Explaining Neural Networks with Reasons

Levin Hornischer[1] and Hannes Leitgeb[1]

[1]*Munich Center for Mathematical Philosophy, LMU Munich*

Abstract We propose a new interpretability method for neural networks, which is based on a novel mathematico-philosophical theory of reasons. Our method computes a vector for each neuron, called its *reasons vector*. We then can compute how strongly this reasons vector speaks for various *propositions*, e.g., the proposition that the input image depicts digit 2 or that the input prompt has a negative sentiment. This yields an interpretation of neurons, and groups thereof, that combines a logical and a Bayesian perspective, and accounts for polysemanticity (i.e., that a single neuron can figure in multiple concepts). We show, both theoretically and empirically, that this method is: (1) grounded in a philosophically established notion of explanation, (2) uniform, i.e., applies to the common neural network architectures and modalities, (3) scalable, since computing reason vectors only involves forward-passes in the neural network, (4) faithful, i.e., intervening on a neuron based on its reason vector leads to expected changes in model output, (5) correct in that the model's reasons structure matches that of the data source, (6) trainable, i.e., neural networks can be trained to improve their reason strengths, (7) useful, i.e., it delivers on the needs for interpretability by increasing, e.g., robustness and fairness.*

Keywords Deep learning, interpretability, explainable AI.

1. Introduction

Neural networks, the drivers of the recent boom in artificial intelligence (AI), excel at learning patterns from data. However, they are also notoriously opaque: the parameters that they find during training are difficult to interpret in human-understandable terms. Solving this problem is the goal of AI interpretability research [9, 31, 39, 38]. A prominent and rapidly growing approach is *mechanistic interpretability*. It aims to analyze the internal mechanisms of the neural network in order to understand and improve it [45, 44, 40, 19]. At AAAI 2025, Chalmers [4] argued that this should be done specifically in terms of *propositional attitudes*: using propositions, phrased in our language, that describe the AI system's goals and models of the world. Finding and logging these propositions is a research program that is "highly nontrivial" and "we don't yet have any broad and reliable techniques" [4, p. 10].

*The source code will eventually be made available here: https://github.com/LevinHornischer/ReasonsMethod.

In this paper, we suggest a first step. We build on a recent theory of reasons [30] that formalizes the language of *reasons*, which we ordinarily use to make sense of the world and the mechanisms in it—be it physical processes, the behavior of others, or engineering artifacts. Based on this theory, we develop a new interpretability method for neural networks. It makes sense of individual neurons, and groups thereof, as epistemic reasons that favor certain propositions with certain numerical strengths. We compute, for each neuron, its reasons vector, from which we can compute how much it speaks for each proposition. For example, for a neural network solving the MNIST task, we will compute, e.g., that this particular neuron speaks with strength 2.36 for the proposition that the input image depicts digit 3. Or in an LLM, we can compute that this group of five neurons speaks most strongly for the prompt having a positive sentiment.

We find that our reasons method satisfies ten desiderata for interpretability that we identify in the literature (section 2). The method applies to both individual neurons and groups thereof, and it is rooted in a fundamental conceptual framework of making sense of the world (section 3). The method can account for polysemanticity, since a single reason can speak for multiple propositions. It connects to the logico-symbolic tradition of understanding cognition by associating neurons with propositions, and it also connects to the Bayesian tradition by describing how neurons update subjective probabilities. In experiments (section 4), we see that the method applies across the common neural network architectures and modalities. It is scalable, since computing the reasons vector only involves forward-passes. The method is faithful, i.e., intervening according to the reasons brings about the expected change in behavior; and it is correct in the sense that the reason structure of a well-trained model matches that of the world. Reasons not only interpret trained models, but we can also train a model via backpropagation to improve its reasons strengths, and this also increases robustness and fairness.

2. Background: Desiderata for interpretability

We identify desiderata for any interpretability method that aims for mechanistic—or even propositional [4] —interpretability. Afterward, we discuss them and the respective literature.

1. *Understandable*: The interpretation should be in human-understandable terms.
2. *Local and distributed*: The interpretability method should interpret a neural network in both a local (individual neurons) and a distributed (groups of neurons) way.
3. *Mechanism-compatible*: The interpretation should reflect: (a) the encoding of inputs, (b) the real-valued activations of neurons across different inputs, (c) the decoding of outputs, and (d) how neurons interact via weights and activation functions with other neurons.
4. *Uniform*: The method should apply to all neural network architectures and data modalities.
5. *Scalable*: The method should work both for small and large neural networks.
6. *Transparent*: The interpretability method should not require further interpretation of its results or black-box training.

7. *Grounded*: The method should provide a philosophically deep notion of interpretation that supports comparisons between a neural network, its interpretation, and reality.

8. *Faithful*: The method should assign interpretations that represent faithfully, i.e., which track features of the network under relevant interventions.

9. *Correct*: The method should assign interpretations that represent correctly, i.e., where the structure of the interpretations tracks the intended structure of reality (the data).

10. *Useful*: The method should deliver on the needs for interpretability, i.e., trust, causality, transferability, fairness, privacy, robustness/reliability, recourse, and debugging.

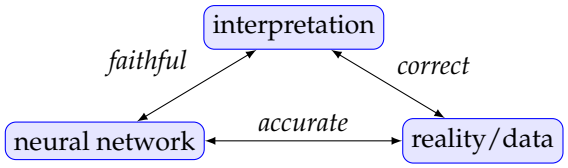
While 1 is uncontested, the literature is undecided on 2: whether it is individual neurons that should be interpreted (as in the first neural networks [34]) or rather groups thereof (distributed representation). See, e.g., [46, 42, 17] for discussion. Although interpretations of individual neurons have been suggested in some cases [41, 1], a *complete* localist representation is difficult, since neurons often are polysemantic, i.e., participate in several concepts—they are in 'superposition' [46, 42, 12]. But 2 asks for as much of a *partial* localist representation as possible, which ideally explains how distributed representations are built up. Our reasons method provides this: the reasons vector is a local representation since it is associated with a specific neuron, and it also partial since it 'pushes' into different conceptual dimensions rather than a single one (cf. superposition). Aggregating the reasons vectors of a group of neurons builds a distributed representation of the group.

Regarding 3, part (b) precludes an interpretation of a neuron's activation as a classical truth-value, but it allows more complex interpretations in terms of truth and falsity [24, 33]. The reason method will interpret activations as providing the components of the reasons vectors. Regarding (d), [51] discusses desiderata for circuit discovery. For us, weights and activation functions determine the reasons vector of a neuron based on the reasons vectors of the neurons in the preceding layer.

Regarding 4–6, one of the most prominent approaches to mechanistic interpretability, *sparse auto-encoders* (SAEs) [6, 3], required much work to scale [48]. Our reasons method only requires forward passes of the model, while SAEs require extensive training, which make them expensive to compute and difficult to evaluate [16, 23].

Desiderata 7–9 are explained in what we call the *triangle of interpretability* in figure 1 (left). The interpretability method should connect three components: (1) the neural network; (2) the human-understandable interpretations of the network parameters; (3) the reality, which is available via data. The connection (1)–(3) is measured as accuracy: it requires no interpretability since it just demands that the network's behavior matches reality. The connection (1)–(2) requires that the network's internal *mechanism* is captured by the interpretation. This is known as *faithfulness* and tested in terms of interventions [37, 19, 21, 50, 53, 36]. Intervening on the network parameters according to the interpretation should result in the expected change in behavior. While accuracy requires *behavioral correctness*, connection (2)–(3) requires *mechanistic correctness*. Given an interpretation of the network, we not only want it faithfully representing the internal mechanisms, we also want that the model's mechanisms match those of reality. In section 4.1, we operationalize this notion for our reasons interpretation via dimensionality reductions.

Finally, 10 is commonplace [31, 9], and we will find that reasons improve robustness and fairness.



	u_1	...	u_n
w_1	a_{11}	...	a_{1n}
$\vdots$	$\vdots$		$\vdots$
w_{2^m}	$a_{2^m 1}$	...	$a_{2^m n}$

Figure 1: *Left*: The triangle of interpretability. *Right*: The activation matrix.

3. Methodology: Reasons and the Reasons Method

We review the philosophical understanding of reasons and then the recently axiomatic theory of reasons [30]. Afterward, we apply it to develop our reasons method for interpreting neural networks.

Philosophy of reasons Talk of reasons for action (practical reasons) and reasons for belief (epistemic reasons) is omnipresent in everyday communication and much researched in philosophy [26]. Given our focus on interpretability, we consider epistemic reasons. Different reasons may support the same proposition, and one and the same reason may support different propositions. Moreover, reasons may do so with different strengths. Having a reason for a proposition A is a particular propositional attitude, which is required for an agent to believe A and which increases the probability the agent assigns to A (proportional to the reason's strength). Since reasons can act against each other and even defeat each other [25], a rational agent needs to aggregate all available reasons before forming a belief on their basis. One understanding of reasons is as 'epistemic forces' that should aggregate additively much like physical forces do. This suggests that reasons might have a vector structure, and aggregation and attenuation of reasons is vector addition and scalar multiplication, respectively.

Theory of reasons The theory of reasons [30] formalizes these philosophical ideas. It axiomatizes the following primitive notions:

- 'x is a direct epistemic reason of strength α for proposition A' (written $R(x, A, \alpha)$). E.g., x might be a strong reason for *there will be rain* presented by black clouds.
- '$x \circ y$ is the aggregation of reasons x and y'. E.g., y could be another reason presented by a weather forecast, and $x \circ y$ would be the aggregation of the two reasons.
- '$b * x$ is the result of updating the agent's current beliefs b with (the available) reason x'. E.g., if x speaks strongly for $A =$ *there will be rain*, the posterior subjective probability $b * x(A)$ should be significantly greater than the prior probability $b(A)$.

For a given reason x, it is not necessarily the case that for every proposition A there is an α, such that $R(x, A, \alpha)$. To generalize R to all propositions A, additional probabilistic weighing is required. For that purpose, the theory allows for the definition of a more inclusive reason relation:

- 'x is a doxastic reason of strength α for proposition A that is inferred relative to belief b' (written $S(x, A, \alpha, b)$). E.g., given my belief that tonight's party is sensitive to bad weather, black clouds speak with high strength for the party being canceled.

The axioms for these primitive notions include, e.g., the additivity of $\circ$ with respect to R: $R(x \circ y, A, \alpha)$ iff there are B, C, β, γ such that $R(x, B, \beta)$, $R(y, C, \gamma)$, $A = B \cap C \neq \emptyset$, and $\alpha = \beta + \gamma$. In contrast, $\circ$ is not generally additive with respect to S. The main mathematical result is that, surprisingly, the models of the overall axiomatic theory of reasons are unique up to a multiplicative constant $c > 0$. Hence we will work here directly with the models given for $c = 1$.

The models of the reasons theory are determined by a choice of a finite set $W = \{w_1, \ldots, w_{2^m}\}$, for some positive integer m. Its elements will be called *possible worlds* or *samples*. (They will be, as we will soon see, the situations in which the neural network can be applied, e.g., input-label pairs.) Together with the powerset $\mathcal{P}(W)$, it forms a measurable space. The elements of $\mathcal{P}(W)$ are called *propositions* or *events*—i.e., a proposition is identified with the set of possible worlds at which it is true. The *negation* or *complement* of A is given by: $A^c := W \setminus A$.[1] A *belief* is a probability measure on $(W, \mathcal{P}(W))$. A *reason* x is a vector in $\mathbb{R}^{2^m}$. Intuitively, x is the reason that speaks with strength x_k for w_k being the actual world—i.e., $R(x, \{w_k\}, x_k)$. Given a proposition $A \subseteq W$, the *elementary reason* for A, written el_A, is the vector in $\mathbb{R}^{2^m}$ which is 1 at component k if $w_k \in A$ and -1 otherwise. Reason aggregation $\circ$ is vector addition. (So aggregations of elementary reasons need not be elementary again.) Given a probability measure b on W and a reason $x \in \mathbb{R}^{2^m}$, the *update* of b by x is the probability measure defined by (for $A \subseteq W$):

$$b * x(A) := \frac{\sum_{k=1}^{2^m} e^{x_k} b(A \cap \{w_k\})}{\sum_{k=0}^{2^m} e^{x_k} b(\{w_k\})}.^{2} \tag{1}$$

The *doxastic reason strength* α with which x speaks for a proposition A relative to b (i.e., $S(x, A, \alpha, b)$) is defined by:

$$\alpha := D(x, A, b, W) := \frac{1}{2} \log \Big(\frac{b * x(A)/b * x(A^c)}{b(A)/b(A^c)} \Big). \tag{2}$$

which is defined if, and only if, the proposition A is *nontrivial*, i.e., $0 < b(A) < 1$.

Reasons method for interpretability Our reasons methods uses the reasons theory to interpret neurons and groups thereof as follows. The main conceptual choice is the set $W = \{w_1, \ldots, w_{2^m}\}$ of situations in which the neural network can be applied. For example, in an image classification task, this could be input-label pairs. (We will see many more examples in section 4.) Given a neuron u of the neural network, its *reasons vector* $r_u \in \mathbb{R}^{2^m}$ has, as value at component k, the activation that neuron u has in the possible world w_k.[3] In the example, if $w = (x, y)$ is an input-label pair, then r_u's value at w is simply the activation of neuron u after inputting x to the neural network.

Once the reasons vector r_u is computed, we can use it to interpret the neuron u in two ways—in line with logical and Bayesian tradition, respectively. (1) *Logico-symbolically*: The neuron u represents the proposition A consisting of those possible worlds at which r_u has a positive value. (2) *Probabilistically*: Relative to a prior probability measure b on W, the neuron u represents the probability distribution $b * r_u$. We get a combination of both—which we hence will use below—via the reasons theory. (3) *Strength-based*: Relative

[1] The terms 'possible worlds', 'propositions', 'negation' are used in philosophy, while 'sample', 'event', 'complement' are used in statistics.

[2] If b is the uniform measure, $b * x$ is the well-known softmax of x.

[3] We do not use the variable 'x' since it commonly refers to the input to the neural network.

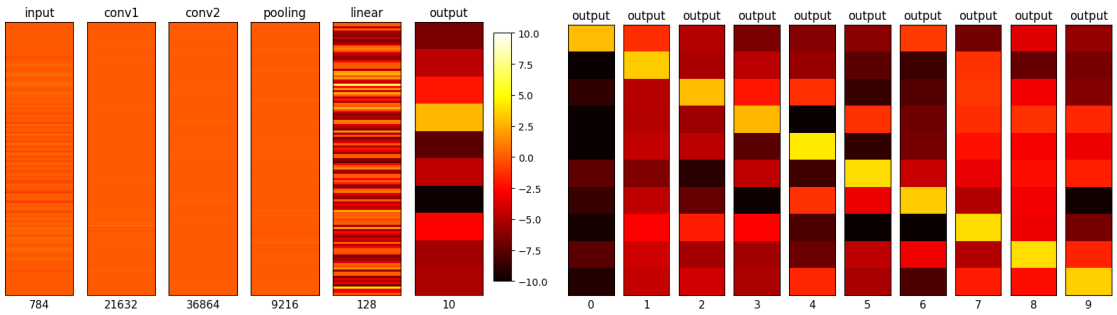


Figure 2: *Left*: For each neuron in the different layers of LeNet, the strength with which it speaks for (positive) or against (negative) the proposition 'The input depicts digit 3'. The number below the bars indicates the number of neurons in the layer. *Right*: For each digit d (shown below each bar), the reasons strength of the output neurons speak for 'The input depicts digit d'.

to a prior probability measure b on W, the neuron u represents a strength profile, i.e., how much it speaks for and against any nontrivial proposition. For example, if l is a label in the classification task, the proposition 'The input has label l' is the set $A = \{(x, y) \in W : y = l\}$, and neuron u speaks for it with strength $D(r_u, A, b, W)$.

Finally, we interpret a group of neurons $u_1, \ldots, , u_n$ by the aggregated reasons vector $r := r_{u_1} + \ldots + r_{u_n}$. This choice is corroborated by the general result of the reasons theory that update and aggregation commute: $b * r$ is the same as the convex combination of the $b * r_k$'s.

4. Experiments

To experimentally test our new interpretability method, we apply it in a wide range of tasks: involving different architectures (convolutional neural networks, multi-layer perceptrons, and transformer-based LLMs) and different modalities (images, tabular data, and text).

4.1. Interpreting a classic: LeNet for MNIST

Given the method's novelty, we first test it on a task—the MNIST task—with the classic architecture that solved it: the convolutional neural network *LeNet* [29]. The task is to classify images of handwritten digits according to which digit $(0, \ldots, 9)$ they depict. We train the LeNet architecture on the MNIST training set and achieve $> 99\%$ accuracy on the test set (details in appendix A).

Reason strengths To apply our reasons method, we choose the possible worlds as input-label pairs.[4] We sample a set W of 1024 such pairs (x, l) from the test set, so the model has not seen them.[5] The propositions of interest are 'The image depicts digit d', i.e., $A_d := \{(x, l) \in W : l = d\}$.

[4]We could add more information to a world: e.g., who wrote the digit; when and where it was written; or an intended label in addition to the correct label (in case, say, someone wrote what looks like a '7' but meant a '1').

[5]Appendix A establishes statistical robustness and shows no qualitative difference to using the training set.

Now, for each neuron u in the trained LeNet model, we can compute its reasons vector $r_u := (u(x) : (x, l) \in W)$, where $u(x)$ is the activation of neuron u in the model on input x. By taking the uniform measure b on W, we can compute, for each proposition A_d, the strength $D(r_u, A_d, b, W)$ with which neuron u speaks for the proposition A_d. This is shown, for $d = 3$, in figure 2 (left). We observe fairly low reason strengths in earlier layers and stronger ones (either positive or negative) in the later layers. This is in line with CNNs possessing a hierarchy of features: with earlier layers corresponding to low-level features such as basic shapes, while later layers correspond to more abstract features [52]. Focusing on the output neurons, figure 2 (right) shows their reasons strength for the different digits. As desired, for each digit d, the output neuron corresponding to d strongly speaks for the proposition 'The image depicts digit d', while the other output neurons strongly speak against it.

When it comes to interpreting groups of neurons, the layers make a natural choice. In appendix A, figure 8 shows how the layers (after aggregating the reasons vectors of their neurons) update an initially uniform prior probability distribution over the possible worlds. Even though we start with the uniform measure and use a balanced dataset, the input layer introduces some bias among the worlds—and this bias is amplified by later layers.

Faithfulness Next, we test the faithfulness of our interpretation via *causal interventions* [21, 19, 37] or, more precisely, *activation patching* [50, 53, 36, 18]. Specifically, we test this in two versions, for the hidden linear layer of our trained LeNet model.

Version 1: pos2neg. Fixing a digit d, we go through the test dataset considering images x that are labeled with d. We input x into the model, which, due to its high accuracy, will classify x almost always correctly as d. Now we consider the activations of the 20 neurons in the linear layer that most strongly speak against digit d. We intervene and set their activations to $a' := m - 3a$, where m is that neuron's mean activation and a is its current activation.[6] From these intervened activations in the linear layer, we forward propagate to calculate the intervened model output. It is a success if the model now predicts a digit different from d. Figure 3 (left) shows that, for all digits except 1, we have a 100% success rate. It also shows (as orange dots) the KL divergence between the originally outputted probability distribution over the digits and the one after intervention.

Version 2: neg2pos. Fixing a digit d, we now consider test images x that are *not* labeled with d. We input x into the model and now consider the activations of the 20 neurons in the linear layer that most strongly speak *for* digit d. We intervene to set their activations to $a' := m - 5a$ and calculate the intervened model output. It is a success if the model now predicts digit d. Note that this is much harder: intervening to do *anything* else is easier than doing something *specific*. Still, figure 3 (right) shows that, for all digits except $3, 4, 9$, we have an approximately 60% success rate and, for all digits, a KL divergence of around 30. In the next subsection, when we train the model's reasons, we see that these success rates improve—thus further corroborating faithfulness.

Correctness We need to operationalize the idea that the reasons structure of the neural network should match the reasons structure of the world. Inspired by theories of scientific

[6] Taking the mean—aka *mean ablating*—effectively 'knocks out' the neuron; and it does so better than *zero ablating*, i.e., setting the neuron to zero [51, 53]. Adding $-3a$ points the neuron in the opposite direction.

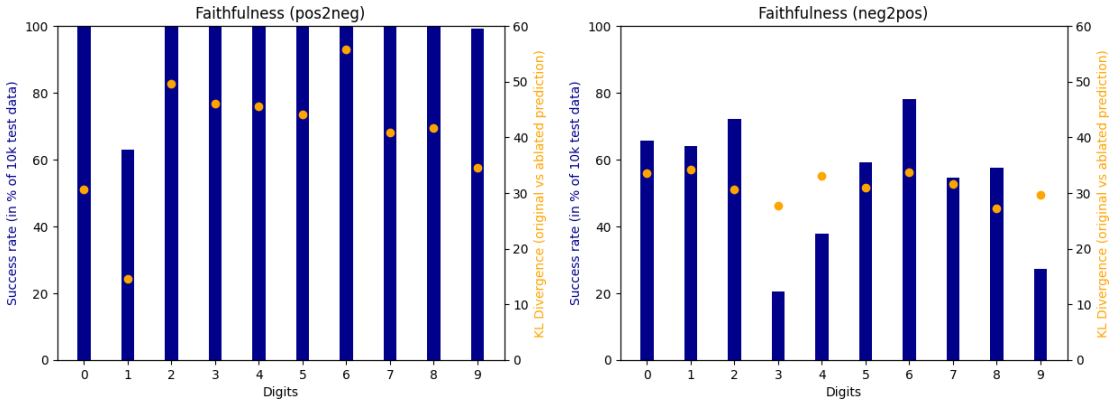


Figure 3: *Left*: intervening on neurons speaking against a digit to flip the prediction away from that digit. *Right*: intervening on neurons speaking for a digit to flip the prediction to that digit.

representation [15], we measure how much the representational similarity between possible worlds matches their objective similarity. This is done via the activation matrix in figure 1 (right). Given neurons $u_1, \ldots, u_n$ and possible worlds $w_1, \ldots, w_{2^m}$, the value a_{ij} is the activation of neuron u_j at world w_i. Thus, the j-th *column* is the reasons vector of neuron u_j, and we call the i-th *row* the *reasons-character* of the world w_i (cf. C^* algebras). Two worlds are *internally similar* if their reasons-characters are close as vectors in $\mathbb{R}^n$: the reason structure of the neural network almost cannot tell these worlds apart. Two worlds are *externally similar* if they have the same objective properties, i.e., the same label. Correctness requires that internal similarity typically (i.e., *defeasibly*) entails external similarity. Thus, worlds with the same label should form clusters in the space $\mathbb{R}^n$ of reasons-characters.

To observe potential clusters, we need to reduce the dimension from n to 2. So we perform a Principal Component Analysis (PCA). (Appendix A shows similar results for t-SNE and UMAP.) We sample 2^{12} worlds from the test set and consider the neurons of the hidden linear layer. We associate each of the 10 labels with a color. So if correctness holds, we should find monochromatic clusters—which is the case, as figure 4 (left) shows. If we do the same for the first convolutional layer, where we saw lower reasons strengths, figure 4 (right) indeed does not show such clusters.

4.2. Improving reasons: do good reasons lead to more robustness and fairness?

We used the reasons method to interpret a trained model, but can we also improve the model's reasons? So the model not just performs well, but does so, literally, "for the right reasons" [2, p. 5192]? The suggestive hope is that this delivers on the needs for interpretability: if the model has good reasons for its output, it should be more robust and fair.

Training reasons To improve a model's reasons via backpropagation, we need a loss function to measure the quality of its reasons with its current weights. Given weights w and a batch $x = (x_1, \ldots, x_N)$ of inputs with corresponding labels $y = (y_1, \ldots, y_N)$, we define the *doxastic reasons loss* $L(w, x, y)$. Let $\{l_1, \ldots, l_C\}$ be the set of classes (for MNIST this is the set of digits). Let $\hat{y}_k^w$ be the C-dimensional vector of logits produced by the model

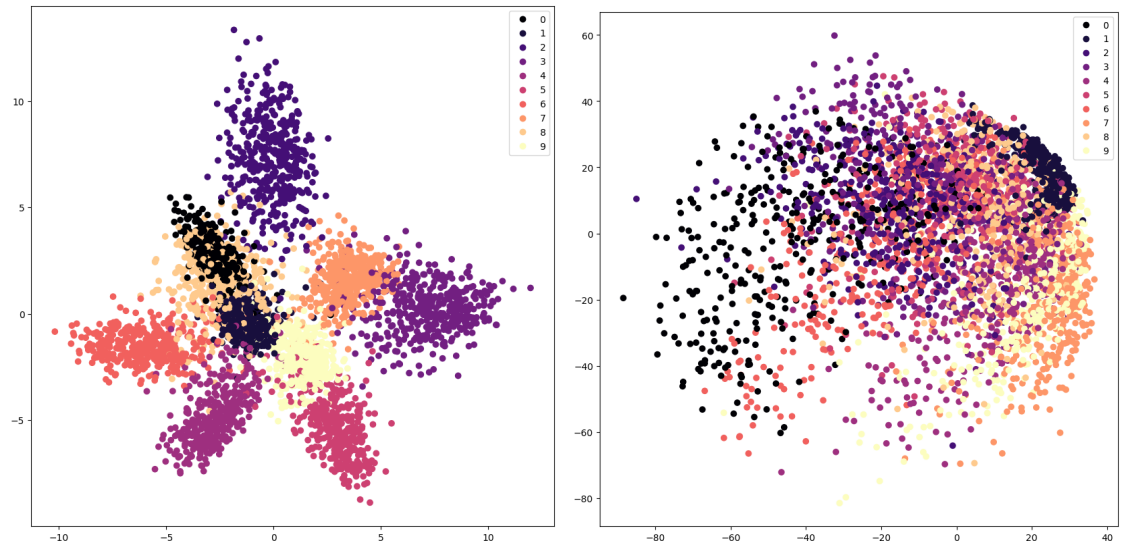


Figure 4: *Left*: After clustering together worlds (using PCA) that are internally similar according to the neurons in the hidden linear layer, they also are externally similar, i.e., have the same label. *Right*: This is not yet true for neurons in the first convolutional layer.

on input x_k using its weights w. We want that the d-th output neuron is a 'good' reason for label l_d. To formalize that, define $W := \{(x_k, y_k) : k = 1, \ldots, N\}$ as the set of worlds. For $d = 1, \ldots, C$, the reasons vector of the d-th output neuron is $r_d = (\hat{y}_k^{w}{}_d : k = 1, \ldots, N)$ and $A_d = \{(x, y) \in W : y = l_d\}$ is the proposition that the input has label l_d. The strength with which r_d speaks for A_d should be high, so

$$L(w, x, y) = \sum_{d=1}^{C} e^{-D(r_d, A_d, b, W)}. \tag{3}$$

We instantiate a LeNet model and train it using the sum of the usual loss (i.e., cross entropy) and this reasons loss. For comparison, we make a copy of the initial model and train it on the very same sequence of batches but with only the usual loss. Both models achieve $> 99\%$ accuracy. While correctness only marginally improves, faithfulness improves more: in the more difficult 'neg2pos' version, the model now achieves success rates between 60% and 80% for all digits (previously 6 digits were below 60%). More details are in appendix B.[7]

Robustness We can improve a model's reasons structure via training. But do good reasons make it harder to trick the model? To test this, we adversarially attack both the reasons-trained model and the comparison model with a FGSM attack [20]. This adds ϵ-much adversarially crafted noise to the input images. We check, for different choices of ϵ, how much accuracy decreases due to these attacks. We find that the model trained for reasons is, for all considered ϵ's, more immune to FGSM attacks than the comparison

[7] There, we also consider an alternative loss function, which we call the *elementary reasons loss*. Curiously, it improves faithfulness and correctness more than the doxastic reasons loss, but it does not improve robustness unlike the doxastic reasons loss (as we will see next). So these notions interact with reasons nontrivially.

model. For $\epsilon = 0.15$, this is 78.6% vs 69.9%, and for $\epsilon = 0.25$, this is 44.6% vs 27.1% (more details in appendix B). This is remarkable: First, nothing in the reasons training is specific to defending adversarial attacks.[8] Second, the reasons method increases interpretability and robustness while maintaining the same high accuracy. Thus, it defies general tradeoffs between accuracy and interpretability [11] and between accuracy and stability [5].

Fairness Moving to a different modality, we consider the task of predicting whether a person's income is above a given threshold based on tabular data about their age, occupation, sex, etc. We use the modernized *Adult* dataset due to [7], here focusing on US census data from Alabama in 2018. We consider two income thresholds: 25k and 50k. We train multi-layer perceptrons (MLPs) for this task. Treating sex as a protected attribute, we measure the MLPs' fairness using standard metrics: disparate impact (DI) [13] and equality of opportunity (EoO) [22]. We add a reasons-based fairness metric.

Given a list of inputs $x = (x_1, \ldots, x_N)$, let $\hat{y}^w = (\hat{y}^w_1, \ldots, \hat{y}^w_N)$ be the corresponding model outputs computed with its weights w. So $\hat{y}^w_k$ is the value of the single output neuron; if the sigmoid of it is > 0.5, the model predicts the income to be above the threshold. Let $W = \{x_1, \ldots, x_N\}$ be the set of worlds, let A_+ be the set of $x \in W$ which the model predicts to have an income above the threshold, and let A_p (resp., A_u) be the set of $x \in W$ that belong to the privileged (resp., unprivileged) group. The reasons vector of the single output neuron is $\hat{y}^w$. To be fair, the model's reasons strength for a positive prediction should be the same regardless of conditioning on the privileged group or the unprivileged group. So, with the uniform measure b on W, the *reasons difference* is:

$$\mathrm{RD}(w, x) := \big(\mathrm{D}(\hat{y}^w, A_+, b(\cdot|A_p), W) - \mathrm{D}(\hat{y}^w, A_+, b(\cdot|A_u), W)\big)^2. \tag{4}$$

This not only measures trained models (smaller is better), but also serves as a loss function to train models (an unsupervised one, since no labels are needed). To do so, we again initialize an MLP, make a copy, train the original model with the sum of the usual loss and the RD loss, and train the comparison model with only the usual loss (details in appendix B). We find that, in the 25k version, the model trained for reasons performs equally well as the comparison model, but it improves on RD. So there is a dimension of fairness in addition to DI and EoO that could still be improved. In the 50k version, the two models get perfect RD scores and perform equally well on the other metrics except DI: here the reasons trained model fares better. This again shows that the comparison model was not yet on the Pareto front of fairness. In sum, the reasons training could improve along fairness dimensions while keeping the same accuracy—again defying general fairness–accuracy tradeoffs [10].

4.3. Reasons in LLMs: mechanistic interpretability

A prominent tool for mechanistic interpretability of large language models (LLMs) are *sparse auto-encoders* (SAEs) [6, 3], which were scaled in [48] to also find abstract features represented by the neural network (e.g., sadness or sycophancy). As mentioned, SAEs are expensive to train and difficult to evaluate, so we test if our reasons method—which only needs forward-passes—also can identify such abstract features. For concreteness, we focus on one abstract feature—*sentiment*—since it is well-studied in NLP with established

[8] The point here also is not to introduce a new defense to adversarial attacks. Rather, we wanted to answer whether improved reasons lead to more robustness.

datasets and baselines.[9] We analyze the LLM Qwen2.5-0.5B-Instruct. Although small by today's standards, it solves the sentiment classification task (see below) and can easily be run on a laptop.

We identify which neurons in the residual stream of the model speak most strongly for positive and negative sentiment, respectively. To do so, we sample 1024 sentences from the SST2 dataset [47], which contains movie review excerpts (e.g., "contains no wit, only labored gags"). We use a two-shot prompt template asking about the sentence's sentiment (appendix C). The set W of prompts constructed from the selected sentences forms the set of worlds. Now, for each 'neuron'—or, rather, position—in the model's residual stream, we can compute its reasons vector: For each position $\mathfrak{n}$ (of the 896 embedding dimensions) and for each layer $\mathfrak{l}$ (of the 24 layers), the reasons vector $\mathfrak{r}_{\mathfrak{n},\mathfrak{l}}$ has, at component $w \in W$, the value $e_{\mathfrak{n}}$, where e is the embedding vector for the last token in layer $\mathfrak{l}$ given prompt w. We use the `NLTK` *SentimentIntensityAnalyzer* to rank the selected sentences by positivity and by negativity. Let A_+ (resp., A_-) be the set of worlds using the 25 most positive (resp., negative) sentences. Figure 5 (left) shows, for each position $\mathfrak{n}$ in layer $\mathfrak{l}$, the reason strength for positivity $D(\mathfrak{r}_{\mathfrak{n},\mathfrak{l}}, A_+, \mathfrak{b}, W)$ and for negativity $D(\mathfrak{r}_{\mathfrak{n},\mathfrak{l}}, A_-, \mathfrak{b}, W)$, respectively (with $\mathfrak{b}$ the uniform measure on W). Most of the strength is again found in the later layers.

Thus, we formed, in an automated way, hypotheses about the roles of the model's neurons. Next, we need to validate if this description of those components is correct [45]. Here, we again do this via causal interventions (cf. section 4.1). We sample 500 sentences from SST2 (different from those used for the worlds). We use a three-shot prompt template to classify a sentence as 'a) positive' or 'b) negative' (appendix C). Thus, the model achieves an accuracy of 91.2%. For each sentence, we also perform the following intervention in the last layer (before the unembedding). *Pos2neg*: if the model classifies the sentence correctly as positive, we set each of the 5 neurons that most strongly speak for positivity to $\mathfrak{a}' := \mathfrak{m} - 5\mathfrak{a}$, where $\mathfrak{m}$ is the neuron's mean activation and $\mathfrak{a}$ its current activation; and we set the 5 neurons speaking most against positivity to $\mathfrak{a}' = \mathfrak{m}$. *Neg2pos*: if the model classifies the sentence correctly as negative, we set the 5 neurons speaking most for negativity to $\mathfrak{a}' := \mathfrak{m} - 7\mathfrak{a}$; and we set the 5 neurons peaking most against negativity to $\mathfrak{a}' = \mathfrak{m}$. In 97.9% of the cases, the pos2neg intervention indeed flips the model prediction from 'positive' to 'negative'. The intervention drops the model's average next-token probability for 'a' from 67.2% to 27.8%. In 98.6% of the cases, the neg2pos intervention indeed flips the model prediction from 'negative' to 'positive'. The average probability for 'b' drops from 69.1% to 2.4%.

In figure 5 (right), we can also see this in generation. Given a prompt about the movie *Titanic*, we generate output with the model first as is, then with a positive intervention and then with a negative intervention. The positive (resp., negative) intervention sets the 5 neurons speaking most for positivity (resp., negativity) to $\mathfrak{a}' = 2\mathfrak{m}$ (resp., $\mathfrak{a}' = 20\mathfrak{m}$) and the 5 neurons speaking most against positivity (resp., negativity) to $\mathfrak{a}' = \mathfrak{m}$ (resp., $\mathfrak{a}' = \mathfrak{m}$). After the interventions, the output becomes noticeably more positive and negative, respectively, and this can also be observed statistically using the SentimentIntensityAnalyzer across 100 generations (see appendix C).

[9] For an older discussion of 'sentiment neurons' using LSTMs, see [43, 8].

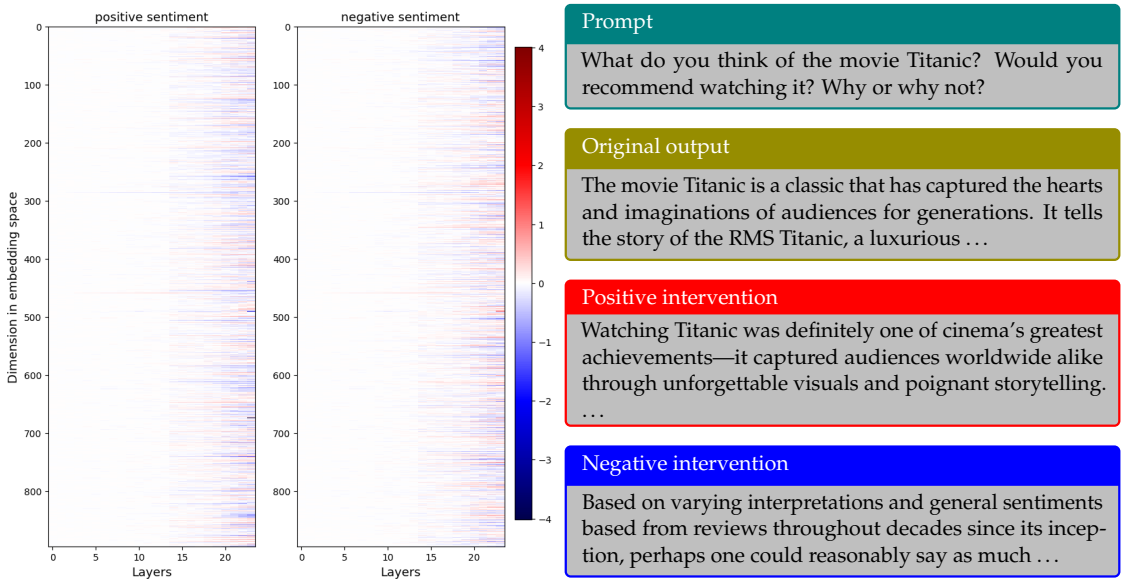


Figure 5: *Left*: Reason strength of every neuron in the residual stream. *Right*: Generating output with intervention on the 'positivity' neurons and the 'negativity' neurons, respectively.

5. Discussion and conclusion

We introduced a new interpretability method based on a formalized notion of reasons. We have shown, both theoretically and empirically, that the method scores well on our desiderata for interpretability.

Limitations and future work As a new method, we established its promise by focusing on breadth rather than depth. Accordingly, future work should continue our experiments in more depth: testing bigger models and harder tasks in experiments 4.1 and 4.3, investigating robustness for more adversarial attacks with a comparison to known defenses in experiment 4.2, mapping the space of possible loss functions (cf. footnote 7), and connecting to more metrics and tasks in the algorithmic fairness literature. Other questions include: Which theoretical guarantees on faithfulness and correctness can one derive under plausible assumptions? The reasons vectors of the neurons are connected by the model's weights into a *network of reasons*: can one abstract from it a high-level and human-understandable network (analogous to [19]) or identify circuits (analogous to [42])?

References

[1] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, Sept. 2020. ISSN 1091-6490. doi: https://doi.org/10.1073/pnas.1907375117.

[2] E. M. Bender and A. Koller. Climbing towards NLU: On meaning, form, and under-

standing in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, 2020.

[3] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[4] D. J. Chalmers. Propositional interpretability in artificial intelligence, 2025. URL https://arxiv.org/abs/2501.15740.

[5] M. J. Colbrook, V. Antun, and A. C. Hansen. The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and smale's 18th problem. *Proceedings of the National Academy of Sciences*, 119(12):e2107151119, 2022. doi: 10.1073/pnas.2107151119. URL https://www.pnas.org/doi/abs/10.1073/pnas.2107151119.

[6] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600. Also see [27].

[7] F. Ding, M. Hardt, J. Miller, and L. Schmidt. Retiring adult: New datasets for fair machine learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6478–6490. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/32e54441e6382a7fbacbbbaf3c450059-Paper.pdf.

[8] J. Donnelly and A. Roegiest. On interpretability and feature representations: an analysis of the sentiment neuron. In *Advances in Information Retrieval, ECIR 2019*, pages 795–802. Springer, 2019. doi: https://doi.org/10.1007/978-3-030-15712-8_55.

[9] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning, 2017. URL https://arxiv.org/abs/1702.08608.

[10] S. Dutta, D. Wei, H. Yueksel, P.-Y. Chen, S. Liu, and K. Varshney. Is there a trade-off between fairness and accuracy? A perspective using mismatched hypothesis testing. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2803–2813. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/dutta20a.html.

[11] G. K. Dziugaite, S. Ben-David, and D. M. Roy. Enforcing interpretability and its statistical impacts: Trade-offs between accuracy and interpretability, 2020. URL https://arxiv.org/abs/2010.13764.

[12] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

[13] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783311. URL https://doi.org/10.1145/2783258.2783311.

[14] J. Fiotto-Kaufman, A. R. Loftus, E. Todd, J. Brinkmann, C. Juang, K. Pal, C. Rager, A. Mueller, S. Marks, A. S. Sharma, F. Lucchetti, M. Ripa, A. Belfki, N. Prakash, S. Multani, C. Brodley, A. Guha, J. Bell, B. Wallace, and D. Bau. Nnsight and ndif: Democratizing access to foundation model internals, 2024. URL https://arxiv.org/abs/2407.14561.

[15] R. Frigg and J. Nguyen. Scientific Representation. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2021 edition, 2021.

[16] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

[17] A. D. Garcez and L. C. Lamb. Neurosymbolic ai: The 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023. doi: https://doi:10.1007/s10462-023-10448-w.

[18] A. Geiger, H. Lu, T. Icard, and C. Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586, 2021. URL https://arxiv.org/abs/2106.02997.

[19] A. Geiger, D. Ibeling, A. Zur, M. Chaudhary, S. Chauhan, J. Huang, A. Arora, Z. Wu, N. Goodman, C. Potts, and T. Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability, 2025. URL https://arxiv.org/abs/2301.04709.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[21] J. Harding. Operationalising representation in natural language processing. *The British Journal for the Philosophy of Science*, forthcoming. URL https://www.journals.uchicago.edu/doi/abs/10.1086/728685?journalCode=bjps.

[22] M. Hardt, E. Price, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

[23] T. Heap, T. Lawson, L. Farnik, and L. Aitchison. Sparse autoencoders can interpret randomly initialized transformers, 2025. URL https://arxiv.org/abs/2501.17727.

[24] D. A. Herrmann and B. A. Levinstein. Standards for belief representations in llms. *Minds and Machines*, 35(5), 2025. doi: https://doi.org/10.1007/s11023-024-09709-6.

[25] J. F. Horty. *Reasons as Defaults*. Oxford University Press, 2012.

[26] N. Howard and M. Schroeder. *The Fundamentals of Reasons*. Oxford University Press, 2024.

[27] R. Huben, H. Cunningham, L. R. Smith, A. Ewart, and L. Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

[28] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

[29] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995.

[30] H. Leitgeb. The additive logic of epistemic reasons: An axiomatic account, 2025. Manuscript under review.

[31] Z. C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, sep 2018. ISSN 0001-0782. doi: 10.1145/3233231. URL https://doi.org/10.1145/3233231.

[32] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[33] S. Marks and M. Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2024. URL https://arxiv.org/abs/2310.06824.

[34] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943. doi: https://doi.org/10.1017/S0140525X00052791.

[35] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020. URL https://arxiv.org/abs/1802.03426.

[36] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in gpt. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

[37] R. Millière and C. Buckner. A philosophical introduction to language models – part ii: The way forward, 2024. URL https://arxiv.org/abs/2405.03207.

[38] C. Molnar. Interpretable machine learning, 2025. URL https://christophm.github.io/interpretable-ml-book.

[39] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, 116(44):22071–22080, 2019. doi: https://doi.org/10.1073/pnas.1900654116.

[40] C. Olah. Interpretability dreams. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/interpretability-dreams/index.html. Informal note.

[41] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. URL https://distill.pub/2017/feature-visualization.

[42] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in.

[43] A. Radford, R. Jozefowicz, and I. Sutskever. Learning to generate reviews and discovering sentiment, 2017. URL https://arxiv.org/abs/1704.01444.

[44] D. Rai, Y. Zhou, S. Feng, A. Saparov, and Z. Yao. A practical review of mechanistic interpretability for transformer-based language models, 2025. URL https://arxiv.org/abs/2407.02646.

[45] L. Sharkey, B. Chughtai, J. Batson, J. Lindsey, J. Wu, L. Bushnaq, N. Goldowsky-Dill, S. Heimersheim, A. Ortega, J. Bloom, S. Biderman, A. Garriga-Alonso, A. Conmy, N. Nanda, J. Rumbelow, M. Wattenberg, N. Schoots, J. Miller, E. J. Michaud, S. Casper, M. Tegmark, W. Saunders, D. Bau, E. Todd, A. Geiger, M. Geva, J. Hoogland, D. Murfet, and T. McGrath. Open problems in mechanistic interpretability, 2025. URL https://arxiv.org/abs/2501.16496.

[46] P. Smolensky. On the proper treatment of connectionism. *Behavioral and brain sciences*, 11(1):1–74, 1988. doi: https://doi.org/10.1017/S0140525X00052791.

[47] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

[48] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[49] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[50] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

[51] K. R. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *NeurIPS ML Safety Workshop*, 2022. URL https://openreview.net/forum?id=rvi3Wa768B-.

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

[53] F. Zhang and N. Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

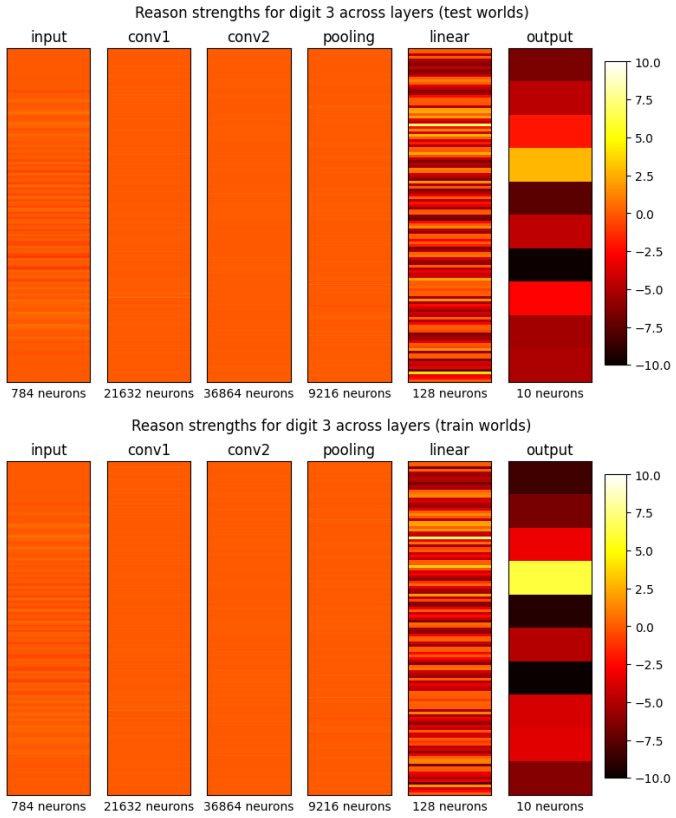


Figure 6: *Top*: Computing reasons strengths using worlds from the test dataset, so the model has not seen them. *Bottom*: Using the training dataset instead.

A. Experiment 1: Interpreting a classic

Architecture and training details The LeNet architecture is as follows:

- a convolutional layer of dimension (in-channel: 1, out-channel: 32, kernel: 3×3),
- a convolutional layer of dimension (in-channel: 32, out-channel: 64, kernel: 3×3),
- a max-pooling layer followed by dropout (25%) and flattening,
- a linear layer to 128 neurons followed by ReLU and dropout (50%),
- a linear layer to the 10 output neurons (for the 10 digits) followed by log-softmax.

We train with a batch size of 64 and 20 epochs, using the AdamW optimizer [32] with the default learning rate. We load the MNIST dataset via `torchvision.datasets`, which contains a balanced 60k images in the training set and 10k images in the test set.

Test vs training worlds Figure 6 shows the difference between using worlds from the test dataset (as done in the main text, so the model has not seen them) and from the training dataset. Using training worlds, the reasons strengths get higher values than with test worlds. After all, the model has seen these worlds during training. However, there is no qualitative difference.

Stastical robustness Figure 7 shows the statistical robustness of the reasons strengths. We compute the reason strength for three different instances of the LeNet architecture. They all were trained on MNIST data and achieved accuracies $> 99\%$ but used different random seeds. Which neurons in, say, the hidden layer strongly speak for digit 3 and which against can vary between the models. So there is no implicit bias in the LeNet architecture that would force a given neuron in the linear layer to play a specific role with respect to digit classification. However, qualitatively speaking, all models show the same behavior in the output layer: that neuron d strongly speaks for digit d and the others strongly against digit d. But, again, the quantity especially of the negative strength can vary.

Groups of neurons We consider the layers of the trained LeNet model as groups of neurons. We compute how the layers update an initially uniform prior probability distribution over the possible worlds. Specifically, this is done as follows. We again sample 1024 input-label pairs from the MNIST test dataset to form the set of worlds W. The prior b is the uniform distribution on W. We start with layer 0, the input layer. For each neuron in this layer, we compute its reasons vector as before. To aggregate all these reasons vectors, according to the reasons theory, we sum all these vectors, to obtain the reasons vector r_0 of layer 0. To update the prior probability b with layer 0, we compute $b_0 := b * r_0$.[10] Similarly, we update b_0 with layer 1, the first convolutional layer, to obtain b_2, and so on for the other layers. The resulting probability distributions are shown in figure 8. Even though we start with the uniform measure and use a balanced dataset, the input layer introduces some bias among the worlds—and this bias is amplified by later layers.

Different dimensionality reduction In operationalizing correctness, we used PCA as a dimensionality reduction technique. Other popular such techniques are *t-SNE* [49] and *UMAP* [35]. Figure 9 shows the results of the correctness experiment when using those dimensionality reductions instead. Qualitatively, the results are the same: clear monochromatic clusters form for the linear layer, but things are more chaotic in the early convolutional layer. For the convolutional layer, the more sophisticated dimensionality reduction methods t-SNE and UMAP can identify more monochromatic clusters compared to PCA, but they are still more chaotic compared to the clear clusters that these methods identify for the linear layer.

B. Experiment 2: Improving reasons

Training reasons After initializing a LeNet model and making a copy, we train the original model with the sum of the usual loss (cross entropy) and the doxastic reasons loss (equation 3)—adding both summands with equal weight—, while we train the comparison model with only the usual loss. We use the same batches for both models, taken from the MNIST train dataset, with a batch size of 2048 and 20 epochs. We again use the AdamW optimizer [32].

The achieved accuracy with reasons training is 99.12% on the test set, while without reasons training it is 99.11%. Figure 10 shows how the two models compare regarding faithfulness, in the more difficult 'neg2pos' version. The success rates now have less

[10] For numerical stability, we first normalize r_0 before computing $b * r_0$.

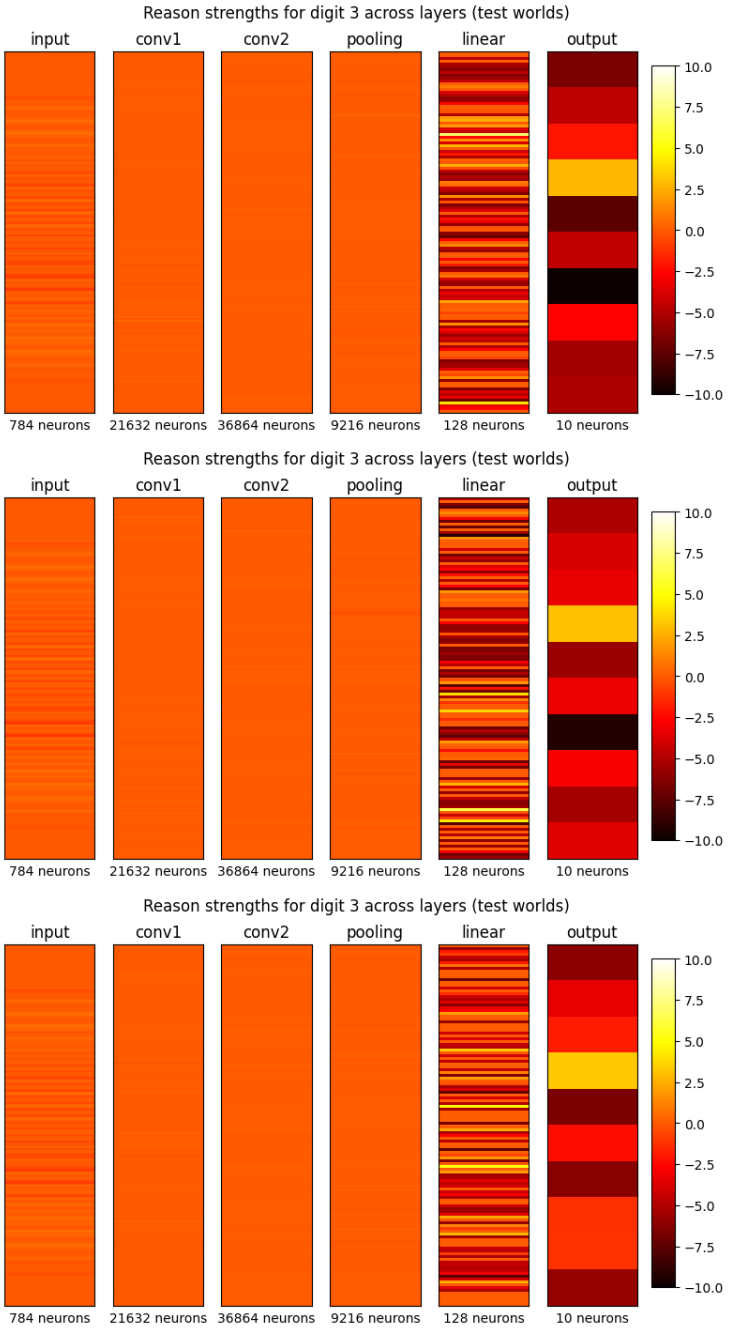


Figure 7: The reason strength of three different models (one plot for one model): all are instances of the LeNet architecture trained on MNIST data but with different seeds.

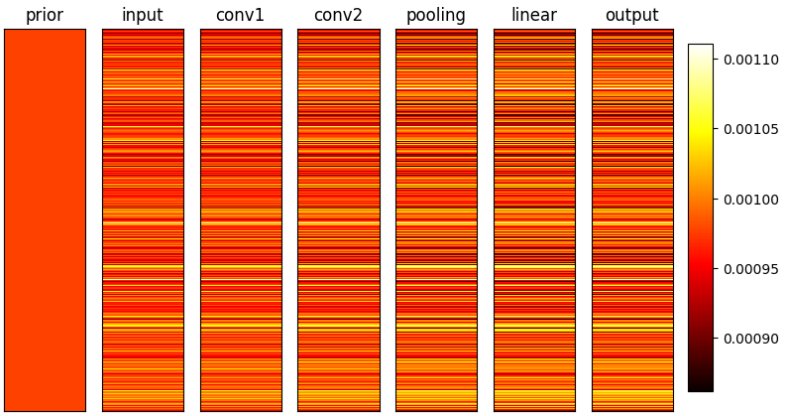


Figure 8: Updating a uniform prior distribution over possible worlds layer by layer.

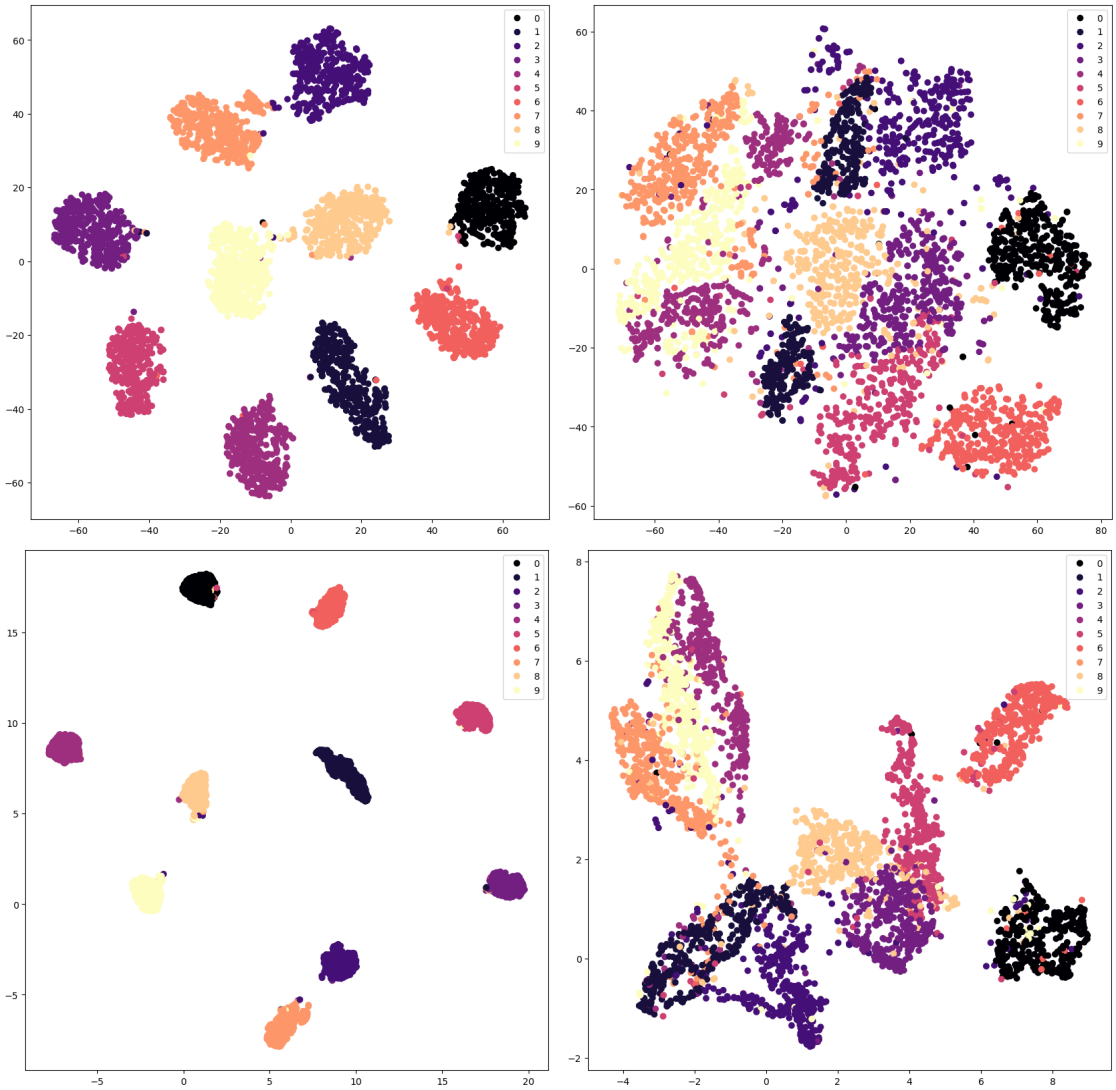


Figure 9: *Top left*: TSNE for linear layer. *Top right*: TSNE for first convolutional layer. *Bottom left*: UMAP for linear layer. *Bottom right*: UMAP for first convolutional layer.

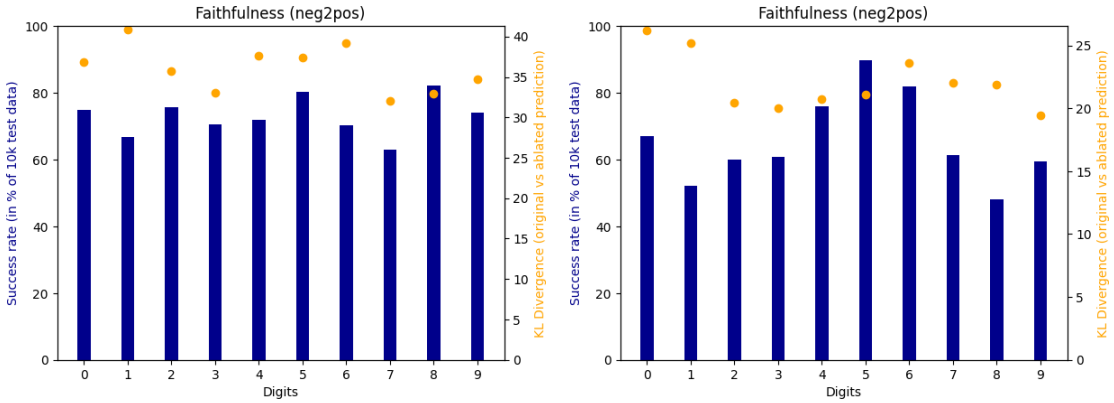


Figure 10: *Left*: Faithfulness with doxastic reasons training. *Right*: Faithfulness without reasons training.

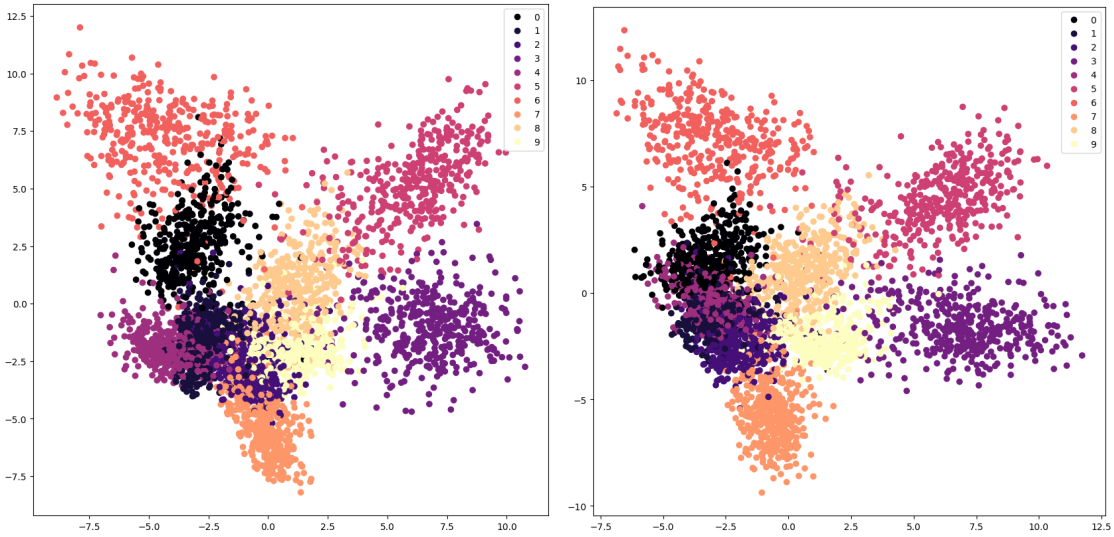


Figure 11: *Left*: Correctness with doxastic reasons training. *Right*: Correctness without reasons training.

variance and are all reliably above 60%, and the KL divergences have almost doubled. Figure 11 shows how they compare regarding correctness. The clusters only marginally became more separated.

Alternative loss function Using the same notation as for the doxastic reasons loss defined in equation 3, we define here an alternative loss $L'(w, x, y)$, which we call the *elementary reasons loss*. Given weights w and a batch $x = (x_1, \ldots, x_N)$ of inputs with corresponding labels $y = (y_1, \ldots, y_N)$, again let r_d be the reasons vector of the d-th output neuron, and let $A_d = \{(x, y) \in W : y = l_d\}$. Another way to formalize that r_d is a 'good' reason for A_d is by saying that it is similar to the elementary reason el_{A_d}, which is—in a sense—the canonical reason for A_d. We measure similarity by cosine similarity CosSim, which takes values in the interval $[-1, 1]$, where $+1$ means most similar (i.e., codirectional). Hence

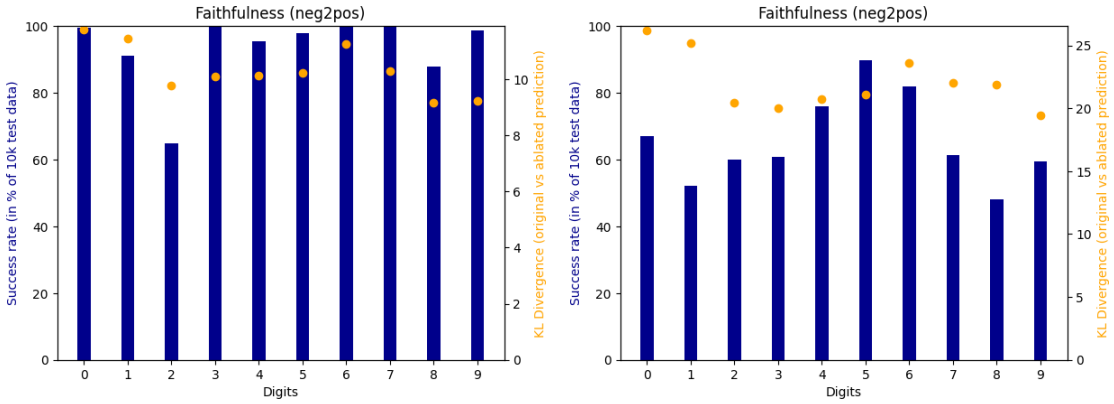


Figure 12: *Left*: Faithfulness with elementary reasons training. *Right*: Faithfulness without reasons training.

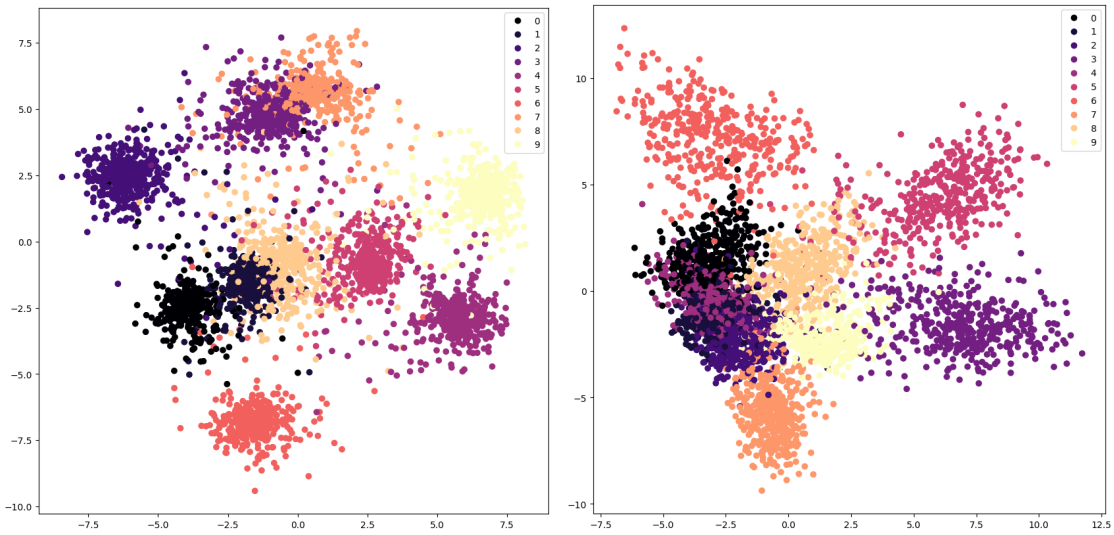


Figure 13: *Left*: Correctness with elementary reasons training. *Right*: Correctness without reasons training.

$1 - \text{CosSim}$ takes values in $[0, 2]$ and we want to minimize it. So we define:

$$L'(w, x, y) := \sum_{d=1}^{C} 1 - \text{CosSim}\left(r_d, \text{el}_{A_d}\right). \tag{5}$$

When training with this loss, the achieved accuracy with reasons training is 99.05% on the test set, while without reasons training it is 99.11%. Figure 12 shows how the two models compare regarding faithfulness, in the more difficult 'neg2pos' version. The model now achieves, for all digits except 2, a success rate of well above 90% (compared to around 60% without reasons training). Figure 13 shows how they compare regarding correctness. We get yet clearer monochromatic clusters.

Robustness Figure 14 shows the effectiveness of an FGSM attack on the reasons trained model compared to the comparison model that has not been trained for reasons. On the left, the reasons training is done with the doxastic reasons loss, and on the right

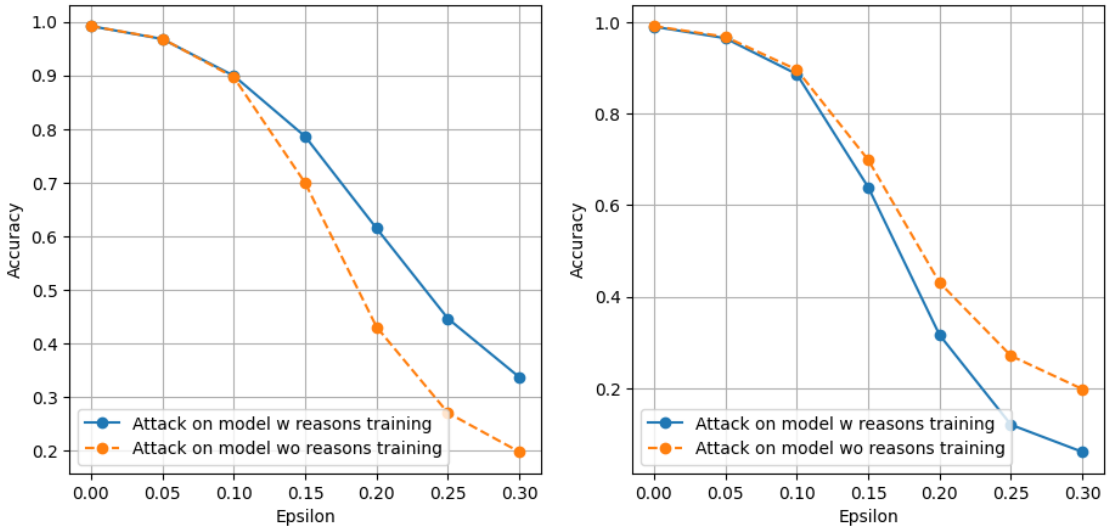


Figure 14: *Left*: Robustness with and without doxastic reasons loss. *Right*: Robustness with and without elementary reasons loss.

with the elementary reasons loss. Curiously, even though the elementary loss is better at improving the model's reasons structure (as seen in the preceding paragraph) compared to the doxastic loss, it is the doxastic loss which yields more robustness to adversarial attacks, while the elementary loss does not. Thus, there is a nontrivial relationship between faithfulness, correctness, and robustness.

Training fairness via reasons difference The task is to predict, based on certain information about a person, whether they earn more than a threshold amount. The two threshold amounts that we test are 25k and 50k. We use the modernized *Adult* dataset due to [7], available via the `folktables` package.[11] The data can be chosen to come from different US states and years; here we choose 2018 Alabama. The features in the dataset are the following [7, appendix B]:

- AGEP: Age
- COW: Class of worker
- SCHL: Educational attainment
- MAR: Marital status
- OCCP: Occupation
- POBP: Place of birth
- RELP: Relationship
- WKHP: Usual hours worked per week past 12 months
- SEX: Sex (1: Male, 2: Female)

[11] Available at https://github.com/socialfoundations/folktables under the MIT licese.

- RAC1P: Recoded detailed race code

In this experiment, we will treat sex as the protected attribute. The dataset has 22,268 entries. The percentage of the privileged group (male) is 52.2%. In the 25k version, the percentage of positive classification is 62.6%; and in the 50k version, it is 31.1%.

We first train the following models on this task (without any reasons training) to get an understanding of the baseline performance. The first group are the following standard models (available in `scikit-learn`):

1. Logistic regression
2. Random Forest Classifier
3. C-Support Vector Classifier

The second group is the following MLPs. Each has 10 input neurons (for the 10 features) and 1 output neuron (indicating positive or negative classification) and uses ReLU as activation function.

1. MLP_s ('small'): One hidden layer of size 100 followed by a second hidden layer of size 50.
2. MLP_v ('vanilla'): Four hidden layers each of size 128.
3. MLP_dn ('dropnorm'): Also four hidden layers each of size 128, but with 20%-dropout and batch norm [28].

We test all combinations of the following hyperparameters:

1. Learning rates: 1e-4, 1e-3, 1e-2.
2. Number of epochs: 5, 10, 20.

We train with binary cross entropy loss (with logits) using AdamW [32]. The training-test split is 20% test data, and we scale the data using `scikit-learn`'s StandardScaler. Since the dataset is unbalanced, we report the F1 scores (rather than accuracy) achieved by each model in figure 15. Except for some outliers on the smallest learning rate, all models achieve a very similar performance.

Based on this, we choose, for further reasons training, the MLP_dn model with a learning rate of 1e-3 and 20 epochs: it achieves a performance comparable to the other models, but it has higher numerical stability due to batch normalization, which is useful for computing reason strengths (since this requires taking exponentials of neuron activations).

After initializing the MLP_dn model and making a copy, we train the original model with the sum of the usual loss (binary cross entropy) and the reasons difference loss (equation 4)—adding both summands with equal weight—, while we train the comparison model with only the usual loss. We use the same batches for both models, with a batch size of 1024, again using AdamW [32]. We do this for both the 25k task and the 50k task, and we repeat each 100 times. Figure 16 shows the results.

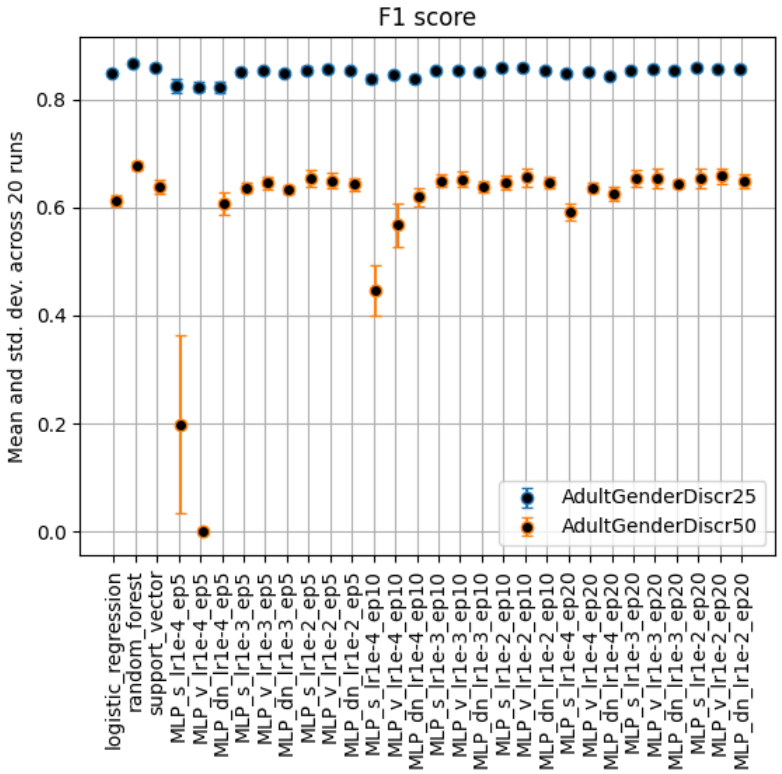


Figure 15: F1 scores for different models for the two fairness tasks.

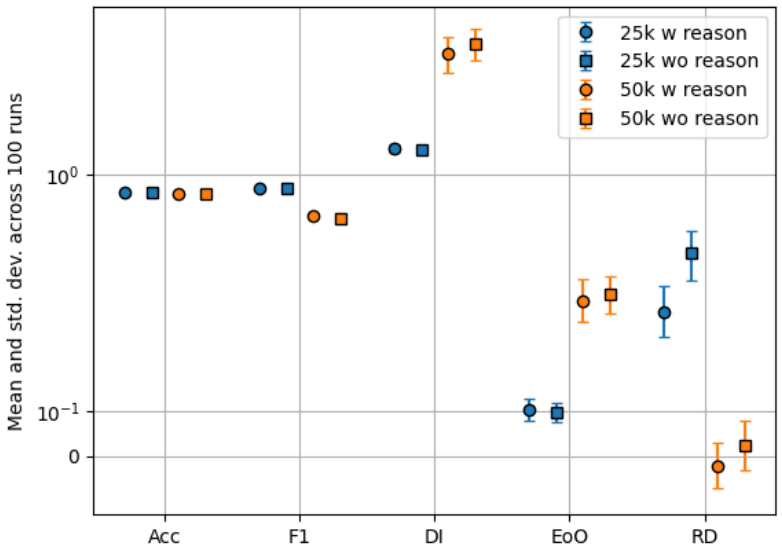


Figure 16: Improving fairness through reasons training. The metrics are: accuracy (Acc), F1 score (F1), disparate impact (DI), equality of opportunity (EoO), reasons difference (RD). Note the logarithmic scale.

C. Experiment 3: Reasons in LLMs

Setup We use the `nnsight` library [14] to load the model *Qwen2.5-0.5B-Instruct* and to access its residual stream.[12] We use the popular *Stanford Sentiment Treebank* dataset SST2 [47].[13]

Prompt-template world construction Given a sentence s from the SST2 training dataset, we form the following few-shot prompt:

```
Input: This was a truly amazing movie.
Classify the sentiment of the message: positive

Input: One of the worst films I saw lately.
Classify the sentiment of the message: negative

Input: {s}
Classify the sentiment of the message:
```

Prompt-template sentiment classification Given a sentence s from the SST2 validation dataset, we form the following few-shot prompt:

```
You have to classify sentences as either 'positive' or 'negative'.

Input: This was a thought-provoking movie
The sentiment of the message is:
a) positive
b) negative
Answer: a)

Input: rather mixed acting with a mediocre story line
The sentiment of the message is:
a) positive
b) negative
Answer: b)

Input: feel-good story with rich characters
The sentiment of the message is:
a) positive
b) negative
Answer: a)

Input: {s}
The sentiment of the message is:
```

[12] Available at https://nnsight.net/ under the MIT license.
[13] Available at https://huggingface.co/datasets/stanfordnlp/sst2.

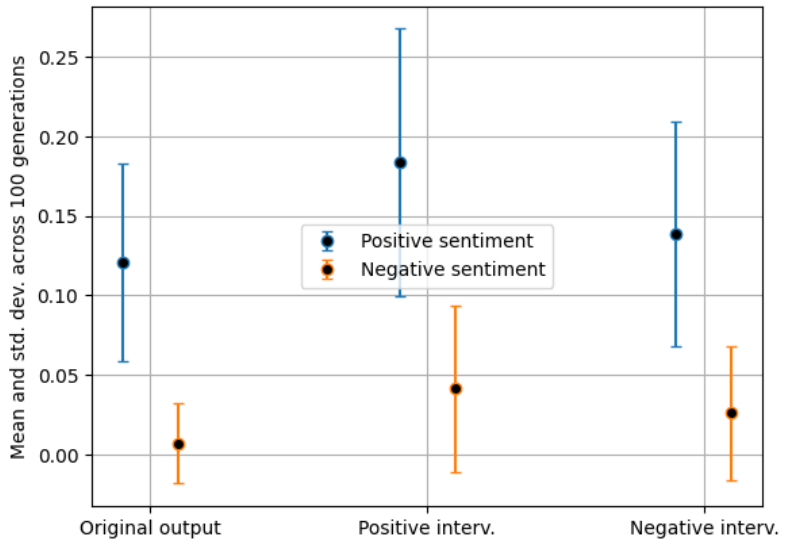


Figure 17: Sentiment statistics when generating 100 responses to the prompt 'What do you think of the movie Titanic? Would you recommend watching it? Why or why not?'.

```
a) positive
b) negative
Answer:
```

Sentiment statistics for generation We consider the prompt 'What do you think of the movie Titanic? Would you recommend watching it? Why or why not?'. We generate output with the model first as is ('Original output'), then with a positive intervention ('Positive interv.') and then with a negative intervention ('Negative interv.'). The positive (resp., negative) intervention sets the 5 neurons speaking most for positivity (resp., negativity) to $a' = 2m$ (resp., $a' = 20m$) and the 5 neurons speaking most against positivity (resp., negativity) to $a' = m$ (resp., $a' = m$). To test for statistical variance, we generate these outputs 100 times. For each output, we measure, using the `NLTK` SentimentIntensityAnalyzer, both its positivity score and its negativity score.[14] Figure 17 shows the mean and standard deviation of these scores. In figure 5 (right) in the main text, we saw *qualitatively* that the generated output changes according to the positive or negative intervention. But now we can also see *quantitatively* that the positive intervention generates outputs with higher positivity scores than the original model, and the negative intervention generates outputs with higher negativity scores than the original model.

D. Compute

All experiments are performed using a regular laptop (CPU with 16 GB memory). The majority of experiments take less than 20 minutes, with a few experiments taking a couple of hours. Initial experiments included approximately 10 days of compute time on the aforementioned setup.

[14] Available at https://www.nltk.org/index.html under the Apache-2.0 license.