Unit Testing for Concepts in Neural Networks

Charles Lovering
Department of Computer Science
Brown University, USA
charles_lovering@brown.edu

Ellie Pavlick
Department of Computer Science
Brown University, USA
ellie_pavlick@brown.edu

Abstract

Many complex problems are naturally understood in terms of symbolic concepts. For example, our concept of *''cat''* is related to our concepts of *''ears''* and *''whiskers''* in a non-arbitrary way. Fodor (1998) proposes one theory of concepts, which emphasizes symbolic representations related via constituency structures. Whether neural networks are consistent with such a theory is open for debate. We propose unit tests for evaluating whether a system's behavior is consistent with several key aspects of Fodor's criteria. Using a simple visual concept learning task, we evaluate several modern neural architectures against this specification. We find that models succeed on tests of groundedness, modularity, and reusability of concepts, but that important questions about causality remain open. Resolving these will require new methods for analyzing models' internal states.

1 Introduction

Understanding language requires having representations of the world to which language refers. Prevailing theories in linguistics and cognitive science hold that these representations, or *concepts*, are structured in a compositional way—for example, the concept of *''car''* can be combined with other concepts (*''gray''*, *''new''*)—and that the meanings of composite concepts (*''gray car''*) are inherited predictably from the meanings of the parts. State-of-the-art models for natural language processing (NLP) use neural networks (NNs), in which internal representations are points in high-dimensional space. Whether such representations can in principle reflect the abstract symbolic structure presupposed by theories of human language and cognition is an open debate. This paper maintains that the question of whether a model contains the desired type of symbolic conceptual representations is best answered at the *computation level* (Marr, 2010): that is, the diagnostics of ''symbolic concepts'' concern what a system does and why, rather than the details of how that behavior is achieved (e.g., whether it stores vectors vs. explicit symbols ''on disk''). Even Fodor and Pylyshyn (1988), in their vocal criticism of NNs, assert that ''a connectionist neural network can perfectly well *implement* a classical architecture at the cognitive level'',[1] but do not say how to know if such an implementation has been realized.

To this end, we propose an API-level specification based on criteria of ''what concepts have to be'' (Fodor, 1998). Our specification (§3) defines the required behaviors and operations, but is agnostic about implementation. We then consider fully connectionist systems equipped with modern evaluation methods (e.g., counterfactual perturbations, probing classifiers) as candidate systems. We present evidence that the evaluated models learn conceptual representations that meet a number of the key criteria (§5–§7) but fail on those related to causality (§8–§9). We argue that more powerful tools for analyzing NNs' internal states may be sufficient to close this gap (§10). Overall, our primary contribution is a framework for seeking converging evidence from multiple evaluation techniques in order to determine whether modern neural models are consistent with a specific theory of concepts. Our experiments offer an updated perspective in the debate about whether neural networks can serve as the substrate of a linguistically competent system.

2 ''What Concepts Have To Be''

2.1 Criteria

There is no single agreed-upon standard for what ''concepts'' are (Margolis et al., 1999). We base

[1] Italics added. Here, classical architecture = symbolic architecture, and cognitive level = computational level.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 1193–1208, 2022. https://doi.org/10.1162/tacl_a_00514
Action Editor: Katrin Erk. Submission batch: 5/2022; Revision batch: 6/2022; Published 11/2022.

our criteria on those put forth in Fodor (1998) as part of a theory which advocates symbolic representations and prioritizes explaining phenomena such as syntactic productivity and semantic compositionality. Fodor (1998) argues for five conditions required for a conceptual representation to be viable as a model of human-level cognition: **C1:** ''Concepts are mental[2] particulars; specifically, they...function as mental causes and effects''; **C2:** ''Concepts...apply to things in the world; things in the world ''fall under them''; **C3:** ''Concepts are constituents of thoughts and... of one another. Mental representations inherit their contents from the contents of their constituents''; **C4:** ''Quite a lot of concepts [are] learned''; **C5:** ''Concepts are public...to say that two people share a concept [means] they have tokens of literally the same concept type.''

2.2 Assumptions and Limitations

We focus on Fodor's (1998) criteria since they are concordant with ideas from formal linguistics which have recently been highlighted as weaknesses of NNs (Pavlick, 2022). We don't claim that Fodor's theories should necessarily serve as the standard for NLP systems (indeed, his theories face criticisms). The subset of Fodor's criteria on which we focus (§3) are fairly uncontroversial, and arguably would transfer to alternative theories of conceptual structure—for example, Bayesian causal models (Sloman, 2005). We view our tests as necessary but alone insufficient to meet Fodor's criteria. For example, our composite concepts depend on simple conjunction and thus do not address issues about constituency structure in which the argument order matters. Even so, our results offer a valuable starting point on which subsequent theoretical and empirical work can build.

[2]''Mental'' here implies that the representations are divorceable from the external world. One can token a concept in the absence of relevant perceptual stimuli. For example, thinking ''If it were raining...'' entails thinking about ''raining'' precisely when it is *not* raining. This distinction is subtle but important. Our unit tests operationalize this via the fact, in 3 out of 4 tests, the perceptual input is held fixed and the intervention is applied to internal state. This is only a first step. Future work will need to explore this issue in more detail to determine what type of perceptual-conceptual distinction suffices to meet this criterion, and how it can be demonstrated empirically.

3 System Specification

We translate key ideas from Fodor's conditions into concrete unit tests for evaluating computational models. Our mapping is not one-to-one: We combine C2 and C5 into a single test focused on whether a concept grounds consistently to perception; we split C3 into two tests and leave aspects to future work; we omit C4 since there is likely little controversy that modern NLP systems ''learn'' concepts. Our tests apply to a system holistically, including implementations of diagnostic functions, not just the internal representations. Thus, it is possible for one system to fail our tests, but for a different system with the same internal representation but different implementations of the functions to succeed. See discussions in §4.1.1 and §10.

3.1 Data Types and Basic Functions

Our domain consists of things in the perceptual world (`type X`) to which humans assign discrete words (`type Y`). We follow Fodor (1998) in treating word meaning and concepts as interchangeable.[3] Internal concepts may be either *atomic* (without an internal structure) or *composite*, which, in our setting, means they obey a simple conjunctive syntax over atomic constituents (e.g., *''ice''*$\models$*''water''*&*''solid''*). We assume two ground-truth functions: `gt_label` which returns the name for a given thing, and `gt_describe` which describes a composite concept (type `Y`) in terms of its constituents (type `Set[Y]`).

We require that the system supports an `encode` operation to map `X` to an internal representation of type `Z`, as well as a `predict` operation to map `Z` to `Y`. We also require that the system implements two diagnostic functions, that is, functions unnecessary for the system's usual operations (here, assigning words to inputs), but necessary for measuring properties of the system's internal structure. `has_concept` returns true if the system considers the internal representation (`Z`) to encode a concept (`Y`); `ablate` removes the part of the internal representation considered to encode the concept.

[3]This is a common assumption. Of course, in reality, there are things for which humans may have a concept but do not have the ability to express precisely in language.

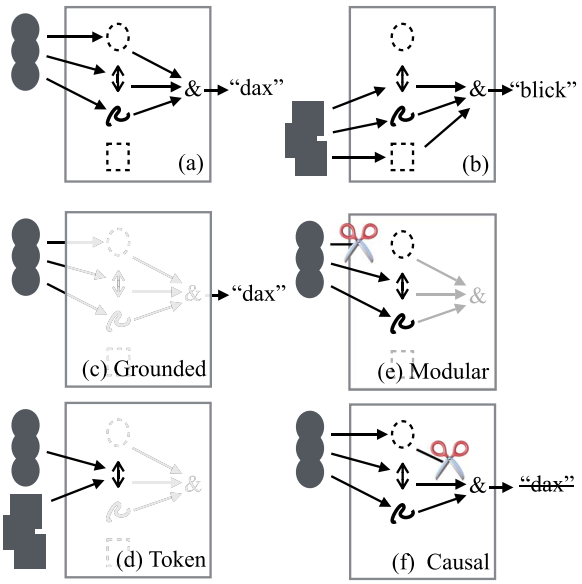

Figure 1: (a) and (b): Visualization of unit tests as operations on a symbolic graphical model (c): Changes in input features lead to expected changes in output. (d): Internal nodes are reused across tokens of the same type. (e): Removing one internal concept does not damage others. (f): Removing internal concepts impacts the model's predictions.

3.2 Unit Tests

Our specification requires not only that a system supports the above operations, but that its implementation obeys certain constraints, which we formalize via unit tests. Intuitively, it is helpful to think of these unit tests by picturing a symbolic system (e.g., a graphical model) which would pass the tests by construction (Figure 1). In practice, we run our tests on NNs, not graphical models. However, if the models pass our tests, the implication is that the NN has implemented something that, for our purposes, is functionally equivalent to the symbolic model shown in Figure 1.

3.2.1 `is_grounded`

Our test `is_grounded` (analyzed in §5) is derived from the requirements that internal concepts are tied to the external world (C2) in a way that is shared (C5).[4] Our test requires that models respond to changes in perceptual inputs in the same way that an (idealized) human would respond to those changes, namely, that `predict(encode(x)) == gt_label(x)`. Effectively, this test simply requires that a model performs well on the labeling task, but does not care about the representations involved in producing those labels.

3.2.2 `is_token_of_type`

C3 requires that concepts have constituency structure. We define two tests which probe aspects of this requirement (see §2.2 for caveats).

First, `is_token_of_type` (evaluated in §6) tests whether different token instances of a concept evaluate to the same semantic type. Fodor and Pylyshyn (1988) claim this property is required for systematicity and compositionality, arguing that the inference *''Turtles are slower than rabbits$_1$''; ''Rabbits$_2$ are slower than Ferraris''*→*''Turtles are slower than Ferraris''* only follows if, among other things, *''rabbits$_1$''* is treated as the same as (not merely ''similar to'') *''rabbits$_2$''*. We thus require that there exists a computational procedure for mapping models' internal representations into a discrete space, and that this procedure applies in the same way to all token instances of a concept. Concretely, $\forall$`c` $\in$ `gt_describe(gt_label(x))` we require that `has_concept(encode(x), c)`.

3.2.3 `is_modular`

Second, `is_modular` (evaluated in §7) is based on requirements for productivity; for example, for an NP (e.g., *''John''*) to fit into arbitrarily many contexts (*''John loves Mary''*, *''Joe loves John''*), the representation of the NP must be fully disentanglable from the other words and syntax. We frame this requirement as a test of whether representations support ''slot filling''. That is, given a representation of a composite concept, removal of one constituent concept should produce an unfilled ''slot'' but otherwise

[4] `is_grounded` tests whether changes in the input lead to changes in the model's behavior. This is different from Fodor's criteria, which require that the *concept*—i.e., the internal representation–is grounded, and that the representation (not necessarily the behavior) changes in response to external features. We can make this shift because (1) our models' behavior is by definition a function of its internal representation and (2) our test, `is__causal`, requires that changes in behavior are explained by changes in the internal representation. Thus success on both `is_causal` and `is__grounded` entails Fodor's criteria that those things which serve as mental causes and effects are grounded. However, it is plausible that other models could pass using a ''loophole'' in which the behavior is grounded but the internal concepts are not, or could fail due to a technicality in which the representation changes but the model ''decides'' not to change it behavior (though the latter assumes a highly competent system, see Block [1981]).

leave the remaining constituent concepts intact, namely, *''_ loves Mary''*. Concretely, given `z = ablate(encode(x), y)`, we require that `has_concept(z, y)` is false, and that $\forall c \in$ `gt_describe(gt_label(x))` such that $c \neq$ `y`, `has_concept(z, c)` is true.

3.2.4 `is_causal`

Finally, `is_causal` (evaluated in §8) checks that C1 is met by testing that internal conceptual representations themselves serve as ''mental causes and effects''. As in Fodor and Pylyshyn (1988), ''state transitions in Classical machines are causally determined by *the structure–including the constituent structure–of the symbol arrays that the machines transform*: change the symbols and the system behaves quite differently''. To operationalize this, we consider the case in which a model's behavior (e.g., its use of a label) is assumed to be in response to having tokened a composite concept 'A&B'. We require that changes in the representation, such that the constituent concept 'A' is no longer tokened (or, that the constituent concept which is tokened is no longer labeled as type 'A'), result in corresponding changes in the model's behavior. In practice, this amounts to requiring that ablating a constituent concept results in expected degradation in model performance. That is, `predict(ablate(encode(x),c))` should perform at chance if `c` is a constituent of `gt_label(x)`, and should perform equivalently to `predict(encode(x))` otherwise.

4 Implementation

Our code, data, and results are available at: `bit.ly/unit-concepts-drive`.

4.1 Functions

We implement **`encode`** with five different models: three pretrained and two from-scratch.[5] For pretrained models, we use a residual network trained over ImageNet (RN_{IMG}) (He et al., 2016) and two architectures from CLIP Radford et al., 2021—a vision transformer (ViT_{CLIP}) (Dosovitskiy et al., 2020) and a residual network (RN_{CLIP}). For from-scratch models, we use a randomly initialized residual network model (RN_{NoPre}) and a CNN model[6] ($\text{CNN}_{\text{NoPre}}$). We use the pretrained encoders with no additional training. For the other models, we finetune on a classification task on our data.

To implement **`predict`**, we train linear ''probing classifiers'' (Sinha et al., 2021) over the outputs of `encode` using the Adam optimizer (Kingma and Ba, 2014). **`has_concept`** is also implemented with linear classifiers. Thus, our system considers the output of `encode` to ''have'' a concept if a probing model can learn to discriminate instances according to the concept.

To implement **`ablate`**, we use Iterative Nullspace Linear Projection (INLP) (Ravfogel et al., 2020), which repeatedly collapses directions that linearly separate the instances of one concept from those of another. INLP has been used to remove concepts like parts of speech from word representations (Elazar et al., 2020).

4.1.1 Limitations

We make a few important simplifying assumptions in our implementations, which are necessary in order to employ the available analysis tools at the time of writing. First, since INLP—our implementation for `ablate`—only removes linear information, we restrict our implementations of `predict` and `has_concept` to be linear models. However, since writing, new methods have been introduced which could in principle be used in place of INLP in our experiments, and would likely yield different results. We discuss possible implications in §8.3.

Second, in most experiments, we treat the `encode` function as a block, only analyzing its outputs, rather than ablating concepts in its internal layers. However, looking at individual layers could tell a different story. We provide initial results in §9, but a complete investigation warrants significant experiments and is left for future work.

Finally, INLP is iterative, each step removing a direction from the input representation. Our experiments report the results after the first iteration of INLP, as it removes the most salient direction of the concept. Again, future work may find insights in analyzing the removal of subsequent directions.

[5] We do claim pretraining is analogous to human learning. Success on our tests is interesting because it provides an existence proof of *one particular* recipe by which the desired representations arise. This is similar to work on syntax in LMs (Linzen and Baroni, 2021), which is valuable despite LM training being very different from how children learn.

[6] Four layers: filters=(64, 32, 16, 8), kernels=3, stride=2; batch norm (Ioffe and Szegedy, 2015) and ReLU activations.

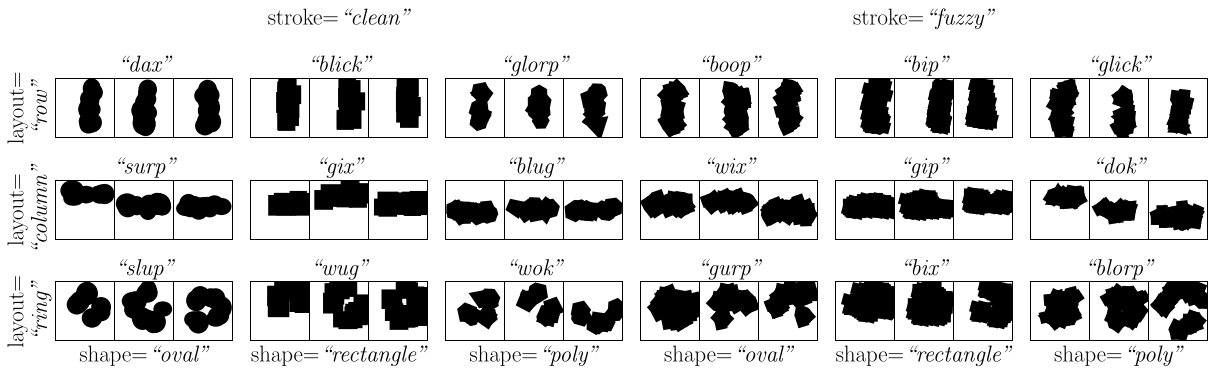

Figure 2: Dataset. Three samples from each class; the right nine classes have fuzzy borders (although this admittedly hard to see in these small images).

4.2 Dataset

4.2.1 Description

Our **default dataset** is a synthetic image[7] dataset with 1000 training examples of each of 18 classes, where each class is composed of three from a set of eight **atomic concepts** {3 layouts: horizontal, vertical, ring} x {3 shapes: rectangle, oval, polygon} x {2 strokes: clean, fuzzy}. Thus, each class is a **composite concept** made up of three constituent atomic concepts. See Figure 2 for examples.[8]

We also create a **colors dataset** in which the color of the shapes is correlated with the class label. We do this because, in `is_grounded` (§5), we find very strong results in the default setting and want to better understand the conditions under which those results hold. The colors dataset emulates a situation where there are spurious features, making it more difficult for a model to ground to the correct perceptual inputs. This dataset is not directly tied to any of Fodor's criteria, but allows us get a more nuanced understanding of our `is_grounded` results. Here, each of the 18 classes is correlated with a different color, such that for $p \in \{\text{RAND}, 90, 99, 100\}$, a given instance has probability p of expressing that paired color, with remaining $1-p$ probability distributed uniformly over the other colors. RAND $= 5.6\%$ (i.e., $1/18$).

4.2.2 Seen and Unseen Examples

To test the generality of a model's representations, we train the diagnostic functions `has_concept` and `ablate` on a subset of the full 18 classes. We define **slice** to mean a set of composite concepts that share the same atomic concepts except along a given dimension. For instance, *''dax''*, *''surp''*, *''slup''* form a slice that delineates layout (i.e., the classes differ in layout but otherwise are the same in terms of shape and stroke). All classes that the diagnostic functions are trained on are considered **seen** and the other classes are considered **unseen**. We experiment with two training settings, which, like the colors dataset, are not directly tied to Fodor's criteria, but which allow us to tell a more nuanced story about what it takes for models to pass our tests. In the first setting (**1 slice**), the probes used to implement `has_concept` are trained on a dataset with one class per concept. So, in this setting, instances that fall under the concept *''horizontal''* would all be drawn from *''dax''*. In the second setting (**N-1 slices**), probes are trained on many classes per concept. Here, instances of *''horizontal''* would be drawn from several classes (*''blick''*, *''glorp''*, etc). In §5–§8, we focus on the results over unseen classes; performance over seen classes is generally high across all evaluations.

[7] Each image is saved as a PNG and resized to the highest resolution supported by the given model; ImgNet uses 256×256 pixels. There is exactly 1 duplicate image in the *colors* dataset for seed = 10.

[8] This paper asks whether NNs are consistent with (key aspects of) Fodor's theory of concepts, *not* whether NNs are equivalent to humans. Synthetic data allows us to study models in a setting where we can guarantee that the desired structure is ''correct''. That is, we give Fodor the benefit of the doubt and assume his theory of concepts is correct. Ultimately, we care about the latter question: are NN's human-like? However, in our view, we don't have the data and theories (just yet) to tackle this in a deep, meaningful way. While Fodor's theory is certainly *not* a perfect theory of human concepts, at least some aspects of his theory are likely to be present in whatever the ''right'' theory is, even if not exactly as Fodor envisioned it (e.g., most credible theories appeal to compositionality and causality). Future work can and should relax our generous assumptions, work on non-synthetic data, and analyze NNs through the lens of competing theories of concepts.

4.2.3 Human Performance

We run a Mechanical Turk study with 150 individuals. Subjects are given three exemplars of each class (equivalent to Figure 2), and are then asked to assign a novel instance to one of the 18 classes. Across 1500 predictions, the majority label agrees with our ground truth label 63% of the time (over a 5.6% random baseline). We find that mistakes are systematic and predictable: For example, subjects routinely confusing *''gix''* and *''gip''* as the *''clean''* versus *''fuzzy''* edge is difficult to discern in this setting.

Thus, some of our class distinctions rely on perceptual features that are difficult for humans to distinguish, but which models are able to differentiate well. This is an important discussion point, but does not undermine the validity of the present study. In general, conceptual representation is considered to be divorceable from perception: The fact that one might mistake a cat for a skunk does not mean they do not have the concept of `cat`. By similar logic, the fact that our models have super-human perception in this domain need not prevent us from analyzing the structure of the concepts that they represent, or comparing them to a ground truth that imagines humans to have perfect perception.

5 Test 1: Predictions are Grounded

`Is_grounded` requires that if, definitionally, the difference between *''dax''* and *''blick''* is roundness, then this visual attribute should dictate predictions.

5.1 Experimental Design

We use counterfactual minimal pairs, which have been used in both NLP (Huang et al., 2020) and computer vision (Goyal et al., 2019b). Our dataset (§4.2) is generated using a set of background parameters (i.e., locations and sizes of the underlying shapes) in addition to the atomic concepts (shape, stroke, and layout). To generate minimal pairs, we sample 1000 sets of these background parameters, and then render each sampled set of parameters for every combination of shape×stroke×layout. This ensures the instances in a pair are equivalent in all visual features (total surface area covered by shapes, relative distance between shapes, etc.) except those features which change as a direct consequence of manipulating the target atomic concept. We generate minimal pairs in the colors dataset (§4.2) in the same way, treating color as another background parameter. After setting up the minimal pairs, we measure the probability that `predict(encode(·)) == gt_label(·)`.

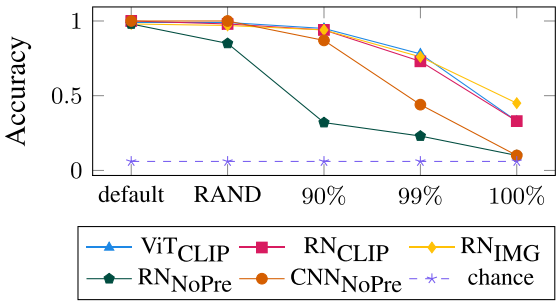

Figure 3: Results for `is_grounded` on the colors dataset. Performance for all models degrades when trained on data in which color is spuriously correlated with the target concepts, and then tested on out-of-distribution minimal pairs. However, pretrained models still perform well above chance.

If the model grounds concepts to the desired perceptual features, then it should perform perfectly at classifying the images across all settings. If the model performs poorly, we interpret this as evidence that the model grounds the concept to some features in a way that would not be ''shared'' with (idealized) humans, for example, the model considers *''dax''* to ground to color or size of shapes, rather than solely to *''circle''* &*''horizontal''*&*''smooth''*.

5.2 Results

The models perform well on the default dataset ($\sim$ 98%). When the classes are highly correlated with a spurious color feature, performance degrades (Figure 3). However, notably, even when models are trained on highly imbalanced data (e.g., with 99% of *''dax''*s being red), the pre-trained models still perform well above random out-of-distribution (75% over a 5.6% random baseline).

5.3 Discussion

We interpret this as a positive result: The results on the default dataset demonstrate that the pretrained models' behavior is explained by the expected perceptual features, satisfying `is_grounded`. The degradation in performance when using the colors dataset raises two issues worthy of discussion.

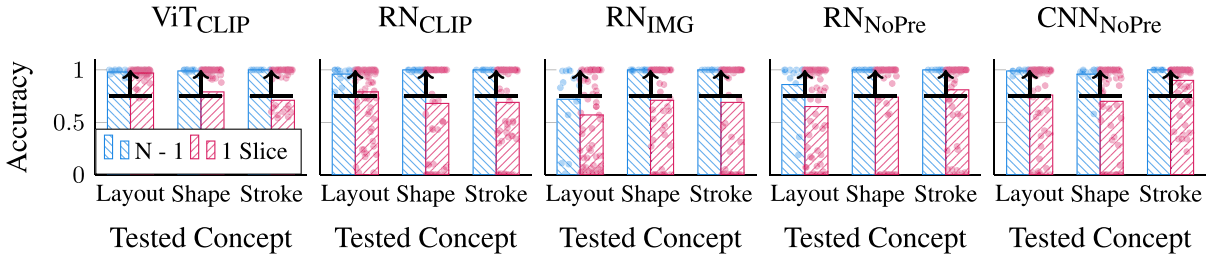

Figure 4: `is_token_of_type` on unseen classes. The points show the accuracies over seeds and the unseen test classes; the bar shows the mean over these points. Black arrows indicate expectations–we want to see models performing well, as high accuracy is indicative of a reusable type representation that generalizes to unseen concepts.

First, across our unit tests, this result is the one of only places in which we see a real difference between pretrained and from-scratch models. These results suggest that the pretrained models (which have been trained with access to linguisitic information, i.e., category labels for ImageNet and captions for CLIP) encode an inductive bias for shape over color. That is, even in the setting in which color is perfectly correlated with the class label, the models still generalize based on shape rather than color around half of the time. Such findings echo previously published arguments that pretraining can encode inductive biases that help models learn language more efficiently (Lovering et al., 2020; Warstadt et al., 2020; Mueller et al., 2022).

Second, while poor out-of-distribution generalization is not desirable, it is important to emphasize that it is *not* inconsistent with the use of symbolic concepts. For example, a model which explicitly represents symbols (e.g., Naive Bayes) could exhibit a similar drop in performance as the prior given the correlation in the training data makes the correct class less likely. As written, Fodor's criteria do not adjudicate on this issue. Thus, with respect to grounding, fully characterizing neural networks in terms of their symbolic representations (or lack thereof) requires refined criteria which can discriminate between models which represent grounded symbols (but make errors in learning) from models that do not represent grounded symbols at all.

6 Test 2: Representations Encode Types

`Is_token_of_type` requires that the system's representations of concepts can be mapped to discrete types in a reusable way.

6.1 Experimental Design

We train `has_concept` on a subset of the slices from the dataset (see §4.2.2). For example, we can train `has_concept` to predict the layout (*''vertical''*, *''horizontal''*, or *''ring''*) by training it on examples of *''dax''*, *''surp''* and *''slup''*, which differ only in the layout constituent, but are identical in the other constituents, (*''oval''*, *''smooth''*). We then evaluate on unseen classes, such as *''blick''*, *''gix''*, and *''wug''*, which exemplify the same variation in layout, but do so in the context of other constituents not seen in training (e.g., *''rectangle''*).

We take good generalization as evidence that the model's representations of a concept can be viewed as tokens of the same concept type. For example, whenever the model receives an input that falls under the concept *''vertical''*, the concept of *''vertical''* is tokened in the model's internal representations in a way which can be reliably localized by a single, fixed *''vertical''*-type detector. Generalization to unseen classes indicates that the tokening of *''vertical''* is not dependent on the other concepts that might be tokened simultaneously (e.g., *''oval''* or *''rectangle''*). Poor generalization suggests that models' internal representations are context dependent: *''vertical''* in the context of *''oval''* is not of the same type as *''vertical''* in the context of *''rectangle''*.

6.2 Results

The results are overall positive. All models show near-perfect accuracies on seen classes ($> 99\%$, not shown). Over the unseen classes (Figure 4), the models perform better in the easier N - 1 slices setting (when generalizing from 15 seen classes to 3 unseen classes). For 1 slice, the accuracies are lower but still well above chance—around 75%.

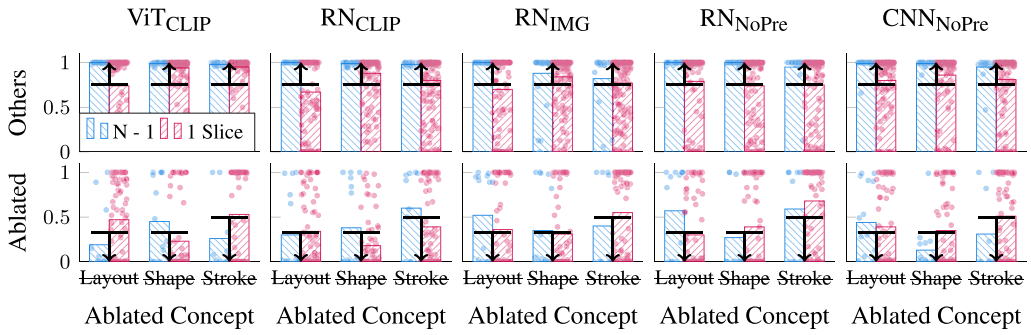

Figure 5: `is_modular` on unseen classes. Arrows indicate expectations: Performance for the ablated concepts (top row) should be at or below random and performance on other concepts (bottom row) should be high. Points show the accuracies over seeds and unseen test classes; the bar shows the mean.

6.3 Discussion

Overall, representations of atomic concepts appear to be ''the same'' across contexts, generalizing well to unseen compositions. The performance differential between 1 slice and N - 1 slice suggests (intuitively) that more varied data enables the `has_concept` probe to better identify the stable, defining features of the concept: That is, seeing *''vertical''* in the context of both *''oval''* and *''rectangle''* makes it easier to recognize *''vertical''* in the context of previously unseen *''polygon''*. As was the case with the out-of-distribution generalization results discussed in §5.3, these results about the amount and variety of training data required are interesting, but do not speak directly to the question of symbolic representations. Rather, our results on 1 slice vs. N - 1 slices correspond to a question about acquisition, and is an issue on which Fodor's criteria are silent. Other theories of concepts focus on acquisition (Spelke and Kinzler, 2007; Carey, 2009) and make empirical predictions about the amount and distribution of data from which certain concepts should be acquirable. Future work could expand our unit tests to reflect such empirical predictions, in addition to the in-principle criteria proposed by Fodor.

7 Test 3: Representations are Modular

`Is_modular` tests that removing one constituent concept from the representation of a composite concept does not harm the other constituents.[9]

[9]Whether concepts should be entangled, i.e., ''holism'' (Jackman, 2020), is an area of extensive debate. We make some strong assumptions following Fodor's ideals. See footnote 8.

7.1 Experimental Design

We use `ablate` to remove a given constituent and then assert that `has_concept` is unable to detect the removed concept, but still able to detect the remaining constituents. For example, *''dax''* $\models$ *''oval''&''horizontal''&''smooth''* is a composite concept. We require that ablating *''horizontal''* from a tokened representation of *''dax''* results in a representation of the form *''oval''&_&''smooth''*, which leaves the layout ''slot'' empty, but otherwise preserves the information about the structure and type of the composition. In our implementation, without loss of generality, we ablate sets of atomic concepts (e.g., ablating all three layout concepts together) rather than a single concept at a time.

High accuracy on the ablated concept means the system failed to implement `ablate` correctly. Low accuracy on the concepts that were not ablated (e.g., if removing layout means `has_concept` no longer can distinguish *''rectangle''* from *''oval''*) means that constituent representations are entangled in a way likely incompatible with, for example, productivity. Thus, for each atomic concept dimension (layout, shape, stroke) we run three tests—one to check that performance at detecting the ablated concepts is low and two to check that performance at detecting the other two dimensions is high. We consider ''high'' to be >75% accuracy;[10] random is 33% for layout and shape, and 50% for stroke.

7.2 Results

All the models are largely successful (Figure 5). Overall, performance is low on the removed

[10]This threshold is arbitrary, but allows us to talk in terms of explicit pass/fail criteria for our unit tests.

concept but high on the remaining concepts, as desired. Performance is higher variance in the harder 1 slice setting. For example, when layout is ablated in ViT_{CLIP}, the accuracy for detecting layout is far below random in the N - 1 slice setting, but marginally above random in the 1-slice setting.

7.3 Discussion

Across models and training configurations, the trends are in the expected direction: Performance on the ablated concept is low (near random) and performance on other concepts is high. In the harder 1-slice setting, performance on the not-ablated concepts sometimes degrades, meaning, for example, its not possible to remove the constituent *''vertical''* from *''dax''* without also damaging the representation of *''oval''* to some extent. In terms of Fodor's criteria for constituency, this suggests a problem, as the lack of modularity would make it difficult to explain phenomena such as infinite productivity—that is, if *''oval''* cannot be fully divorced from *''vertical''*, it becomes difficult to explain how the same *''oval''* is able to combine with arbitrarily many different layouts (*''horizontal''*, *''ring''*, etc.). However, the evidence is hardly damning—the patterns are largely consistent with expectations. As in §6.3, this represents a direction in need of future work and discussion. These results could become unambiguously positive if we concede that models might require sufficient training in order to learn modular concept representations. Fodor's theory does not offer criteria for what is ''sufficient'', but subsequent experiments could draw on other theories from developmental psychology to determine such criteria, and then refine the unit tests accordingly.

8 Test 4: Representations are *Not* Causal

`Is_causal` tests that the internal representations serve as ''mental causes and effects''. Where `is_token_of_type` and `is_modular` demonstrated that models' representations can be labeled and manipulated according to discrete types, we now test that those types are causally implicated in model behavior—for example, if the constituent concept *''oval''* is no longer tokened, will this prevent the model from producing the label *''dax''*? Similar to `is_grounded`, this test relies on counterfactual perturbations, but differs in that the perturbations are applied to the model's internal representations, rather than to the perceptual input.

8.1 Experimental Design

We evaluate `predict` after removing a concept with `ablate`. We expect this to impair the model's ability to reason about the ablated concept, but not others. For example, if we remove the layout dimension, the model should be able to distinguish between *''blick''* and *''dax''* (as they differ in shape), but be unable to distinguish between *''blick''* and *''slup''* (as they differ in layout). We thus distinguish two measures of accuracy: the rate at which the model's predicted concept matches the true concept along the removed dimension (which should be at random), and the rate at which the model's predicted concept matches the true concept along the other dimensions (which should be high). We take >75% accuracy to be high; random is 33% for layout and shape, and 50% for stroke.

8.2 Results

All of our models fail this test (Figure 6). Accuracies with respect to the ablated features stay far above random. The pattern holds whether we train on 1 or N - 1 slices, and whether we evaluate on seen (not shown) or unseen classes. Increasing the iterations of INLP (§4) (not shown) causes performance to deteriorate for all concepts (even those which we are not trying to ablate), a different pattern which nonetheless constitutes a failure on our unit test.

8.3 Discussion

These models in general pass `is_modular`, meaning that there exists a localizable representation of each atomic concept. Thus, this subsequent failure suggests that `predict` ends up using different representations than those which are used by `has_concept`. That is, while there exists a part of the internal representation that encodes the atomic concepts, `predict` relies on a different part of the internal representation to make decisions about composite concepts.

One possible explanation for this result is that the model tokens *both* the atomic concepts and the composite ones simultaneously, with each concept (composite or not) represented as its own symbol, and `predict` uses only the composite ones directly. For example, observing an instance

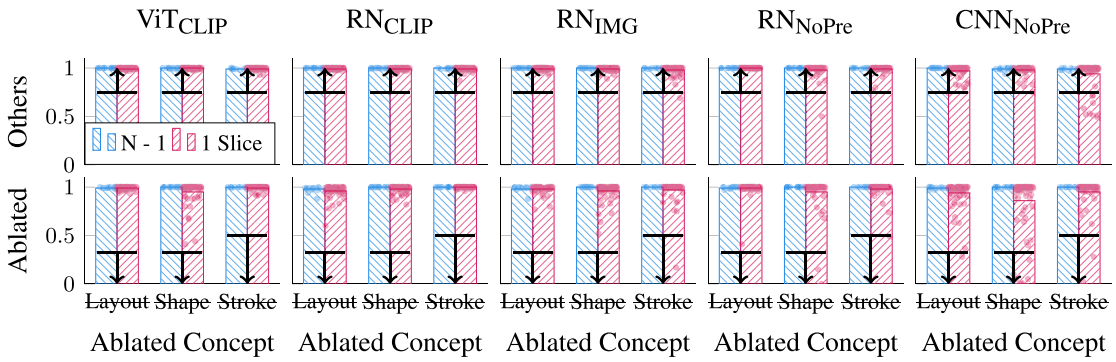

Figure 6: `is_causal` on unseen classes. Arrows indicate expectations: performance for the ablated concepts (top) should be at chance and performance on other concepts (bottom) should be high. Bars show mean accuracies over seeds and unseen test classes. Accuracies are over classes (composite concepts).

of *''dax''* causes the model to token the atomic *''oval''* and *''horizontal''* but also a composite concept *''oval''*&*''horizontal''*, which is a symbol in and of itself. Whether or not such behavior is consistent with Fodor's criteria depends on the causal relationship between these tokenings—that is, does tokening *''oval''*&*''horizontal''* entail tokening *''oval''*? Future work could answer this question by looking more closely at the way representations evolve during training or across layers during processing. We present initial investigations on the latter in §9.

Finally, as discussed in §3, our specification applies not just to the representations, but to the system as a whole. Thus, the implementation of `ablate` (INLP in our case), is part of the evaluated system. When a model fails this test, we cannot say whether there was a critical flaw with the representation or rather that the concept ablation itself failed (e.g., because of assumptions of linearity, of treating `encode` as a block, etc.) It is possible that, if new techniques are used to instantiate `ablate`, the same representations might fare better (or worse) according to our tests. For example, since writing, new techniques for applying non-linear perturbations (Tucker et al., 2021; Meng et al., 2022) have been proposed. Such methods could potentially be incorporated into our framework to yield new insights on this particular test.

9 Analysis: Concepts Across Layers

9.1 Hypothesis

Here, we conduct a preliminary investigation into one hypothesis about the reason for our models' failure on `is_causal`. Specifically, we hypothesize that the causal structure exists, but it unfolds across layers. The constituent concepts (e.g., *''oval''* and *''horizontal''*) are tokened in early layers, and are subsequently composed such that the composite concept (*''dax''* = *''oval''* &*''horizontal''*) is tokened at the final layer as its own symbol and is the direct effect of the model's predicted label. Below, we investigate two predictions of this hypothesis, and observe mixed results.

9.2 Aggregate Analysis

If our hypothesis is true, we would expect to see 1) that concepts should emerge in the expected order across layers, that is, constituent concepts before composite concepts, and 2) errors in labeling the composite concept at a given layer should be explained by errors in identifying the constituents at that layer. That is, if the model cannot recognize *''oval''* vs. *''rectangle''* until layer 4, it should not be able to differentiate *''dax''* from *''blick''* (which depend on the shape distinction) before that layer. Moreover, if the model's failure to recognize *''oval''* vs. *''rectangle''* is the reason for the mislabel, the observed error in labeling the composite concepts should be equal to the product of the errors the constituents. That is, considering *''dax''*, if errors in the constituents cause errors in the composite, the model should mislabel *''dax''* as *''blick''* exactly as often as `has_concept` mistakenly returns *''rectangle''* instead of *''oval''*.

Figure 7 shows predictions from probing models for each concept at each layer. It also shows the *composed probe* accuracy, computed by combining the predictions of each of the probing

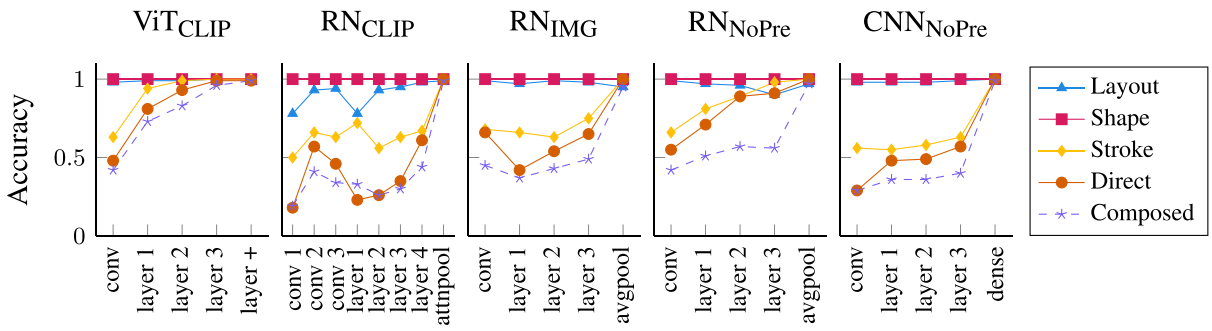

Figure 7: Probing performance explains downstream performance across layers. Composed Probes: accuracy that would result by directly composing the predictions of the probes for each constituent concept; Direct Classification: accuracy of a classifier trained at the given layer to predict the *composite* concept. The remaining lines show the probing performance for the constituent concepts.

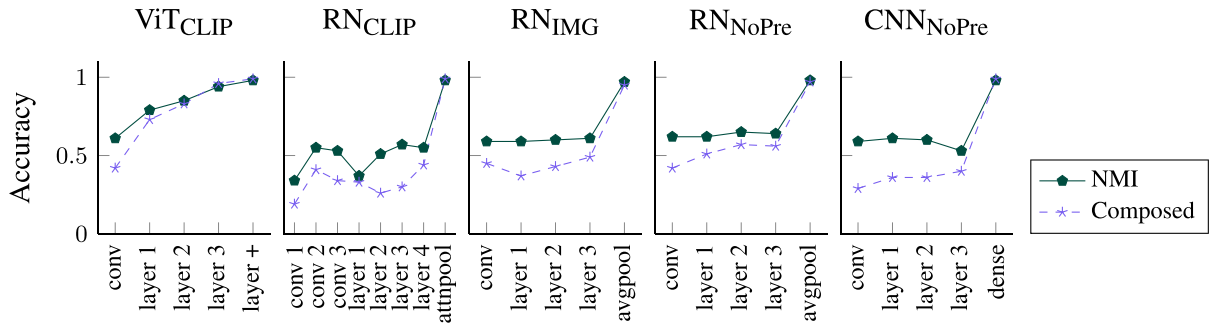

Figure 8: Expected vs observed mistakes are different. Mutual information between the probing and downstream predictions at the instance level. If there were a direct causal connection between the constituent concept and the composite prediction, we would expect high NMI across all layers. Instead, for most models, NMI is only high at the final layer.

classifiers, as well as the *direct classification* accuracy, computed by measuring the performance of a new classifier trained to predict the final class *at each layer*.[11] The trend is promising: Composite concepts are recognized only after constituents are recognized, and, for most models, the direct classification accuracy is close to what we expect based on composed probes (though often slightly higher, especially on the from-scratch models).[12]

9.3 Instance-Level Analysis

If our hypothesis holds, not only should the error rates be similar, but the direct class prediction should be predicted by the composed probes. That is, if at a given layer, the model is given an image of a *''dax''* and mistakenly detects *''rectangle''*(according to the probe) instead of *''oval''*, then the model should label the input as *''blick''*.

To quantify whether the instance-level predictions behave this way, we compute the normalized pointwise mutual information (NMI, which ranges from 0 to 1) between the direct prediction and the composition of the probe predictions. If the direct prediction is indeed a function of the constituent probes, we would expect to see high NMI (near 1.0) across the board—that is, even when the model's accuracy is low, the NMI would be high if it was erring in the expected way. However, Figure 8 shows there is relatively little mutual information until the final layer of the network (ViT$_{CLIP}$ might be an exception). In other words, while the probing and downstream models have similar error rate in aggregate, they make *different mistakes* on individual instances.

This result is inconclusive: While high NMI would have been suggestive of a causal connection between the probes and the classifier, low NMI

[11]Because CNNs and ViTs have multiple dimensions, to get a vector representation for a given layer, we mean across the channels and then flatten into a vector. There are many other possible approaches we did not evaluate.

[12]Note: In early layers, models make the same mistakes people do, e.g., confusing fuzzy ovals and polygons (§4.2).

doesn't necessarily mean such a link does not exist. For example, if a model is altogether failing to differentiate *''rectangle''*s and *''oval''*s, and thus failing to differentiate *''dax''*s and *''blicks''*s, then both the probe and the classifier might resort to pure guessing between these labels, and thus appear to disagree even though they in fact depend on the same (underdetermined) conceptual representation.

10 Summary

Overall, our experiments suggest that models exhibit grounded behavior and possess conceptual representations that encode modular, context-independent types. However, we don't find evidence of a direct causal connection between the representations of constituent concepts and those of composite concepts, an essential feature of Fodor's theory on which our specification is based. Our discussions of each individual experiment (§5.3, §6.3, §7.3, §8.3) together raise several general themes.

First, success on our tests often depends on granting assumptions about how concepts are acquired: How should concepts be learned in the face of spurious correlations, how many training examples are necessary, and so forth? While Fodor does not focus on acquisition in his criteria, other theories exist which make empirical predictions about how and when specific conceptual representations develop in humans (Spelke and Kinzler, 2007; Carey, 2009). Future work could translate such predictions into additional unit tests (measuring learning curves, processing times, etc.), in order to diagnose whether current models' errors should be interpreted as failures vs. expected signatures of conceptual learning.

Second, our proposed tests evaluate a system as a whole. Thus, our ability to make claims about neural networks as an implementation of conceptual reasoning is dependent on the quality of the tools available for inspecting neural networks' internals. A particularly fruitful area for future work is finding alternative implementations of `ablate`. Recent work by Tucker et al. (2021) and Meng et al. (2022) could be promising places to start.

Finally, we observe interesting trends about the effect of pretraining on conceptual representations. The models we evaluate share the same architecture but have different pretraining regimes. Only for `is_grounded`, and possibly in our layerwise analysis, was there a clear benefit from pretraining. Our results suggested that the pretrained models had an inductive bias for shape over color, and may show more promise in subsequent studies of causality. On other tests, pretraining did not translate to a clear improvement in conceptual structure.

11 Related Work

Our study follows work on distributional models of semantics, which seeks to interpret computational models based on vectors and neural networks in terms of linguistic and cognitive theories (Erk, 2012; Lenci, 2018; Boleda, 2020). However, we do not take a stand on how vector spaces compare to symbols as models of human language/cognition *at the computational level*. Rather, our study assumes that one prefers a symbolic model at the computational level, and asks whether neural networks could serve as the implementation of such a model.

Closely related is recent work which seeks to answer whether neural networks exhibit properties such as systematicity and compositionality both in NLP (Lake and Baroni, 2018; Yanaka et al., 2019; Goodwin et al., 2020; Kim and Linzen, 2020) and in computer vision (Johnson et al., 2017; Andreas et al., 2016). In contrast to these studies, which assess the final model behavior (analogous to `predict`), we have additional criteria for how the representations behave (like `is_modular`). Also related is prior work which attempts to define mappings between humans' and neural networks' conceptual spaces, for example, by defining measures of compositionality or groundedness based on how well similarity in vector space reflects similarity according to a symbolic representation (Andreas, 2019; Chrupała and Alishahi, 2019; Merrill et al., 2021). Our work differs in that we use a multifaceted suite of evaluation techniques in order to operationalize a specific theory of concepts.

We use techniques from the broad area of interpretability and analysis of neutral networks. First, work on **identifying concepts in neural networks** seeks interpretable patterns in the activations and gradients of neural networks, for example, that unsupervised CNNs encode concepts such as edges (Sermanet et al., 2013; Le, 2013). Many techniques have been proposed in

order to determine which input features are ‘‘important’’ to model decisions (Ribeiro et al., 2016; Sundararajan et al., 2017; Kim et al., 2018; Wiegreffe and Pinter, 2019). We employ the method of ‘‘diagnostic classifiers’’ (Veldhoen et al., 2016; Ettinger et al., 2016; Adi et al., 2017; Hupkes et al., 2018), with the goal of finding high-level concepts which are not directly reducible to input features (Kim et al., 2018; Tenney et al., 2019). Second, work on **counterfactual perturbations** attempts to provide causal explanations of model predictions in terms of input features or concepts. Most such work relies on controlled perturbations of the model's input—for example, manipulating pixels in an image (Fong and Vedaldi, 2017; Chang et al., 2018; Goyal et al., 2019a,b) or tokens in a string of text (Ribeiro et al., 2018; Webster et al., 2020; Huang et al., 2020), though recent methods operate on models' internal representations (Vig et al., 2020; Ravfogel et al., 2020; Tucker et al., 2021; Meng et al., 2022). We employ both types of counterfactual manipulations (we manipulate inputs in §5 and representations in §8). Unlike prior work, which often treats these counterfactual manipulations as different measures of the same thing, we connect each evaluation to a different aspect of Fodor's theory of concepts. Finally, our work uses the idea of **unit testing for neural networks** (Adebayo et al., 2020; Ribeiro et al., 2020).

12 Conclusion

We introduce a specification for symbolic conceptual reasoning based on Fodor's theory of concepts. We find evidence that current neural network models are consistent with many predictions of this theory but don't demonstrate a causal connection between the representations of constituent concepts and those of composite concepts. Further investigation into methods for manipulating models' internal representations may illuminate whether this inconsistency is fundamental to neural networks, or rather a limitation of current analysis tools.

Acknowledgments

We would like to thank Roman Feiman, Carsten Eickhoff, Gabor Brody, and Jack Merullo for the helpful discussions, as well as the Brown LUNAR lab. We want to thank Sheridan Feucht for organizing and running the Mechanical Turk study which helped us contextualize our findings. Furthermore, we want to thank our reviewers for their thorough feedback, helping us better present our work. This research was conducted using computational resources and services at the Center for Computation and Visualization, Brown University. This research was supported by the DARPA GAILA program.

References

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. 2020. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *International Conference on Learning Representations (ICLR)*.

Jacob Andreas. 2019. Measuring compositionality in representation learning. In *International Conference on Learning Representations*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48.

Ned Block. 1981. Psychologism and behaviorism. *The Philosophical Review*, 90(1):5–43. https://doi.org/10.2307/2184371

Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6:213–234. https://doi.org/10.1146/annurev-linguistics-011619-030303

Susan Carey. 2009. *The Origin of Concepts*. Oxford series in cognitive development. Oxford University Press.

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. 2018. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*.

Grzegorz Chrupała and Afra Alishahi. 2019. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, pages 2952–2962, Florence, Italy. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. When BERT forgets how to POS: Amnesic probing of linguistic properties and MLM predictions. *arXiv preprint arXiv:2006.00995*.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653. https://doi.org/10.1002/lnco.362

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

Jerry A. Fodor. 1998. *Concepts: Where Cognitive Science Went Wrong*. Oxford University Press. https://doi.org/10.1093/0198236360.001.0001

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1–2):3–71. https://doi.org/10.1016/0010-0277(88)90031-5

Ruth C. Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437. https://doi.org/10.1109/ICCV.2017.371

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of ACL*.

Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. 2019a. Explaining classifiers with causal concept effect. *arXiv preprint arXiv:1907.07165*.

Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019b. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2020. Reducing sentiment bias in language models via counterfactual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 65–83, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.7

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926. https://doi.org/10.1613/jair.1.11196

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR.

Henry Jackman. 2020. Meaning Holism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Winter 2020 edition. Metaphysics Research Lab, Stanford University.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2017.215

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with

concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2668–2677. PMLR.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of EMNLP*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of ICML*.

Quoc V. Le. 2013. Building high-level features using large scale unsupervised learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598. IEEE.

Alessandro Lenci. 2018. Distributional models of word meaning. *Annual Review of Linguistics*, 4:151–171. https://doi.org/10.1146/annurev-linguistics-030514-125254

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7:195–212. https://doi.org/10.1146/annurev-linguistics-032020-051035

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2020. Predicting inductive biases of pre-trained models. In *International Conference on Learning Representations*.

Eric Margolis and Stephen Laurence, editors. 1999. *Concepts: Core Readings*. MIT Press.

David Marr. 2010. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual knowledge in GPT. *arXiv preprint arXiv:2202.05262*.

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *arXiv preprint arXiv:2104.10809*. https://doi.org/10.1162/tacl_a_00412

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. 2022. Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models. *arXiv preprint arXiv:2203.09397*. https://doi.org/10.18653/v1/2022.findings-acl.106

Ellie Pavlick. 2022. Semantic structure in deep learning. *Annual Review of Linguistics*, 8:447–471. https://doi.org/10.1146/annurev-linguistics-031120-122924

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR. https://doi.org/10.48550/arXiv.2103.00020

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256. https://doi.org/10.18653/v1/2020.acl-main.647

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. ''Why should I trust you?'' Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. https://doi.org/10.1145/2939672.2939778

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1079

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics, pages 4902–4912, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.442

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*. https://doi.org/10.18653/v1/2021.emnlp-main.230

Steven Sloman. 2005. *Causal Models: How People Think About the World and its Alternatives*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195183115.001.0001

Elizabeth S. Spelke and Katherine D. Kinzler. 2007. Core knowledge. *Developmental Science*, 10(1):89–96. https://doi.org/10.1111/j.1467-7687.2007.00569.x, PubMed: 17181705

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Mycal Tucker, Peng Qian, and Roger Levy. 2021. What if this modified that? Syntactic interventions with counterfactual embeddings. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 862–875. https://doi.org/10.18653/v1/2021.findings-acl.76

Sara Veldhoen, Dieuwke Hupkes, and Willem Zuidema. 2016. Diagnostic classifiers: Revealing how neural networks process hierarchical structure. In *CEUR Workshop Proceedings*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in Neural Information Processing Systems*, 33:6–12.

Alex Warstadt, Yian Zhang, Haau-Sing Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). *arXiv preprint arXiv:2010.05358*. https://doi.org/10.18653/v1/2020.emnlp-main.16

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP*. https://doi.org/10.18653/v1/W19-4804