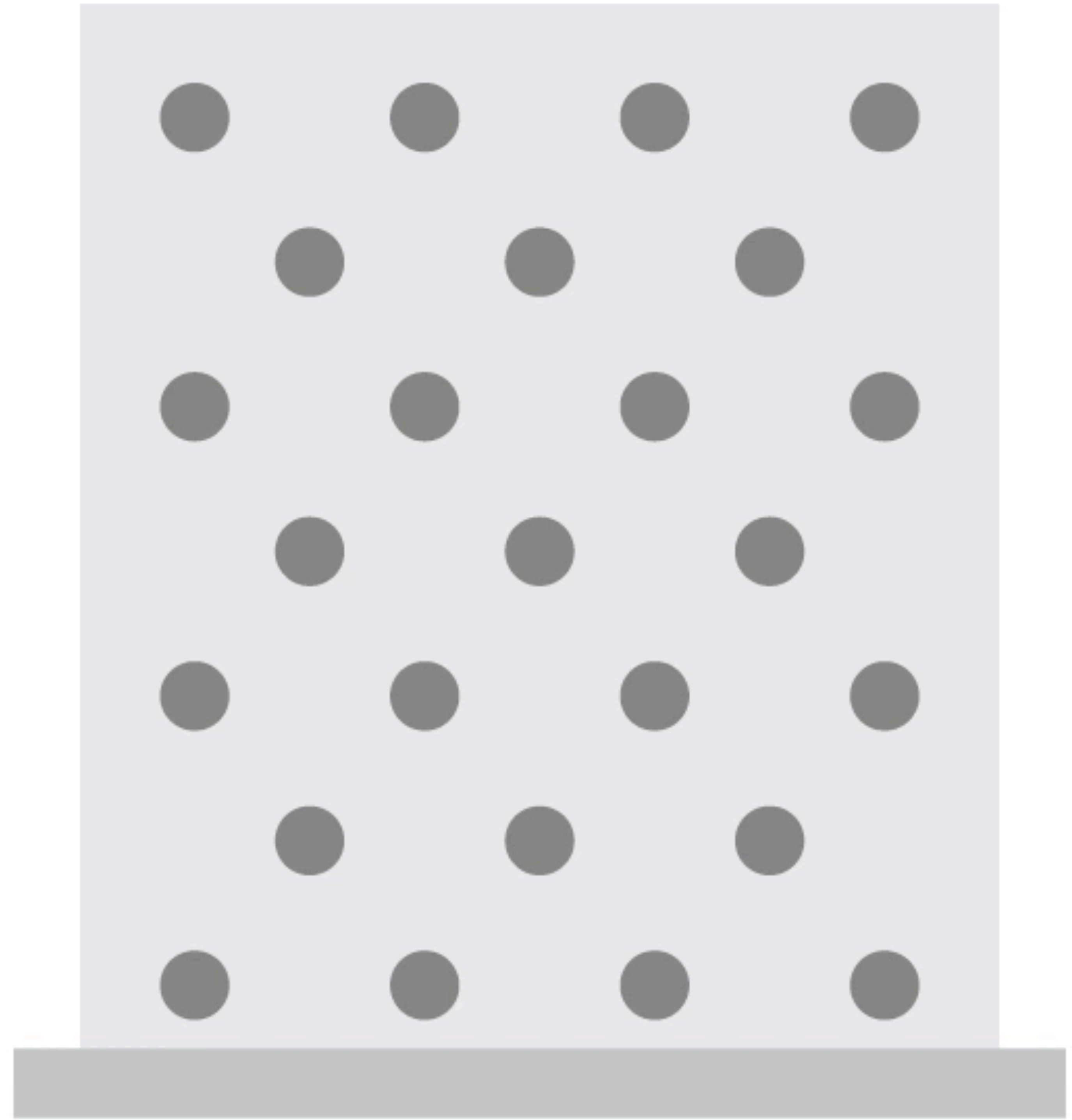
Microsoft

Find x, y

4x+3y=17, 2y-3x=0



Microsoft Blog \ Algorithms \ Next-generation architectures bridge gap between neural and symbolic representations with neural symbols

Strategy today

Compositionality and NNs: Where to start?

- Human cognition: what notion of compositionality does it instantiate?
- NNs: what very general notion of compositionality naturally applies to them?
- Historical source of prominence of compositionality notion: Strong definition $\mathcal{D}$
  - No one would deny that satifying $\mathcal{D}$ constitutes compositionality
  - Idealization, rather than empirically-validated characterization, of human cognition
  - May be too strong to apply to all desired cases
  - BUT: **if NNs can meet this strong definition**, we can dismiss in-principle arguments claiming the impossibility of NNs displaying compositionality
  - AND: (i) Identify the primitive NN competences which enable strong compositionality
    (ii) Endow deep learning with these primitives

Fodor & Pylyshyn 1988. Connectionism and cognitive architecture: A critical analysis *Cognition* 28: 3–71

Strategy today

Compositionality and NNs: Where to start?

- Historical source of prominence of compositionality notion: Strong definition $\mathcal{D}$
  - ➤ BUT: **if NNs can meet this strong definition**, we can dismiss in-principle arguments

Smolensky 1987 The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn *Southern Journal of Philosophy*, 26: 137–161

Fodor & Pylyshyn 1988. Connectionism and cognitive architecture: A critical analysis *Cognition* 28: 3–71

Smolensky 1988 On the proper treatment of connectionism *Behavioral and Brain Sciences* 11: 1–23. *Also:* 11: 59–74; 13: 407–411

Fodor & McLaughlin 1990 Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work *Cognition* 35: 183–204

Smolensky 1991 Connectionism, constituency, and the language of thought. In Loewer & Rey (Eds.) *Meaning in Mind: Fodor and his Critics* 201–227

Smolensky 1995 Constituent structure and explanation in an integrated connectionist/symbolic cognitive architecture. In Macdonald & Macdonald (Eds.) *Connectionism: Debates on Psychological Explanation* Vol 2 221–290

Fodor 1997 Connectionism and the problem of systematicity (continued): Why Smolensky's solution still doesn't work *Cognition* 62: 109–119

Smolensky 2006 Computational levels and integrated connectionist/symbolic explanation. In Smolensky & Legendre *The Harmonic Mind* Vol 2 503–592

Strategy today

Compositionality and NNs: Where to start?

- Historical source of prominence of compositionality notion: $\boxed{\text{Strong definition } \mathcal{D}}$
  - ➤ BUT: **if NNs can meet this strong definition**, we can dismiss in-principle arguments claiming the impossibility of NNs displaying compositionality

- An infinite universe $\mathcal{U}$ of discrete structures: labeled binary trees
- A recursive formal rewrite-rule grammar $\mathcal{G}$ that generates an infinite subset $\mathcal{L}$ of $\mathcal{U}$
- A system $\mathcal{M}$ has *compositional behavior* if it computes f: $\mathcal{L} \rightarrow \mathcal{X}$ where
  f is recursively defined w.r.t. grammar rules in $\mathcal{G}$: $\mathtt{X} \rightarrow \mathtt{A\ B}$; $\mathtt{A} \rightarrow \mathtt{a}$; $\mathtt{B} \rightarrow \mathtt{b}$:
  $f([_{\mathtt{X}}\ \mathtt{a}\ \ \mathtt{b}]) = f_{\mathtt{X}}\,(f_{\mathtt{A}}(\mathtt{a}), f_{\mathtt{B}}(\mathtt{b}))$
- $\mathcal{M}$ has *compositional processing*: procedure for computing f is built from subprocesses computing $f_{\mathtt{X}}$'s
- $\mathcal{M}$ has *compositional representation*: F&P footnote 9, p. 14 "physical instantiation mapping of combinatorial structure" $\mathbf{F}(P\&Q) = \mathbf{B}_{\&}\,[\mathbf{F}(P), \mathbf{F}(Q)]$
- $\mathcal{M}$ has *compositional learning*: ?? Need an induction principle: data $\rightarrow \mathcal{G}$, e.g., MDL

What we know for certain

Capabilities of **KNOWLEDGE** & **PROCESSING**, *not* **LEARNING**

From work of the previous millenium
Next: work of this millenium on learning

What we know for certain

Capabilities of **KNOWLEDGE** & **PROCESSING**, *not* **LEARNING**

Contra F&P 1988: Symbolic ("Classical") computation ***cannot*** explain "systematicity, compositionality, inferential coherence"

These are ***stipulated***, not ***explained***

What we know for certain

Capabilities of **KNOWLEDGE** & **PROCESSING**, *not* **LEARNING**

Contra F&P 1988: Symbolic ("Classical") computation ***cannot*** explain "systematicity, compositionality, inferential coherence"

Massively parallel numerical computation over distributed (dense vectorial) representations ***can***

Incorporate

- Type/token distinction
- Variables which can be bound to values

Smolensky 1988 Analysis of distributed representation of constituent structure in connectionist systems. *NIPS-1987* 730–739

Smolensky 1990 Tensor product variable binding and the representation of symbolic structures in connectionist networks *Artificial Intelligence* 46: 159–216

What we know for certain

Capabilities of **KNOWLEDGE** & **PROCESSING**, *not* **LEARNING**

Contra F&P 1988: Symbolic ("Classical") computation ***cannot*** explain "systematicity, compositionality, inferential coherence"

Massively parallel numerical computation over distributed (dense vectorial) representations ***can***

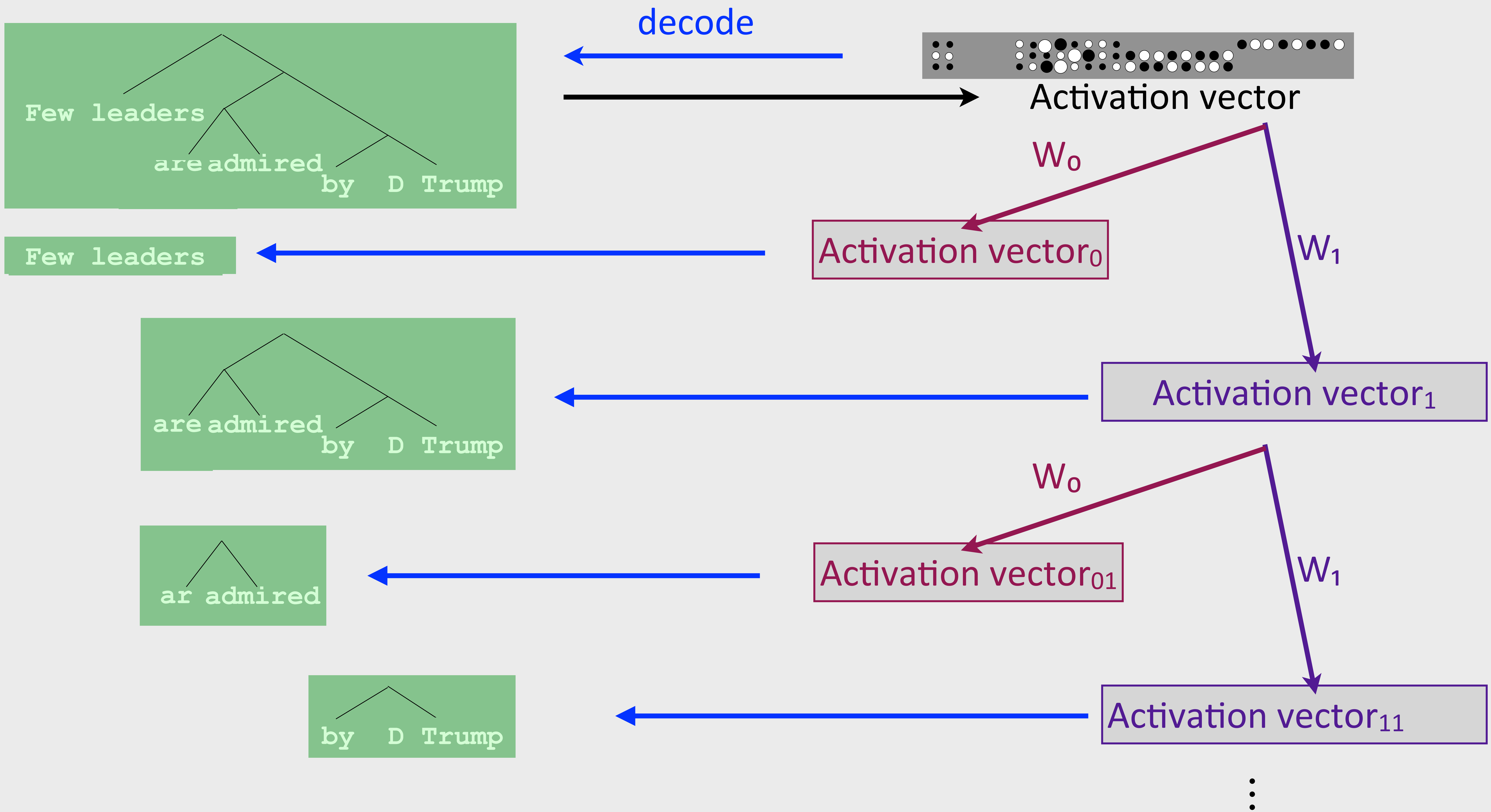
Incorporate

- Type/token distinction
- Variables which can be bound to values
- Embedding of combinatorial constituents within others
- Recursive structure (e.g., trees)

Legendre, Miyata & Smolensky 1991 Distributed recursive structure processing. *NIPS-1990* 591–597

Smolensky & Legendre 2006 *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. Vol. 1: Cognitive Architecture* MIT Press.

Recursive compositional structure of activation vectors



What we know for certain

Capabilities of **KNOWLEDGE** & **PROCESSING**, *not* **LEARNING**

Legendre, Miyata & Smolensky 1990. Harmonic Grammar — A formal multi−level connectionist theory of linguistic well−formedness: Theoretical foundations. *CogSci-1990* 388–395

Smolensky 1993 Harmonic Grammars for formal languages. ***NIPS-1992*** 847–854.

Prince & Smolensky 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*

Smolensky & Legendre 2006 *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. Vol. 2: Linguistic and Philosophical Implications* MIT Press

Cho, Goldrick & Smolensky 2017 Incremental parsing in a continuous dynamical system: Sentence processing in Gradient Symbolic Computation *Linguistics Vanguard* 3:1

- Embedding of combinatorial constituents within others
- Recursive structure (e.g., trees)
- Grammars controlling constituent combination
- New grammar formalisms that have transformed parts of formal linguistic theory

What we know for certain

Capabilities of **KNOWLEDGE** & **PROCESSING**, *not* **LEARNING**

Smolensky 2012 Symbolic functions from neural computation *Philosophical Transactions of the Royal Society — A: Mathematical, Physical and Engineering Sciences* 370: 3543–3569

Massively parallel numerical computation over distributed (dense vectorial) representations ***can***

Incorporate

- Type/token distinction
- Variables which can be bound to values
- Embedding of combinatorial constituents within others
- Recursive structure (e.g., trees)
- Grammars controlling constituent combination
- New grammar formalisms that have transformed parts of formal linguistic theory

Compute

- Structure sensitive functions
- Recursive functions in formally specified families

What we know for certain

Capabilities of **KNOWLEDGE** & **PROCESSING**, *not* **LEARNING**

Smolensky 1995 Constituent structure and explanation in an integrated connectionist/symbolic cognitive architecture. In Macdonald & Macdonald (Eds.) *Connectionism: Debates on Psychological Explanation* Vol 2 221–290

Massively parallel numerical computation over distributed (dense vectorial) representations

Incorporate

- Type/token distinction
- Variables which can be bound to values
- Embedding of combinatorial constituents within others
- Recursive structure (e.g., trees)
- Grammars controlling constituent combination
- New grammar formalisms that have transformed parts of formal linguistic theory

Compute

- Structure sensitive functions
- Recursive functions in formally specified families

All this is enabled by Tensor Product Representations (TPRs): primitives enabling strong compositionality

What we know for certain

Capabilities of **LEARNING**

~~From work of the previous millenium~~
Now: work of this millenium on learning

What we know for certain

Capabilities of **LEARNING**

Standard DNNs learning highly structure-sensitive functions can create combinatorial distributed representations (TPRs) that can be explicitly specified

Enhance: DNNs specially-designed with hidden representations that are TPRs **can invent their own types of symbol structures**

These invented symbol structures improve performance on compositional tasks

† Chen, Huang, Palangi, Smolensky, Forbus, Gao 2019 Natural-to formal-language generation using Tensor Product Representations arXiv:1910.02339

Talk @KR2ML yesterday

McCoy, Linzen, Dunbar, Smolensky 2019 RNNs Implicitly Implement Tensor Product Representations *ICLR-2019* arXiv:1812.08718

MathQA: Example answers — Kezhen Chen†

† Chen, Huang, Palangi, Smolensky, Forbus, Gao 2019 Natural-to formal-language generation using Tensor Product Representations arXiv:1910.02339

what is the sum of the multiples of 4 between 38 and 127 inclusive ?

(add n1 const_2) (subtract n2 const_3) (add #0 #1) (subtract #1 #0) (divide #3 n0) (divide #2 const_2) (add #4 const_1) (multiply #6 #5)

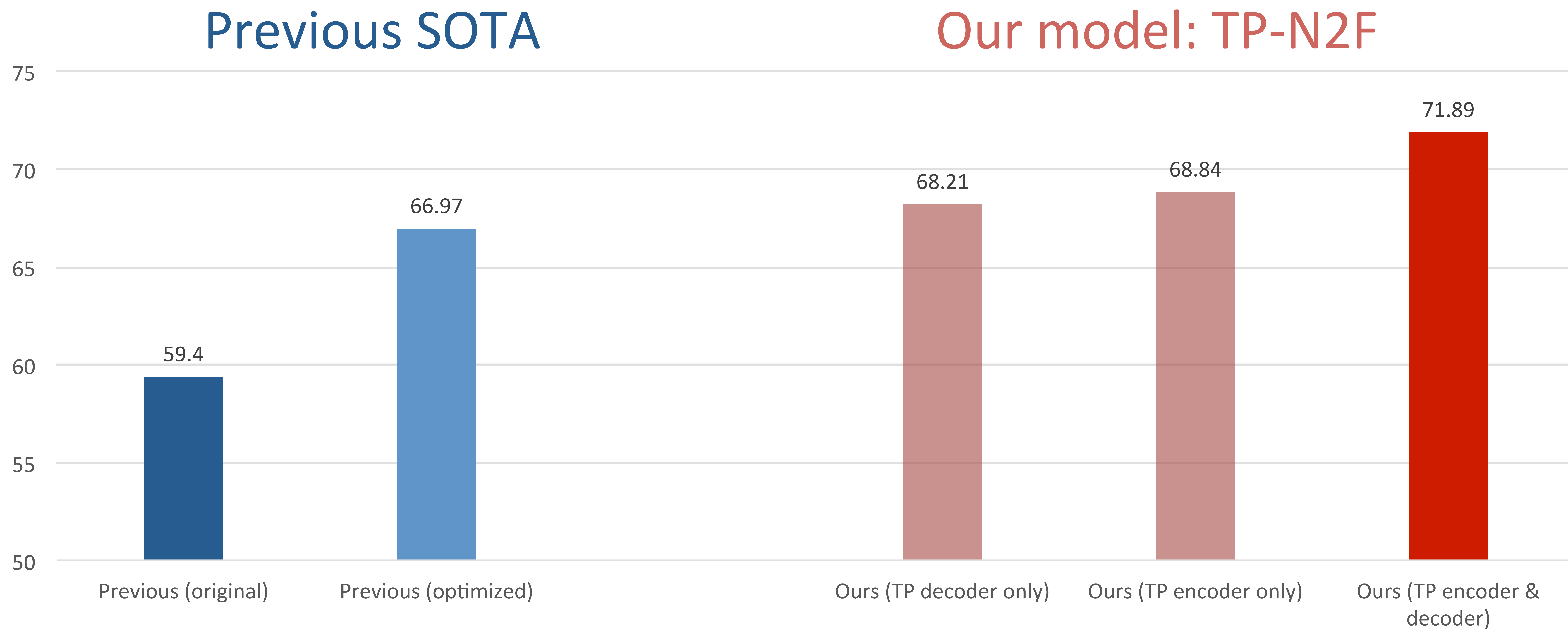
this year , mbb consulting fired 6 % of its employees and left remaining employee salaries unchanged . sally , a first - year post - mba consultant , noticed that that the average (arithmetic mean) of employee salaries at mbb was 10 % more after the employee headcount reduction than before . the total salary pool allocated to employees after headcount reduction is what percent of that before the headcount reduction ?

(multiply n1 const_100) (subtract const_100 n0) (add #0 const_100) (add #1 const_4) (multiply #2 #3) (divide #4 #0)

a high school has 360 students 1 / 2 attend the arithmetic club , 5 / 8 attend the biology club and 3 / 4 attend the chemistry club . 3 / 8 attend all 3 clubs . if every student attends at least one club how many students attend exactly 2 clubs .

(multiply n0 n1) (multiply n0 n3) (multiply n0 n5) (divide #0 n2) (divide #1 n4) (divide #2 n6) (divide #2 n4) (add #3 #4) (multiply n2 #6) (add #7 #5) (subtract #9 #8) (subtract #10 n0)

MATH QA ACCURACY: EXACTLY MATCHING PROGRAM



ALGOLISP: EXAMPLE ANSWERS

consider a number , your task is to find the given number factorial

(<= arg1 1) (− arg1 1) (self #1) (* #2 arg1) (if #0 1 #3) (lambda1 #4) (invoke1 #5 a)

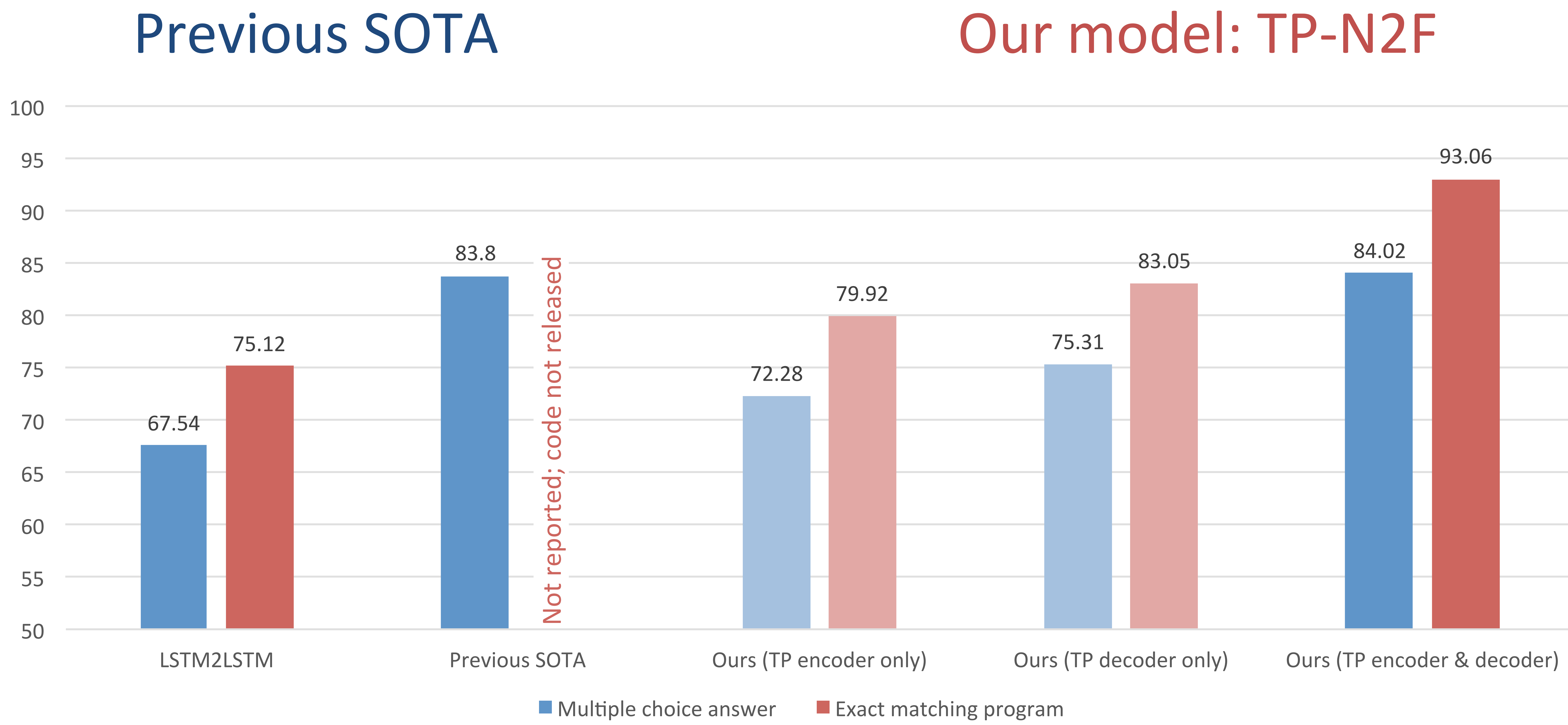
consider a number , your task is to find the given number factorial

(<= arg1 1) (− arg1 1) (self #1) (* #2 arg1) (if #0 1 #3) (lambda1 #4) (invoke1 #5 a)

you are given numbers a , b and d and an array of numbers c , let how many times you can replace a with sum of its digits before it becomes a single digit number and b be the coordinates of one end and the length of the longest subsequence of c with the first value of the subsequence equal to one and all values except for the first equal to the previous value plus one and d be the coordinates of another end of segment e , what is the length of segment e rounded down

(digits arg1) (len #0) (== #1 1) (digits arg1) (reduce #3 0 +) (self #4) (+ 1 #5)
(if #2 0 #6) (lambda1 #7) (invoke1 #8 a) (== arg1 arg2) (+ arg1 1) (if #10 #11 arg1)
(lambda2 #12) (reduce c 1 #13) (− #14 1) (− #9 #15) (digits arg1) (len #17) (== #18 1)
(digits arg1) (reduce #20 0 +) (self #21) (+ 1 #22) (if #19 0 #23) (lambda1 #24)
(invoke1 #25 a) (== arg1 arg2) (+ arg1 1) (if #27 #28 arg1) (lambda2 #29)
(reduce c 1 #30) (− #31 1) (− #26 #32) (* #16 #33) (− b d) (− b d) (* #35 #36)
(+ #34 #37) (sqrt #38) (floor #39)

ALGOLISP ACCURACY: ANSWER / PROGRAM



What we know for certain

Capabilities of **LEARNING**

Standard DNNs learning highly structure-sensitive functions can create combinatorial distributed representations (TPRs) that can be explicitly specified

Enhance: DNNs specially-designed with hidden representations that are TPRs **can invent their own types of symbol structures**

These invented symbol structures improve performance on compositional tasks

* Schlag, Smolensky, Fernandez, Jojic, Schmidhuber, Gao 2019 Enhancing the Transformer with Explicit Relational Encoding for Math Problem -Solving arXiv:1910.06611†

Poster here today

Mathematics Dataset (DeepMind) — Imanol Schlag+

* Schlag, Smolensky, Fernandez, Jojic, Schmidhuber, Gao 2019 Enhancing the Transformer with Explicit Relational Encoding for Math Problem -Solving arXiv:1910.06611

Suppose 0 = 2*a + 3*a – 150. Let p = 106 – 101. Suppose –3*b + w + 544 = 3*w, –p*b – 5*w = –910. What is the greatest common factor of b and a?

30

Let q(r) = 33*r. Let a(y) = –y**2 + 2*y – 2. Let p be a(1). Let d be q(p). Let n = 38 + d. Solve –5*v – 11 = –3*c – 0*v, –4*c = n*v + 32 for c.

–3

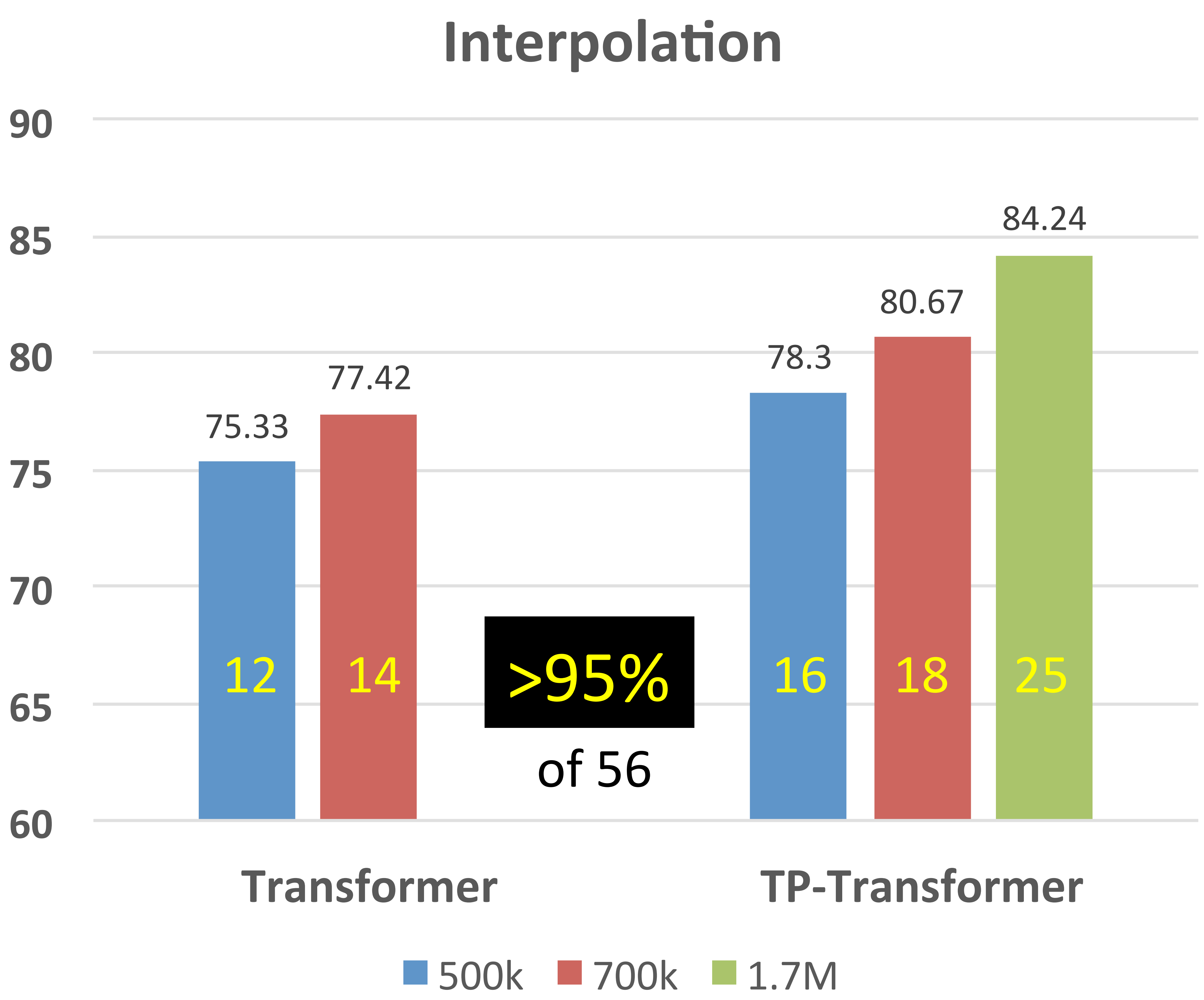
Let r(g) be the second derivative of 2*g**3/3 – 21*g**2/2 + 10*g. Let z be r(7). Factor –z*s + 6 – 9*s**2 + 0*s + 6*s**2.

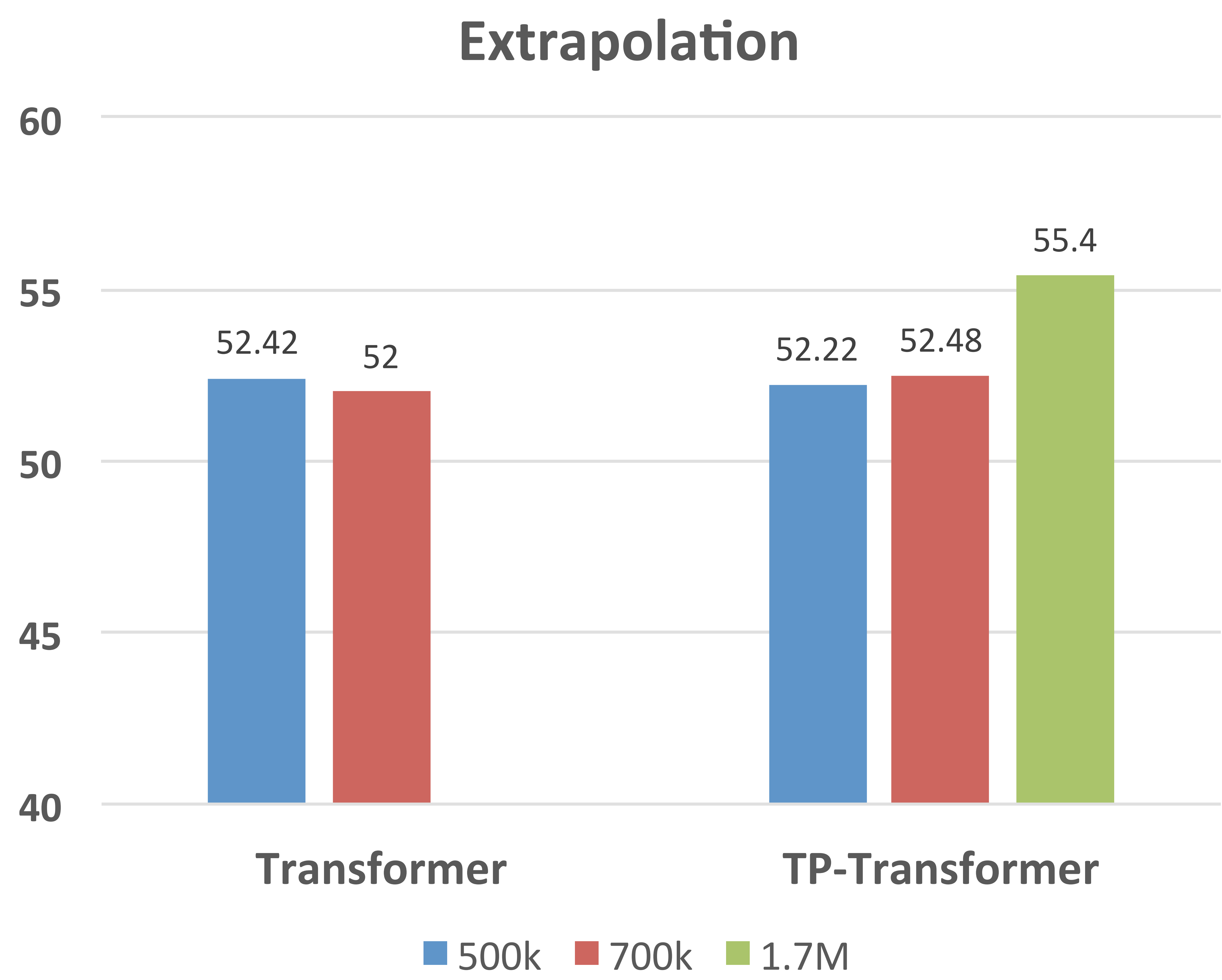
–(s + 3)*(3*s – 2)

Let m(i) be the first derivative of 3/55*i**5 + 0*i – 73 + 0*i**3 + 0*i**2 – 5/66*i**6 + 1/22*i**4. Let m(d) = 0. Calculate d.

–2/5, 0, 1

Math Dataset Accuracy: Average EM





What we know for certain

Capabilities of **LEARNING**

Standard DNNs learning highly structure-sensitive functions can create combinatorial distributed representations (TPRs) that can be explicitly specified

Enhance: DNNs specially-designed with hidden representations that are TPRs **can invent their own types of symbol structures**

These invented symbol structures improve performance on compositional tasks

We can interpret these invented symbol structures (partially)

- Grammatical structure
- Algebraic structure

* Schlag, Smolensky, Fernandez, Jojic, Schmidhuber, Gao 2019 Enhancing the Transformer with Explicit Relational Encoding for Math Problem -Solving arXiv:1910.06611†

Palangi, Smolensky, He, Deng 2018 Question-answering with grammatically-interpretable representations *AAAI-2018* arXiv:1705.08432

Huang, Smolensky, He, Deng, Wu 2018 Tensor Product Generation Networks for deep NLP learning *NAACL-2018* arXiv:1709.09118

Interpreting Learned Structures

TP-Transformer model, arthmetic structure

- digits in the denominator of a fraction are assigned one set of structural relations
- digits in the numerator are assigned a different relations

TP-N2F model of MathQA solution-program generation, vectors for operators:

- general-purpose operators in one region of the vector space: add, negate, log
- shape-specific geometric computations in a different region: square_area, volume_cylinder, surface_cube
- At one edge of the space: max, min; at another: factorial, choose

What we know for certain

Capabilities of **LEARNING**

Standard DNNs learning highly structure-sensitive functions can create combinatorial distributed representations (TPRs) that can be explicitly specified

Enhance: DNNs specially-designed with hidden representations that are TPRs **can invent their own types of symbol structures**

These invented symbol structures improve performance on compositional tasks

We can interpret these invented symbol structures (partially)

We can directly alter hidden constituents to control network outputs

* Soulos, McCoy, Linzen, Smolensky 2019 Discovering the Compositional Structure of Vector Representations with Role Learning Networks arXiv:1910.09113

Talk/poster here today

Precision surgery on hidden representations — Paul Soulos+

* Soulos, McCoy, Linzen, Smolensky 2019 Discovering the Compositional Structure of Vector Representations with Role Learning Networks arXiv:1910.09113

We can directly alter hidden constituents to control network outputs: SCAN task

run left twice after jump opposite right thrice

run:11 left:36 twice:8 after:43 jump:10 opposite:17 right:4 thrice:46 →
TR TR JUMP TR TR JUMP TR TR JUMP TL RUN TL RUN

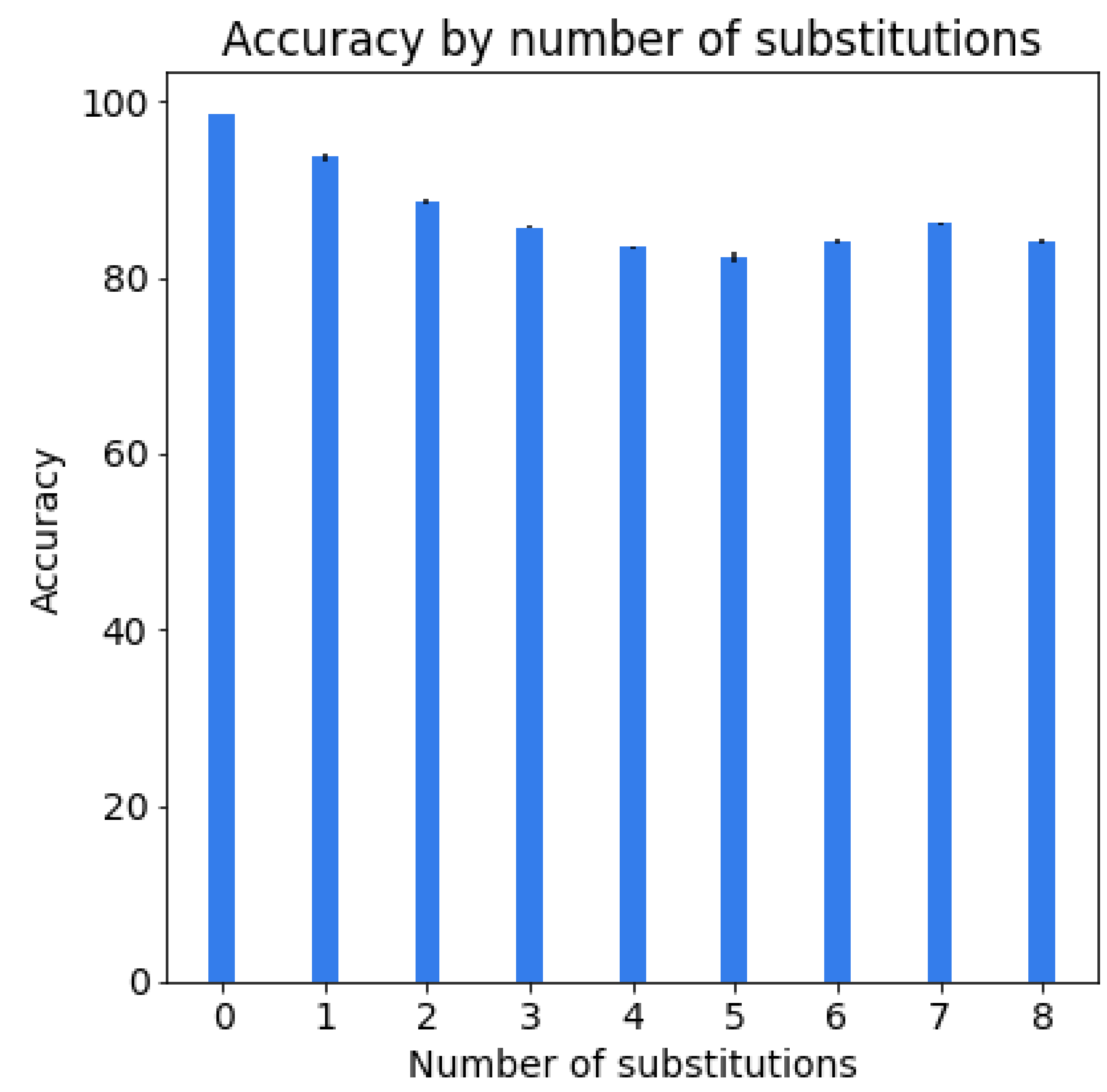
− run:11 + look:11 →
TR TR JUMP TR TR JUMP TR TR JUMP TL LOOK TL LOOK

− jump:10 + walk:10 →
TR TR WALK TR TR WALK TR TR WALK TL LOOK TL LOOK

− left:36 + right:36 →
TR TR WALK TR TR WALK TR TR WALK TR LOOK TR LOOK

− twice:8 + thrice:8 →
TR TR WALK TR TR WALK TR TR WALK TR LOOK
TR LOOK TR LOOK

− opposite:17 + around:17 →
TR WALK TR WALK TR WALK TR WALK TR WALK TR WALK
TR WALK TR WALK TR WALK TR WALK TR WALK TR WALK
TR LOOK TR LOOK TR LOOK



What we know for certain

Capabilities of **LEARNING**

Standard DNNs learning highly structure-sensitive functions can create combinatorial distributed representations (TPRs) that can be explicitly specified

Enhance: DNNs specially-designed with hidden representations that are TPRs **can invent their own types of symbol structures**

These invented symbol structures improve performance on compositional tasks

We can interpret these invented symbol structures (partially)

We can directly alter hidden constituents to control network outputs

What we don't yet know

Can we interpret these invented structures sufficiently to (i) understand how DNNs create and process them and (ii) explain the DNNs' task performance?

Can the newly-invented symbol systems inform our understanding of the tasks and our theories of how human cognition performs them?

Thank you for your attention