Causal Abstractions of Neural Networks

Atticus Geiger[*], **Hanson Lu**[*], **Thomas Icard, and Christopher Potts**
Stanford
Stanford, CA 94305-2150
{atticusg, hansonlu, icard, cgpotts}@stanford.edu

Abstract

Structural analysis methods (e.g., probing and feature attribution) are increasingly important tools for neural network analysis. We propose a new structural analysis method grounded in a formal theory of *causal abstraction* that provides rich characterizations of model-internal representations and their roles in input/output behavior. In this method, neural representations are aligned with variables in interpretable causal models, and then *interchange interventions* are used to experimentally verify that the neural representations have the causal properties of their aligned variables. We apply this method in a case study to analyze neural models trained on Multiply Quantified Natural Language Inference (MQNLI) corpus, a highly complex NLI dataset that was constructed with a tree-structured natural logic causal model. We discover that a BERT-based model with state-of-the-art performance successfully realizes parts of the natural logic model's causal structure, whereas a simpler baseline model fails to show any such structure, demonstrating that BERT representations encode the compositional structure of MQNLI.

1 Introduction

Explainability and interpretability have long been central issues for neural networks, and they have taken on renewed importance as such models are now ubiquitous in research and technology. Recent structural evaluation methods seek to reveal the internal structure of these "black box" models. Structural methods include probes, attributions (feature importance methods), and interventions (manipulations of model-internal states). These methods can complement standard behavioral techniques (e.g., performance on gold evaluation sets), and they can yield insights into how and why models make the predictions they do. However, these tools have their limitations, and it has often been assumed that more ambitious and systematic causal analysis of such models is beyond reach.

Although there is a sense in which neural networks are "black boxes", they have the virtue of being completely closed and controlled systems. This means that standard empirical challenges of causal inference due to lack of observability simply do not arise. The challenge is rather to identify high-level causal regularities that *abstract away* from irrelevant (but arbitrarily observable and manipulable) low-level details. Our contribution in this paper is to show that this challenge can be met. Drawing on recent innovations in the formal theory of causal abstraction [1, 2, 5, 22], we offer a methodology for meaningful causal explanations of neural network behavior.

Our methodology *causal abstraction analysis*[2] consists of three stages. (1) Formulate a hypothesis by defining a causal model that might explain network behavior. Candidate causal models can be naturally adapted from theoretical and empirical modeling work in linguistics and cognitive sciences. (2) Search for an alignment between neural representations in the network and variables in the

[*] equal contribution

[2] We provide tools for causal abstraction analysis at `http://github.com/hansonhl/antra` and the code base for this paper at `http://github.com/atticusg/Interchange`

35th Conference on Neural Information Processing Systems (NeurIPS 2021)

high-level causal model. (3) Verify experimentally that the neural representations have the same causal properties as their aligned high-level variables using the *interchange intervention* method of Geiger et al. [11].

As a case study, we apply this methodology to LSTM-based and BERT-based natural language inference (NLI) models trained on the logically complex Multiply Quantified NLI (MQNLI) dataset of Geiger et al. [10]. This challenging dataset was constructed with a tree-structured natural logic causal model [17, 29, 14]. Our BERT-based model has the structure of a standard NLI classifier, and yet it is able to perform well on MQNLI (88%), a result Geiger et al. achieved only with highly customized task-specific models. By contrast, our LSTM-based model is much less successful (46%).

The obvious scientific question in this case study is what drives the success of the BERT-based model on this challenging task. To answer this we employ our methodology. (1) We formulate hypotheses by defining simplified variants of the natural logic causal model. (2) We search over potential alignments between neural representations in BERT and variables in our high-level causal models. (3) We perform interchange interventions on the BERT model for each alignment. We find that our BERT model partially realizes the causal structure of the natural logic causal model; crucially, the LSTM model does not. High-level causal explanation for system behavior is often considered a gold standard for interpretability, one that may be thought quixotic for complex neural models [16]. The point of our case study is to show that this high standard can be achieved.

We conclude by comparing our methodology to probing and the attribution method of integrated gradients [27]. We argue probing is unable to provide a causal characterization of models. We show formally that attribution methods do measure causal properties, and in that way they are similar to the tool of interchange interventions. However, our methodology of causal abstraction analysis provides a framework for systematically measuring and aggregating such causal properties in order to evaluate a precise hypothesis about abstract causal structure.

2 Related Work

Probes Probes are generally supervised models trained on the internal representations of networks with the goal of determining what those internal representations encode [7, 13, 20, 28]. Probes are fundamentally unable to directly measure causal properties of neural representations, and Ravichander et al. [21], Elazar et al. [9], and Geiger et al. [11] have argued that probes are limited in their ability to provide even indirect evidence of causal properties.

We now present an analytic example in which probing identifies seemingly crucial information in representations that have no causal impact on behavior. We assume the structure of the simple addition network N_+ in Figure 1. For our embedding, we simply map every integer i in $\mathbb{N}_9$ to the 1-dimensional vector $[i]$. The weight matrices are

$$W_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad W_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad W_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

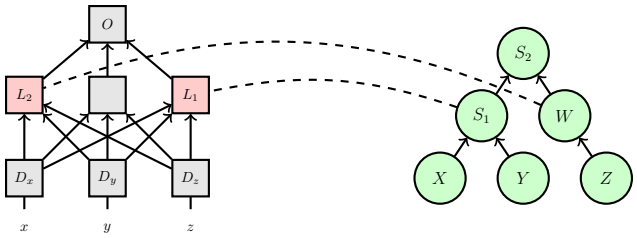
The output for an input sequence $\mathbf{x} = (i, j, k)$ is given by $(\mathbf{x}W_1; \mathbf{x}W_2; \mathbf{x}W_3)\,\mathbf{w}$.

In this network, $\mathbf{x}W_1$ perfectly encodes $i + j$, and $\mathbf{x}W_3$ perfectly encodes k. Thus, the identity model probe will be perfect in probing those representations for this information. However, neither representation plays a causal role in the network behavior; only $\mathbf{x}W_2$ contributes to the output.

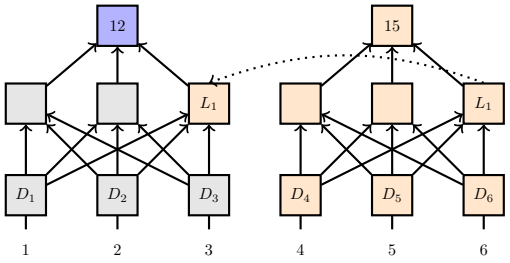
Attribution Methods Attribution methods aim to quantify the degree to which a network representation contributes to the output prediction of the model, for a specific example or set of examples [3, 24, 26, 27, 32]. In contrast to probing, the well known integrated gradients method (IG) can be given an unambiguous causal interpretation. Following [27] we define the vector $IG(\mathbf{x})$, for an input $\mathbf{x}$ relative to a baseline $\mathbf{b}$, to have ith component $IG_i(\mathbf{x})$ given by the expression on the left:

$$(x_i - b_i) \cdot \int_{\alpha=0}^{1} \frac{\partial F(\alpha\mathbf{x} + (1-\alpha)\mathbf{b})}{\partial x_i} d\alpha \quad = \quad (x_i - b_i) \cdot \int_{\alpha=0}^{1} \lim_{\epsilon \to 0} \frac{F(\mathbf{x}^{\alpha,\epsilon}) - F(\mathbf{x}^{\alpha})}{\epsilon} d\alpha$$

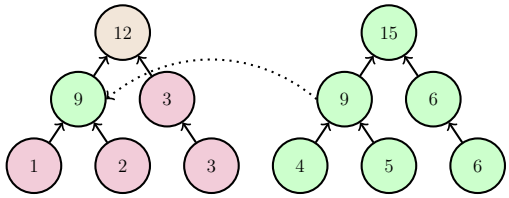
Abbreviating the weighted average $\alpha\mathbf{x} + (1-\alpha)\mathbf{b}$ by $\mathbf{x}^{\alpha}$, letting $\mathbf{x}^{\alpha,\epsilon}$ be the vector that differs from $\mathbf{x}^{\alpha}$ in that the ith coordinate is increased by ϵ, and then expanding the definition of partial derivative, this can be written in the form given on the right. The difference $F(\mathbf{x}^{\alpha,\epsilon}) - F(\mathbf{x}^{\alpha})$ is known in the



(a) The causal model C_+ (right) that first computes $S_1 = X + Y$ and $W = Z$, before computing the final output $S_2 = W + S_1$ aligned with the neural network N_+ (left) with L_1 highlighted as the hypothesized location encoding $S_1 = X + Y$ and L_2 as the location encoding $W = Z$.



(b) Low-level neural network interchange intervention. The network processes two different input sequences. The neural representation created at L_1 for input sequence $(1, 2, 3)$ is replaced by the corresponding representation created for input sequence $(4, 5, 6)$.



(c) High-level symbolic computation interchange intervention. The computation processes two different input sequences. The sum at S_1 for input sequence $(1, 2, 3)$ is replaced by the corresponding representation created for input sequence $(4, 5, 6)$.

Figure 1: Our motivating example where we hypothesis that a symbolic computation C_+ is a causal abstraction of a neural network N_+ under a particular alignment (top). We can experimentally confirm this hypothesis by conducting an interchange intervention on both the network and the computation with every pair of inputs and evaluating whether the intervened network and intervened computation have the same counterfactual output behavior. We schematically depict an interchange intervention on the network N_+ (bottom left) and the computation C_+ (bottom right) with the base input $(1, 2, 3)$ and the source input $(4, 5, 6)$. Observe that the output of the intervened neural network matches the output of the intervened symbolic computation, so we have success for this pair of inputs.

causal literature as the (individual) *causal effect* on the output (e.g., [15]) of increasing neuron i by ϵ relative to the fixed input $\mathbf{x}^{\alpha}$. So, essentially, $IG_i(\mathbf{x})$ is measuring the average "limiting" causal effect of increasing neuron i along the straight line from the baseline vector to the input vector $\mathbf{x}$, weighted by the difference at i between input and baseline. More recently, Chattopadhyay et al. [6] develop an attribution method that explicitly treats neural models as structured causal models and directly computes the individual causal effect of a feature to determine its attribution.

Attribution methods can measure causal properties, and, in that way, they are similar to the tool of interchange interventions. However, our methodology of causal abstraction analysis provides a framework for systematically measuring and aggregating such causal properties in order to evaluate a precise hypothesis about abstract causal structure.

Causal Abstraction Our goal is to evaluate whether the internal structure of a neural network realizes an abstract causal process. To concretize this, we turn to formal, broadly interventionist theories of causality [25, 19], in which causal processes are characterized by effects of interventions, and theories of abstraction [1, 2, 5, 22] where relationships between two causal processes are determined by the presence of systematic correspondences between the effects of interventions.

The notion of abstraction that we employ here is a relatively simple one called *constructive abstraction* [1]. Informally, a high-level model is a constructive abstraction of a low-level model if there is a way to partition the variables in the low-level model where each high-level variable can be assigned to a low-level partition cell, such that there is a systematic correspondence between interventions on the low-level partition cells and interventions on the high-level variables.

There are two properties of constructive abstraction that make it ideal for neural network analysis. First, the information content of partition cells of low-level variables can be determined by the high-level variables that they correspond to. For neural networks, the partition cells of low-level variables are sets of neurons, and our method supports reasoning at the level of vector representations (sets of neurons). Second, the causal dependencies between partitions of low-level variables are not necessarily preserved as causal dependencies between the high-level variables corresponding to these partitions. For example, the low-level model might be a fully connected neural network, whereas the high-level model might have much sparser connections. For neural network analysis, this means we can find causal abstractions that have far simpler causal structures than the underlying neural networks. We provide an example in the next section.

3 Causal Abstraction Analysis of Neural Networks

We now describe our methodology in more detail, illustrating the relevant concepts with an example of a neural network performing basic arithmetic. Specifically, suppose that we have a neural network N_+ that takes in three vector representations D_x, D_y, D_z representing the integers x, y, and z, and outputs the sum of the three inputs: $N_+(D_x, D_y, D_z) = x + y + z$. We seek an informative causal explanation of this network's behavior.

Formulating a Hypothesis A human performing this task might follow an algorithm in which they add together the first two numbers and then add that sum to the third number. We can hypothesize that the behavior of N_+ is explained by this symbolic computation. Specifically, the network combines D_x and D_y to create an internal representation at some location L_1 encoding $x + y$; it encodes z at some location L_2; and L_1 and L_2 are composed to encode $a + z$ at the location of the output representation. This hypothesis is given schematically in Figure 1a.

Following our methodology, we first define the causal model C_+ in Figure 1a. Our informal hypothesis that a neural network's behavior is explained by a simple algorithm can then be restated more formally: C_+ is a constructive abstraction of the neural network N_+.

Alignment Search Now that we have hypothesized that the causal model C_+ is a causal abstraction of the network N_+, the next step is to align the neural representations in N_+ with the variables in C_+. The input embeddings D_x, D_y, and D_z must be aligned with the input variables X, Y, and Z and the output neuron O must be aligned with the output variable S_2. That leaves the intermediate variables S_1 and W to be aligned with neural representations at some undetermined locations L_1 and L_2. If this were an actual experiment (see below), we would perform an *alignment search* to consider many possible values for L_1 and L_2. Each alignment is a hypothesis about where the network N_+ stores and uses the values of S_1 and W. For the example, we assume the alignment in Figure 1a.

Interchange Interventions Finally, for a given alignment, we experimentally determine whether the neural representations at L_1 and L_2 have the same causal properties as S_1 and W. The basic experimental technique is an *interchange intervention*, in which a neural representation created during prediction on a "base" input is interchanged with the representation created for a "source" input [11]. We now show informally that this method can be used to prove that the causal model C_+ is a constructive abstraction of the neural network N_+ (Appendix G has formal details).

We first intervene on the causal model. Consider two inputs $\mathbf{a}, \mathbf{a}' \in (\mathbb{N}_9)^3$ where $\mathbb{N}_9$ is the set of integers 0–9. Let $\mathbf{a} = (x, y, z)$ be the base input and $\mathbf{a}' = (x', y', z')$ be the source input. Define

$$C_+^{S_1 \leftarrow \mathbf{a}'}(\mathbf{a}) = x' + y' + z \tag{1}$$

to be the output provided by C_+ when S_1, the variable representing the intermediate sum, is intervened on and set to the value $x' + y'$. Thus, for example, if the base input is $C_+(1, 2, 3) = 6$, and the source input is $\mathbf{a}' = (4, 5, 6)$, then $C_+^{S_1 \leftarrow \mathbf{a}'}(1, 2, 3) = 4 + 5 + 3 = 12$. This process is depicted in Figure 1c.

Next, we intervene on the neural network N_+. Let $\mathbf{D}$ be an embedding space that provides unique representations for $\mathbb{N}_9$, and consider two inputs $D = (D_x, D_y, D_z)$ and $D' = (D_{x'}, D_{y'}, D_{z'})$, where all D_i and $D_{i'}$ are drawn from $\mathbf{D}$. In parallel with (1), define

$$N_+^{L_1 \leftarrow D'}(D) \tag{2}$$

to be the output provided by N_+ processing the input D when the representation at location L_1 is replaced with the representation at location L_1 created when N_+ is processing the input D'. This process is depicted in Figure 1b.

With these two definitions, we can define what it means to test the hypothesis that N_+ computes $x + y$ at position L_1. Where $D_{\mathbf{a}}$ is an embedding for $\mathbf{a}$ and $D_{\mathbf{a}'}$ is an embedding for $\mathbf{a}'$, we test:

$$C_+^{S_1 \leftarrow \mathbf{a}'}(\mathbf{a}) = N_+^{L_1 \leftarrow D_{\mathbf{a}'}}(D_{\mathbf{a}}) \tag{3}$$

If this equality holds for all source and base inputs $\mathbf{a}$ and $\mathbf{a}'$, then we can conclude that, for every intervention on S_1, there is an equivalent intervention on L_1. If we can establish a corresponding claim for W and L_2, then we have shown that C_+ is a constructive abstraction of N_+, since the inputs' relationships are established by our embedding and there are no other interventions on C_+ to test.

Analysis Suppose that all of our intervention experiments verify our hypothesis that C_+ is a constructive abstraction of N_+ with variables S_1 and W aligned to neural representations at L_1 and L_2. This explains network behavior by resolving two crucial questions.

First, we learn what information is encoded in the representations L_1 and L_2. Neural representations encode the values of the high-level variables they are aligned with. The location L_1 encodes the variable S_1 and the location L_2 encodes the variable W. This is similar to what probing achieves. However, our method is crucially different from probing. In probing, information content is established through purely correlational properties, meaning a neural representation with no causal role in network behavior can be successfully probed, as we showed in Section 2. In causal abstraction analysis, information content is established through purely causal properties, ensuring that the neural representation is actually implicated in model behavior.

Second, we learn what causal role L_1 and L_2 play in network behavior. Neural representations play a parallel causal role to their aligned high-level variables. At the location L_1, D_x and D_y are composed to form a neural representation with content $x + y$ that is then composed with L_2 to create an output. The fact that S_1 doesn't depend on z tells us that while L_1 depends on D_z and representations at L_1 may even correlate with z, the information about z is not causally represented at L_1. At the location L_2, the value of z is simply repeated and then composed with L_1 to create a final output.

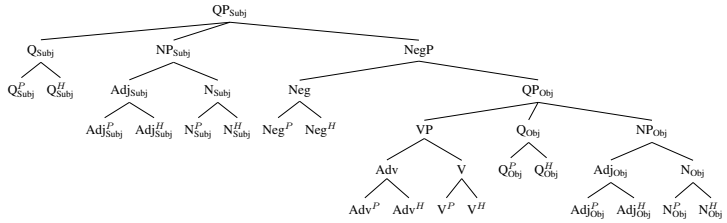
Our method assigns causally impactful information content, but also identifies the abstract causal structure along which representations are composed. It thus encompasses and improves on both correlational (probing) and attribution methods.

4 The Natural Language Inference Task and Models

Multiply Quantified NLI Dataset The Multiply Quantified NLI (MQNLI) dataset of Geiger et al. [10] contains templatically generated English-language NLI examples that involve very complex interactions between quantifiers, negation, and modifiers. We provide a few examples in Figure 2b; the empty-string symbol ε ensures perfect alignments at the token level both between premises and hypotheses and across all examples.

The MQNLI examples are labeled using an algorithmic implementation of the natural logic of MacCartney and Manning [18] over tree structures, and MQNLI has train/dev/test splits that vary in their difficulty. In the hardest setting, the train set is provably the minimal set of examples required to ensure that the dev and test sets can be perfectly solved by a simple symbolic model; in the easier settings, the train set redundantly encodes necessary information, which might allow a model to perform perfectly in assessment by memorization despite not having found a truly general solution. For a fuller review of the dataset, see Appendix A.

MQNLI is a fitting benchmark given our goals for a few reasons. First, we can focus on the hardest splits that can be generated, which will stress-test our NLI architectures in a standard behavioral way. Second, the MQNLI labeling algorithm itself suggests an appropriate causal model of the data-generating process. Figure 2a summarizes this model in tree form, and it is presented in full detail in Geiger et al. [10]. This allows us to rigorously assess whether a neural network has learned to implement variants of this causal model. The complexity of the MQNLI examples creates many opportunities to do this in linguistically interesting ways.



(a) The causal structure of the high-level natural logic causal model C_{NatLog} that performs inference on MQNLI. The superscripts P and H stand for 'premise' and 'hypothesis' and the subscripts 'Subj' and 'Obj' stand for 'Subject' and 'Object'. The node labels are used to explain the experimental results in Section 5

ε every ε baker ε ε ε eats ε no ε bread
contradiction
ε no angry baker ε ε ε eats ε no ε bread

ε every silly professor ε ε ε sells not every ε book
neutral
ε every silly professor ε ε ε sells not every ε chair

not every sad baker ε ε fairly admits not every odd idea
entailment
ε some ε baker does not ε admits ε no ε idea

(b) MQNLI examples. The ε token serves as padding (but still attended to by the model) and ensures a perfect alignment between both premises and hypotheses and across all examples. It is semantically an identity element.

Model	Train	Dev	Test
CBoW	88.04	54.18	53.99
TreeNN	67.01	54.01	53.73
CompTreeNN	99.65	80.17	80.21
BiLSTM	99.42	46.41	46.32
BERT	99.99	**88.25**	**88.50**

(c) MQNLI results. The first three models are from Geiger et al. 10, where the CompTreeNN is a task-specific model not suitable for general NLI and functions as an idealized upperbound. Our results show that BERT-based models can surpass this without such alignments.

Figure 2: The natural logic causal model (top), MQNLI examples (left) and MQNLI results (right).

Models We evaluated two models on MQNLI: a randomly initialized multilayered Bidirectional LSTM (BiLSTM; [23]) and a BERT-based classifier model in which the English `bert-base` parameters [8] are fine-tuned on the MQNLI train set. Output predictions are computed using the final representation above the [CLS] token. Models are trained to predict the relation of every pair of aligned phrases in Figure 2a. Additional model and training details are given in Appendix B.

Results Figure 2c summarizes the results of our BERT and BiLSTM models on the hardest fair generalization task Geiger et al. [10] creates with MQNLI. We find that our BiLSTM model is not able to learn this task, and that our BERT model is able to achieve high accuracy. The only models in Geiger et al. [10] able to achieve above 50% accuracy were task-specific tree-structured models with the structure of the tree in Figure 2a. Thus, our BERT-based model is the first general-purpose model able to achieve good performance on this hard generalization task. Without pretraining, the BERT-based model achieves $\approx$49.1%, confirming that pretraining is essential, as expected.

A natural hypothesis is that the BERT-based model achieves this high performance *because* it has in effect induced some approximation to the tree-like structure of the data-generating process in its own internal layers. With causal abstraction analysis, we are actually in a position to test this hypothesis.

5 A Case Study in Structural Neural Network Analysis

5.1 Causal Abstractions of Neural NLI models

Formulating Our Hypotheses We proceed just as we did for the simple motivating example in Section 3, except that we are now seeking to assess the extent to which the natural logic algebra in Figure 2a is a causal abstraction of the trained neural models in the above section.

The hallmark of Figure 2a is that it defines an alignment between premise and hypothesis at both lexical and phrasal levels. This permits us to run interchange interventions in a naturally compositional way. For a given non-leaf node N in Figure 2a, let C^N_{NatLog} be a submodel of C_{NatLog} that computes the relation between the aligned phrases under N and uses them to compute the final output relation between premise and hypothesis. For example, let $C^{\text{NP}_{\text{Obj}}}_{NatLog}$ be the submodel of C_{NatLog} that computes

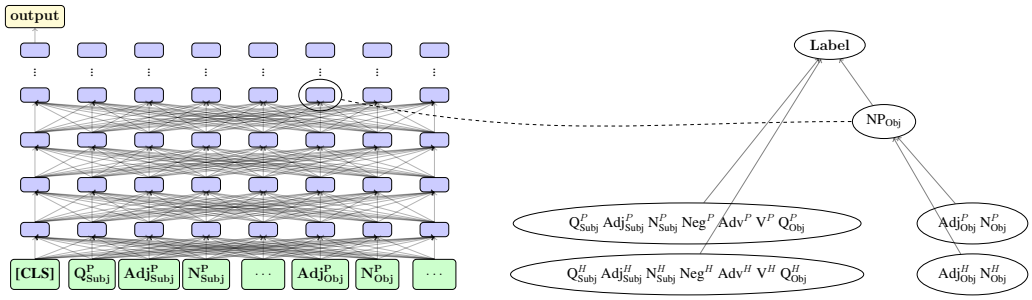


Figure 3: A BERT-based NLI model (left) aligned with the natural logic causal model $C^{\text{NP}_{\text{Obj}}}_{NatLog}$ (right), where the fourth vector representation above the $\text{Adj}^P_{\text{Obj}}$ token in the network is aligned with NP_{Obj}, the variable representing the relation between the object noun phrases. When analyzing a sample of 1000 examples, we found a subset of 383 where $C^{\text{NP}_{\text{Obj}}}_{NatLog}$ is an abstraction of N_{NLI} under this alignment.

the relation between the two aligned object noun phrases and then uses that relation in computing the final output relation between premise and hypothesis (see Figure 3 right). We would like to ask whether our trained neural models also compute this relation between object noun phrases and use it to make a final prediction. We can pose this same question for other nodes which correspond to a pair of aligned subphrases.

Alignment Search For each N, we search for an alignment between a neural representation in N_{NLI} and the variable N in C^N_{NatLog}. In principle, any location in the network could be the right one for any causal model. Testing every hypothesis in this space would be intractable. Thus, for each C^N_{NatLog}, we consider a restricted set of hidden representations based on the identity of N. The BERT model we use has 12 Transformer layers [30], meaning that there are 12 hidden representations for each input token. Each alignment search considers aligning the intermediate high-level variable with dozens of possible locations in the grid of BERT representations. Specifically, the following locations were considered for each N:

- Q_{Subj}, Adj_{Subj}, N_{Subj}, Neg, Adv, V, Q_{Obj}, Adj_{Obj}, N_{Obj}: hidden representations above the two descendant leaf tokens.
- NP_{Subj}, VP, and NP_{Obj}: same but above the four descendant leaf tokens.
- QP_{Obj}: hidden representations above Q^P_{Obj} and Q^H_{Obj}.
- NegP: same but above Neg^P and Neg^H.
- All nodes (for BERT): same but above [CLS] and [SEP].

For each alignment considered, we performed a full causal abstraction analysis. We report the results from the best alignments in Table 1, and we summarize the results from all alignments in Appendix D.

Interchange Interventions We first focus on our high-level causal models. Consider a non-leaf node N from Figure 2a and two input token sequences e and e' from MQNLI. Define

$$C^{N \leftarrow e'}_{NatLog}(e) \tag{4}$$

to be the output provided by the causal model C^N_{NatLog} when processing input e where the relation between the aligned subphrases under the node N is changed to the relation between those subphrases in e'. For example, simplifying for the sake of exposition, suppose e is (*some happy baker*, *no* ϵ *baker*), which has output label **contradiction**, and suppose e' is (*every happy person*, *some happy baker*), which has output label **entailment**. We wish to intervene on the noun phrase, so $N = \text{NP}$. In e, the noun phrase relation is entailment; in e', it is reverse entailment. Thus, $C^{\text{NP} \leftarrow e'}_{NatLog}(e)$ changes the object noun phrase relation in e to entailment while holding everything else about e constant. This results in the output label for the example (*some happy person*, *no* ϵ *baker*), which is **neutral**.

Next, we consider interventions in a neural model N_{NLI}. Define

$$N^{L \leftarrow e'}_{NLI}(e) \tag{5}$$

Table 1: Largest subsets of examples on which specific models C^N_{NatLog} are abstractions of an LSTM and BERT model trained on MQNLI. We record the size of such subsets as a percentage of the total 1000 examples. On this subset, we know that the neural models compute a representation of the relation between the aligned subphrases under N and use this information to make a final prediction.

Causal Model	LSTM	BERT
Q_{Subj}	0.7	13.1
Q_{Obj}	0.9	7.3
Neg	0.7	21.4
Adj_{Subj}	2.5	6.7
N_{Subj}	1.2	5.5
Adj_{Obj}	0.9	14.1
N_{Obj}	0.7	8.8
V	0.4	11.4
Adv	1.4	7.9
NP_{Subj}	1.0	6.7
NP_{Obj}	0.7	**38.3**
VP	0.4	11.4
NegP	0.9	11.8

(a) Main results (clique sizes) for non-leaf nodes of the tree in Figure 2a. The hypothesis we have most evidence for is that the BERT model computes a representation of the NP_{Obj} node with the alignment shown in Figure 3. Remarkably, with 1000 examples sampled, we found a subset of 383 examples where $C^{\text{NP}_{\text{Obj}}}_{NatLog}$ is an abstraction of BERT.

Nodes removed	BERT
N^H_{Obj}	31.9
A^H_{Obj}	15.7
N^P_{Obj}	33.8
A^P_{Obj}	15.8
$\text{N}^H_{\text{Obj}}, \text{A}^H_{\text{Obj}}$	31.9
$\text{N}^H_{\text{Obj}}, \text{N}^P_{\text{Obj}}$	14.1
$\text{N}^H_{\text{Obj}}, \text{A}^P_{\text{Obj}}$	32.2
$\text{N}^P_{\text{Obj}}, \text{A}^H_{\text{Obj}}$	31.6
$\text{A}^H_{\text{Obj}}, \text{A}^P_{\text{Obj}}$	8.8
$\text{N}^P_{\text{Obj}}, \text{A}^P_{\text{Obj}}$	32.1

Nodes added	BERT
$\text{Adj}^P_{\text{Subj}}$	30.5
N^P_{Subj}	37.2
Neg^P	14.9
Adv^P	26.9
V^P	35.6
Q^H_{Obj}	16.2
$\text{Adj}^H_{\text{Subj}}$	13.4
N^H_{Subj}	12.0
Neg^H	34.4
Adv^H	16.2
V^H	13.4
Q^H_{Obj}	12.0

(b) Detailed results (clique sizes) for Alternative causal models in a "neighborhood" around the model $C^{\text{NP}_{\text{Obj}}}_{NatLog}$, which has a single intermediate variable composed of four lexical items (See Figure 3). At left, we have alternative causal models where one or two of those lexical items are removed from the composition. At right, we have alternatives obtained by adding one lexical item to the composition. We observe that no alternative hypothesis about causal structure considered has more evidence.

to be the output provided by N_{NLI} processing the input e when the representation at location L is replaced with the representation at location L created when N_{NLI} is processing e'. This is exactly the process depicted in Figure 1, except now the networks are the complex trained networks of Section 4.

Our hypothesis linking Figure 2a with a model N_{NLI} takes the same form as (3). The causal model C^N_{NatLog} is a constructive abstraction of N_{NLI} when, for some representation location L, it is the case that, for all MQNLI examples e and e', we have

$$C^{N \leftarrow e'}_{NatLog}(e) = N^{L \leftarrow e'}_{NLI}(e) \tag{6}$$

This asserts a correspondence between interventions on the representations at L in network N_{NLI} and interventions on the variable N in the causal model C^N_{NatLog}. If it holds, then N_{NLI} computes the relation between the aligned phrases under the node N and uses this information to compute the relation between the premise and hypothesis.

We call a pair of examples (e, e') *successful* if it satisfies equation (6), i.e., interventions in both the target causal model and neural model produce equal results. In addition, to isolate the causal impact of our interventions, we specifically focus on pairs (e, e') for which performing the intervention produces a different output value than without the intervention. We call a pair (e, e') *impactful* if:

$$C^{N \leftarrow e'}_{NatLog}(e) \neq C^N_{NatLog}(e) \tag{7}$$

Quantifying Partial Success Equation (6) universally quantifies over all examples. We do not expect this kind of perfect correspondence to emerge in practice for real problems: neural network training is often approximate and variable in nature, and even our best model does not achieve *perfect* performance. However, we can still ask how widely (6) holds for a given model. To do this, we seek to find the *largest subset* of MQNLI on which C^N_{NatLog} is an abstraction of our neural models, for each non-leaf node N in C_{NatLog}.

More specifically, considering each example in MQNLI as a vertex in a graph, we add an undirected edge between two examples e_i and e_j if and only if both the ordered pairs (e_i, e_j) and (e_j, e_i) satisfy (6). In other words, C^N_{NatLog} is an abstraction of a neural model on a subset of examples S of MQNLI if and only if all examples in S form a *clique*.

The number of interventions we need to run scales quadratically with the number of inputs we consider, so we sample 1000 MQNLI examples, producing a total of $1000^2 = 1$M ordered pairs. We only consider examples for which the neural network outputs a correct label. For each node N and each of its corresponding neural network locations L, we perform interventions on all of these pairs.

We choose to measure the largest clique with at least one impactful edge, because (1) the causal abstraction relation holds with full force on that clique, but other measures such as the total number of connections lack this theoretical grounding, and (2) if a clique has at least one impactful edge, that guarantees the high-level variable is being used.

Results and Analysis For each target causal model node N and neural network representation location L, we construct a graph as described above with 1000 examples as vertices and add an edge between two examples e_i and e_j if and only if *both* (e_i, e_j) and (e_j, e_i) are successful. We then find the largest clique in this graph with at least one impactful edge and record its size.

Table 1a shows, for each causal model node N, the maximum size of cliques found among all neural locations. With this stricter *impactful* criterion (as opposed to simply using intervention success), our results show that, for almost all nodes N, our target causal model C^N_{NatLog} is indeed a causal abstraction of BERT on a significant number of examples in our dataset. These subsets are much smaller for the BiLSTM model.

We also investigated alternative high-level causal structures that are not variants of C_{NatLog} from Figure 2a. Specifically, we consider alternative models in a "neighborhood" around the model $C^{\text{NP}_{\text{Obj}}}_{NatLog}$ that can be obtained by adding one leaf, or by removing one or two leaves to the composition. These results are in Table 1b. Remarkably, all of these alternative models result in smaller clique sizes, significantly so for many of them. This further supports the significance of our results.

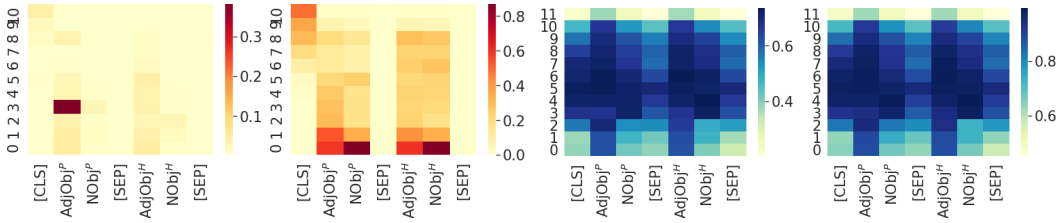
This analysis is similar to the analysis of our hypothetical addition example in Section 3, except for two crucial differences. First, for each variable N, we are hypothesizing that the causal model C^N_{NatLog} is an abstraction of N_{NLI}, whereas in the addition example there was only one model. To investigate this difference, we take $N = \text{NP}_{\text{Obj}}$ as a paradigm case, as it is the model with the strongest results. (The results for other nodes are in Appendix D.) Second, we only achieved partial experimental success, whereas in the addition example we assumed complete success. Crucially, this means that the following analysis will be valid only on subsets of the input space on which the abstraction relation holds between N_{NLI} and $C^{\text{NP}_{\text{Obj}}}_{NatLog}$.

We visualize the results of our intervention experiments for the node NP_{Obj} in Figure 4. The alignment with the largest subset of inputs aligns the NP_{Obj} variable in $C^{\text{NP}_{\text{Obj}}}_{NatLog}$ with the neural representation on the fourth layer of BERT above the $\text{Adj}^P_{\text{Obj}}$ token (see Figure 3). Because neural representations encode the value of their aligned variables and play a parallel causal role to their high-level variables, we know that, on this subset of input examples, at the fourth neural representation above the $\text{Adj}^P_{\text{Obj}}$ token, the four input embeddings for the object nouns and adjectives in the premise and hypothesis are composed to form a neural representation with information content of the relation between the object noun phrases in the premise and hypothesis. Then this representation is composed with the other input-embeddings to create an output representing the relation between the premise and hypothesis.

5.2 Comparison with Other Structural Analysis Methods

Probes We probed neural representation locations for the relation between aligned subexpressions on a subset of 12,800 randomly selected MQNLI examples. For a pair of aligned subexpressions below a node N in Figure 2a, we probe the columns above the same set of restricted class of tokens as described in Section 5.1.

To evaluate these probes, we report accuracy as well as *selectivity* as defined by Hewitt and Liang [12]: probe accuracy minus control accuracy, where *control accuracy* is the train set accuracy of a probe with the same architecture but trained on a control task to factor out probe success that can be



(a) Clique size. (b) Interchange success. (c) Probe selectivity. (d) Probe Accuracy.

Figure 4: Interchange intervention and probing results for the NP_{Obj} position. Vertical axes denote layers of BERT and horizontal axes denote the token position of hidden representations. The intervention success rates reported here are calculated based on intervention experiments with a change in the output label. Clique sizes are reported as % of 1000 examples.

attributed to the probe model itself. Our control task is to learn a random mapping from node types to semantic relations; see Appendix C for full details on how this task was constructed.

Figure 4 summarizes our probing results for $N = \text{NP}_{\text{Obj}}$, along with corresponding interchange intervention results for comparison. Probes tell us that information about the relation between the aligned noun phrases is encoded in nearly all of the locations we considered, and using the selectivity metric does not result in any qualitative change. In contrast, our intervention heatmaps indicate only a small number of locations store this information in a causally relevant way. Clearly, our intervention experiments are far more discriminating than probes. Appendix D provides examples involving other variables along with the intervention experiments, where the general trend of interchange interventions being more discriminating holds.

Integrated Gradients Attribution methods that estimate feature importance can measure causal properties of neural representations, but a single feature importance method is an impoverished characterization of a representation's role in network behavior. Whereas our interchange interventions gave us high-level information about how a neural representation is composed and what it is composed into, attribution methods simply tell us "how much" a representation contributes to the network output on a give input. Moreover, intervention interchanges provide a rich, high-level characterization of causal structure on a space of inputs.

We use integrated gradients on our models to verify the intuitive hypothesis that if a premise and hypothesis differ by a single token, then the neural representations above that token should be more causally responsible for the network output than other representations. For example, given premise 'Every sleepy cat meows' and hypothesis 'Some hungry cat meows', the attributive modifier position is different and the rest are matched. The neural representations above the adjective tokens *sleepy* and *hungry* should be more important for the network output than others, because if those adjectives were the same, the example label would change from **neutral** to **entailment**. We summarize the results of our integrated gradient experiments in Appendix E, where we confirm our intuitive hypothesis.

6 Conclusion

We have introduced a methodology for deriving interpretable causal explanations of neural network behaviors, grounded in a formal theory of causal abstraction. The methodology involves first *formulating a hypothesis* in the form of a high-level, interpretable causal model, then *searching for an alignment* between the neural network and the causal model, and finally *verifying experimentally* that the neural representations encode the same causal properties and information content as the corresponding components of the high-level causal model. As a case study demonstrating the feasibility of the approach, we analyzed neural models trained on the semantically formidable MQNLI dataset. Guided by the intuition that success on this challenging task may call for a way of recapitulating the causal structure of the natural logic model that generates the MQNLI data, we were able to verify the hypothesis that a state-of-the-art BERT-based model partially realizes this structure, whereas baseline models that do not perform as well fail to do so. This suggestive case study demonstrates that our theoretically grounded methodology can work in practice.

Acknowledgments and Disclosure of Funding

Our thanks to Amir Feder, Noah Goodman, Elisa Kreiss, Josh Rozner, Zhengxuan Wu, and our anonymous reviewers. This research is supported in part by grants from Facebook and Google.

References

[1] S. Beckers and J. Y. Halpern. Abstracting causal models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):2678–2685, Jul. 2019. doi: 10.1609/aaai.v33i01.33012678. URL https://ojs.aaai.org/index.php/AAAI/article/view/4117.

[2] S. Beckers, F. Eberhardt, and J. Y. Halpern. Approximate causal abstractions. In R. P. Adams and V. Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 606–615, Tel Aviv, Israel, 22–25 Jul 2020. PMLR. URL http://proceedings.mlr.press/v115/beckers20a.html.

[3] A. Binder, G. Montavon, S. Bach, K. Müller, and W. Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. *CoRR*, abs/1604.00825, 2016. URL http://arxiv.org/abs/1604.00825.

[4] S. Bongers, P. Forré, J. Peters, B. Schölkopf, and J. M. Mooij. Foundations of structural causal models with cycles and latent variables. *arXiv.org preprint*, arXiv:1611.06221v4 [stat.ME], Oct. 2020. URL https://arxiv.org/abs/1611.06221v4.

[5] K. Chalupka, F. Eberhardt, and P. Perona. Multi-level cause-effect systems. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 361–369, Cadiz, Spain, 09–11 May 2016. PMLR. URL http://proceedings.mlr.press/v51/chalupka16.html.

[6] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian. Neural network attributions: A causal perspective. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 981–990, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/chattopadhyay19a.html.

[7] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://www.aclweb.org/anthology/W19-4828.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[9] Y. Elazar, S. Ravfogel, A. Jacovi, and Y. Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. In *Proceedings of the 2020 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Nov. 2020. doi: 10.18653/v1/W18-5426.

[10] A. Geiger, I. Cases, L. Karttunen, and C. Potts. Posing fair generalization tasks for natural language inference. 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 4475–4485, Stroudsburg, PA, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1456. URL https://www.aclweb.org/anthology/D19-1456.

[11] A. Geiger, K. Richardson, and C. Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://www.aclweb.org/anthology/2020.blackboxnlp-1.16.

[12] J. Hewitt and P. Liang. 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, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://www.aclweb.org/anthology/D19-1275.

[13] D. Hupkes, S. Bouwmeester, and R. Fernández. Analysing the potential of seq-to-seq models for incremental interpretation in task-oriented dialogue. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 165–174, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5419. URL https://www.aclweb.org/anthology/W18-5419.

[14] T. F. Icard and L. S. Moss. Recent progress on monotonicity. *Linguistic Issues in Language Technology*, 9(7):1–31, January 2013.

[15] G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[16] T. P. Lillicrap and K. P. Kording. What does it mean to understand a neural network?, 2019.

[17] B. MacCartney and C. D. Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 193–200, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1654536.1654575.

[18] B. MacCartney and C. D. Manning. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140–156, Tilburg, The Netherlands, Jan. 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W09-3714.

[19] J. Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 411–420, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001.

[20] M. Peters, M. Neumann, L. Zettlemoyer, and W.-t. Yih. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1179. URL https://www.aclweb.org/anthology/D18-1179.

[21] A. Ravichander, Y. Belinkov, and E. Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance?, 2020.

[22] P. K. Rubenstein, S. Weichwald, S. Bongers, J. M. Mooij, D. Janzing, M. Grosse-Wentrup, and B. Schölkopf. Causal consistency of structural equation models. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*. Association for Uncertainty in Artificial Intelligence (AUAI), Aug. 2017. URL http://auai.org/uai2017/proceedings/papers/11.pdf. *equal contribution.

[23] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[24] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *CoRR*, abs/1605.01713, 2016. URL http://arxiv.org/abs/1605.01713.

[25] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2nd edition, 2001. ISBN 9780262194402.

[26] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, 12 2014.

[27] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/sundararajan17a.html.

[28] I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

[29] J. van Benthem. A brief history of natural logic. In M. Chakraborty, B. Löwe, M. Nath Mitra, and S. Sarukki, editors, *Logic, Navya-Nyaya and Applications: Homage to Bimal Matilal*, 2008.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. 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 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[31] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[32] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1.

A Additional Details on MQNLI

A.1 Dataset Description

The MQNLI dataset contains sentences of the form

$$\mathrm{Q_S\ Adj_S\ N_S\ Neg\ Adv\ V\ Q_O\ Adj_O\ N_O}$$

where $\mathrm{N_S}$ and $\mathrm{N_O}$ are nouns, V is a verb, $\mathrm{Adj_S}$ and $\mathrm{Adj_O}$ are adjectives, and Adv is an adverb. These categories all have 100 words. Neg is *does not*, and $\mathrm{Q_S}$ and $\mathrm{Q_O}$ can be *every*, *not every*, *some*, or *no*. Additionally, $\mathrm{Adj_S}$, $\mathrm{Adj_O}$, Adv, and Neg can be the empty string ε.

NLI examples are constructed so that non-identical non-empty nouns, adjectives, verbs, and adverbs with identical positions in s_p and s_h are semantically unrelated. This means that the learning task is trivial for these lexical items, as the correct relation is equivalence when they are identical and independence when they are not identical.

For our experiments, we used a train set with 500K examples, a dev set with 60k examples, and a test set with 10K examples – the most difficult generalization scheme of Geiger et al. [10].

A.2 A Natural Logic Causal Model

Geiger et al. [10] construct a natural logic model that solves MQNLI using a formalization they call *composition trees*, which is easily translated into the causal model we call C_{NatLog}. Natural logic is a flexible approach to doing logical inference directly on natural language expressions [14, 17, 29] where the *semantic relations* between phrases are compositionally computed from the semantic relations between aligned subphrases and *projectivity signatures*, which encode how semantic operators interact compositionally with their arguments (which are semantic relations). The causal model C_{NatLog} performs inference on aligned semantic parse trees that represent both the premise and hypothesis as a single structure and calculates semantic relations between all subphrases compositionally.

B Model Training and Interchange Experiment Details

We evaluated two models on MQNLI: a multi-layered bidirectional LSTM baseline and a Transformer-based model trained to do masked language modeling and next-sentence prediction [8]. We rely on the uncased BERT-base initial parameters from Hugging Face `transformers` [31]. For both models, we concatenate the premise s_p and hypothesis s_h into one string with special separator tokens: [CLS] s_p [SEP] s_h [SEP].

For the BiLSTM, we concatenate the hidden state above the last [SEP] and the [CLS] in the last layer for the forward and backward directions respectively to obtain a representation for the whole input, and then apply three linear transformations on top of that. The final transformation outputs a logit score for each class in the label space.

For the BERT model, we apply one linear transformation to the final layer's hidden representation above the [CLS] token to obtain a logit score for each label class.

B.1 Tokenization

In the original setting of MQNLI, some positions in the premise and hypothesis consist of two words such as *not every* in $\mathrm{Q_S}$ and $\mathrm{Q_O}$ and *does not* in the leaf nodes $\mathrm{Neg^P}$ and $\mathrm{Neg^H}$ (as shown in the beginning of Section A.1). We treat them as two separate tokens in order to utilize BERT's knowledge of these function words. To ensure all sentences have identical length, we introduce one extra empty string tokens ε to single-word quantifiers and two such tokens in the place of Neg^P and Neg^H for sentences without negation.

For consistency, we use the same tokenization method for both models.

Table 2: Ablation results.

Model	Dev	Test
Fine-tuned BERT	**88.25**	**88.50**
Without augmented examples	55.42	54.51

B.2 Dataset Augmentation with Labeled Subphrases

The *hard but fair* MQNLI generalization task requires the dataset to explicitly expose the model to labels for each intermediate node that is a relation in C_{NatLog}. For each training example $(s_p, s_h, y) \in \mathcal{S}$, we create an additional example (s_p^N, s_h^N, y^N) for each node N. (s_p^N, s_h^N) is a *subphrase* pair made up of all the leaf tokens under node N in the original input (s_p, s_h), and y^N is the relation computed by C_{NatLog} for that subphrase pair. The set of labels we use for these subphrase examples is disjoint from that of the full-sentence examples. During training, the augmented examples are coupled with original examples in each batch. For BERT, the subphrase pairs occupy their original positions in the sentence, while we pad and apply an attention mask over all other positions. For the BiLSTM, we align them to the left, with [SEP] in between the two parts of the pair.

We performed an ablation experiment to test whether removing the augmented examples would affect BERT's performance. Using the same grid-search setting, we see that BERT's dev set accuracy decreased from 88.25% to 55.42%, and test set accuracy decreased from 88.50% to 54.51%. This indeed shows that the above data augmentation method is important for BERT to learn the type of generalization required for the hard MQNLI task.

B.3 Training Procedure

For the BiLSTM, we use 256 dimensions for token embeddings and 128 dimensions for the hidden states in each LSTM direction. We grid search for $\{2, 4, 6\}$ layers. We randomly initialize each element in the token embeddings from the distribution $\mathcal{N}(0, 1)$ scaled down by a factor of 0.1. We use a batch size of $768 = 64 \times 12$, with 64 original examples per batch and 11 augmented examples for each one. We apply a dropout of 0.1, and grid search for learning rates in $\{0.001, 0.0001\}$. We train for a maximum of 400 epochs and perform early stopping when the dev set accuracy does not increase for 20 epochs. We train each grid search setting 3 times with different random seeds.

For BERT, we use the same model architecture for the uncased base variant. We use a batch size of $192 = 16 \times 12$, and grid search for learning rates in $\{2.0 \times 10^{-5}, 5.0 \times 10^{-5}\}$. We train for a maximum of $\{3, 4\}$ epochs. We warm up the learning rate linearly from 0 to the specified value in the first 25% of steps of the first epoch, and linearly decrease the learning rate to 0 following that until the end of training.

All models were trained with 1 GPU core on a cluster with models including GeForce RTX 2080 Ti, GeForce GTX Titan X, Titan XP and Titan V, each with 11-12GB memory. Each instance of the grid search took on average 5.5 hours to train. We repeated each grid search setting with 4 different random seeds and took the instance with the highest dev set accuracy.

B.4 Interchange experiment details

There are 14 intermediate nodes in the high-level causal model (NegP, QP_{Obj}, Q_{Subj}, NP_{Subj}, Adj_{Subj}, N_{Subj}, Neg, VP, Adv, V, Q_{Obj}, NP_{Obj}, Adj_{Obj}, N_{Obj}). For each high-level node, we conducted a set of interchange experiments on each one of 11 BERT layers (excluding the final layer, since only the [CLS] token causally impacts the output). Each high-level node has its own fixed set of hand-specified intervention locations in the time-step/sentence length dimension, and we use the same intervention locations on each layer. For each of the $14 \times 11 = 154$ interchange experiments, it took on average 1.15 hours to run using the same computation resources mentioned above.

C Probing Details

C.1 Probe Models

Our probe models are single-layer softmax classifiers: $y_i \propto \mathrm{softmax}(Ah_i + b)$ where h_i is a hidden representation and $y_i \in \mathbb{R}$. Following Hewitt and Liang [12], to control the dimensionality of A, we factorize it in the form $A = LR$ where $L \in \mathbb{R}^{|\mathcal{R}| \times \ell}$ and $R \in \mathbb{R}^{\ell \times d}$ where d is the dimensionality of h_i.

We train the probes on hidden representations of a set of 12,800 examples that are randomly selected from the model's original training set. We additionally take 2,000 examples to form a development set for early stopping. We filter out examples for which the model outputs a wrong prediction.

For training, we perform a grid search, maximizing for selectivity. We set a dropout of 0.1, and apply early stopping when the development set loss does not increase for 4 epochs. We train for a maximum of 40 epochs. We also anneal the learning rate by a factor of 0.5 if the dev set loss did not increase in the last epoch. We use a batch size of 512, learning rates in $\{0.001, 0.01\}$, weight decay regularization constants in $\{0.01, 0.1\}$. We set $\ell \in \{8, 32\}$ for restricting the maximum rank of the linear matrix A.

Using the same computation resources described above, each grid search setting took approximately 5 hours to run. For each grid search setting we trained a separate probe for every possible $\langle$causal model node, BERT representation$\rangle$ combination, where for the latter we use the intervention locations outlined in the "Alignment Search" part of Section 5.1 on each BERT layer.

C.2 Control Task

For each high-level node N, we construct a random mapping $\mathrm{Control}_N : \mathcal{S}_N \mapsto \mathcal{L}_N$ where $\mathcal{S}_N$ is the set of all aligned subexpressions under the node N and $\mathcal{L}_N$ is the output label space. For phrasal nodes (VP, NegP, etc.) and aligned verbs and nouns, $\mathcal{L}_N$ is the set of 7 possible relations $\{\#, \equiv, \sqsubset, \sqsupset, |, \hat{}, \smile\}$ from MacCartney and Manning [17]. For aligned quantifiers, the label space is the set of all projectivity signatures that can be produced by their composition.

Similar to Hewitt and Liang [12], $\mathrm{Control}_N$ will assign the same control label regardless of the context as long as its input consists of the same tokens. Consequently, the possible input space $\mathcal{S}_N$ grows exponentially larger if N corresponds to longer subphrases (such as NegP and $\mathrm{QP}_{\mathrm{Obj}}$), and the control task becomes much more difficult to solve, resulting in near random accuracies.

C.3 Extended Probe Analysis

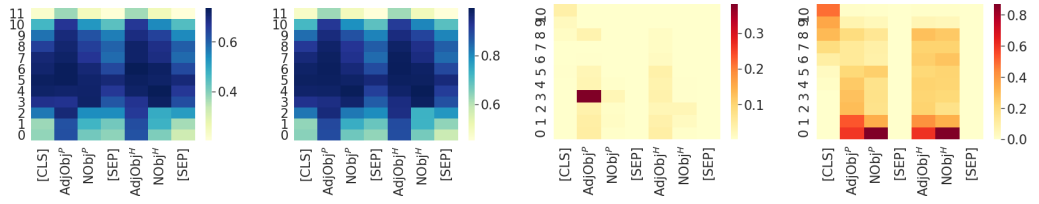
In Figures 5–7 we report some more representative selectivity and accuracy results for our probing experiments on BERT trained on the hard variant, juxtaposed against intervention experiments on the same model. For open-class words and full phrases, probing and intervention show similar trends. For aligned closed-class words, we find near-zero selectivity because the domain of the control function is so small.

In general, probing and intervention experiments for relations between aligned single open-class words (i.e., $\mathrm{N}_{\mathrm{Subj}}$, $\mathrm{Adj}_{\mathrm{Subj}}$, $\mathrm{N}_{\mathrm{Obj}}$, $\mathrm{Adj}_{\mathrm{Obj}}$, Adv, V) show similar trends, which can be seen in Figures 4c–4b. Every location except those above the [CLS] and [SEP] tokens has a near-100% accuracy, while selectivity is only high in the last few layers. Lower layers of BERT contains more information about word identity and hence may allow the probe to memorize each input pair, resulting in higher control task accuracy and lower selectivity for lower layers.

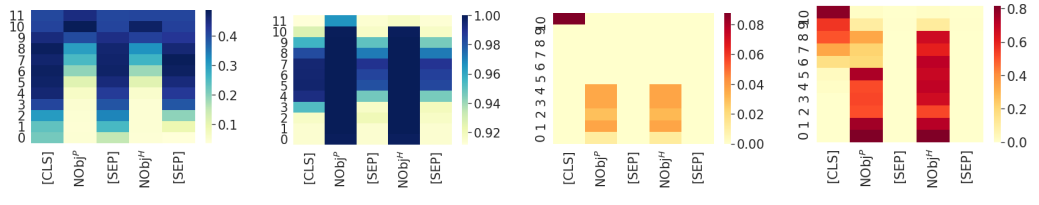
Probing experiments for relations between aligned multi-word subphrases (i.e., $\mathrm{NP}_{\mathrm{Subj}}$, VP, $\mathrm{NP}_{\mathrm{Obj}}$, $\mathrm{QP}_{\mathrm{Obj}}$ and NegP) show similar trends as shown in the row of figures 6m to 6h. As described in Section C.2, all control probes for these achieve near-random performance, so selectivity and accuracy differ by the random baseline accuracy, which is evident by comparing figures 6m and 6n.

On the other hand, probing experiments for aligned closed-class words (quantifiers and negation) have near-zero selectivity, as shown in Figure 6a. This is because the domain of the control function is the small set of closed-class word pairs, so memorizing the identity of these words becomes trivial for the probe.

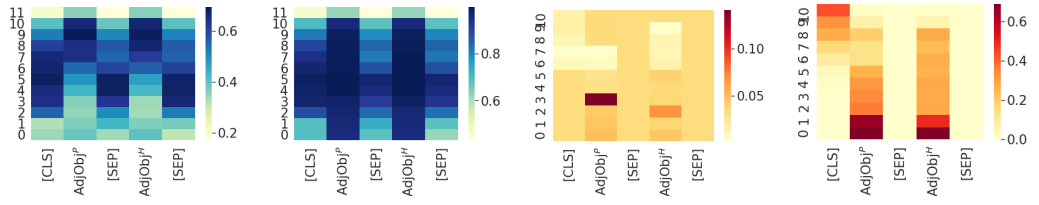
D Probing and Intervention Heatmaps



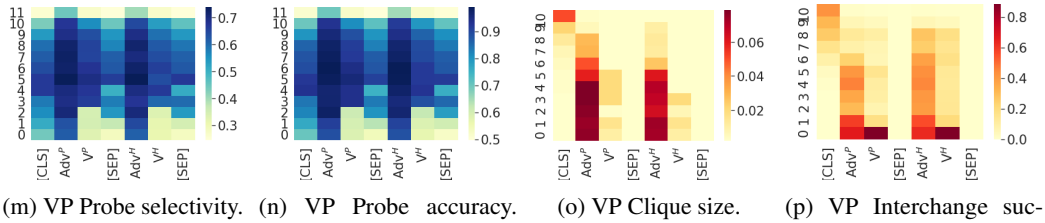
(a) NP_{Obj} Probe selectivity. (b) NP_{Obj} Probe accuracy. Random baseline: 25% (c) NP_{Obj} Clique size. (d) NP_{Obj} Interchange success.



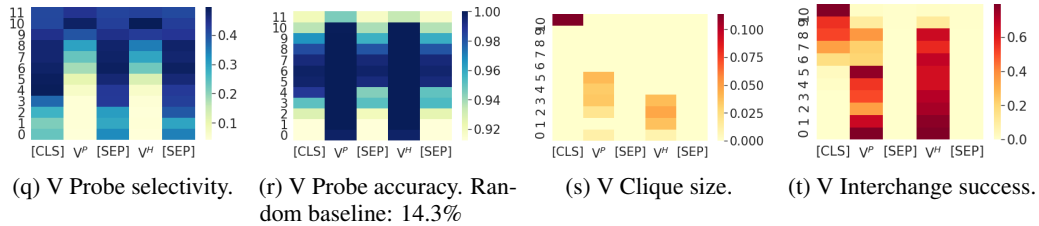
(e) N_{Obj} Probe selectivity. (f) N_{Obj} Probe accuracy. Random baseline: 14.3% (g) N_{Obj} Clique size. (h) N_{Obj} Interchange success.



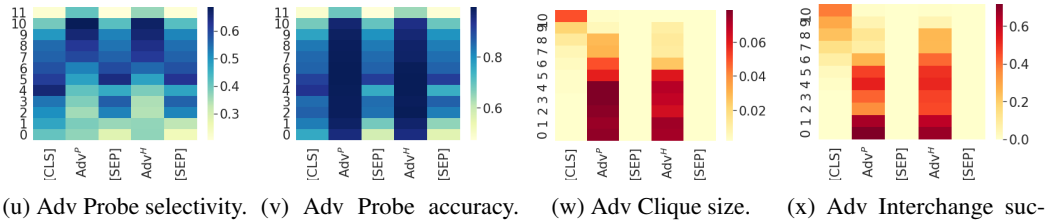
(i) Adj_{Obj} Probe selectivity. (j) Adj_{Obj} Probe accuracy. Random baseline: 14.3% (k) Adj_{Obj} Clique size. (l) Adj_{Obj} Interchange success.



(m) VP Probe selectivity. (n) VP Probe accuracy. Random baseline: 25% (o) VP Clique size. (p) VP Interchange success.

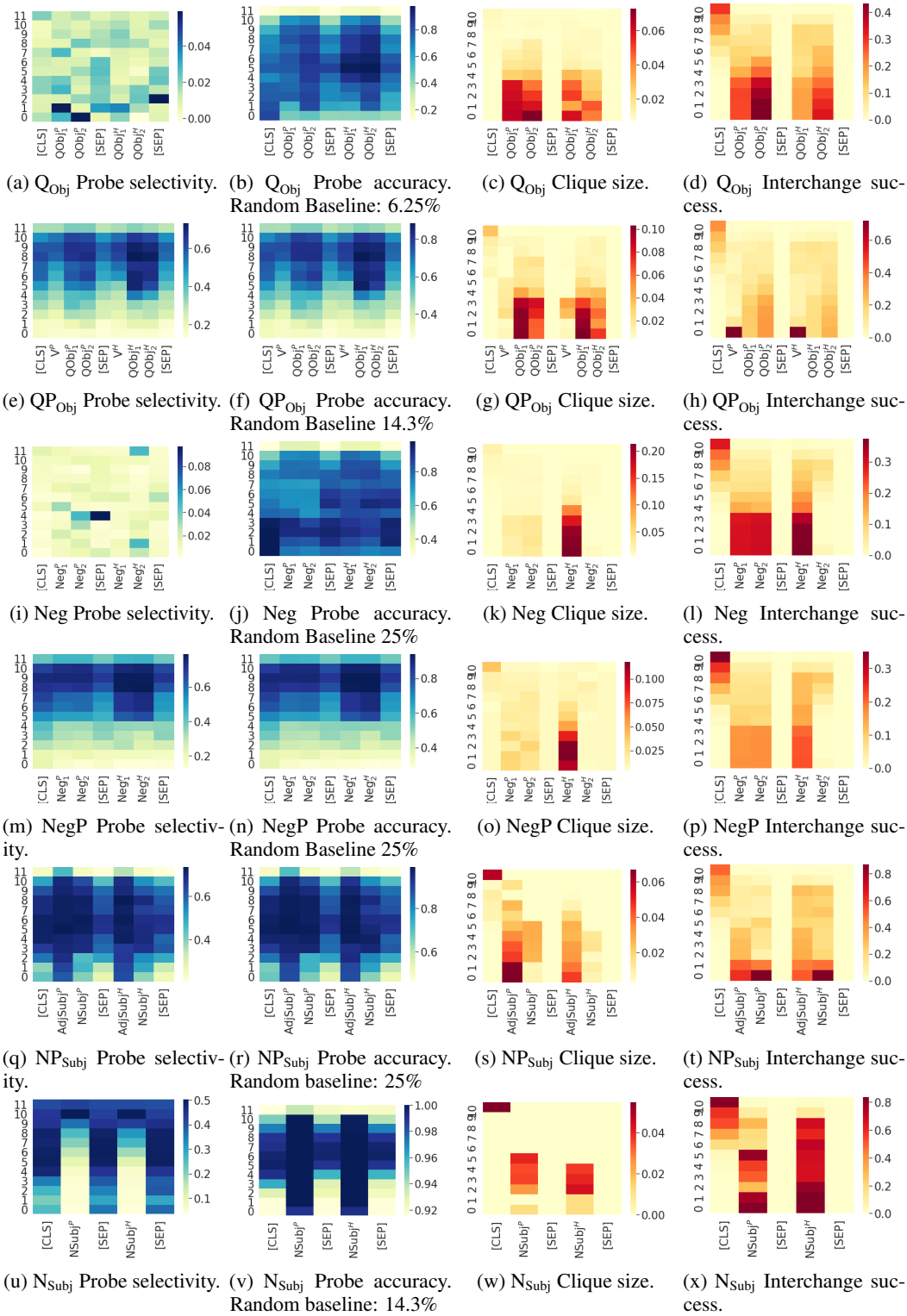


(q) V Probe selectivity. (r) V Probe accuracy. Random baseline: 14.3% (s) V Clique size. (t) V Interchange success.



(u) Adv Probe selectivity. (v) Adv Probe accuracy. Random baseline: 14.3% (w) Adv Clique size. (x) Adv Interchange success.

Figure 5: Full probing and interchange intervention results for high-level nodes NP_{Obj}, N_{Obj}, Adj_{Obj}, VP, V, and Adv. Vertical axes denote BERT layers and horizontal axes denote the token position of hidden representations. Intervention success rates are based on experiments with a change in the output label. Clique sizes are reported as a percentage of all examples.



(a) Q_{Obj} Probe selectivity. (b) Q_{Obj} Probe accuracy. Random Baseline: 6.25% (c) Q_{Obj} Clique size. (d) Q_{Obj} Interchange success.

(e) QP_{Obj} Probe selectivity. (f) QP_{Obj} Probe accuracy. Random Baseline 14.3% (g) QP_{Obj} Clique size. (h) QP_{Obj} Interchange success.

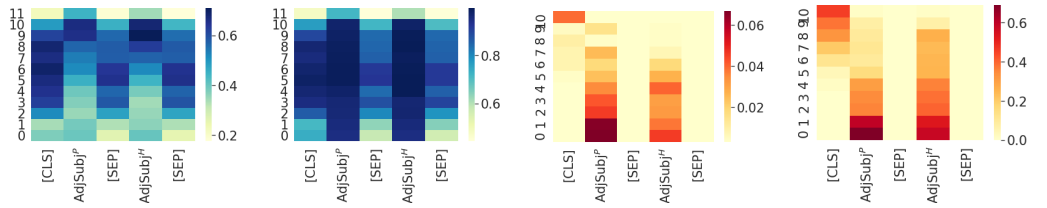
(i) Neg Probe selectivity. (j) Neg Probe accuracy. Random Baseline 25% (k) Neg Clique size. (l) Neg Interchange success.

(m) NegP Probe selectivity. (n) NegP Probe accuracy. Random Baseline 25% (o) NegP Clique size. (p) NegP Interchange success.

(q) NP_{Subj} Probe selectivity. (r) NP_{Subj} Probe accuracy. Random baseline: 25% (s) NP_{Subj} Clique size. (t) NP_{Subj} Interchange success.

(u) N_{Subj} Probe selectivity. (v) N_{Subj} Probe accuracy. Random baseline: 14.3% (w) N_{Subj} Clique size. (x) N_{Subj} Interchange success.

Figure 6: Full probing and interchange intervention results on the high level nodes Q_{Obj}, QP_{Obj}, Neg, NegP, NP_{Subj} and N_{Subj}. Vertical axes denote BERT layers and horizontal axes denote the token position of hidden representations. Intervention success rates are based on experiments with a change in the output label. Clique sizes are reported as a percentage of all examples.

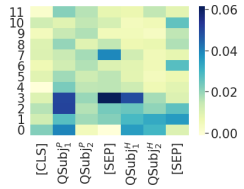


(a) Adj_{Subj} Probe selectivity.

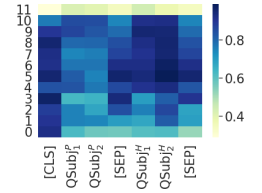
(b) Adj_{Subj} Probe accuracy. Random baseline: 14.3%

(c) Adj_{Subj} Clique size.

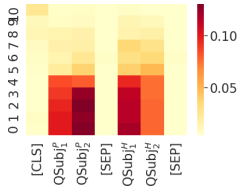
(d) Adj_{Subj} Interchange success.



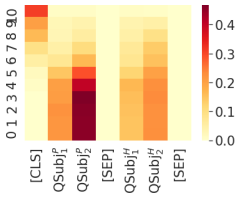
(e) Q_{Subj} Probe selectivity.



(f) Q_{Subj} Probe accuracy. Random Baseline: 6.25%



(g) Q_{Subj} Clique size.



(h) Q_{Subj} Interchange success.

Figure 7: Full probing and interchange intervention results for high-level nodes Adj_{Subj} and Q_{Subj}. Vertical axes denote BERT layers and horizontal axes denote the token position of hidden representations. Intervention success rates are based on experiments with a change in the output label. Clique sizes are reported as a percentage of all examples.

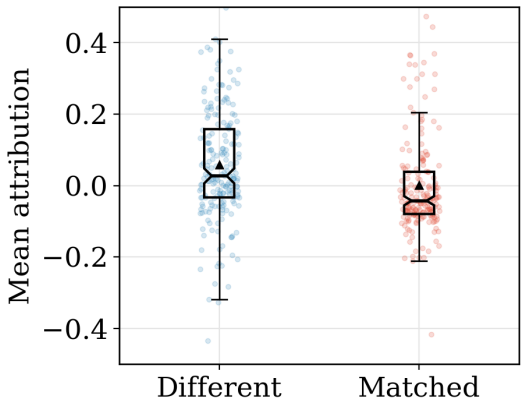


Figure 8: Integrated Gradients values for examples in which the premise and hypothesis differ by in exactly one aligned position. 'Different' refers to the IG value for this position, and 'Matched' is a randomly selected different position from each example. The two populations are different according to a Wilcoxon signed-rank test ($p < 0.00001$). The 'Different' positions have positive attribution on average, aligning with our expectation that they tend to be decisive for the output prediction.

E Integrated Gradients

We report attributions for the first BERT layer; later layers tend to concentrate importance onto the [CLS] token, since it is the direct basis for the classifier head in our model. To simplify the analysis, we restrict attention to examples in which exactly one position is different across the premise and hypothesis, and 'Matched' is a randomly selected position from elsewhere in the example. We see that the 'Matched' are positive in general, which aligns with our expectation that they are the most important positions in these examples (Figure 8).

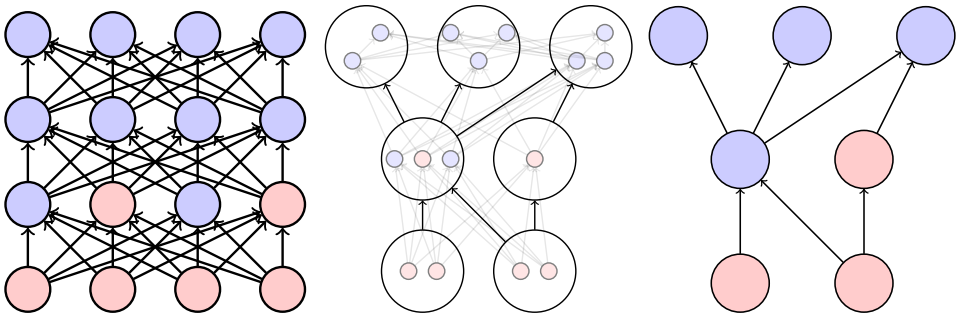


Figure 9: Schematic depicting constructive abstraction [1]. The variables of the low-level model (left) are divided into partitions (center) such that each low-level partition corresponds to a high level variable from the high-level model (right). The circles represent variables and the arrows represent causal dependencies. Blue circles are variables that are not being intervened on and red circles are variables that are being intervened on. Observe that a low-level causal dependence between partitions does not necessarily result in a high-level causal dependence between variables and that not every low-level intervention results in a high level intervention.

F Background on Causal Models and Causal Abstraction

In this appendix we provide relevant background on causal models and causal abstraction, sufficient to define the notion of *constructive abstraction*.

F.1 Causal Models

Definition F.1. (Signatures) A signature S is a pair $(\mathcal{V}, \mathcal{R})$, where $\mathcal{V}$ is a set of variables and $\mathcal{R}$ is a function that associates with every variable $X \in \mathcal{V}$ a nonempty set $\mathcal{R}(X)$ of possible values. If $\mathbf{X} = (X_1, \ldots, X_n)$, $\mathcal{R}(\mathbf{X})$ denotes the cross product $\mathcal{R}(X_1) \times \cdots \times \mathcal{R}(X_n)$.

Definition F.2. (Causal models) A causal model M is a pair $(\mathcal{S}, \mathcal{F})$, where $\mathcal{S}$ is a signature and $\mathcal{F}$ defines a function that associates with each variable X a structural equation $\mathcal{F}^X$ giving the value of X in terms of the values of other variables. Formally, the equation $\mathcal{F}^X$ maps $\mathcal{R}(\mathcal{V} - \{X\})$ to $\mathcal{R}(X)$, so $\mathcal{F}^X$ determines the value of X, given the values of all the other variables in $\mathcal{V}$.

Definition F.3. (Dependence) X causes Y according to M, denoted $X \rightsquigarrow Y$, if there is some setting of the variables other than X and Y such that varying the value of X results in a variation in the value of Y; that is, there is a setting $\mathbf{z}$ of the variables $\mathbf{Z} = \mathcal{V} - \{X, Y\}$ and values x and x' of X $\mathcal{F}^Y(x, \mathbf{z}) \neq \mathcal{F}^Y(x', \mathbf{z})$.

Definition F.4. (Intervention) An intervention i has the form $\mathbf{X} \leftarrow \mathbf{x}$, where $\mathbf{X}$ is a vector of variables. Intuitively, this means that the values of the variables in $\mathbf{X}$ are set to $\mathbf{x}$. Setting the value of some variables $\mathbf{X} \leftarrow \mathbf{x}$ in a causal model $M = (\mathcal{S}, \mathcal{F})$ results in a new causal model, denoted $i(M)$, which is identical to M, except that $\mathcal{F}$ is replaced by $i(\mathcal{F})$: for each variable $Y \notin \mathbf{X}$, $i(\mathcal{F}^Y) = \mathcal{F}^Y$ (i.e., the equation for Y is unchanged), while for each $X' \in \mathbf{X}$, $i(F^{X'})$ is the constant function sending all arguments to x' (where x' is the value in $\mathbf{x}$ corresponding to X_i).

When we write out the structured equations for a variable X, for simplicity's sake, we treat $\mathcal{F}^X$ as a map from $\mathcal{R}(\{Y \in \mathcal{V} : Y \rightsquigarrow X\})$ to $\mathcal{R}(X)$.

Note that interventions $\mathbf{X} \leftarrow \mathbf{x}$ correspond 1–1 with variable settings $\mathbf{x}$. We make use of this in what follows.

F.2 Constructive Abstraction

The following definitions are in agreement with the definitions from Beckers and Halpern [1], but differ somewhat in presentation. We additionally omit exogenous variables, as they play no role in our deterministic setting. In this section we take causal models to be pairs $(M, \mathcal{I})$, with a set $\mathcal{I}$ of *admissible interventions* made explicit.

Definition F.5. (Projection and Inverse Projection) Given some $\mathbf{v} \in \mathcal{R}(\mathcal{V})$ and $\mathbf{X} \subseteq \mathcal{V}$, define $\mathsf{Proj}(\mathbf{v}, \mathbf{X})$ to be the restriction of $\mathbf{v}$ to the variables in $\mathbf{X}$. Given some $\mathbf{x} \subseteq \mathcal{V}(\mathbf{X})$, the inverse

$\mathsf{Proj}^{-1}(\mathbf{x})$ is defined as usual:

$$\{\mathbf{v} \in \mathcal{R}(\mathcal{V}) : \mathbf{x} \text{ is the restriction of } \mathbf{v} \text{ to } \mathbf{X}\}.$$

We are interested in (possibly partial) functions $\tau : \mathcal{R}_L(\mathcal{V}_L) \to \mathcal{R}_H(\mathcal{V}_H)$ mapping settings of low-level variables to settings of high-level variables. Such a function τ naturally induces a function ω_τ between sets of interventions, where $\omega_\tau(\mathbf{x}) = \mathbf{y}$ just in case

$$\tau(\mathsf{Proj}^{-1}(\mathbf{x})) = \mathsf{Proj}^{-1}(\mathbf{y}).$$

We are now in a position to define τ-abstraction:

Definition F.6. (τ-abstraction) Fix a function $\tau : \mathcal{R}_L(\mathcal{V}_L) \to \mathcal{R}_H(\mathcal{V}_H)$, which in turn fixes $\omega_\tau : \mathcal{I}_L \to \mathcal{I}_H$. We say $(M_H, \mathcal{I}_H)$ is a τ-abstraction of $(M_L, \mathcal{I}_L)$ if the following hold:

1. τ is surjective.
2. ω_τ is surjective.
3. for all $i_L \in \mathcal{I}_L$ we have $\tau(i_L(M_L)) = \omega_\tau(i_L)(M_H)$.

One way to think of this is: τ is a map from $\mathcal{R}(\mathcal{V}_L)$ to $\mathcal{R}(\mathcal{V}_H)$, which in turn induces a map ω_τ from the space of *projections* on $\mathcal{R}(\mathcal{V}_L)$ to *projections* on $\mathcal{R}(\mathcal{V}_H)$. The conditions on τ-abstraction below then simply become that τ and ω_τ are both total and surjective on their respective (co)domains, and a second condition that can be easily encoded in terms of potential outcomes. For any setting/projection $\mathbf{x}$ at the low-level, we require that $M_L \models \mathbf{v}_\mathbf{x}$ iff $M_H \models \tau(\mathbf{v})_{\omega_\tau(\mathbf{x})}$.

Finally, to be a *constructive* τ-abstraction we simply require that τ decompose into a family of "component" functions, as below.

Definition F.7 (Constructive τ-abstraction)**.** $(M_H, \mathcal{I}_H)$ is a constructive τ-abstraction of $(M_L, \mathcal{I}_L)$ if, in addition to being a τ-abstraction, we can associate with each X_H a subset P_{X_H} of $\mathcal{V}_L$, such that the mapping $\tau : \mathcal{R}(\mathcal{V}_L) \to \mathcal{R}(\mathcal{V}_H)$ decomposes into a family of functions $\tau_{X_H} : \mathcal{R}(P_{X_H}) \to \mathcal{R}(X_H)$. We say M_H is a constructive abstraction of M_L if it is a constructive τ-abstraction for some τ.

In other words, for a constructive abstraction it suffices to define the component functions τ_{X_H}, as these completely determine τ. In fact, the maps τ_{X_H} more generally induce a (partial) function from projections of $\mathcal{R}(\mathcal{V}_L)$ to (in fact, onto) projections of $\mathcal{R}(\mathcal{V}_H)$ in the following sense. For any setting $\mathbf{h} = [h_1 \dots h_k]$ of high-level variables $H_1, \dots, H_k$ we can find low-level setting $\mathbf{y}$ such that projections of $\mathbf{y}$ map via τ_{H_i} to h_i. Slightly abusing notation, denote this (partial) low-level setting $\mathbf{y}$ as $\tau^{-1}(\mathbf{h})$. So, in particular when $\mathbf{h}$ corresponds to an intervention in $\mathcal{I}_H$, the setting $\tau^{-1}(\mathbf{h})$ should specify a corresponding intervention in $\mathcal{I}_L$. Indeed, point (2) of Def. F.6 tells us that (the intervention corresponding to) $\tau^{-1}(\mathbf{h})$ should be mapped via ω_τ to (the intervention corresponding to) $\mathbf{h}$.

G Causal Abstraction Analysis of C_+

G.1 Formal Definition of C_+

We define the causal model $C_+ = (\mathcal{V}_+, \mathcal{R}_+, \mathcal{F}_+)$ as follows (where $\mathbb{N}_k = \{0, \dots, k\}$):

$$\begin{aligned}
\mathcal{V}_+ &= \{X, Y, Z, W, S_1, S_2\} \\
\mathcal{R}_+(V) &= \mathbb{N}_9, \text{ for } V \in \{X, Y, Z, W\} \\
\mathcal{R}_+(S_1) &= \mathbb{N}_{18} \\
\mathcal{R}_+(S_2) &= \mathbb{N}_{27} \\
\mathcal{F}_+^X &= \mathcal{F}_+^Y = \mathcal{F}_+^Z = 0 \\
\forall z \in \mathcal{R}(Z) : \mathcal{F}_+^W(z) &= z \\
\forall (x, y) \in \mathcal{R}(X) \times \mathcal{R}(Y) : \mathcal{F}_+^{S_1}(x, y) &= x + y \\
\forall (s_1, w) \in \mathcal{R}(S_1) \times \mathcal{R}(W) : \mathcal{F}_+^{S_2}(s_1, w) &= s_1 + w
\end{aligned}$$

G.2 Formal Definition of N_+

In the main text, we did not provide a specific identity for N_+. Here, we define N_+ to be a feed forward network, which we represent directly as a causal model $C_{N_+} = (\mathcal{V}_{N_+}, \mathcal{R}_{N_+}, \mathcal{F}_{N_+})$. The location L_1 from Figure 1 is the hidden unit H_3, the location L_2 is the hidden unit H_1.

Let $W \in \mathbb{R}^{30\times 3}$; for $k \in \{1,3\}$ let $W_{jk} = j \bmod 10$ if $0 \leq j \leq 20$, otherwise $W_{jk} = 0$, and let $W_{j2} = 0$ if $0 \leq j \leq 20$, otherwise $W_{j2} = j \bmod 10$. Let $U \in \mathbb{R}^3$ and $U = [1,1,0]$.

$$\mathcal{V}_{N_+} = \{D_x, D_y, D_z, H_1, H_2, H_3, O\}$$

$$\mathcal{R}_{N_+}(D_x) = \mathcal{R}_{N_+}(D_y) = \mathcal{R}_{N_+}(D_z) = \{0,1\}^{10}$$
$$\mathcal{R}_{N_+}(O) = \mathcal{R}_{N_+}(H_1) = \mathcal{R}_{N_+}(H_2) = \mathcal{R}_{N_+}(H_3) = \mathbb{R}$$

$$\mathcal{F}^{D_x}_{N_+} = \mathcal{F}^{D_y}_{N_+} = \mathcal{F}^{D_z}_{N_+} = 0$$
$$\forall \mathbf{x} \in \mathcal{R}_{N_+}(D_x) \times \mathcal{R}_{N_+}(D_y) \times \mathcal{R}_{N_+}(D_z) : [\mathcal{F}^{H_1}_{N_+}(\mathbf{x}), \mathcal{F}^{H_2}_{N_+}(\mathbf{x}), \mathcal{F}^{H_3}_{N_+}(\mathbf{x})] = \text{ReLU}(\mathbf{x}W)$$
$$\forall \mathbf{h} \in \mathcal{R}_{N_+}(H_1) \times \mathcal{R}_{N_+}(H_2) \times \mathcal{R}_{N_+}(H_3) : \mathcal{F}^{O}_{N_+}(\mathbf{h}) = \text{ReLU}(\mathbf{h}U)$$

This network uses one-hot representations $d_x, d_y, d_z \in \{0,1\}^{10}$ to represent inputs from $\mathbb{N}_9$.

G.3 Proving C_+ is an abstraction of N_+

We now prove that C_+ is an abstraction C_{N_+}

We define the mapping $\tau : \mathcal{R}_{N_+}(\mathcal{V}_{N_+}) \to \mathcal{R}_+(\mathcal{V}_+)$ as follows. We first partition the variables of N_+ into cells: $P_X = \{D_x\}$, $P_Y = \{D_y\}$, $P_Z = \{D_z\}$, $P_W = \{H_1\}$, $P_{S_1} = \{H_3\}$, $P_{S_2} = \{O\}$, $P_\varnothing = \{H_2\}$. To define τ it suffices to define the component functions τ_V for $V \in \mathcal{V}_+$. Let $B : \{0,1\}^{10} \to \mathbb{N}_9$ be the partial function s.t. $B([v_1, v_2, \ldots, v_{10}]) = k$ if $v_k = 1$ and $v_j = 0$ for $j \neq k$. Set τ_X, τ_Y, τ_Z all equal to B, and let $\tau_W, \tau_{S_1}, \tau_{S_2}$ all be the identity function.

Let $\mathcal{I}_+$ be the set of all interventions on C_+ that determine values for (at least) X, Y, and Z. Let $\mathcal{I}_{N_+} = \mathit{dom}(\omega_\tau)$. That is, $\mathcal{I}_{N_+}$ includes exactly the (interventions corresponding to) projections of $\mathcal{R}_{N_+}(\mathcal{V}_{N_+})$ that map via ω_τ to some admissible intervention on C_+. Because elements of $\mathcal{I}_+$ always determine values for X, Y, Z, every intervention in $\mathcal{I}_{N_+}$ determines a value for each of D_x, D_y, D_z. In fact, these values are guaranteed to be in the domains of τ_X, τ_Y, τ_Z, respectively.

We now prove the three conditions guaranteeing $(C_+, \mathcal{I}_+)$ is a τ-abstraction of $(C_{N_+}, \mathcal{I}_{N_+})$.

(1) The first point is that the map τ is surjective. Take an arbitrary $(x, y, z, w, s_1, s_2) \in \mathcal{R}_+(\mathcal{V}_+)$. We determine an element of $\mathcal{R}_{N_+}(\mathcal{V}_{N_+})$ as follows: $[d_x d_y d_z] = B^{-1}([x, y, z])$, $[h_1 h_2 h_3] = [s_1 d_2 s_1]$, and $o = s_2$. It's then clear that $\tau(d_x, d_y, d_z, h_1, h_2, h_3, o) = (x, y, z, w, s_1, s_2)$. As (x, y, z, w, s_1, s_2) was chosen arbitrarily, τ is surjective.

(2) The second point is that ω_τ must also surject onto the set $\mathcal{I}_+$ of all interventions on C_+. Any intervention $i_+ \in \mathcal{I}_+$ can be identified with a vector $\mathbf{i}^+$ of values of variables in $\mathcal{V}_+$. By definition of $\mathcal{I}_+$, i_+ fixes at least the values of X, Y, Z. Consider the intervention i_{N_+} that sets D_x, D_y, and D_z to the one-hot representations of X, Y, and Z for the values they were set. Furthermore, if i_+ sets W to w then i_{N_+} sets H_1 to w and if i_+ sets S_2 to s_2, then i_{N_+} sets H_3 to s_2. It suffices to show that $\omega_\tau(i_{N_+}) = i_+$. In other words, we need to show that $\tau(\mathsf{Proj}^{-1}(\mathbf{i}^{N_+})) = \mathsf{Proj}^{-1}(\mathbf{i}^+)$.

First, we show for all $\mathbf{v}_L \in \mathsf{Proj}^{-1}(\mathbf{i}^{N_+})$ that $\tau(\mathbf{v}_L) \in \mathsf{Proj}^{-1}(\mathbf{i}^+)$. By construction of i_+, any variables fixed by i_{N_+} will correspond (via τ component functions) to values of variables fixed by i_+, except for the variable H_3, which has no corresponding high level variable. We merely need to observe that for any values of variables *not* set by i_{N_+}, there exist corresponding values for the variables that are *not* set by i_+, such that the appropriate τ component functions map the former to the latter (with the exception of H_3, which has no corresponding high level variable). This is obvious from the definition of the components of τ.

Second, we show for all $\mathbf{v}_H \in \mathsf{Proj}^{-1}(\mathbf{i}^+)$ there is $\mathbf{v}_L \in \mathsf{Proj}^{-1}(\mathbf{i}^{N_+})$ such that $\tau(\mathbf{v}_L) = \mathbf{v}_H$. Again, by construction of i_+, any variables fixed by i_+ will correspond (via τ component functions) to values of variables fixed by i_{N_+}. We merely need to observe that for any values of variables *not* set by i_+, there exist corresponding values for the variables *not* set by i_{N_+}, such that the appropriate τ

component functions map the former to the latter, with H_3 taking on any value. This is obvious from the definition of the components of τ. This concludes the argument that $\omega_\tau(i_{N_+}) = i_+$.

(3) Finally, we need to show for each $i_{N_+} \in \mathit{dom}(\omega_\tau)$ that $\tau(i_{N_+}(C_{N_+})) = \omega_\tau(i_{N_+})(C_+)$. The point here is that the two causal processes unfold in the same way, under any intervention.

Indeed, pick any i_{N_+} and suppose that $i_+ = \omega_\tau(i_{N_+})$. We know that i_+ fixes values x, y, z of X, Y, Z, and likewise that i_{N_+} fixes values d_x, d_y, d_z of D_x, D_y, D_z such that $\tau_{D_j}(x_j) = d_j$ for $j \in \{1, 2, 3\}$. Any other variables fixed by i_+ from among W, S_1, S_2 will likewise correspond (via $\tau_W, \tau_{S_1}, \tau_{S_2}$) to values of H_2, H_1, O fixed by i_{N_+}. We merely need to observe that any variables that are *not* set by i_+ and i_{N_+} will still correspond via the appropriate τ-component, given their settings in $i_+(C_+)$ and $i_{N_+}(C_{N_+})$. The mechanisms in C_{N_+} were devised precisely to guarantee this.

Thus we have fulfilled the three requirements and we have shown that C_+ is an abstraction C_{N_+}.

The proof that C_{NatLog} is a constructive abstraction of N_{NLI} follows this same pattern.

H Causal Abstraction Analysis of C_{NatLog}

H.1 Formal Definition of C_{NatLog}

We formally define the model $C_{\mathit{NatLog}} = (\mathcal{V}_{\mathit{NatLog}}, \mathcal{R}_{\mathit{NatLog}}, \mathcal{F}_{\mathit{NatLog}})$ as follows:

$$\mathcal{V}_{\mathit{NatLog}} = \left\{ \begin{array}{c} \mathrm{Q}^P_{\mathit{Subj}}, \mathrm{Q}^H_{\mathit{Subj}}, \mathrm{Neg}^P_{\mathit{Subj}}, \mathrm{Neg}^H_{\mathit{Subj}}, \mathrm{N}^P_{\mathit{Subj}}, \mathrm{N}^H_{\mathit{Subj}}, \mathrm{Neg}^P, \mathrm{Neg}^H, \mathrm{Adv}^P, \mathrm{Adv}^H, \\ \mathrm{V}^P, \mathrm{V}^H, \mathrm{Q}^P_{\mathit{Obj}}, \mathrm{Q}^H_{\mathit{Obj}}, \mathrm{Neg}^P_{\mathit{Obj}}, \mathrm{Neg}^H_{\mathit{Obj}}, \mathrm{N}^P_{\mathit{Obj}}, \mathrm{N}^H_{\mathit{Obj}}, Q_{\mathit{Subj}}, \mathrm{Neg}_{\mathit{Subj}}, N_{\mathit{Subj}}, \mathrm{Neg}, \mathrm{Adv} \\ Q_{\mathit{Obj}}, \mathrm{Neg}_{\mathit{Obj}}, \mathrm{N}_{\mathit{Obj}}, \mathrm{NP}_{\mathit{Subj}}, \mathrm{VP}, \mathrm{NP}_{\mathit{Obj}}, \mathrm{QP}_{\mathit{Obj}}, \mathrm{NegP}, \mathrm{QP}_{\mathit{Subj}} \end{array} \right\}$$

$$\begin{aligned}
\mathcal{R}_{\mathit{NatLog}}(\mathrm{Q}^P_{\mathit{Subj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{Q}^H_{\mathit{Subj}}) = \mathcal{R}_{\mathit{NatLog}}(\mathrm{Q}^H_{\mathit{Subj}}) = \mathcal{R}_{\mathit{NatLog}}(\mathrm{Q}^H_{\mathit{Subj}}) \\
&= \{\textit{no, some, every, not every}\} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}^P) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}^H) = \{not, \epsilon\} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}^P_{\mathit{Subj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}^H_{\mathit{Subj}}) = \mathrm{Neg}_{\mathit{Subj}} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{N}^P_{\mathit{Subj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{N}^H_{\mathit{Subj}}) = \mathbf{N}_{\mathit{Subj}} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{Adv}^P) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{Adv}^H) = \mathrm{Adv}_{\mathit{Subj}} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{V}^P) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{V}^H) = \mathbf{V}_{\mathit{Subj}} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}^P_{\mathit{Obj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}^H_{\mathit{Obj}}) = \mathrm{Neg}_{\mathit{Obj}} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{N}^P_{\mathit{Obj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{N}^H_{\mathit{Obj}}) = \mathbf{N}_{\mathit{Obj}} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{Q}_{\mathit{Obj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{Q}_{\mathit{Subj}}) = \mathcal{Q} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}) &= \mathcal{N} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}_{\mathit{Obj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{Neg}_{\mathit{Subj}}) = \mathcal{R}_{\mathit{NatLog}}(\mathrm{Adv}) = \mathcal{A} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{N}_{\mathit{Obj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{N}_{\mathit{Subj}}) = \mathcal{R}_{\mathit{NatLog}}(\mathrm{V}) = \{\#, \equiv\} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{NP}_{\mathit{Subj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{NP}_{\mathit{Obj}}) = \mathcal{R}_{\mathit{NatLog}}(\mathrm{VP}) = \{\#, \equiv, \sqsubset, \sqsupset\} \\
\mathcal{R}_{\mathit{NatLog}}(\mathrm{QP}_{\mathit{Obj}}) &= \mathcal{R}_{\mathit{NatLog}}(\mathrm{NegP}) = \mathcal{R}_{\mathit{NatLog}}(\mathrm{QP}_{\mathit{Subj}}) = \{\#, \equiv, \sqsubset, \sqsupset, |, \mathbin{\hat{}}, \smile\}
\end{aligned}$$

$$\begin{aligned}
\mathcal{F}_{\mathrm{N}} &= \mathrm{COMP\ for\ N} \in \{\mathrm{VP}, \mathrm{NP}_{\mathit{Subj}}, \mathrm{NP}_{\mathit{Obj}}, \mathrm{NegP}, \mathrm{QP}_{\mathit{Obj}}, \mathrm{QP}_{\mathit{Subj}}\} \\
\mathcal{F}_{\mathrm{N}} &= \mathrm{REL\ for\ N} \in \{\mathrm{V}, \mathrm{N}_{\mathit{Subj}}, \mathrm{N}_{\mathit{Obj}}\} \\
\mathcal{F}_{\mathrm{N}} &= \mathrm{PROJ\ for\ N} \in \{\mathrm{Q}_{\mathit{Obj}}, \mathrm{Q}_{\mathit{Subj}}, \mathrm{Adv}, \mathrm{Neg}_{\mathit{Subj}}, \mathrm{Neg}_{\mathit{Obj}}, \mathrm{Neg}\}
\end{aligned}$$

The set $\{\#, \equiv, \sqsubset, \sqsupset, |, \mathbin{\hat{}}, \smile\}$ contains the seven relations used in the natural logic of MacCartney and Manning [17]. The set $\mathbf{N}_{\mathit{Subj}}$ contains the subject nouns used to create MQNLI, $\mathbf{N}_{\mathit{Obj}}$ the set of object nouns, $\mathbf{Adj}_{\mathit{Subj}}$ the subject adjectives, $\mathbf{Adj}_{\mathit{Obj}}$ the object adjectives, $\mathbf{V}$ the verbs, and Adv the adverbs. Additionally, $\mathcal{Q}$ is the set of joint projectivity signatures between *every*, *some*, *not every*, and *no*, $\mathcal{N}$ is the set of joint projectivity signatures between *not* and ϵ, $\mathcal{A}$ is the set of joint projectivity signatures between intersective adjectives and adverbs and ϵ. $\mathrm{REL}(x, y)$ outputs the lexical relation

between x and y. Finally, $\text{COMP}(f, x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n)$ and $\text{PROJ}(f, g) = P_{f/g}$ where $P_{f/g}$ is the joint projectivity signature between f and g. See Geiger et al. [10] for details about these sets and functions.

H.2 Formal Definition of $C^N_{\textit{NatLog}}$

For some non-leaf node N of the tree in Figure 2a, we define $C^N_{\textit{NatLog}}$ to be the marginalization of $C_{\textit{NatLog}}$ where all variables are removed other than the input variables

$$\mathcal{V}^{\textit{Input}}_{\textit{NatLog}} = \text{Q}^P_{\textit{Subj}}, \text{Q}^H_{\textit{Subj}}, \text{Neg}^P_{\textit{Subj}}, \text{Neg}^H_{\textit{Subj}}, \text{N}^P_{\textit{Subj}}, \text{N}^H_{\textit{Subj}}, \text{Neg}^P, \text{Neg}^H, \text{Adv}^P, \text{Adv}^H, \text{V}^P, \text{V}^H,$$
$$\text{Q}^P_{\textit{Obj}}, \text{Q}^H_{\textit{Obj}}, \text{Neg}^P_{\textit{Obj}}, \text{Neg}^H_{\textit{Obj}}, \text{N}^P_{\textit{Obj}}, \text{N}^H_{\textit{Obj}}$$

along with the output variable $\text{QP}_{\textit{Subj}}$ and the intermediate variable N. For a definition of marginalization, see Bongers et al. [4].

H.3 Formal definition of $N_{\textit{NLI}}$

In the main text, $N_{\textit{NLI}}$ could represent either our BERT model or our LSTM model. We will maintain this ambiguity, because while these two models are drastically different at the highest level of detail, for the sake of our analysis we can view them both as creating a grid of neural representations where each representation in the grid is caused by all representations in the previous row and causes all representations in the following row. We will now formally define the causal model $C_{N_{\textit{NLI}}}$.

$$\mathcal{V}_{N_{\textit{NLI}}} = \{R_{11}, R_{12}, \ldots, R_{1m}, \ldots R_{nm}, O\}$$

For the LSTM model $n = 2$ and for the BERT model $n = 12$. m is the number of tokens in a tokenized version of an MQNLI example.

$$\mathcal{R}_{N_{\textit{NLI}}}(R_{jk}) = \mathbb{R}^d \quad \mathcal{R}_{N_{\textit{NLI}}}(O) = \{\text{entailment, contradiction, neutral}\}$$

For all j and k and where d is the dimension of the vector representations.

$$\forall (r_{(j-1)1}, r_{(j-1)2}, \ldots, r_{(j-1)m}) \in \mathcal{R}_{N_{\textit{NLI}}}(R_{(j-1)1} \times R_{(j-1)2} \times \cdots \times R_{(j-3)m})$$
$$\mathcal{F}^{R_{jk}}_{N_{\textit{NLI}}}(r_{(j-1)1}, r_{(j-1)2}, \ldots, r_{(j-1)m}) = \mathbf{NN}_{jk}(r_{(j-1)1}, r_{(j-1)2}, \ldots, r_{(j-1)m})$$

where $\mathbf{NN}_{jk}$ is either the LSTM function or the BERT function that creates the neural representation at the jth row and kth column.

$$\forall r_{n1} \in \mathcal{R}_{N_{\textit{NLI}}}(R_{n1}) \mathcal{F}^{O}_{N_{\textit{NLI}}}(r_{n1}) = \mathbf{NN}_O(r_{n1})$$

where $\mathbf{NN}_O$ is the neural network that makes a three class prediction using the final representation of the [CLS] token.

See Appendix B for details about these functions.

H.4 Proving $C^N_{\textit{NatLog}}$ is an abstraction of $N_{\textit{NLI}}$

We will now formally prove that that $C^N_{\textit{NatLog}}$ is a constructive abstraction of $N_{\textit{NLI}}$ if the following holds for all $e, e' \in$ MQNLI, where the representation location L is equivalent to the variable R_{jk} for some j and k. This would mean that every single one of our intervention experiments at this location are successful.

$$C^{N \leftarrow e'}_{\textit{NatLog}}(e) = N^{L \leftarrow e'}_{\textit{NLI}}(e) \tag{8}$$

We define the mapping $\tau : \mathcal{R}_{N_{\textit{NLI}}}(\mathcal{V}_{N_{\textit{NLI}}}) \rightarrow \mathcal{R}^N_{\textit{NatLog}}(\mathcal{V}_{\textit{NatLog}})$ as follows. We first partition the "low level" variables of $N_{\textit{NLI}}$ into partition cells:

$$P_N = \{L\} \qquad P_{\text{QP}_{\textit{Subj}}} = \{O\} \quad \forall X \in \mathcal{V}^{\textit{Input}}_{\textit{NatLog}}$$
$$P_X = \{R_{1j}, R_{1(j+1)}, \ldots, R_{1(j+k)}\}$$

where $R_{1j}, R_{1(j+1)}, \ldots, R_{1(j+k)}$ are the token vectors associated with the input variable X. Some of our causal model's input variables are tokenized into several tokens (see Appendix B for details).

To define τ, it then suffices to define the component functions τ_V for each $V \in \mathcal{V}_{\textit{NatLog}}$. Let $T : (\mathbb{R}^d)^+ \to \mathcal{V}^{\textit{Input}}_{\textit{NatLog}}$ be the partial function mapping sequences of token vectors to the input variable they correspond to, where $+$ is the Kleene plus operator. Let $P : \mathcal{R}^3 \to \{\text{entailment, neutral, contradiction}\}$ be the partial function mapping a vector of logits to the output prediction they correspond to. Finally, let $Q_L : \mathbb{R}^d \to \mathcal{R}_{\textit{NatLog}}(N)$ be the partial function such that for all $e \in$ MQNLI, if $\mathbf{v}$ is the vector created by $N_{\textit{NLI}}$ at location L when processing input e and x is the value realized by $C_{\textit{NatLog}}$ for the variable N when processing input e, then $Q_L(\mathbf{v}) = x$.

For all $\forall X \in \mathcal{V}^{\textit{Input}}_{\textit{NatLog}}$, we set τ_X to be T. We additionally set τ_N to be Q_L and $\tau_{\text{QP}_{\textit{Subj}}}$ to be P.

Let $\mathcal{I}_{\textit{NatLog}}$ be the set of all interventions on $C_{\textit{NatLog}}$ that intervene on (i.e., determine the values for) at least the elements of $\mathcal{V}^{\textit{Input}}_{\textit{NatLog}}$. Let $\mathcal{I}_{N_{\textit{NLI}}}$ be the set of interventions that is the domain of the partial function ω_τ. In other words, $\mathcal{I}_{N_{\textit{NLI}}}$ includes exactly the projections of $\mathcal{R}_{N_{\textit{NLI}}}(\mathcal{V}_{N_{\textit{NLI}}})$ that map via ω_τ to some intervention on C_+. The fact that P, Q_L, an T are all proper partial functions prevent $\mathcal{I}_{N_{\textit{NLI}}}$ from including all possible interventions on $C_{N_{\textit{NLI}}}$.

We now prove the three conditions that must hold for $(C_{\textit{NatLog}}, \mathcal{I}_{\textit{NatLog}})$ to be a τ-abstraction of $(C_{N_{\textit{NLI}}}, \mathcal{I}_{N_{\textit{NLI}}})$.

(1) The first point is to show the map τ is surjective. So take an arbitrary element $(\vec{v}^{\textit{input}}, n, q) \in \mathcal{R}_{\textit{NatLog}}(\mathcal{V}_{\textit{NatLog}})$. We specify an element of $\mathcal{R}_{N_{\textit{NLI}}}(\mathcal{V}_{N_{\textit{NLI}}})$ as follows:

$$l = Q_L^{-1}(n) \qquad\qquad o = P^{-1}(q)$$

$$\forall v^{\textit{input}} \in \vec{v}^{\textit{input}} T^{-1}(v^{\textit{input}}) = (r_{1j}, r_{1(j+1)}, \ldots, r_{1(j+k)})$$

where $r_{1j}, r_{1(j+1)}, \ldots, r_{1(j+k)}$ are the token vectors corresponding to the input variable $v^{\textit{input}}$.

It's then patent that $\tau(r_{11}, \ldots, r_{n1}, r_{12}, \ldots r_{nm}, o) = (\vec{v}^{\textit{input}}, n, q)$. As $(\vec{v}^{\textit{input}}, n, q)$ was chosen arbitrarily, we have shown τ is surjective.

(2) The second point is that ω_τ must also be surjective onto the set $\mathcal{I}_{\textit{NatLog}}$ of interventions on $C_{\textit{NatLog}}$. Any intervention $i_{\textit{NatLog}} \in \mathcal{I}_{\textit{NatLog}}$ can be identified with with a vector $\mathbf{i}^{\textit{NatLog}}$ of values of variables in $\mathcal{V}_{\textit{NatLog}}$. By the definition of $\mathcal{I}_{\textit{NatLog}}$, $i_{\textit{NatLog}}$ fixes the values of the variables in $\mathbf{V}^{input}$ and may also determine N and/or QP_{Subj}. Consider the intervention $i_{N_{\textit{NLI}}}$ corresponding to $\mathbf{i}^{\mathbf{N}_{\textit{NLI}}} = \tau^{-1}(\mathbf{i}^{\textit{NatLog}})$ as described in Section F.2. It suffices to show that $\omega_\tau(i_{N_{\textit{NLI}}}) = i_{\textit{NatLog}}$. In other words, we need to show parts 1, 2, and 3 from the definition above.

Part 1 is clear, since by the definition of $\mathcal{I}_{\textit{NatLog}}$ we are guaranteed that $\mathbf{i}^{\textit{NatLog}}$ determines values for $\mathbf{V}^{\textit{input}}$, and hence $\mathbf{i}^{N_{\textit{NLI}}}$ fixes values for $R_{11}, \ldots, R_{1m}$ in the domains of $\tau_{\mathbf{V}^{\textit{input}}}$ for $V \in \mathbf{V}^{\textit{input}}$. Then any intervention that intervenes only on the values of

Part 2 requires that for every $\mathbf{v}_{N_{\textit{NLI}}} \in \mathsf{Proj}^{-1}(\mathbf{i}^{N_{\textit{NLI}}})$, we have $\tau(\mathbf{v}_{N_{\textit{NLI}}}) \in \mathsf{Proj}^{-1}(\mathbf{i}^{\textit{NatLog}})$. Because of how we defined $i_{\textit{NatLog}}$, any variables fixed by $i_{N_{\textit{NLI}}}$ will correspond (via τ component functions) to values of variables fixed by $i_{\textit{NatLog}}$, except for the variables $R_{jk} \notin \mathbf{V}^{\textit{input}} \cup \{L\}$, which have no corresponding high level variables. We merely need to observe that, for any values for the variables that are *not* set by $i_{N_{\textit{NLI}}}$, there exists corresponding values for the variables that are *not* set by $i_{\textit{NatLog}}$ such that the appropriate τ component functions map the former to the latter, except for the variables $R_{jk} \notin \mathbf{V}^{input} \cup \{L\}$, which, again, have no corresponding high level variables. This is plainly obvious from the definition of the components of τ.

Part 3 requires that for any $\mathbf{v}_{\textit{NatLog}} \in \mathsf{Proj}^{-1}(\mathbf{i}^{\textit{NatLog}})$, there exists a $\mathbf{v}_{N_{\textit{NLI}}} \in \mathsf{Proj}^{-1}(\mathbf{i}^{N_{\textit{NLI}}})$ such that $\tau(\mathbf{v}_{N_{\textit{NLI}}}) = \mathbf{v}_{\textit{NatLog}}$. Again, because of how we defined $i_{\textit{NatLog}}$, any variables fixed by $i_{\textit{NatLog}}$ will correspond (via τ component functions) to values of variables fixed by $i_{N_{\textit{NLI}}}$. We merely need to observe that for any values for the variables that are *not* set by $i_{\textit{NatLog}}$, there exists corresponding values for the variables that are *not* set by $i_{N_{\textit{NLI}}}$, such that the appropriate τ component functions map the former to the latter, with $R_{jk} \notin \mathbf{V}^{\textit{input}} \cup \{L\}$ taking on any value. This is plainly obvious from the definition of the components of τ.

Thus, we have shown that $\omega_\tau(i_{N_{NLI}}) = i_{NatLog}$.

(3) Finally, we need to show for each $i_{N_{NLI}} \in dom(\omega_\tau)$ that $\tau(i_{N_{NLI}}(C_{N_{NLI}})) = \omega_\tau(i_{N_{NLI}})(C_{NatLog})$. The point here is that the two causal processes unfold in the same way, under any intervention. Indeed, pick any $i_{N_{NLI}}$ and suppose that $i_{NatLog} = \omega_\tau(i_{N_{NLI}})$. We know that i_{NatLog} fixes values for the variables in $\mathbf{V}^{input}$, and likewise that $i_{N_{NLI}}$ fixes values for the variables $R_{11}, \ldots, R_{1m}$. Any other variables fixed by i_{NatLog} from among N, QP_{Subj} will likewise correspond (via the component functions of τ) to values of L and O. We merely need to observe that any variables that are *not* set by i_{NatLog} and $i_{N_{NLI}}$ will still correspond via the appropriate τ-component, given their settings in $i_{NatLog}(C_{NatLog})$ and $i_{N_{NLI}}(C_{N_{NLI}})$. The intervention experiments on N_{NLI} that we are assuming were successful were devised precisely to guarantee this.

We have thus fulfilled the three requirements and shown that C_{NatLog} is an abstraction of $C_{N_{NLI}}$.