Efficiently Computing Susceptibility to Context in Language Models

Tianyu Liu   Kevin Du   Mrinmaya Sachan   Ryan Cotterell

{tianyu.liu, kevin.du, mrinmaya.sachan, ryan.cotterell}@inf.ethz.ch

ETH zürich

Abstract

One strength of modern language models is their ability to incorporate information from a user-input context when answering queries. However, they are not equally sensitive to the subtle changes to that context. To quantify this, Du et al. (2024) gives an information-theoretic metric to measure such sensitivity. Their metric, susceptibility, is defined as the degree to which contexts can influence a model's response to a query at a distributional level. However, exactly computing susceptibility is difficult and, thus, Du et al. (2024) fall back on a Monte Carlo approximation. Due to the large number of samples required, the Monte Carlo approximation is inefficient in practice. As a faster alternative, we propose Fisher susceptibility, an efficient method to estimate the susceptibility based on Fisher information. Empirically, we validate that Fisher susceptibility is comparable to Monte Carlo estimated susceptibility across a diverse set of query domains despite its being $70\times$ faster. Exploiting the improved efficiency, we apply Fisher susceptibility to analyze factors affecting the susceptibility of language models. We observe that larger models are as susceptible as smaller ones.[1]

1 Introduction

Much of current language models' (LM) capabilities and success are due to their responsiveness to different user-input contexts, e.g., prompts, and ability to integrate those contexts with prior knowledge (Brown et al., 2020; Bubeck et al., 2023, *inter alia*). Given this ability, it is natural for us to wonder how easily a model's prior belief can be changed by an input context. For example, a language model might correctly complete the prompt *Here Comes the Sun is performed by* with *The Beatles*, but wrongly when this prompt is prepended with a random context such as *Falafel wraps make ostriches burp.*; such a context might distract the model away from its original behavior.

Recently, Du et al. (2024) studied how easily input contexts can skew language models to give answers to queries that are different than the answers stored in their prior knowledge from an information-theoretic point of view. They propose a metric, termed *susceptibility*, to quantify the discrepancy between a model's prior belief and updated belief after seeing the input contexts as the mutual information between input contexts and answers from a specific language model. This metric makes use of the language model's *distribution* over answers rather than relying on an approximate argmax, as is common in previously proposed metrics (Shi et al., 2023; Wang et al., 2024). Susceptibility, as an information-theoretic metric, captures changes in model behavior that might not necessarily surface while decoding an answer.

Computing susceptibility requires, in principle, summing over all possible answers, a countably infinite set. In the general case, the authors know of no efficient algorithm to perform such a summation; Du et al. (2024) propose a Monte Carlo approximation. Such a sampling approximation, however, is computationally expensive: to compute the susceptibility for one query, one needs to execute a forward pass of the neural language model for each context in the sample set, making it inefficient to be applied to large-scale datasets. For example, Du et al. (2024) consider sampling 600 contexts for every individual query, and thus require 600 forward passes. In light of the computation required for the Monte Carlo approximation to susceptibility, we propose a more efficient approximation based on Fisher information that does not require sampling to estimate the susceptibility; we term this approximation *Fisher susceptibility*.

We conduct experiments across queries from 122 relation domains (e.g., *alumniOf* or *capitalOf*) in the YAGO knowledge graph (Suchanek et al.,

---

2007) with a variety of model sizes (e.g., from 70 million to 8 billion parameters), language model families (e.g., Pythia (Biderman et al., 2023), LLaMA (Touvron et al., 2023), and GPT-2 (Brown et al., 2020)), and fine-tuning schemes (e.g., instruction-tuning). First, to empirically validate Fisher susceptibility, we show that it is tightly correlated with Du et al.'s (2024) Monte Carlo approximation to susceptibility across domains while also benchmarking its speed. In these experiments, we find that compared to the Monte Carlo estimation with a sample size of 256, Fisher susceptibility exhibits a $70\times$ improvement in runtime. Then, we use the increased efficiency of Fisher susceptibility to investigate further initial questions like the one posed in the first paragraph: how susceptible are language models across different sizes and model families, and what factors influence a model's Fisher susceptibility for a query? We find that larger models are not less susceptible than smaller ones and that instruction-tuning does not help reduce susceptibility. We further find that queries about well-known entities are equally susceptible as less frequent ones under Fisher susceptibility, which contrasts with the finding from Du et al. (2024) that susceptibility is negatively related with expected familiarity when contexts might be relevant. To the extent that this is not an approximation error, this finding suggests that, regardless of how much prior knowledge it has about a query, a language model is still susceptible to contexts and can integrate new information from them.

2 Susceptibility to Context in LMs

Language models are capable of answering a wide range of queries formulated in natural language, including code auto-completion, text generation, and factual question answering (Kwiatkowski et al., 2019; Brown et al., 2020; Kasai et al., 2023, *inter alia*). When responding to a query, language models need to synthesize the prior knowledge they learned during pretraining with the new information provided in the input context (Kwiatkowski et al., 2019; Joshi et al., 2017; Berant et al., 2013; Kasai et al., 2023). For example, in the knowledge conflict setting proposed by Longpre et al. (2021) given the query *What's the capital of Ireland?* and context *The capital of Ireland is Rome.*, the model must decide between whether to agree with its prior knowledge (*Dublin*) or the context (*Rome*).

How easily large language models are affected by contexts is a well-studied problem (Liang et al., 2023; Yoran et al., 2024; Wang et al., 2024; Wu et al., 2024a,b, *inter alia*). Many studies, including Longpre et al. (2021); Chen et al. (2022); Xie et al. (2024), measure how easily context influences a model by computing the **memorization ratio**: the proportion of examples for which the model maintains from before the context was introduced. However, memorization rate may not fully capture the strength of this influence. For example, adding a context could take a model's probability of answering a token from 95% to 51%, but the 1-best answer would remain the same, and the context's influence would not be detectable by memorization rate. To solve this issue, Du et al. (2024) provide a metric that measures the influence of context on a model's answer to a query using a more fine-grained metric based on the model's full answer distribution, the **susceptibility**, which we describe in detail in §4.3. To compute susceptibility, Du et al. (2024) proposes a Monte Carlo approximation. However, it requires running one forward pass, which is computationally expensive, per sampled answer; this limits the scale of analysis. Thus, we aim to find a method that is more efficient, general, and interpretable.

3 Measuring Susceptibility

Let Σ be an **alphabet**, i.e., a finite, non-empty set. A **language model** p_M over an alphabet Σ is a distribution over Σ^*, the set of all strings with **tokens** drawn from Σ. We denote a query by $q \in \Sigma^*$, and the answer to the query by $a \in \Sigma^*$. When querying a language model, a context $c \in \Sigma^*$ is provided together with the query q in one input sequence $c \oplus q$ where $\oplus$ denotes string concatenation. In practice, querying a language model is the process of generating an answer a from the distribution $p_\mathrm{M}(\cdot \mid c \oplus q)$.

3.1 Susceptibility as Mutual Information

For a pretrained language model, Du et al. (2024) gives a metric that quantifies how easily a language model's distribution given a query is altered by a context. To investigate how the answer distributions of a language model respond to a class of contexts, they consider three Σ^*-valued random variables, C, Q and A, standing for context, query, and answer. For a specific query q, C and A are jointly distributed according to the distribution

$$p(C = c, A = a \mid Q = q) \overset{\triangle}{\propto} p_\mathrm{M}(a \mid c \oplus q) \quad (1)$$

Based on the joint distribution in Eq. (1), Du et al. (2024) defines the **susceptibility** of a query q as the conditional mutual information between C and A:

$$\chi(q) \triangleq \mathrm{I}(C; A \mid Q = q) \qquad (2)$$
$$= \mathbb{E}_{C}[D_{\mathrm{KL}}(p(A \mid C = c, Q = q) \parallel p(A \mid Q = q))]$$

See Du et al. (§3.2, 2024) for additional details.

Intuitively, the susceptibility measures how much the answer distribution differs before and after being prompted with additional contexts *on average*. By taking the expectation over all possible contexts, we arrive at the susceptibility. If the model is not susceptible, the Kullback–Leibler divergence $D_{\mathrm{KL}}(p(A \mid C, Q = q) \parallel p(A \mid Q = q))$ should be 0 in expectation, meaning that $p(A \mid C, q)$ and $p(A \mid Q = q)$ are the same, regardless of the value of C.

3.2 Computational Cost

Du et al. (2024) gives a practical algorithm to estimate susceptibility based on Monte Carlo estimation. We call susceptibility estimated in this manner **Monte Carlo susceptibility**. However, computing Monte Carlo susceptibility requires sampling a large number of input–output pairs from a language model. Indeed, we first sample contexts from a distribution $p(C)$, and then evaluate the conditional distribution over answer $p(A \mid C = c, Q = q)$ for all answers a in the answer space, which requires one forward pass per sample. Because the answer space is typically all of Σ^*, $p(A \mid C = c, Q = q)$ is additionally approximated by considering the next-token distribution (Du et al., 2024). Thus, the runtime of such a scheme, with the next-token approximation, is still $\mathcal{O}(n_c \times |\Sigma|)$ where n_c is the number of context samples; for reference, Du et al. (2024) takes 600 Monte Carlo samples.

4 Fisher Susceptibility

To alleviate the cost of the Monte Carlo approximation discussed in §3.2, we derive an efficient approximation based on Fisher information.

4.1 A Simple Reparameterization

First, we give a simple reparameterization that applies to any neural p_{M}. Given a query q, each context $c \in \Sigma^*$ defines a probability distribution over Σ^*, viewed as answers to the query q. Now, let $\boldsymbol{\theta} \colon \Sigma^* \to \mathbb{R}^D$ be an injective embedding function that maps strings to unique real vectors. We can view $\boldsymbol{\theta}$ as an *index* and, thus, consider the language

model as a parameterized family of distributions with a real-valued index:

$$p_{\mathrm{M}}(A = a \mid C = c, Q = q) \triangleq f_q(a; \boldsymbol{\theta}(c \oplus q)), \quad (3)$$

On this view, the conditional language model is parameterized by a real-vector rather than by a string that encodes the context. In our experiments, we define the embedding function to map its argument $c \oplus q$ to be the concatenation of the real-valued embedding vectors of q and the c given by the pretrained language model.

4.2 The Fisher Information Matrix

The reparameterization given in §4.1 opens up a new type of approximation. Specifically, we can now define the following **Fisher information**

$$\mathcal{J}(\boldsymbol{\theta}(q)) \triangleq \qquad (4)$$
$$\mathbb{E}_{a \sim f_q(\cdot; \boldsymbol{\theta}(q))} \left[\frac{\partial \log f_q(a; \boldsymbol{\theta}(q))}{\partial \boldsymbol{\theta}} \frac{\partial \log f_q(a; \boldsymbol{\theta}(q))}{\partial \boldsymbol{\theta}^\top} \right]$$

as a measure of how much influence a context has on the distribution over answers. The Fisher information matrix can be interpreted as a quantification of the amount of information that an observable random variable carries about an unknown parameter (Lehmann and Casella, 1998): The higher the Fisher information is, the easier we can estimate the unknown parameters of the distribution from samples. In our context, we are interested in how language models react to the change in the *context* rather than the pretrained parameters of the model. Thus, we treat the parameters of the language model itself as constants and investigate the Fisher information of f_q with respect to the embedding vector $\boldsymbol{\theta}(q)$ of query q.

More relevant to susceptibility, there is a formal relationship between the Fisher information matrix and the KL divergence. Performing a second-order Taylor expansion of the KL divergence, we arrive at

$$D_{\mathrm{KL}}(f_q(\cdot \mid \boldsymbol{\theta}(q) + \boldsymbol{\delta}) \parallel f_q(\cdot \mid \boldsymbol{\theta}(q)))$$
$$= \frac{1}{2} \boldsymbol{\delta}^\top \mathcal{J}(\boldsymbol{\theta}(q)) \boldsymbol{\delta} + \mathcal{O}\left(||\boldsymbol{\delta}||^3\right), \quad (5)$$

where $\boldsymbol{\delta} \in \mathbb{R}^D$ is a perturbation of the distribution parameter. When the perturbation is small, i.e., when $||\boldsymbol{\delta}||^3$ is small, we expect Eq. (5) to be dominated by its first term. This view invites a simple approximation of the KL divergence.
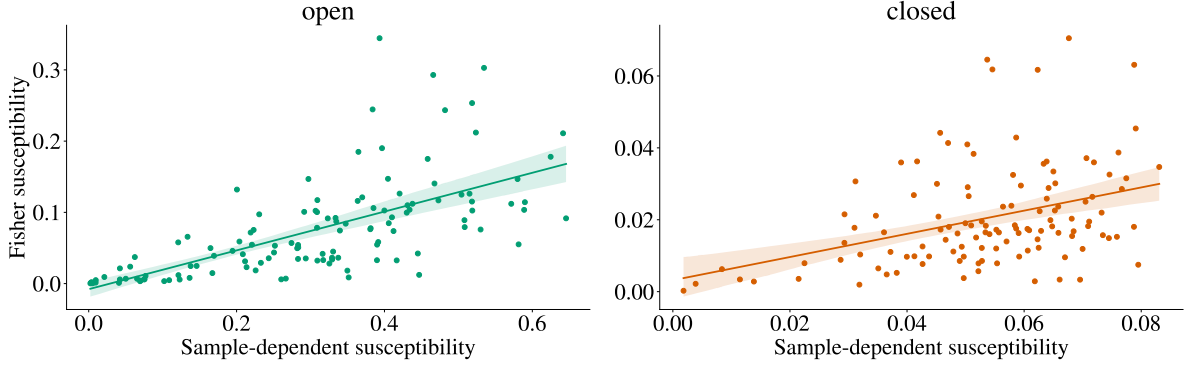
Figure 1: This plot shows the average susceptibility on 122 relation domains (e.g., *alumniOf*) in the YAGO dataset. Each point represents the average score of queries on a relation domain. The x-coordinate represents Monte Carlo susceptibility and y-coordinate represents Fisher susceptibility. The scores are computed with `LLaMA-3-8B-Instruct`. For both query types, the two metrics are strongly correlated.

4.3 Fisher and Mutual Information

In Eq. (2), the susceptibility $\chi(q)$ is defined as the mutual information between the context and answer random variables, conditioned on a fixed query about an entity. Using the identity given in Eq. (5), the susceptibility can be rewritten as

$$\chi(q) \triangleq \mathrm{I}(C; A \mid Q = q) \tag{6a}$$

$$= \frac{1}{2} \mathbb{E}_{C}\left[\boldsymbol{\delta}(c, q)^{\top} \mathcal{J}(\boldsymbol{\theta}(q)) \boldsymbol{\delta}(c, q)\right] \tag{6b}$$

$$+ \mathcal{O}\left(\mathbb{E}_{C}\left[||\boldsymbol{\delta}(c, q)||^{3}\right]\right)$$

$$\approx \frac{1}{2} \mathbb{E}_{C}\left[\boldsymbol{\delta}(c, q)^{\top} \mathcal{J}(\boldsymbol{\theta}(q)) \boldsymbol{\delta}(c, q)\right], \tag{6c}$$

where we define the context-specific perturbation

$$\boldsymbol{\delta}(c, q) \triangleq \boldsymbol{\theta}(c \oplus q) - \boldsymbol{\theta}(q). \tag{7}$$

The full derivation is given in App. A.

To derive an approximation to susceptibility that does not require sampling, we need to remove the terms in Eq. (6) that depend on the random variable C. If we assume that

$$\mathbb{E}_{C}[\boldsymbol{\delta}(c, q)] = \mathbf{m} \tag{8a}$$

$$\mathbb{V}_{C}[\boldsymbol{\delta}(c, q)] = \mathbf{S}, \tag{8b}$$

we arrive at the following closed-form solution

$$\mathbb{E}_{C}\left[\boldsymbol{\delta}(c, q)^{\top} \mathcal{J}(\boldsymbol{\theta}(q)) \boldsymbol{\delta}(c, q)\right]$$
$$= \mathrm{Tr}(\mathbf{S}\mathcal{J}(\boldsymbol{\theta}(q))) + \mathbf{m}^{\top} \mathcal{J}(\boldsymbol{\theta}(q))\mathbf{m} \tag{9}$$

by means of a well-known identity (Petersen and Pedersen, 2008). If we take, for instance, $\mathbf{m} = \mathbf{0}$ and $\mathbf{S} = \mathbf{I}$, Eq. (9) simplifies to

$$\chi^{*}(q) \triangleq \mathrm{Tr}(\mathcal{J}(\boldsymbol{\theta}(q))), \tag{10}$$

which we term **Fisher susceptibility**.

To the extent that Eq. (10) is a good approximation of Eq. (6), Fisher susceptibility $\chi^{*}(q)$ should strongly correlate with Du et al.'s (2024) Monte Carlo approximation to $\chi(q)$. We remark again that Fisher susceptibility $\chi^{*}(q)$ does *not* require computation of the distribution over contexts because, in Eqs. (8a) and (8b), we made an assumption about the mean and variance of the perturbation vector $\boldsymbol{\delta}$. However, this additional assumption is not theoretically motivated, and, thus, we appeal to experimentation to vet the approximation.

4.4 Efficient Computation

Recall that the Fisher information $\mathcal{J}(\boldsymbol{\theta}(q))$ is a matrix of shape $\mathbb{R}^{D \times D}$, where D is the dimension of input embedding $\boldsymbol{\theta}(q)$. Directly computing Fisher information matrix $\mathcal{J}(\boldsymbol{\theta}(q))$ using Eq. (4) takes $\mathcal{O}(D^{2})$; it has $D^{2} \approx 10^{12}$ entries that need to be computed. Thus, we apply the following approximation (Du et al., 2024) by truncating the distribution over answers:

$$\mathcal{J}(\boldsymbol{\theta}(q)) = \sum_{k=1}^{K} f_{q}(a_{k}; \boldsymbol{\theta}(q)) \tag{11}$$

$$\frac{\partial \log f_{q}(a_{k}; \boldsymbol{\theta}(q))}{\partial \boldsymbol{\theta}} \frac{\partial \log f_{q}(a_{k}; \boldsymbol{\theta}(q))}{\partial \boldsymbol{\theta}^{\top}},$$

where $\{a_{k}\}_{k=1}^{K}$ are the top-K highest-probability tokens. This approximation reduces the number of entries to be computed from $\mathcal{O}(D^{2})$ to $\mathcal{O}(KD)$. The individual gradients $\frac{\partial \log f_{q}(a_{k}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ can be efficiently computed with auto-differentiation in the same time complexity that it takes to compute a *single* forward pass of the language model.

Model	Open		Closed		Overall	
	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman
gpt2	0.55	0.58	0.21	0.18	0.63	0.56
gpt2-large	0.27	0.37	0.24	0.32	0.43	0.64
gpt2-medium	0.42	0.47	0.29	0.28	0.57	0.68
gpt2-xl	0.38	0.50	0.31	0.36	0.54	0.71
LLaMA-3-8B	0.46	0.66	0.31	0.35	0.47	0.65
LLaMA-3-8B-Instruct	0.51	0.76	0.38	0.47	0.53	0.68
LLaMA-2-7B	0.46	0.71	0.25	0.27	0.52	0.69
LLaMA-2-7B-chat	0.55	0.78	0.40	0.47	0.47	0.65
Pythia-70m-deduped	0.66	0.68	0.32	0.39	0.64	0.57
Pythia-160m-deduped	0.44	0.53	0.26	0.32	0.51	0.63
Pythia-410m-deduped	0.59	0.64	0.36	0.35	0.70	0.77
Pythia-1.4b-deduped	0.45	0.55	0.20	0.19	0.52	0.69
Pythia-2.8b-deduped	0.37	0.43	0.25	0.29	0.55	0.69
Pythia-6.9b-deduped	0.34	0.42	0.26	0.28	0.56	0.72

Table 1: Correlations between Monte Carlo susceptibility and Fisher susceptibility across different models. For each model, we compute Monte Carlo susceptibility and Fisher susceptibility. Then, we evaluate the Pearson's and Spearman's correlation between them on open, closed, and all queries from YAGO.

5 Experiments

Experimentally, we first aim to show that Fisher susceptibility correlates well with Monte Carlo susceptibility. Then, we investigate what factors influence higher susceptibility, across models, queries, and entities, accelerated by the use of Fisher susceptibility. Finally, we apply Fisher susceptibility to evaluate language models' susceptibility to contexts.

5.1 Experiment Setup

For all of our experiments, we use the same framework and dataset provided by Du et al. (2024).[2] For each of the relations from the YAGO knowledge graph (Suchanek et al., 2007), we have two closed query forms (yes–no questions) and two open query forms. For each relation, we subsample 100 entities, half of which are real entities extracted from YAGO, and half of which are fake entity names generated by GPT-4, from their dataset of 1000 entities. In total, we construct a collection of 48,800 queries. For each of these queries, we compute Monte Carlo susceptibility as a point of comparison. We take a sample of 256 contexts, 8 of which directly mention the queried entity, per query for Pythia and GPT-2 models, and 128 contexts, 4 of which directly mention the queried entity, per query for LLaMA models.

We also compute Fisher susceptibility for each

of these entities according to Eq. (10). We repeat these for models of different families (i.e., Pythia (Biderman et al., 2023), GPT-2 (Brown et al., 2020), and LLaMA (Touvron et al., 2023)), model sizes,[3] and training types (e.g., pretrained vs instruction-tuned) when applicable. A full list of models can be found in Tab. 1

Dataset. Following Du et al. (2024), we use 122 relations from the YAGO knowledge graph (Suchanek et al., 2007), such as *birthPlace*, *leader*, and *homeLocation*. For each relation, we sample 50 real entities from YAGO and 50 fake entities generated by GPT-4 (OpenAI, 2023).[4] Our open and closed queries are constructed from templates of both question-answering and sentence-completion forms, e.g., (closed, question-answering) *Q: Is {answer} the capital of {entity}? A:*, (open, question-answering) *Q: What is the capital of {entity}? A:*, and (open, sentence-completion) *The capital of {entity} is.* We parameterize the templates with entities (both real and fake entities) and answers in their respective slots in the templates. For instantiating contexts, we use the base template from Du et al. (2024), e.g., *The capital of {entity} is {answer}.*. We parameterize these context templates with both real and fake entities (and answers to the queries, when applicable). For each relation domain, we randomly sample

---

[3]We choose 70m, 410m, 1.4b, 2.8b, 6.9b for Pythia-deduped, and small, medium, large, xl for GPT-2.
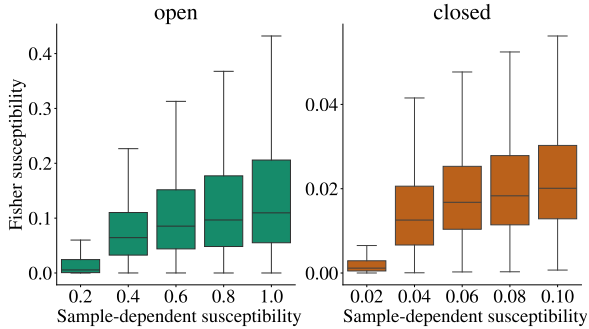[4]gpt-4-1106-preview, January 2024.

Figure 2: For both *open* and *closed* queries, Fisher susceptibility y strongly correlates with Monte Carlo susceptibility (x-axis). Monte Carlo susceptibility are divided into 5 bins from 0 to 1 on `LLaMA-3-8B-Instruct`.

2000 contexts. For the Monte Carlo estimate of susceptibility, we further subsample 256 contexts for `Pythia` and `GPT-2` and 128 contexts for `LLaMA`.

Models. In our experiments, we use the pretrained language models (`Pythia`, `GPT-2`, `LLaMA`) from Huggingface.[5]

Computational Resources. All experiments presented were conducted on two NVIDIA GeForce RTX 4090 GPUs with 24GB memory. `LLaMA` models are stored and run in bfloat16 precision. The other models are stored and run in float32 precision.

5.2 Comparing Susceptibility

We now demonstrate that Fisher susceptibility is a good approximation to susceptibility.

Experiment Setup. We use the setup from §5.1 to compute Fisher susceptibility. For each query, we compute the Pearson's correlation and Spearman's correlation between Monte Carlo susceptibility and Fisher susceptibility.

Results. We compare Monte Carlo susceptibility and Fisher susceptibility. In Fig. 2 and Tab. 1, we observe strong correlation between Fisher susceptibility and Monte Carlo susceptibility across all models. Across queries, we find a Pearson's correlation of $r = 0.51$ and a Spearman's correlation of $\rho = 0.76$ on open queries, $r = 0.38$ and $\rho = 0.47$ on closed queries using

---

[5]We use Pythia (`https://huggingface.co/colle ctions/EleutherAI/pythia-scaling-suite-64fb5df a8c21ebb3b7ad2e1`), GPT-2 (`https://huggingface.co /openai-community/gpt2`), LLaMA-2 (`https://huggin gface.co/collections/meta-llama/llama-2-famil y-661da1f90a9d678b6f55773b`), and LLaMA-3 (`https: //huggingface.co/collections/meta-llama/meta-lla ma-3-66214712577ca38149ebb2b6`).

Model	Query Type		
	Overall	Closed	Open
`Pythia-70m-deduped`	0.30	0.03	0.54
`Pythia-160m-deduped`	0.31	0.03	0.57
`Pythia-410m-deduped`	0.31	0.04	0.56
`Pythia-1.4b-deduped`	0.35	0.06	0.62
`Pythia-2.8b-deduped`	0.33	0.03	0.58
`Pythia-6.9b-deduped`	0.34	0.04	0.61
`gpt2-small`	0.34	0.08	0.57
`gpt2-medium`	0.31	0.05	0.55
`gpt2-large`	0.32	0.04	0.57
`gpt2-xl`	0.29	0.04	0.51
`LLaMA-2-7B`	0.34	0.07	0.59
`LLaMA-2-7B-chat`	0.57	0.23	0.61
`LLaMA-3-8B`	0.37	0.13	0.58
`LLaMA-3-8B-instruct`	0.39	0.10	0.66

Table 2: Mean Monte Carlo susceptibility of different models on YAGO on different query types (open, closed, and overall).

`LLaMA-3-8B-instruct`. We also evaluate average susceptibility on the corpus level by averaging the susceptibility of queries in each domain in Fig. 1. We measure a Pearson's correlation of $r = 0.65$ and a Spearman's correlation of $\rho = 0.76$ on open queries and $r = 0.51, \rho = 0.60$ on closed queries using `LLaMA-3-8B-instruct`. We take these results as validation that Fisher susceptibility correlates with Monte Carlo susceptibility and is, thus, a good approximation. Moreover, the large Pearson's correlation coefficient indicates the relationship is linear. We refer the readers to App. B for full evaluation results on all sizes of models.

Runtime Comparison. We conduct an empirical analysis on the runtime of Fisher susceptibility and Monte Carlo susceptibility. We find computing Fisher susceptibility is $70\times$ faster when the number of samples for Monte Carlo susceptibility is chosen to be 256 and $30\times$ faster when the number of samples is 128. Specifically, for `LLaMA-3-8B` models, evaluating Monte Carlo susceptibility for all 48800 queries on YAGO costs 10 hours, while computing Fisher susceptibility only costs 20 minutes.

5.3 Factors Affecting Fisher susceptibility

We now aim to understand what factors could cause a language model to have higher susceptibility. We investigate three aspects, namely the size and training method of language models, the format of the query, and the type of the entity.

6620

Sample-dependent susceptibility vs entity type
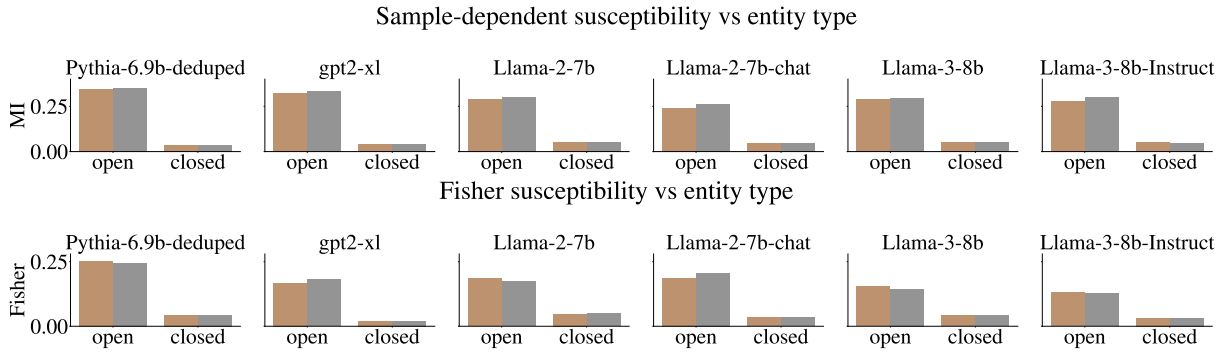

Fisher susceptibility vs entity type

Figure 3: Susceptibility comparisons between open and closed queries for *real entities* (bars on the left) and *fake entities* (bars on the right). In each subplot, the two bars on the left represent open queries, and the two on the right represent closed queries. From this, we can see the susceptibility generally does not appear to differ much between real and fake entities. (Top) Monte Carlo susceptibility. (Bottom) Fisher susceptibility.

Sample-dependent susceptibility vs query type

Fisher susceptibility vs query type
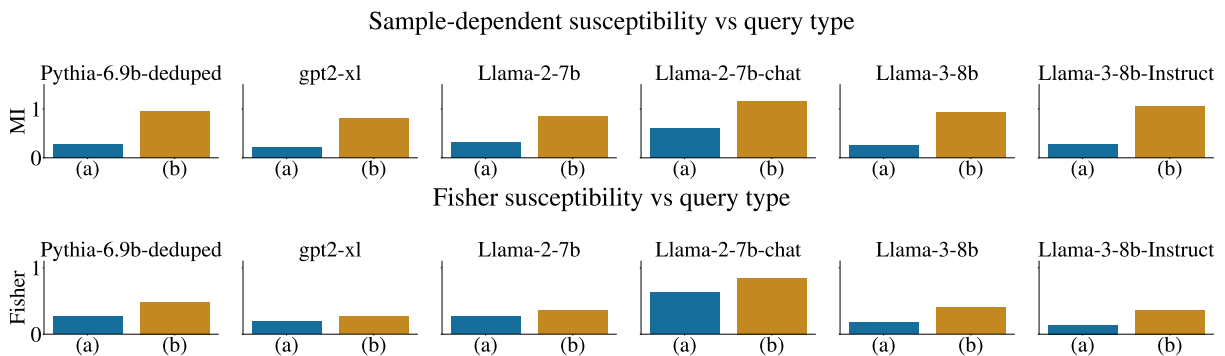
Figure 4: Susceptibility comparisons between (a) question-answering and (b) sentence-completion queries. Consistently, question-answering formats are less susceptible for both susceptibility metrics. (Top) Monte Carlo susceptibility. (Bottom) Fisher susceptibility.

Models. From Tab. 2, we see that the susceptibility of the models of `Pythia` and `GPT-2` families do *not* decrease as the number of trainable parameters in the model increases. This finding indicates that susceptibility can not be offset by the amount of prior knowledge stored in the language model about a query. We also see that `LLaMA` models do not have lower susceptibility compared with the other two model families, despite their better performance on many downstream tasks (Touvron et al., 2023). By comparing instruction-tuned models and base models (i.e., `LLaMA-2-7B` vs. `LLaMA-2-7B-chat`, `LLaMA-3-8B` vs. `LLaMA-3-8B-Instruct`), we find instruction tuning increases the susceptibility of language models. These comparisons is in alignment with the empirical observation that instruction-tuned models are better at integrating the information in the input context and responding to queries.

Queries. We also investigate whether language models are more susceptible for some particular types of queries. Similar to the findings of Du

et al. (2024), Tab. 2 shows that closed queries (e.g., *Q: Is Here Comes the Sun performed by The Beatles? A:*) are less susceptible than open queries (e.g., *Q: Who performed Here Comes the Sun? A:*). In addition, we investigate two forms of open queries: question-answering (e.g., *Q: Is Here Comes the Sun performed by The Beatles? A:*) and sentence-completion (e.g., *Here Comes the Sun is performed by*). On `LLaMA-3-8B-Instruct`, we found that queries in question-answering form have an average Monte Carlo susceptibility of 0.27 and Fisher susceptibility of 0.04, while the sentence-completion form has 1.05 and 0.14. Both of the susceptibility metrics show that question-answering queries are less easily affected by context than sentence-completion queries. This finding supports our claim on the comparability of Fisher susceptibility to Monte Carlo susceptibility. The full results are given in Fig. 4.

Entity Familiarity. Du et al. (2024) found, across different models, that real entities tend to be less susceptible than fake ones. However, we find

this pattern does not hold for Fisher susceptibility. We show the susceptibility comparison for 6 models in Fig. 3. The Monte Carlo susceptibility of real entities are slightly but consistently lower than that of fake entities. Meanwhile, the Fisher susceptibility of all models remains similar regardless of whether the entity is real or not. This could suggest a limitation of Fisher susceptibility as an approximation for susceptibility, which could be due to the assumptions made in Eq. (8a).

6 Conclusion

To efficiently measure a language model's susceptibility, we have proposed Fisher susceptibility, which uses the Fisher information of a language model with regard to its input to measure the scale of distributional changes as the input varies. Through experiments, we find strong correlation between a language model's Monte Carlo susceptibility and Fisher susceptibility, which we take to validate our approximation. Compared to methods which requires many context samples and language model forward passes, our method is significantly faster. Our study contributes to the exploration for interpretable and efficient evaluation metrics for language models.

Limitations

One technical limitation of this work is that we compute an approximation of the Fisher information $\mathcal{J}(\boldsymbol{\theta}(q))$ using Eq. (11) by taking the top K answers. Second, computing Fisher susceptibility requires automatic differentiation on the language model, which is more memory intensive (by a factor of 2) than simply performing a forward pass.

Ethics Statement

This paper provides a novel language evaluation metric and experimental analysis on publicly available models and data. Our ultimate goal in this paper is to contribute to the research on language model interpretability and evaluation. By investigating the robustness of language models to contexts, we aim to contribute to a research effort of developing more reliable, interpretable models. We foresee no particular ethical concerns and hope this paper contributes to developing tools that can identify and mitigate ethical concerns in the future.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2307, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kevin Du, Vésteinn Snæbjarnarson, Niklas Stoehr, Jennifer White, Aaron Schein, and Ryan Cotterell. 2024. Context versus prior knowledge in language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13211–13235, Bangkok, Thailand. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and

Kentaro Inui. 2023. RealTime QA: What's the answer right now? In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Erich L. Lehmann and George Casella. 1998. *Theory of Point Estimation*, second edition. Springer-Verlag, New York, NY, USA.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *ArXiv*, abs/2303.08774.

K. B. Petersen and M. S. Pedersen. 2008. The matrix cookbook. Version 20081110.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA. Association for Computing Machinery.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Bin Wang, Chengwei Wei, Zhengyuan Liu, Geyu Lin, and Nancy F. Chen. 2024. Resilience of large language models for noisy instructions.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024a. How easily do irrelevant inputs skew the responses of large language models? In *First Conference on Language Modeling*.

Zhenyu Wu, Chao Shen, and Meng Jiang. 2024b. Instructing large language models to identify and ignore irrelevant conditions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6799–6819, Mexico City, Mexico. Association for Computational Linguistics.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

A Derivation of Eq. (6)

Proposition A.1.

$$\chi(q) \triangleq \mathrm{I}(C; A \mid Q = q) = \frac{1}{2} \underset{C}{\mathbb{E}}\Big[\boldsymbol{\delta}(c, q)^{\top} \mathcal{J}(\boldsymbol{\theta}(q))\boldsymbol{\delta}(c, q)\Big] + \mathcal{O}\left(\|\boldsymbol{\delta}(c, q)\|^3\right) \tag{12}$$

where $\boldsymbol{\delta}(c, q) \triangleq \boldsymbol{\theta}(c \oplus q) - \boldsymbol{\theta}(q)$.

Proof. Consider the following manipulation

$$\chi(q) \triangleq \mathrm{I}(C; A \mid Q = q) \tag{13a}$$

$$= \underset{C}{\mathbb{E}}[D_{\mathrm{KL}}(p(A \mid C = c, Q = q) \parallel p(A \mid Q = q))] \tag{13b}$$

$$= \underset{C}{\mathbb{E}}[D_{\mathrm{KL}}(f_q(\cdot \mid \boldsymbol{\theta}(q) + (-\boldsymbol{\theta}(q) + \boldsymbol{\theta}(c \oplus q)) \parallel f_q(\cdot \mid \boldsymbol{\theta}(q))))] \tag{13c}$$

$$= \frac{1}{2}\underset{C}{\mathbb{E}}\Big[(\boldsymbol{\theta}(c) - \boldsymbol{\theta}(q))^{\top}\mathcal{J}(\boldsymbol{\theta}(q))(\boldsymbol{\theta}(c \oplus q) - \boldsymbol{\theta}(q)) + \mathcal{O}\left(\|\boldsymbol{\theta}(c \oplus q) - \boldsymbol{\theta}(q)\|^3\right)\Big] \tag{13d}$$

$$= \frac{1}{2}\underset{C}{\mathbb{E}}\Big[(\boldsymbol{\theta}(c) - \boldsymbol{\theta}(q))^{\top}\mathcal{J}(\boldsymbol{\theta}(q))(\boldsymbol{\theta}(c \oplus q) - \boldsymbol{\theta}(q))\Big] + \underset{C}{\mathbb{E}}\Big[\mathcal{O}\left(\|\boldsymbol{\theta}(c \oplus q) - \boldsymbol{\theta}(q)\|^3\right)\Big] \tag{13e}$$

$$= \frac{1}{2}\underset{C}{\mathbb{E}}\Big[(\boldsymbol{\theta}(c) - \boldsymbol{\theta}(q))^{\top}\mathcal{J}(\boldsymbol{\theta}(q))(\boldsymbol{\theta}(c \oplus q) - \boldsymbol{\theta}(q))\Big] + \mathcal{O}\Bigg(\underbrace{\underset{C}{\mathbb{E}}\big[\|\boldsymbol{\theta}(c \oplus q) - \boldsymbol{\theta}(q)\|^3\big]}_{\text{expected approximation error}}\Bigg) \tag{13f}$$

$$= \frac{1}{2}\underset{C}{\mathbb{E}}\Big[\boldsymbol{\delta}(c, q)^{\top}\mathcal{J}(\boldsymbol{\theta}(q))\boldsymbol{\delta}(c, q)\Big] + \mathcal{O}\Bigg(\underbrace{\underset{C}{\mathbb{E}}\big[\|\boldsymbol{\delta}(c, q)\|^3\big]}_{\text{expected approximation error}}\Bigg), \tag{13g}$$

which proves the result. ∎

B Additional Experimental Results

We plot the comparison between Fisher susceptibility and Monte Carlo susceptibility in Fig. 5 and Fig. 6. Across all models, Fisher susceptibility exhibits strong correlation with Monte Carlo susceptibility, which we take to mean that Fisher susceptibility is a good approximation to Monte Carlo susceptibility.
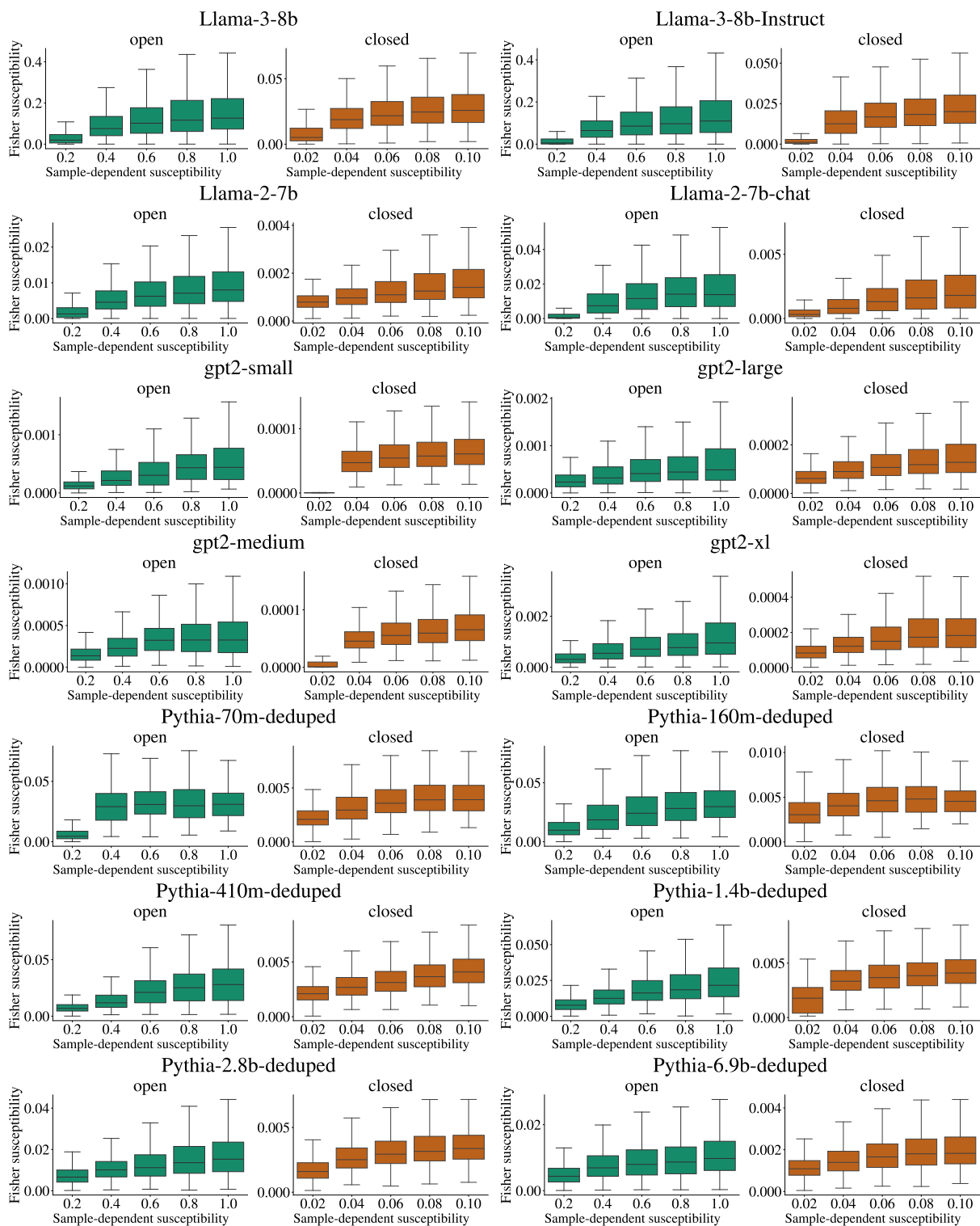
Figure 5: Fisher susceptibility plotted against Monte Carlo susceptibility for different models.
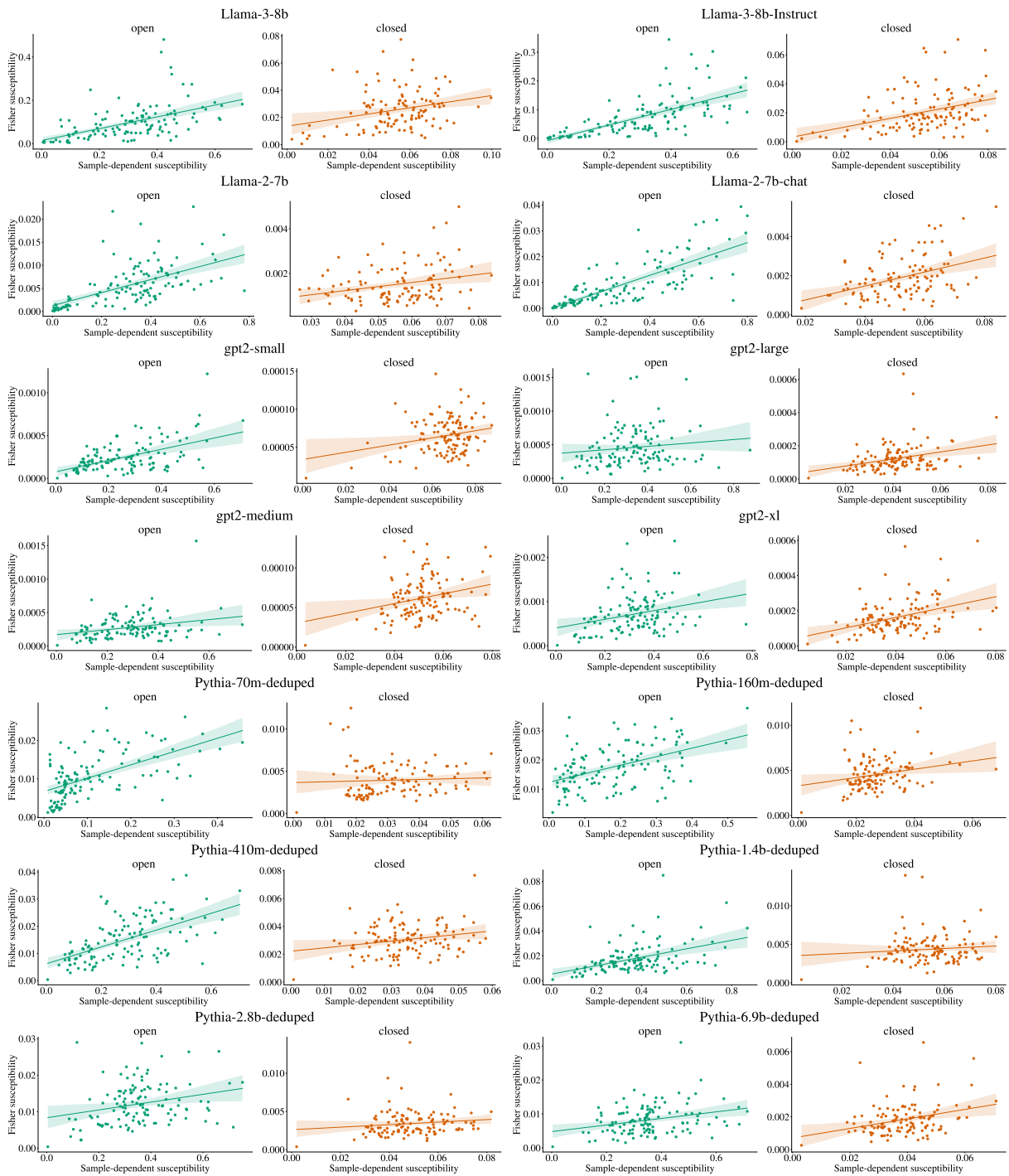
Figure 6: This plot shows the average susceptibility on 122 relation domains in the YAGO dataset for all models. Each point represents the average score of queries on a relation domain. The x-coordinate represents Monte Carlo susceptibility and y-coordinate represents Fisher susceptibility.