Bridging the Divide Between Linguistics and NLP: From Vectors to Symbols and Back Again

Tom McCoy

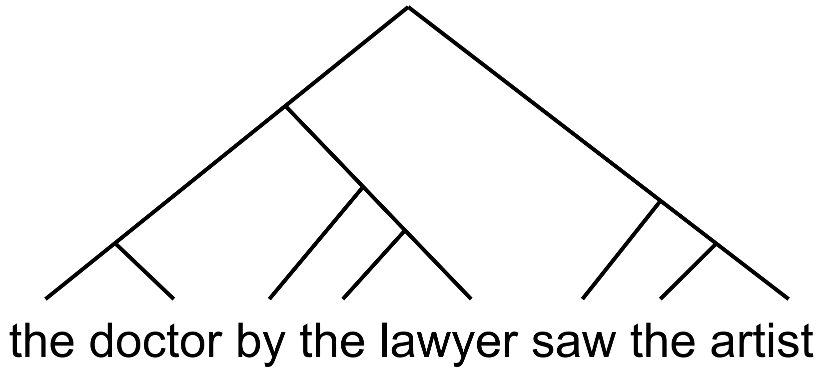
June 11, 2025

Yale *Department of Linguistics*

What type of system should we use to represent language?

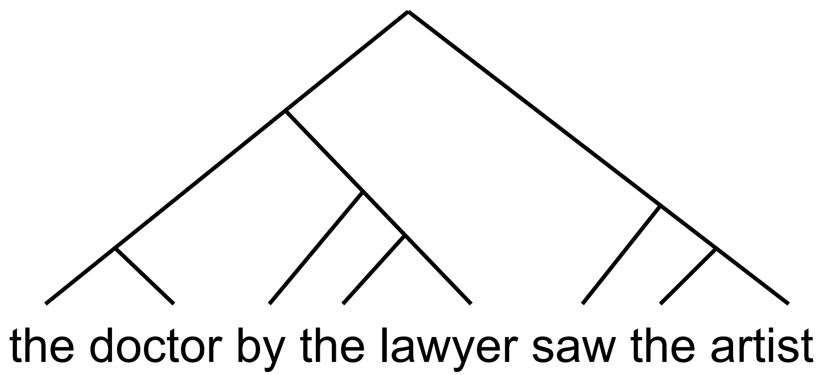
An apparent paradox

- Traditional view: Symbolic structure



An apparent paradox

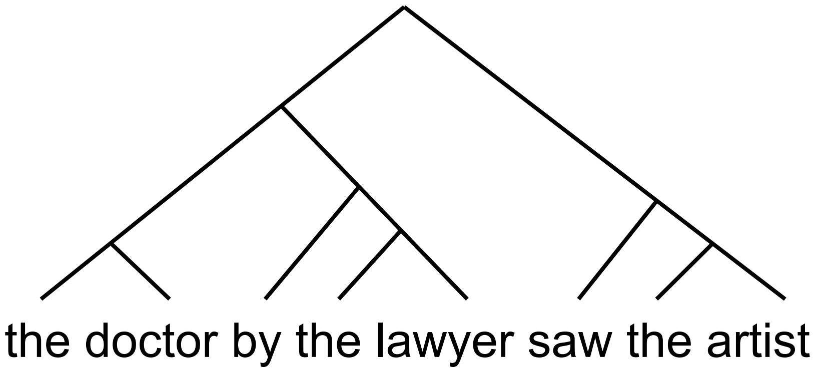
- Traditional view: Symbolic structure



- Neural networks: Vector representations

An apparent paradox

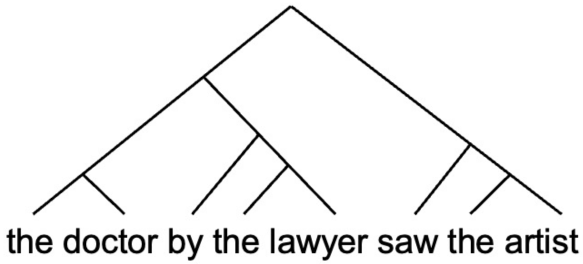
- Traditional view: Symbolic structure



- Neural networks: Vector representations

```
[-1.4792, 1.7206, -0.74454, -0.66112, -0.24197, 0.54995, -0.553
69, -0.35791, -0.27327, 0.56298, 0.48538, 0.1956, 0.76609, 0.00
14633, -0.49319, -0.25365, -0.60558, 1.3433, -0.47967, 1.0888,
0.93805, 1.1932, -0.03101, -0.29201, -0.4451, -1.252, -0.09721,
 -1.168, -0.37394, 0.24645, 1.5268, 0.12353, -0.98737, -0.82833
, 0.84111, 0.48287, -0.45142, 0.9825, 1.3721, -0.34363, -0.1575
8, -0.34484, 0.0048065, 0.84408, -1.3145, -0.61091, 0.7188, -0.
19112, -0.090826, 0.26399]
```

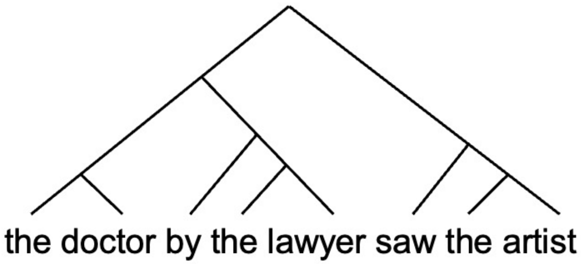
Symbolic structures



Vectors

```
[-1.4792, 1.7206, -0.74454, -0.66112, -0.24197, 0.54995, -0.553
69, -0.35791, -0.27327, 0.56298, 0.48538, 0.1956, 0.76609, 0.00
14633, -0.49319, -0.25365, -0.60558, 1.3433, -0.47967, 1.0888,
0.93805, 1.1932, -0.03101, -0.29201, -0.4451, -1.252, -0.09721,
 -1.168, -0.37394, 0.24645, 1.5268, 0.12353, -0.98737, -0.82833
, 0.84111, 0.48287, -0.45142, 0.9825, 1.3721, -0.34363, -0.1575
8, -0.34484, 0.0048065, 0.84408, -1.3145, -0.61091, 0.7188, -0.
19112, -0.090826, 0.26399]
```

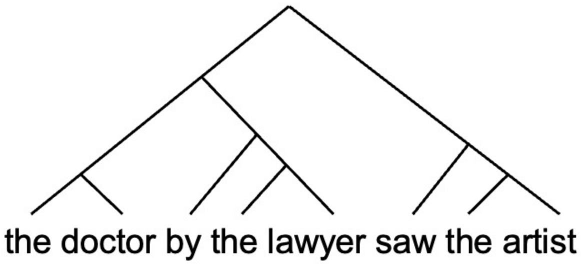
Symbolic structures



Vectors

[−1.4792, 1.7206, −0.74454, −0.66112, −0.24197, 0.54995, −0.553
69, −0.35791, −0.27327, 0.56298, 0.48538, 0.1956, 0.76609, 0.00
14633, −0.49319, −0.25365, −0.60558, 1.3433, −0.47967, 1.0888,
0.93805, 1.1932, −0.03101, −0.29201, −0.4451, −1.252, −0.09721,
−1.168, −0.37394, 0.24645, 1.5268, 0.12353, −0.98737, −0.82833
, 0.84111, 0.48287, −0.45142, 0.9825, 1.3721, −0.34363, −0.1575
8, −0.34484, 0.0048065, 0.84408, −1.3145, −0.61091, 0.7188, −0.
19112, −0.090826, 0.26399]

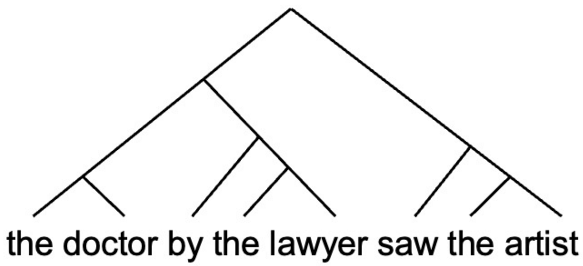
Symbolic structures



Vectors

[-1.4792, 1.7206, -0.74454, -0.66112, -0.24197, 0.54995, -0.553
69, -0.35791, -0.27327, 0.56298, 0.48538, 0.1956, 0.76609, 0.00
14633, -0.49319, -0.25365, -0.60558, 1.3433, -0.47967, 1.0888,
0.93805, 1.1932, -0.03101, -0.29201, -0.4451, -1.252, -0.09721,
 -1.168, -0.37394, 0.24645, 1.5268, 0.12353, -0.98737, -0.82833
, 0.84111, 0.48287, -0.45142, 0.9825, 1.3721, -0.34363, -0.1575
8, -0.34484, 0.0048065, 0.84408, -1.3145, -0.61091, 0.7188, -0.
19112, -0.090826, 0.26399]

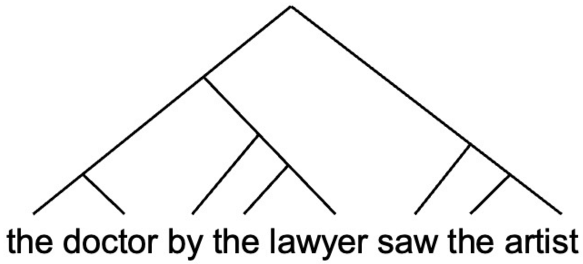
Symbolic structures



Vectors

```
[-1.4792, 1.7206, -0.74454, -0.66112, -0.24197, 0.54995, -0.553
69, -0.35791, -0.27327, 0.56298, 0.48538, 0.1956, 0.76609, 0.00
14633, -0.49319, -0.25365, -0.60558, 1.3433, -0.47967, 1.0888,
0.93805, 1.1932, -0.03101, -0.29201, -0.4451, -1.252, -0.09721,
 -1.168, -0.37394, 0.24645, 1.5268, 0.12353, -0.98737, -0.82833
, 0.84111, 0.48287, -0.45142, 0.9825, 1.3721, -0.34363, -0.1575
8, -0.34484, 0.0048065, 0.84408, -1.3145, -0.61091, 0.7188, -0.
19112, -0.090826, 0.26399]
```

Symbolic structures

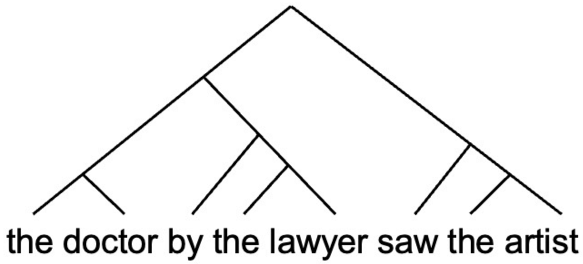


Girl with a Pearl Earring

Vectors

```
[-1.4792, 1.7206, -0.74454, -0.66112, -0.24197, 0.54995, -0.553
69, -0.35791, -0.27327, 0.56298, 0.48538, 0.1956, 0.76609, 0.00
14633, -0.49319, -0.25365, -0.60558, 1.3433, -0.47967, 1.0888,
0.93805, 1.1932, -0.03101, -0.29201, -0.4451, -1.252, -0.09721,
 -1.168, -0.37394, 0.24645, 1.5268, 0.12353, -0.98737, -0.82833
, 0.84111, 0.48287, -0.45142, 0.9825, 1.3721, -0.34363, -0.1575
8, -0.34484, 0.0048065, 0.84408, -1.3145, -0.61091, 0.7188, -0.
19112, -0.090826, 0.26399]
```

Symbolic structures



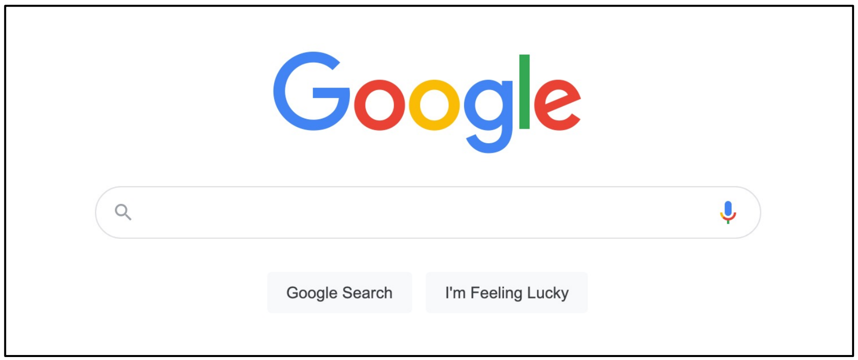
Girl with a Pearl Earring

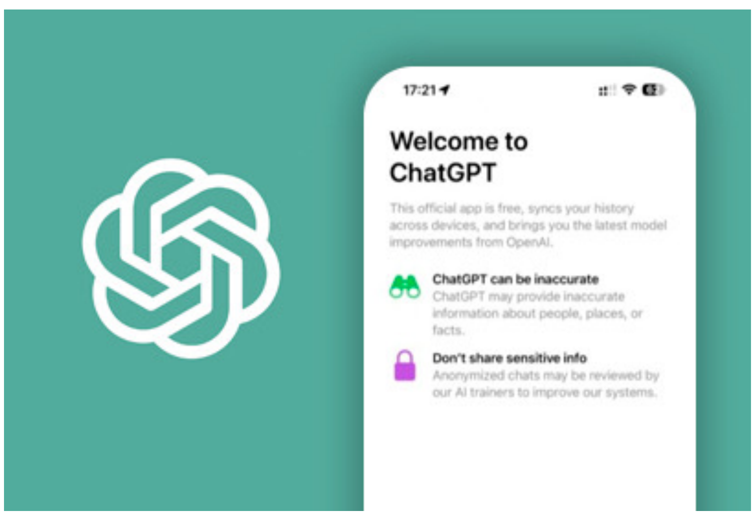
Vectors

[-1.4792, 1.7206, -0.74454, -0.66112, -0.24197, 0.54995, -0.553
69, -0.35791, -0.27327, 0.56298, 0.48538, 0.1956, 0.76609, 0.00
14633, -0.49319, -0.25365, -0.60558, 1.3433, -0.47967, 1.0888,
0.93805, 1.1932, -0.03101, -0.29201, -0.4451, -1.252, -0.09721,
-1.168, -0.37394, 0.24645, 1.5268, 0.12353, -0.98737, -0.82833
, 0.84111, 0.48287, -0.45142, 0.9825, 1.3721, -0.34363, -0.1575
8, -0.34484, 0.0048065, 0.84408, -1.3145, -0.61091, 0.7188, -0.
19112, -0.090826, 0.26399]

Jackson Pollock

The surprising success of neural nets





Q: What type of system should we use to represent language?

Q: What type of system should we use to represent language?

A: Vectors that act like symbols

Q: What type of system should we use to represent language?

A: Vectors that <u>act like symbols</u>

Q: What type of system should we use to represent language?

A: Vectors that act like symbols

Representations

Learning

Q: What type of system should we use to represent language?

A: Vectors that act like symbols

Representations

Learning

“Vectors that act like symbols”

- Why not just “vectors”?
- Deep learning already excels in:
  - Language
  - Math
  - Board games
  - …
- Answer: Despite appearances, deep learning systems already have implicit symbolic structure!

Simplified case

- Number-to-word mapping:

16 5 40 → sixteen five forty

Simplified case

- Number-to-word mapping:

16 5 40 → sixteen five forty

Requires symbolic structure: Which numbers and where?

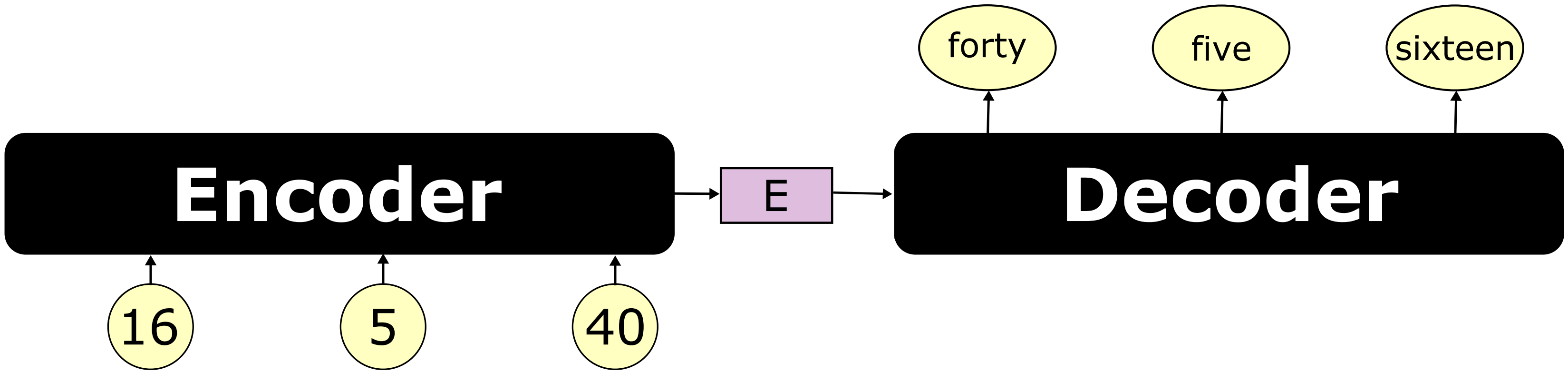
- Number-to-word mapping (reversed):

16 5 40 → forty five sixteen

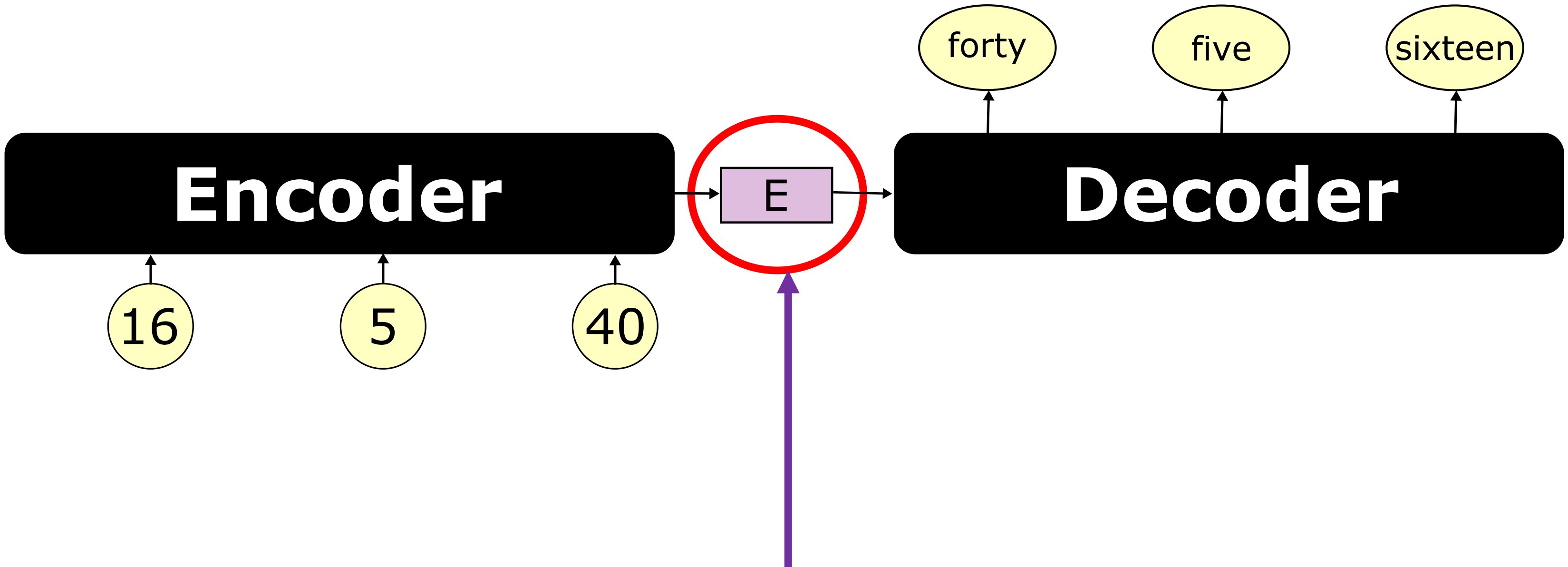
Models

- Two sub-networks:
  - Encoder
  - Decoder

Models



Models



Goal: Understand the encoding vector

Models

- RNN
- Transformer
- MLP

Hypothesis: Neural networks implicitly implement symbolic representations

Hypothesis: Neural networks implicitly implement symbolic representations

But how?

Hypothesis: Neural networks implicitly implement symbolic representations

But how?

By constructing Tensor Product Representations

Tensor Product Representations

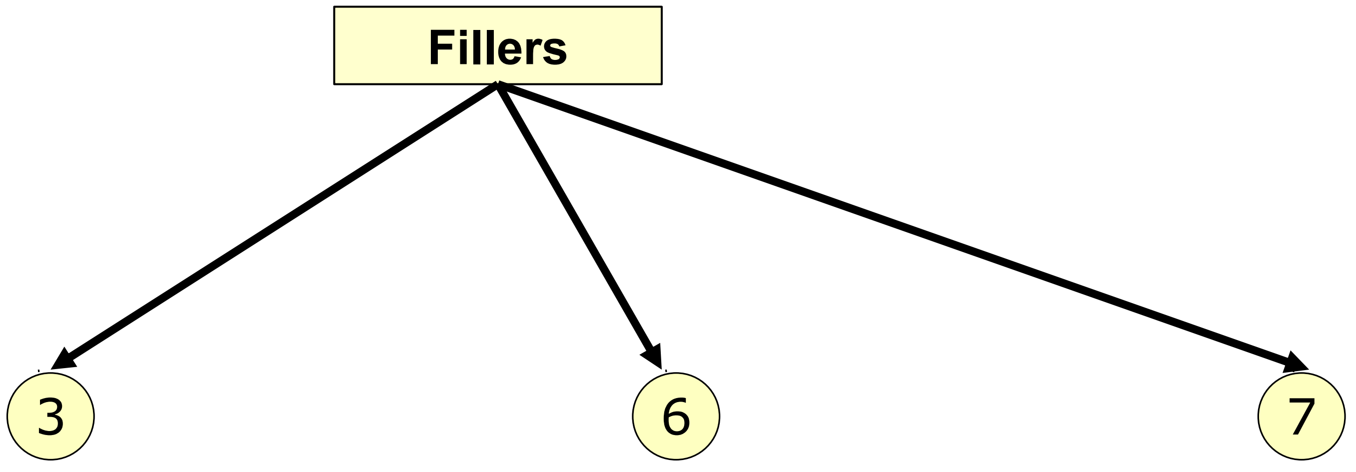
Smolensky (1987)

Tensor Product Representations

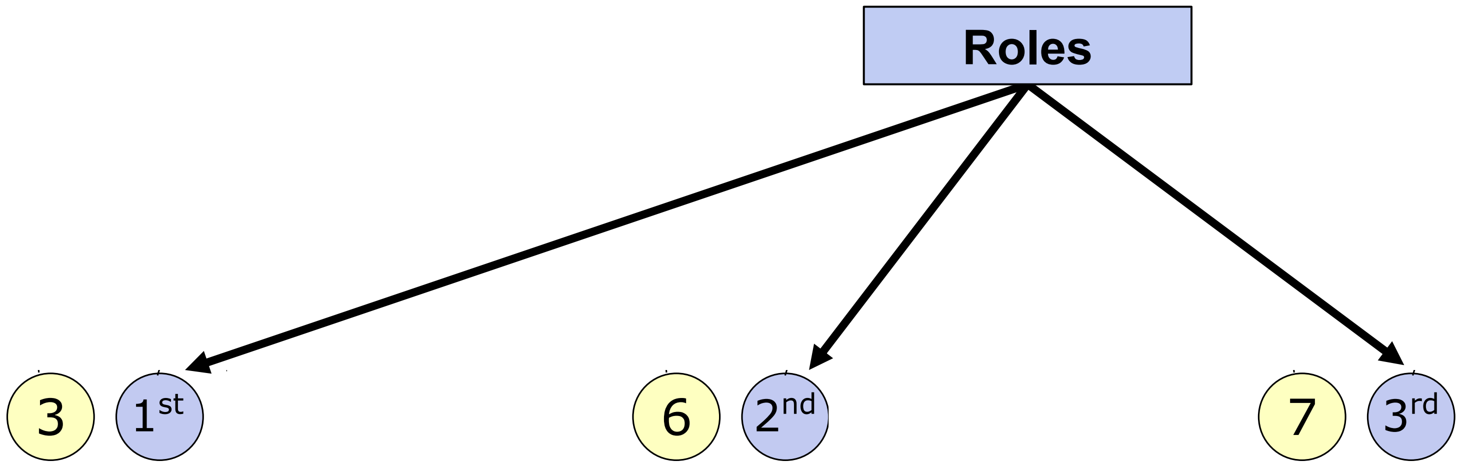
- Example: Representing a sequence of numbers: 3, 6, 7

Tensor Product Representations

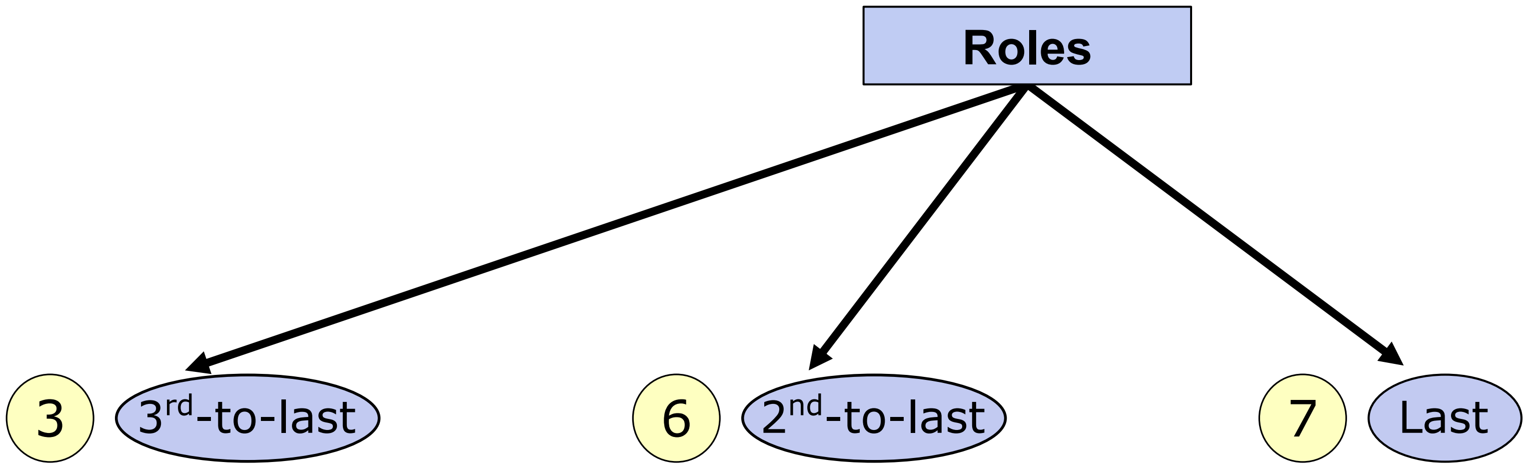
- Example: Representing a sequence of numbers: 3, 6, 7



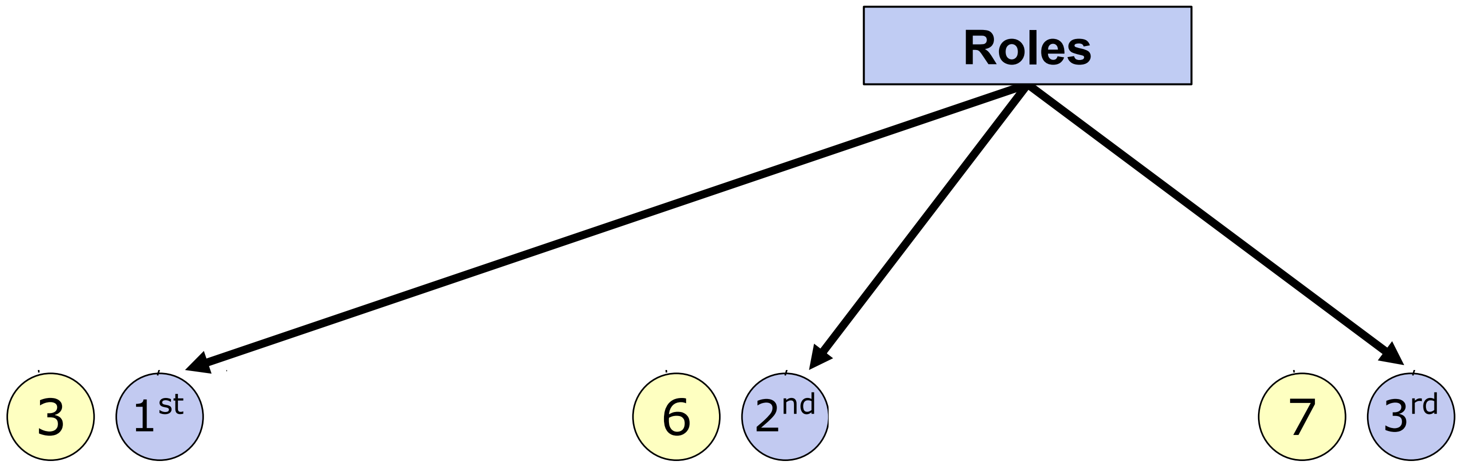
Tensor Product Representations



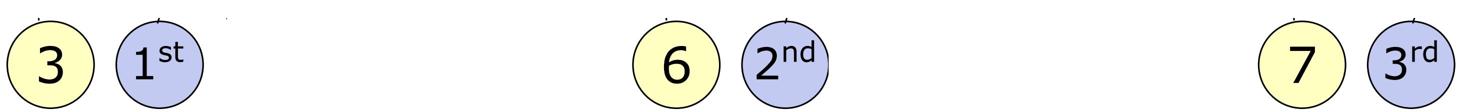
Tensor Product Representations



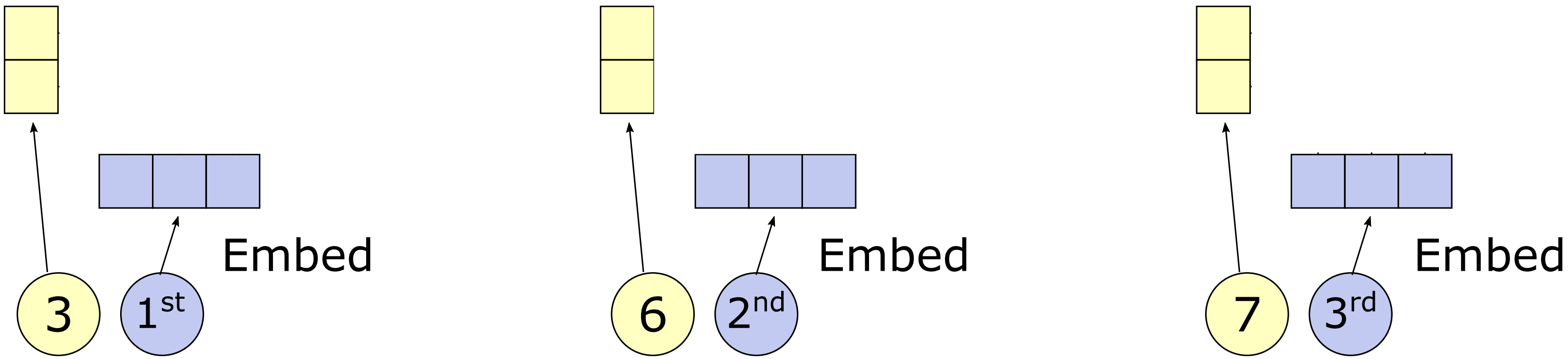
Tensor Product Representations



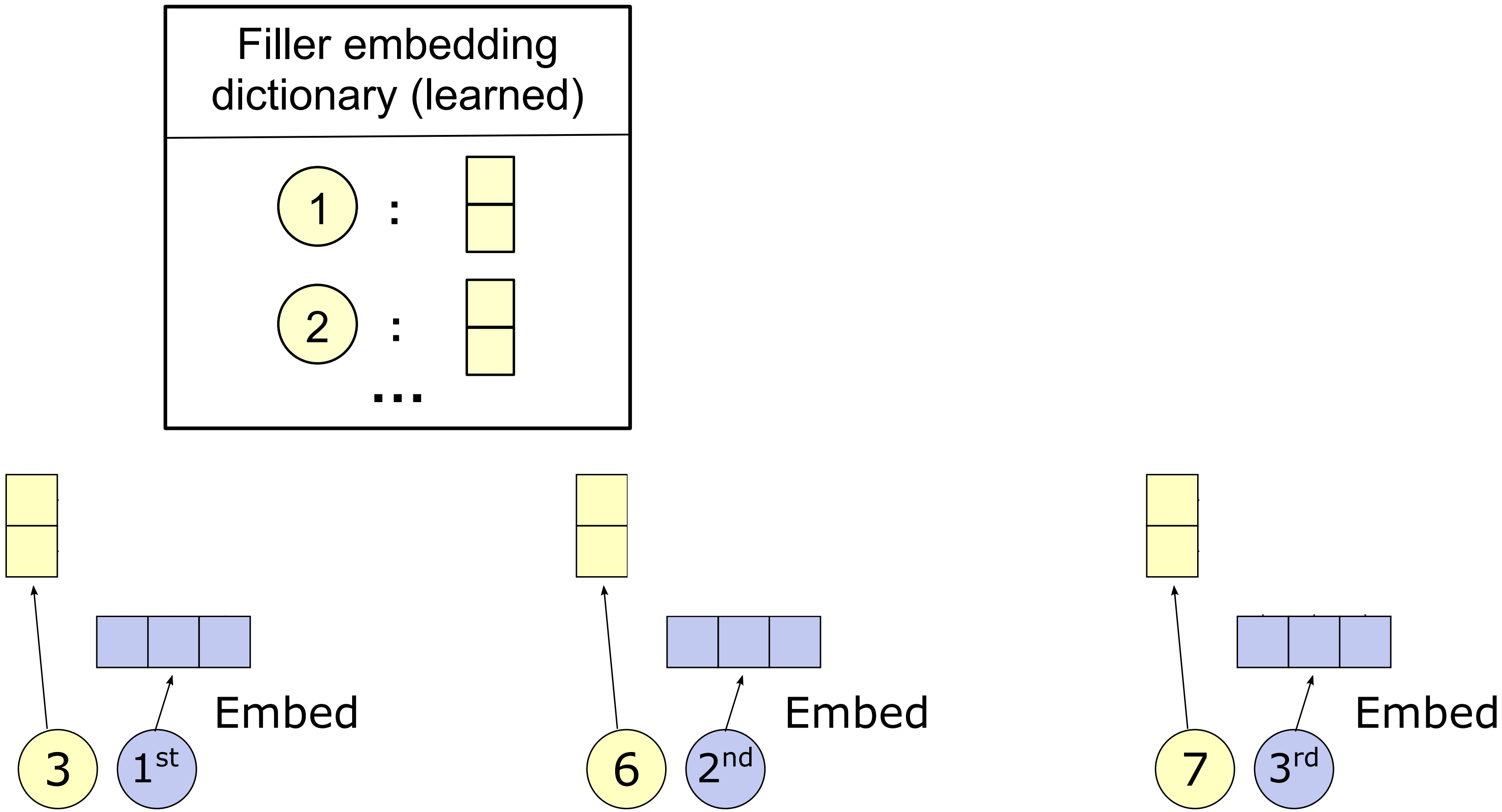
Tensor Product Representations



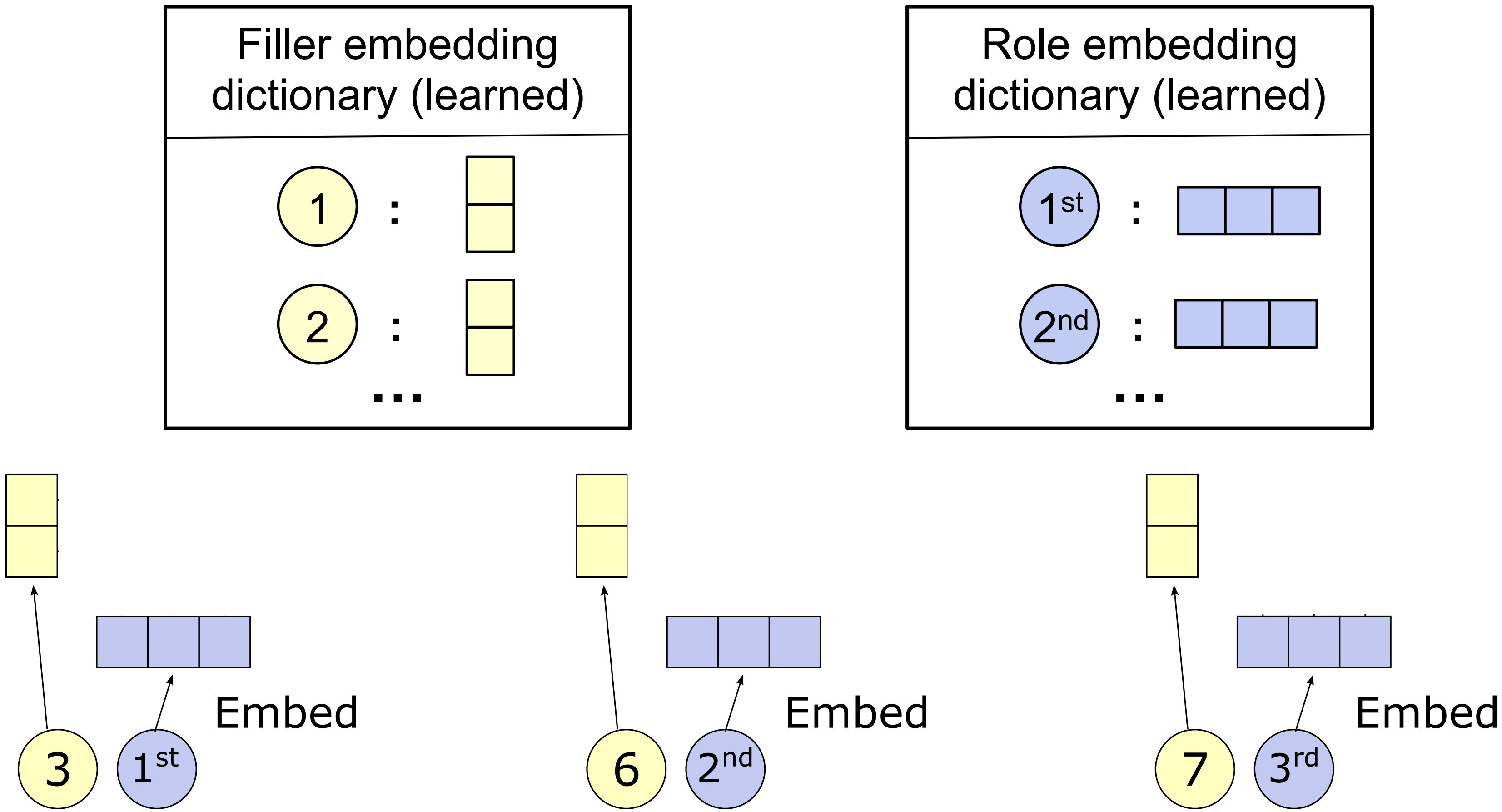
Tensor Product Representations



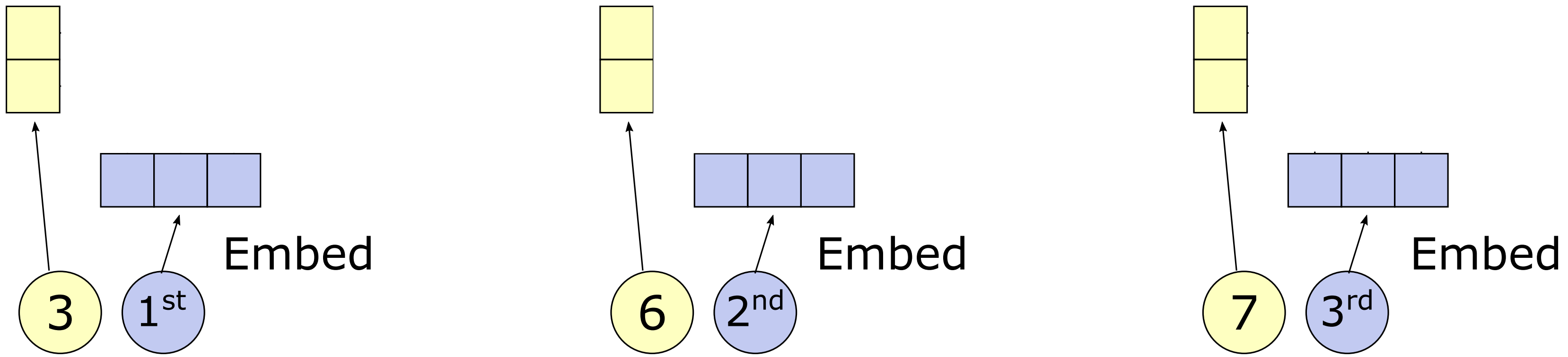
Tensor Product Representations



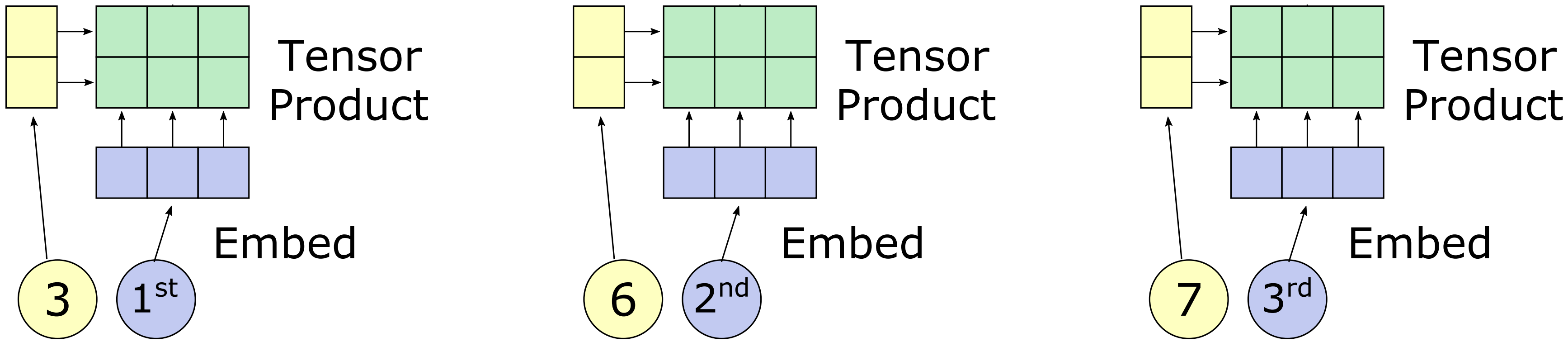
Tensor Product Representations



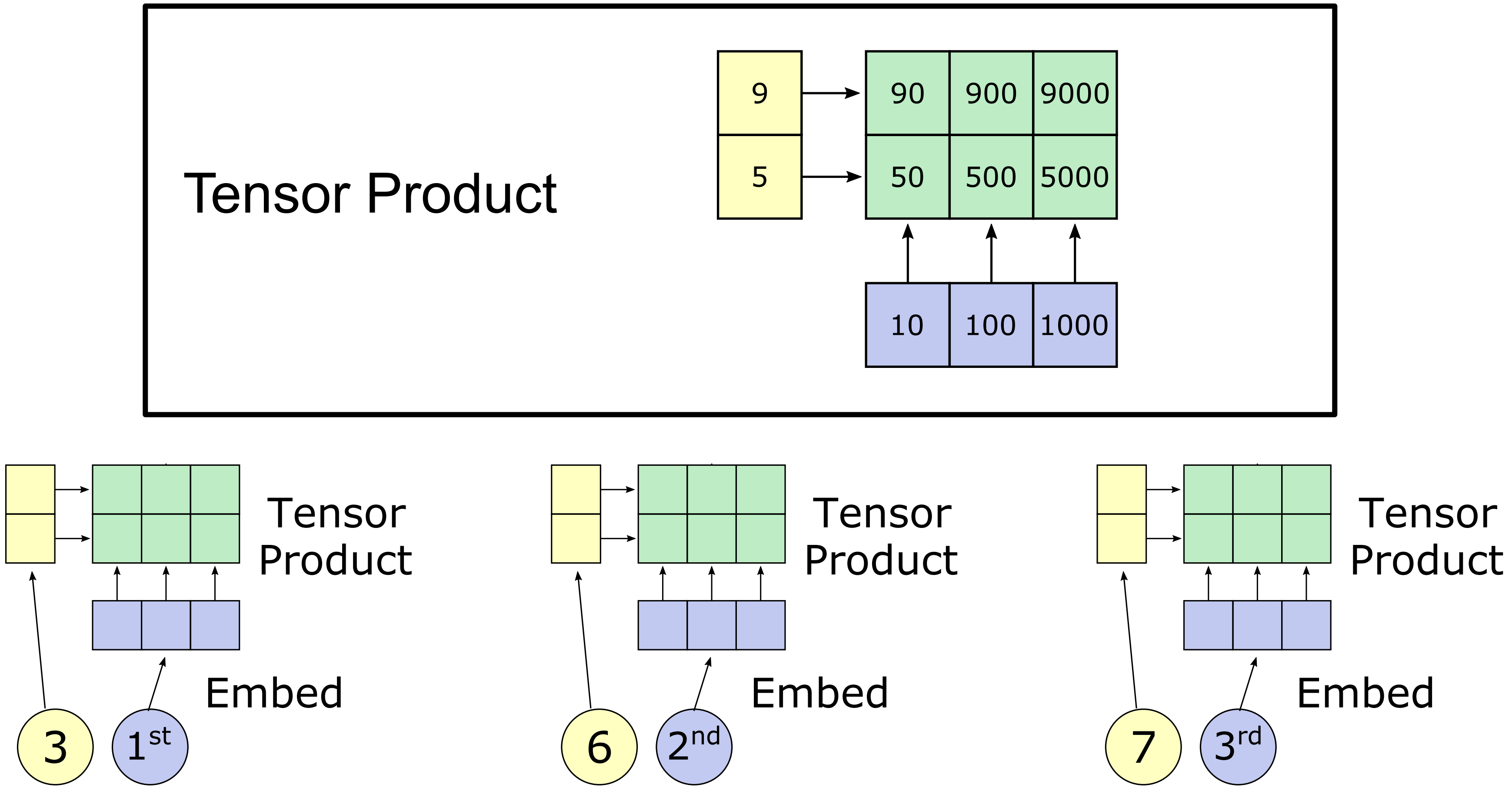
Tensor Product Representations



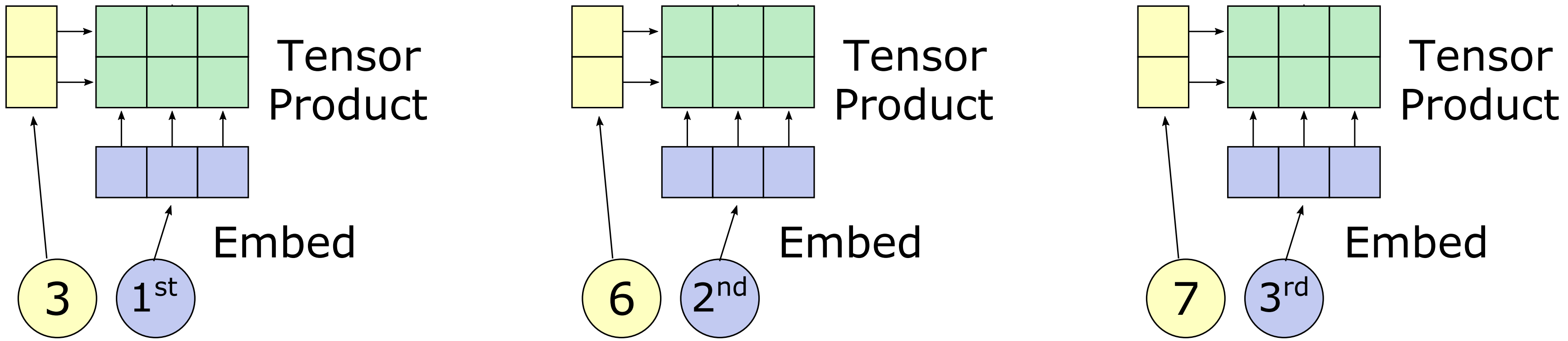
Tensor Product Representations



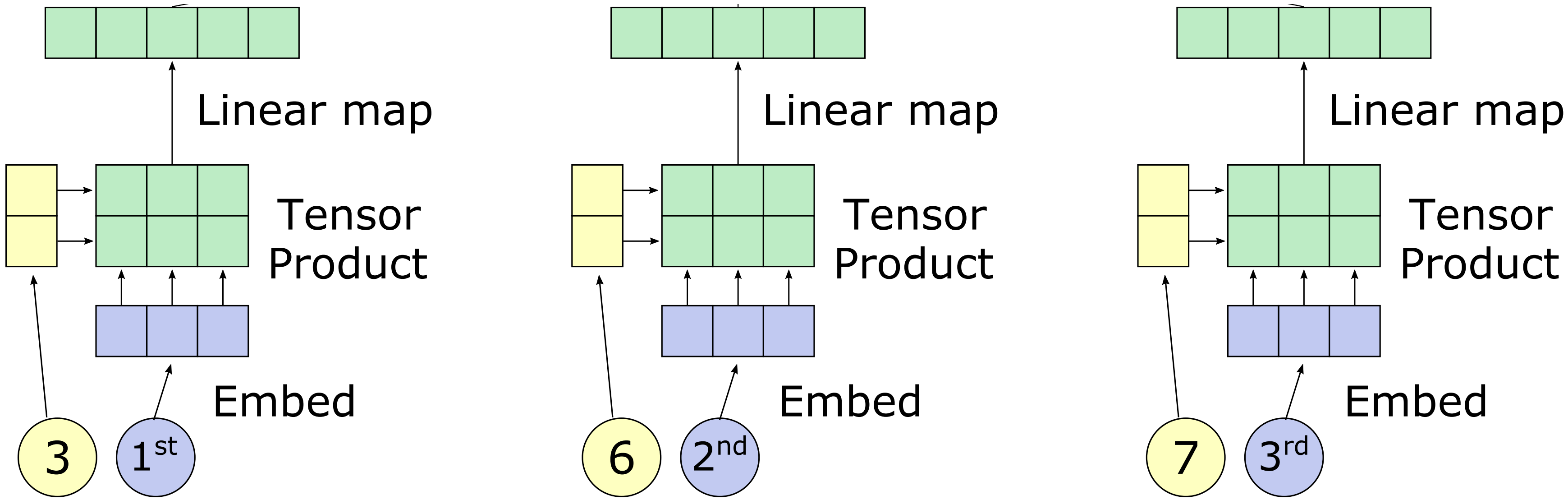
Tensor Product Representations



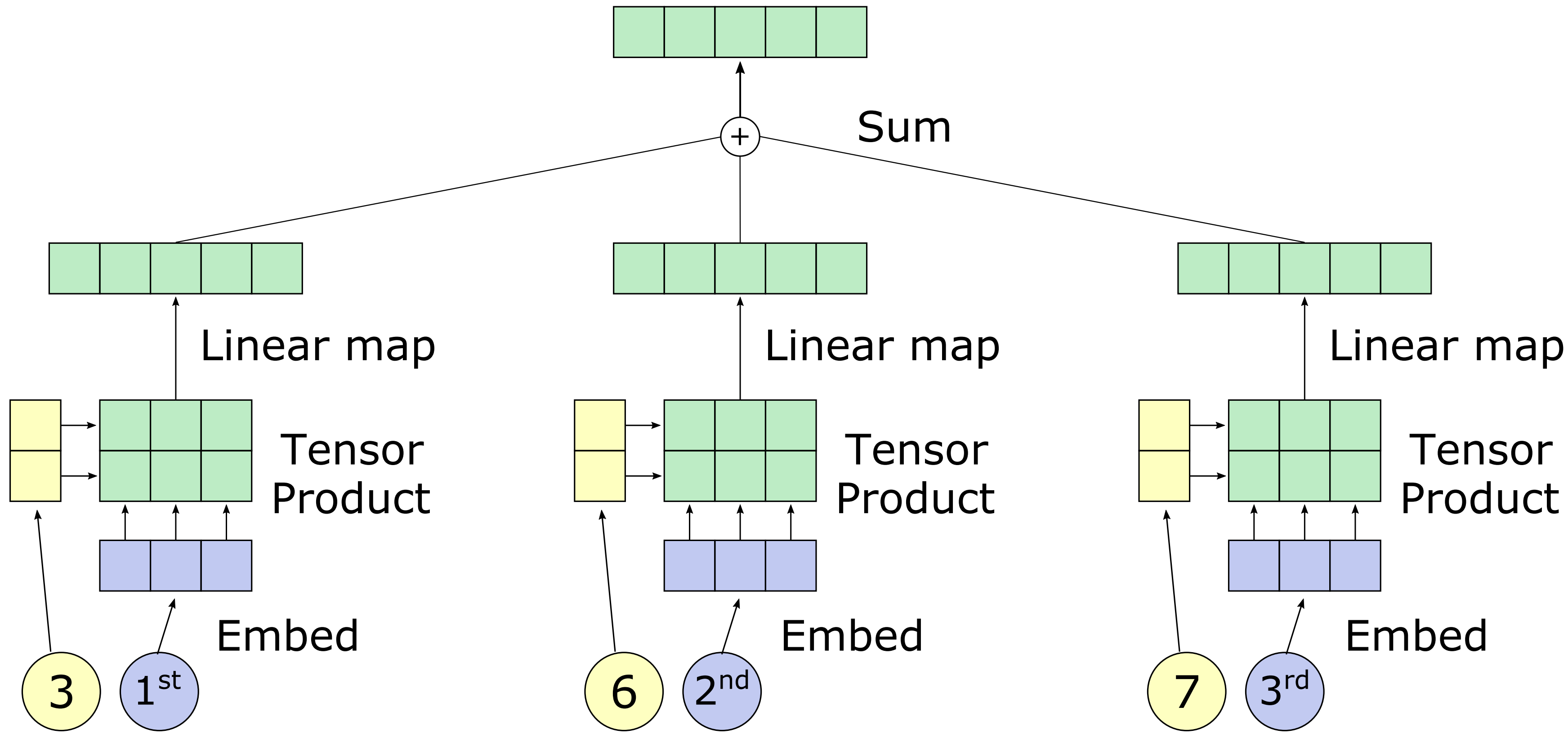
Tensor Product Representations



Tensor Product Representations

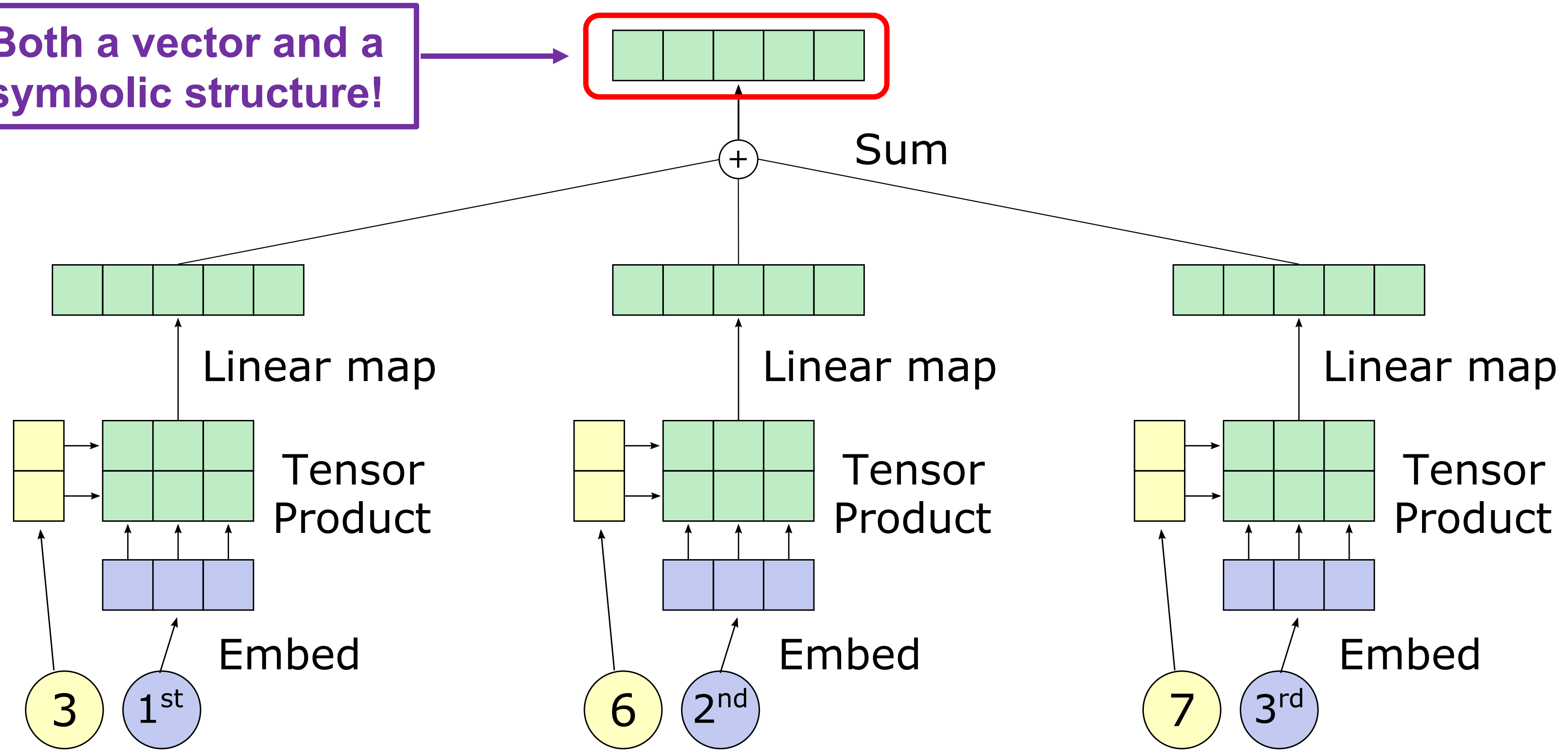


Tensor Product Representations



Tensor Product Representations

Both a vector and a symbolic structure!



Tensor Product Representations

- Hypothesis: Neural network representations are implicitly structured as Tensor Product Representations
  - Even without being designed to have this structure!

Symbolic structures

Classical painting

Vectors

Jackson Pollock

Symbolic structures

Classical painting

Vectors

Jackson Pollock

Tensor Product Representations

Pointillism

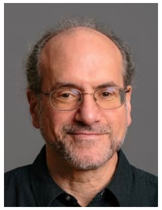
Tensor Product Representations

- Hypothesis: Neural network representations are implicitly structured as Tensor Product Representations
  - Even without being designed to have this structure!
- To test the hypothesis: train Tensor Product Representations to approximate the neural network representations

(McCoy, Linzen, Dunbar, Smolensky 2019: ICLR)

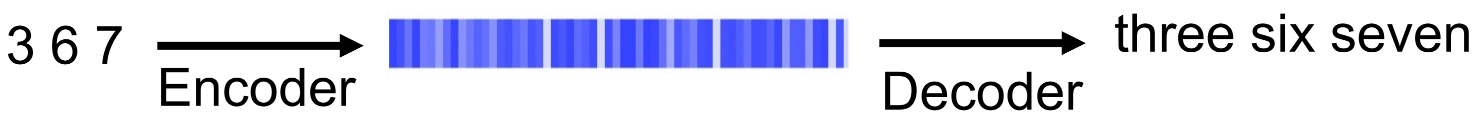
(McCoy 2022: Dissertation)



Testing the hypothesis

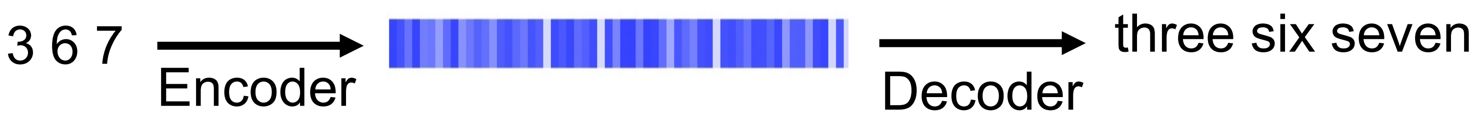
- Model being analyzed:



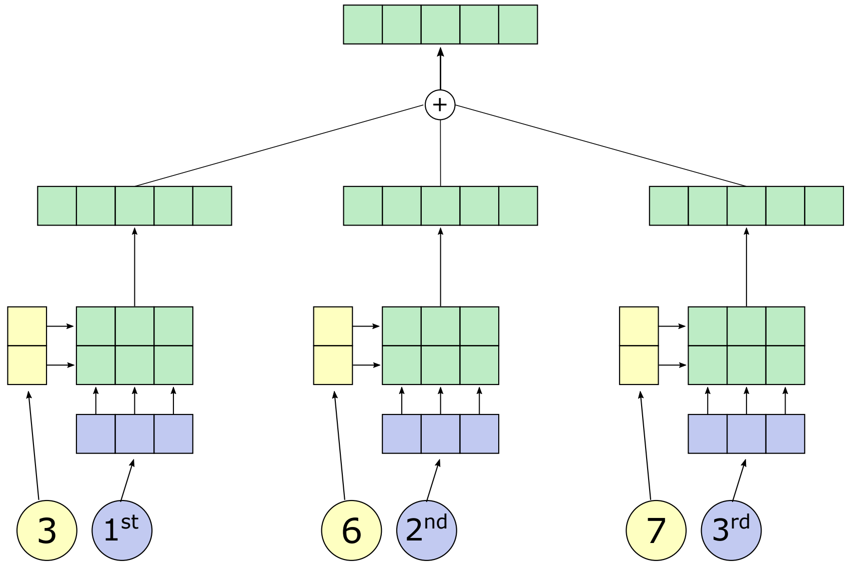
- Train a Tensor Product Representation to approximate encodings:

Testing the hypothesis

- Model being analyzed:

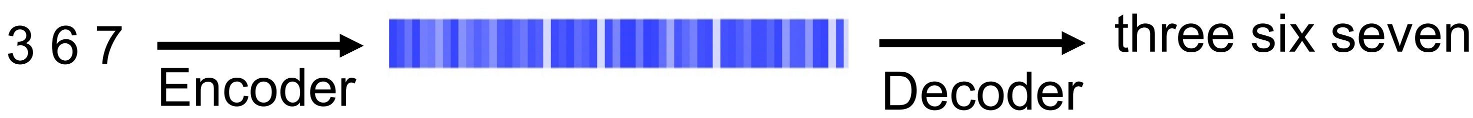


- Train a Tensor Product Representation to approximate encodings:

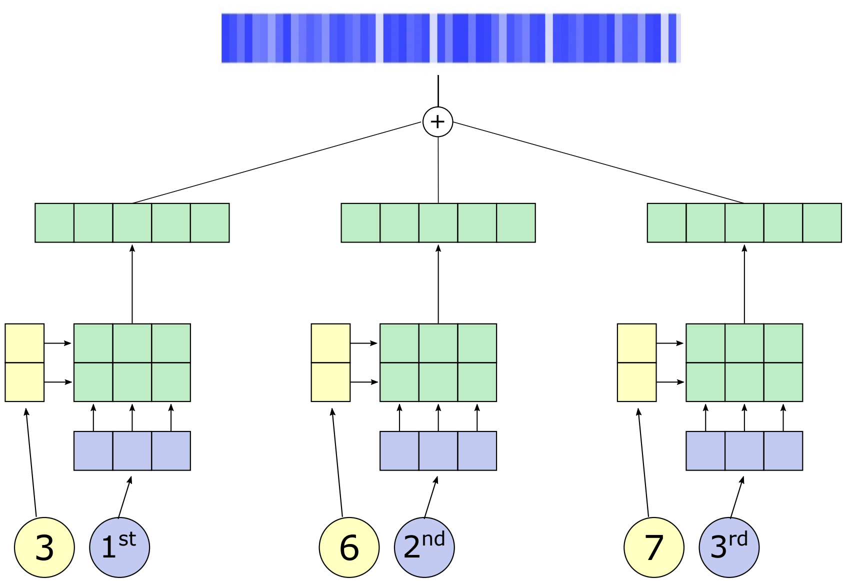


Testing the hypothesis

- Model being analyzed:

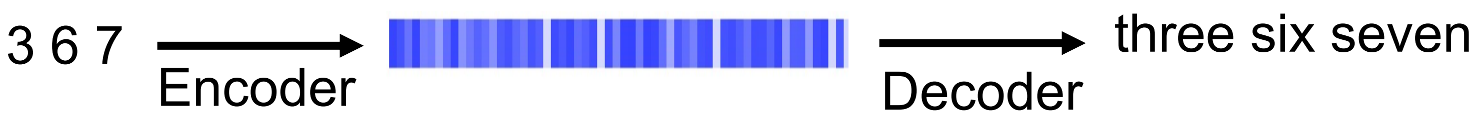


- Train a Tensor Product Representation to approximate encodings:

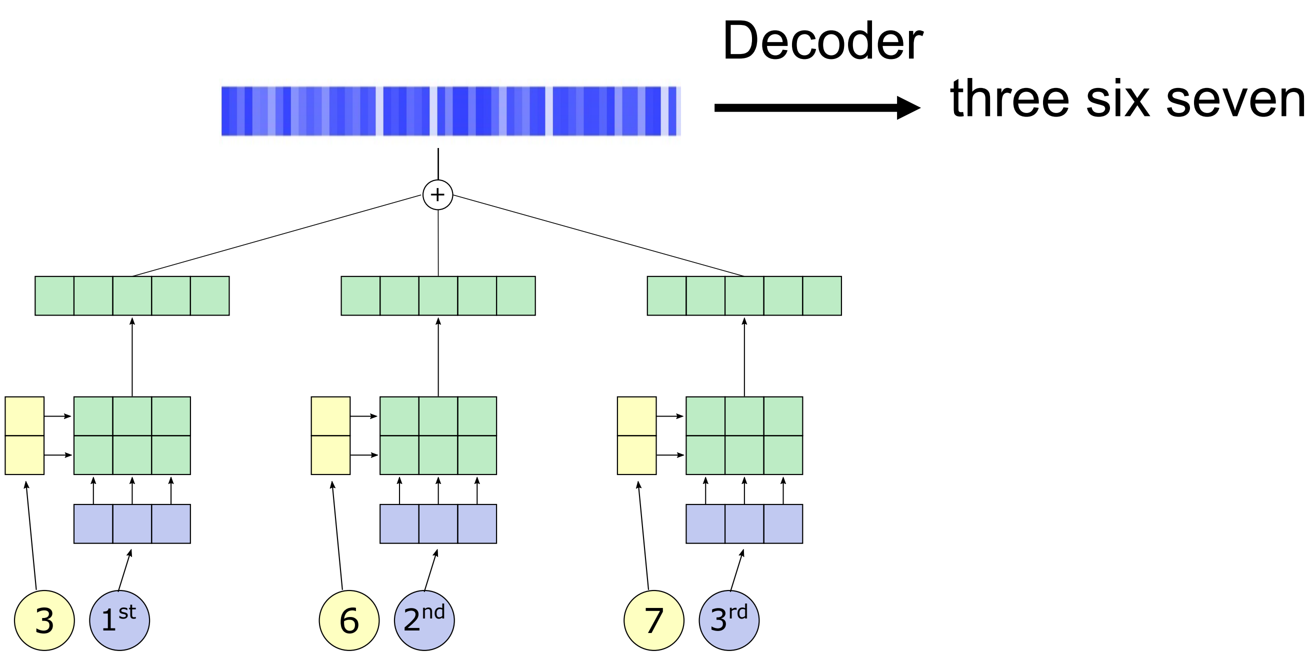


Testing the hypothesis

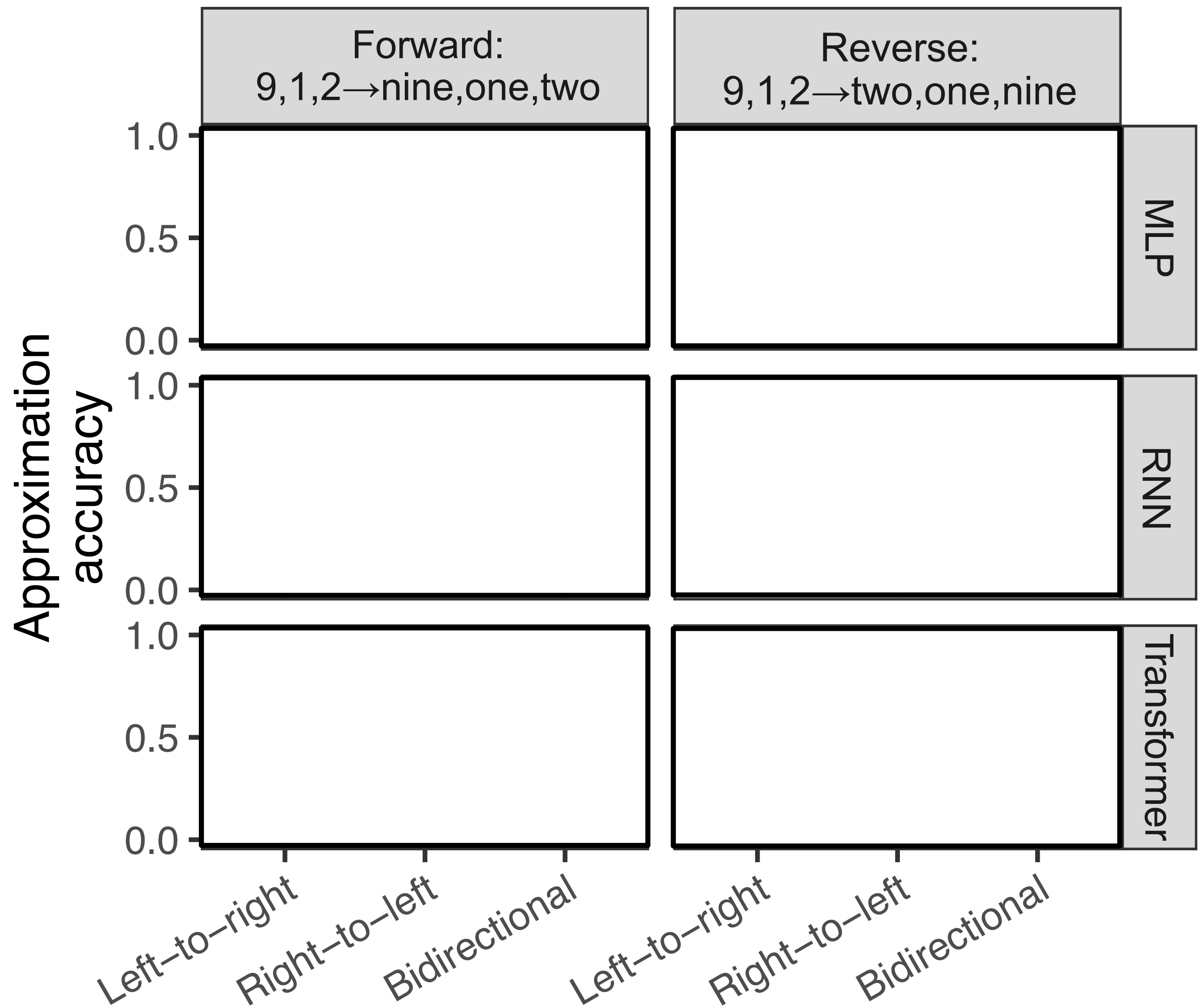
- Model being analyzed:



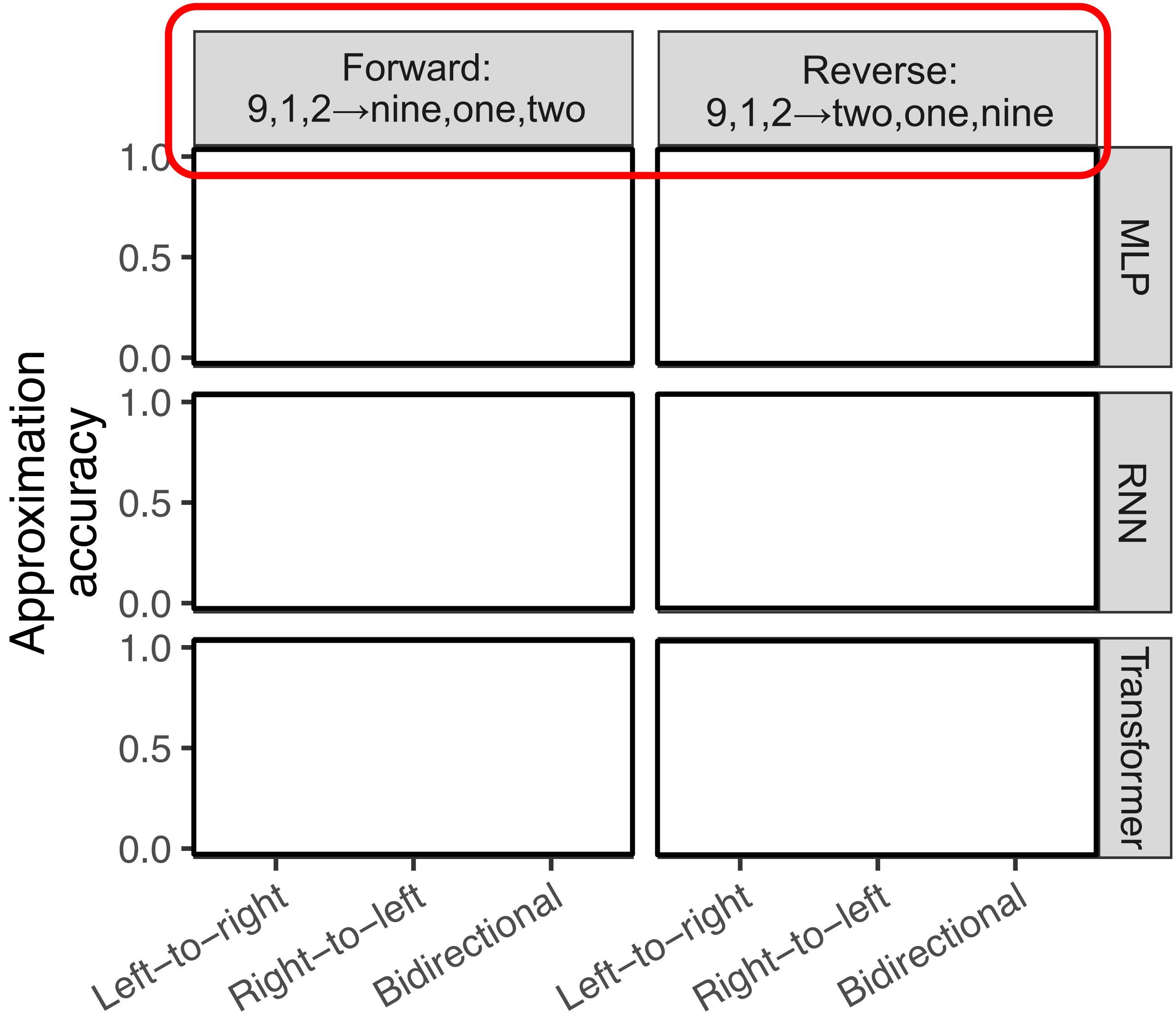
- Train a Tensor Product Representation to approximate encodings:



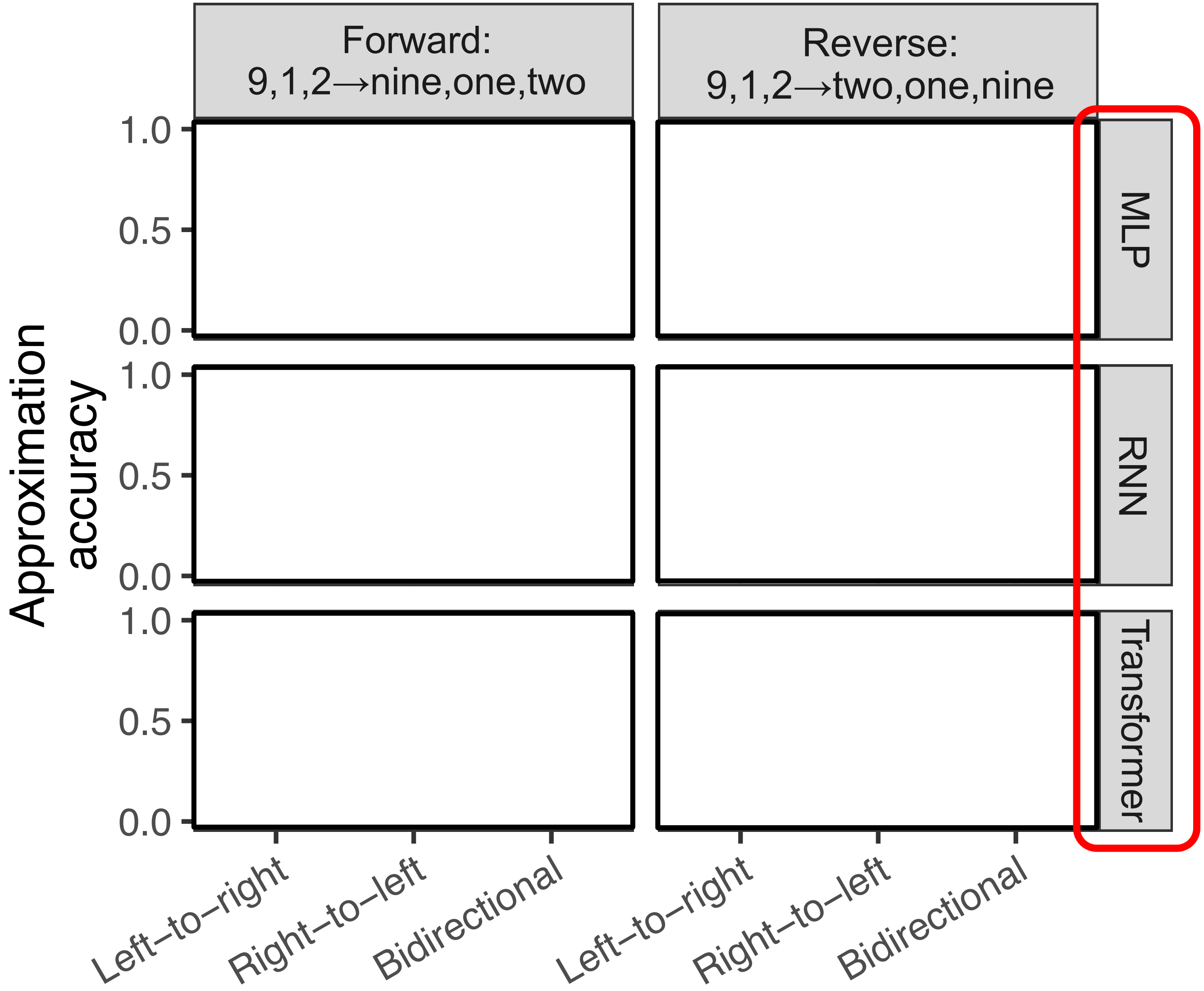
Results


Forward:
9,1,2→nine,one,two
Reverse:
9,1,2→two,one,nine
MLP
RNN
Transformer
Approximation accuracy
1.0
0.5
0.0
Left–to–right
Right–to–left
Bidirectional

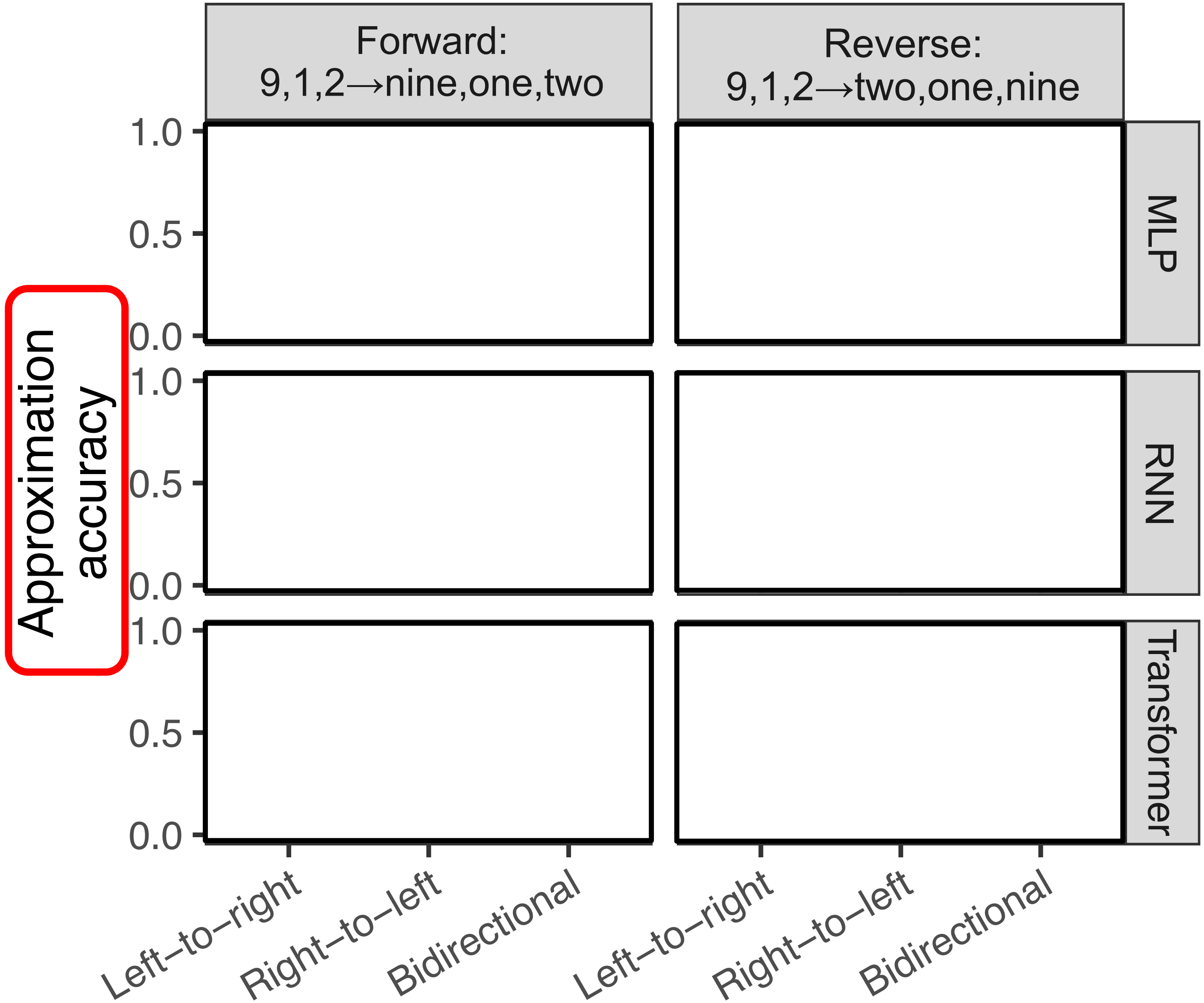
Results


Forward:
9,1,2→nine,one,two
Reverse:
9,1,2→two,one,nine
Approximation accuracy
1.0
0.5
0.0
MLP
RNN
Transformer
Left–to–right
Right–to–left
Bidirectional

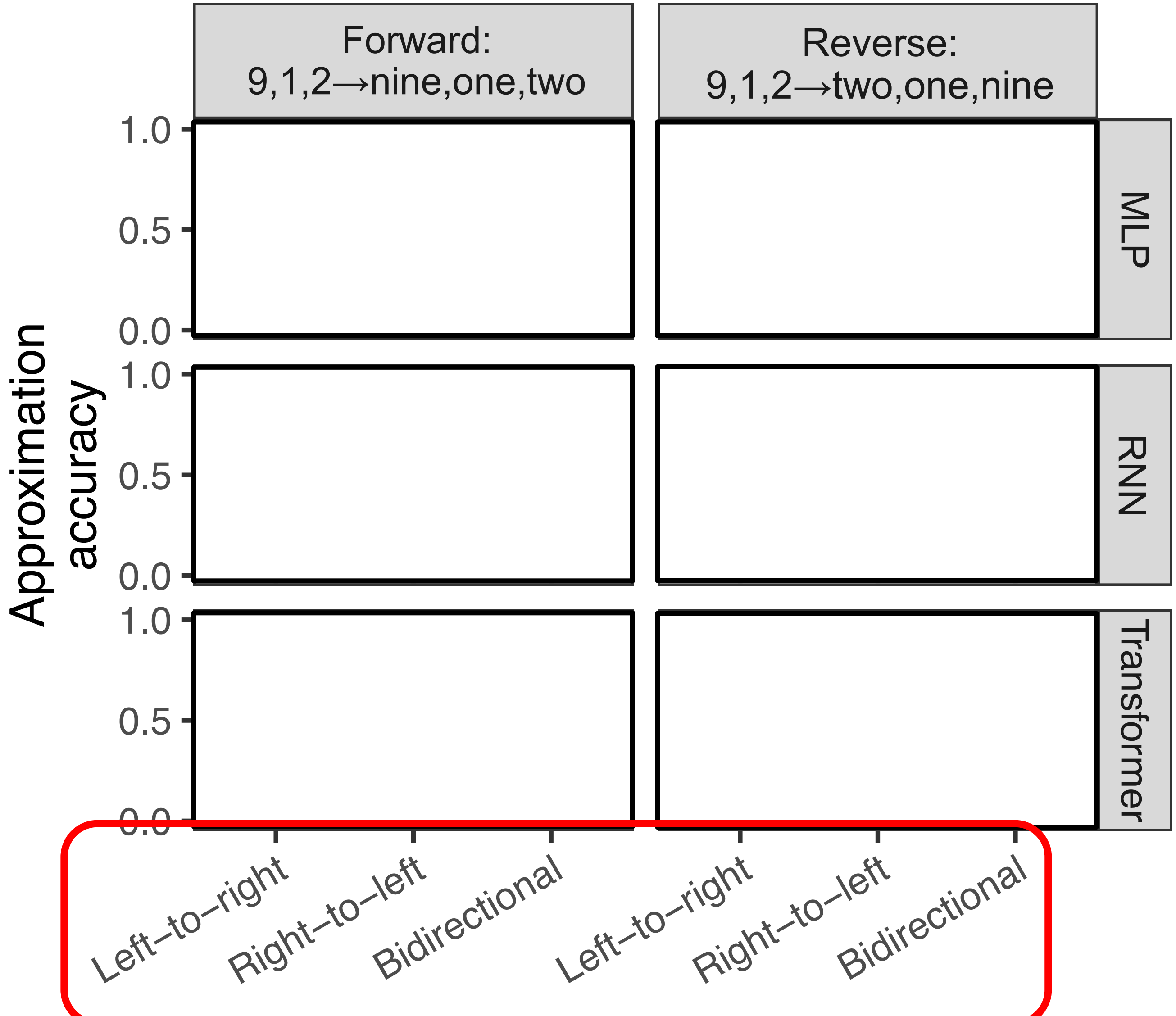
Results


Forward:
9,1,2→nine,one,two
Reverse:
9,1,2→two,one,nine
Approximation accuracy
1.0
0.5
0.0
MLP
RNN
Transformer
Left–to–right
Right–to–left
Bidirectional

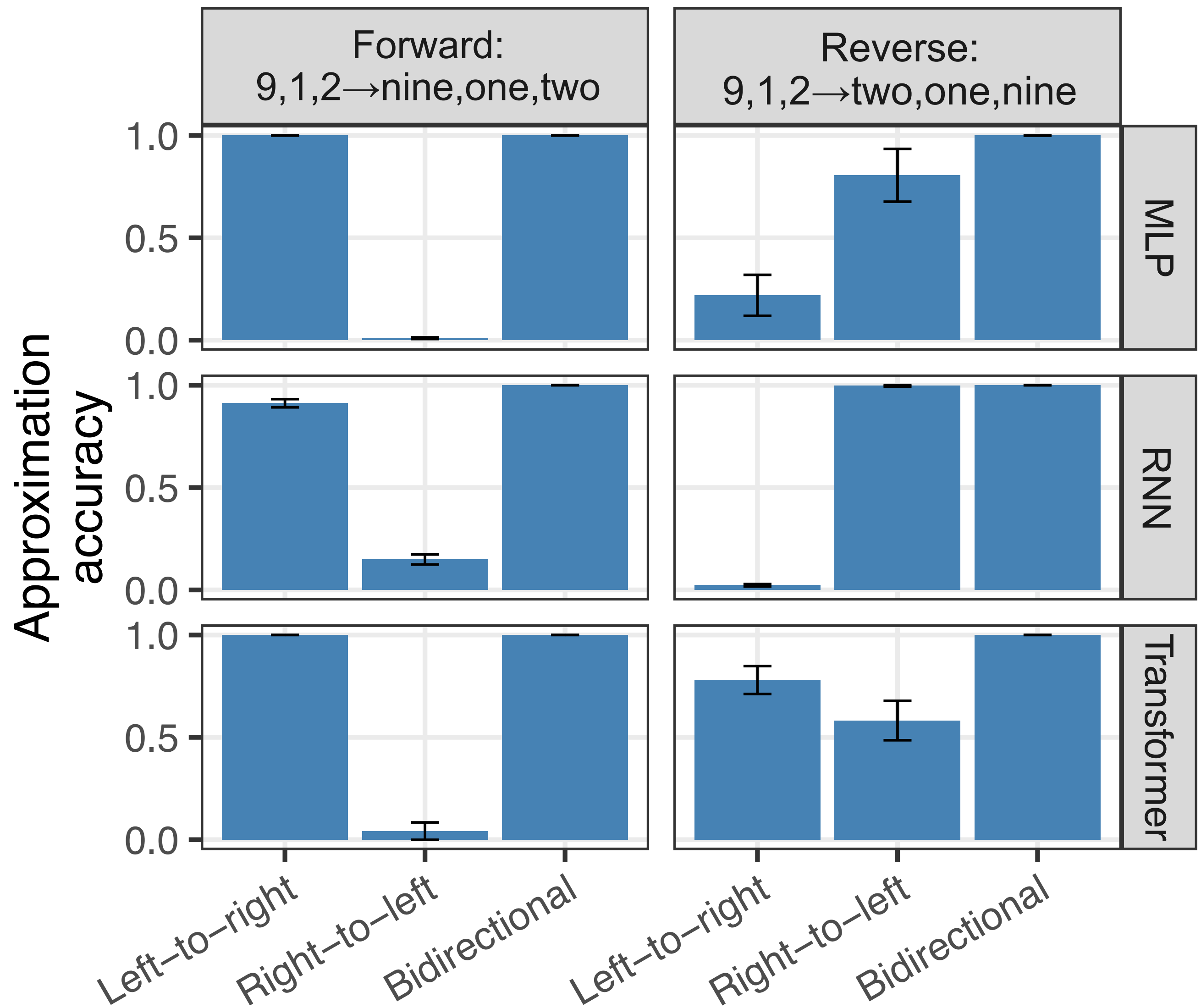
Results


Forward:
9,1,2→nine,one,two
Reverse:
9,1,2→two,one,nine
MLP
RNN
Transformer
Approximation accuracy
1.0
0.5
0.0
Left–to–right
Right–to–left
Bidirectional

Results


Forward:
9,1,2→nine,one,two
Reverse:
9,1,2→two,one,nine
Approximation accuracy
1.0
0.5
0.0
MLP
RNN
Transformer
Left–to–right
Right–to–left
Bidirectional
Left–to–right
Right–to–left
Bidirectional

Results



Very strong approximation: Shows that the representations of these neural networks have implicit symbolic structure!

Results with natural language

- Have also applied this approach to neural networks trained on natural language
  - And have gotten promising results

Q: What type of system should we use to represent language?

A: Vectors that <u>act like symbols</u>

Representations

Learning

Q: What type of system should we use to represent language?

A: Vectors that <u>act like symbols</u>

<u>Compositional representations</u>

Tensor Product Representations

<u>Learning</u>

Q: What type of system should we use to represent language?

A: Vectors that <u>act like symbols</u>

<u>Compositional representations</u>

Tensor Product Representations

<u>Learning</u>

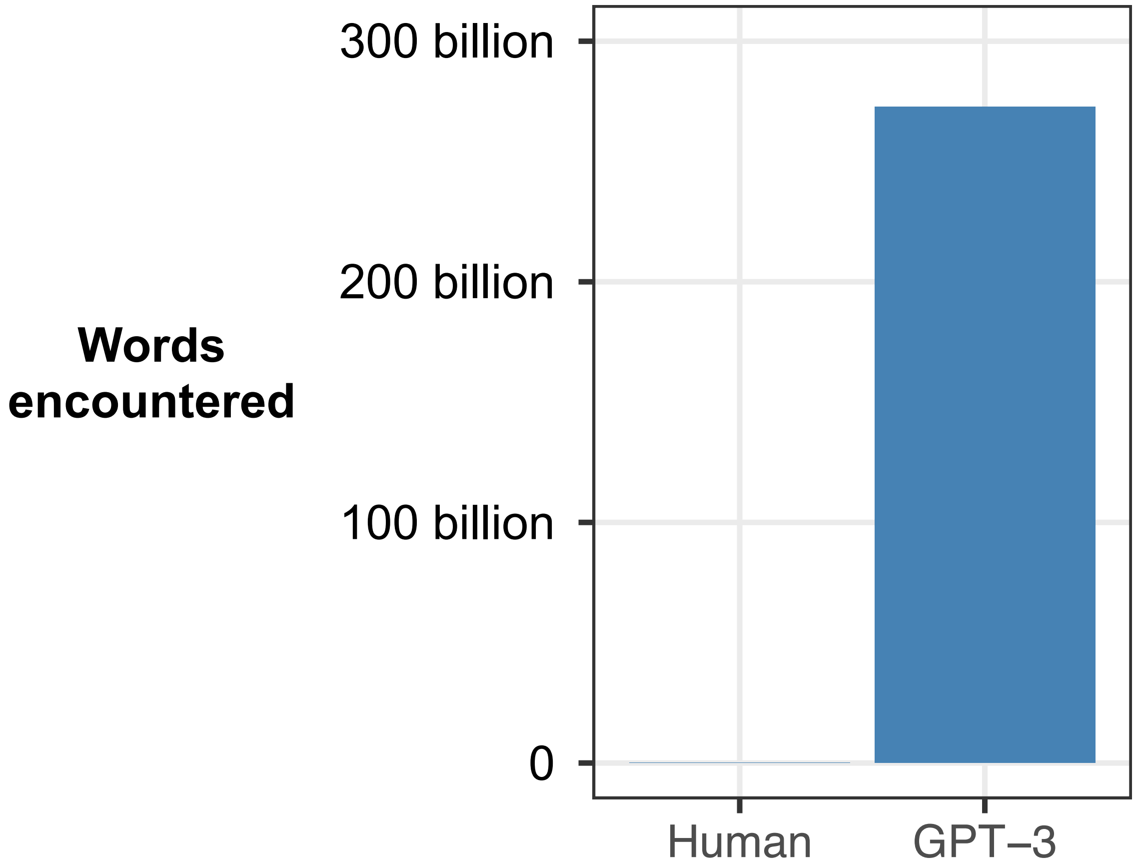
Modeling language learning

- Representations are one area where vectors and symbols seem mismatched
- Learning is another such area

Modeling language learning

- Neural networks have some attractive properties as models of learning:
  - Trained on naturalistic corpora
  - By the end of training: They capture many aspects of linguistic structure

Problem: Data quantity



Another candidate: Probabilistic models

- Represent hypotheses using symbolic grammars
- E.g., a context-free grammar

$S \rightarrow NP\ VP$
$NP \rightarrow Det\ N$
$VP \rightarrow V\ NP$
$Det \rightarrow$ the
$Det \rightarrow$ a
…

Probabilistic models

- Goal: Given a corpus, find the grammar that best describes the corpus

Probabilistic models

- Goal: Given a corpus, find the grammar that best describes the corpus
- Need some way to decide what it means to “best describe”
  - Answer: Probability

Probabilistic models

- Goal: Given a corpus, find the grammar that best describes the corpus
- Need some way to decide what it means to “best describe”
  - Answer: Probability

Prominent type of probabilistic model: Bayesian model

Probabilistic models

- Can learn effectively from small amounts of data
  - Why? Using structured grammars as hypotheses guides the search

Probabilistic models

- Can learn effectively from small amounts of data
  - Why? Using structured grammars as hypotheses guides the search
- But they are typically intractable in naturalistic settings

Probabilistic models

- Can learn effectively from small amounts of data
  - Why? Using structured grammars as hypotheses guides the search
- But they are typically intractable in naturalistic settings
  - Hard to find a grammar that captures all the complexities of the data

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	
Strong inductive biases	

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	
Strong **inductive biases**	

The factors that guide generalization

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	
Strong inductive biases	
Effective generalization from limited data	

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	
Strong inductive biases	
Effective generalization from limited data	
Struggle to tractably handle natural data	

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	Representational flexibility: Vectors
Strong inductive biases	
Effective generalization from limited data	
Struggle to tractably handle natural data	

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	Representational flexibility: Vectors
Strong inductive biases	
Effective generalization from limited data	
Struggle to tractably handle natural data	Can handle complex, natural data

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	Representational flexibility: Vectors
Strong inductive biases	Weak inductive biases
Effective generalization from limited data	
Struggle to tractably handle natural data	Can handle complex, natural data

Modeling learning

Probabilistic models (e.g., Bayesian)	Neural networks
Strong representational commitments: Symbols	Representational flexibility: Vectors
Strong inductive biases	Weak inductive biases
Effective generalization from limited data	Require lots of data
Struggle to tractably handle natural data	Can handle complex, natural data

Proposal: Inductive bias distillation

- Distill Bayesian inductive biases into a neural network
- Neural network → flexible
- Strong inductive bias → rapid learning

Sounds nice…but how can
we actually do it?

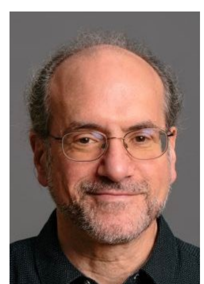
Modeling rapid language learning by distilling Bayesian priors into artificial neural networks

R. Thomas McCoy[1*] and Thomas L. Griffiths[2,1]


Universal linguistic inductive biases via meta-learning

R. Thomas McCoy,[1] Erin Grant,[2] Paul Smolensky,[3,1] Thomas L. Griffiths,[4] and Tal Linzen[1]





Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

$h = \text{plus(D)}$

$h = \text{concat(}$
or(A, C),
plus(A),
Σ,
or(ε, B))

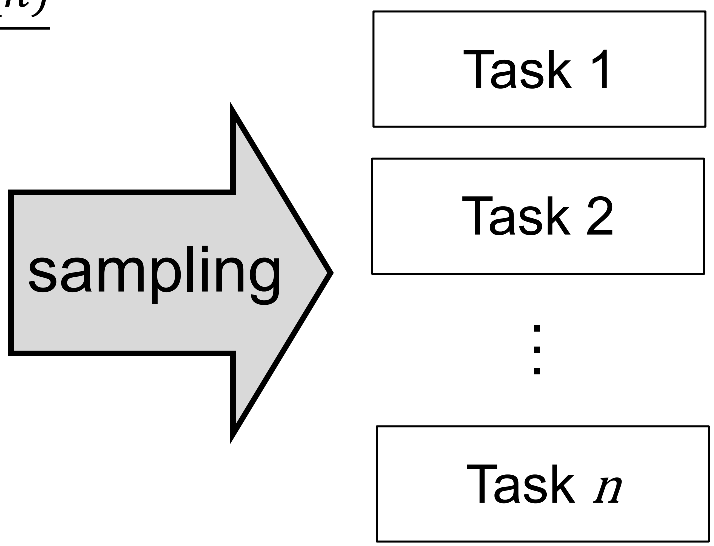
Probabilistic
model

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

$h = \mathrm{plus(D)}$

$h = \mathrm{concat(}$
$\quad \mathrm{or(A, C),}$
$\quad \mathrm{plus(A),}$
$\quad \Sigma,$
$\quad \mathrm{or(\varepsilon, B))}$



Probabilistic model

Training data

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

h = plus(D)

h = concat(
or(A, C),
plus(A),
Σ,
or(ε, B))

sampling

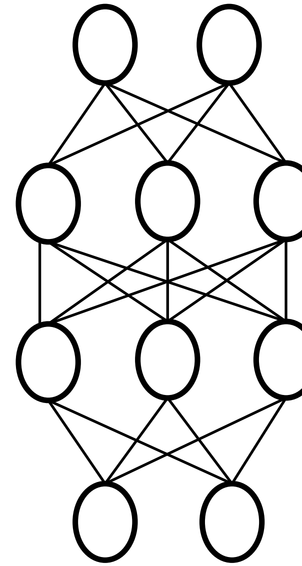
Task 1

Task 2

⋮

Task n

meta learning



Probabilistic model

Training data

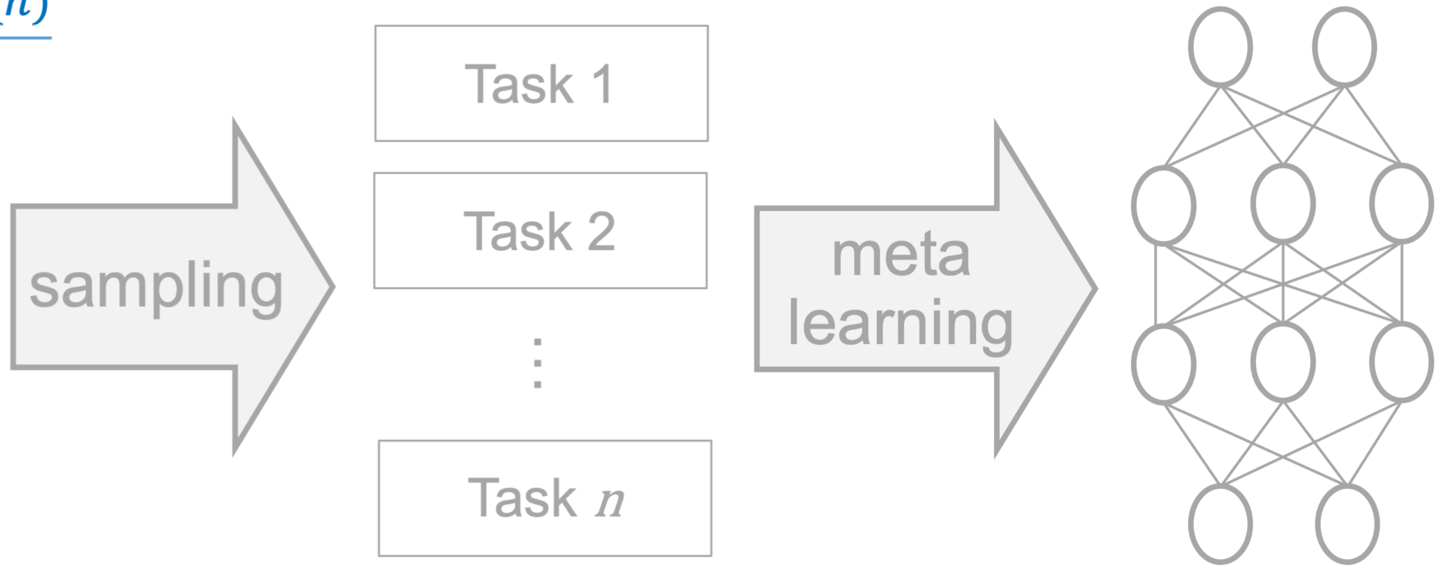
Neural network

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

$h =$ plus(D)

$h =$ concat(
or(A, C),
plus(A),
Σ,
or(ε, B))



Probabilistic model

Training data

Neural network

One model for the learning of language

Yuan Yang[a] **and Steven T. Piantadosi**[b,1]

[a]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332; and [b]Department of Psychology, Hele
of California, Berkeley, CA 94720

Formal languages

- plus(concatenate(A, B))
  - AB
  - ABAB
  - ABABAB
  - …

Formal languages

- Inspired by the structure of natural language

Formal languages

- Inspired by the structure of natural language
- Consider **plus(concatenate(A, B))**: AB, ABAB, ABABAB, …

Formal languages

- Inspired by the structure of natural language
- Consider **plus(concatenate(A, B))**: AB, ABAB, ABABAB, …
  - A → *preposition*
  - B → <u>noun phrase</u>

Formal languages

- Inspired by the structure of natural language
- Consider **plus(concatenate(A, B))**: AB, ABAB, ABABAB, …
  - A → *preposition*
  - B → noun phrase
  - *on* the table *by* the door *in* the kitchen

Probabilistic model

- Probabilistically combine primitives into formal languages

B

plus

or

A

plus(or(A,B))

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

$h = $ plus(D)

$h = $ concat(
or(A, C),
plus(A),
Σ,
or(ε, B))

sampling

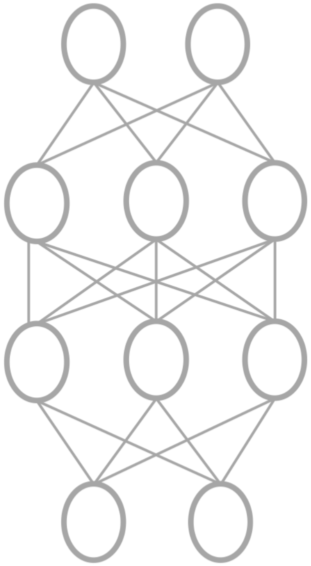
Task 1

Task 2

⋮

Task n

meta learning



Probabilistic model

Training data

Neural network

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

$h =$ plus(D)

$h =$ concat(
or(A, C),
plus(A),
Σ,
or(ε, B))

sampling

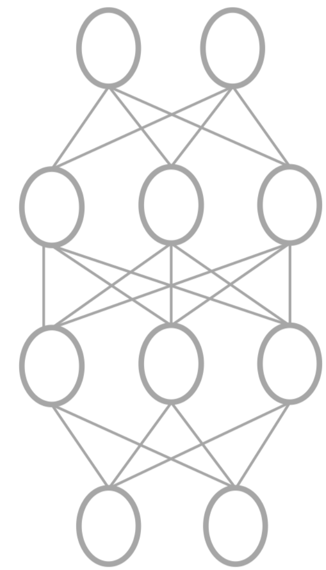
Task 1

Task 2

⋮

Task n

meta learning



Probabilistic model

Training data

Neural network

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

$h = \text{plus(D)}$

$h = \text{concat(}$
$\quad\text{or(A, C),}$
$\quad\text{plus(A),}$
$\quad\Sigma,$
$\quad\text{or(}\varepsilon\text{, B))}$

sampling

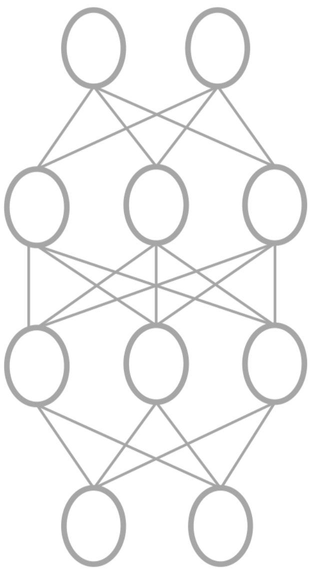
Language 1

Language 2

⋮

Language n

meta learning



Probabilistic model

Training data

Neural network

Sampling formal languages

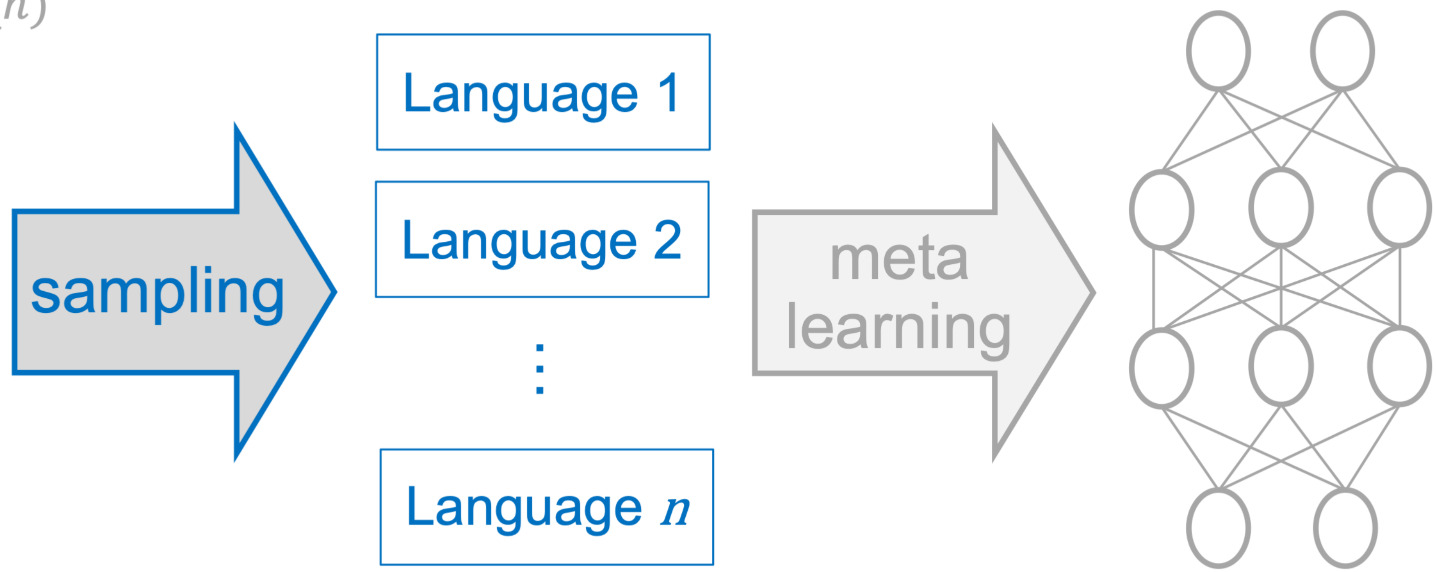
- plus(A)
- D
- concat(A, B, A)
- concat(plus(or(A, concat(C, A, C)), or(plus(D), A))
- or(A, A)
- B
- …

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

$h =$ plus(D)

$h =$ concat(
or(A, C),
plus(A),
Σ,
or(ε, B))



Probabilistic model

Training data

Neural network

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

$h =$ plus(D)

$h =$ concat(
or(A, C),
plus(A),
Σ,
or(ε, B))

sampling

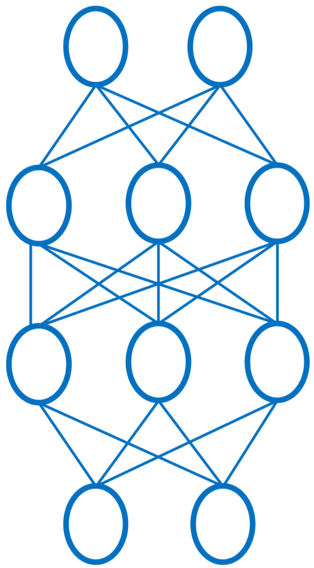
Language 1

Language 2

⋮

Language n

meta learning



Probabilistic model

Training data

Neural network

Meta-learning

Meta-learning

- Learning to learn

Meta-learning

- Approach:

Meta-learning

- Approach:
  - Show the model many languages
    - Giving it linguistic inductive biases (which types of languages are likely/unlikely?)

Meta-learning

- Approach:
  - Show the model many languages
    - Giving it linguistic inductive biases (which types of languages are likely/unlikely?)
  - By controlling the languages, we control the model's inductive biases

Meta-learning

- Approach:
  - Show the model many languages
    - Giving it linguistic inductive biases (which types of languages are likely/unlikely?)
  - By controlling the languages, we control the model's inductive biases
- Variant that we use: MAML (Finn et al. 2017)

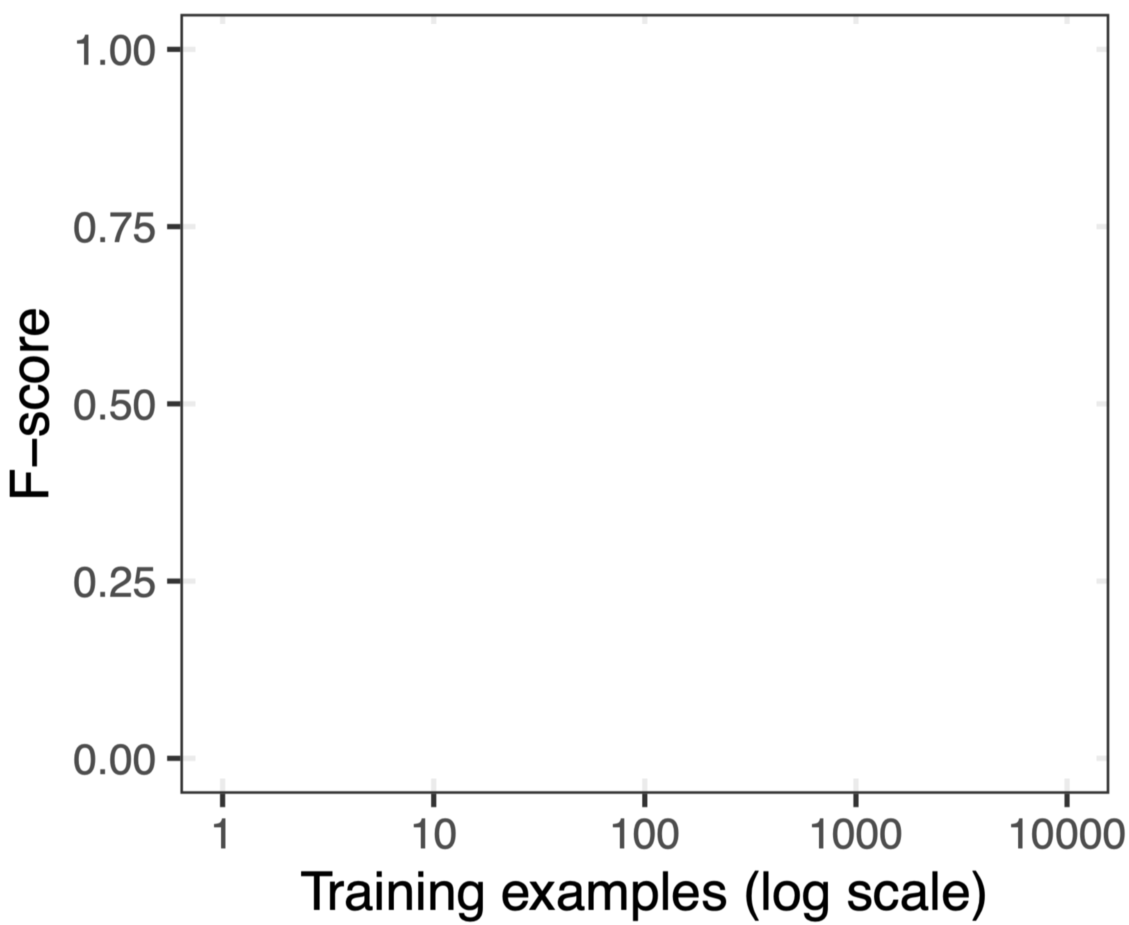
Meta-learning

- Result: A **prior-trained** neural network
  - Trained to have a particular prior

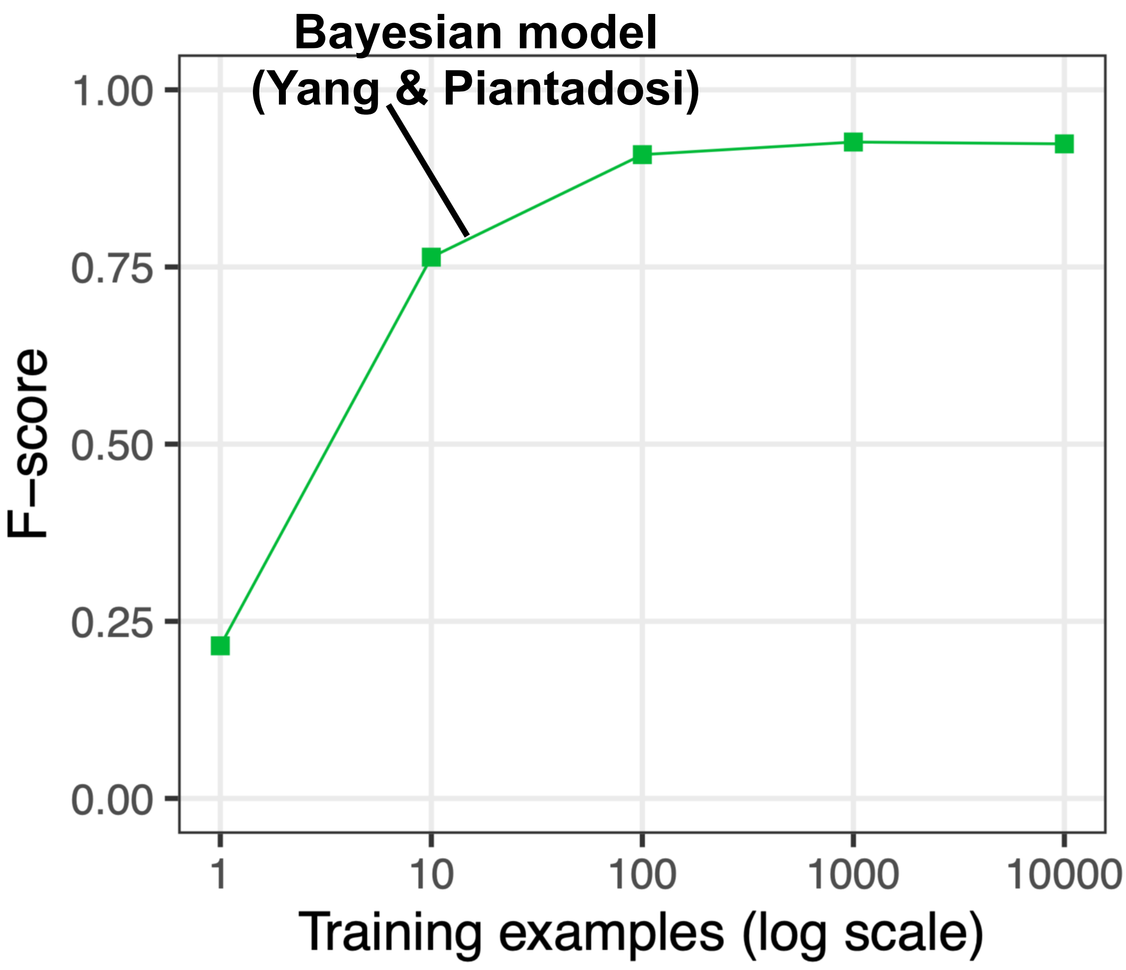
Meta-learning

- Result: A **prior-trained** neural network
  - Trained to have a particular prior
- Different from **pre-trained**:
  - Trained to learn aspects of the intended task – learning, not meta-learning
  - The prior is an indirect byproduct, not a direct target

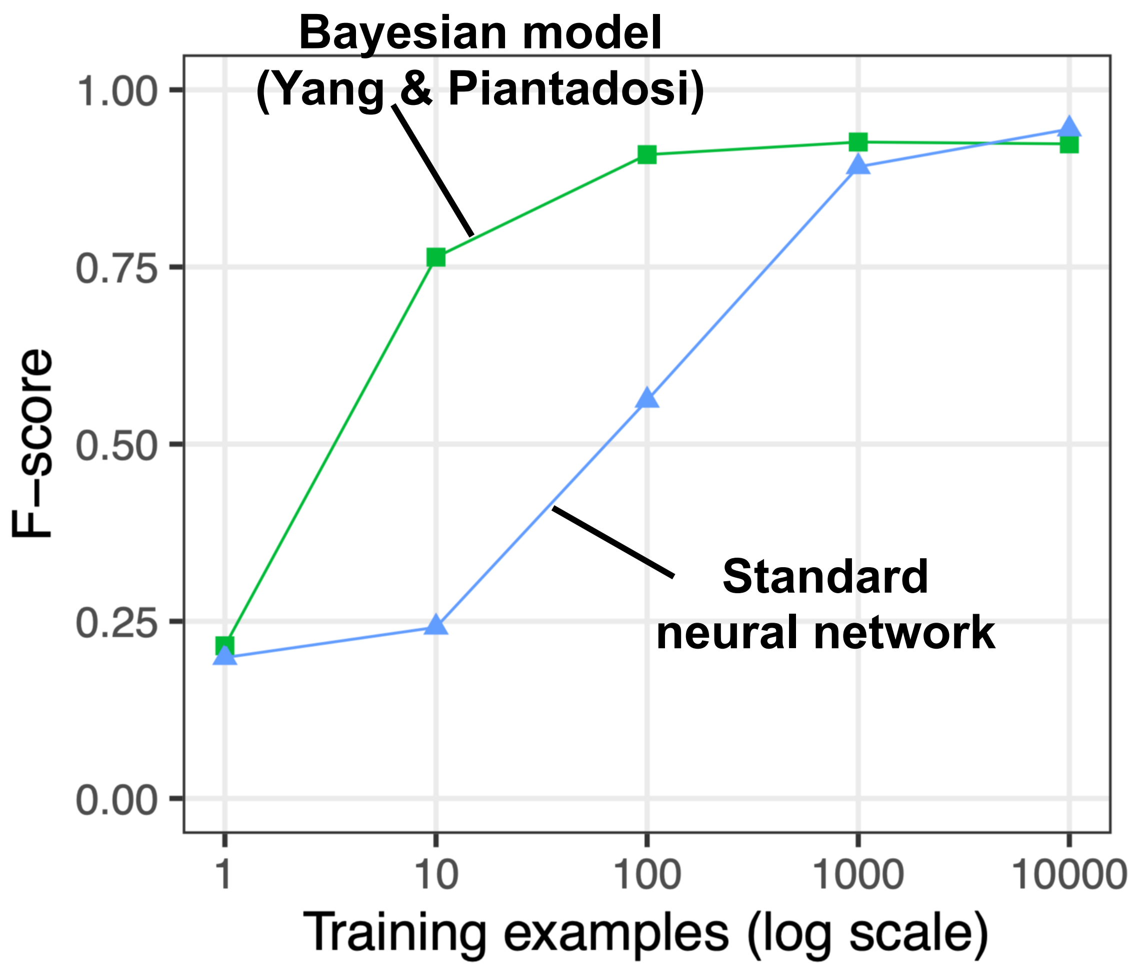
Results: Learning formal languages



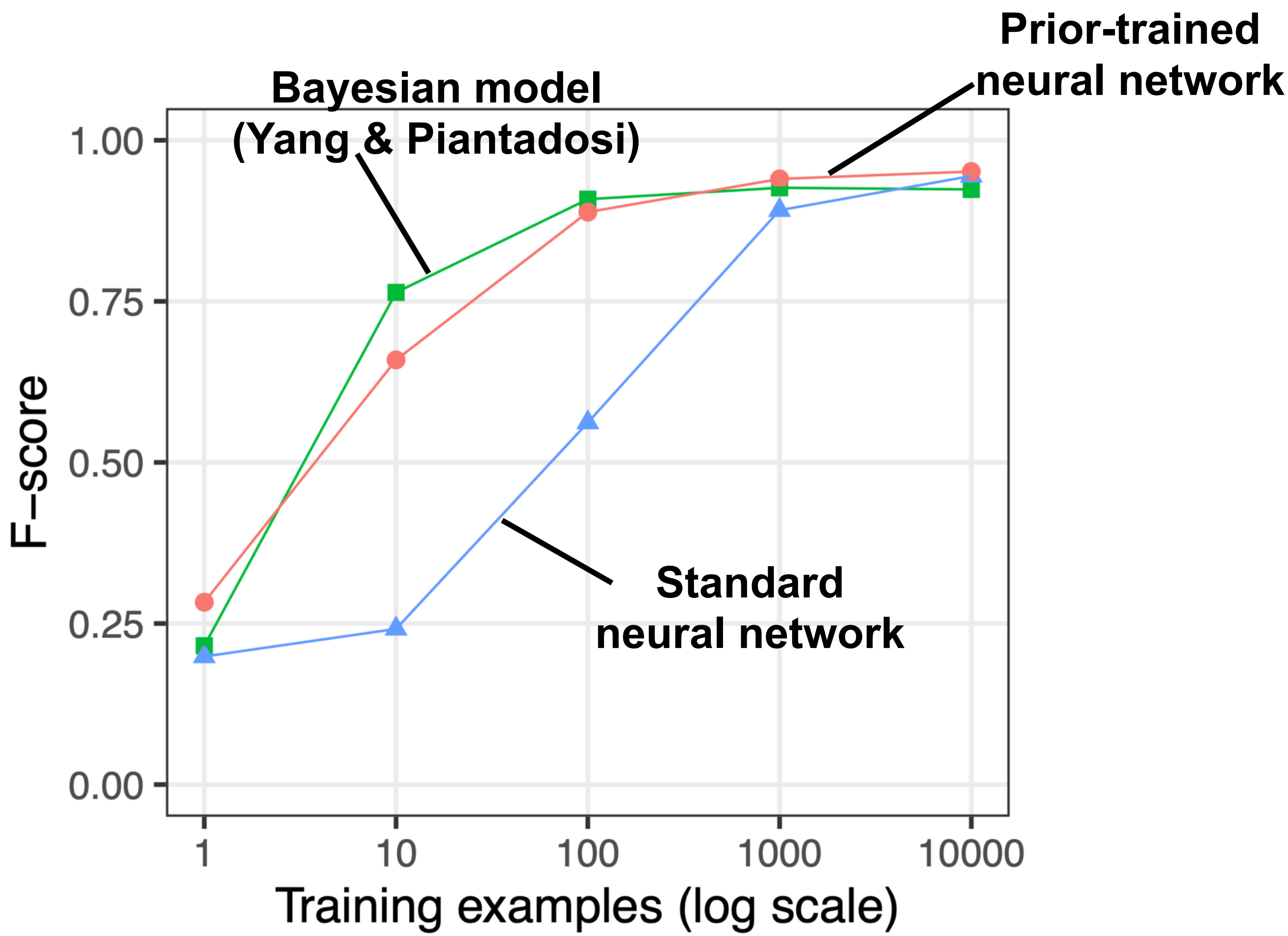
Results: Learning formal languages



Results: Learning formal languages



Results: Learning formal languages



Time

- Bayesian model: 1 minute to 7 days
  - Not feasible to train on naturalistic data

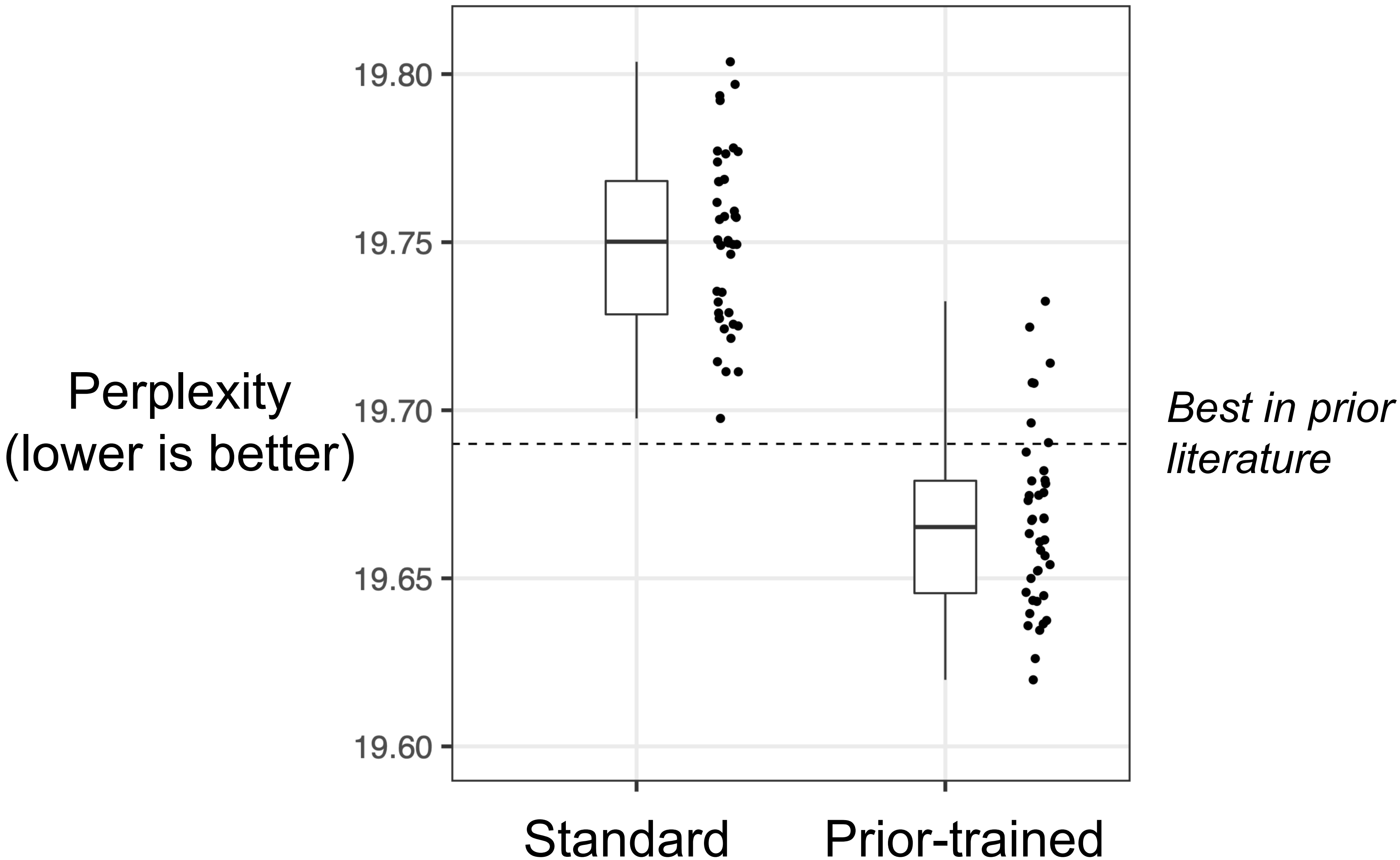
Time

- Bayesian model: 1 minute to 7 days
  - Not feasible to train on naturalistic data
- Prior-trained neural network: 10 milliseconds to 3 minutes
  - Can train on naturalistic data!

Training on English

- Child-directed speech
- 8 million words

Training on English: Results



Recursion

- The prior included the Kleene **plus** primitive
  - plus(B) = {B, BB, BBB, …}

Recursion

- The prior included the Kleene **plus** primitive
  - plus(B) = {B, BB, BBB, …}
- Does the prior-trained model handle recursion well?

Recursion

✅ 1. The book sitting on the table is blue.

❌ 2. The book sits on the table is blue.

Recursion

✅ 1. The book sitting on the table is blue.

❌ 2. The book sits on the table is blue.

<u>Test:</u> Do models recognize that (1) is better than (2)?

Recursion

✅ 1. The book sitting <u>on the table</u> is blue.

❌ 2. The book sits <u>on the table</u> is blue.

<u>Test:</u> Do models recognize that (1) is better than (2)?

Recursion

✅ 1. The book sitting <u>on the table</u> <u>in the kitchen</u> is blue.

❌ 2. The book sits <u>on the table</u> <u>in the kitchen</u> is blue.

<u>Test:</u> Do models recognize that (1) is better than (2)?

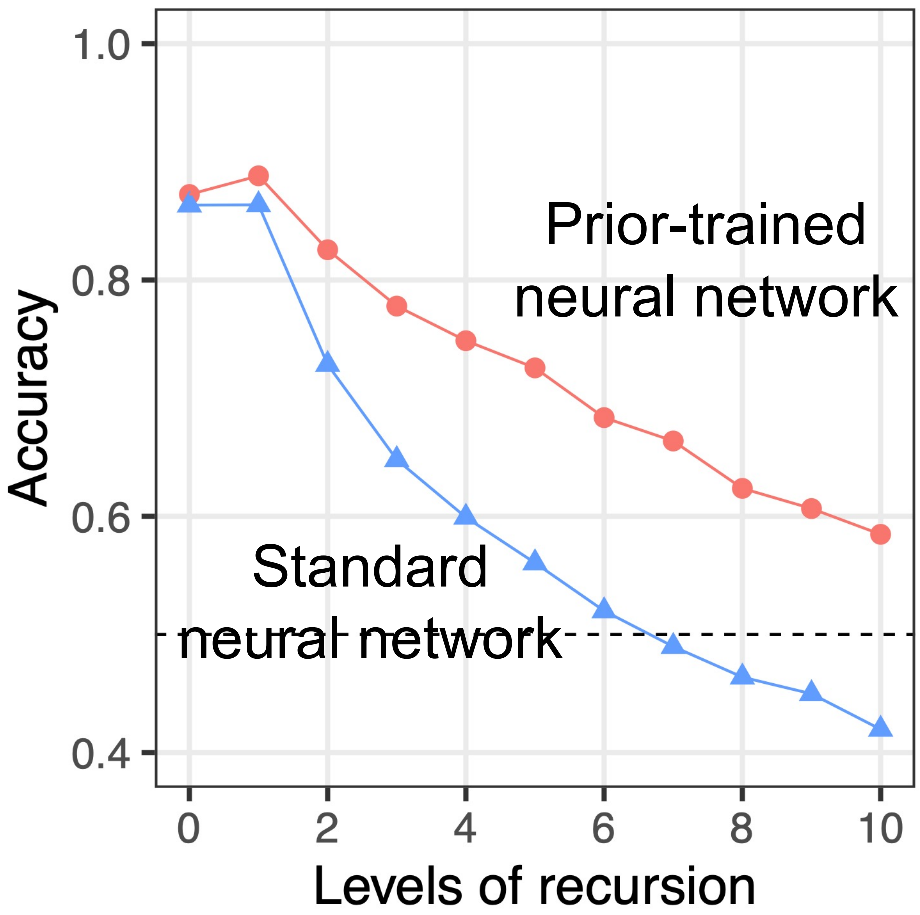
Recursion

✅ 1. The book sitting <u>on the table</u> <u>in the kitchen</u> <u>by the door</u> is blue.

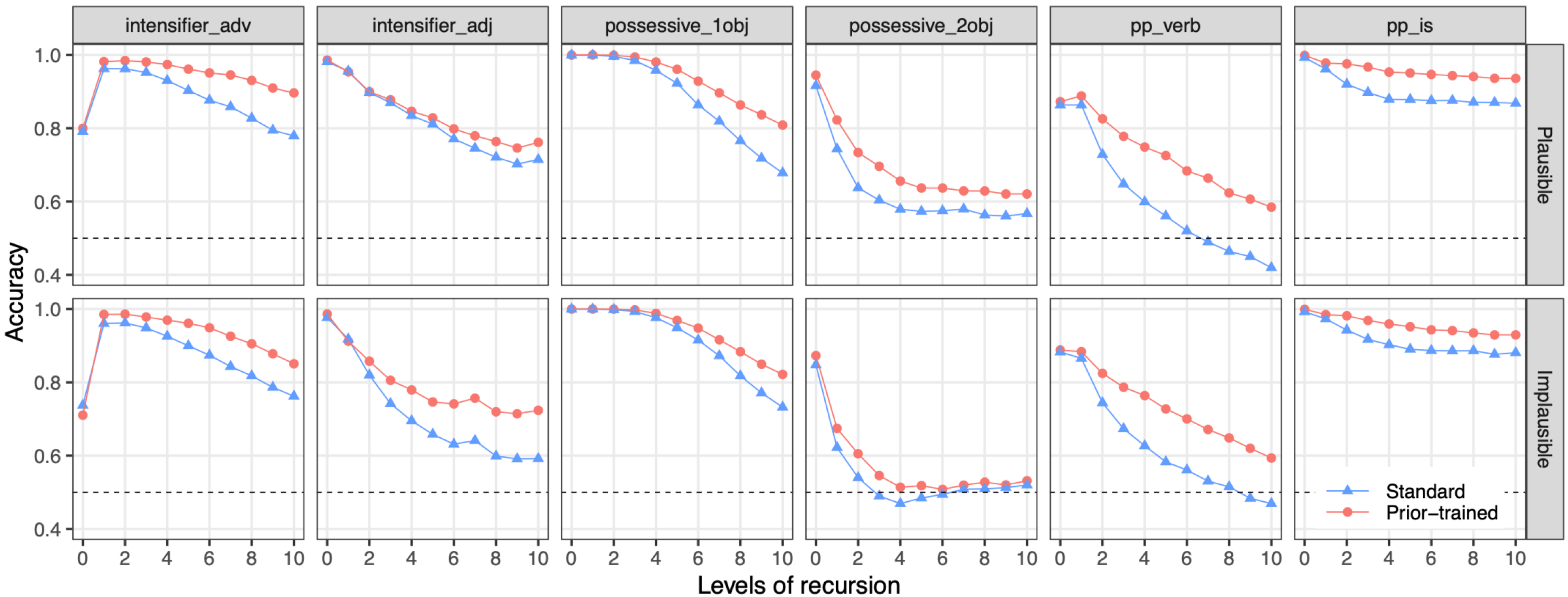
❌ 2. The book sits <u>on the table</u> <u>in the kitchen</u> <u>by the door</u> is blue.

<u>Test:</u> Do models recognize that (1) is better than (2)?

Recursion: Results



Recursion: Results



Learning formal languages from few examples			
Learning aspects of English from naturalistic data			

	Bayesian model		
Learning formal languages from few examples	✅		
Learning aspects of English from naturalistic data	❌		

	Bayesian model	Standard neural network	
Learning formal languages from few examples	✅	❌	
Learning aspects of English from naturalistic data	❌	✅	

	Bayesian model	Standard neural network	Prior-trained neural network
Learning formal languages from few examples	✅	❌	✅
Learning aspects of English from naturalistic data	❌	✅	✅

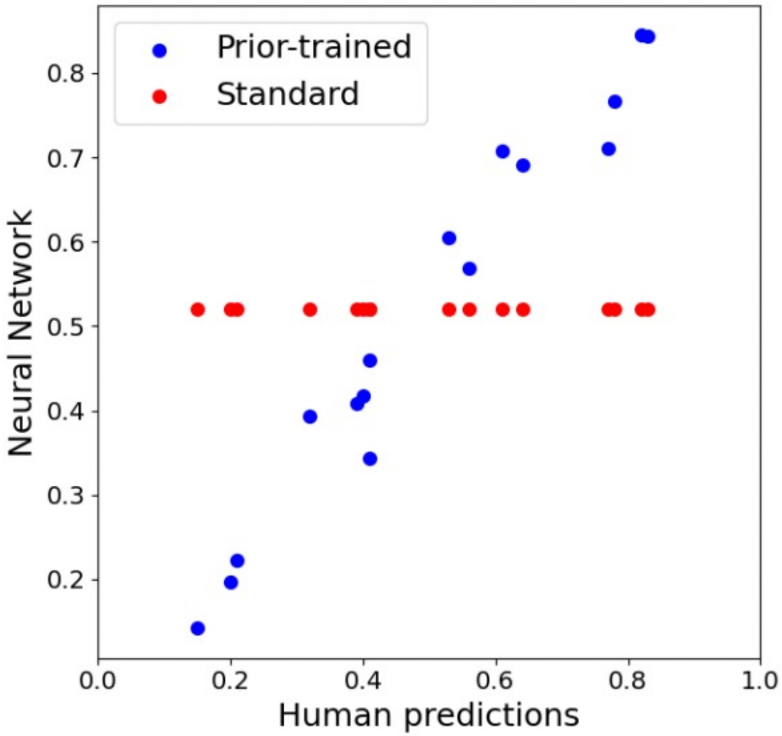
Concept learning

Distilling Symbolic Priors for Concept Learning into Neural Networks

Ioana Marinescu,[1] R. Thomas McCoy,[2] Thomas L. Griffiths[1,3]
ioanam@princeton.edu, tom.mccoy@yale.edu, tomg@princeton.edu
[1]Department of Computer Science, Princeton University
[2]Department of Linguistics, Yale University
[3]Department of Psychology, Princeton University



Concept learning



Conclusion

Q: What type of system should we use to represent language?

A: Vectors that act like symbols

Q: What type of system should we use to represent language?

A: Vectors that <u>act like symbols</u>

Q: What type of system should we use to represent language?

A: Vectors that act like symbols

Compositional representations

Tensor Product Representations

Q: What type of system should we use to represent language?

A: Vectors that act like symbols

Compositional representations
Tensor Product Representations

Strong inductive biases
Inductive bias distillation

Symbolic structures

Classical painting

Vectors

Jackson Pollock

Tensor Product Representations
and
Neural Networks with Meta-Learning

Pointillism

A symbolic victory?

- The framing I’ve used might seem like a resounding victory for the symbolic perspective
  - I.e., the role of vectors is just to implement symbols!
- But the truth is probably more complex
- The fact that the symbols are implemented in vectors might have important consequences
  - Ability to deviate from pure symbolic structure to handle exceptions, context, etc.
  - Flexibility for learning

How to get there?

- On one hand, neural networks naturally develop compositional structure on their own
  - So maybe we don't need to consciously have symbols in mind when developing systems
- On the other hand: Incorporating soft versions of symbols might be useful as an inductive bias
  - Architectures with compositional structure
  - Training paradigms that encourage symbolic processing

Thank you!

- Collaborators:

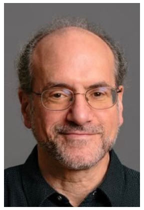
Ewan Dunbar

Erin Grant

Tom Griffiths

Tal Linzen

Ioana Marinescu



Paul Smolensky

Abi Tenenbaum

- Funding: NSF GRFP #1746891, NSF SPRF #2204152
- You!