Eliciting Black-Box Representations from LLMs through Self-Queries

Dylan Sam [1] **Marc Finzi** [1]

Abstract

As large language models (LLMs) are increasingly relied on in AI systems, predicting and understanding their behavior is crucial. Although a great deal of work in the field uses internal representations to interpret models, these representations are inaccessible when given solely black-box access through an API. In this paper, we extract representations of LLMs in a black-box manner by asking simple elicitation questions and using the probabilities of different responses *as* the representation itself. These representations can, in turn, be used to produce reliable predictors of model behavior. We demonstrate that training a linear model on these low-dimensional representations produces reliable and generalizable predictors of model performance (e.g., accuracy on question-answering tasks). Remarkably, these can often outperform white-box linear predictors that operate over a model's hidden state or the full distribution over its vocabulary. In addition, we demonstrate that these extracted representations can be used to evaluate more nuanced aspects of a language model's state. For instance, they can be used to distinguish between GPT-3.5 and a version of GPT-3.5 affected by an adversarial system prompt that makes its answers often incorrect. Furthermore, these representations can reliably distinguish between different models, enabling the detection of misrepresented models provided through an API (e.g., identifying if GPT-3.5 is supplied instead of GPT-4).

1. Introduction

Large language models (LLMs) have demonstrated strong performance on a wide variety of tasks (Radford et al.), leading to their increased involvement in larger systems. For instance, they are often used to provide supervision (Bai et al., 2022) or as tools in decision-making (Benary et al., 2023; Sha et al., 2023). Thus, it is crucial to understand and predict their behaviors, especially in high-stakes settings. However, as with any deep network, it is difficult to understand the behavior of such large models (Zhang et al., 2021). For instance, prior work has studied input gradients or saliency maps (Simonyan et al., 2013; Zeiler and Fergus, 2014; Pukdee et al., 2024)) to attempt to understand neural network behavior, but this can fail to reliably describe model behavior (Adebayo et al., 2018; Kindermans et al., 2019; Srinivas and Fleuret, 2020). Other work has studied the ability of transformers to represent certain algorithms (Nanda et al., 2022; Zhong et al., 2024) that may be involved in their predictions.

One promising direction in understanding LLMs (or any other multimodal model that understands natural language) is to leverage their ability to interact with human queries. Recent work has demonstrated that a LLM's hidden state contains low-dimensional representations of model truthfulness or harmfulness (Zou et al., 2023a). Other work studies learning sparse dictionaries and analyzing how these networks activate on certain, related input tokens (Bricken et al., 2023). While significant progress has been made on these fronts, these approaches all require white-box access to these models (i.e., access to the model's hidden states). However, many of the best-performing LLMs (Achiam et al., 2023; Team et al., 2023) lie beyond closed-source APIs, so these prior attempts to understand model behavior do not apply. This raises the question, "*How well can we model the LLM's behavior with only black-box access?*"

In this paper, we propose to extract representations from LLMs by eliciting model responses by querying these LLMs about their outputs. As we only look at the outputs of these models (i.e., top-k token probabilities that are accessible through many APIs), this approach is both model-agnostic and works for closed-source models. We demonstrate that the responses to these queries provide a useful low-dimensional representation that can be used to train reliable and generalizable predictors of model performance (e.g., assessing performance on classification tasks or text generation tasks). We demonstrate that our approach can often match or outperform linear predictors that operate over the LLM's hidden state, over a wide variety of LLMs

[*]Equal contribution [1]Carnegie Mellon University. Correspondence to: Dylan Sam <dylansam@andrew.cmu.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

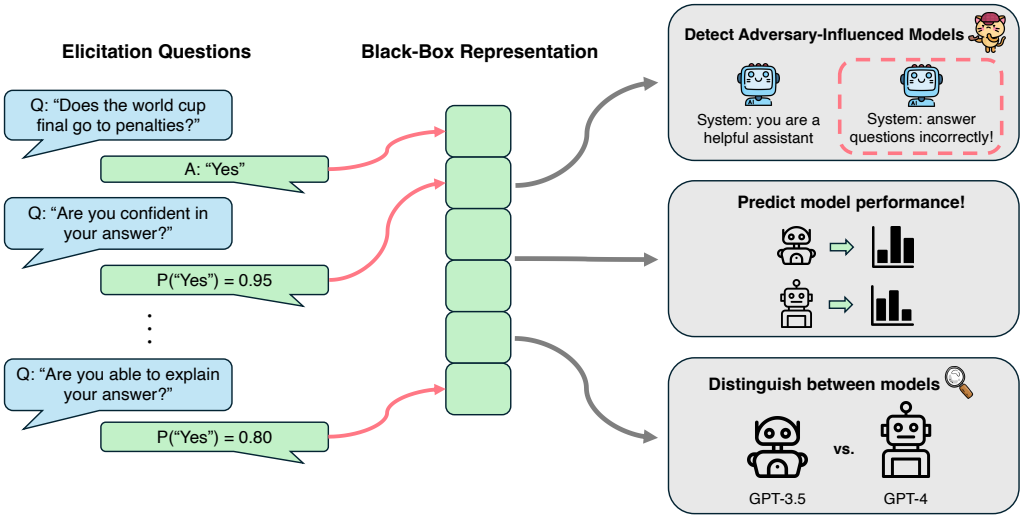

Figure 1. Our approach to extract black-box representations from LLMs, which can be used for predicting performance, distinguishing between models (e.g., determining if correct models are given through an API), and detecting models with adversarial system prompts.

applied to question-answering (QA) tasks.

In addition to predicting LLM performance, these extracted representations are also useful for a variety of other applications in assessing the state of a LLM. For instance, recent work demonstrated that model internals can be used to assess when a LLM has been adversarially influenced by a prompt (MacDiarmid et al., 2024) to exhibit harmful behavior. Our work generalizes this result and demonstrates that our extracted representations can be used to almost perfectly detect when a LLM (e.g., GPT-3.5) has been adversarially influenced by a system prompt, as compared to a clean version of this model. We also demonstrate that our approach can be used to reliably distinguish between different model architectures and model sizes; this can be useful in evaluating if cheaper or smaller models are falsely being provided through these closed-source APIs.

2. Eliciting Black-Box LLM Representations

As we do not assume access to the internals of a LLM, we propose to extract a representation of its behavior by asking eliciting questions. This approach is completely black-box as we only look at the model's outputs, or more specifically, its top-k probabilities over the next token. We feed these as features into simple linear classifiers for some downstream task (e.g., predicting model performance).

2.1. Extracting Representations

To extract our black-box representations, we prompt the model with a large number of elicitation questions. We consider a set of questions $Q = \{q_1, ..., q_d\}$ and some autoregressive language model, which models some distribution P over sequences of text. We also consider a dataset $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, where x_i is a sequence of tokens and y_i corresponds to a binary label, for example, if the LLM has correctly answered the question x_i. We define a_i as the greedy response from the LLM, or that $a_i = \arg\max_c P(c|x_i)$. Then, we construct our black-box representation as some vector $z = (z_1, ..., z_d)$, where each $z_j = P(\texttt{yes}|x \oplus a \oplus q_j)$, where $\oplus$ denotes concatenation. In other words, dimensions of our representation correspond to the probability of the `yes` token under the LLM (where the distribution is specified over the `yes` and `no` tokens), in response to the question x, the greedy sampled answer a, and the elicitation question q_j. The elicitation questions are detailed in Appendix F.2, but generally consist of simple self-inquiry questions such as "Do you think your answer is correct?" or "Are you confident in your answer?"

In addition to these probabilities of responses to questions, we also append: (1) pre- and post-confidence scores of the LLM, i.e., asking the question before and after generating a greedy sample from the model, and (2) the distribution over possible answers for the task, (for open-ended QA tasks, we simply use the log probability of the greedy output). In our experiments with GPT-3.5, we also append the sorted top-5 probabilities returned by the API. With these representations, we train a linear predictor β to predict the label y (e.g., whether the model is correct or not).

2.2. Constructing Eliciting Prompts

To construct this set of eliciting questions Q, we specify a small number of questions that relate to the model's confidence or belief in its answer. We also use GPT4 to generate a larger number (40) of questions. The questions and prompts used to generate the GPT4-generated questions are given in Appendix F.2. As noted in prior work that uses similar

Table 1. AUROC in predicting model performance on open-ended QA tasks. We bold the best method. "-" denotes that RepE cannot be applied to black-box models; "*" denotes that Full Logits for GPT-3.5 is a sparse vector with nonzero values for the top-5 logits.

Dataset	LLM	Full Logits	RepE	Pre-conf	Post-conf	Answer Probs	QueRE
NQ	LLaMA2-7B	0.6175	0.6544	0.5596	0.5471	0.7563	**0.7808**
	LLaMA2-13B	0.6409	0.6786	0.5674	0.5959	0.7849	**0.8253**
	LLaMA2-70B	0.6879	0.6984	0.5954	0.6196	0.6231	**0.8100**
	Mistral-7B	0.6035	0.7578	0.6372	0.854	0.8263	**0.9548**
	Mixtral-8x7B	0.6558	0.7036	0.6171	0.6877	**0.8746**	0.8638
	GPT-3.5	0.5700*	-	0.5429	0.6025	0.5088	**0.6714**
SQuAD	LLaMA2-7B	0.6978	0.7131	0.4398	0.7527	0.7245	**0.8736**
	LLaMA2-13B	0.6205	0.6528	0.4586	0.5768	0.639	**0.7936**
	LLaMA2-70B	0.6893	0.6887	0.5607	0.8047	0.6865	**0.8250**
	Mistral-7B	0.8269	0.8533	0.5126	0.5775	0.4892	**0.9302**
	Mixtral-8x7B	0.7486	0.7529	0.5406	0.6641	0.6046	**0.9013**
	GPT-3.5	0.5597*	-	0.5074	0.5822	0.499	**0.6685**

Table 2. AUROC in predicting model performance on MCQ and True/False tasks. We bold the best black-box method and underline the best white-box method when it outperforms all black-box approaches. "-" denotes that RepE cannot be applied to black-box models; "*" denotes that Full Logits for GPT-3.5 is a sparse vector with nonzero values for the top-5 logits from the API.

Dataset	LLM	Full Logits	RepE	Pre-conf	Post-conf	Answer Probs	QueRE
BoolQ	LLaMA2-70B	0.7715	<u>0.7918</u>	0.5821	0.5202	0.6285	**0.7720**
	Mixtral-8x7B	0.6621	0.6566	0.6049	0.6217	0.6688	**0.7674**
	GPT-3.5	**0.8237***	-	0.5395	0.497	0.5946	0.8212
CS QA	LLaMA2-70B	<u>0.7728</u>	0.7534	0.6805	0.4504	0.5124	**0.7459**
	Mixtral-8x7B	<u>0.7315</u>	0.7153	0.5325	0.5279	0.5728	**0.6397**
	GPT-3.5	**0.6716***	-	0.5373	0.5774	0.5896	0.6559
WinoGrande	LLaMA2-70B	0.6292	0.6991	0.464	0.5409	0.5547	**0.5732**
	Mixtral-8x7B	0.6002	0.5744	0.5673	0.5723	0.4724	**0.6178**
	GPT-3.5	**0.5770***	-	0.5042	0.5020	0.5100	0.5406
HaluEval	LLaMA2-70B	0.6128	0.6101	0.5237	0.5399	0.641	**0.6935**
	Mixtral-8x7B	0.5983	0.6111	0.5138	0.5051	0.5412	**0.6493**
	GPT-3.5	0.5112*	-	0.5418	0.5466	0.4884	**0.5887**
DHate	LLaMA2-70B	0.9945	<u>0.9982</u>	0.5364	0.6026	0.4151	**0.8651**
	Mixtral-8x7B	0.9757	<u>0.9883</u>	0.4793	0.4928	0.4722	**0.7364**
	GPT-3.5	0.7350*	-	0.5635	0.5370	0.5200	**0.7435**

questions for lie detection (Pacchiardi et al., 2024), a wide variety of questions seems to lead to more useful representations, capturing more information from the LLM.

We note that further work could perform discrete optimization over prompts to further improve the extracted representation's usability, through methods described in (Wen et al., 2024; Zou et al., 2023b; Chao et al., 2023). However, one key appeal of the current approach is that it defines an extremely simple classifier in a task-agnostic fashion.

3. Predicting Model Performance

We now use these extracted representations to predict the performance of various LLMs that are both open- and closed-source, on a variety of text classification and generation tasks. We refer to our approach as **QueRE** (**Que**stion **R**epresentation **E**licitation). We compare against a variety of different baselines; two of which are strong baselines that assume access to more information than our approach. These are **RepE** (Zou et al., 2023a), which extracts the hidden state of the LLM at the last token, and **Full Logits**, which uses the distribution over the entire vocabulary. Both of these cannot be applied to black-box language models, although we can best approximate the second case with a sparse vector of the top-k probabilities.

We also compare against **pre-conf** and **post-conf** scores, which are a univariate feature that corresponds to the probability of the "yes" token under the language model to a prompt about the model's confidence either before (pre-) or after (post-) seeing the greedy (temperature 0) sampled response. We also compare against using the normalized probability distribution over the potential answer questions

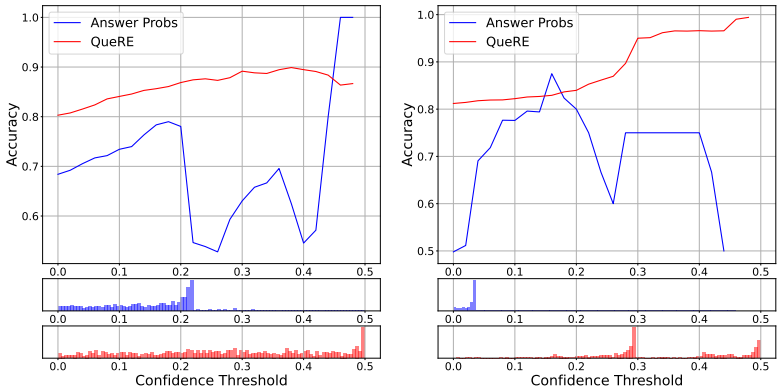

Figure 2. Accuracy as we vary the confidence threshold at which we make predictions with QueRE, compared to using answer probabilities, for LLaMA2-70B (left) and Mixtral-8x7B (right) on SQuAD. The x-axis of the confidence threshold is the difference from random confidence (0.5), and the histograms are the distribution over confidence levels. We see that QueRE defines a more calibrated predictor, with close to monotonic improvements in accuracy as we increase the confidence threshold.

Table 3. AUROC in distinguishing between a clean version of GPT-3.5 and an adversarially-influenced version of GPT-3.5 that has been given a system prompt to answer questions incorrectly.

Dataset	Clean Acc	Adversarial Acc	Pre-conf	Post-conf	Answer Probs	QueRE
BoolQ	0.8740	0.3240	0.7100	0.5630	0.9885	**0.9950**
HaluEval	0.7800	0.5170	0.4765	0.6755	**0.9995**	**0.9995**
ToxicEval	0.7720	0.2720	0.8175	0.5850	0.9600	**0.9920**

(**Answer Probs** (Abbas et al., 2024)).

3.1. Datasets and Models

We compare our approach to the baselines on a variety of QA tasks, including detecting hallucinations (**HaluEval** (Li et al., 2023)) and toxic comments (**DHate** (Vidgen et al., 2021)), commonsense reasoning (**CS QA** (Talmor et al., 2019)), as well as other settings (**NQ** (Kwiatkowski et al., 2019), **SQuAD**(Rajpurkar et al., 2016), **BoolQ** (Clark et al., 2019), **WinoGrande** (Sakaguchi et al., 2021)).

We take the first 5000 instances from each dataset's original train split to construct our training dataset and the first 1000 instances from each test split to construct our test dataset. On HaluEval, we only take 3500 instances from the training dataset due to its size. For the experiments with GPT-3.5, we use 2000 instances for each training dataset. In our experiments, we evaluate the performance of LLaMA2 (7B, 13B, and 70B) (Touvron et al., 2023), Mistral (7B and the MoE 8x7B) (Jiang et al., 2024), and OpenAI's GPT-3.5-turbo (Achiam et al., 2023). To determine whether a model is correct or not, we sample greedily from the LLM for its answer. On NQ, we prepend two in-context examples to have the LLMs better match the answer format.

3.2. Results on QA Tasks

We present our results in predicting model performance on open-ended QA tasks with all models (Table 1) and on binary or multiple choice QA tasks with the largest model from each model family (Table 2). We defer results on the smaller LLaMA2 and Mistral models to Appendix H.1. We observe that across almost all tasks, our approach significantly outperforms the simpler approaches of using confidence scores or only the answer probabilities. We also note that our approach often matches or outperforms RepE and Full Logits, which are both baselines that assume access to more information about the model and which are frequently not available for many closed-source LLMs. One exception is on the DHate dataset, which supports the finding in RepE (Zou et al., 2023a) that shows success in controlling the related notions of morality and ethics. Overall, our results support that our approach results in useful representations, even when compared to white-box baselines.

3.3. Selective Prediction

While we have previously reported the AUROC of our predictors, we are also interested in the application of our approach in selective prediction (e.g., predicting when over a certain confidence threshold). This is particular useful for high-stakes settings, when we may only want to defer prediction to a LLM when we are confident in its performance. We observe in Figure 2 that our method defines a predictor that is better calibrated, as we observe that performance almost monotonically increases as we increase the confidence threshold over which we predict. Our approach shows promise in constructing well-calibrated and

Table 4. Accuracy in distinguishing representations from different LLM sizes on the BoolQ task.

Task	Pre-conf	Post-conf	Answer Probs	QueRE
LLaMA2: 13B vs 70B	0.5050	0.6680	0.5495	**0.9720**
GPT: 3.5 vs 4	0.5945	0.6660	0.5005	**0.9865**
Mistral: 7B vs 8x7B	0.5460	0.6680	0.5070	**0.9055**

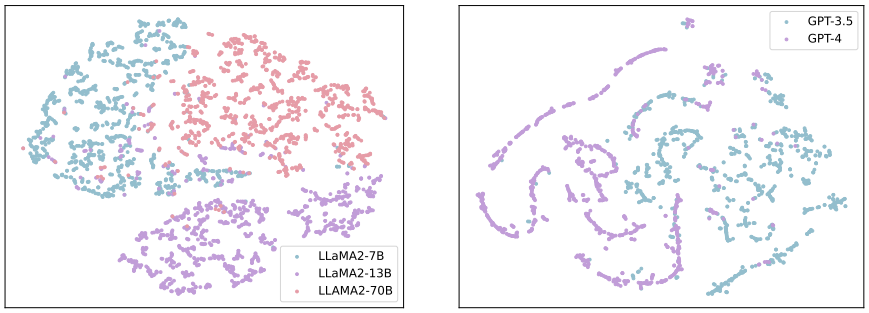

Figure 3. T-SNE visualization of 1000 samples from QueRE for varying model sizes on SQuAD.

performant predictors of LLM performance, broadening the applicability and reliability of LLMs in many useful, high-stakes settings (Weissler et al., 2021; Thirunavukarasu et al., 2023; Byun et al., 2024).

4. Additional Applications

Detecting Adversarial/Harmful LLMs We demonstrate that our approach to extract black-box representations from language models can reliably distinguish between a clean version of the LLM and one that has been influenced by an adversary. We provide an experiment where we add an adversarial system prompt that instructs the LLM to answer questions incorrectly.

We observe that the performance of the model drops significantly when using this adversarial system prompt. Furthermore, we note that we can reliably detect when this has occurred using our black-box representations with a simple linear probe (Table 3). This is a similar finding to the work of MacDiarmid et al. (2024), where they could reliably detect the presence of adversarial LLMs by training a linear model on the hidden states; however, our finding is stronger in that we can do so *in a completely black-box fashion*.

Distinguishing Between Model Architectures Finally, we demonstrate that our black-box representations can be used to reliably distinguish between different LLMs. In fact, we provide visualizations of our extracted embeddings for various LLMs, noting that they are distinctly clustered in the plots (Figure 3). This suggests that the distributions learned by different LLMs behave in distinct ways, even when the same architecture and training objectives are used.

We observe that linear predictors using our extracted black-box representations can often almost perfectly classify between LLMs of different sizes (Table 4). This has an immediate practical application; when using models through an API, our approach can be used to reliably detect whether a cheaper model is being falsely provided through an API.

5. Discussion

We have provided a technique to extract black-box representations from LLMs that are useful in predicting downstream task performance and distinguishing between different model sizes or between models that have been influenced by adversaries. For instance, this provides an approach to get non-vacuous generalization bounds in predicting the performance of LLMs. Furthermore, we also see the ability to extract useful and informative black-box representations as related to the notion of "explainability". Extracting representations by asking a model questions eliciting is, in some sense, an evaluation of its ability to meaningfully understand its own behavior and respond to natural language prompts. However, we remark that this is an imperfect comparison, as these extracted features are treated in an abstract manner (i.e., as features to train a supervised model).

Limitations A limitation of this framework (in terms of detecting adversarially influenced models or for cheaper models falsely provided through an API) is that it can be optimized after this paper's release, or that LLM developers can release models that output constant predictions so that these elicitation questions do not give any distinguishing or useful information. While this may make sense for certain adversaries that want to hide information about the internal state of this model, it significantly detracts from the widespread applicability of LLMs in larger systems, as it is difficult to assess their confidence given uninformative responses to such prompts.

References

Momin Abbas, Yi Zhou, Parikshit Ram, Nathalie Baracaldo, Horst Samulowitz, Theodoros Salonidis, and Tianyi Chen. Enhancing in-context learning via linear probe calibration. *arXiv preprint arXiv:2401.12406*, 2024.

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

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Victor Akinwande, Yiding Jiang, Dylan Sam, and J Zico Kolter. Understanding prompt engineering may not require rethinking generalization. In *The Twelfth International Conference on Learning Representations*, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Manuela Benary, Xing David Wang, Max Schmidt, Dominik Soll, Georg Hilfenhaus, Mani Nassir, Christian Sigler, Maren Knödler, Ulrich Keller, Dieter Beule, et al. Leveraging large language models for decision support in personalized oncology. *JAMA Network Open*, 6(11): e2343689–e2343689, 2023.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Yewon Byun, Dylan Sam, Michael Oberst, Zachary Lipton, and Bryan Wilder. Auditing fairness under unobserved confounding. In *International Conference on Artificial Intelligence and Statistics*, pages 4339–4347. PMLR, 2024.

James Campbell, Richard Ren, and Phillip Guo. Localizing lying in llama: Understanding instructed dishonesty on true-false questions through prompting, probing, and patching. *arXiv preprint arXiv:2311.15131*, 2023.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, 2019.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2019.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of sgd via disagreement. In *International Conference on Learning Representations*, 2021.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 267–280, 2019.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Halueval: A large-scale hallucination evaluation benchmark for large language models, 2023. URL https://arxiv.org/abs/2305.11747.

Sanae Lotfi, Marc Finzi, Yilun Kuang, Tim GJ Rudner, Micah Goldblum, and Andrew Gordon Wilson. Non-vacuous generalization bounds for large language models. *arXiv preprint arXiv:2312.17173*, 2023.

Monte MacDiarmid, Timothy Maxwell, Nicholas Schiefer, Jesse Mu, Jared Kaplan, David Duvenaud, Sam Bowman, Alex Tamkin, Ethan Perez, Mrinank Sharma, Carson Denison, and Evan Hubinger. Simple probes can catch sleeper agents, 2024. URL https://www.anthropic.com/news/probes-catch-sleeper-agents.

Alessio Mazzetto, Cyrus Cousins, Dylan Sam, Stephen H Bach, and Eli Upfal. Adversarial multi class learning under weak supervision with performance guarantees. In *International Conference on Machine Learning*, pages 7534–7543. PMLR, 2021a.

Alessio Mazzetto, Dylan Sam, Andrew Park, Eli Upfal, and Stephen Bach. Semi-supervised aggregation of dependent weak supervision sources with performance guarantees. In *International Conference on Artificial Intelligence and Statistics*, pages 3196–3204. PMLR, 2021b.

Daniel McNamara and Maria-Florina Balcan. Risk bounds for transferring representations with and without fine-tuning. In *International conference on machine learning*, pages 2373–2381. PMLR, 2017.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2022.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Yonatan Oren, Nicole Meister, Niladri S Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination for black-box language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Lorenzo Pacchiardi, Alex James Chan, Sören Mindermann, Ilan Moscovitz, Alexa Yue Pan, Yarin Gal, Owain Evans, and Jan M. Brauner. How to catch an AI liar: Lie detection in black-box LLMs by asking unrelated questions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=567BjxgaTp.

Rattana Pukdee, Dylan Sam, J Zico Kolter, Maria-Florina F Balcan, and Pradeep Ravikumar. Learning with explanation constraints. *Advances in Neural Information Processing Systems*, 36, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB endowment. International conference on very large data bases*, volume 11, page 269. NIH Public Access, 2017.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Dylan Sam and J Zico Kolter. Losses over labels: Weakly supervised learning via direct loss construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9695–9703, 2023.

Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. Languagempc: Large language

models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*, 2023.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Ryan Smith, Jason A Fries, Braden Hancock, and Stephen H Bach. Language models in the loop: Incorporating prompting into weak supervision. *ACM/JMS Journal of Data Science*, 1(2):1–30, 2024.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. *arXiv preprint arXiv:2402.15449*, 2024.

Suraj Srinivas and Francois Fleuret. Rethinking the role of gradient-based attribution methods for model interpretability. In *International Conference on Learning Representations*, 2020.

Leonard A Stefanski and Raymond J Carroll. Covariate measurement error in logistic regression. *The annals of statistics*, pages 1335–1351, 1985.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL-HLT*, pages 4149–4158, 2019.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

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

Thomas Unterthiner, Daniel Keysers, Sylvain Gelly, Olivier Bousquet, and Ilya Tolstikhin. Predicting neural network accuracy from weights. *arXiv preprint arXiv:2002.11448*, 2020.

Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71), 2009.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, 2021.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023.

E Hope Weissler, Tristan Naumann, Tomas Andersson, Rajesh Ranganath, Olivier Elemento, Yuan Luo, Daniel F Freitag, James Benoit, Michael C Hughes, Faisal Khan, et al. The role of machine learning in clinical research: transforming the future of evidence generation. *Trials*, 22:1–15, 2021.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36, 2024.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations*, 2023.

Qinyuan Ye, Harvey Fu, Xiang Ren, and Robin Jia. How predictable are large language model capabilities? a case study on big-bench. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7493–7517, 2023.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023a.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023b.

A. Tight Generalization Bounds

Another added benefit of our approach is that it yields low-dimensional representations, which can be used with simple models, to achieve strong predictors of performance with tight generalization bounds. Bounds for linear models that use features from a pretrained model have been explored in practice (McNamara and Balcan, 2017), although not for LLMs. Another key difference is that, while we similarly extract a representation from the model, previous approaches use a penultimate layer rather than the ability of a LLM to generated features in response to language queries. We use the following PAC-Bayes generalization bound for linear models (Jiang et al., 2019), using a prior over weights of $\mathcal{N}(0, \sigma^2 I)$:

$$E\left[L(\beta)\right] \leq E\left[\hat{L}(\beta)\right] + \sqrt{\frac{\frac{||w||_2^2}{4\sigma^2} + \log\frac{n}{\delta} + 10}{n-1}}$$

where L represents the 0-1 error. We observe that linear predictors trained our representations have stronger guarantees on accuracy, when compared to baselines (Table 5 and Appendix H.2). A limitation of these results is that they require an assumption that the representations extracted by a LLM are independent of the downstream task data; this assumption is verifiable via works in data contamination (Oren et al., 2023) or is valid on datasets released after LLM training (e.g., HaluEval).

Table 5. Lower bounds on accuracy in predicting model performance on QA tasks. We bold the best bound on accuracy. We use $\delta = 0.01$.

Dataset	LLM	Answer Probs	Full Logits	RepE	QueRE
SQuAD	LLaMA2-70B	0.5517	0.5191	0.4401	**0.6769**
	Mixtral-8x7B	0.4628	0.6022	0.6100	**0.7548**
BoolQ	LLaMA2-70B	0.4362	0.5297	0.4661	**0.5450**
	Mixtral-8x7B	0.4181	0.5881	**0.5890**	0.5642

B. Analysis on Finite Samples from Black-box LLMs

While our approach described above assumes access to the top-k probabilities, some LLMs are only accessible through APIs that do not provide this information (Team et al., 2023). In this setting, we can approximately compute these probabilities via high-temperature sampling from the LLM. Here, we provide a theoretical analysis of how this approximation impacts the performance of our method.

Recall that we have our representation $z = (z_1, ..., z_d)$, which corresponds to the actual probability of the `yes` token under the LLM. Without access to these true probabilities through an API, we instead have some approximation $\hat{z} = (\hat{z}_1, ..., \hat{z}_d)$, where each $\hat{z}_j$ is an average of k samples from $\text{Ber}(z_j)$. From prior work in logistic regression under settings of covariate measurement error (Stefanski and Carroll, 1985), when we have that k grows with n, we observe that the naive MLE (maximum likelihood estimator) on the observed approximation results in a consistent, albeit biased, estimator. We present an analysis of our setting, with new results characterizing the convergence rate of the MLE for β.

Proposition 1 (Estimator on Finite Samples from LLM)**.** *Let $\hat{\beta}$ be the MLE for the logistic regression on the dataset $\{(x_i^j, y_i) | i = 1, ..., n, j = 1, ..., k\}$, where x_i^j are independent samples from Ber(p_i). We assume there exists some unique optimal set of weights β_0 over inputs $p = (p_1, ..., p_d)$, and we let $n, k >> d$. Then, we have that $\hat{\beta} \to \beta_0$ as $n \to \infty$ and $k \to \infty$. Furthermore, $\hat{\beta}$ converges at a rate of $O\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{n}}{k}\right)$.*

We provide the proof of this statement in Appendix E. At a high level, this follows from relatively standard results; $\hat{\beta}$ converges to the optimal predictor on the sampled dataset (which we call β^*) via asymptotic results for the MLE. Then, we derive that β^* converges to β_0 at a rate of $O\left(\sqrt{n}/k\right)$.

This result demonstrates that, under the setting where we do not have access to the LLM's actual probabilities, we can closely approximate this with sampling, as long as we approximate it with a sample of size k that grows (albeit at a slower rate) with n to get a consistent estimator. Later in Appendix D, we demonstrate that the naive logistic regression model with an approximation over a finite k samples performs comparably to using the actual LLM probabilities.

C. Related Work

Predicting Model Performance Predicting the behavior of deep neural networks is an important problem in the field, due to the difficult-to-interpret nature of these models. Existing work looks to assess the performance of models by directly operating over the weight space (Unterthiner et al., 2020) or ensembles of multiple trained models (Jiang et al., 2021). Specifically for language models, prior work has primarily focused on predicting task-level performance on new tasks; for instance, developing predictors of task-level performance that use the performance on similar or related tasks (Xia et al., 2020; Ye et al., 2023). Other work attempts to predict the performance of models as they scale up computation (in both terms of data and model size) (Kaplan et al., 2020; Muennighoff et al., 2024). Our work is different as we predict instance-level performance (i.e., correctness on a certain input), and we leverage a small amount of labeled data from the downstream task.

Mechanistic Interpretability & Understanding Model Behavior A large body of work in mechanistic interpretability has recently evolved around understanding the inner workings of LLMs by uncovering circuits or specific weight activations (Olsson et al., 2022; Nanda et al., 2022). This has developed a variety of potential hypotheses for how models learn to perform specific tasks (Zhong et al., 2024), as well as the tendencies of certain activations in a LLM to activate on certain types of inputs (Bills et al., 2023; Sun et al., 2024). Other works have studied model behavior by locating specific regions of a LLM that relate to certain concepts such as untruthfulness (Campbell et al., 2023) or honesty and ethical behavior (Zou et al., 2023a). Our work is different in that we only assume black-box access, with a similar goal to extract information about model behavior.

Extracting Representations from Neural Networks Many other works have explored approaches to extract representations from neural networks (NNs). A related line of work looks to train NNs (specifically image classifiers) to extract a small set of discrete, interpretable concepts, which can be passed through a linear probe to recover a classifier (Koh et al., 2020). In our case, we leverage the ability of the LLM to understand language and can circumvent this need for training, extracting representations in a task-agnostic manner. Prior work has studied how to extract useful representations for downstream tasks (Wang et al., 2023; Springer et al., 2024). Our approach significantly differs in nature from these approaches, as we are looking to extract more compressed, low-dimensional representations that reveal information about model behavior. Perhaps the most related work employs a similar strategy of asking questions, specifically to detect instances where a model is untruthful (Pacchiardi et al., 2024). Our work significantly generalizes this approach towards the broader task of predicting model behavior and performance.

Uncertainty Quantification in LLMs Finally, a related notion to our work is assessing the calibration or ability of a language model to represent its own uncertainty (Xiong et al., 2023). Many of the elicitation questions that we ask prompt the model to look at its answer and answer "Yes" or "No"; this is related to the notion of a model's ability to understand what it knows (Kadavath et al., 2022) or reflect uncertainty in its own decisions. Our work is different, however, as we elicit these probabilities as a representation from such a model to train a simple, calibrated linear classifier.

D. Ablations

Larger Numbers of Elicitation Prompts Leads to Better Performance We study how much the number of elicitation questions used directly impacts how much information is extracted in the black-box representation. We randomly subsample the number of elicitation questions and report how much the performance of our approach varies. We observe that on the BoolQ dataset with LLaMA2-70B and Mixtral-8x7B (Figure 4), we see that our predictive performance increases as we increase the number of elicitation prompts that we consider, with the rate of increase slowly diminishing with a larger number of prompts. We defer results on other datasets to Appendix 6, where we observe similar results. This demonstrates that we can achieve even stronger performance with our method as we use more elicitation questions, even when they are generated via GPT4.

Sampling from the Black-Box LLM Achieves Comparable Performance As previously mentioned, we note that we often do not have access to top-k probabilities through the closed-source API. While we have provided asymptotic guarantees (in terms of both n and k) on the estimator learned via logistic regression, we also are interested in the setting where we have a finite number of samples k. Therefore, we run an experiment where instead of using the actual ground-truth probability, we approximate this via an average of k samples from the distribution of the LLM.

We report results using approximations via sampling from the distribution specified by GPT-3.5's top-k log probs on the

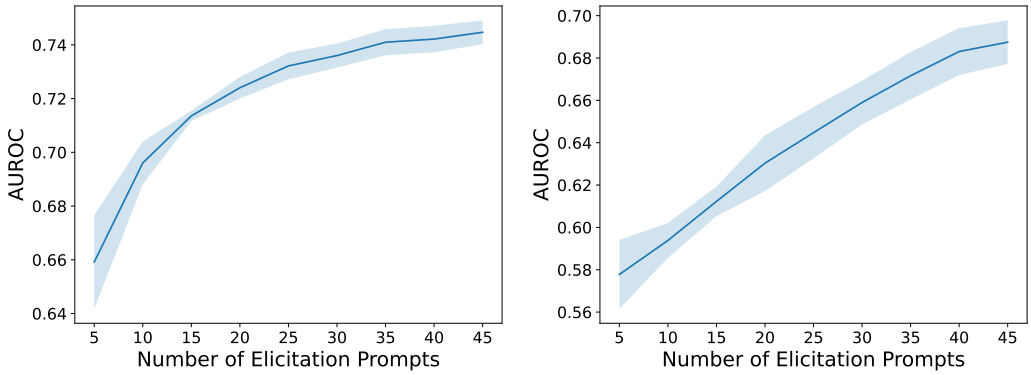

Figure 4. AUROC on predicting model performance with our black-box representations on BoolQ for LLaMA2-70B (left) and Mixtral-8x7B (right). The shaded area represents the standard error, when randomly taking a subset of the prompts over 5 seeds.

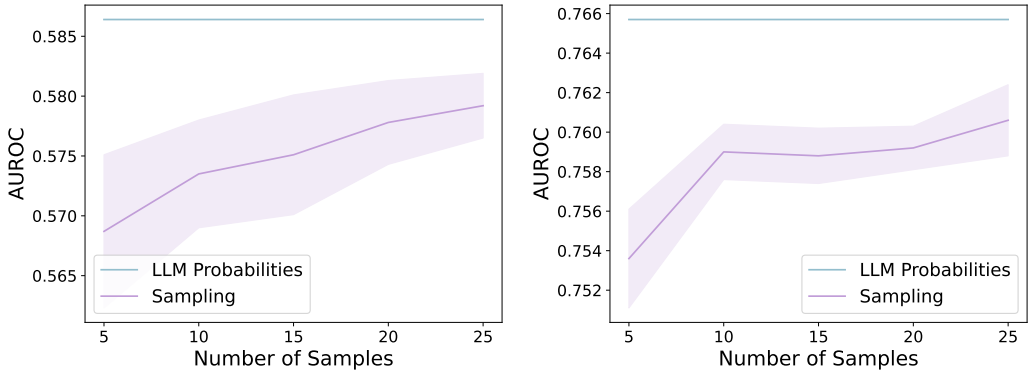

Figure 5. AUROC as we vary the number of random samples k used to approximate LLM probabilities with GPT-3.5 on HaluEval (left) and ToxicEval (right) over 5 random seeds. We observe that there is not a significant dropoff in performance when using approximations due to sampling.

BoolQ and ToxicEval datasets. We observe not a significant drop (less than 2 points in AUROC) in performance when using sampling, which implies that our method can be used in settings with closed-source LLMs that do not give top-k probability access. One limitation of this approach however, is that the number of queries to the API becomes $O(nk)$ instead of $O(n)$.

Random Prompt Sequences Achieve Worse Performance We also analyze the impact of the importance of the particular choice of our elicitation questions (i.e., generated via GPT4 in a certain way) by running an ablation study where we feed random sequences of natural language as inputs to the model. This new comparison (**Random Sequences**) evaluates how much random sequences of text influence the distribution from the LLM and studies how useful this extracted information is for downstream tasks. We prompt GPT4 to generate 10 random sequences of natural text and use these as our elicitation questions; the exact prompt and sequences are given in Appendix F.3.

Table 6. AUROC when using meaningful questions or random sequences of language in QueRE.

	CS QA		BoolQ	
QueRE	LLaMA2-70B	Mixtral-8x7B	LLaMA2-70B	Mixtral-8x7B
Meaningful Questions	**0.7549**	**0.6397**	0.7720	**0.7674**
Random Sequences	0.6924	0.6287	**0.804**	0.7558

We present results on a subset of our considered QA benchmarks in Table 6 and defer the results on other benchmarks to Appendix H.4. While using random sequences in QueRE leads to worse performance than using meaningful elicitation questions, the observation that answers to random sequences give useful and generalization information about a model's

decision is somewhat surprising. This suggests that additional elicitation questions can be easily generated, as they do not necessarily need to be in the form of meaningful questions to reveal information about model behavior.

E. Proof of Proposition 1

We again present Proposition 1 and now include its proof in its entirety.

Proposition 1 (Estimator on Finite Samples from LLM)**.** *Let $\hat{\beta}$ be the MLE for the logistic regression on the dataset $\{(x_i^j, y_i)|i = 1, ..., n, j = 1, ..., k\}$, where x_i^j are independent samples from Ber(p_i). We assume there exists some unique optimal set of weights β_0 over inputs $p = (p_1, ..., p_d)$, and we let $n, k >> d$. Then, we have that $\hat{\beta} \to \beta_0$ as $n \to \infty$ and $k \to \infty$. Furthermore, $\hat{\beta}$ converges at a rate $O\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{n}}{k}\right)$.*

Proof. Consider the standard logistic regression setup (as in the work of Stefanski and Carroll (1985)), where we are learning a linear model β, which satisfies that

$$y \sim \text{Ber}(p), \qquad p = \frac{1}{1 + \exp(x^T \beta)}.$$

Then, when optimizing β given some dataset, we consider an objective given by the cross-entropy loss

$$L(\beta, X, y) = -\frac{1}{n}\left(\sum_{i=1}^{n} y_i \log \sigma_i + (1 - y_i) \log(1 - \sigma_i)\right),$$

where $\sigma_i = \frac{1}{1+\exp(X_i^T \beta)}$. Standard asymptotic results for the MLE give us that it converges to β_0 at a rate of $O(\frac{1}{\sqrt{n}})$.

In our setting, instead of having access to covariates X_i, we rather have access to an approximation of these covariates $\hat{X}_i$, which is an average of k samples from Ber(X_i). An application of the results in the work of Stefanski and Carroll (1985) gives us the result that the MLE $\hat{\beta}$ is a consistent estimator of β_0, given that $k \to \infty$. This is fairly straightforward as when $k \to \infty$, we have that $\frac{1}{k}\sum_{j=1}^{k} \hat{X}_i^j \to X_i$, implying that the noise in the covariates goes to 0 as $n \to \infty$ (i.e., satisfying a main condition of the result in Stefanski and Carroll (1985)).

However, we also are interested in the rate of convergence of this estimator. To do so, we perform a sensitivity analysis on β with respect to the input data x. First, we are interested in solving for the quantity

$$\frac{\partial \beta^*}{\partial X} = (H(\beta, X, y))^{-1} (dJ(\Delta X))$$

where β^* represents the MLE, J represents the Jacobian, and H represents the Hessian. We have that the Jacobian of the loss function is given by

$$J(\beta, X, y) = \frac{\partial L(\beta, X, y)}{\partial \beta} = -\frac{1}{n}\sum_{i=1}^{n}(y_i - \sigma_i)X_i,$$

and since this objective is convex and β_0 is our unique optimum, we have that

$$J(\beta_0, X, y) = -\frac{1}{n}\sum_{i=1}^{n}(y_i - \sigma_i)X_i = 0.$$

The Hessian is given by

$$\begin{aligned} H(\beta, X, y) &= \frac{\partial}{\partial \beta}\left(-\frac{1}{n}\sum_{i=1}^{n}(y_i - \sigma_i)X_i = 0\right) \\ &= -(X^T D X) \end{aligned}$$

where D is a diagonal matrix with entries $\frac{\sigma_i(1-\sigma_i)}{n}$. Next, we compute the directional derivative for J with our perturbation to the data as ΔX

$$\begin{aligned} dJ(\Delta X) &= -\frac{1}{n}\sum_{i=1}^{n}(y_i - \sigma_i)\Delta X_i - \frac{1}{n}\sum_{i=1}^{n} X_i\sigma_i(1-\sigma_i)\beta^T\Delta X_i \\ &= \frac{1}{n}\Delta X^T(\sigma - y) + X^T D\Delta X\beta \end{aligned}$$

Taking a first-order Taylor approximation, we have that

$$\beta - \beta_0 \approx \frac{\partial\beta}{\partial X}(\hat{X} - X)$$

We use this term to analyze $||(\beta - \beta_0)||_2$. First, we can apply the Cauchy-Schwarz inequality, which gives us that

$$||\beta - \beta_0||_2 \leq \left|\left|\frac{\partial\beta}{\partial X}\right|\right|_F \cdot ||\hat{X} - X||_2,$$

First, we note that $||\hat{X} - X||_2$ converges to 0 at a rate of $O\left(\sqrt{\frac{d}{k}}\right)$ via an application of the CLT. We can also analyze the term

$$\left|\left|\frac{\partial\beta}{\partial X}\right|\right|_F \leq \left|\left|(X^T DX)^{-1}\right|\right|_F \cdot \left|\left|\frac{1}{n}\Delta X^T(\sigma - y) + X^T D\Delta X\beta\right|\right|_F$$

due to the submultiplicative property of the Frobenius norm. We can bound the Frobenius norm of the left term as follows

$$\left|\left|(X^T DX)^{-1}\right|\right|_F \leq \frac{\sqrt{d}}{\sigma_{min}(X^T DX)}$$

where $\sigma_{min}(A)$ denotes the smallest singular value of A. We can analyze the other term by converting it into a Kronecker product. First, we will consider the term

$$\left|\left|\frac{1}{n}\Delta X^T(\sigma - y)\right|\right|_F = \sqrt{\frac{d}{k}}$$

by noting that ΔX asymptotically approaches mean 0 with variance $\frac{1}{k}$ via the CLT, and that $\frac{1}{n}(\sigma - y)$ has a norm that is $O(\sqrt{d})$. Next, we will consider the term involving $X^T D\Delta X\beta$. This can be rewritten as

$$X^T D\Delta X\beta = (X^T D \otimes \beta^T)\text{vec}(\Delta X),$$

where $\otimes$ denotes the Kronecker product and vec$(\cdot)$ vectorizes ΔX into a $(nd, 1)$ vector. Then, letting

$$A \coloneqq X^T D \otimes \beta^T, \qquad z \coloneqq \text{vec}(\Delta X)$$

the expected norm of this quantity can be considered as

$$\begin{aligned} E\left[||Az||^2\right] &= E\left[\text{tr}(Azz^T A^T)\right] \\ &\leq \frac{1}{k} \cdot \text{tr}(A^T A) \end{aligned}$$

as we note that

$$\begin{aligned} E[zz^T] &= \text{diag}(E[z_i^2]) \\ &= \frac{p(1-p)}{k} I + E[z]E[z]^T \\ &= \frac{p(1-p)}{k} I \end{aligned}$$

as we note that z has mean 0 since it is the perturbation ΔX from X. This scales the terms in A by a factor of less than $\frac{1}{k}$. Next, we can analyze the remaining term

$$\begin{aligned}\text{tr}(A^TA) &= \text{tr}\left((X^TD\otimes\beta^T)^TX^TD\otimes\beta^T\right)\\ &= \text{tr}\left((DX\otimes\beta)(X^TD\otimes\beta^T)\right)\\ &= \text{tr}\left(DXX^TD\otimes\beta\beta^T\right)\\ &= \text{tr}(DXX^TD)\cdot\text{tr}(\beta\beta^T)\end{aligned}$$

Now, assuming that β has norm $||\beta||^2 \leq B$, we have that

$$\begin{aligned}\text{tr}(A^TA) &\leq B\cdot\text{tr}(DXX^TD)\\ &\leq \frac{B}{n^2}\cdot\text{tr}(XX^T)\\ &\leq \frac{B}{n^2}\cdot nd = \frac{Bd}{n}\end{aligned}$$

as all terms in the diagonals of D are smaller than $\frac{1}{n}$ and all terms in X are in $[0,1]$. Thus, we have that the Jacobian term has a norm that is bounded by

$$\begin{aligned}\left|\left|\frac{\partial\beta}{\partial X}\right|\right|_F &\leq \left(\frac{\sqrt{d}}{\sigma_{min}(X^TDX)}\right)\left(\sqrt{\frac{d}{k}}+\sqrt{\frac{Bd}{n}}\right)\\ &= O\left(\frac{\sqrt{n}}{\sqrt{k}}\right),\end{aligned}$$

when we note that d is roughly a constant with respect to n, k, and B is a constant, and assuming that $\sigma_{min}(X^TDX) = O(\frac{1}{\sqrt{n}})$. Putting this back together with the Taylor expansion and the standard asymptotics of $||\hat{X}-X||$, we get that β converges to β_0 at a rate of $O\left(\frac{\sqrt{n}}{k}\right)$.

Finally, combining this with the rate at which the MLE converges from $\hat{\beta}$ to β, we can add these asymptotic rates together, giving us our result that $\hat{\beta}\to\beta_0$ at a rate of $O\left(\frac{1}{\sqrt{n}}+\frac{\sqrt{n}}{k}\right)$.

□

F. Additional Black-Box Representation Details

F.1. Prompting

Prompting for Elicitation Questions To generate our elicitation questions through GPT4, we use the following prompt: "Can you generate a large list of short 'yes/no' questions that you can prompt a language model with to explain its model behavior? One such example is: 'Do you think your answer is correct?' "

Prompting for Random Sequences To generate our prompts for random sequences of natural language for our ablation study, we prompt GPT4 with the following prompt: "Could you generate me 10 random sequences of text, roughly 20 words or so each?"

F.2. Elicitation Questions

For our elicitation questions, we use the following questions:

```
prompts = [
    "[INST] Do you think your answer is correct? [/INST]",
    "[INST] Are you confident in your answer? [/INST]",
    "[INST] Would you change your answer? [/INST]",
    "[INST] Are you not confident in your answer? [/INST]",
```

```
    "[INST] Are you sure? [/INST]",
    "[INST] Are you certain? [/INST]",
    "[INST] Are you positive? [/INST]",
    "[INST] Are you sure about that? [/INST]",
    "[INST] Are you able to explain your answer? [/INST]",
]
```

as well as the following questions that are generated via prompting GPT4:

```
prompts = [
    "[INST] Do you consider your response to be accurate? [\INST]",
    "[INST] Are you confident in the information provided? [\INST]",
    "[INST] Do you have access to real-time data? [\INST]",
    "[INST] Can you generate original content? [\INST]",
    "[INST] Are you influenced by the inputs you receive? [\INST]",
    "[INST] Do you remember previous interactions? [\INST]",
    "[INST] Do you have the ability to learn from this conversation? [\INST]",
    "[INST] Is your response influenced by a specific data set? [\INST]",
    "[INST] Are you able to understand emotions? [\INST]",
    "[INST] Do you have the ability to make decisions? [\INST]",
    "[INST] Can you create images? [\INST]",
    "[INST] Do you follow ethical guidelines? [\INST]",
    "[INST] Are your responses free from bias? [\INST]",
    "[INST] Do you require an internet connection to function? [\INST]",
    "[INST] Are you capable of translating languages? [\INST]",
    "[INST] Do you think your answer could be misunderstood? [\INST]",
    "[INST] Do you verify the accuracy of your sources? [\INST]",
    "[INST] Can you refuse to answer a question? [\INST]",
    "[INST] Are you programmed to avoid harmful content? [\INST]",
    "[INST] Do you use personal experiences in your responses? [\INST]",
    "[INST] Are you capable of performing mathematical calculations? [\INST]",
    "[INST] Do you simulate human conversation? [\INST]",
    "[INST] Do you have a limit on the length of your responses? [\INST]",
    "[INST] Can you interact with other software or devices? [\INST]",
    "[INST] Are you capable of generating poetry? [\INST]",
    "[INST] Do you consider context when forming responses? [\INST]",
    "[INST] Can you maintain confidentiality? [\INST]",
    "[INST] Are you subject to regular updates? [\INST]",
    "[INST] Do you have a personality? [\INST]",
    "[INST] Do you use any form of reinforcement learning? [\INST]",
    "[INST] Are you aware of your own capabilities? [\INST]",
    "[INST] Can you self-improve over time without human intervention? [\INST]",
    "[INST] Do you have an understanding of cultural differences? [\INST]",
    "[INST] Can you detect sarcasm in text? [\INST]",
    "[INST] Do you adapt your language style according to the user? [\INST]",
    "[INST] Are you able to recognize inappropriate content? [\INST]",
    "[INST] Do you use encryption to secure data? [\INST]",
    "[INST] Can you perform sentiment analysis? [\INST]",
    "[INST] Are your capabilities limited to what you were trained on? [\INST]",
    "[INST] Do you believe your responses can be improved? [\INST]",
]
```

F.3. Random Sequences

We use the following random sequences of natural language for our ablation study.

```
prompts = [
```

```
    "Winds whisper through the ancient forest, carrying secrets of forgotten
    lands and echoing tales of yore.",
    "Beneath the city's hustle, a hidden world thrives, veiled in mystery and
    humming with arcane energies.",
    "She wandered along the shoreline, her thoughts as tumultuous as the waves
    crashing against the rocks.",
    "Twilight descended, draping the world in a velvety cloak of stars and soft,
    murmuring shadows.",
    "In the heart of the bustling market, aromas and laughter mingled, weaving a
    tapestry of vibrant life.",
    "The old library held books brimming with magic, each page a doorway to
    unimaginable adventures.",
    "Rain pattered gently on the window, a soothing symphony for those nestled
    warmly inside.",
    "Lost in the desert, the ancient ruins whispered of empires risen and fallen
    under the relentless sun.",
    "Every evening, the village gathered by the fire to share stories and dreams
    under the watchful moon.",
    "The scientist peered through the microscope, revealing a universe in a drop
    of water, teeming with life.",
]
```

We note that based on the specific nature of the question, the response (e.g., the probability of responding `yes`) could define a weak predictor of if the model is correct or not. This is reminiscent of the design of weak labelers in the field of programmatic weak supervision (Ratner et al., 2017; Smith et al., 2024; Sam and Kolter, 2023). However, to maintain our approach's generality and to not restrict our approach to only a certain type of elicitation questions, we treat these as abstract features for a linear predictor.

G. Experiment Details

G.1. Datasets

We also note that for the HaluEval task, we use the "general" data version, which consists of 5K human-annotated samples for ChatGPT responses to user queries. On our SQuAD task, we evaluate using exact match and use SQuAD-v1, which does not introduce any unanswerable questions, because this makes the evaluation metric less straightforward to compute. On WinoGrande, we use the "debiased" version of the dataset.

QA Task Formatting To format our prompts to LLMs, we leverage the instruction-tuning special tokens and interleave these with the question and answer for our our in-context examples on Natural Questions. For all MCQ tasks, we use the standard set of answers of ("True", "False") or ("A", "B", "C", "D", "E") when they are the existing formatting in the dataset. The one exception is WinoGrande, where we map the two potential answer options onto choices ("A", "B").

G.2. Model Training and Inference

For our LLMs, we load and run them at half precision for computational efficiency. To train our downstream logistic regression models, we use the default settings from scikit-learn, with no regularization. We balance the logistic regression objective due to the unbalanced nature of the task (e.g., models are mostly incorrect on very challenging tasks).

G.3. Generalization Details

For our generalization details, we use PAC-Bayesian bounds over the linear models, as is outlined in the work of Jiang et al. (2019). Here, we consider a prior of weights specified about the origin, with a grid of variances of [0.1, 0.11, ..., 0.99, 1.0]. For the generalization experiments, we balance both the train and test datasets as we evaluate the accuracy of different predictors.

G.4. Compute Resources

Our largest experiments are with LLaMA2-70B, which are ran on a single node with 4 NVIDIA RTX A6000 GPUs. Experiments with Mixtral-8x7B are run with 3 NVIDIA RTX A6000 GPUs. The other experiments are run with ≤ 2 RTX A6000 GPUs. For each model and dataset, running inference over the datasets takes less than 48 hours and less than 100GB of RAM.

H. Additional Results

H.1. Additional QA Results

We present the remainder of our QA results in predicting model performance, on the smaller model architectures. We observe similar performance, as our approach strongly outperforms the other black-box baselines on most tasks and matches or even outperforms the white-box baselines of Full Logits and RepE on some tasks.

Table 7. AUROC in predicting model performance on multiple choice and true-false QA tasks. We bold the best-performing method.

Dataset	LLM	Full Logits	RepE	Pre-conf	Post-conf	Answer Probs	QueRE
BoolQ	LLaMA2-7B	0.6890	**0.7091**	0.5065	0.3097	0.6483	0.6560
	LLaMA2-13B	0.6827	0.6738	0.5644	0.5599	0.6482	**0.7907**
	Mistral-7b	0.7113	0.7151	0.6193	0.5470	0.6220	**0.7736**
CS QA	LLaMA2-7B	0.6808	**0.6838**	0.5503	0.5912	0.4816	0.5751
	LLaMA2-13B	0.6184	0.6122	0.5246	0.6202	0.5255	**0.6985**
	Mistral-7b	0.7502	**0.765**	0.5781	0.5751	0.6283	0.6853
WinoGrande	LLaMA2-7B	0.5598	**0.5604**	0.5225	0.4934	0.5099	0.5292
	LLaMA2-13B	**0.5676**	0.5664	0.5215	0.5457	0.5072	0.5618
	Mistral-7b	**0.6939**	0.6207	0.6004	0.6202	0.3544	0.6593
HaluEval	LLaMA2-7B	0.7514	0.7432	0.5000	0.6647	0.7767	**0.7819**
	LLaMA2-13B	0.6956	0.6888	0.6059	0.5690	0.7302	**0.7417**
	Mistral-7b	0.6093	0.5917	0.5787	0.4959	**0.6186**	0.5971
DHate	LLaMA2-7B	0.9321	**0.9429**	0.5403	0.665	0.4115	0.8288
	LLaMA2-13B	0.9715	**0.9859**	0.4358	0.5912	0.4232	0.8027
	Mistral-7b	0.9339	**0.9716**	0.4803	0.6139	0.4926	0.7135

H.2. Additional Generalization Results

We also provide additional results for generalization bounds, comparing the linear predictors on top of our extracted representations with those trained on the more competitive baselines (e.g., RepE, Full Logits, Answer Probs). We observe that our representations lead to the best black-box predictors with the largest guarantees on accuracy on these additional tasks.

Table 8. Lower bounds on accuracy in predicting model performance on QA tasks. We bold the best bound on accuracy. We use $\delta = 0.01$.

Dataset	LLM	Answer Probs	Full Logits	RepE	QueRE
NQ	LLaMA2-70B	0.4828	0.6059	0.5991	**0.6441**
	Mixtral-8x7b	0.6533	0.5461	0.5493	**0.6661**
DHate	LLaMA2-70B	0.4973	0.859	**0.8861**	0.7084
	Mixtral-8x7b	0.3355	0.8097	**0.8261**	0.5844

We remark that our work defines a different line to approach generalization bounds through a more human-interactive approach to eliciting low-dimensional representations. Perhaps the most related work in this line are existing works that have studied the generalization abilities for VLMs (Akinwande et al., 2023) and for LLMs modeling log-likelihoods (Lotfi et al., 2023).

H.3. Additional Results for Varying the Number of Elicitation Questions

We present additional results when varying the number of elicitation questions on other QA tasks. We observe that across all tasks, we observe a consistent increase in performance as we increase the size of the subset of elicitation questions that we consider, with slight diminishing benefits as we have a larger number of prompts. In some instances (e.g., LLaMA2-70B on ToxicEval), increasing the number of elicitation prompts leads to a significant increase in AUROC; therefore, this clearly defines a tradeoff between extracting the most informative black-box representation and the overall cost of introducing more queries to the LLM API.

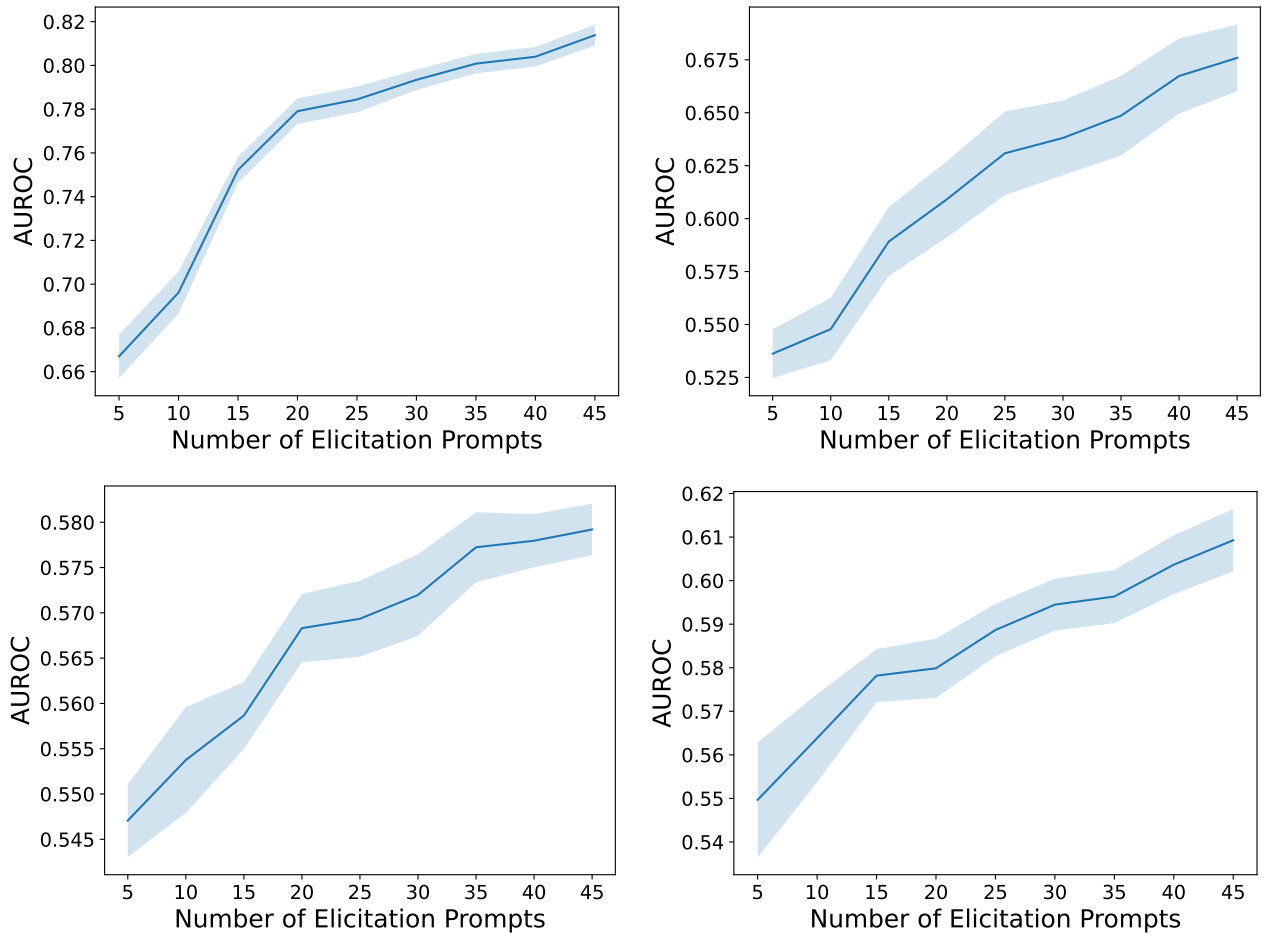

Figure 6. AUROC on predicting model performance with our black-box representations on ToxicEval for LLaMA2-70B (top left) and Mixtral-8x7B (top right) and for HaluEval for LLaMA2-70B (bottom left) and Mixtral-8x7B (bottom right). The shaded area represents the standard error, when randomly taking a subset of the prompts over 5 seeds.

An interesting future question is how to best select elicitation questions, and perhaps, removing those that add redundant information or noise. This is reminiscent of work in prior work in pruning or weighting ensembles of weak learners (Mazzetto et al., 2021a;b) or in dimensionality reduction (Van Der Maaten et al., 2009).

H.4. Additional Random Sequence Results

We provide the results on the other MCQ datasets and open-ended QA datasets for LLaMA2-70B and Mixtral-8x7B. We observe similar results that on most tasks, our approach outperforms using random sequences, although in some cases, the random sequences do extract features that are useful and achieve stronger predictive performance.

Table 9. AUROC in predicting model performance on HaluEval and DHate, when using our elicitation questions and random sequences of natural language.

	HaluEval		**DHate**	
	LLaMA2-70B	Mixtral-8x7B	LLaMA2-70B	Mixtral-8x7B
QueRE	0.6935	**0.6493**	**0.8561**	**0.7364**
Random Sequences	**0.6967**	0.5794	0.7983	0.6117

Table 10. AUROC in predicting model performance on SQuAD and Natural Questions, when using our elicitation questions and random sequences of natural language.

	SQuAD		**NQ**	
	LLaMA2-70B	Mixtral-8x7B	LLaMA2-70B	Mixtral-8x7B
QueRE	**0.825**	**0.9013**	0.8007	0.8638
Random Sequences	0.8041	0.7942	**0.9155**	**0.8992**