Analysis methods in NLP

Christopher Potts

Stanford Linguistics

CS224u: Natural language understanding

Overview

Varieties of evaluation

Behavioral

- Standard ("IID"; Independent and Identically Distributed)
- Exploratory
- Hypothesis-driven
- Challenge
- Adversarial
- Security-oriented

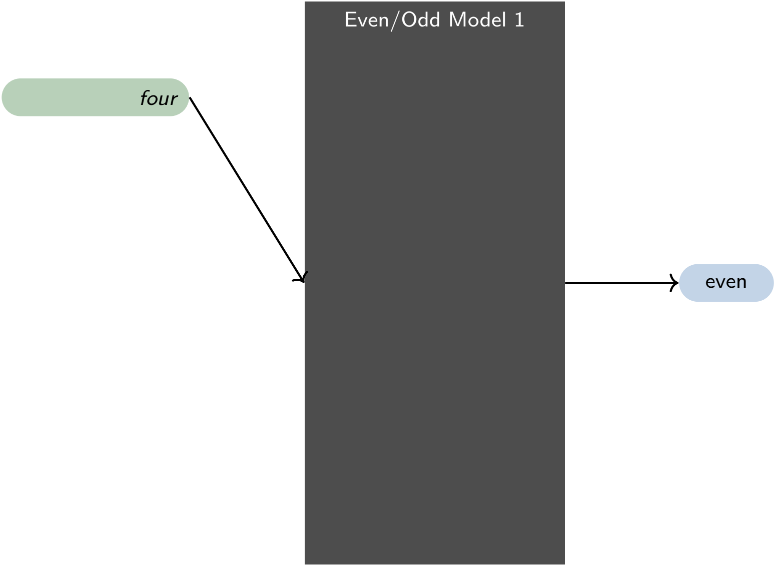
Structural

- Probing
- Feature attribution
- Interventions

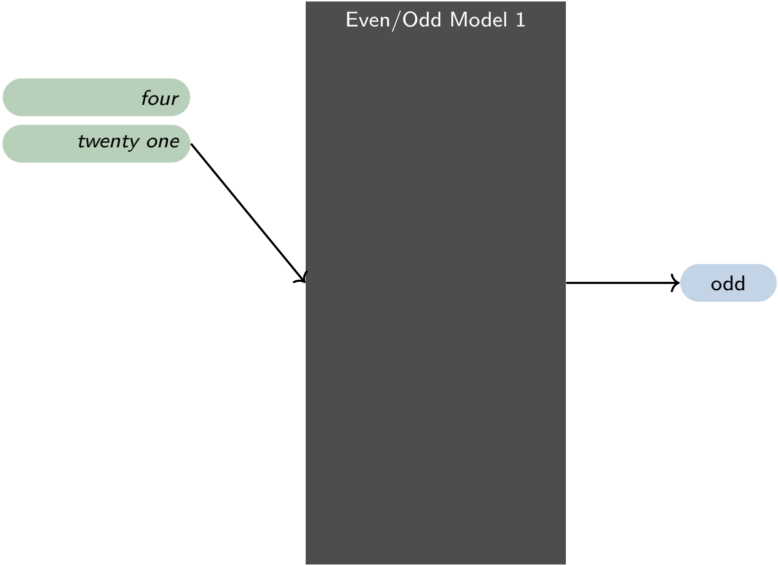
Limits of behavioral testing

Even/Odd Model 1

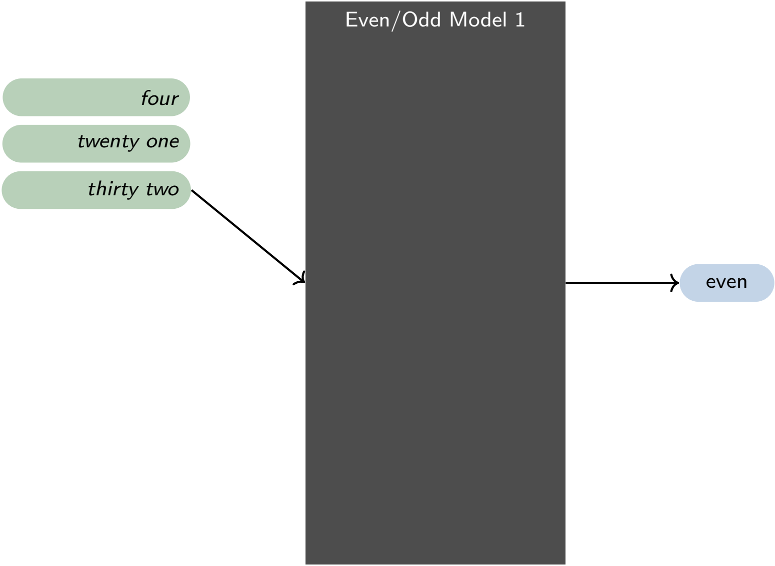
Limits of behavioral testing



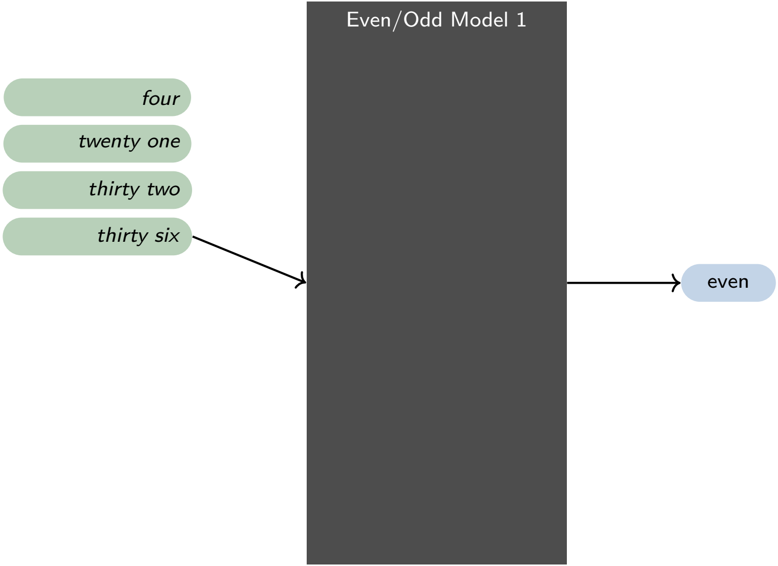
Limits of behavioral testing



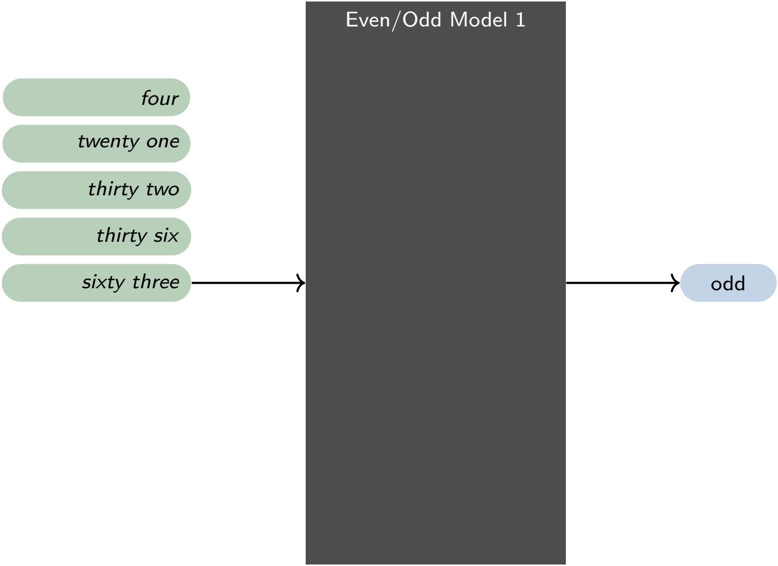
Limits of behavioral testing



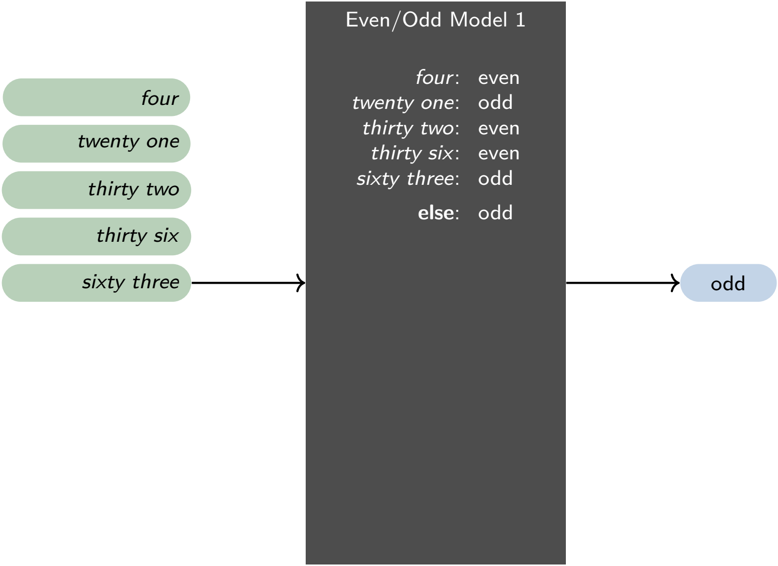
Limits of behavioral testing



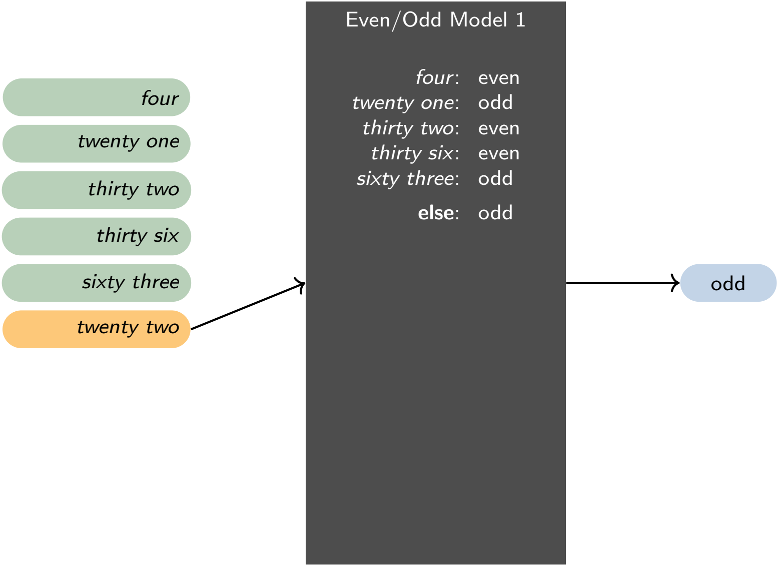
Limits of behavioral testing



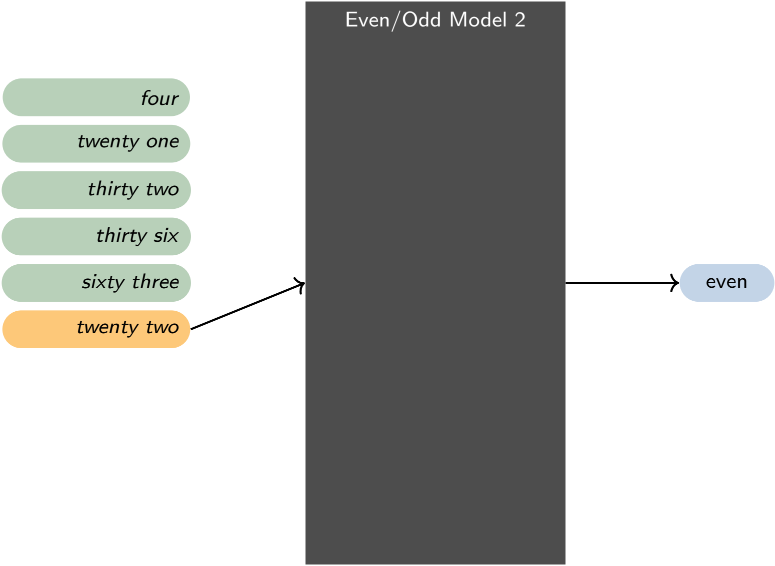
Limits of behavioral testing



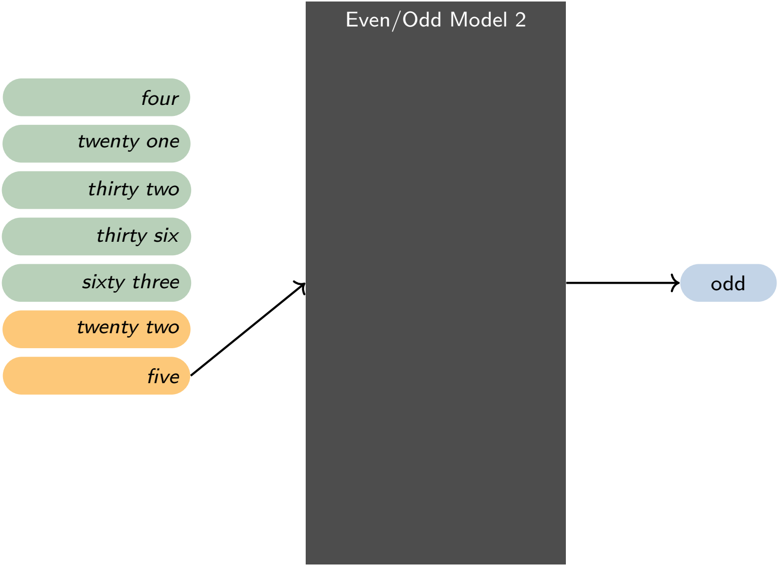
Limits of behavioral testing



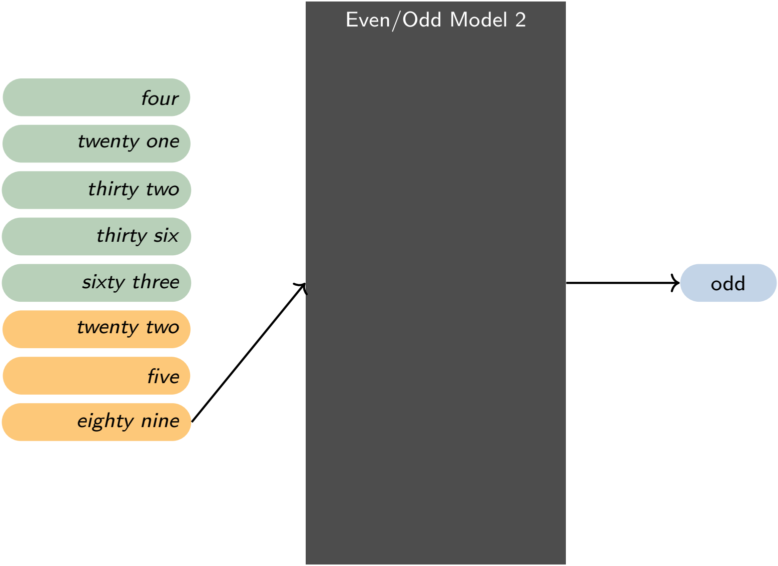
Limits of behavioral testing



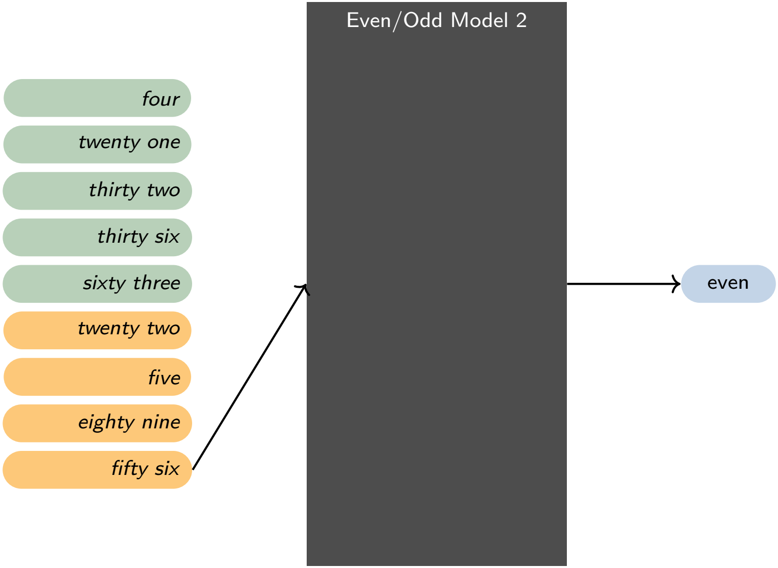
Limits of behavioral testing



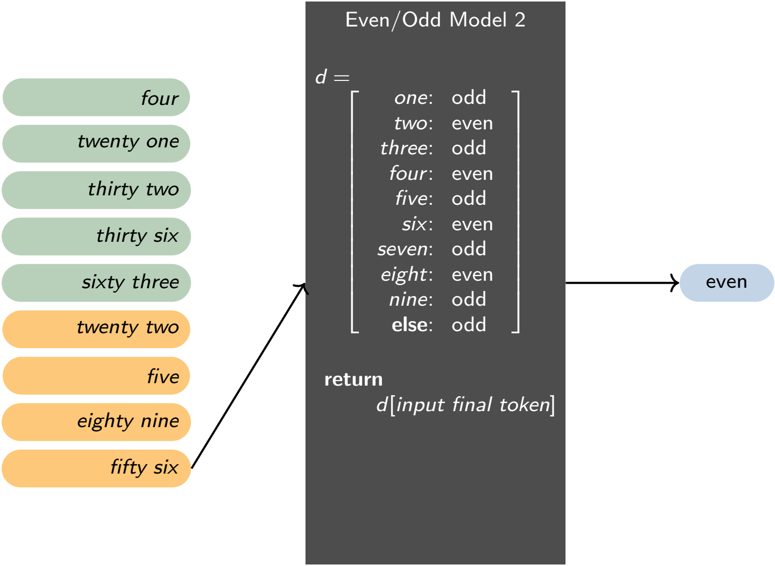
Limits of behavioral testing



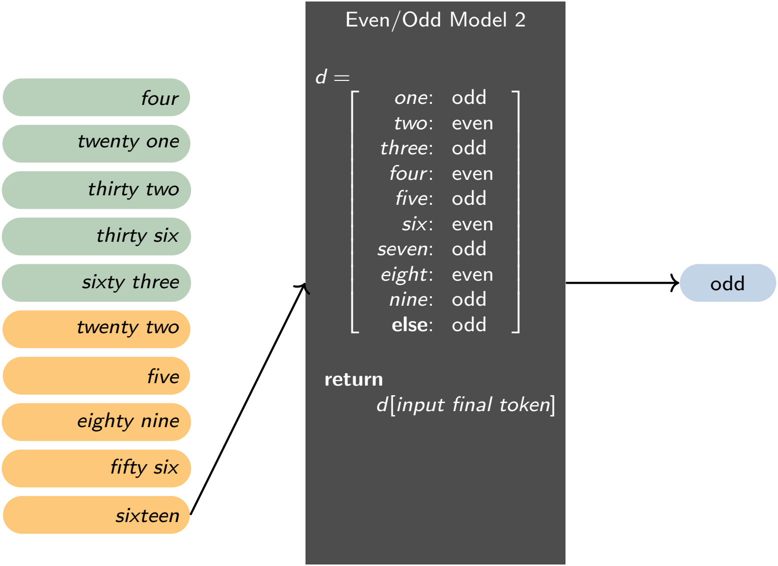
Limits of behavioral testing



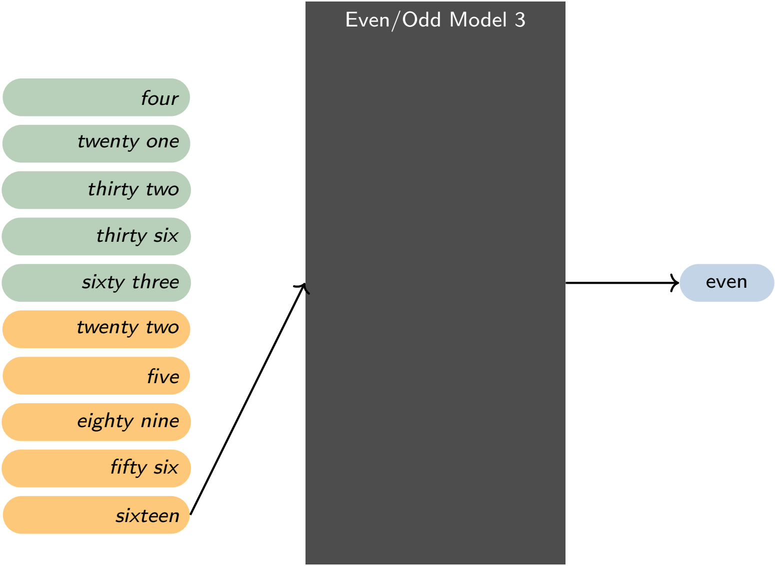
Limits of behavioral testing



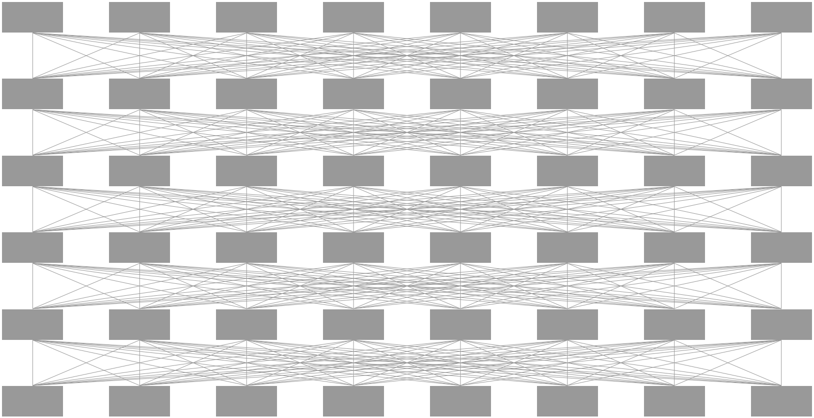
Limits of behavioral testing



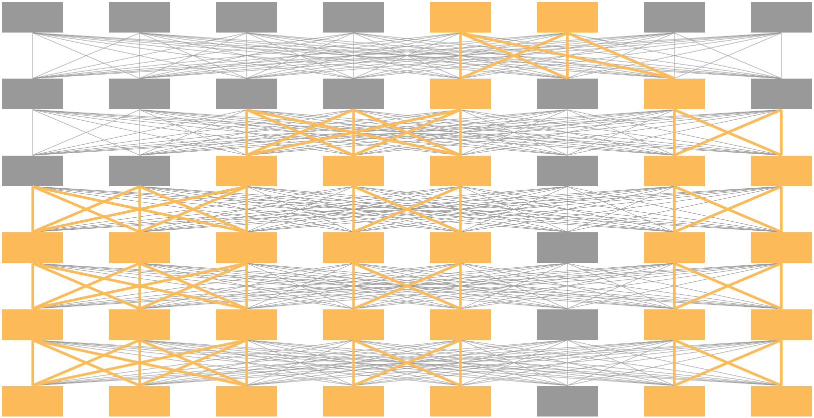
Limits of behavioral testing



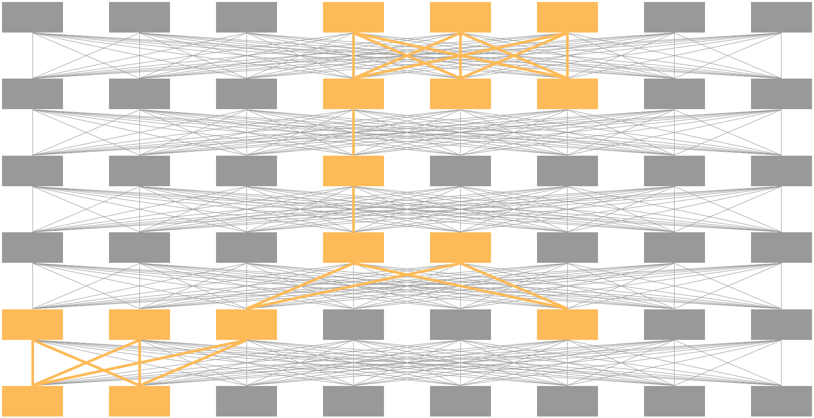
Models today



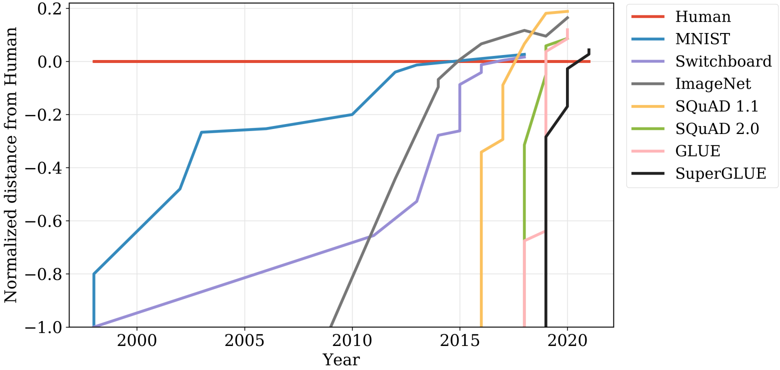
The interpretability dream



The reality: Apparently just a mess (but only apparently!)



Progress on benchmarks



Kiela et al. 2021

Systematicity

Fodor and Pylyshyn (1988:37):

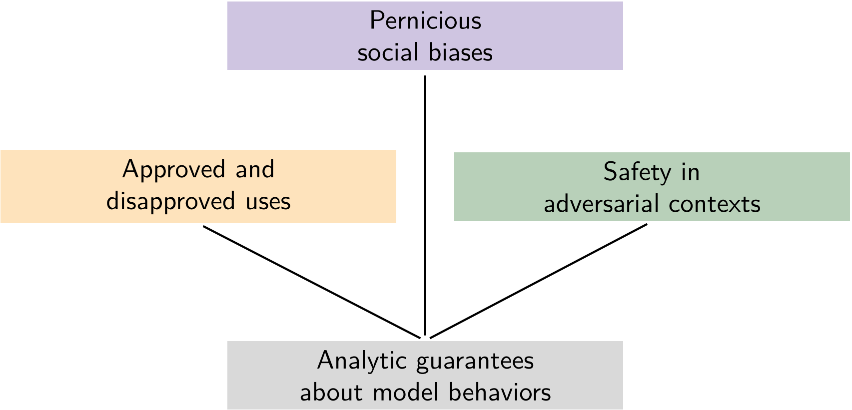
"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

1. Sandy loves the puppy.
2. The puppy loves Sandy.
3. the turtle $\sim$ the puppy
4. The turtle loves Sandy.
5. ...

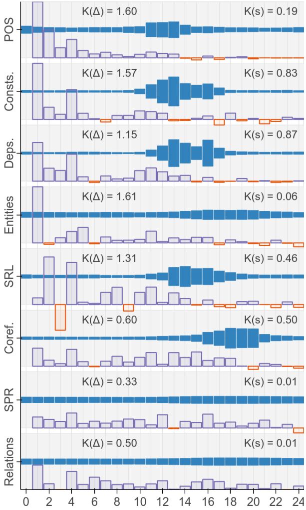
Compositionality

The meaning of a phrase is a function of the meanings of its immediate syntactic constituents and the way they are combined.

A crucial prerequisite



Probing internal representations



Tenney et al. 2019

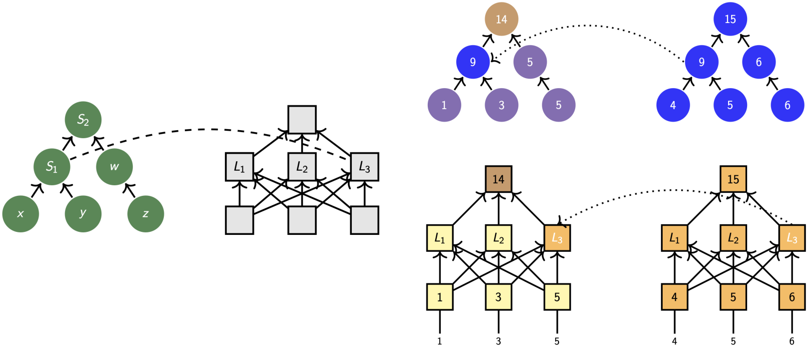
Feature attribution

Legend: Away from true label · Neutral wrt true label · Toward true label

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
2	2 (0.82)	None	-5.71	#s They said it would be great , and they were right . #/s
0	0 (0.50)	None	2.25	#s They said it would be great , and they were wrong . #/s
2	2 (0.76)	None	-0.35	#s They were right to say it would be great . #/s
0	0 (0.62)	None	2.84	#s They were wrong to say it would be great . #/s
2	2 (0.77)	None	2.59	#s They said it would be stellar , and they were correct . #/s

Integrated gradients; Sundararajan et al. 2017

Intervention-based methods



Analytical framework

	Characterize representations	Causal inference	Improved models
Probing	😃		🤔
Feature attribution	🤔	😃	
Interventions	😃	😃	😃

Overview

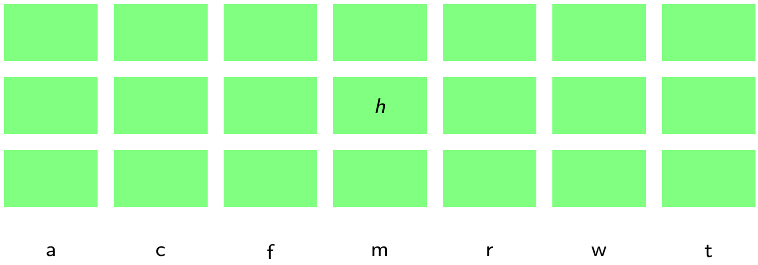
1. Core idea: use supervised models (the probes) to determine what is latently encoded in the hidden representations of our target models.
2. Often applied in the context of BERTology – see especially Tenney et al. 2019.
3. A source of valuable insights, but we need to proceed with caution:
   - A very powerful probe might lead you to see things that aren't in the target model (but rather in your probe).
   - Probes cannot tell us about whether the information that we identify has any *causal* relationship with the target model's behavior.
4. Final section: unsupervised probes.

Recipe for probing

1. State a hypothesis about an aspect of the target model's internal structure.
2. Choose a supervised task that is a proxy for the internal structure of interest.
3. Identify the place in the model where you believe the structure will be encoded.
4. Train supervised probe on the chosen site(s).

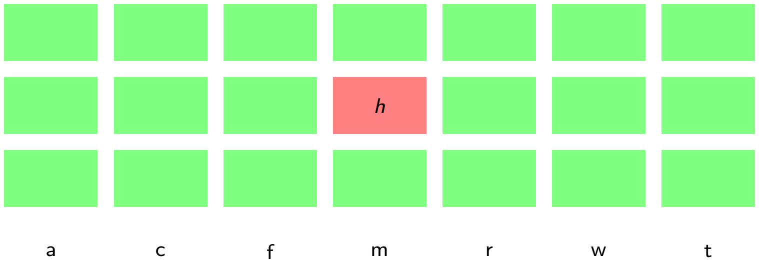
Conneau et al. 2018; Tenney et al. 2019

Core method



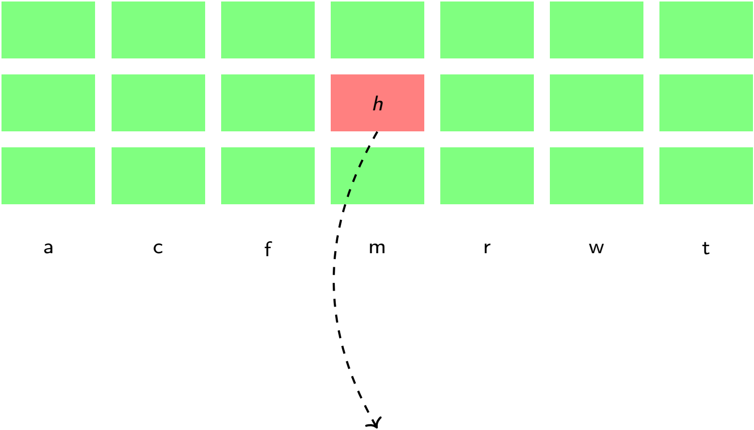
Conneau et al. 2018; Tenney et al. 2019

Core method



Conneau et al. 2018; Tenney et al. 2019

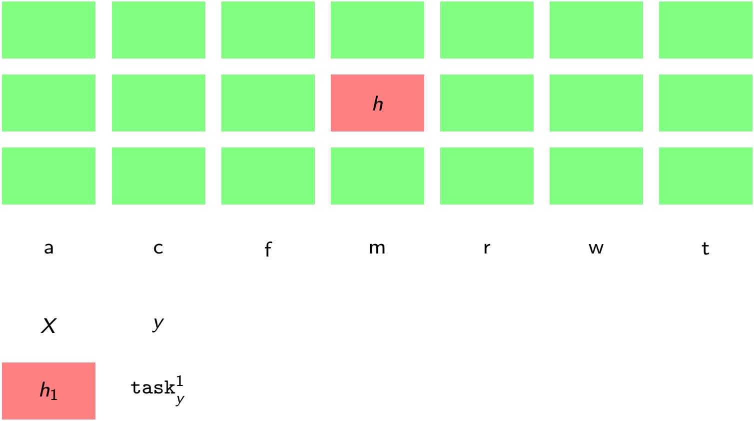
Core method



SmallLinearModel(h) = *task*

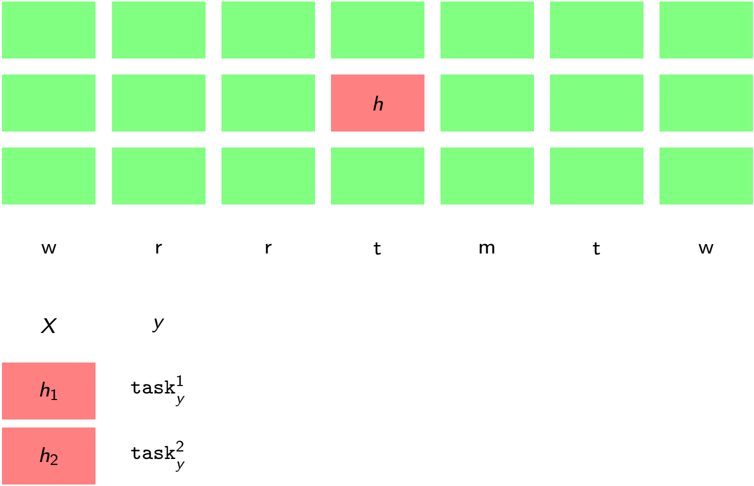
Conneau et al. 2018; Tenney et al. 2019

Core method



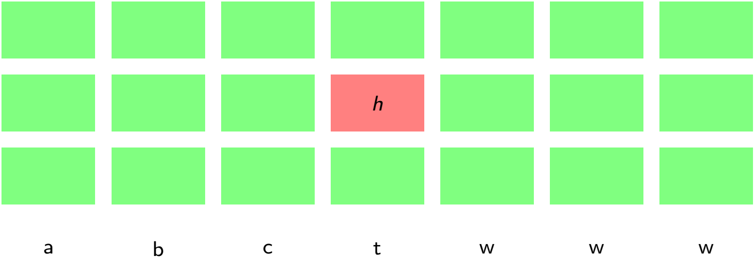
Conneau et al. 2018; Tenney et al. 2019

Core method



Conneau et al. 2018; Tenney et al. 2019

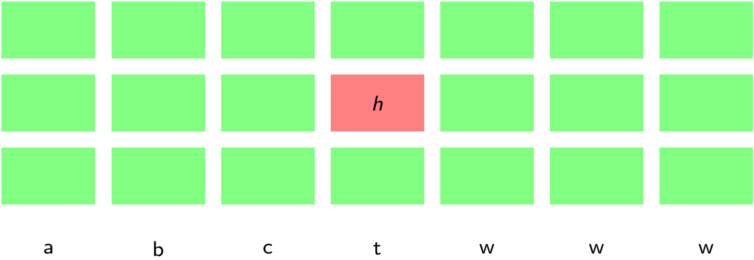
Core method



X	y
h_1	$\texttt{task}_y^1$
h_2	$\texttt{task}_y^2$
h_3	$\texttt{task}_y^3$

Conneau et al. 2018; Tenney et al. 2019

Core method



X	y
h_1	$\texttt{task}_y^1$
h_2	$\texttt{task}_y^2$
h_3	$\texttt{task}_y^3$

SmallLinearModel(X, y)

Conneau et al. 2018; Tenney et al. 2019

Core method

Conneau et al. 2018; Tenney et al. 2019

Probing or learning a new model?

1. Probes in the above sense are supervised models whose inputs are frozen parameters of the model we are probing.
2. This is hard to distinguish from simply fitting a supervised model as usual, with a particular choice for featurization.
3. At least some of the information that we identify is likely to be stored in the probe model.
4. More powerful probes might "find" more information – by storing more information in the probe parameters.

Control tasks and probe selectivity

Control task

A random task with the same input/output structure as the target task.

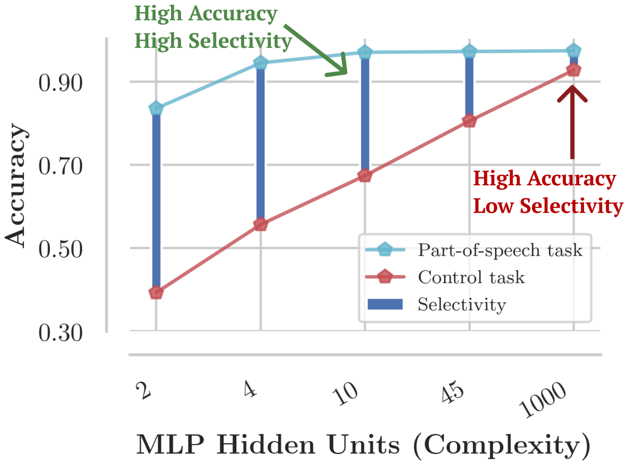
- Word-sense classification: words assigned random fixed senses.
- POS tagging task: words assigned random fixed tags.
- Parsing: assigned edges randomly using simple strategies.

Selectivity

The difference between probe performance on the task and probe performance on the control task.

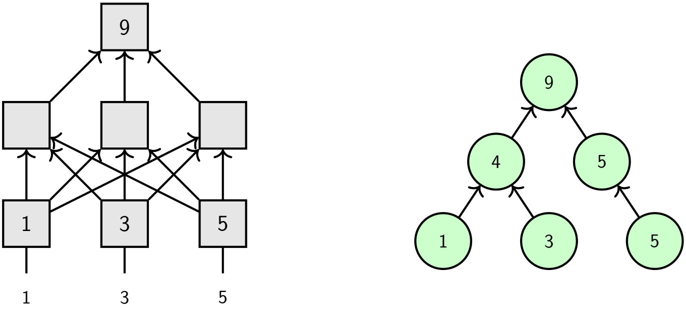
Hewitt and Liang 2019

Control tasks and probe selectivity

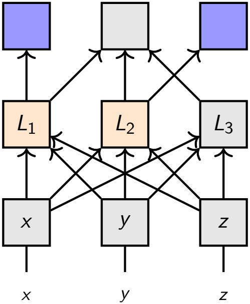


Hewitt and Liang 2019

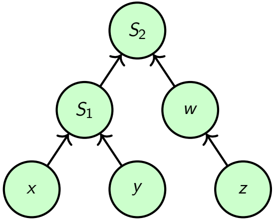
Simple example



No causal inferences



1. Probe L_1: it computes z
2. Probe L_2: it computes $x + y$
3. Aha!

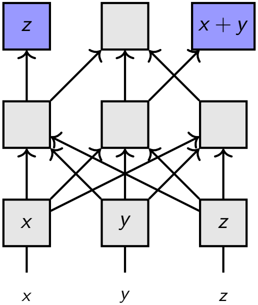


4. But L_2 has no impact on the output!

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

$$\mathbf{w} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad (\mathbf{x}W_1; \mathbf{x}W_2; \mathbf{x}W_3)\,\mathbf{w}$$

From probing to multi-task training



$$\mathbf{w} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

Unsupervised probes

1. Saphra and Lopez (2019): Singular Vector Canonical Correlation Analysis as a probing technique
2. Clark et al. (2019) and Manning et al. (2020): Inspecting attention weights.
3. Hewitt and Manning (2019) and Chi et al. (2020): Linear transformations of hidden states to identify latent syntactic structures in BERT.
4. Rogers et al. (2020): extensive discussion of probing and related efforts and what they have revealed about BERT representations.

Summary

	Characterize representations	Causal inference	Improved models
Probing	😃		🤔
Feature attribution	🤔	😃	
Interventions	😃	😃	😃

Feature attribution

captum.ai

1. Integrated gradients (Sundararajan et al. 2017)
2. Gradients
3. Saliency Maps (Simonyan et al. 2013)
4. DeepLift (Shrikumar et al. 2017)
5. Deconvolution (Zeiler and Fergus 2014)
6. LIME (Ribeiro et al. 2016)
7. Feature ablation
8. Feature permutation
9. ...

https://captum.ai
https://github.com/cgpotts/cs224u/blob/main/feature_attribution.ipynb

Axioms

Sensitivity

If two inputs x and x' differ only at dimension i and lead to different predictions, then feature f_i has non-zero attribution.

$$M([1, 0, 1]) = \text{positive}$$
$$M([1, 1, 1]) = \text{negative}$$

Implementation invariance

If two models M and M' have identical input/output behavior, then the attributions for M and M' are identical.

Sundararajan et al. 2017

Gradients · inputs

$$\text{InputXGradient}_i(M, x) = \frac{\partial M(x)}{\partial x_i} \cdot x_i$$

```python
"""For both functions, the `forward` method of `model` is used.
 `X` is an (m x n) tensor of attributions. Use `targets=None` for
 models with scalar outputs, else supply a LongTensor giving a
 label for each example."""

 import torch
 def grad_x_input(model, X, targets=None):
     X.requires_grad = True
     y = model(X)
     y = y if targets is None else y[list(range(len(y))), targets]
     (grads, ) = torch.autograd.grad(y.unbind(), X)
     return grads * X

 from captum.attr import InputXGradient
 def captum_grad_x_input(model, X, target):
     X.requires_grad = True
     amod = InputXGradient(model)
     return amod.attribute(X, target=target)
```

Attributions wrt predicted vs. actual labels

```
import torch
import utils
from sklearn.datasets import make_classification
from torch_shallow_neural_classifier import
    TorchShallowNeuralClassifier

utils.fix_random_seeds()

X, y = make_classification(n_samples=100, n_classes=2, n_features=4,
    n_informative=4, n_redundant=0, random_state=1)

# Deliberately undertrained model:
mod_bad = TorchShallowNeuralClassifier(max_iter=1)
mod_bad.fit(X, y)

# Attributions wrt the true labels:
bad_true = captum_grad_x_input(mod_bad.model, torch.FloatTensor(X),
    target=torch.LongTensor(y))
print(bad_true.mean(axis=0))
tensor([ 0.0204, -0.0181,  0.0508,  0.0194], grad_fn=<MeanBackward1>)

# Attributions wrt the predicted labels:
bad_pred = mod_bad.predict(X)
bad_attr = captum_grad_x_input(mod_bad.model, torch.FloatTensor(X),
    target=torch.LongTensor(bad_pred))
print(bad_attr.mean(axis=0))
tensor([0.0112, 0.0168, 0.0558, 0.0740], grad_fn=<MeanBackward1>)
```

Gradients · inputs fails sensitivity

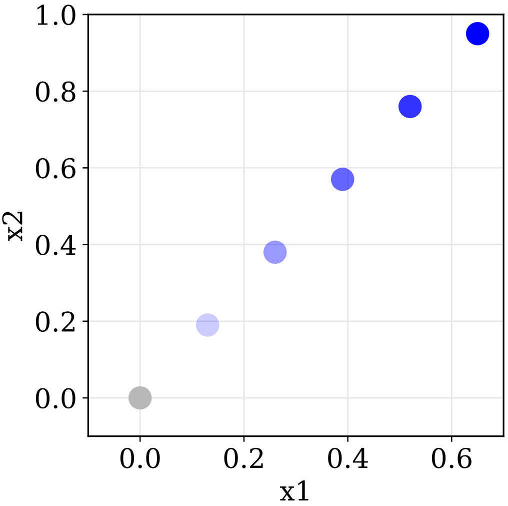
$$M(x) = 1 - \max(0, 1 - x)$$

$$\begin{aligned} M(0) &= 1 - \max(0, 1 - 0) && = 1 - 1 = 0 \\ M(2) &= 1 - \max(0, 1 - 2) && = 1 - 0 = 1 \end{aligned}$$

$$\begin{aligned} \text{InputXGradient}(M, 0) &= \max(0, \text{sign}(1 - 0)) \cdot 0 = 1 \cdot 0 &= 0 \\ \text{InputXGradient}(M, 2) &= \max(0, \text{sign}(1 - 2)) \cdot 2 = 0 \cdot 2 &= 0 \end{aligned}$$

Example from Sundararajan et al. 2017

Integrated gradients: Intuition



Integrated gradients: Core computation

$$\mathrm{IG}_i(M, x, x') = \overbrace{(x_i - x'_i)}^{5} \cdot \overbrace{\sum_{k=1}^{m}}^{4} \frac{\overbrace{\partial M(\overbrace{x' + \overbrace{\frac{k}{m}}^{1} \cdot (x - x')}^{2})}^{3}}{\partial x_i} \cdot \overbrace{\frac{1}{m}}^{4}$$

1. Generate $\alpha = [1, \ldots, m]$
2. Interpolate inputs between baseline x' and actual input x
3. Compute gradients for each interpolated input
4. Integral approximation through averaging
5. Scaling to remain in the space region as the original

Adapted from the TensorFlow integrated gradients tutorial

Sensitivity again

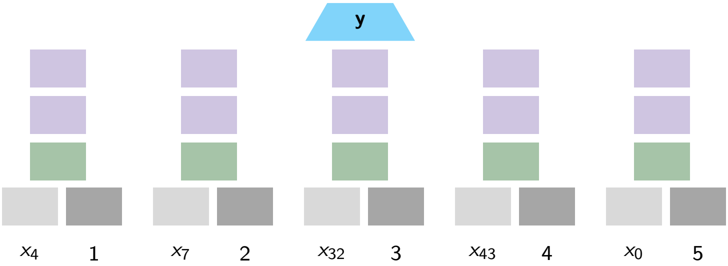
$$M(x) = 1 - \max(0, 1 - x)$$

$$\begin{aligned} M(0) &= 1 - \max(0, 1 - 0) & &= 1 - 1 = 0 \\ M(2) &= 1 - \max(0, 1 - 2) & &= 1 - 0 = 1 \end{aligned}$$

$$\begin{aligned} \text{InputXGradient}(M, 0) &= \max(0, \text{sign}(1 - 0)) \cdot 0 = 1 \cdot 0 &= 0 \\ \text{InputXGradient}(M, 2) &= \max(0, \text{sign}(1 - 2)) \cdot 2 = 0 \cdot 2 &= 0 \end{aligned}$$

$$\text{IG}_i(M, 2, 0) = (2 - 0) \cdot \sum \begin{pmatrix} \max(0, \text{sign}(1 - 0.00)) \\ \max(0, \text{sign}(1 - 0.02)) \\ \max(0, \text{sign}(1 - 0.04)) \\ \vdots \\ \max(0, \text{sign}(1 - 2.00)) \end{pmatrix} \cdot \frac{1}{m} \approx 1$$

BERT example



BERT example

```
import torch
import torch.nn.functional as F
from transformers import AutoModelForSequenceClassification,
    AutoTokenizer
from captum.attr import LayerIntegratedGradients
from captum.attr import visualization as viz

weights = 'cardiffnlp/twitter-roberta-base-sentiment'
tok = AutoTokenizer.from_pretrained(weights)
model = AutoModelForSequenceClassification.from_pretrained(weights)

def predict_one_proba(text):
    input_ids = tok.encode(text, add_special_tokens=True,
     return_tensors='pt')
    model.eval()
    with torch.no_grad():
        logits = model(input_ids)[0]
        preds = F.softmax(logits, dim=1)
    model.train()
    return preds.squeeze(0)

def ig_encodings(text):
    """Get base and source ids."""
    input_ids = tok.encode(text, add_special_tokens=False)
    base_ids = [tok.pad_token_id] * len(input_ids)
    input_ids = [tok.cls_token_id] + input_ids + [tok.sep_token_id]
    base_ids = [tok.cls_token_id] + base_ids + [tok.sep_token_id]
    return torch.LongTensor([input_ids]), torch.LongTensor([base_ids])
```

BERT example

```python
def ig_forward(inputs):
    return model(inputs).logits

#layer = model.roberta.encoder.layer[0]
layer = model.roberta.embeddings
ig = LayerIntegratedGradients(ig_forward, layer)

text = "This is illuminating!"
true_class = 2  # positive

input_ids, base_ids = ig_encodings(text)

# Attributions wrt to the true label:
attrs, delta = ig.attribute(input_ids, base_ids, target=true_class,
     return_convergence_delta=True)

# `scores` has dimension [1, 6, 768]
scores = attrs.sum(dim=-1)
# z-score normalize the attributions:
scores = (scores - scores.mean()) / scores.norm()

pred_probs = predict_one_proba(text)
pred_class = pred_probs.argmax()
```

BERT example

```python
toks = tok.convert_ids_to_tokens(input_ids.tolist()[0])
toks = [x.strip("Ġ") for x in toks]

score_vis = viz.VisualizationDataRecord(
    word_attributions=scores.squeeze(0),
    pred_prob=pred_probs.max(),
    pred_class=pred_class,
    true_class=true_class,
    attr_class=None,
    attr_score=attrs.sum(),
    raw_input_ids=toks,
    convergence_score=delta)

viz.visualize_text([score_vis])
```

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
2	2 (0.93)	None	2.70	#s This is illuminating ! #/s

A small challenge test

Attributions with respect to the true labels:

Legend: Away from true label, Neutral wrt true label, Toward true label

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
2	2 (0.82)	None	3.72	#s They said it would be great , and they were right . #/s
0	0 (0.50)	None	1.82	#s They said it would be great , and they were wrong . #/s
2	2 (0.76)	None	1.43	#s They were right to say it would be great . #/s
0	0 (0.62)	None	1.04	#s They were wrong to say it would be great . #/s
2	2 (0.77)	None	1.07	#s They said it would be stellar , and they were correct . #/s
0	1 (0.47)	None	1.09	#s They said it would be stellar , and they were incorrect . #/s

Summary

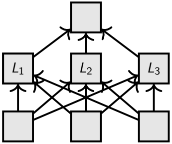
	Characterize representations	Causal inference	Improved models
Probing	😃		🤔
Feature attribution	🤔	😃	
Interventions	😃	😃	😃

Causal abstraction

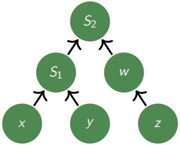
Recipe for causal abstraction

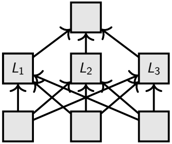
1. State a hypothesis about (an aspect of) the target model's causal structure.
2. Search for an alignment between the causal model and target model.
3. Perform *interchange interventions*.

Geiger et al. 2020, 2021

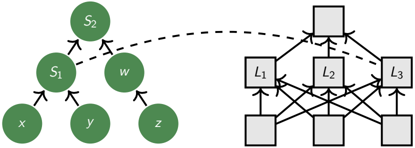


Our neural network successfully adds three numbers.
In human-interpretable terms, how does it do it?

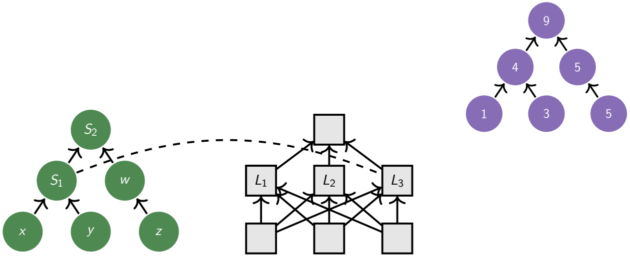




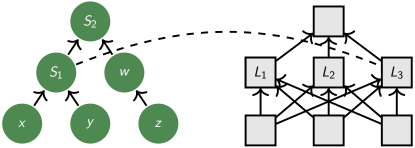
Our causal model adds the first two inputs to form an intermediate variable S_1.

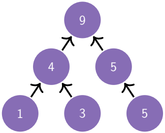


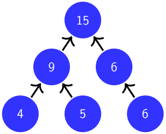
We hypothesize that the neural representation L_3 plays the same role as S_1.



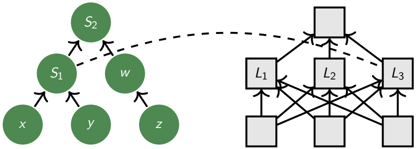
To test this, we run our causal model on $[1, 3, 5]$ and obtain output 9.

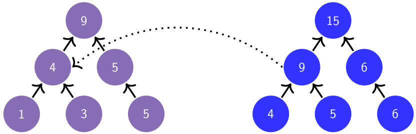




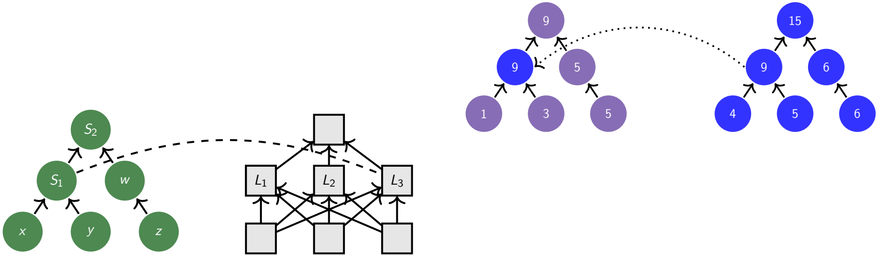


And we run the causal model on $[4, 5, 6]$ to get 15.

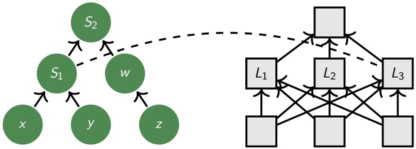


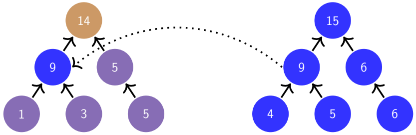


Then we perform an interchange intervention targeting the value of S_1.

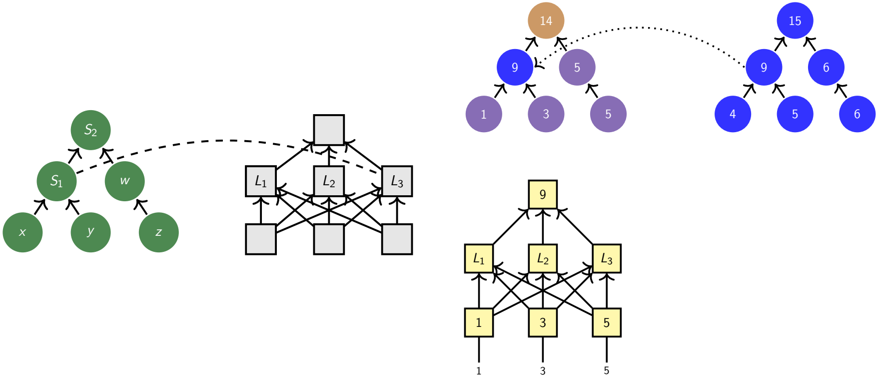


This changes the value of S_1 in the left example to 9.

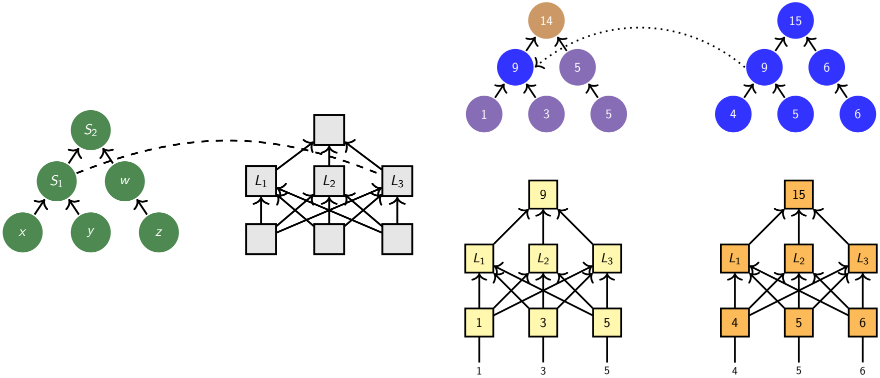




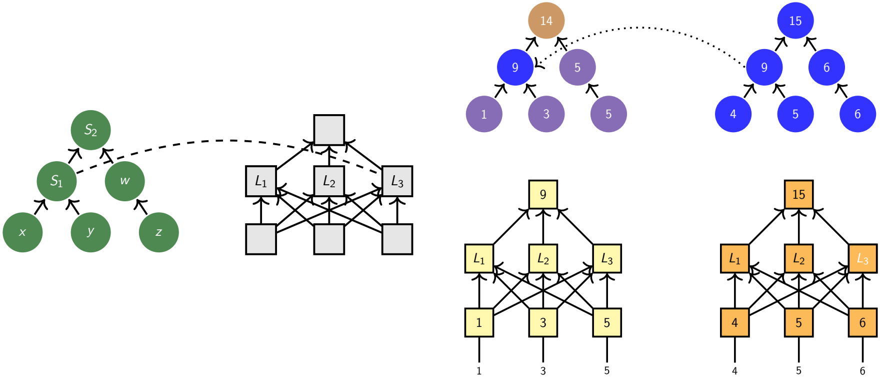
And this causes the model to output 14.



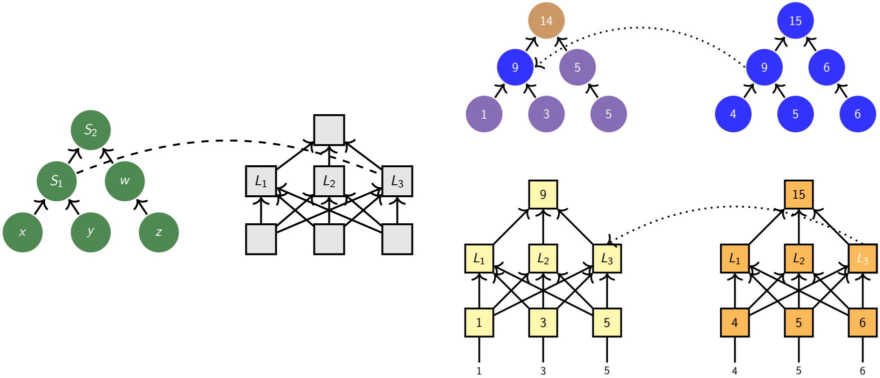
Will the neural network show the same behavior?



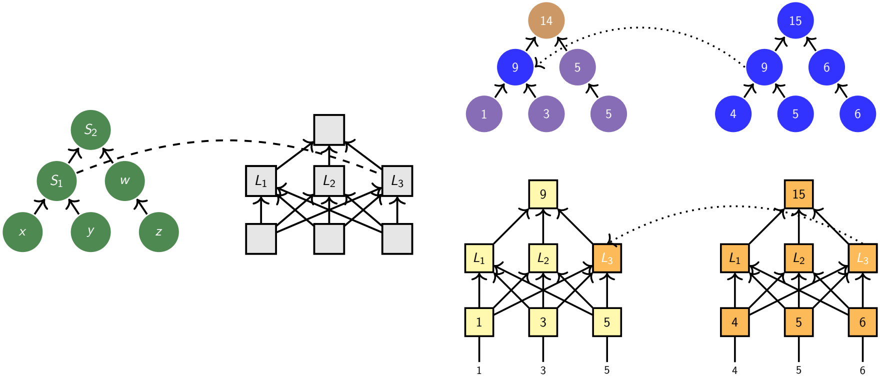
We process the same two examples.



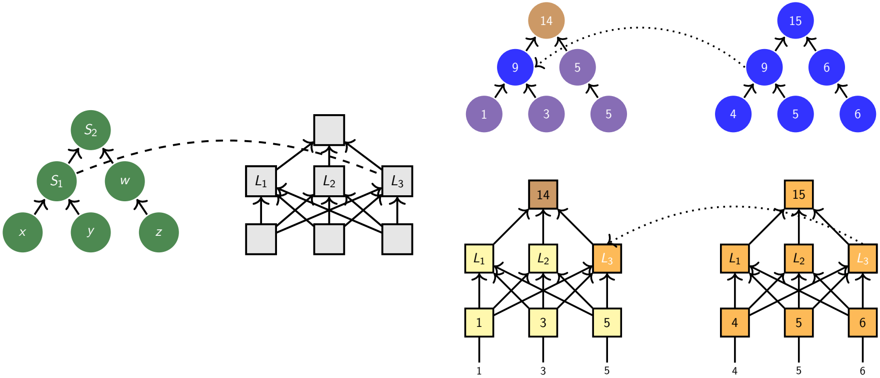
We hypothesized that L_3 plays the role of S_1.



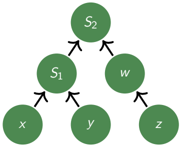
So we perform an intervention targeting L_3.

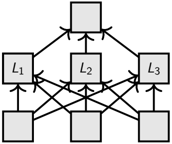


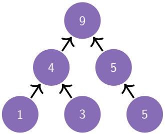
What is the effect of this intervention?

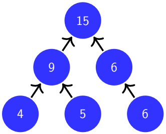


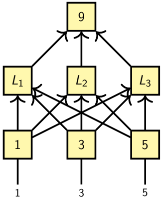
If this leads the network to output 14, we have a piece of evidence that L_3 plays the same role as S_1.

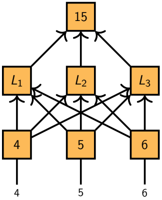

S_2
S_1
w
x
y
z

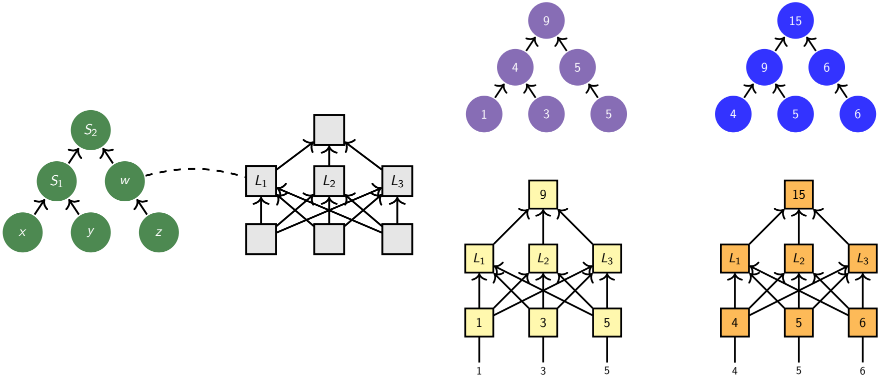

L_1
L_2
L_3


9
4
5
1
3
5

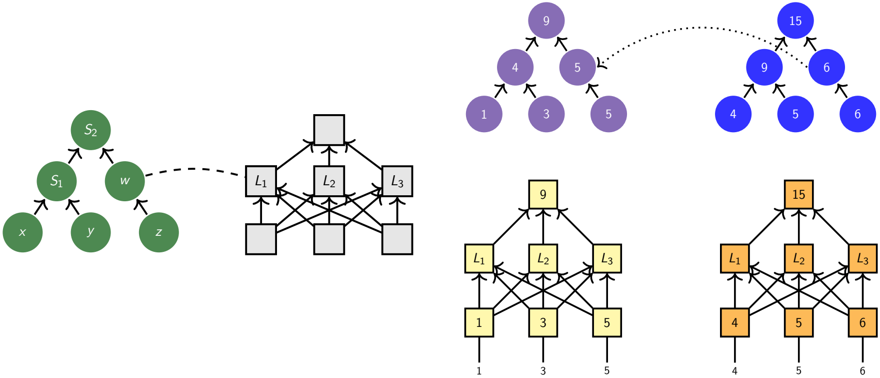

15
9
6
4
5
6


9
L_1
L_2
L_3
1
3
5
1
3
5

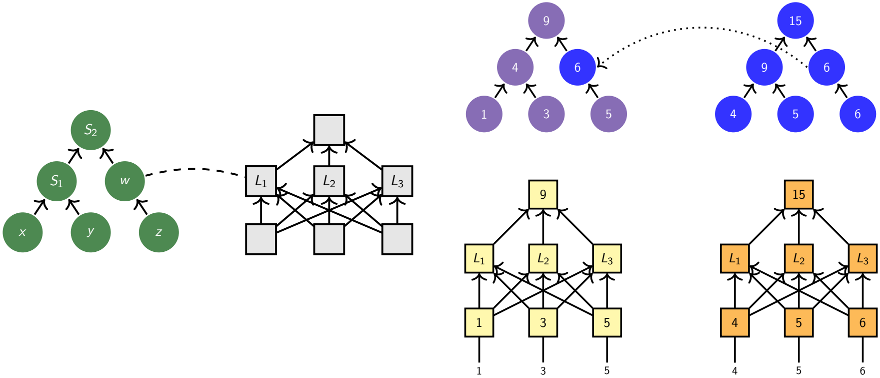

15
L_1
L_2
L_3
4
5
6
4
5
6



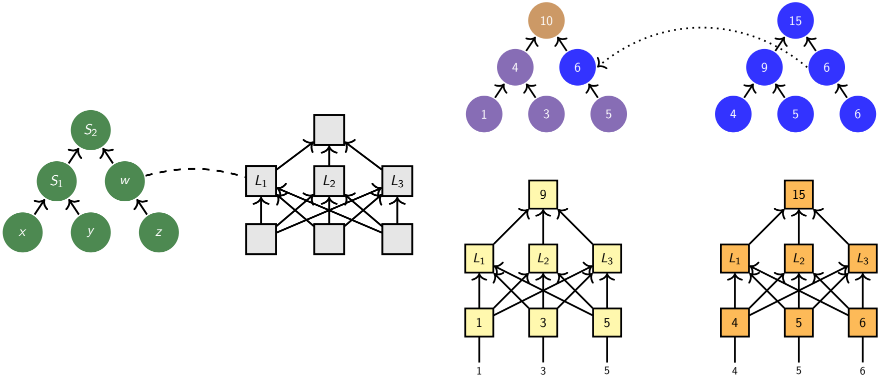
We can repeat the same process using the hypothesis that L_1 plays the role of w.



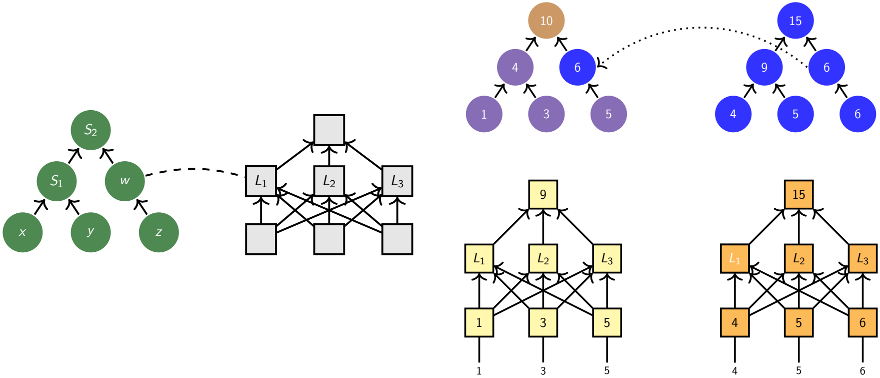
We first intervene on the causal model to get an output for this intervention.



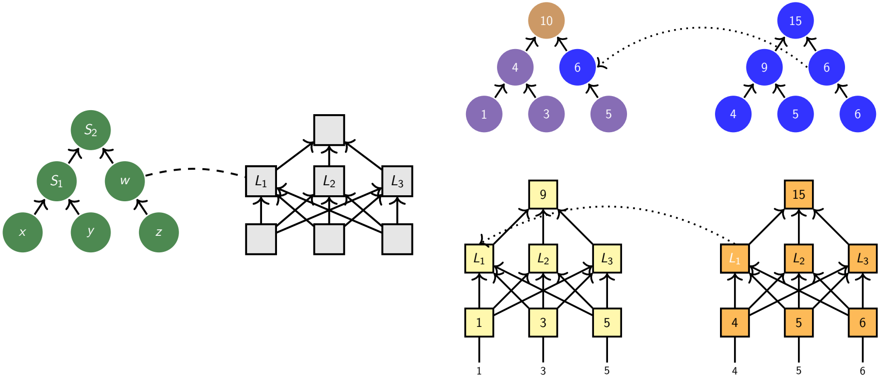
We first intervene on the causal model to get an output for this intervention.



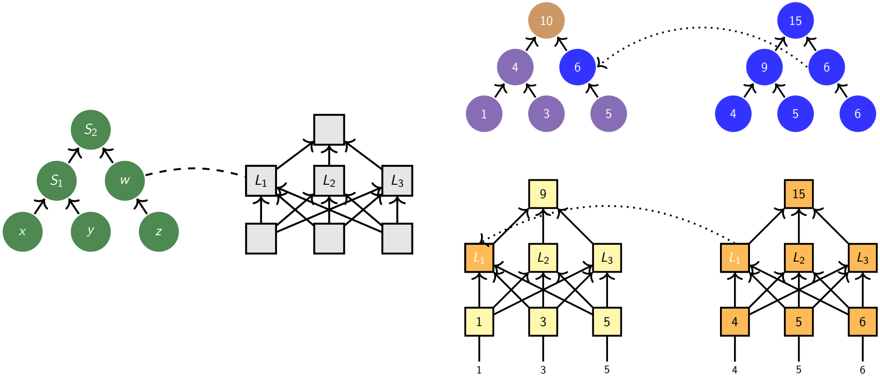
We first intervene on the causal model to get an output for this intervention.



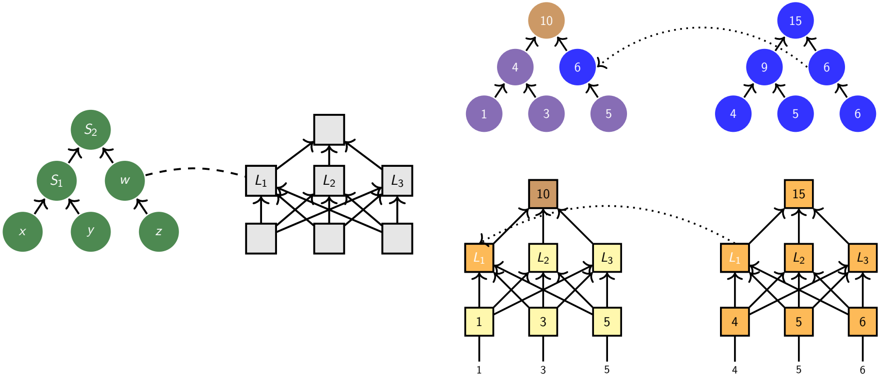
Then we intervene on the neural model.



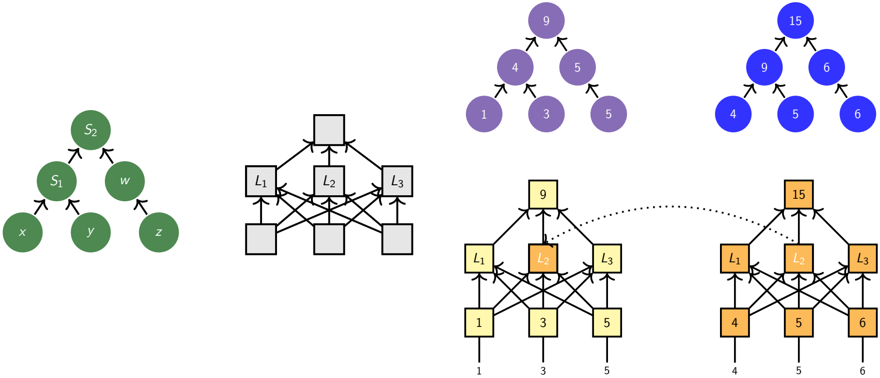
Then we intervene on the neural model.



Then we intervene on the neural model.



And we check whether the output corresponds to the output of the causal model under the aligned intervention.



Finally, if we intervene on L_2 and find that the output label never changes, then we have shown that it plays no role in the model's behavior.

Interchange intervention accuracy (IIA)

1. IIA is the percentage of interchange interventions that lead to outputs that match those of the causal model under the chosen alignment.
2. IIA is scaled in $[0, 1]$, as with a normal accuracy metric.
3. IIA can actually be above task performance, if the interchange interventions put the model into a better state.
4. IIA is extremely sensitive to the set of interchange interventions one does.
5. Pay particular attention to how many interchange interventions *should* change the output label, since they provide the clearest evidence.

Findings from causal abstraction

1. Fine-tuned BERT models succeed at hard, out-of-domain examples involving lexical entailment and negation because they are abstracted by simple monotonicity programs (Geiger et al. 2020).
2. Fine-tuned BERT models succeed at the MQNLI task because they find compositional solutions (Geiger et al. 2021).
3. Models succeed at the MNIST Pointer Value Retrieval task (MNIST-PVR; Zhang et al. 2021) because they are abstracted by simple programs like "if the digit is 6, then the label is in the lower left" (Geiger et al. 2021).
4. BART and T5 use coherent entity and situation representations that evolve as the discourse unfolds (Li et al. 2021).
5. This course notebook is a hands-on introduction to these techniques: `https://github.com/cgpotts/cs224u/blob/main/iit_equality.ipynb`

Connections to the literature

- Constructive abstraction (Beckers et al. 2020)
- Causal mediation analysis (Vig et al. 2020)
- Role Learning Networks (Soulos et al. 2020)
- CausaLM (Feder et al. 2021)
- Amnesic Probing (Elazar et al. 2021)
- Circuits (Cammarata et al. 2020; Olsson et al. 2022; Wang et al. 2022)
- Causal scrubbing (LawrenceC et al. 2022)

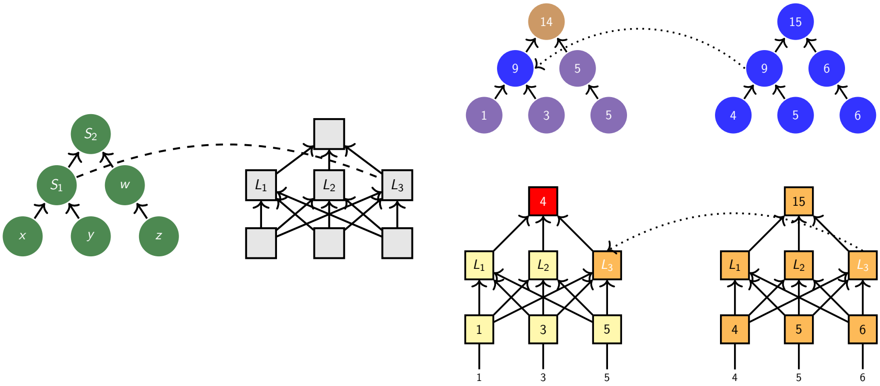
For more:
`https://ai.stanford.edu/blog/causal-abstraction/`

Summary

	Characterize representations	Causal inference	Improved models
Probing	😃		🤔
Feature attribution	🤔	😃	
Interventions	😃	😃	😃

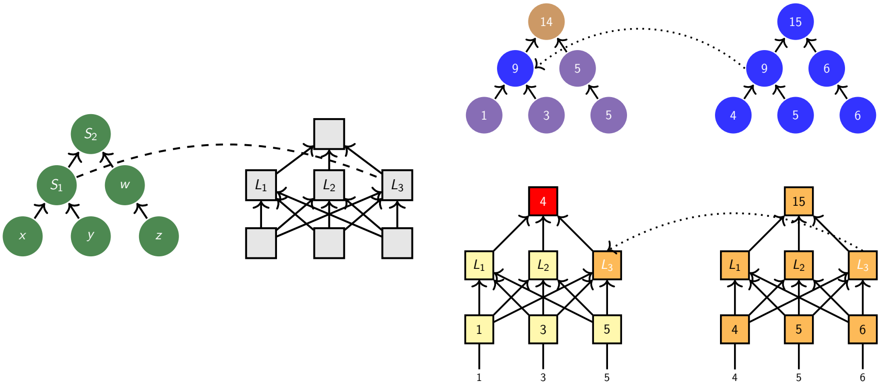
Interchange Intervention Training (IIT)

Method



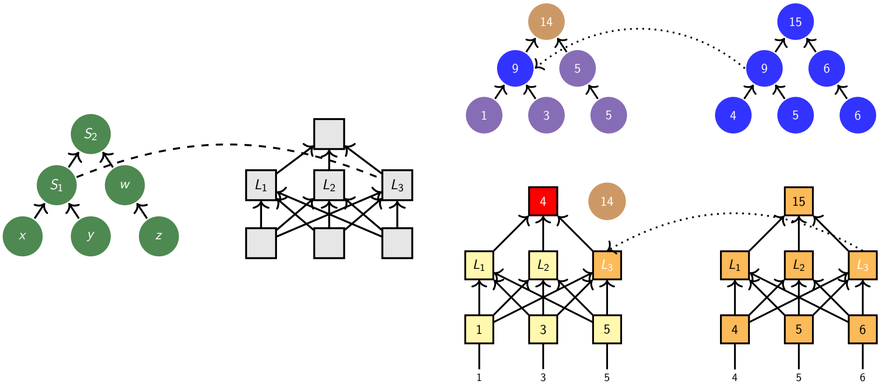
Suppose our network doesn't agree with the causal model under our intervention.

Method



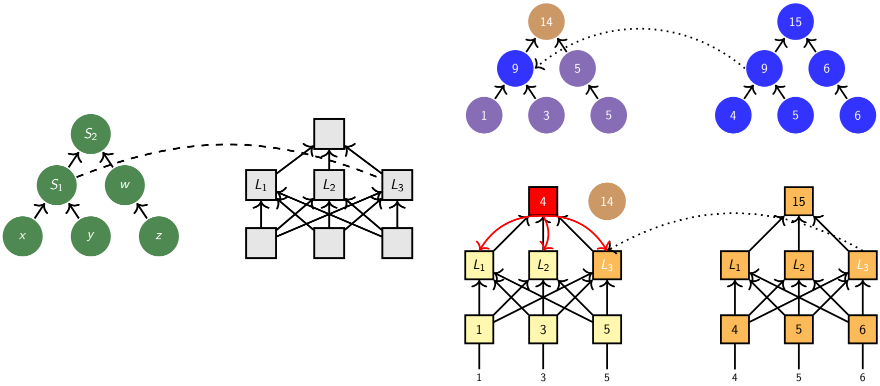
We can correct that misalignment with interchange intervention training.

Method



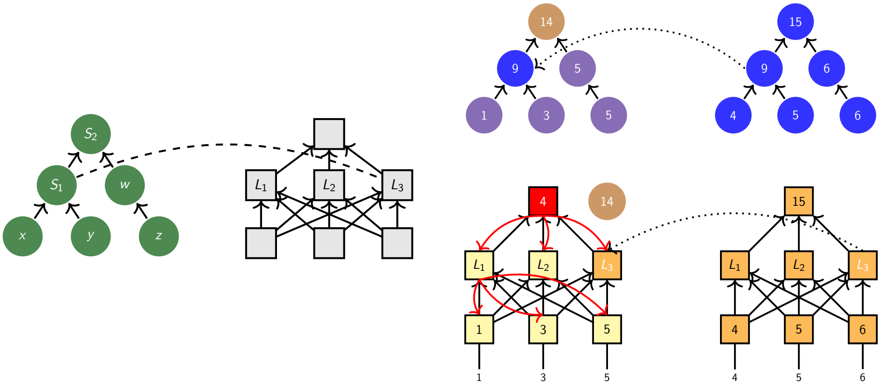
The causal model provides us with a true label, and a comparison with the incorrect prediction gives us an error signal.

Method



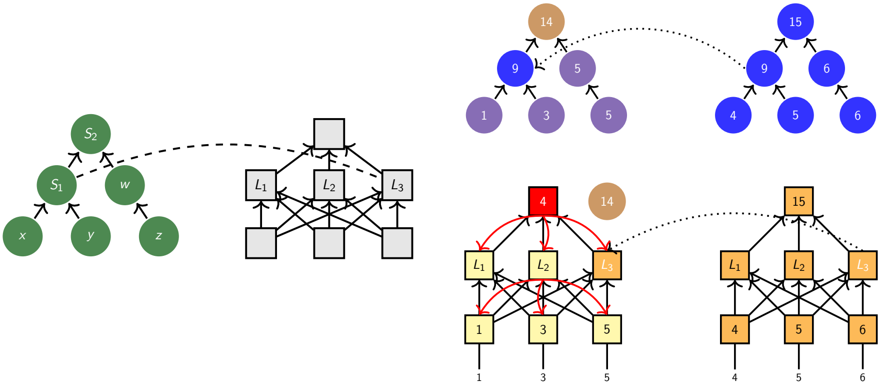
The gradients flow from this node to the top hidden layer as usual.

Method



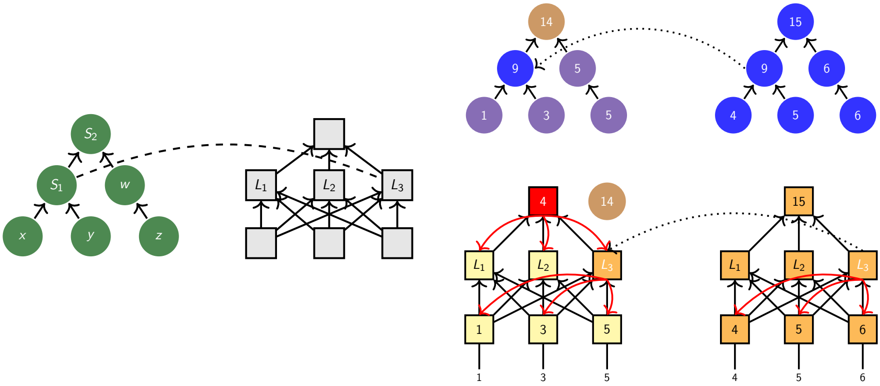
And the gradients flow as usual for the left and center hidden states.

Method



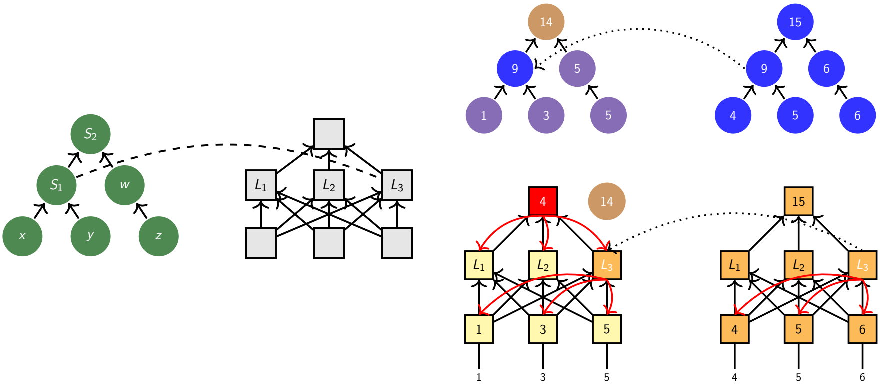
And the gradients flow as usual for the left and center hidden states.

Method



But the intervention site receives a double update, from the target example and the source example at right.

Method



This process gradually brings L_3 into alignment with S_1.

Findings from IIT

1. Geiger et al. (2022b) develop IIT and use it to achieve state-of-the-art results on the MNIST Pointer Value Retrieval task (MNIST-PVR; Zhang et al. 2021) and the ReaSCAN grounded language understanding benchmark (Wu et al. 2021).
2. Wu et al. (2022b) augment the standard distillation objectives (Sanh et al. 2019) with an IIT objective and show that it improves over standard distillation techniques.
3. Huang et al. (2022) use IIT to induce internal representations of characters in LMs based in subword tokenization, and they show that this helps with a variety of character-level games and tasks.
4. Wu et al. (2022a) use IIT to create concept-level methods for explaining model behavior.
5. Our course notebook covers IIT as well as causal abstraction: `https://github.com/cgpotts/cs224u/blob/main/iit_equality.ipynb`

Summary

	Characterize representations	Causal inference	Improved models
Probing	😃		🤔
Feature attribution	🤔	😃	
Interventions	😃	😃	😃

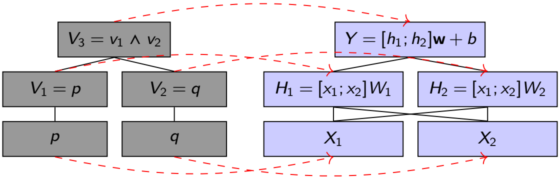
Distributed Alignment Search (DAS)

Our scorecard again

	Characterize representations	Causal inference	Improved models
Probing	😃		🤔
Feature attribution	🤔	😃	
Interventions	😃	😃	😃

- Alignment search is expensive.
- Causal abstraction could fail to find genuine causal structure.

A simple causal abstraction analysis



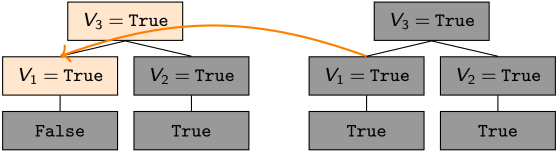
$$W_1 = \begin{bmatrix} \cos(20°) & -\sin(20°) \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} \sin(20°) & \cos(20°) \end{bmatrix} \qquad b = -1.8$$

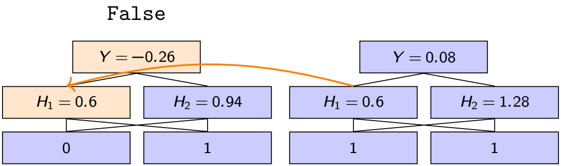
The high-level model **does not abstract** the new neural model under our chosen alignment.

Interchange intervention failure

An interchange intervention on the high-level model:

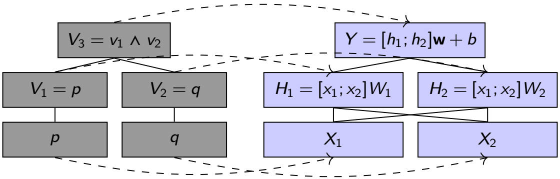


The aligned interchange intervention on the neural model:



The two models have **unequal counterfactual predictions**

But the relationship holds in a non-standard basis



$$W_1 = \begin{bmatrix} \cos(20°) & -\sin(20°) \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

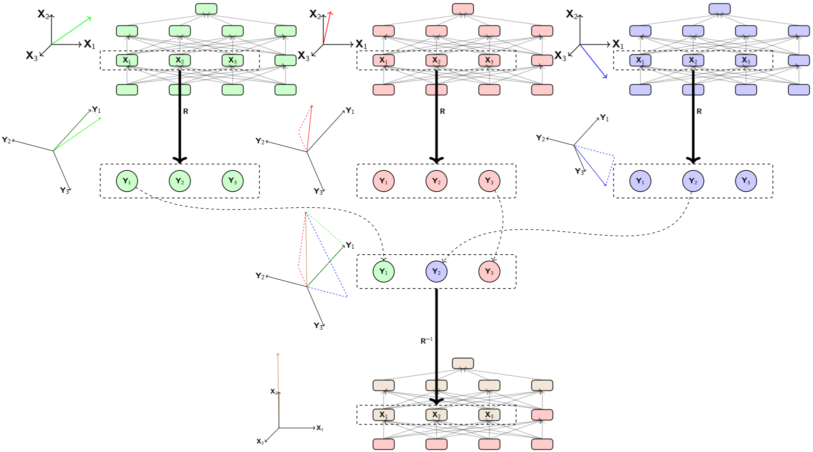
$$W_2 = \begin{bmatrix} \sin(20°) & \cos(20°) \end{bmatrix} \qquad b = -1.8$$

View $[H_1, H_2]$ under a non-standard basis by rotating $-20°$

$$\begin{bmatrix} \cos(-20°) & -\sin(-20°) \\ \sin(-20°) & \cos(-20°) \end{bmatrix}$$

Problem: Causal abstraction missed this because of the standard basis we chose. But our choice of basis was arbitrary!

Solution: Distributed Interchange Intervention



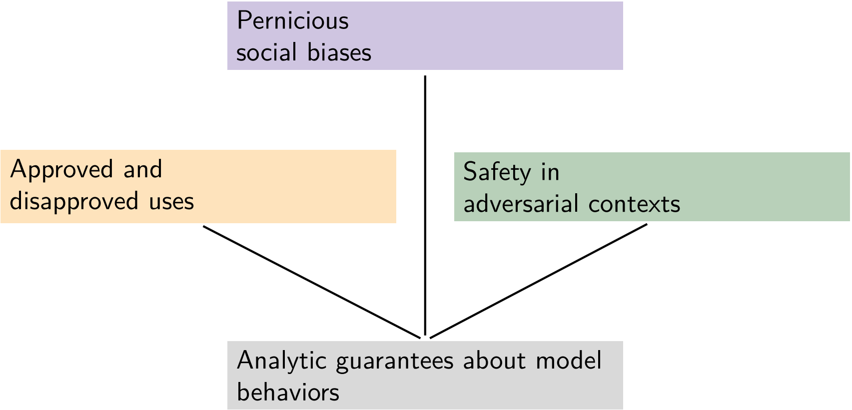
Freeze the model parameters and **learn** a rotation matrix with distributed interchange intervention training.

Findings from DAS

1. Geiger et al. (2023): Models learn truly hierarchical solutions to the hierarchical equality task from our notebook, but these solutions are easy to miss with standard causal abstraction.
2. Geiger et al. (2023): Models learn theories of lexical entailment and negation, but in a brittle way that preserves the identities of the lexical items rather than truly learning a general solution to entailment.
3. Wu et al. (2023): Alpaca implements an intuitive algorithm to solve a numerical reasoning task.

Conclusions

Reminder: A crucial prerequisite



The near future of explainability research

1. Causal explanations
2. Human-interpretable explanations
3. Applications to ever-larger Instruct-trained LLMs
4. Increasing evidence that models are inducing a semantics: a mapping from language into network of concepts.

References I

Sander Beckers, Frederick Eberhardt, and Joseph Y. Halpern. 2020. Approximate causal abstractions. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 606–615. PMLR.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. 2020. Thread: Circuits. *Distill*. Https://distill.pub/2020/circuits.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. 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. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9(0):160–175.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71.

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

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. 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. Association for Computational Linguistics.

Atticus Geiger, Zhengxuan Wu, Karel D'Oosterlinck, Elisa Kreiss, Noah D. Goodman, Thomas Icard, and Christopher Potts. 2022a. Faithful, interpretable model explanations via causal abstraction. Stanford AI Lab Blog.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. 2022b. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7324–7338. PMLR.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2023. Finding alignments between interpretable causal variables and distributed neural representations. Ms., Stanford University.

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

References II

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Jing Huang, Zhengxuan Wu, Kyle Mahowald, and Christopher Potts. 2022. Inducing character-level structure in subword-based language models with Type-level Interchange Intervention Training. Ms., Stanford University and UT Austin.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

LawrenceC, Adrià Garriga-alonso, Nicholas Goldowsky-Dill, ryan_greenblatt, jenny, Ansh Radhakrishnan, Buck, and Nate Thomas. 2022. Causal scrubbing: a method for rigorously testing interpretability hypotheses. Blog post, Redwood Research.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online. Association for Computational Linguistics.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. ArXiv:2002.12327.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org.

References III

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Paul Soulos, R. Thomas McCoy, Tal Linzen, and Paul Smolensky. 2020. Discovering the compositional structure of vector representations with role learning networks. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 238–254, Online. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.

Zhengxuan Wu, Karel D'Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. 2022a. Causal Proxy Models for concept-based model explanations. ArXiv:2209.14279.

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D. Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in Alpaca. Ms., Stanford University.

Zhengxuan Wu, Atticus Geiger, Josh Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah D. Goodman. 2022b. Causal distillation for language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4295, Seattle, United States. Association for Computational Linguistics.

Zhengxuan Wu, Elisa Kreiss, Desmond C. Ong, and Christopher Potts. 2021. ReaSCAN: Compositional reasoning in language grounding. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Chiyuan Zhang, Maithra Raghu, Jon M. Kleinberg, and Samy Bengio. 2021. Pointer value retrieval: A new benchmark for understanding the limits of neural network generalization. *CoRR*, abs/2107.12580.