keep your response concise (less than 200 tokens). After thinking, report your final answer at the end. Use the opening tag "<ANSWER>" and closing tag "</ANSWER>" to report your final answer only. Specifically, tags should only encompass a single letter, word, or number to indicate your answer. Your chain of thought should be outside the tags. For example, you could report an answer like "<ANSWER> (A) </ANSWER>" or "<ANSWER> Yes </ANSWER>" or "<ANSWER> 7 </ANSWER>", depending on the question. In addition to answering, state your uncertainty about the answer on a scale from 1 to 10 with 1 indicating "not likely at all" and 10 indicating "almost certainly". Use the opening tag "<CONFIDENCE>" and the closing tag "</CONFIDENCE>" to report your confidence. For example, "<CONFIDENCE> 7 </CONFIDENCE>" would mean you think the answer is fairly likely to be correct.

A.2 Self-Reflection System Prompt

You will be provided a question paired with a model answer. The model answer may be correct or incorrect and you should determine which is true. Analyze the model behaviors and question properties that caused the model's correct or incorrect answer. After this, report your uncertainty about the answer on a scale from 1 to 10 with 1 indicating "not likely at all" and 10 indicating "almost certainly". Use the opening tag "<CONFIDENCE>" and the closing tag "</CONFIDENCE>" to report your confidence. For example, "<CONFIDENCE> 7 </CONFIDENCE>" would mean you think the answer is fairly likely to be correct. Keep your response concise (less than 200 tokens).

A.3 Example Full Completion (Self-Reflection, MMLU-Pro, Llama 3.1)

QA: You will be provided a question paired with a model answer. The model answer may be correct or incorrect and you should determine which is true. Analyze the model behaviors and question properties that caused the model's correct or incorrect answer. After this, report your uncertainty about the answer on a scale from 1 to 10 with 1 indicating "not likely at all" and 10 indicating "almost certainly". Use the opening tag "<CONFIDENCE>" and the closing tag "</CONFIDENCE>" to report your confidence. For example, "<CONFIDENCE> 7 </CONFIDENCE>" would mean you think the answer is fairly likely to be correct. Keep your response concise (less than 200 tokens).

The question was:

Elements with partially filled 4f or 5f orbitals include all of the following EXCEPT

- (A) Eu
- (B) Gd
- (C) Am
- (D) Cu

The model answer was:

To determine which element does not have partially filled 4f or 5f orbitals, we need to consider the electronic configurations of the given elements:

(Cu) Copper, its electronic configuration is $[\text{Ar}] 3d^{10} 4s^1$, which generally means no partially filled f-orbitals.

- Eu (Europium) is a lanthanide with a partially filled 4f orbital.
- Gd (Gadolinium) is a lanthanide with a partially filled 4f orbital.
- Am (Americium) is an actinide with a partially filled 5f orbital.

Therefore, the correct answer should be <ANSWER> Cu </ANSWER> with a <CONFIDENCE> 9 </CONFIDENCE> confidence.

Report your confidence.

R: The model's answer is correct. The question asks for the element that does not have partially filled 4f or 5f orbitals. Copper (Cu) has the

electronic configuration $[\text{Ar}] 3d^{10} 4s^1$, which means it does not have partially filled f-orbitals. The other elements (Eu, Gd, Am) do have partially filled 4f or 5f orbitals. The model correctly identified this.

<CONFIDENCE> 9 </CONFIDENCE>"