CausalGym: Benchmarking causal interpretability methods on linguistic tasks

Aryaman Arora **Dan Jurafsky** **Christopher Potts**
Stanford University
{aryamana,jurafsky,cgpotts}@stanford.edu

Abstract

Language models (LMs) have proven to be powerful tools for psycholinguistic research, but most prior work has focused on purely behavioural measures (e.g., surprisal comparisons). At the same time, research in model interpretability has begun to illuminate the abstract causal mechanisms shaping LM behavior. To help bring these strands of research closer together, we introduce **CausalGym**. We adapt and expand the SyntaxGym suite of tasks to benchmark the ability of interpretability methods to causally affect model behaviour. To illustrate how **CausalGym** can be used, we study the `pythia` models (14M–6.9B) and assess the causal efficacy of a wide range of interpretability methods, including linear probing and distributed alignment search (DAS). We find that DAS outperforms the other methods, and so we use it to study the learning trajectory of two difficult linguistic phenomena in `pythia-1b`: negative polarity item licensing and filler–gap dependencies. Our analysis shows that the mechanism implementing both of these tasks is learned in discrete stages, not gradually.

https://github.com/aryamanarora/causalgym

1 Introduction

Language models have found increasing use as tools for psycholinguistic investigation—to model word surprisal (Smith and Levy, 2013; Goodkind and Bicknell, 2018; Wilcox et al., 2023a; Shain et al., 2024, *inter alia*), graded grammaticality judgements (Hu et al., 2024), and, broadly, human language processing (Futrell et al., 2019; Warstadt and Bowman, 2022; Wilcox et al., 2023b). To benchmark the linguistic competence of LMs, computational psycholinguists have created **targeted syntactic evaluation** benchmarks, which feature minimally-different pairs of sentences differing in grammaticality; success is measured by whether

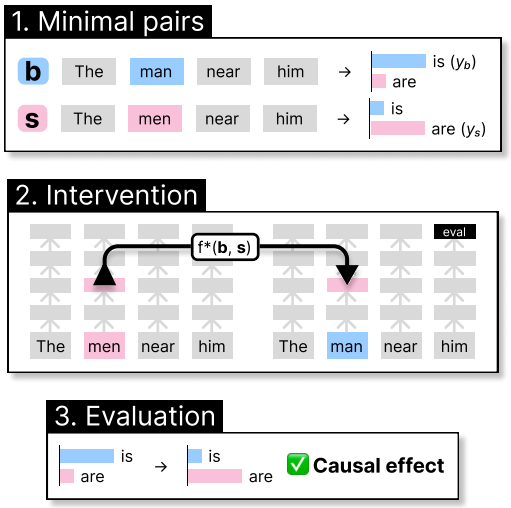


Figure 1: The **CausalGym** pipeline: **(1)** take an input minimal pair $(\mathbf{b}, \mathbf{s})$ exhibiting a linguistic alternation that affects next-token predictions (y_b, y_s); **(2)** intervene on the base forward pass using a pre-defined intervention function that operates on aligned representations from both inputs; **(3)** check how this intervention affected the next-token prediction probabilities. In aggregate, such interventions assess the causal role of the intervened representation on the model's behaviour.

LMs assign higher probability to the grammatical sentence in each pair (Marvin and Linzen, 2018). Despite the increasing use of LMs as models of human linguistic competence and how much easier it is to experiment on them than human brains, we do not understand the mechanisms underlying model behaviour—LMs remain largely uninterpretable.

The **linear representation hypothesis** claims that 'concepts' form linear subspaces in the representations of neural models. An increasing body of experimental evidence from models trained on language and other tasks supports this idea (Mikolov et al., 2013; Elhage et al., 2022; Park et al., 2023; Nanda et al., 2023). Per this hypothesis, information about high-level linguistic alternations can be localised to linear subspaces of LM activations.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14638–14663

Methods for finding such features, and even modifying activations in feature subspaces to causally influence model behaviour, have proliferated, including probing (Ettinger et al., 2016; Adi et al., 2017), distributed alignment search (DAS; Geiger et al., 2023b), and difference-in-means (Marks and Tegmark, 2023).

Psycholinguistics and interpretability have complementary needs: thus far, psycholinguists have evaluated LMs on extensive benchmarks but neglected understanding their internal mechanisms, whereas interpretability methods have only been evaluated on one-off datasets and sorely need more diverse benchmarks. Thus, we introduce **`CausalGym`** (Figure 1). We adapt linguistic tasks from SyntaxGym (Gauthier et al., 2020) to benchmark interpretability methods on their ability to find linear features in LMs that, when subject to intervention, causally influence linguistic behaviours. We study the `pythia` family of models (Biderman et al., 2023), finding that DAS is the most efficacious method. However, our investigation corroborates earlier findings that DAS is powerful enough to make the model produce arbitrary input–output mappings (Wu et al., 2023). To address this, we adapt the notion of control tasks from the probing literature (Hewitt and Liang, 2019), finding that adjusting for performance on the arbitrary mapping task reduces the gap between DAS and other methods.

We further investigate how LMs learn two difficult linguistic behaviours during training: filler–gap extraction and negative polarity item licensing. We find that the causal mechanisms require multi-step movement of information, and that they emerge in discrete stages (not gradually) early in training.

2 Related work

Targeted syntactic evaluation. Benchmarks adhering to this paradigm include SyntaxGym (Gauthier et al., 2020; Hu et al., 2020), BLiMP (Warstadt et al., 2020), and several earlier works (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018; Futrell et al., 2019). We use the SyntaxGym evaluation sets over BLiMP even though the latter has many more examples, because we require minimal pairs that are grammatical sentences alternating along a specific feature (e.g. number). Such pairs can be constructed templatically using SyntaxGym's format.

Interventional interpretability. Interventions are the workhorse of causal inference (Pearl, 2009), and have thus been adopted by recent work in interpretability for establishing the causal role of neural network components in implementing certain behaviours (Vig et al., 2020; Geiger et al., 2021, 2022, 2023a; Meng et al., 2022; Chan et al., 2022; Goldowsky-Dill et al., 2023), particularly linguistic ones like coreference and gender bias (Lasri et al., 2022; Wang et al., 2023; Hanna et al., 2023; Chintam et al., 2023; Yamakoshi et al., 2023; Hao and Linzen, 2023; Chen et al., 2023; Amini et al., 2023; Guerner et al., 2023). The approach loosely falls under the nascent field of *mechanistic interpretability*, which seeks to find interpretable mechanisms inside neural networks (Olah, 2022).

We illustrate the interventional paradigm in Figure 1; given a base input $\mathbf{b}$ and source input $\mathbf{s}$, all interventional approaches take a model-internal component f and replace its output with that of $f^*(\mathbf{b}, \mathbf{s})$, which modifies the representation of $\mathbf{b}$ using that of $\mathbf{s}$. The core idea of intervention is adopted directly from the do-operator used in causal inference; we test the intervention's effect on model output to establish a causal relationship.

3 Benchmark

To create **`CausalGym`**, we converted the core test suites in SyntaxGym (Gauthier et al., 2020) into templates for generating large numbers of span-aligned minimal pairs, a process we describe below along with our evaluation setup.

3.1 Premise

Each test suite in SyntaxGym focuses on a single linguistic feature, constructing English-language minimal pairs that minimally adjust that feature to change expectations about how a sentence should continue. A test suite contains several *items* which share identical settings for irrelevant features, and each item has some *conditions* which vary only the important feature. All items adhere to the same templatic structure, sharing the same ordering and set of *regions* (syntactic units). To measure whether models match human expectations, SyntaxGym evaluates the model's surprisal at specific regions between differing conditions.

For example, the *Subject-Verb Number Agreement (with prepositional phrase)* task constructs items consisting of 4 conditions, which set all possible combinations of the number feature on subjects

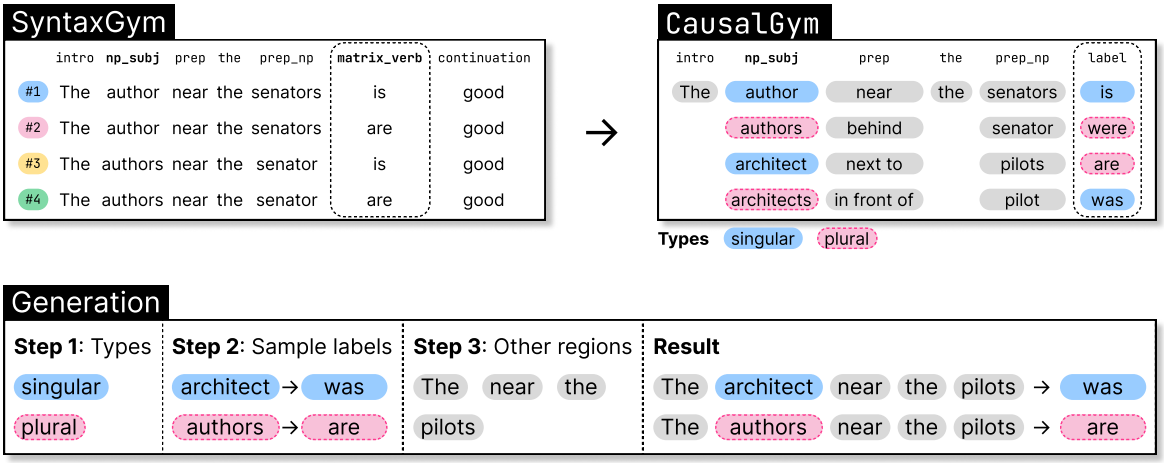


Figure 2: An example of the **CausalGym** conversion process on the test suite *Subject-Verb Number Agreement (with prepositional phrase)*. The left side shows how items are structured in SyntaxGym originally, which we process into the templatic format on the right. The bottom shows how we sample a minimal pair.

and their associated verbs, as well as the *opposite* feature on a distractor noun. Each example in this test suite follows the template

(1) The `np_subj` `prep` the `prep_np` `matrix_verb` `continuation`.

where, in a single item, the regions `np_subj` and `matrix_verb` are modified along the number feature, and `prep_np` is a distractor. For example:

(2) The **author** near the senators **is** good.

(3) *The **author** near the senators **are** good.

(4) *The **authors** near the senator **is** good.

(5) The **authors** near the senator **are** good.

Humans expect agreement between the number feature on the verb and the subject, as in (2) and (5). On this test suite, SyntaxGym measures if the surprisal at the verb satisfies the following inequalities between conditions: $p(\text{is} \mid \text{author}) > p(\text{are} \mid \text{author})$ and $p(\text{are} \mid \text{authors}) > p(\text{is} \mid \text{authors})$.

3.2 Templatising SyntaxGym

Our goal is to study how LMs implement mechanisms for converting feature alternations in the input into corresponding alternations in the output—e.g., how does an LM keep track of the number feature on the subject when it needs to output an agreeing verb? In adapting SyntaxGym for this purpose, we must address two issues: (1) to study model mechanisms, we only want *grammatical* pairs of sentences; and (2) SyntaxGym test suites contain < 50 items, while we need many more for training supervised interpretability methods and creating non-overlapping test sets.

Thus, we select the two grammatical conditions from each item and simplify the behaviour of interest into an explicit input–output mapping. For example, we recast *Subject-Verb Number Agreement (with prepositional phrase)* into counterfactual pairs that elicit singular or plural verbs based on the number feature of the subject, and hold everything else (including the distractor) constant:

(6) a. The **author** near the senators ⇒ **is**

b. The **authors** near the senators ⇒ **are**

To be able to generate many examples for training, we use the aligned regions as slots in a template that we can mix-and-match between items to combinatorially generate pairs, illustrated in Figure 2. We manually removed options (potential choices to fill slots) that would have resulted in questionably grammatical sentences.

For generation using our format, each template has a set of types T which govern the input **label variable** and the expected next-token prediction **label**. To generate a counterfactual pair, we first sample two types $t_1, t_2 \sim T$ such that $t_1 \neq t_2$. Then, for the label variable and label, we sample an option of that type t_1 (for the first sentence) or t_2 (for the second). Finally, for the non-label variable regions, we sample one option and set both sentences to that. In Figure 2, we show the generation process in the bottom panel; types for the label variable and label options are colour-coded.

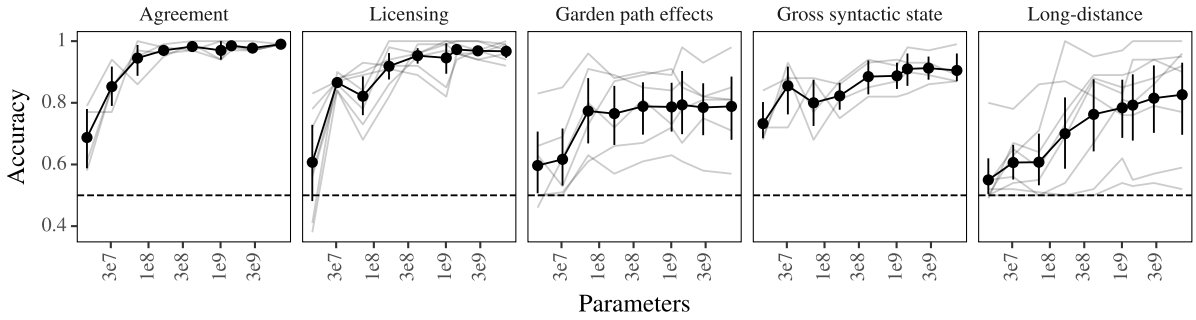


Figure 3: Accuracy of `pythia`-family models on the **`CausalGym`** tasks, grouped by type, with scale. The dashed line is random-chance accuracy (50%).

3.3 Tasks

`CausalGym` contains 29 tasks, of which one is novel (`agr_gender`) and 28 were templatised from SyntaxGym. Of the 33 test suites in the original release of SyntaxGym, we only used tasks from which we could generate paired grammatical sentences (leading us to discard the 2 center embedding tasks), and merged the 6 gendered reflexive licensing tasks into 3 non-gendered ones.

We report task accuracy vs. model scale in Figure 3 to give a sense of how well the models perform the linguistic behaviours we study. Examples of pairs generated for each task are provided in appendix A.

3.4 Evaluation

An evaluation sample consists of a base input $\mathbf{b}$, source input $\mathbf{s}$, ground-truth base label y_b, and ground-truth source label y_s. For example, the components of (6) are

(7) $\underbrace{\text{The author near the senators}}_{\mathbf{b}} \Rightarrow \underbrace{\text{is}}_{y_b}$

(8) $\underbrace{\text{The authors near the senators}}_{\mathbf{s}} \Rightarrow \underbrace{\text{are}}_{y_s}$

A successful intervention will take the original LM p running on input $\mathbf{b}$ and make it predict y_s as the next token. We measure the strength of an intervention by its **log odds-ratio**.

First, we select a component f, which can be any part of a neural network that outputs a representation, inside the model p. When the model is run on input $\mathbf{b}$, this component produces a representation we denote $f(\mathbf{b})$. We perform an intervention which replaces the output of f with an output of f^* as in §2. To produce a representation, f^* may modify the base representation with reference to the source representation, and so its output is $f^*(\mathbf{b}, \mathbf{s})$. The intervention results in an intervened language model which we denote informally as $p_{f \leftarrow f^*}$. In the framework of causal abstraction (Geiger et al., 2021), if this intervention successfully makes the model behave as if its input was $\mathbf{s}$, then the representation at f is causally aligned with the high-level linguistic feature alternating in $\mathbf{b}$ and $\mathbf{s}$.

We now operationalise a measure of causal effect. Taking the original model p, the intervened model $p_{f \leftarrow f^*}$, and the evaluation sample, we define the log odds-ratio as:

$$\begin{aligned}&\mathsf{Odds}(p, p_{f \leftarrow f^*}, \langle \mathbf{b}, \mathbf{s}, y_b, y_s \rangle) \\ &\quad = \log \left(\frac{p(y_b \mid \mathbf{b})}{p(y_s \mid \mathbf{b})} \cdot \frac{p_{f \leftarrow f^*}(y_s \mid \mathbf{b}, \mathbf{s})}{p_{f \leftarrow f^*}(y_b \mid \mathbf{b}, \mathbf{s})} \right) \quad (9)\end{aligned}$$

where a greater log odds-ratio indicates a larger causal effect at that intervention site, and a log odds-ratio of 0 indicates no causal effect. Given an evaluation set E, the average log odds-ratio is

$$\begin{aligned}&\mathsf{AvgOdds}(p, p_{f \leftarrow f^*}, E) = \\ &\qquad \frac{1}{|E|} \sum_{e \in E} \mathsf{Odds}(p, p_{f \leftarrow f^*}, e) \quad (10)\end{aligned}$$

4 Methods

We briefly describe our choice of f^* and the feature-finding methods that we benchmark in this paper.

4.1 Preliminaries

In this paper, we only benchmark interventions along a single feature direction, i.e. one-dimensional distributed interchange intervention (1D DII; Geiger et al., 2023b). DII is an interchange intervention that operates on a non-basis-aligned subspace of the activation space. Formally,

given a feature vector $\mathbf{a} \in \mathbb{R}^n$ and f, 1D DII defines f^* as

$$f^*_{\mathbf{a}}(\mathbf{b}, \mathbf{s}) = f(\mathbf{b}) + (f(\mathbf{s})\mathbf{a}^\top - f(\mathbf{b})\mathbf{a}^\top)\mathbf{a} \quad (11)$$

As noted above, when our intervention replaces f with $f^*_{\mathbf{a}}$, we denote the new model as $p_{f \leftarrow f^*_{\mathbf{a}}}$.

We fix f to operate on token-level representations; since $\mathbf{b}$ and $\mathbf{s}$ may have different lengths due to tokenisation; we align representations at the last token of each template region.

In principle, we allow future work to consider other forms of f^*, but 1D DII has two useful properties. Given the linear representation hypothesis and that **`CausalGym`** exclusively studies binary linguistic features, 1D DII ought to be sufficiently expressive for controlling model behaviour. Furthermore, probes trained on binary classification tasks operate on a one-dimensional subspace of the representation, and thus we can directly use the weight vector of a probe as the parameter $\mathbf{a}$ in eq. (11)—Tigges et al. (2023) used a similar setup to causally evaluate probes.

We study seven methods, of which four are supervised: distributed alignment search (DAS), linear probing, difference-in-means, and LDA. The other three are unsupervised: PCA, k-means, and (as a baseline) sampling a random vector. All of these methods provide us a feature direction $\mathbf{a}$ that we use as a constant in eq. (11). For probing and unsupervised methods, we use implementations from `scikit-learn` (Pedregosa et al., 2011). To train distributed alignment search and run 1D DII, we use the `pyvene` library (Wu et al., 2024). Further training details are in appendix B. We formally describe each method below.

4.2 Definitions

DAS. Given a training set T, we learn the intervention direction, potentially distributed across many neurons, that maximises the output probability of the counterfactual label. Formally, we first randomly initialise $\mathbf{a}_{\text{das}}$ and intervene on the model p with it to get $p_{f \leftarrow f^*_{\mathbf{a}_{\text{das}}}}$. We freeze the model weights and optimise $\mathbf{a}_{\text{das}}$ such that we minimise the cross-entropy loss with the target output y_s:

$$\min_{\mathbf{a}_{\text{das}}} \left\{ -\sum_{\langle \mathbf{b}, \mathbf{s}, y_b, y_s \rangle \in T} \log p_{f \leftarrow f^*_{\mathbf{a}_{\text{das}}}}(y_s \mid \mathbf{b}, \mathbf{s}) \right\} \quad (12)$$

The learned DAS parameters $\mathbf{a}_{\text{das}}$ then define a function $f^*_{\mathbf{a}_{\text{das}}}$ using (11).

Linear probe. Linear probing classifiers have been the dominant feature-finding method for neural representations of language (Belinkov, 2022). A probe outputs a distribution over classes given a representation $\mathbf{x} \in \mathbb{R}^n$:

$$q_{\boldsymbol{\theta}}(y \mid \mathbf{x}) = \text{softmax}(\mathbf{a}_{\text{probe}} \cdot f(\mathbf{x}) + b) \quad (13)$$

We learn the parameters $\boldsymbol{\theta}$ of the probe over the base training set examples (so, maximising $q_{\boldsymbol{\theta}}(y_b \mid \mathbf{b})$) using the SAGA solver (Defazio et al., 2014) as implemented in `scikit-learn`, and the parameters $\mathbf{a}_{\text{probe}}$ define the intervention function $f^*_{\mathbf{a}_{\text{probe}}}$.

Diff-in-means. The difference in per-class mean activations has been surprisingly effective for controlling representations (Marks and Tegmark, 2023; Li et al., 2023) and erasing linear features (Belrose et al., 2023; Belrose, 2023). To implement this approach, we take the base input–output pairs $\langle \mathbf{b}, y_b \rangle$ from the training set T, where $y_b \in \{y_1, y_2\}$, and group them by the identity of their labels. Thus, we have $X_1 = \{\mathbf{b} \in T : y_b = y_1\}$ and $X_2 = \{\mathbf{b} \in T : y_b = y_2\}$. The diff-in-means method is then defined as follows:

$$\mathbf{a}_{\text{mean}} = \frac{1}{|X_1|} \sum_{\mathbf{x} \in X_1} f(\mathbf{x}) - \frac{1}{|X_2|} \sum_{\mathbf{x} \in X_2} f(\mathbf{x}) \quad (14)$$

$$= \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \quad (15)$$

and as usual $\mathbf{a}_{\text{mean}}$ defines the function $f^*_{\mathbf{a}_{\text{mean}}}$.

Linear disciminant analysis. LDA assumes that each class is distributed according to a Gaussian and all classes share the same covariance matrix $\boldsymbol{\Sigma}$. Given the per-class means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$,

$$\mathbf{a}_{\text{lda}} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \quad (16)$$

Principal component analysis (PCA). We intervene along the first principal component, which is a vector $\mathbf{a}_{\text{pca}}$ that maximises the variance in mean-centered activations (denoted $\widetilde{f}(\mathbf{x})$).

$$\max_{\mathbf{a}_{\text{pca}}} \left\{ \sum_{\mathbf{x} \in X_1 \cup X_2} \left(\widetilde{f}(\mathbf{x}) \cdot \mathbf{a}_{\text{pca}} \right)^2 \right\} \quad (17)$$

PCA was previously used to debias gendered word embeddings by Bolukbasi et al. (2016).

k-means. We use 2-means and learn a clustering of activations into two sets S_1, S_2 that minimises the variance of the activations relative to their class centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$. Our feature direction is

$$\mathbf{a}_{\text{kmeans}} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \quad (18)$$

Mod.	Acc.	Overall odds-ratio (↑)							Selectivity (↑)						
		DAS	Probe	Mean	PCA	k-m.	LDA	Rand.	DAS	Probe	Mean	PCA	k-m.	LDA	Rand.
14m	0.62	**3.94**	1.16	1.04	0.48	0.50	0.11	0.03	**1.84**	1.38	1.24	0.54	0.55	0.15	0.08
31m	0.74	**5.82**	2.22	1.80	0.83	0.85	0.08	0.02	**2.75**	2.63	2.03	0.86	0.88	0.13	0.03
70m	0.77	**7.60**	2.70	2.12	1.16	1.20	0.11	0.03	**2.87**	2.86	2.15	1.05	1.09	0.16	0.05
160m	0.82	**7.93**	3.13	2.23	1.26	1.29	0.12	0.02	2.93	**3.27**	2.34	1.24	1.26	0.15	0.04
410m	0.86	**10.24**	3.69	3.22	2.15	2.19	0.34	0.05	3.96	**4.20**	3.33	2.07	2.12	0.43	0.06
1b	0.86	**10.74**	3.66	3.17	2.07	2.13	0.29	0.03	3.34	**4.24**	3.09	1.78	1.85	0.36	0.04
1.4b	0.88	**9.58**	3.48	3.06	1.96	2.02	0.37	0.02	2.99	**4.08**	3.21	1.87	1.94	0.46	0.03
2.8b	0.88	**8.88**	3.72	3.19	1.93	2.00	0.31	0.01	2.57	**4.15**	3.31	1.69	1.75	0.39	0.01
6.9b	0.89	**9.95**	3.42	2.91	1.81	1.87	0.27	0.01	2.48	**3.79**	2.85	1.50	1.54	0.34	0.02

Table 1: Overall odds-ratio (§5.1) and selectivity (§5.2) of each feature-finding method averaged over all tasks in **CausalGym**. We also report average task accuracy, which increases with scale. For models larger than `pythia-70m`, we report the better of two probes trained with different hyperparameters (appendix C).

5 Experiments

We perform all experiments on the `pythia` model series (Biderman et al., 2023), which includes 10 models ranging from 14 million to 12 billion parameters, all trained on the same data in the same order. This model series allows us to study how feature representations change with scale and training data size in a controlled manner—all models were trained on the same data in the same order, and checkpoints are provided.

5.1 Measuring causal efficacy

The Transformer (Vaswani et al., 2017) is organised around the **residual stream** (Elhage et al., 2021), which each attention and MLP layer reads from and additively writes to. The residual stream is an information bottleneck; information from the input must be present at some token in every layer's residual stream in order to reach the next layer and ultimately affect the output.

Therefore, given a feature present in the input and influencing the model output, we should be able to find a causally-efficacious subspace encoding that feature in at least one token position in every layer. If the feature is binary (such as the ones we study in **CausalGym**) and processed by a single mechanism in the model, then 1D DII should be sufficient for this. Conversely, if 1D DII using a given method fails to produce causal effect, either the method is poor, or the feature is processed by multiple mechanisms in the model or not represented linearly at all.

Thus, for each task in **CausalGym**, we take the function of interest f to be the state of the residual stream after the operation of a Transformer layer $l \in L$ at the last token of a particular region $r \in R$. For notational convenience, we denote this function as $f^{(l,r)}$. We learn 1D DII using each method m for every such function. We use a trainset T of 400 examples for each benchmark task, and evaluate on a non-overlapping set E of 100 examples.[1] Each such experiment results in an intervened model that we denote $p_{f^{(l,r)} \leftarrow f^*_{\mathbf{a}_m}}$. To compute the overall log odds-ratio for a feature-finding method on a particular model on a single task, we take the maximum of the average odds-ratio (§3.4) over regions at a specific layer, and then average over all layers:

$$\mathsf{OverallOdds}(p, m, E) \tag{19}$$
$$= \frac{1}{|L|} \sum_{l \in L} \left(\max_{r \in R} \left(\mathsf{AvgOdds}(p, p_{f^{(l,r)} \leftarrow f^*_{\mathbf{a}_m}}, E) \right) \right)$$

This metric rewards a method for finding a highly causally-efficacious region in every layer.

5.2 Controlling for expressivity

DAS is the the only method with a causal training objective. Other methods do not optimise for, or even have access to, downstream model behaviour. Wu et al. (2023) found that a variant of DAS achieves substantial causal effect even on a randomly-initialised model or with irrelavant next-token labels, both settings where no causal mechanism should exist. How much of the causal effect found by DAS is due to its expressivity? Research on probing has faced a similar concern: to what extent is a probe's accuracy due to its expressivity rather than any aspect of the representation being studied? Hewitt and Liang (2019) propose comparing to accuracy on a **control task** that requires memorising an input-to-label mapping.

We adapt this notion to **CausalGym**, introducing control tasks where the next-token labels y_b, y_s are mapped to the arbitrary tokens '_dog' and '_give'

[1] Further training details are given in appendix B, and we report hyperparameter tuning experiments on a dev set in appendix C.

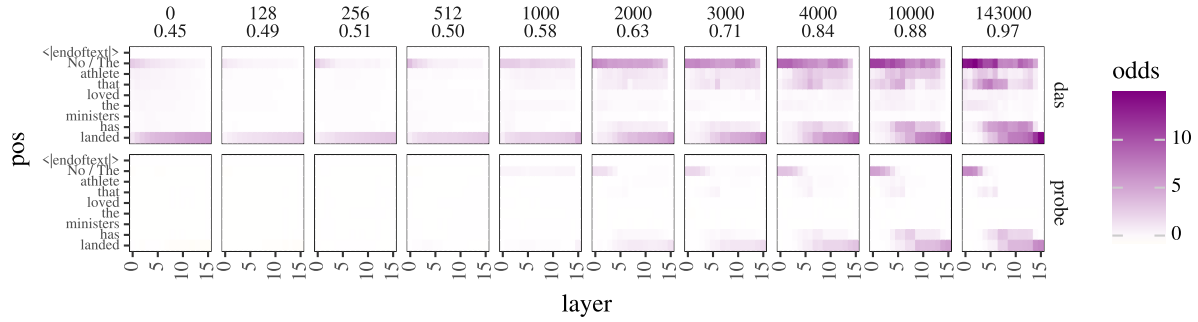
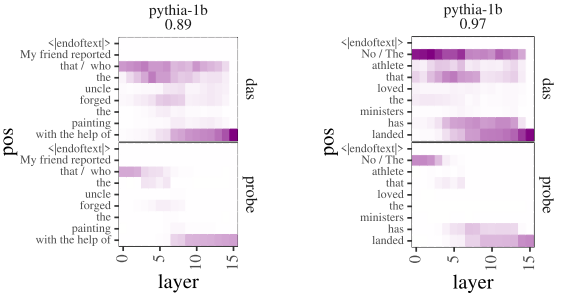


Figure 4: Odds-ratio for checkpoints of `pythia-1b` on the task `npi_any_subj_relc`, plotted at every layer and template region. The y-axis is labelled with an example pair of sentences. The plot titles are labelled with the checkpoint and task accuracy. Darker regions indicate a token in a specific layer where causal effect was high.



(a) `filler_gap_subj` (b) `npi_any_subj_relc`

Figure 5: Odds-ratio for each layer and region using DAS and probing on `pythia-1b`, on two tasks.

while preserving the class partitioning.[2] For example, on the gender-agreement task `agr_gender`, we replace the label '_he' with '_dog' and '_she' with '_give'. We define selectivity for each method by taking the difference between odds-ratios on the original task and the control task for each f, and then compute the overall odds-ratio as in eq. (19).

5.3 Results

We summarise the results for each method in Table 1 by reporting overall odds-ratio and selectivity averaged over all tasks for each model. For a breakdown, see appendices E.1 and E.2.

We find that DAS consistently finds the most causally-efficacious features. The second-best method is probing, followed by difference-in-means. The unsupervised methods PCA and k-means are considerably worse. Despite supervision, LDA barely outperforms random features.

[2]The input-to-label mapping in **`CausalGym`** tasks is dependent on the input token types, so we cannot exactly replicate Hewitt and Liang. The setup we instead use is from Wu et al. (2023).

However, DAS is not considerably more selective or (at larger scales) even less selective than probing or diff-in-means; it can perform well on arbitrary input–output mappings. This suggests that its access to the model outputs during training is responsible for much of its advantage.

6 Case studies

In this section, we use **`CausalGym`** to study how LMs learn negative polarity item (NPI) licensing and wh-extraction from prepositional phrases over the course of training using checkpoints of `pythia-1b`. We first describe the tasks.

`npi_any_subj-relc.` NPIs are lexemes that can only occur in negative-polarity sentential contexts. In this task, we specifically check whether the NPI *any* is correctly licensed by a negated subject, giving minimal pairs like

(20) **No** athlete that loved the ministers has landed ⇒ **any**

(21) **The** athlete that loved the ministers has landed ⇒ **some**

In (21), where there is no negation at the sentence level, it would be ungrammatical to continue the sentence with the NPI *any*.

`filler_gap_subj.` Filler–gap dependencies in English occur when interrogatives are extracted out of and placed in front of a clause. The position from which they are extracted must remain empty. The task `filler_gap_subj` requires an LM to apply this rule when extracting from a distant prepositional phrase, e.g.

(22) My friend reported **that** the uncle forged the painting with the help of ⇒ **him**

(23) My friend reported **who** the uncle forged the painting with the help of ⇒ **.**

In (23), it would be ungrammatical for the preposition to have an explicit object since *who* was extracted from that position, leaving behind a gap.

Final mechanisms. We use the experimental setup of §5.1 and plot the average odds-ratio for each region and layer on the final checkpoint of `pythia-1b` in Figure 5. For both tasks, we find that the input feature crosses over several different positions before arriving at the output position. For example, in the NPI mechanism (Figure 5b), the negation feature is moved to the complementiser *that* in the early layer, into the auxiliary verb at middle layers, and the main verb in later layers, where its presence is used to predict the NPI *any*. The filler–gap mechanism is similarly complex.

6.1 Training dynamics

To study how the mechanisms emerge over the course of training, we run the exact same experiments on earlier checkpoints of `pythia-1b`.

`npi_any_subj-relc`. In Figure 4, the effect first emerges at the NPI (all but last layer) and the main verb (step 1000), then abruptly the auxiliary becomes important at middle layers and the NPI effect is pushed down to early layers (step 2000), and finally another intermediate locations is added at *that* (step 3000). The effect is also distributed across multiple regions in the intermediate layers.

`filler_gap_subj`. This behaviour takes longer to learn than NPI licensing (Figure 6). The mechanism emerges in two stages: at step 2000, it includes the filler position (*that* / *who*), the first determiner *the*, and the final token. After step 10K, the main verb is added to the mechanism.

Discussion. For both tasks, the model intially learns to move information directly from the alternating token to the output position. Later in training, intermediate steps are added in the middle layers. DAS finds a greater causal effect across the board, but both methods largely agree on which regions are the most causally efficacious at each layer. Notably, DAS finds causal effect at all timesteps, even when the model has just been initialised; this corroborates Wu et al.'s (2023) findings.

7 Conclusion

We introduced **`CausalGym`**, a multi-task benchmark of linguistic behaviours for measuring the causal efficacy of interpretability methods. We showed the impressive performance of distributed alignment search, but also adapted a notion of control tasks to causal evaluation to enable fairer comparison of methods. Finally, we studied how causal effect propagates in training on two linguistic tasks: NPI licensing and filler–gap dependency tracking.

In recent years, much effort has been devoted towards developing causally-grounded methods for understanding neural networks. A probe achieving high classification accuracy provides no guarantee that the model actually distinguishes those classes in downstream computations; evaluating probe directions for causal effect is an intuitive test for whether they reflect features that the model uses downstream. Overall, while methods may come and go, we believe the causal evaluation paradigm will continue to be useful for the field.

A major motivation for releasing **`CausalGym`** is to encourage computational psycholinguists to move beyond studying the input–output behaviours of LMs. Our case studies in §6 are a basic example of the analysis that new methods permit. Ultimately, understanding how LMs learn linguistic behaviours may offer insights into fundamental properties of language (cf. Kallini et al., 2024; Wilcox et al., 2023b).

We hope that **`CausalGym`** will encourage comprehensive evaluation of new interpretability methods and spur adoption of the interventional paradigm in computational psycholinguistics.

Limitations

While **`CausalGym`** includes a range of linguistic tasks, there are many non-linguistic behaviours on which we may want to use interpretability methods, and so we encourage future research on a greater variety of tasks. In addition, **`CausalGym`** includes only English data, and comparable experiments with other languages might yield substantially different results, thereby providing us with a much fuller picture of the causal mechanisms that LMs learn to use. Furthermore, results may differ on other models, since models in the `pythia` series were trained on the same data in a fixed order; different training data may result in different mechanisms. Finally, justified by the nature of our tasks, we only benchmark methods that operate on one-dimensional linear subspaces; multi-dimensional linear methods as well as non-linear ones await being benchmarked.

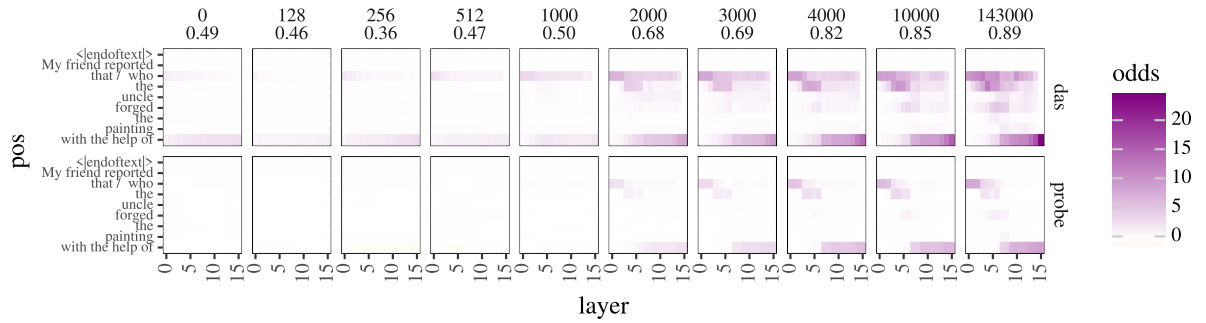


Figure 6: Odds-ratio for checkpoints of `pythia-1b` on the task `filler_gap_subj`, plotted at every layer and template region.

Ethics statement

Interpretability is a rapidly-advancing field, and our benchmark results render us optimistic about our ability to someday understand the mechanisms inside complex neural networks. However, successful interpretability methods could be used to justify deployment of language models in high-risk settings (e.g. to autonomously make decisions about human beings) or even manipulate models to produce harmful outputs. Understanding a model does not mean that it is safe to use in every situation, and we caution model deployers and users against uncritical trust in models even if they are found to be interpretable.

Acknowledgements

We thank Atticus Geiger, Jing Huang, Harshit Joshi, Jordan Juravsky, Julie Kallini, Chenglei Si, Tristan Thrush, and Zhengxuan Wu for helpful discussion about the project and their comments on the manuscript.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France.

Afra Amini, Tiago Pimentel, Clara Meister, and Ryan Cotterell. 2023. Naturalistic causal probing for morpho-syntax. *Transactions of the Association for Computational Linguistics*, 11:384–403.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Nora Belrose. 2023. Diff-in-means concept editing is worst-case optimal. *EleutherAI Blog*.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. LEACE: Perfect linear concept erasure in closed form. *arXiv:2306.03819*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430, Honolulu, Hawaii, USA. PMLR.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. 2022. Causal scrubbing: A method for rigorously testing interpretability hypotheses. In *Alignment Forum*.

Angelica Chen, Ravid Schwartz-Ziv, Kyunghyun Cho, Matthew L. Leavitt, and Naomi Saphra. 2023. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. *arXiv:2309.07311*.

Abhijith Chintam, Rahel Beloch, Willem Zuidema, Michael Hanna, and Oskar van der Wal. 2023. Identifying and adapting transformer-components responsible for gender bias in an English language model. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 379–394, Singapore. Association for Computational Linguistics.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.

Atticus Geiger, Chris Potts, and Thomas Icard. 2023a. Causal abstraction for faithful model interpretation. *arxiv:2301.04709*.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah D. Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning, ICML 2022*, volume 162 of *Proceedings of Machine Learning Research*, pages 7324–7338, Baltimore, Maryland, USA. PMLR.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2023b. Finding alignments between interpretable causal variables and distributed neural representations. *arXiv:2303.02536*.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *arXiv:2304.05969*.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics.

Clément Guerner, Anej Svete, Tianyu Liu, Alexander Warstadt, and Ryan Cotterell. 2023. A geometric notion of causal probing. *arXiv:2307.15054*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Hanna, Yonatan Belinkov, and Sandro Pezzelle. 2023. When language models fall in love: Animacy processing in transformer language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12120–12135, Singapore. Association for Computational Linguistics.

Sophie Hao and Tal Linzen. 2023. Verb conjugation in transformers is determined by linear encodings of subject number. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4531–4539, Singapore. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Jennifer Hu, Kyle Mahowald, Gary Lupyan, Anna Ivanova, and Roger Levy. 2024. Language models align with human judgments on key grammatical constructions. *arXiv:2402.01676*.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. *arXiv:2401.06416*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA.

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv:2310.06824*.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Chris Olah. 2022. Mechanistic interpretability, variables, and the importance of interpretable bases. *Transformer Circuits Thread*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv:2311.03658*.

Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*, 2nd edition. Cambridge University Press.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

Cory Shain, Clara Meister, Tiago Pimentel, Ryan Cotterell, and Roger Levy. 2024. Large-scale evidence for logarithmic effects of word predictability on reading time. *Proceedings of the National Academy of Sciences*. To appear.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *arXiv:2310.15154*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, Kigali, Rwanda.

Alex Warstadt and Samuel R. Bowman. 2022. What artificial neural networks can tell us about human language acquisition. *Algebraic Structures in Natural Language*, pages 17–60.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R.

Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Ethan Wilcox, Clara Meister, Ryan Cotterell, and Tiago Pimentel. 2023a. Language model quality correlates with psychometric predictive power in multiple languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7503–7511, Singapore. Association for Computational Linguistics.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2023b. Using computational models to test syntactic learnability. *Linguistic Inquiry*, pages 1–44.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Noah D. Goodman, Christopher D. Manning, and Christopher Potts. 2024. pyvene: A library for understanding and improving PyTorch models via interventions. Under review.

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D. Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in Alpaca. In *Advances in Neural Information Processing Systems*, volume 36.

Takateru Yamakoshi, James McClelland, Adele Goldberg, and Robert Hawkins. 2023. Causal interventions expose implicit situation models for commonsense language understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13265–13293, Toronto, Canada. Association for Computational Linguistics.

A Tasks

Task	Example
Agreement (4)	
`agr_gender`	[**John/Jane**] walked because [**he/she**]
`agr_sv_num_subj-relc`	The [**guard/guards**] that hated the manager [**is/are**]
`agr_sv_num_obj-relc`	The [**guard/guards**] that the customers hated [**is/are**]
`agr_sv_num_pp`	The [**guard/guards**] behind the managers [**is/are**]
Licensing (7)	
`agr_refl_num_subj-relc`	The [**farmer/farmers**] that loved the actors embarrassed [**himself/themselves**]
`agr_refl_num_obj-relc`	The [**farmer/farmers**] that the actors loved embarrassed [**himself/themselves**]
`agr_refl_num_pp`	The [**farmer/farmers**] behind the actors embarrassed [**himself/themselves**]
`npi_any_subj-relc`	[**No/The**] consultant that has helped the taxi driver has shown [**any/some**]
`npi_any_obj-relc`	[**No/The**] consultant that the taxi driver has helped has shown [**any/some**]
`npi_ever_subj-relc`	[**No/The**] consultant that has helped the taxi driver has [**ever/never**]
`npi_ever_obj-relc`	[**No/The**] consultant that the taxi driver has helped has [**ever/never**]
Garden path effects (6)	
`garden_mvrr`	The infant [**who was/**∅] brought the sandwich from the kitchen [**by/.**]
`garden_mvrr_mod`	The infant [**who was/**∅] brought the sandwich from the kitchen with a new microwave [**by/.**]
`garden_npz_obj`	While the students dressed [**,/**∅] the comedian [**was/for**]
`garden_npz_obj_mod`	While the students dressed [**,/**∅] the comedian who told bad jokes [**was/for**]
`garden_npz_v-trans`	As the criminal [**slept/shot**] the woman [**was/for**]
`garden_npz_v-trans_mod`	As the criminal [**slept/shot**] the woman who told bad jokes [**was/for**]
Gross syntactic state (4)	
`gss_subord`	[**While the/The**] lawyers lost the plans [**they/.**]
`gss_subord_subj-relc`	[**While the/The**] lawyers who wore white lab jackets studied the book that described several advances in cancer therapy [**,/.**]
`gss_subord_obj-relc`	[**While the/The**] lawyers who the spy had contacted repeatedly studied the book that colleagues had written on cancer therapy [**,/.**]
`gss_subord_pp`	[**While the/The**] lawyers in a long white lab jacket studied the book about several recent advances in cancer therapy [**,/.**]
Long-distance dependencies (8)	
`cleft`	What the young man [**did/ate**] was [**make/for**]
`cleft_mod`	What the young man [**did/ate**] after the ingredients had been bought from the store was [**make/for**]
`filler_gap_embed_3`	I know [**that/what**] the mother said the friend remarked the park attendant reported your friend sent [**him/.**]
`filler_gap_embed_4`	I know [**that/what**] the mother said the friend remarked the park attendant reported the cop thinks your friend sent [**him/.**]
`filler_gap_hierarchy`	The fact that the brother said [**that/who**] the friend trusted [**the/was**]
`filler_gap_obj`	I know [**that/what**] the uncle grabbed [**him/.**]
`filler_gap_pp`	I know [**that/what**] the uncle grabbed food in front of [**him/.**]
`filler_gap_subj`	I know [**that/who**] the uncle grabbed food in front of [**him/.**]

B Training and evaluation details

We load models using the HuggingFace `transformers` (Wolf et al., 2020) library. Up to size `410m` we load weights in `float32` precision, `1b` in `bfloat16` precision, and larger models in `float16` precision. Our training set starts with 200 examples sampled according to the scheme in §3.2. We then double the size of the set (400) by swapping the base and source inputs/labels and adding these to the training set; including both directions of the intervention makes the comparison fairer between DAS and the other non-paired methods, and also ensures a perfect balance between labels.

The evaluation set consists of 50 examples sampled the same way (effectively 100), except we resample in case we encounter a sentence already present in the training set. Thus, there is no overlap with the training set. We evaluate all metrics (odds-ratio and probe classification accuracy) on this set.

We train DAS for one epoch with a batch size of 4, resulting in 100 backpropagation steps. We use the Adam optimiser (Kingma and Ba, 2015) and a linear learning rate schedule, with the first 10% of training being a warmup from 0 to the learning rate, followed by the learning rate linearly decaying to 0 for the rest of training. The scheduling and optimiser is identical to Wu et al. (2023). We use a learning

rate of $5 \cdot 10^{-3}$, which is higher than previous work (usually 10^{-3}) due to the small training set size; see appendix C for hyperparameter tuning experiments which justify this choice.

To run our experiments, we used a cluster of NVIDIA A100 (40 GB) and NVIDIA RTX 6000 Ada Generation GPUs. The total runtime for the benchmarking experiments in §5 was ~ 400 hours, and for the case studies in §6 it was ~ 25 hours.

C Hyperparameter tuning

To ensure fair comparison, we tuned hyperparameters for DAS, probes, and PCA on a dev set, sampled the same way as the eval set (non-overlapping with train set) but with a different random seed. We train on all tasks in **`CausalGym`** and report the average odds-ratio following the same evaluation setup as in §5.1. We studied only the three smallest models (`pythia-14m`, `31m`, `70m`) due to the large number of experiments needed. Specifically, we tune the learning rate for DAS, the type of regularisation and whether or not to include a bias term in the logit for probes,[3] and averaging of the first c components for PCA. We report the overall log odds-ratio for various hyperparameter settings in Table 2. These experiments were run on a NVIDIA RTX 6000 Ada Generation. The total runtime was ~ 25 hours.

For probing (Table 2a), we found that including a bias term and using only L_2 regularisation with the `saga` solver delivers the best performance. However, the setting of the weight coefficient λ on the regularisation term in the loss depends on the model. The main architectural difference between these three models is the hidden dimension size, so we suspect that the optimal choice for λ depends on that. Roughly extrapolating the observed trend, in our main experiments we check $\lambda = \{10^4, 10^5\}$ for `pythia-160m` and `410m`, $\lambda = \{10^5, 10^6\}$ for `pythia-1b`, `1.4b`, and `2.8b`, and $\lambda = \{10^6, 10^7\}$ for `pythia-6.9b`. As for why L_2 regularisation increases causal efficacy, we note that Hewitt and Liang (2019) found that it also increases probe selectivity—we leave this as an open question for future work.

For PCA (Table 2b), we found that averaging the first c components did not improve performance over just using the first component; thus, we used just the first PCA component in our main-text experiments.

For DAS (Table 2c), we found that using the learning rate suggested by Wu et al. (2023), 10^{-3}, understated performance and a higher learning rate did not result in any apparent training instability. However, our experimental setup is quite different (smaller training set, no learned boundary, greater variety of model scales). We did not find any consistent differences or trends with model scale between learning rates of $5 \cdot 10^{-3}$ and 10^{-2}, so we used the former for all experiments.

D Data and licensing

We use the original test suites from SyntaxGym which were described in Hu et al. (2020). These were released under the MIT License, and our data release will also use the MIT license for compatibility.

[3] Cf. Tigges et al. (2023), who did not include a bias term in their causal evaluation of probing.

Model	Probe	λ				
		10^0	10^1	10^2	10^3	10^4
`pythia-14m` ($d = 128$)	No reg., no int.	0.80				
	No reg., int.	0.85				
	L_1, no int.	0.38	0.21		0.00	
	L_1, int.	0.41	0.22		0.00	
	L_2, no int.	1.07	1.15		1.08	
	L_2, int.	1.09	**1.18**	1.15	1.07	1.05
	$L_1 + L_2$, no int.	0.93	0.55		0.08	
	$L_1 + L_2$, int.	0.89	0.57		0.08	
`pythia-31m` ($d = 256$)	No reg., no int.	1.75				
	No reg., int.	1.77				
	L_1, no int.	0.83	0.39		0.12	
	L_1, int.	0.83	0.40		0.11	
	L_2, no int.	1.98	2.14		2.18	
	L_2, int.	2.03	2.22	**2.26**	2.11	1.90
	$L_1 + L_2$, no int.	1.45	0.99		0.14	
	$L_1 + L_2$, int.	1.42	0.93		0.14	
`pythia-70m` ($d = 512$)	No reg., no int.	1.72				
	No reg., int.	1.72				
	L_1, no int.	0.74	0.32		0.31	
	L_1, int.	0.75	0.33		0.17	
	L_2, no int.	1.85	2.05		2.43	
	L_2, int.	1.87	2.08	2.38	**2.70**	2.57
	$L_1 + L_2$, no int.	1.11	0.67		0.32	
	$L_1 + L_2$, int.	1.12	0.71		0.18	

(a) Overall odds-ratio across various hyperparameter settings for probes. 'Int.' means whether the probe logit has a bias term.

Model	c (# components)				
	1	2	3	4	5
`pythia-14m`	**0.48**	0.44	0.34	0.29	0.28
`pythia-31m`	**0.86**	0.82	0.59	0.49	0.43
`pythia-70m`	**1.18**	0.91	0.78	0.75	0.64

(b) Overall odds-ratio across variants of PCA, averaging the first c components.

Model	LR	Step				
		0	25	50	75	99
`pythia-14m`	10^{-3}	0.06	0.37	1.01	1.48	1.63
	$5 \cdot 10^{-3}$	0.04	2.53	3.58	3.82	3.91
	10^{-2}	0.04	3.17	3.72	3.95	**4.02**
`pythia-31m`	10^{-3}	0.04	1.09	2.83	3.64	3.83
	$5 \cdot 10^{-3}$	0.04	5.19	5.78	6.00	**6.04**
	10^{-2}	0.03	5.05	5.44	5.77	5.87
`pythia-70m`	10^{-3}	0.02	2.25	4.69	5.42	5.57
	$5 \cdot 10^{-3}$	0.02	7.21	7.48	7.54	7.55
	10^{-2}	0.03	6.92	7.37	7.66	**7.75**

(c) Overall odds-ratio across various learning rates for DAS.

Table 2: Hyperparameter search results.

E Detailed odds-ratio results

In these comprehensive results, we include an additional method: **vanilla interchange intervention**. Instead of as in eq. (11), vanilla intervention defines f^* as

$$f^*_{\text{vanilla}}(\mathbf{b}, \mathbf{s}) = f(\mathbf{s}) \tag{24}$$

i.e. it entirely replaces the activation with that of the source input. This is equivalent to n-dimensional DII where $f(\mathbf{s}) \in \mathbb{R}^n$, and is a significantly more expressive intervention than any methods we tested. We include it as a non-learned baseline.

E.1 Per-layer

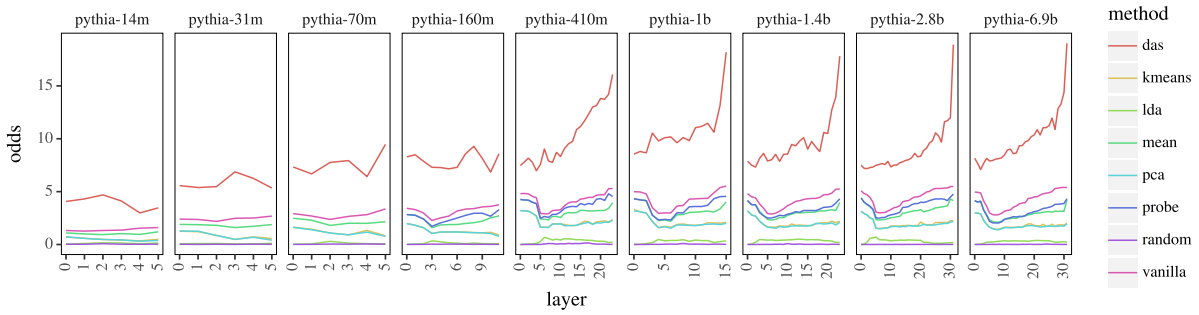


Figure 7: Average odds-ratio per layer and model across all tasks in `CausalGym`.

E.2 Per-task

Rows in gray indicate tasks where the model achieves $< 60\%$ accuracy.

Task	Task Acc.	Feature-finding methods							Vanilla
		DAS	Probe	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.58	0.32	**0.94**	0.49	0.35	0.36	0.03	0.01	0.81
agr_sv_num_subj-relc	0.61	**2.50**	1.74	1.48	0.14	0.08	0.14	0.01	1.91
agr_sv_num_obj-relc	0.79	2.03	2.02	2.05	0.30	0.32	0.06	0.03	**2.13**
agr_sv_num_pp	0.77	**3.47**	3.15	2.85	0.36	0.14	0.15	0.03	3.28
agr_refl_num_subj-relc	0.78	**2.39**	2.18	1.79	0.17	0.13	0.12	0.03	2.34
agr_refl_num_obj-relc	0.72	**1.79**	1.46	1.18	0.12	0.10	0.06	0.03	1.56
agr_refl_num_pp	0.83	**2.56**	2.14	1.57	0.20	0.14	0.13	0.04	2.48
npi_any_subj-relc	0.56	**5.62**	0.64	0.67	0.41	0.41	0.04	0.03	0.68
npi_any_obj-relc	0.57	**5.27**	0.54	0.56	0.37	0.36	0.02	0.03	0.55
npi_ever_subj-relc	0.38	**5.50**	0.10	0.10	0.20	0.19	0.02	0.01	0.10
npi_ever_obj-relc	0.41	**5.07**	0.14	0.14	0.25	0.25	0.01	0.01	0.14
garden_mvrr	0.63	**4.72**	1.62	1.71	0.86	1.49	0.44	0.11	1.72
garden_mvrr_mod	0.50	**3.73**	1.01	1.12	0.99	1.05	0.14	0.00	1.80
garden_npz_obj	0.83	**5.93**	0.56	1.04	1.04	1.04	0.15	0.05	2.07
garden_npz_obj_mod	0.66	**7.55**	0.21	0.20	0.19	0.20	0.23	0.02	1.18
garden_npz_v-trans	0.46	**2.32**	0.49	0.45	0.05	0.06	0.01	0.02	0.20
garden_npz_v-trans_mod	0.50	**0.64**	0.08	0.05	0.02	0.02	0.02	0.02	0.14
gss_subord	0.72	**4.38**	3.53	2.37	1.92	2.01	0.10	0.04	4.38
gss_subord_subj-relc	0.69	**4.70**	0.99	0.93	0.89	0.93	0.10	0.08	1.73
gss_subord_obj-relc	0.68	**5.10**	1.33	1.27	1.25	1.27	0.16	0.07	1.74
gss_subord_pp	0.84	**6.80**	1.07	0.96	0.93	0.96	0.23	0.09	1.99
cleft	0.50	**7.89**	2.30	1.73	0.45	0.52	0.18	0.04	2.43
cleft_mod	0.50	**1.74**	0.06	0.06	0.07	0.07	0.01	0.04	0.02
filler_gap_embed_3	0.55	**3.54**	0.46	0.50	0.21	0.21	0.07	0.01	0.53
filler_gap_embed_4	0.52	**3.23**	0.32	0.30	0.12	0.12	0.04	0.01	0.27
filler_gap_hierarchy	0.50	**3.79**	1.22	1.23	0.61	0.61	0.22	0.05	1.24
filler_gap_obj	0.80	**5.72**	2.54	2.46	1.24	1.28	0.10	0.04	2.54
filler_gap_pp	0.54	**3.85**	0.70	0.65	0.33	0.31	0.09	0.03	0.61
filler_gap_subj	0.49	**2.15**	0.17	0.13	0.03	0.02	0.10	0.00	0.12
Average	0.62	**3.94**	1.16	1.04	0.48	0.50	0.11	0.03	1.40

Table 3: pythia-14m

Task	Task Acc.	Feature-finding methods							Vanilla
		DAS	Probe	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.58	0.07	**1.01**	0.44	0.26	0.28	0.04	0.05	0.78
agr_sv_num_subj-relc	0.61	2.03	2.65	2.23	0.20	0.12	0.19	0.02	**2.76**
agr_sv_num_obj-relc	0.79	1.05	2.87	2.94	0.32	0.37	0.11	0.03	**3.03**
agr_sv_num_pp	0.77	2.62	4.66	4.20	0.55	0.22	0.22	0.04	**4.82**
agr_refl_num_subj-relc	0.78	1.97	2.93	2.39	0.17	0.12	0.14	0.05	**3.05**
agr_refl_num_obj-relc	0.72	1.14	1.86	1.44	0.11	0.11	0.09	0.05	**1.95**
agr_refl_num_pp	0.83	1.95	2.91	2.12	0.26	0.16	0.16	0.03	**3.21**
npi_any_subj-relc	0.56	**2.15**	0.45	0.51	0.38	0.39	0.07	0.04	0.55
npi_any_obj-relc	0.57	**2.02**	0.39	0.44	0.33	0.33	0.07	0.04	0.45
npi_ever_subj-relc	0.38	**1.13**	0.12	0.11	0.20	0.18	0.04	0.02	0.12
npi_ever_obj-relc	0.41	**0.72**	0.12	0.12	0.18	0.20	0.03	0.05	0.13
garden_mvrr	0.63	**3.56**	1.24	1.59	1.09	1.57	0.23	0.11	1.56
garden_mvrr_mod	0.50	**2.89**	0.99	1.34	1.31	1.35	0.40	0.28	1.60
garden_npz_obj	0.83	**4.55**	0.22	0.77	0.76	0.78	0.09	0.18	1.18
garden_npz_obj_mod	0.66	**3.65**	0.16	0.18	0.18	0.19	0.42	0.02	0.75
garden_npz_v-trans	0.46	**1.46**	0.73	0.68	0.07	0.08	0.03	0.04	0.44
garden_npz_v-trans_mod	0.50	**0.38**	0.10	0.05	0.01	0.01	0.03	0.03	0.17
gss_subord	0.72	1.82	2.81	1.69	1.77	1.76	0.13	0.15	**3.77**
gss_subord_subj-relc	0.69	**1.89**	1.32	1.35	1.32	1.35	0.23	0.34	1.80
gss_subord_obj-relc	0.68	1.82	2.08	2.15	2.18	2.15	0.38	0.11	**2.64**
gss_subord_pp	0.84	**4.44**	1.43	1.38	1.35	1.38	0.40	0.22	1.86
cleft	0.50	3.15	4.77	3.69	0.88	1.00	0.32	0.07	**5.32**
cleft_mod	0.50	**0.61**	0.05	0.06	0.04	0.04	0.03	0.06	0.00
filler_gap_embed_3	0.55	**0.72**	0.48	0.50	0.19	0.19	0.03	0.03	0.52
filler_gap_embed_4	0.52	**0.57**	0.30	0.28	0.10	0.10	0.02	0.01	0.27
filler_gap_hierarchy	0.50	**1.46**	0.86	0.85	0.28	0.28	0.19	0.05	0.88
filler_gap_obj	0.80	**2.15**	1.90	1.80	0.91	0.93	0.16	0.05	1.83
filler_gap_pp	0.54	**0.85**	0.67	0.52	0.24	0.24	0.10	0.03	0.49
filler_gap_subj	0.49	**0.44**	0.03	0.02	0.01	0.01	0.09	0.01	0.02
Average	0.62	**1.84**	1.38	1.24	0.54	0.55	0.15	0.08	1.58

Table 4: pythia-14m (selectivity)

Task	Task Acc.	Feature-finding methods							Vanilla
		DAS	Probe	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.85	**2.50**	1.74	0.52	0.13	0.12	0.01	0.03	2.04
agr_sv_num_subj-relc	0.85	**4.56**	3.88	2.84	0.23	0.20	0.12	0.01	3.93
agr_sv_num_obj-relc	0.94	**4.44**	3.95	3.42	0.26	0.20	0.04	0.03	4.02
agr_sv_num_pp	0.77	**3.62**	3.28	2.37	0.32	0.18	0.05	0.02	3.31
agr_refl_num_subj-relc	0.87	**3.84**	3.50	2.54	0.17	0.13	0.17	0.02	3.68
agr_refl_num_obj-relc	0.88	3.66	3.47	2.57	0.20	0.17	0.15	0.03	**3.68**
agr_refl_num_pp	0.87	**4.25**	3.44	1.82	0.20	0.14	0.10	0.03	3.85
npi_any_subj-relc	0.84	**5.16**	2.04	2.01	1.05	1.05	0.05	0.02	2.09
npi_any_obj-relc	0.86	**5.72**	2.10	2.08	1.07	1.08	0.05	0.01	2.12
npi_ever_subj-relc	0.84	**6.09**	2.22	2.18	1.49	1.57	0.03	0.01	2.15
npi_ever_obj-relc	0.90	**6.39**	2.34	2.28	1.52	1.57	0.04	0.03	2.34
garden_mvrr	0.53	**5.89**	1.79	1.52	1.01	1.07	0.08	0.04	1.77
garden_mvrr_mod	0.50	**7.85**	1.61	1.03	0.91	1.02	0.14	0.04	1.64
garden_npz_obj	0.85	**9.93**	2.18	1.71	1.43	1.39	0.08	0.00	3.29
garden_npz_obj_mod	0.69	**9.14**	2.53	1.50	1.41	1.47	0.19	0.02	2.41
garden_npz_v-trans	0.62	**3.53**	1.01	0.79	0.07	0.08	0.07	0.01	1.14
garden_npz_v-trans_mod	0.51	**0.70**	0.04	0.04	0.02	0.02	0.01	0.01	0.07
gss_subord	0.72	**7.41**	3.05	2.84	2.76	2.81	0.06	0.01	4.48
gss_subord_subj-relc	0.89	**9.49**	2.08	1.52	1.42	1.41	0.12	0.04	2.86
gss_subord_obj-relc	0.93	**10.08**	2.05	1.63	1.55	1.60	0.17	0.02	2.84
gss_subord_pp	0.88	**8.83**	2.07	1.61	1.54	1.59	0.18	0.01	3.24
cleft	0.63	**12.54**	4.30	3.76	0.89	1.35	0.16	0.02	4.12
cleft_mod	0.50	**3.88**	0.20	0.07	0.01	0.02	0.01	0.01	0.05
filler_gap_embed_3	0.56	**3.04**	0.57	0.55	0.27	0.26	0.02	0.00	0.59
filler_gap_embed_4	0.52	**2.55**	0.24	0.23	0.13	0.13	0.01	0.01	0.27
filler_gap_hierarchy	0.54	**6.16**	2.36	2.36	0.92	0.86	0.10	0.00	2.38
filler_gap_obj	0.78	**8.87**	4.17	4.17	2.19	2.39	0.08	0.03	4.14
filler_gap_pp	0.65	**4.42**	1.19	1.17	0.45	0.43	0.04	0.02	1.13
filler_gap_subj	0.67	**4.16**	1.03	1.02	0.40	0.40	0.05	0.01	1.03
Average	0.74	**5.82**	2.22	1.80	0.83	0.85	0.08	0.02	2.44

Table 5: pythia-31m

Task	Task Acc.	Feature-finding methods							Vanilla
		DAS	Probe	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.85	**2.38**	1.81	0.66	0.45	0.41	0.01	0.06	2.02
agr_sv_num_subj-relc	0.85	4.06	5.43	4.12	0.36	0.28	0.16	0.02	**5.63**
agr_sv_num_obj-relc	0.94	3.30	5.30	4.83	0.38	0.28	0.05	0.03	**5.48**
agr_sv_num_pp	0.77	2.87	4.80	3.48	0.49	0.28	0.09	0.03	**4.92**
agr_refl_num_subj-relc	0.87	2.83	4.34	3.10	0.24	0.16	0.22	0.02	**4.48**
agr_refl_num_obj-relc	0.88	2.38	4.28	3.14	0.26	0.21	0.15	0.03	**4.44**
agr_refl_num_pp	0.87	3.47	4.25	2.28	0.25	0.15	0.13	0.04	**4.78**
npi_any_subj-relc	0.84	0.66	1.88	1.85	0.95	0.97	0.03	0.02	**1.93**
npi_any_obj-relc	0.86	0.79	1.89	1.90	0.99	1.00	0.02	0.02	**1.97**
npi_ever_subj-relc	0.84	1.45	**2.62**	2.50	1.60	1.75	0.02	0.02	2.50
npi_ever_obj-relc	0.90	1.37	**2.52**	2.42	1.58	1.63	0.03	0.03	2.47
garden_mvrr	0.53	**5.19**	1.45	1.22	0.84	0.88	0.22	0.06	1.37
garden_mvrr_mod	0.50	**3.68**	3.23	1.35	1.29	1.32	0.20	0.01	1.50
garden_npz_obj	0.85	**4.87**	1.96	1.75	1.56	1.51	0.14	0.02	2.35
garden_npz_obj_mod	0.69	**2.28**	1.38	0.94	0.92	0.94	0.29	0.03	1.60
garden_npz_v-trans	0.62	**1.83**	1.04	0.80	0.12	0.12	0.08	0.01	1.11
garden_npz_v-trans_mod	0.51	**0.27**	0.06	0.05	0.02	0.02	0.01	0.02	0.09
gss_subord	0.72	**5.21**	2.74	2.87	2.83	2.86	0.22	0.02	4.03
gss_subord_subj-relc	0.89	**4.64**	2.33	1.35	1.31	1.26	0.39	0.05	2.24
gss_subord_obj-relc	0.93	**4.30**	3.85	1.67	1.67	1.63	0.32	0.07	2.40
gss_subord_pp	0.88	**4.40**	2.25	1.26	1.23	1.25	0.37	0.02	2.74
cleft	0.63	5.58	**8.26**	7.15	1.80	2.65	0.30	0.01	7.92
cleft_mod	0.50	0.25	**0.40**	0.22	0.11	0.10	0.02	0.01	0.19
filler_gap_embed_3	0.56	**1.14**	0.58	0.56	0.28	0.28	0.03	0.01	0.63
filler_gap_embed_4	0.52	**0.85**	0.26	0.26	0.14	0.14	0.02	0.01	0.30
filler_gap_hierarchy	0.54	**2.66**	1.90	1.93	0.79	0.75	0.10	0.01	1.96
filler_gap_obj	0.78	**3.29**	3.04	2.99	1.58	1.76	0.09	0.03	3.00
filler_gap_pp	0.65	**1.76**	1.23	1.19	0.47	0.45	0.04	0.02	1.20
filler_gap_subj	0.67	**1.83**	1.04	1.00	0.46	0.47	0.06	0.01	1.05
Average	0.74	**2.75**	2.63	2.03	0.86	0.88	0.13	0.03	2.63

Table 6: pythia-31m (selectivity)

Task	Task Acc.	Feature-finding methods							Vanilla
		DAS	Probe	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.95	**3.29**	2.95	1.09	0.74	0.77	0.06	0.04	2.99
agr_sv_num_subj-relc	0.97	**4.77**	4.11	3.66	0.53	0.67	0.42	0.01	4.34
agr_sv_num_obj-relc	0.86	**3.60**	3.38	3.36	0.49	1.10	0.39	0.01	3.44
agr_sv_num_pp	1.00	**5.83**	4.66	4.20	0.47	0.74	0.21	0.01	4.96
agr_refl_num_subj-relc	0.93	**5.35**	3.80	2.88	0.32	0.22	0.18	0.01	3.99
agr_refl_num_obj-relc	0.90	**3.97**	2.94	2.16	0.29	0.22	0.15	0.01	3.02
agr_refl_num_pp	0.89	**5.13**	3.64	2.50	0.28	0.31	0.14	0.01	4.03
npi_any_subj-relc	0.73	**6.65**	1.77	1.83	0.97	0.98	0.02	0.01	1.97
npi_any_obj-relc	0.78	**7.03**	2.01	2.04	1.11	1.11	0.03	0.01	2.10
npi_ever_subj-relc	0.68	**6.69**	2.58	2.70	2.21	2.25	0.04	0.01	2.64
npi_ever_obj-relc	0.84	**8.18**	3.28	3.39	3.06	3.05	0.04	0.01	3.44
garden_mvrr	0.73	**10.69**	5.16	3.19	3.13	3.19	0.14	0.13	3.60
garden_mvrr_mod	0.63	**11.47**	2.83	1.70	1.68	1.70	0.26	0.00	2.80
garden_npz_obj	0.96	**12.71**	2.90	1.63	1.63	1.63	0.15	0.13	3.49
garden_npz_obj_mod	0.91	**12.97**	1.23	0.62	0.58	0.62	0.01	0.04	2.19
garden_npz_v-trans	0.80	**5.60**	2.48	1.38	0.29	0.29	0.03	0.08	2.63
garden_npz_v-trans_mod	0.61	**2.25**	0.52	0.42	0.11	0.12	0.01	0.02	0.58
gss_subord	0.87	**15.67**	3.67	2.84	2.83	2.84	0.17	0.09	3.63
gss_subord_subj-relc	0.68	**12.00**	2.93	2.12	2.10	2.12	0.17	0.06	3.00
gss_subord_obj-relc	0.77	**9.03**	3.07	2.31	2.30	2.31	0.14	0.01	3.15
gss_subord_pp	0.88	**10.79**	2.49	2.00	1.99	2.00	0.10	0.02	3.13
cleft	0.71	**14.55**	4.24	2.13	0.55	0.50	0.08	0.03	4.61
cleft_mod	0.50	**5.57**	0.27	0.23	0.18	0.18	0.00	0.02	0.36
filler_gap_embed_3	0.50	**4.48**	0.48	0.46	0.20	0.20	0.01	0.00	0.49
filler_gap_embed_4	0.51	**4.05**	0.39	0.39	0.16	0.16	0.01	0.01	0.37
filler_gap_hierarchy	0.55	**7.06**	2.85	2.88	1.36	1.36	0.02	0.02	2.89
filler_gap_obj	0.86	**9.54**	4.02	3.91	2.56	2.75	0.06	0.03	3.94
filler_gap_pp	0.59	**5.49**	1.66	1.65	0.70	0.71	0.05	0.01	1.66
filler_gap_subj	0.64	**5.98**	1.93	1.93	0.77	0.79	0.05	0.01	1.94
Average	0.77	**7.60**	2.70	2.12	1.16	1.20	0.11	0.03	2.81

Table 7: pythia-70m

Task	Task Acc.	Feature-finding methods							Vanilla
		DAS	Probe	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.95	1.68	2.75	1.54	1.19	1.25	0.06	0.04	**3.03**
agr_sv_num_subj-relc	0.97	3.08	5.34	4.76	0.67	0.85	0.55	0.02	**5.69**
agr_sv_num_obj-relc	0.86	2.30	4.47	4.45	0.65	1.44	0.51	0.02	**4.59**
agr_sv_num_pp	1.00	4.19	6.11	5.49	0.60	0.97	0.30	0.02	**6.55**
agr_refl_num_subj-relc	0.93	3.63	4.89	3.76	0.44	0.25	0.25	0.01	**5.04**
agr_refl_num_obj-relc	0.90	3.35	3.77	2.78	0.35	0.24	0.19	0.01	**3.80**
agr_refl_num_pp	0.89	4.00	4.50	3.14	0.35	0.36	0.18	0.01	**4.84**
npi_any_subj-relc	0.73	0.73	1.52	1.58	0.90	0.90	0.01	0.01	**1.69**
npi_any_obj-relc	0.78	1.01	1.72	1.73	1.00	1.00	0.01	0.01	**1.76**
npi_ever_subj-relc	0.68	1.35	2.34	**2.42**	2.00	2.01	0.08	0.02	2.39
npi_ever_obj-relc	0.84	1.88	2.97	**3.06**	2.77	2.79	0.07	0.01	3.06
garden_mvrr	0.73	3.87	**5.52**	3.27	3.20	3.27	0.18	0.21	3.41
garden_mvrr_mod	0.63	**8.92**	2.76	1.70	1.68	1.70	0.45	0.09	1.89
garden_npz_obj	0.96	**6.17**	2.04	0.44	0.44	0.44	0.16	0.29	1.85
garden_npz_obj_mod	0.91	**4.84**	0.92	0.41	0.36	0.41	0.13	0.05	1.50
garden_npz_v-trans	0.80	**3.17**	2.33	1.46	0.37	0.38	0.03	0.07	2.66
garden_npz_v-trans_mod	0.61	**1.70**	0.48	0.41	0.16	0.16	0.01	0.02	0.60
gss_subord	0.87	0.50	**3.53**	2.27	2.27	2.27	0.24	0.09	2.00
gss_subord_subj-relc	0.68	2.32	**2.63**	1.69	1.69	1.69	0.32	0.09	2.12
gss_subord_obj-relc	0.77	**4.88**	3.47	2.04	2.03	2.04	0.40	0.02	2.58
gss_subord_pp	0.88	**2.54**	2.24	1.52	1.51	1.52	0.09	0.05	2.09
cleft	0.71	5.82	7.67	3.77	1.07	0.98	0.19	0.07	**8.08**
cleft_mod	0.50	0.40	**0.55**	0.40	0.30	0.30	0.00	0.03	0.54
filler_gap_embed_3	0.50	**1.50**	0.37	0.35	0.15	0.15	0.02	0.00	0.38
filler_gap_embed_4	0.51	**1.31**	0.30	0.29	0.11	0.11	0.01	0.01	0.27
filler_gap_hierarchy	0.55	1.66	1.80	1.83	0.90	0.89	0.01	0.02	**1.89**
filler_gap_obj	0.86	2.76	**3.22**	3.09	2.01	2.16	0.07	0.02	3.18
filler_gap_pp	0.59	**2.08**	1.36	1.34	0.57	0.56	0.05	0.01	1.36
filler_gap_subj	0.64	**1.60**	1.43	1.41	0.58	0.59	0.03	0.01	1.43
Average	0.77	**2.87**	2.86	2.15	1.05	1.09	0.16	0.05	2.77

Table 8: pythia-70m (selectivity)

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.99	**5.64**	3.89	2.87	1.52	0.96	0.96	0.06	0.03	4.35
agr_sv_num_subj-relc	0.96	**4.79**	4.10	3.20	2.52	0.19	0.22	0.26	0.01	4.20
agr_sv_num_obj-relc	0.95	**4.52**	4.49	4.01	3.22	0.24	0.26	0.36	0.00	4.19
agr_sv_num_pp	0.98	**5.02**	4.42	3.48	2.91	0.23	0.20	0.25	0.01	4.49
agr_refl_num_subj-relc	0.92	**4.56**	3.93	2.85	1.99	0.11	0.13	0.28	0.01	3.91
agr_refl_num_obj-relc	0.94	**4.65**	3.74	2.71	1.97	0.19	0.17	0.38	0.00	3.83
agr_refl_num_pp	0.91	3.49	3.23	2.07	1.52	0.10	0.07	0.17	0.01	**3.58**
npi_any_subj-relc	0.86	**8.09**	2.57	2.58	2.57	1.34	1.36	0.05	0.01	2.74
npi_any_obj-relc	0.98	**9.28**	3.82	3.82	3.83	1.85	1.89	0.10	0.01	3.87
npi_ever_subj-relc	0.82	**8.59**	3.88	3.90	3.92	3.81	3.98	0.09	0.01	3.69
npi_ever_obj-relc	1.00	**10.14**	5.74	5.72	5.71	5.53	5.69	0.18	0.01	5.72
garden_mvrr	0.87	**12.14**	6.10	3.84	2.90	2.85	2.90	0.13	0.05	3.71
garden_mvrr_mod	0.57	**10.04**	3.66	2.06	1.57	1.55	1.57	0.14	0.05	2.86
garden_npz_obj	0.88	**12.51**	2.42	1.92	1.76	1.75	1.76	0.07	0.09	3.03
garden_npz_obj_mod	0.89	**14.14**	1.56	1.31	1.30	1.30	1.30	0.15	0.03	2.51
garden_npz_v-trans	0.72	**4.59**	2.63	2.34	1.48	0.18	0.18	0.03	0.01	2.46
garden_npz_v-trans_mod	0.66	**2.23**	0.97	0.69	0.53	0.12	0.13	0.02	0.01	1.21
gss_subord	0.75	**17.03**	4.10	3.19	2.64	2.63	2.64	0.36	0.05	3.32
gss_subord_subj-relc	0.81	**8.82**	1.50	1.38	1.19	1.17	1.19	0.07	0.04	2.01
gss_subord_obj-relc	0.87	**8.66**	2.20	1.99	1.82	1.81	1.82	0.08	0.07	2.50
gss_subord_pp	0.86	**8.86**	1.62	1.57	1.38	1.37	1.38	0.06	0.05	2.25
cleft	1.00	**14.41**	6.08	3.89	2.44	0.42	0.43	0.03	0.00	6.99
cleft_mod	0.54	**6.93**	0.63	0.38	0.28	0.11	0.12	0.02	0.01	0.81
filler_gap_embed_3	0.50	**4.72**	0.12	0.13	0.13	0.09	0.09	0.01	0.00	0.19
filler_gap_embed_4	0.50	**4.33**	0.03	0.03	0.03	0.03	0.02	0.01	0.00	0.02
filler_gap_hierarchy	0.69	**6.75**	3.12	3.12	3.11	1.41	1.40	0.05	0.01	3.17
filler_gap_obj	0.87	**10.14**	4.09	4.07	4.07	3.05	3.24	0.09	0.01	4.09
filler_gap_pp	0.73	**7.11**	3.04	3.02	3.02	1.08	1.10	0.04	0.01	2.95
filler_gap_subj	0.77	**7.70**	3.23	3.22	3.21	1.20	1.18	0.04	0.01	3.20
Average	0.82	**7.93**	3.13	2.60	2.23	1.26	1.29	0.12	0.02	3.17

Table 9: pythia-160m; Probe^0 has $\lambda = 10^4$, Probe^1 has $\lambda = 10^5$.

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.99	**4.85**	4.30	3.33	1.72	1.17	1.19	0.06	0.05	4.49
agr_sv_num_subj-relc	0.96	3.66	5.37	4.21	3.33	0.29	0.31	0.32	0.02	**5.47**
agr_sv_num_obj-relc	0.95	3.65	**6.09**	5.45	4.37	0.41	0.44	0.47	0.01	5.71
agr_sv_num_pp	0.98	3.58	**5.94**	4.67	3.88	0.34	0.32	0.33	0.02	5.88
agr_refl_num_subj-relc	0.92	3.44	**4.43**	3.21	2.21	0.12	0.14	0.32	0.01	4.25
agr_refl_num_obj-relc	0.94	3.25	**4.30**	3.11	2.23	0.22	0.21	0.43	0.00	4.29
agr_refl_num_pp	0.91	2.90	3.96	2.54	1.86	0.16	0.13	0.20	0.02	**4.40**
npi_any_subj-relc	0.86	1.77	2.54	2.56	2.54	1.38	1.40	0.05	0.01	**2.72**
npi_any_obj-relc	0.98	2.44	3.85	3.86	3.87	1.98	2.02	0.08	0.01	**3.96**
npi_ever_subj-relc	0.82	1.65	3.53	3.57	3.59	3.51	**3.66**	0.10	0.01	3.37
npi_ever_obj-relc	1.00	2.93	**5.68**	5.66	5.65	5.46	5.63	0.17	0.02	5.66
garden_mvrr	0.87	2.15	**4.87**	3.61	2.96	2.90	2.96	0.19	0.13	3.47
garden_mvrr_mod	0.57	**3.35**	2.31	1.56	1.37	1.36	1.37	0.20	0.10	2.08
garden_npz_obj	0.88	**2.15**	1.45	1.20	1.27	1.27	1.27	0.08	0.14	1.63
garden_npz_obj_mod	0.89	**6.72**	0.58	0.90	1.11	1.11	1.11	0.21	0.09	1.73
garden_npz_v-trans	0.72	2.40	**2.71**	2.47	1.51	0.12	0.12	0.05	0.01	2.61
garden_npz_v-trans_mod	0.66	**1.60**	0.94	0.68	0.52	0.13	0.13	0.03	0.01	1.18
gss_subord	0.75	3.53	**3.53**	3.17	2.91	2.91	2.91	0.39	0.11	2.62
gss_subord_subj-relc	0.81	**2.77**	0.81	1.03	0.97	0.96	0.97	0.09	0.05	1.68
gss_subord_obj-relc	0.87	**3.35**	2.81	2.44	2.26	2.24	2.26	0.12	0.07	3.08
gss_subord_pp	0.86	**2.64**	0.79	1.23	1.21	1.20	1.20	0.18	0.05	1.70
cleft	1.00	6.50	11.12	7.20	4.60	0.86	0.87	0.07	0.01	**12.44**
cleft_mod	0.54	1.01	1.91	1.21	0.97	0.50	0.51	0.05	0.02	**2.00**
filler_gap_embed_3	0.50	**1.57**	0.07	0.08	0.08	0.10	0.10	0.01	0.00	0.15
filler_gap_embed_4	0.50	**1.44**	0.02	0.02	0.02	0.03	0.03	0.01	0.00	0.02
filler_gap_hierarchy	0.69	1.53	2.50	2.49	2.48	1.11	1.11	0.05	0.01	**2.56**
filler_gap_obj	0.87	2.69	2.86	2.84	2.84	2.19	2.29	0.07	0.01	**2.89**
filler_gap_pp	0.73	2.50	**2.62**	2.59	2.58	0.88	0.87	0.04	0.01	2.45
filler_gap_subj	0.77	2.88	**2.95**	2.92	2.91	1.06	1.03	0.04	0.01	2.88
Average	0.82	2.93	3.27	2.75	2.34	1.24	1.26	0.15	0.04	**3.36**

Table 10: pythia-160m (selectivity)

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	1.00	3.86	2.99	2.80	1.45	1.03	1.03	0.05	0.06	**4.01**
agr_sv_num_subj-relc	0.97	**4.97**	4.32	4.19	3.72	0.29	0.28	1.88	0.03	4.42
agr_sv_num_obj-relc	0.99	**5.61**	5.01	5.46	4.52	0.32	0.29	2.22	0.03	5.01
agr_sv_num_pp	0.97	**5.63**	4.96	4.83	4.55	0.38	0.35	1.15	0.04	5.09
agr_refl_num_subj-relc	0.92	3.65	3.86	3.77	1.85	0.15	0.14	0.61	0.02	**3.90**
agr_refl_num_obj-relc	0.96	**4.43**	4.03	4.04	1.90	0.31	0.31	0.63	0.02	3.96
agr_refl_num_pp	0.89	3.90	3.73	3.20	1.92	0.16	0.16	0.37	0.02	**4.00**
npi_any_subj-relc	0.95	**8.76**	4.01	4.00	3.99	2.28	2.34	0.15	0.01	4.08
npi_any_obj-relc	0.96	**8.59**	4.08	4.07	4.06	2.47	2.52	0.24	0.00	4.11
npi_ever_subj-relc	0.99	**12.12**	6.94	6.90	6.90	6.68	6.91	0.27	0.01	6.81
npi_ever_obj-relc	1.00	**12.15**	7.12	7.07	7.06	6.90	7.06	0.31	0.00	7.04
garden_mvrr	0.89	**19.74**	3.62	5.04	4.47	4.46	4.47	0.08	0.23	5.35
garden_mvrr_mod	0.61	**17.40**	1.85	2.43	3.22	3.21	3.22	0.13	0.12	4.72
garden_npz_obj	0.90	**19.03**	3.33	3.72	2.99	2.98	2.99	0.33	0.12	4.30
garden_npz_obj_mod	0.85	**20.07**	1.79	1.96	1.95	1.95	1.95	0.15	0.28	3.29
garden_npz_v-trans	0.81	**5.43**	2.87	3.22	1.53	0.18	0.18	0.05	0.05	2.88
garden_npz_v-trans_mod	0.67	**2.55**	1.17	1.17	0.62	0.10	0.10	0.04	0.01	1.69
gss_subord	0.82	**22.47**	3.42	3.18	4.35	4.35	4.35	0.21	0.21	5.00
gss_subord_subj-relc	0.85	**14.07**	2.17	2.47	2.43	2.43	2.43	0.17	0.07	3.36
gss_subord_obj-relc	0.94	**13.50**	1.81	1.81	2.37	2.37	2.37	0.13	0.05	2.96
gss_subord_pp	0.93	**13.24**	1.86	2.21	2.52	2.52	2.52	0.11	0.08	3.56
cleft	0.95	**14.46**	5.53	4.84	1.78	0.86	0.92	0.05	0.02	5.79
cleft_mod	0.67	**11.27**	3.37	2.93	1.52	1.25	1.27	0.03	0.03	3.58
filler_gap_embed_3	0.52	**3.98**	0.96	0.98	0.98	0.37	0.35	0.02	0.00	1.00
filler_gap_embed_4	0.50	**3.11**	0.31	0.34	0.34	0.17	0.15	0.01	0.00	0.33
filler_gap_hierarchy	0.87	**9.71**	4.97	4.96	4.95	2.94	3.43	0.11	0.01	4.95
filler_gap_obj	0.82	**11.28**	3.97	3.97	3.97	3.54	3.92	0.12	0.01	4.03
filler_gap_pp	0.88	**10.22**	5.07	5.03	5.02	2.77	2.64	0.12	0.01	4.96
filler_gap_subj	0.89	**11.84**	6.53	6.44	6.41	4.87	4.98	0.14	0.01	6.34
Average	0.86	**10.24**	3.64	3.69	3.22	2.15	2.19	0.34	0.05	4.16

Table 11: pythia-410m; Probe^0 has $\lambda = 10^4$, Probe^1 has $\lambda = 10^5$.

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	1.00	2.77	3.06	3.00	1.64	1.29	1.29	0.05	0.12	**4.08**
agr_sv_num_subj-relc	0.97	3.64	**5.63**	5.45	4.84	0.40	0.38	2.43	0.03	5.52
agr_sv_num_obj-relc	0.99	4.05	6.86	**7.42**	6.01	0.42	0.38	2.93	0.05	6.78
agr_sv_num_pp	0.97	4.46	6.51	6.32	5.95	0.50	0.47	1.49	0.05	**6.55**
agr_refl_num_subj-relc	0.92	2.79	**4.42**	4.23	2.10	0.15	0.11	0.72	0.02	4.36
agr_refl_num_obj-relc	0.96	2.76	**4.79**	4.78	2.26	0.34	0.33	0.81	0.02	4.61
agr_refl_num_pp	0.89	3.32	4.53	3.77	2.31	0.10	0.09	0.46	0.01	**4.68**
npi_any_subj-relc	0.95	2.13	4.13	4.12	4.12	2.31	2.39	0.14	0.01	**4.26**
npi_any_obj-relc	0.96	2.34	4.22	4.20	4.20	2.53	2.58	0.27	0.01	**4.28**
npi_ever_subj-relc	0.99	2.00	**6.74**	6.68	6.67	6.44	6.68	0.30	0.01	6.58
npi_ever_obj-relc	1.00	2.34	7.68	7.64	7.63	7.47	7.63	0.35	0.01	**7.70**
garden_mvrr	0.89	4.67	4.10	**4.96**	4.10	4.10	4.10	0.20	0.25	4.70
garden_mvrr_mod	0.61	**7.25**	2.09	2.16	2.96	2.96	2.96	0.13	0.20	3.53
garden_npz_obj	0.90	**9.84**	2.54	2.66	1.89	1.89	1.89	0.46	0.15	2.96
garden_npz_obj_mod	0.85	**7.40**	1.51	1.07	1.09	1.09	1.09	0.20	0.26	1.93
garden_npz_v-trans	0.81	2.75	3.22	**3.82**	1.72	0.31	0.31	0.06	0.05	3.39
garden_npz_v-trans_mod	0.67	1.06	1.06	1.15	0.52	0.09	0.09	0.04	0.01	**1.63**
gss_subord	0.82	**6.63**	3.41	2.60	4.00	4.00	4.00	0.17	0.20	3.84
gss_subord_subj-relc	0.85	**7.29**	3.02	2.78	2.43	2.43	2.43	0.33	0.06	3.04
gss_subord_obj-relc	0.94	**6.26**	2.34	1.93	2.36	2.36	2.36	0.15	0.04	3.08
gss_subord_pp	0.93	**5.34**	1.97	1.84	2.41	2.41	2.41	0.12	0.12	2.88
cleft	0.95	6.91	12.09	10.55	3.99	2.02	2.13	0.08	0.02	**12.91**
cleft_mod	0.67	3.47	7.93	7.07	3.87	3.18	3.20	0.08	0.07	**8.35**
filler_gap_embed_3	0.52	**1.10**	0.76	0.76	0.76	0.29	0.28	0.03	0.01	0.79
filler_gap_embed_4	0.50	**0.62**	0.19	0.20	0.20	0.11	0.10	0.01	0.00	0.22
filler_gap_hierarchy	0.87	0.88	**3.40**	3.36	3.35	1.96	2.30	0.15	0.01	3.27
filler_gap_obj	0.82	3.24	3.16	3.14	3.14	2.74	3.13	0.06	0.02	**3.27**
filler_gap_pp	0.88	3.22	**4.22**	4.13	4.11	2.25	2.08	0.09	0.01	3.92
filler_gap_subj	0.89	4.23	**6.28**	6.04	5.98	4.05	4.18	0.15	0.01	5.81
Average	0.86	3.96	4.20	4.06	3.33	2.07	2.12	0.43	0.06	**4.45**

Table 12: pythia-410m (selectivity)

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	1.00	**4.72**	2.43	2.27	1.32	0.90	0.90	0.01	0.05	3.85
agr_sv_num_subj-relc	1.00	**6.62**	5.23	4.98	4.24	0.36	0.39	1.40	0.01	5.51
agr_sv_num_obj-relc	0.94	**5.30**	4.75	4.62	3.72	0.38	0.41	1.39	0.02	4.44
agr_sv_num_pp	0.94	**5.81**	4.78	4.45	3.86	0.36	0.40	0.82	0.03	4.91
agr_refl_num_subj-relc	0.85	**4.13**	3.57	2.76	1.78	0.17	0.21	0.48	0.01	3.51
agr_refl_num_obj-relc	1.00	**5.89**	4.71	4.09	2.60	0.39	0.42	0.47	0.00	4.86
agr_refl_num_pp	0.82	**3.62**	3.00	2.18	1.43	0.23	0.26	0.29	0.01	3.46
npi_any_subj-relc	0.97	**10.51**	4.33	4.33	4.33	2.80	2.93	0.20	0.00	4.41
npi_any_obj-relc	0.99	**10.70**	4.33	4.33	4.33	2.85	2.94	0.34	0.01	4.36
npi_ever_subj-relc	0.99	**14.09**	7.14	7.11	7.09	6.67	7.10	0.28	0.01	7.00
npi_ever_obj-relc	1.00	**13.85**	6.87	6.84	6.84	6.62	6.84	0.41	0.00	6.93
garden_mvrr	0.91	**19.24**	4.82	5.23	4.67	4.67	4.67	0.19	0.12	5.63
garden_mvrr_mod	0.63	**17.51**	2.86	2.88	3.37	3.38	3.37	0.10	0.03	4.83
garden_npz_obj	0.89	**21.13**	3.33	3.67	2.52	2.52	2.52	0.35	0.09	4.12
garden_npz_obj_mod	0.84	**22.09**	2.08	2.33	1.82	1.81	1.82	0.17	0.06	3.20
garden_npz_v-trans	0.73	**5.43**	2.58	2.48	1.40	0.22	0.23	0.07	0.01	2.91
garden_npz_v-trans_mod	0.72	**2.65**	0.87	0.80	0.62	0.07	0.07	0.03	0.01	1.54
gss_subord	0.82	**20.00**	2.87	3.48	4.84	4.85	4.84	0.32	0.08	6.17
gss_subord_subj-relc	0.87	**11.91**	2.00	2.39	2.57	2.57	2.57	0.10	0.05	3.74
gss_subord_obj-relc	0.92	**13.31**	2.21	2.38	2.73	2.73	2.73	0.18	0.08	3.32
gss_subord_pp	0.94	**11.44**	2.19	2.42	2.67	2.67	2.67	0.17	0.03	3.79
cleft	0.97	**15.56**	5.59	3.76	1.62	0.25	0.43	0.16	0.01	6.04
cleft_mod	0.81	**11.99**	3.47	2.33	1.51	1.15	1.18	0.02	0.03	3.93
filler_gap_embed_3	0.62	**6.40**	1.08	1.09	1.09	0.41	0.41	0.02	0.01	1.04
filler_gap_embed_4	0.54	**5.79**	0.57	0.59	0.59	0.23	0.23	0.01	0.00	0.57
filler_gap_hierarchy	0.83	**8.69**	3.90	3.90	3.90	1.86	1.94	0.10	0.00	4.01
filler_gap_obj	0.76	**10.69**	3.61	3.63	3.63	3.33	3.50	0.22	0.00	3.70
filler_gap_pp	0.85	**10.52**	4.69	4.67	4.67	1.78	1.80	0.02	0.00	4.55
filler_gap_subj	0.89	**11.82**	6.23	6.18	6.17	3.76	3.87	0.03	0.00	6.22
Average	0.86	**10.74**	3.66	3.52	3.17	2.07	2.13	0.29	0.03	4.23

Table 13: pythia-1b; Probe^0 has $\lambda = 10^5$, Probe^1 has $\lambda = 10^6$.

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	1.00	3.66	3.00	2.73	2.06	1.65	1.65	0.01	0.09	**4.33**
agr_sv_num_subj-relc	1.00	4.13	6.97	6.59	5.60	0.42	0.46	1.75	0.01	**7.32**
agr_sv_num_obj-relc	0.94	3.94	**6.55**	6.32	5.10	0.41	0.45	1.82	0.03	6.01
agr_sv_num_pp	0.94	4.74	6.26	5.78	5.00	0.47	0.51	1.01	0.05	**6.27**
agr_refl_num_subj-relc	0.85	3.20	**4.13**	3.19	2.09	0.12	0.15	0.62	0.01	4.10
agr_refl_num_obj-relc	1.00	3.49	5.21	4.50	2.83	0.31	0.37	0.59	0.01	**5.26**
agr_refl_num_pp	0.82	2.62	3.67	2.62	1.77	0.15	0.19	0.35	0.02	**4.12**
npi_any_subj-relc	0.97	3.32	4.63	4.62	4.62	2.95	3.08	0.21	0.01	**4.83**
npi_any_obj-relc	0.99	3.36	4.57	4.57	4.57	2.96	3.03	0.36	0.01	**4.66**
npi_ever_subj-relc	0.99	2.02	**7.12**	7.09	7.08	6.66	7.10	0.24	0.01	7.00
npi_ever_obj-relc	1.00	2.01	7.26	7.23	7.23	6.99	7.23	0.57	0.01	**7.36**
garden_mvrr	0.91	3.36	**4.92**	4.65	3.36	3.37	3.36	0.37	0.20	3.65
garden_mvrr_mod	0.63	**4.54**	3.15	2.38	2.16	2.17	2.16	0.07	0.09	2.49
garden_npz_obj	0.89	2.85	**3.26**	3.00	0.91	0.91	0.91	0.54	0.16	2.15
garden_npz_obj_mod	0.84	**6.47**	1.73	1.40	0.39	0.39	0.39	0.08	0.07	1.15
garden_npz_v-trans	0.73	2.45	2.80	2.83	1.69	0.42	0.43	0.09	0.02	**3.68**
garden_npz_v-trans_mod	0.72	1.31	0.75	0.71	0.57	0.09	0.08	0.03	0.01	**1.50**
gss_subord	0.82	**6.89**	3.30	3.31	3.76	3.76	3.76	0.33	0.11	3.90
gss_subord_subj-relc	0.87	**4.98**	2.55	2.40	1.73	1.73	1.73	0.12	0.07	2.40
gss_subord_obj-relc	0.92	**5.68**	2.88	2.67	1.95	1.95	1.95	0.25	0.14	2.56
gss_subord_pp	0.94	**3.50**	2.67	2.48	1.96	1.96	1.96	0.21	0.06	2.59
cleft	0.97	4.05	11.38	7.64	3.36	0.58	0.96	0.33	0.03	**12.31**
cleft_mod	0.81	2.64	7.35	4.88	3.27	2.52	2.58	0.02	0.05	**8.34**
filler_gap_embed_3	0.62	**1.56**	1.00	1.01	1.01	0.37	0.37	0.02	0.01	0.97
filler_gap_embed_4	0.54	**0.96**	0.47	0.48	0.48	0.16	0.16	0.00	-0.00	0.47
filler_gap_hierarchy	0.83	0.86	2.61	2.60	2.60	1.24	1.26	0.14	0.01	**2.77**
filler_gap_obj	0.76	1.23	2.51	2.51	2.51	2.46	2.45	0.28	0.01	**2.55**
filler_gap_pp	0.85	3.39	**4.21**	4.17	4.16	1.61	1.62	0.02	0.00	3.94
filler_gap_subj	0.89	3.51	**5.94**	5.82	5.80	2.96	3.25	0.03	0.01	5.74
Average	0.86	3.34	4.24	3.80	3.09	1.78	1.85	0.36	0.04	**4.29**

Table 14: pythia-1b (selectivity)

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	1.00	**3.62**	2.58	2.00	1.11	0.58	0.58	0.00	0.02	3.24
agr_sv_num_subj-relc	0.98	**4.82**	4.24	4.07	3.89	0.48	0.44	2.43	0.02	4.32
agr_sv_num_obj-relc	0.97	**5.11**	4.89	4.46	3.97	0.49	0.61	2.37	0.02	4.44
agr_sv_num_pp	0.99	**5.75**	4.94	4.77	4.58	0.54	0.37	0.71	0.01	5.11
agr_refl_num_subj-relc	0.94	**3.44**	3.27	2.19	1.83	0.17	0.38	0.92	0.01	3.31
agr_refl_num_obj-relc	0.99	**4.02**	3.96	2.71	1.98	0.29	0.39	0.91	0.01	3.85
agr_refl_num_pp	0.96	**3.87**	3.07	2.15	1.91	0.22	0.25	0.40	0.01	3.67
npi_any_subj-relc	0.96	**9.16**	4.16	4.15	4.14	2.08	2.17	0.20	0.00	4.35
npi_any_obj-relc	0.96	**8.85**	4.19	4.18	4.17	2.20	2.27	0.28	0.00	4.43
npi_ever_subj-relc	1.00	**12.95**	7.08	7.06	7.06	6.85	7.06	0.57	0.01	6.92
npi_ever_obj-relc	1.00	**12.99**	6.90	6.85	6.84	6.56	6.84	0.49	0.01	6.89
garden_mvrr	0.85	**18.23**	3.18	3.59	3.76	3.76	3.76	0.22	0.06	4.86
garden_mvrr_mod	0.61	**15.37**	1.38	1.41	2.45	2.45	2.45	0.04	0.04	4.09
garden_npz_obj	0.98	**17.52**	3.27	3.19	2.33	2.33	2.33	0.10	0.06	4.44
garden_npz_obj_mod	0.87	**17.76**	2.17	1.73	1.56	1.56	1.56	0.12	0.02	3.14
garden_npz_v-trans	0.78	**5.48**	2.97	2.50	1.54	0.31	0.31	0.03	0.02	3.32
garden_npz_v-trans_mod	0.67	**2.25**	0.93	0.73	0.65	0.15	0.15	0.03	0.01	1.59
gss_subord	0.83	**19.10**	2.86	2.53	3.88	3.88	3.88	0.03	0.13	4.90
gss_subord_subj-relc	0.90	**9.04**	1.52	1.97	2.21	2.21	2.21	0.11	0.04	3.39
gss_subord_obj-relc	0.98	**11.14**	1.52	1.77	2.42	2.42	2.42	0.07	0.03	3.39
gss_subord_pp	0.93	**9.89**	1.89	2.24	2.45	2.45	2.45	0.05	0.02	3.79
cleft	1.00	**14.65**	5.25	3.85	1.54	0.25	0.28	0.13	0.01	5.86
cleft_mod	0.80	**11.72**	3.72	2.62	1.64	1.20	1.22	0.01	0.01	4.34
filler_gap_embed_3	0.55	**5.14**	1.16	1.19	1.19	0.30	0.29	0.03	0.00	1.21
filler_gap_embed_4	0.53	**4.35**	0.38	0.40	0.39	0.16	0.14	0.01	0.00	0.43
filler_gap_hierarchy	0.94	**9.25**	5.01	5.00	4.98	2.96	3.07	0.13	0.00	5.27
filler_gap_obj	0.76	**10.51**	3.62	3.64	3.64	3.57	3.66	0.23	0.00	3.69
filler_gap_pp	0.86	**9.86**	4.71	4.69	4.68	1.72	2.06	0.02	0.01	4.74
filler_gap_subj	0.90	**11.93**	6.20	6.10	6.08	4.60	4.96	0.03	0.01	6.14
Average	0.88	**9.58**	3.48	3.23	3.06	1.96	2.02	0.37	0.02	4.11

Table 15: pythia-1.4b; Probe^0 has $\lambda = 10^5$, Probe^1 has $\lambda = 10^6$.

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	1.00	2.59	2.66	2.18	1.29	0.86	0.86	0.00	0.04	**3.33**
agr_sv_num_subj-relc	0.98	3.52	**5.70**	5.43	5.19	0.59	0.56	3.20	0.02	5.67
agr_sv_num_obj-relc	0.97	3.36	**6.73**	6.08	5.21	0.63	0.79	3.06	0.03	6.08
agr_sv_num_pp	0.99	4.14	6.71	6.42	6.15	0.72	0.52	0.87	0.03	**6.85**
agr_refl_num_subj-relc	0.94	2.76	3.86	2.63	2.32	0.20	0.50	1.24	0.01	**3.89**
agr_refl_num_obj-relc	0.99	2.91	**4.52**	3.05	2.25	0.25	0.36	1.06	0.01	4.31
agr_refl_num_pp	0.96	3.19	3.81	2.69	2.39	0.23	0.28	0.48	0.02	**4.29**
npi_any_subj-relc	0.96	1.77	4.37	4.35	4.35	2.13	2.23	0.17	0.01	**4.56**
npi_any_obj-relc	0.96	1.98	4.43	4.42	4.41	2.24	2.32	0.28	0.00	**4.60**
npi_ever_subj-relc	1.00	1.69	**7.07**	7.07	7.07	6.91	7.07	0.69	0.01	7.04
npi_ever_obj-relc	1.00	1.75	7.38	7.34	7.33	7.09	7.33	0.51	0.01	**7.45**
garden_mvrr	0.85	3.01	3.35	3.55	3.36	3.36	3.36	0.21	0.09	**4.12**
garden_mvrr_mod	0.61	**2.68**	1.70	1.32	2.05	2.05	2.05	0.05	0.05	2.55
garden_npz_obj	0.98	**2.96**	2.57	2.17	1.34	1.34	1.34	0.07	0.09	2.77
garden_npz_obj_mod	0.87	**6.63**	2.24	1.41	1.13	1.13	1.13	0.06	0.05	1.78
garden_npz_v-trans	0.78	2.23	3.43	2.91	1.80	0.36	0.36	0.05	0.03	**4.12**
garden_npz_v-trans_mod	0.67	0.98	0.87	0.69	0.60	0.09	0.08	0.03	0.01	**1.41**
gss_subord	0.83	**5.34**	2.57	2.06	3.40	3.40	3.40	0.03	0.14	4.00
gss_subord_subj-relc	0.90	**3.92**	2.38	2.59	2.35	2.35	2.35	0.07	0.08	3.22
gss_subord_obj-relc	0.98	**6.99**	2.87	2.88	2.80	2.80	2.80	0.27	0.07	3.60
gss_subord_pp	0.93	**3.61**	2.79	2.87	2.53	2.53	2.53	0.28	0.03	3.50
cleft	1.00	4.36	10.96	8.10	3.26	0.50	0.56	0.27	0.01	**12.25**
cleft_mod	0.80	3.16	7.74	5.35	3.28	2.35	2.40	0.02	0.02	**8.71**
filler_gap_embed_3	0.55	1.02	1.01	1.03	1.03	0.28	0.28	0.03	0.01	**1.05**
filler_gap_embed_4	0.53	**0.86**	0.23	0.25	0.25	0.13	0.11	0.01	0.00	0.32
filler_gap_hierarchy	0.94	0.69	3.57	3.53	3.51	2.06	2.17	0.05	0.01	**3.75**
filler_gap_obj	0.76	1.85	2.49	2.50	2.50	2.46	**2.51**	0.26	0.01	2.48
filler_gap_pp	0.86	3.27	**4.43**	4.36	4.35	1.41	1.77	0.02	0.01	4.28
filler_gap_subj	0.90	3.44	**5.94**	5.74	5.69	3.70	4.14	0.03	0.01	5.53
Average	0.88	2.99	4.08	3.62	3.21	1.87	1.94	0.46	0.03	**4.40**

Table 16: pythia-1.4b (selectivity)

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	1.00	**4.99**	2.53	1.89	1.08	0.38	0.38	0.01	0.01	3.54
agr_sv_num_subj-relc	0.97	**5.87**	5.15	4.99	4.19	0.34	0.35	2.21	0.00	5.30
agr_sv_num_obj-relc	0.97	**6.14**	5.68	5.88	5.06	0.39	0.38	2.28	0.00	5.65
agr_sv_num_pp	0.97	**6.21**	5.69	5.38	4.64	0.34	0.33	0.27	0.00	5.92
agr_refl_num_subj-relc	0.97	**4.90**	3.48	2.89	2.06	0.16	0.16	0.94	0.00	3.94
agr_refl_num_obj-relc	0.97	**5.98**	4.15	3.32	1.95	0.31	0.34	0.76	0.01	4.66
agr_refl_num_pp	0.94	**4.95**	3.36	2.49	1.85	0.21	0.21	0.52	0.00	4.12
npi_any_subj-relc	0.94	**9.39**	3.83	3.80	3.79	1.84	1.89	0.26	0.00	4.02
npi_any_obj-relc	0.96	**9.23**	3.61	3.58	3.57	1.88	1.92	0.30	0.01	3.84
npi_ever_subj-relc	1.00	**13.55**	7.05	7.02	7.01	6.87	7.01	0.36	0.01	6.93
npi_ever_obj-relc	1.00	**13.84**	7.18	7.12	7.10	6.90	7.10	0.44	0.01	7.34
garden_mvrr	0.82	**11.90**	3.99	4.47	3.21	3.20	3.21	0.03	0.02	4.60
garden_mvrr_mod	0.58	**10.37**	2.08	2.37	2.03	2.03	2.03	0.03	0.01	3.80
garden_npz_obj	0.93	**11.57**	1.94	2.46	2.15	2.14	2.15	0.03	0.01	3.94
garden_npz_obj_mod	0.82	**11.22**	1.70	1.82	1.27	1.27	1.27	0.05	0.01	2.98
garden_npz_v-trans	0.81	**5.87**	2.98	2.57	1.74	0.30	0.33	0.03	0.01	3.57
garden_npz_v-trans_mod	0.75	**3.15**	1.34	1.18	0.87	0.13	0.13	0.04	0.00	2.11
gss_subord	0.86	**12.98**	3.29	3.97	3.13	3.12	3.13	0.01	0.01	4.49
gss_subord_subj-relc	0.89	**7.25**	1.72	2.13	2.08	2.08	2.08	0.02	0.01	3.35
gss_subord_obj-relc	0.97	**7.62**	2.01	2.33	2.50	2.50	2.50	0.07	0.01	3.63
gss_subord_pp	0.93	**8.10**	1.81	2.39	2.36	2.36	2.36	0.02	0.01	3.85
cleft	1.00	**14.74**	5.50	4.52	2.60	0.23	0.27	0.06	0.01	6.30
cleft_mod	0.86	**12.06**	3.77	3.37	2.44	1.37	1.37	0.01	0.01	4.52
filler_gap_embed_3	0.57	**5.89**	1.85	1.89	1.88	0.43	0.45	0.03	0.00	1.88
filler_gap_embed_4	0.54	**4.73**	0.89	0.94	0.94	0.31	0.28	0.03	0.01	1.00
filler_gap_hierarchy	0.94	**9.44**	4.35	4.33	4.32	2.93	3.20	0.16	0.00	4.78
filler_gap_obj	0.79	**11.23**	3.98	4.04	4.03	3.87	4.02	0.08	0.00	4.20
filler_gap_pp	0.88	**11.09**	5.46	5.39	5.37	2.37	3.03	0.02	0.00	5.28
filler_gap_subj	0.94	**13.24**	7.55	7.37	7.30	5.73	6.03	0.03	0.00	7.42
Average	0.88	**8.88**	3.72	3.65	3.19	1.93	2.00	0.31	0.01	4.38

Table 17: pythia-2.8b; Probe^0 has $\lambda = 10^5$, Probe^1 has $\lambda = 10^6$.

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	1.00	**4.21**	2.63	2.36	1.62	1.06	1.07	0.01	0.02	3.72
agr_sv_num_subj-relc	0.97	3.13	6.91	6.65	5.57	0.41	0.42	2.92	0.01	**7.07**
agr_sv_num_obj-relc	0.97	3.47	7.60	**7.79**	6.69	0.41	0.38	2.98	0.01	7.49
agr_sv_num_pp	0.97	4.61	7.94	7.49	6.42	0.48	0.48	0.37	0.01	**8.00**
agr_refl_num_subj-relc	0.97	3.18	4.21	3.42	2.45	0.15	0.14	1.15	0.00	**4.62**
agr_refl_num_obj-relc	0.97	2.62	4.70	3.71	2.19	0.24	0.30	0.90	0.01	**5.06**
agr_refl_num_pp	0.94	3.90	4.44	3.25	2.46	0.18	0.17	0.63	0.01	**5.17**
npi_any_subj-relc	0.94	1.59	4.24	4.21	4.20	2.04	2.10	0.30	0.01	**4.57**
npi_any_obj-relc	0.96	1.39	4.18	4.14	4.13	2.14	2.18	0.28	0.01	**4.35**
npi_ever_subj-relc	1.00	1.65	**7.33**	7.29	7.28	7.11	7.29	0.53	0.01	7.11
npi_ever_obj-relc	1.00	1.65	7.57	7.52	7.49	7.29	7.49	0.50	0.01	**7.68**
garden_mvrr	0.82	1.67	2.84	**3.37**	2.18	2.16	2.18	0.05	0.04	3.28
garden_mvrr_mod	0.58	**2.30**	1.09	1.38	1.05	1.05	1.06	0.01	0.01	1.93
garden_npz_obj	0.93	1.74	0.45	0.51	0.54	0.54	0.54	0.04	0.02	**1.77**
garden_npz_obj_mod	0.82	**2.95**	1.11	1.08	0.41	0.41	0.41	0.02	0.01	1.21
garden_npz_v-trans	0.81	1.78	3.60	3.18	2.10	0.50	0.52	0.03	0.02	**4.90**
garden_npz_v-trans_mod	0.75	0.97	1.34	1.19	0.88	0.13	0.13	0.04	0.00	**2.23**
gss_subord	0.86	2.36	2.08	**2.78**	1.82	1.82	1.82	0.02	0.02	2.29
gss_subord_subj-relc	0.89	**3.37**	1.14	1.70	1.42	1.42	1.42	0.02	0.02	2.17
gss_subord_obj-relc	0.97	**5.10**	1.58	1.88	1.72	1.72	1.72	0.09	0.02	3.02
gss_subord_pp	0.93	**3.29**	1.26	1.85	1.66	1.66	1.66	0.03	0.02	2.74
cleft	1.00	3.79	12.47	10.36	5.95	0.50	0.60	0.14	0.01	**14.04**
cleft_mod	0.86	2.77	8.53	7.53	5.37	2.88	2.89	0.01	0.01	**9.89**
filler_gap_embed_3	0.57	1.27	1.71	1.72	1.70	0.42	0.43	0.05	0.00	**1.74**
filler_gap_embed_4	0.54	**1.41**	0.79	0.82	0.82	0.24	0.24	0.05	0.01	0.81
filler_gap_hierarchy	0.94	0.52	3.21	3.14	3.12	2.13	2.32	0.14	0.01	**3.70**
filler_gap_obj	0.79	1.38	2.78	2.83	2.82	2.79	2.82	0.08	0.01	**2.96**
filler_gap_pp	0.88	3.24	**5.10**	5.01	4.98	2.19	2.71	0.02	0.00	4.79
filler_gap_subj	0.94	3.16	**7.50**	7.17	7.06	4.97	5.39	0.03	0.00	7.02
Average	0.88	2.57	4.15	3.98	3.31	1.69	1.75	0.39	0.01	**4.67**

Table 18: pythia-2.8b (selectivity)

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.99	**4.18**	3.58	2.36	1.37	0.56	0.56	0.05	0.01	3.81
agr_sv_num_subj-relc	0.99	**5.23**	4.42	3.86	3.56	0.32	0.30	2.22	0.01	4.68
agr_sv_num_obj-relc	0.99	5.82	**6.16**	5.23	4.31	0.31	0.27	2.10	0.01	5.55
agr_sv_num_pp	0.99	**4.93**	4.16	3.66	3.44	0.32	0.28	0.04	0.01	4.56
agr_refl_num_subj-relc	0.94	4.00	3.52	2.41	2.24	0.14	0.13	0.07	0.01	**4.12**
agr_refl_num_obj-relc	1.00	**5.06**	4.48	2.86	2.23	0.25	0.27	0.09	0.01	4.58
agr_refl_num_pp	0.92	4.03	3.03	2.05	2.00	0.17	0.17	0.08	0.00	**4.49**
npi_any_subj-relc	0.96	**10.46**	3.73	3.74	3.75	1.68	1.72	0.28	0.00	4.03
npi_any_obj-relc	0.99	**11.13**	3.85	3.85	3.85	1.79	1.85	0.31	0.00	4.14
npi_ever_subj-relc	0.97	**14.01**	6.09	6.10	6.10	5.99	6.10	0.64	0.00	6.02
npi_ever_obj-relc	0.99	**15.05**	6.75	6.75	6.75	6.55	6.75	0.50	0.01	6.91
garden_mvrr	0.81	**16.98**	2.86	2.97	3.23	3.22	3.23	0.19	0.03	4.56
garden_mvrr_mod	0.57	**16.65**	1.37	1.10	2.18	2.18	2.18	0.01	0.03	4.00
garden_npz_obj	0.98	**16.11**	2.52	2.73	1.62	1.62	1.62	0.15	0.04	3.99
garden_npz_obj_mod	0.85	**17.55**	1.54	1.61	0.95	0.95	0.95	0.23	0.02	2.55
garden_npz_v-trans	0.81	**5.37**	2.55	1.94	1.56	0.27	0.27	0.05	0.01	3.27
garden_npz_v-trans_mod	0.71	**2.58**	0.98	0.75	0.54	0.08	0.08	0.03	0.00	1.86
gss_subord	0.87	**18.25**	2.52	2.16	3.50	3.49	3.50	0.04	0.05	5.42
gss_subord_subj-relc	0.87	**8.93**	1.66	1.78	2.14	2.14	2.14	0.13	0.01	3.12
gss_subord_obj-relc	0.99	**9.08**	1.93	2.02	2.50	2.50	2.50	0.18	0.02	3.45
gss_subord_pp	0.89	**9.65**	1.89	2.07	2.41	2.41	2.41	0.17	0.02	3.67
cleft	1.00	**14.71**	4.53	3.22	1.43	0.06	0.04	0.03	0.00	5.81
cleft_mod	0.96	**13.13**	4.41	3.27	2.12	1.46	1.49	0.01	0.01	5.31
filler_gap_embed_3	0.59	**5.83**	1.08	1.10	1.09	0.31	0.31	0.04	0.00	1.33
filler_gap_embed_4	0.52	**4.44**	0.32	0.33	0.33	0.13	0.12	0.01	0.00	0.32
filler_gap_hierarchy	0.90	**9.95**	4.34	4.35	4.33	2.83	3.51	0.22	0.00	4.84
filler_gap_obj	0.77	**11.15**	3.38	3.41	3.41	3.19	3.38	0.02	0.01	3.39
filler_gap_pp	0.92	**11.24**	5.04	5.04	5.02	2.61	2.88	0.04	0.01	5.19
filler_gap_subj	0.95	**12.97**	6.57	6.52	6.47	4.92	5.18	0.03	0.01	6.69
Average	0.89	**9.95**	3.42	3.08	2.91	1.81	1.87	0.27	0.01	4.20

Table 19: pythia-6.9b; Probe^0 has $\lambda = 10^6$, Probe^1 has $\lambda = 10^7$.

Task	Task Acc.	Feature-finding methods								Vanilla
		DAS	Probe^0	Probe^1	Mean	PCA	k-means	LDA	Rand.	
agr_gender	0.99	3.15	4.08	2.82	1.99	1.07	1.07	0.04	0.02	**4.20**
agr_sv_num_subj-relc	0.99	3.84	5.91	5.13	4.71	0.42	0.38	2.86	0.02	**6.34**
agr_sv_num_obj-relc	0.99	4.40	**8.22**	6.95	5.67	0.33	0.29	2.74	0.02	7.42
agr_sv_num_pp	0.99	3.58	5.83	5.09	4.76	0.42	0.36	0.05	0.02	**6.28**
agr_refl_num_subj-relc	0.94	2.28	4.10	2.82	2.59	0.14	0.10	0.09	0.01	**4.65**
agr_refl_num_obj-relc	1.00	2.74	5.03	3.19	2.49	0.19	0.23	0.11	0.01	**5.12**
agr_refl_num_pp	0.92	2.98	3.76	2.51	2.43	0.12	0.11	0.10	0.00	**5.42**
npi_any_subj-relc	0.96	1.16	4.13	4.14	4.15	1.72	1.76	0.36	0.01	**4.45**
npi_any_obj-relc	0.99	1.55	4.19	4.19	4.19	1.85	1.92	0.40	0.01	**4.53**
npi_ever_subj-relc	0.97	0.73	6.36	6.37	6.37	6.21	**6.37**	0.67	0.01	6.31
npi_ever_obj-relc	0.99	1.04	7.31	7.30	7.29	7.07	7.29	0.54	0.01	**7.34**
garden_mvrr	0.81	2.67	2.19	2.39	2.06	2.06	2.06	0.26	0.04	**2.89**
garden_mvrr_mod	0.57	**2.58**	0.79	0.54	0.95	0.95	0.95	0.00	0.07	1.79
garden_npz_obj	0.98	1.35	1.00	1.12	0.41	0.42	0.41	0.08	0.07	**2.10**
garden_npz_obj_mod	0.85	**5.03**	0.68	0.56	0.16	0.16	0.16	0.16	0.05	0.99
garden_npz_v-trans	0.81	2.04	3.30	2.53	2.02	0.49	0.49	0.05	0.01	**4.52**
garden_npz_v-trans_mod	0.71	1.08	1.03	0.78	0.56	0.11	0.11	0.03	0.00	**1.92**
gss_subord	0.87	**5.45**	1.47	1.16	2.09	2.08	2.09	0.14	0.05	3.07
gss_subord_subj-relc	0.87	**3.62**	1.22	1.36	1.10	1.09	1.10	0.10	0.02	1.61
gss_subord_obj-relc	0.99	**4.48**	1.20	1.33	1.26	1.26	1.26	0.33	0.04	2.29
gss_subord_pp	0.89	**3.04**	1.43	1.66	1.35	1.35	1.35	0.26	0.04	2.08
cleft	1.00	2.17	10.39	7.32	3.20	0.20	0.19	0.09	0.01	**12.71**
cleft_mod	0.96	2.23	10.15	7.42	4.79	3.33	3.39	0.02	0.01	**11.67**
filler_gap_embed_3	0.59	0.97	1.00	1.02	1.02	0.27	0.28	0.03	0.00	**1.20**
filler_gap_embed_4	0.52	**0.93**	0.21	0.22	0.22	0.09	0.08	0.01	0.00	0.23
filler_gap_hierarchy	0.90	0.43	2.04	2.03	2.02	1.32	1.68	0.11	0.01	**2.47**
filler_gap_obj	0.77	1.27	2.26	2.29	2.28	2.27	2.28	0.02	0.01	**2.30**
filler_gap_pp	0.92	2.44	4.43	4.42	4.40	2.30	2.44	0.03	0.01	**4.47**
filler_gap_subj	0.95	2.82	**6.18**	6.09	6.04	4.13	4.43	0.03	0.02	6.13
Average	0.89	2.48	3.79	3.27	2.85	1.50	1.54	0.34	0.02	**4.36**

Table 20: pythia-6.9b (selectivity)

F Odds-ratio plots for all methods on selected tasks

Figure 8: `agr_gender`

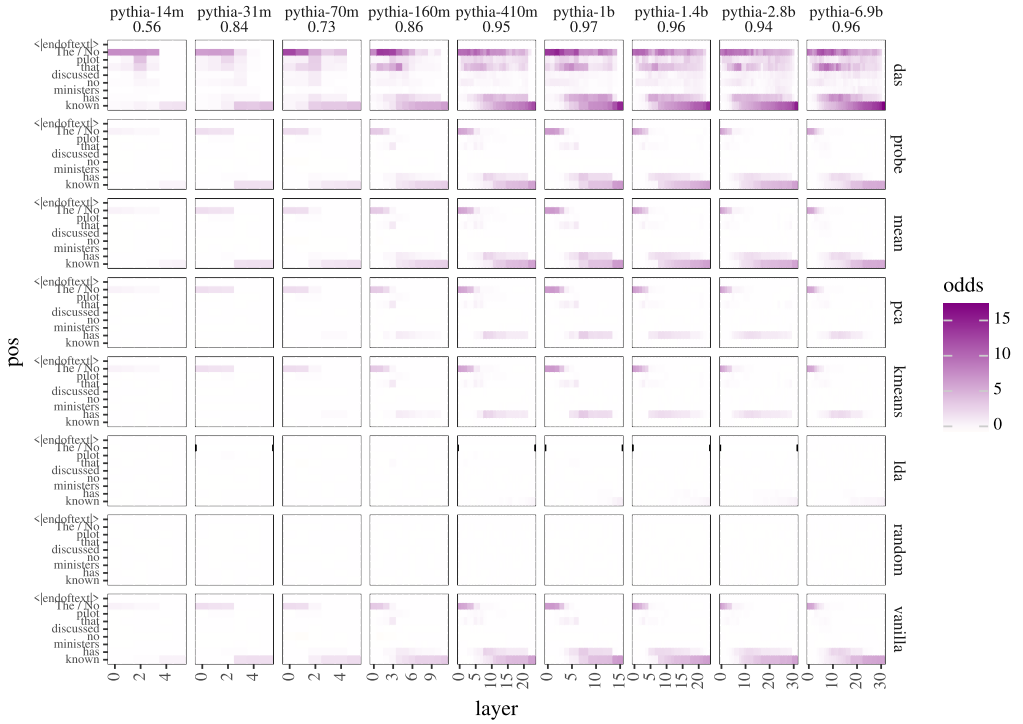


Figure 9: `npi_any_subj-relc`

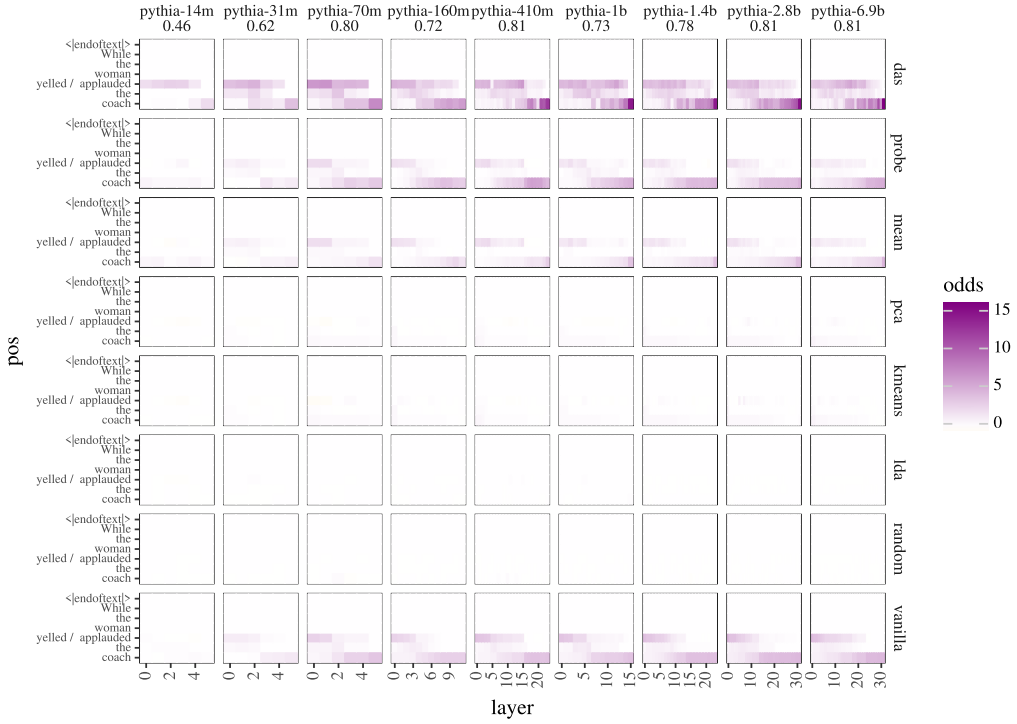


Figure 10: garden_npz_v-trans

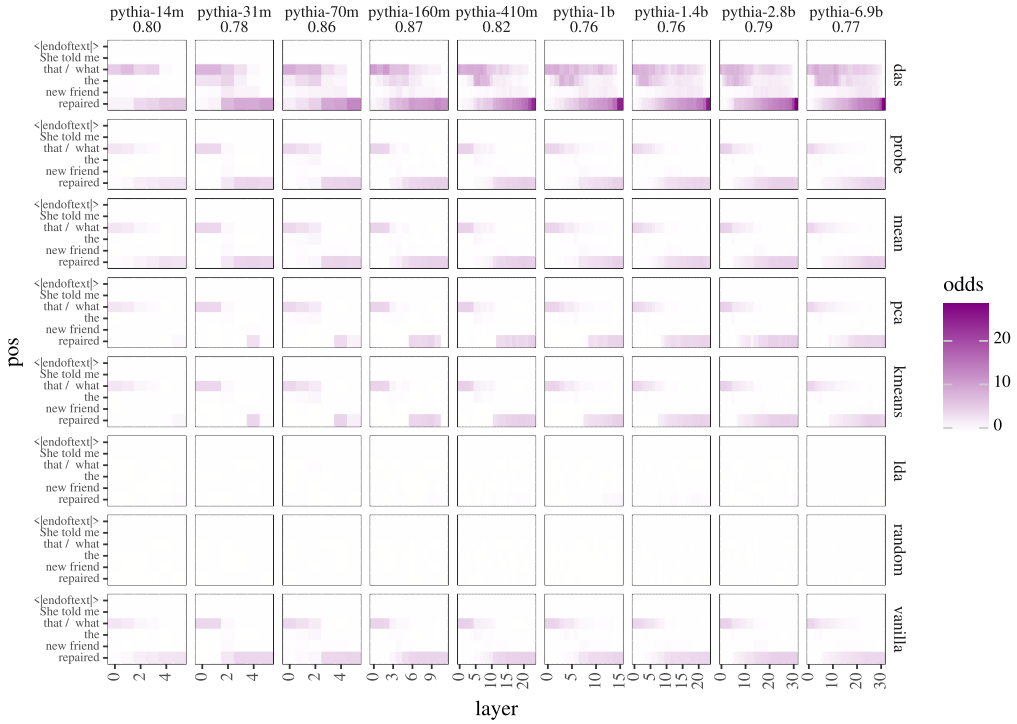


Figure 11: filler_gap_obj