Neurocompositional computing: From the Central Paradox of Cognition to a new generation of AI systems

Paul Smolensky[1,2] | **Richard Thomas McCoy**[1] | **Roland Fernandez**[2] |
Matthew Goldrick[3] | **Jianfeng Gao**[2]

[1]Department of Cognitive Science, Johns Hopkins University, Baltimore, Maryland, USA

[2]Microsoft Research, Redmond, Washington, USA

[3]Department of Linguistics, Northwestern University, Evanston, Illinois, USA

Correspondence
Paul Smolensky, Department of Cognitive Science, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218, USA.
Email: paul.smolensky@jhu.edu

Richard Thomas McCoy, Department of Cognitive Science, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218, USA.
Email: tom.mccoy@jhu.edu

Funding information
NSF, Grant/Award Numbers: BCS-1344269, DGE-0549379, GRFP 1746891; Microsoft Research

Abstract

What explains the dramatic progress from 20th-century to 21st-century AI, and how can the remaining limitations of current AI be overcome? The widely accepted narrative attributes this progress to massive increases in the quantity of computational and data resources available to support statistical learning in deep artificial neural networks. We show that an additional crucial factor is the development of a new type of computation. *Neurocompositional computing* adopts two principles that must be simultaneously respected to enable human-level cognition: the principles of Compositionality and Continuity. These have seemed irreconcilable until the recent mathematical discovery that compositionality can be realized not only through discrete methods of symbolic computing, but also through novel forms of continuous neural computing. The revolutionary recent progress in AI has resulted from the use of limited forms of neurocompositional computing. New, deeper forms of neurocompositional computing create AI systems that are more robust, accurate, and comprehensible.

INTRODUCTION

Artificial Intelligence (AI) has been a long time coming of age. It was Ada Lovelace who recognized in 1843 the profound implications, extending far beyond numerical calculation, of Charles Babbage's design for a general-purpose numerical computer: "in enabling mechanism to combine together *general* symbols, in successions of unlimited variety and extent, a uniting link is established between the operations of matter and … abstract mental processes" (Bowden 1953, Note A, p. 368). Now, more than a century and a half later, the widespread deployment of AI has become a landmark achievement of the 21st century, a turning point that has been likened to the industrial revolution of the 18th century (Thomas 2020). Properly understanding this breakthrough is critical for controlling how AI develops in the remainder of the century, both for maximizing its benefits and minimizing its hazards (Grosz and Stone 2018). Here we present a new analysis of this progress, an analysis which is yielding a new generation of AI systems.

The recent progress in AI—specifically, AI based on artificial neural networks exploiting deep learning—has typically been attributed to merely quantitative technological improvements: greatly increased computing power and data quantity (Sejnowski 2018, p. ix). Correct as far as it goes, this view overlooks a major factor, namely the advent of *neurocompositional computing*

(Smolensky et al. 2022), which, according to a contemporary theory in cognitive science (Smolensky and Legendre 2006), is the form of computing underlying human intelligence. We argue that, in addition to the critical expansion of computing resources, it is the unrecognized emergence of neurocompositional computing—albeit only implicitly, in an acutely restricted form—that has powered the most dramatic recent advances in AI.

Despite the unprecedented progress, it is generally recognized that contemporary AI systems still suffer serious deficiencies, including extreme opacity and weakness in learning general knowledge from limited experience. Here, we show how these problems are being addressed through a deeper development of neurocompositional computing, deployed in the new AI systems we review. In addition to learning more robustly, these new systems are more comprehensible than standard deep learning networks, diminishing the opacity which limits both their utility (Turek 2016) and our ability to control their sometimes societally deleterious behaviors (Bender et al. 2021).

The original contributions of this article, presented in successive sections, are these: identifying key contributors to the power of human intelligence—the Continuity and Compositionality Principles—which together define neurocompositional computing; diagnosing the limitations of 20th-century AI as rooted in the use of computing architectures that failed to respect both these principles; elucidating the recent progress in AI as resulting in significant part from simultaneous adherence to both principles—achieving neurocompositional computing to a limited degree; presenting a general formalism from cognitive science that allows considerably deeper realization of neurocompositional computing: neurally encoded compositionally structured tensor (NECST) computing; showing how new NECST AI models harness the power of neurocompositional computing to mitigate limitations of previous AI systems; illustrating how NECST computing substantially improves both the generalization capacity and the comprehensibility—and consequently the controllability—of AI systems.

THE CENTRAL PARADOX OF COGNITION AND THE PRINCIPLES DEFINING NEUROCOMPOSITIONAL COMPUTING

What type of computing enables human intelligence? Because human intelligence arises from the brain, the obvious answer is: neural computing (Churchland and Sejnowski 2016), in which information is encoded in numerical activation vectors—patterns of activation over groups or layers of neurons (Figure 1A). These encodings form a vector space; the coordinates of an encoding in this space are the numerical activation values of the neurons hosting the activation pattern (Figure 1B). The activation vector that encodes an output results from spreading the activation that encodes an input among multiple layers of neurons through connections of varying strengths or weights. In a typical neural network AI model, the values of these weights are set by training the model on examples of correct input/output pairs. After seeing many such pairs, the model will ideally converge to connection weights that produce the correct output when given an input, not only for the inputs seen during training, but also for novel inputs in a test set.

Once a computational task is encoded in numerical vectors, solving it becomes a problem in statistical inference. A neural network is a complex statistical model with many parameters that are estimated from the statistics of the training data. During learning, the model can compute, and store in its weights, the relations between the numbers in the vector encodings produced when processing the training examples; after learning, during testing, it can use these learned relations to infer an output vector given an input vector that encodes a novel input.

Neural computing respects the *Continuity Principle*: the encoding and processing of information is formalized with real numbers that vary continuously, that is, by amounts that can be arbitrarily small. Continuity means that knowledge about information encoded in one vector automatically generalizes to similar information encoded in nearby vectors (Hinton, McClelland, and Rumelhart 1986). For instance, standard neural systems can readily generalize their knowledge about the word *lock* to the word *fasten* (Figure 1B): these words appear in similar contexts within texts used for training (they have similar "distributional meaning"), so neural-network learning algorithms will encode them as vectors that are close together; as a result, neural processing of these two encodings will produce similar results—similarity-based generalization.

The structure of the vector space can capture other types of relationships as well, such as having a consistent offset that represents a systematic difference between pairs of sentences (Figure 1C). Most importantly, continuity enables deep learning, in which a model's connection weights—even those deeply buried within the network, far from the input and output encodings—can be modified gradually to smoothly improve the model's statistical inference of outputs from inputs in its training set (Rumelhart, Hinton, and Williams 1986).

Although it appears obvious that cognition deploys neural computing, virtually all aspects of human intelligence have been formally and precisely described since antiquity (Kiparsky and Staal 1969) in terms of a very different type of computing: compositional-structure processing (Janssen 2012). In this type of computing
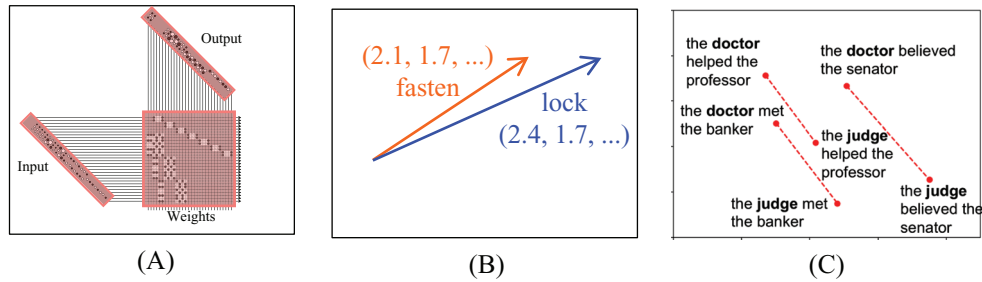
FIGURE 1 Neural computing—encoding information in numerical vector spaces
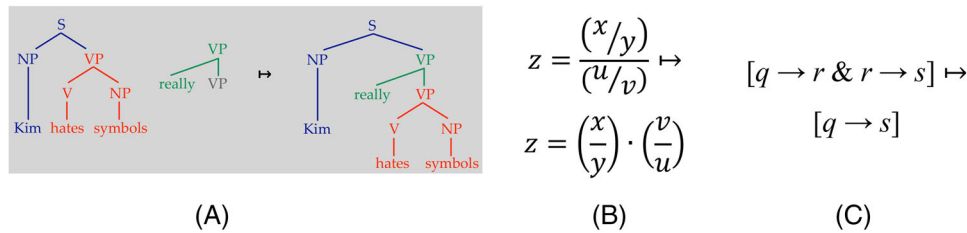

FIGURE 2 Symbolic compositional-structure computing—encoding information in symbol structures

(Figure 2), complex information is encoded in large structures—compositional encodings—which are built by composing together smaller structures that encode simpler information. To process the complex information encoded in a large structure, it suffices to compose together the results of processing the simpler information encoded in its smaller substructures. This is the *Compositionality Principle* (Szabó 2012).

Compositionality has long been seen as key to the power of human cognition: it enables strong *compositional generalization*, enabling us to understand any one of a potentially infinite number of novel situations by encoding the situation internally as a novel composition of familiar, simpler parts, and composing together our understanding of those parts. In virtually all domains of cognition—from vision and speech to reasoning and planning—empirical investigation over millennia has consistently shown how cognitive functions can be well-approximated as computation over appropriate compositional structures, structures with respect to which human cognition exhibits strong compositional generalization (Hinzen, Machery, and Werning 2012). The parts which compose to form such encodings are sometimes familiar elements like objects, words, concepts, and actions, and sometimes scientifically discovered units like phonemes—the basic sounds of speech which combine to form words, each of which is typically denoted by a single letter in alphabetic writing systems.

Robust compositional generalization is all-pervasive in cognition: it underlies the power of both fast, auto-matic, intuitive, largely unconscious cognition (e.g., visual processing of novel arrangements of familiar objects), as well as slow, controlled, deliberative, conscious cognition (e.g., playing chess with a novel initial arrangement of pieces) (Kahneman 2011).

In sum: human cognition gets tremendous power by understanding that *the world is strongly compositional*. Compositional encodings have for centuries been formalized as intricate arrangements of symbols—symbol structures like those we use to represent expressions in algebra or formal logic (Newell 1980). To take a simple example, the two symbol structures [un [lock able]] and [[un lock] able] formalize compositional word encodings that correspond to the two meanings of *unlockable*: not able to be locked (*un-lockable*), and able to be unlocked (*unlock-able*), respectively.

Figure 2 illustrates symbolic compositional-structure processing with three functions (notated "input ↦ output"). In the domain of language, Figure 2A shows the Tree Adjoining function, which here takes as input the phrase-structure trees for the sentence Kim hates symbols and the adverb really and inserts the latter into the middle of the former to produce as output the phrase-structure tree for Kim really hates symbols. This is an operation that is known to give grammars sufficient power to achieve the level of complexity displayed by the syntax of sentences in human languages (Joshi 1985). In the domain of mathematics, Figure 2B shows a function that simplifies a ratio of ratios into a product of ratios. In the domain of logical reasoning, Figure 2C shows a function that chains if-then

implications: it reads *q implies r and r implies s entails q implies s*, for example, knowing that *if it rains then I will drive* ($q \rightarrow r$) and *if I drive then I will need to charge my car* ($r \rightarrow s$) allows us to conclude *if it rains then I will need to charge my car* ($q \rightarrow s$).

The symbols in these structures are abstract entities that are by definition inherently discrete: unlike real numerical values, they cannot be modified by arbitrarily small amounts (although physical drawings of what we perceive as symbols can be). The identity of a symbol, and its position in a symbol structure, are all-or-none: a symbol either is an A or it is not; it either occupies the first position of a symbol sequence or it does not.

As the contrast between Figures 1 and 2 vividly illustrates, discrete symbolic compositional-structure-processing computers and continuous neural computers are profoundly different. Yet, somehow, the computer in our heads apparently is simultaneously a neural computer and a compositional-structure computer (Marcus 2001). How can this be? We call this the Central Paradox of Cognition (Smolensky 1988).

One tempting approach for resolving this paradox would be to posit a hybrid system that combines separate symbolic and neural machines (Andreas et al. 2016; Marcus and Davis 2019; Wang et al. 2020). However, such an approach is only viable if the various components of cognition can be cleanly separated into those that are compositional and those that are continuous. As discussed below, this is not the case: in central cognitive domains, individual encodings respect both the Compositionality and the Continuity Principles. Therefore, a true resolution of the Central Paradox requires a new type of computing—neurocompositional computing—that *simultaneously* satisfies both the Continuity and Compositionality Principles.

20TH-CENTURY AI LIMITATIONS: VIOLATING THE PRINCIPLES

Evidence that human intelligence requires simultaneously respecting both the Continuity and Compositionality Principles comes from a perhaps surprising place: 20th-century AI.

Despite impressive advancements that inspired recurrent optimism, the AI systems built on symbolic computing have consistently fallen far short of human general intelligence (Minsky 1991). The discrete material of symbolic encodings—for example, symbols denoting words—have often proved overly rigid to meet the subtle demands of human cognition; for instance, while such systems can easily generalize from *lock* and *-able* to *lockable*, they cannot easily generalize from *lock* to *fasten* because those two words must be encoded as entirely independent dis-

crete symbols. And the discreteness of the structures themselves also made them insufficiently flexible: even a simple discrete sequential structure for the basic sounds of speech—roughly corresponding to individual letters—turns out to be too rigid, as some sounds cannot be pinned down to a single sequential position but rather simultaneously occupy, to continuously varying degrees, a blend of multiple positions (Smolensky and Goldrick 2016). The strengths and weaknesses of AI systems deploying symbolic computing (Figures 3E–H) arise from these systems respecting the Compositionality Principle, but not the Continuity Principle.

The AI systems (and cognitive models) of the 20th century that employed neural computing typically did the reverse: they respected the Continuity Principle but violated the Compositionality Principle. Continuous vector processing endowed these models with the power to approximate arbitrarily closely any vector-to-vector function (Hornik, Stinchcombe, and White 1989); this suffices for tasks that do not demand a high degree of compositional processing, such as classifying the texture of a single image patch. But violating the Compositionality Principle brought serious limitations which were prominently emphasized by Fodor and Pylyshyn (1988) as well as Marcus (2001). Although neural networks are far less rigid than their symbolic counterparts (Figure 3A,B), they typically suffer from complementary weaknesses (Figure 3C,D), including striking failures of compositional generalization (Baroni 2020; Kim and Linzen 2020). For instance, as noted above, standard neural systems can easily generalize their knowledge about *lock* to *fasten*, but these networks cannot readily generalize to novel compositional structures involving *lock*, such as *lockable*: they lack the compositional encodings that support such generalization.

In sum: traditional neural AI systems suffer because *they do not understand that the world is strongly compositional*—they cannot, because they lack the compositional encodings needed to realize such understanding.

Both symbolic- and neural-based 20th-century AI systems were non-neurocompositional because they violated either the Continuity or the Compositionality Principle—resulting in serious deficiencies. Human intelligence is possible because it deploys a type of computing that simultaneously respects both principles: neurocompositional computing.

21ST-CENTURY AI SUCCESSES: FIRST-GENERATION (1G) NEUROCOMPOSITIONAL COMPUTING

The unprecedented surge in machine intelligence we have witnessed in the 21st century is due largely to progress in AI
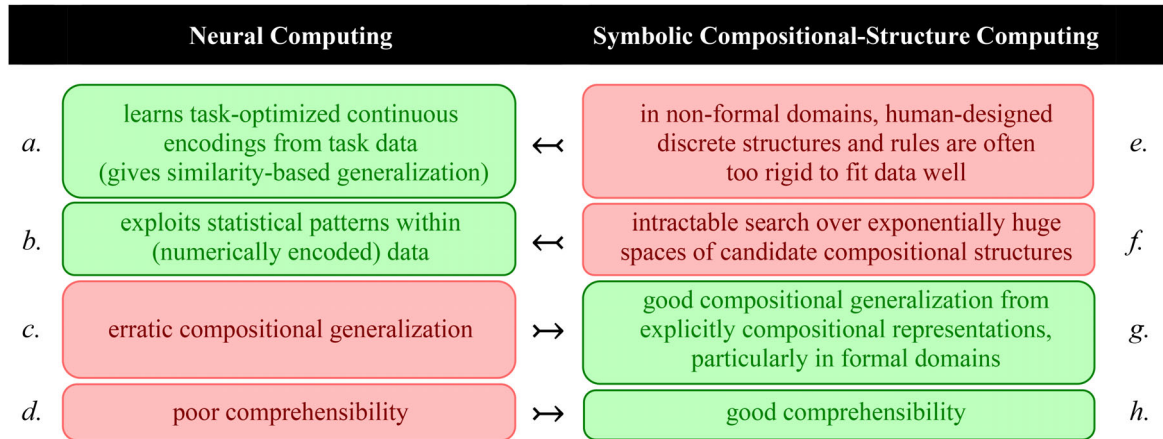
Neural Computing		Symbolic Compositional-Structure Computing	
a.	learns task-optimized continuous encodings from task data (gives similarity-based generalization) ←	in non-formal domains, human-designed discrete structures and rules are often too rigid to fit data well	*e.*
b.	exploits statistical patterns within (numerically encoded) data ←	intractable search over exponentially huge spaces of candidate compositional structures	*f.*
c.	erratic compositional generalization ↣	good compositional generalization from explicitly compositional representations, particularly in formal domains	*g.*
d.	poor comprehensibility ↣	good comprehensibility	*h.*

FIGURE 3 Neural and symbolic compositional-structure computing afford complementary strengths

systems exploiting neural computing. The field has been growing exponentially, and the diversity of network architectures is enormous. However, we can gain insight into the roots of the progress by focusing on two of the most successful and widely used types of networks: Convolutional Neural Networks (CNNs) and Transformers.

CNNs (Fukushima and Miyake 1982; LeCun et al. 1998) have been enormously important in systems operating on visual input. Inspired by the neural structure of mammalian visual systems, CNNs process an input image by analyzing patches of the image with feature detectors. There are multiple layers of feature detectors, each computed by a set of neurons with a preferred stimulus implicitly encoded in its weights. Taking advantage of the Continuity Principle, a CNN's feature detectors are learned by slowly tuning the continuous connection weights that implement them. Each layer is structured as a 2-d grid that mirrors the 2-d spatial structure of the image patches being processed. Within a given layer, each feature detector analyzes all patches; processing is, by design, position-invariant.

Critically, CNN processing also incorporates the Compositionality Principle using spatial structure. At each layer, the analysis of the whole image is gotten by composing together—spatially arranging—analyses of larger patches of the previous layer's analysis of its smaller patches.

Another, more recent, architecture for neural computing, the Transformer (Vaswani et al. 2017), has given rise to a quantum leap in machine language processing (Qiu et al. 2020). A Transformer takes an entire sequence of symbols as input and produces a sequence of symbols as output. To compute the activation vector for a given symbol at a given layer, the Transformer calculates a weighted sum of information from vectors encoding other symbols. It can be shown that this is equivalent to a graph with weighted links between symbols: links along which data flows. In state-of-the-art models, these links partially align with abstract interword relations posited in linguistic theory (Manning et al. 2020).

Thus both of these truly seminal network types—CNNs and Transformers—derive much of their power from their additional compositional structure (Henderson 2020): spatial structure and a type of graph structure. However, these *first-generation (1G) neurocompositional computing* architectures are insufficient for general higher cognition (e.g., reasoning, language). In these domains, compositional structure is not physical but abstract, like a planning structure in which plans contain subplans, which contain sub-subplans (Hendler, Tate, and Drummond 1990). CNNs, limited to spatial structure, are clearly unable to capture such structure.

Transformers, on the other hand, use graphs, which in principle can encode general, abstract structure, including webs of inter-related concepts and facts. However, in Transformers, a layer's graph is defined by its data flow, yet this data flow cannot be accessed by the rest of the network—once a given layer's data-flow graph has been used by that layer, the graph disappears. For the graph to be a bona fide encoding, carrying information to the rest of the network, it would need to be represented with an activation vector that encodes the graph's abstract, compositionally-structured internal information. The technique we introduce next—NECST computing—provides exactly this type of activation vector.

DEEPENING NEUROCOMPOSITIONALITY THROUGH A SOLUTION TO THE CENTRAL PARADOX: NECST COMPUTING

1G neurocompositional computing, illustrated by the CNN and Transformer architectures, has contributed greatly
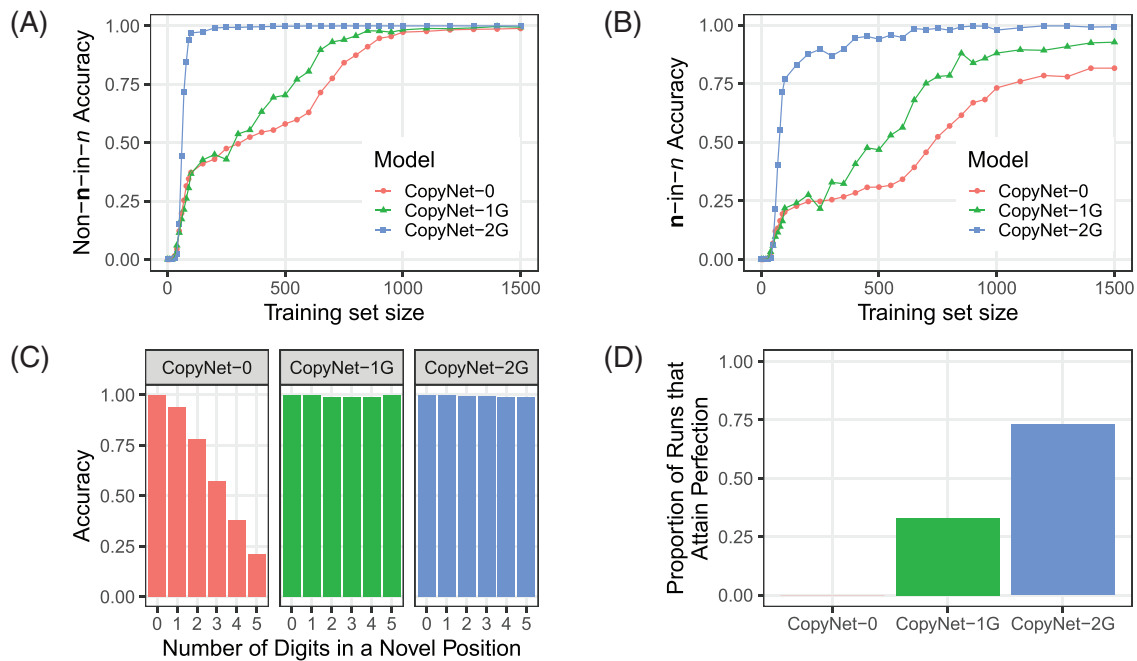
FIGURE 4 Successive increases in a model's degree of neurocompositional structure bring improvements in compositional generalization

to the progress of AI in the 21st century. The work we now present pursues neurocompositional computing more aggressively, aiming for stronger incorporation of the Compositionality Principle by instilling network capabilities that enable explicit construction and processing of general, abstract, compositionally structured activation-vector encodings—while staying strictly within the confines of neural computing to respect the Continuity Principle. This is 2G neurocompositional computing.

Neurocompositionality and compositional generalization

Increasingly deep implementation of neurocompositionality brings increasingly robust compositional generalization. To provide a straightforward initial example, we consider an extremely simple task: taking as input a sequence of five digits (e.g., $\langle 3, 9, 7, 4, 7 \rangle$), encoding the entire sequence internally, and then reproducing that sequence as the output (Figure 4). This copying task demands compositional generalization because it requires learning that if the digit 4 appears in position n in the input, then 4 must be placed in position n of the output, for every possible position n. Similarly, if the digit n appears in position 4 in the input, then n must be placed in position 4 of the output, for every possible digit n.

We test such compositional generalization by withholding from training all sequences of a certain type: "n-in-n" sequences, in which there is a 1 in position 1, or a 2 in

position 2, and so forth. After training, correctly copying the unseen n-in-n sequences requires compositional generalization because digit n has never been seen in position n before.

To see how increasing the degree of neurocompositionality improves compositional generalization, we train a sequence of successively more neurocompositional networks on the copying task (withholding n-in-n examples): COPYNET-0, a pre-Transformer (long-short-term memory, LSTM) neural network (Hochreiter and Schmidhuber 1997), which has little built-in neurocompositional structure; COPYNET-1G, a Transformer, which uses a mildly neurocompositional data-flow-graph structure (1G neurocompositional computing); and COPYNET-2G, a more thoroughly neurocompositional NECST model (2G neurocompositional computing: a NECSTransformer, described below).

Figure 4 gives several illustrations of the improved compositional generalization that comes with increased neurocompositional structure. Figures 4A,B show how models with greater degrees of neurocompositionality perform better with less data—in Figure 4B, when compositional generalization is required, and in Figure 4A when it is not. In the latter case, all models eventually reach 100% correct, but we see that COPYNET-2G reaches this with an order of magnitude fewer training examples. Figure 4C shows that COPYNET-0 does progressively worse on sequences with more and more digits in positions they never occupied during training, while COPYNET-1G and COPYNET-2G show little degradation in performance. Given an n-in-n input,

for example, $\langle 3, 9, 7, 4, 7 \rangle$ (with 4 in position *4*), COPYNET-0 is prone to produce an erroneous output in which neighboring digits are flipped to move n out of position *n*, for example, $\langle 3, 9, 4, 7, 7 \rangle$ (with 4 now in position *3*). In contrast, COPYNET-1G and COPYNET-2G have no difficulty performing the compositional generalization required to produce n-in-*n* outputs.

How do these neural learners' final performance levels compare to a symbolic learner, which would be expected to reach perfect accuracy? We trained 100 distinct instances of each type of model and measured how many of the 100 instances learned the task perfectly (scoring 100% on the set of all possible digit sequences of length 5—including all n-in-*n* sequences). As Figure 4D shows, no COPYNET-0 instance performed perfectly; about one third of COPYNET-1G instances reached perfect performance; and over two thirds of COPYNET-2G instances attained perfection.

In sum, greater neurocompositionality brings faster learning (Figure 4A,B), more robust compositional generalization (Figure 4C), and an increased chance of learning the task perfectly (Figure 4D). All of these improvements lie in areas that have traditionally been strengths of symbolic systems yet weaknesses of neural systems (as exhibited by COPYNET-0). These results, along with findings on more complex tasks reported below, support the conclusion that neurocompositionality endows neural models with some of the strengths of symbolic models while retaining the power of deep learning.

So how, exactly, has neurocompositional computing progressed to 2G?

NECST computing defined

At the foundation of 2G neurocompositional computing is the question: How can fully general, abstract compositional structures be encoded in neural activation vectors so that compositional-structure processing can be carried out on these encodings using only continuous neural computing? The answer we adopt from cognitive science—NECST (Smolensky and Legendre 2006)—is a solution to the Central Paradox of Cognition: How can our own cognitive encodings simultaneously be neural encodings and compositional encodings (Hofstadter 1979)?

The first insight exploited by NECST is that buried within compositional structures are two distinct types of information that are tightly bound together: *what* information and *where* information. Crucially, *where* refers to a position in an abstract structure, not literally a position in physical space.

Consider a simple, discrete compositional structure: a sequence of two symbols, that is, an ordered pair. In [lock able], the *what* comprises two symbols, lock and able;

the *where* comprises two abstract positions: *L* (left) and *R* (right). *L* and *R* are *structural roles*, and they define a structural type: the ordered pair. We can visualize *L* as [☐ —] and *R* as [— ☐]. In [lock able], role *L* is filled by lock and role *R* is filled by able. This structure comprises two *filler-role bindings*, each of which we write as filler:role. Our pair can be written as lock:L *&* able:R and visualized as [lock —] *&* [— able]; the two bindings are aggregated (by the *&* operator) to form a single structure. In NECST computing, this formal characterization of the structure is neurally encoded as a kind of vector, as follows.

Each filler symbol is encoded by an activity vector, a list of real numbers (potentially resulting from deep learning). This is commonplace in neural network modeling: as previously observed, replacing discrete symbols with continuous vectors is at the heart of the 21st-century revolution in machine language processing. A key innovation of NECST is that each *structural role* is similarly encoded by an activity vector. Then the two operations, : for binding each filler to its role and & for aggregating the bindings, are realized as operations on vectors. The aggregation operation & becomes the summation of vectors, +. The binding operation : is a little more complex; it becomes the tensor product $\otimes$ (possibly compressed).[1]

The resulting neural encoding is called the tensor product representation (TPR) of the compositional structure (Smolensky 1990). Hence the name, neurally-encoded compositionally-structured tensor (NECST) computing. There are several neurocompositional architectures known by different names (Eliasmith 2013; Garcez et al. 2019; Gayler 2003), but many of these turn out to be based on TPRs that have been compressed by a linear transformation (Smolensky and Legendre 2006, Ch. 7).

Crucial to the power of symbol structures is that symbols maintain their identity as they play different roles (Mitchell and Lapata 2010, p. 1394). The role *R* of [un lock] and the role *L* of [lock able] are filled by the same symbol, lock; in a symbolic language-processing model, this is why *lock* contributes the same meaning to both *unlock* and *lockable*—making compositional generalization possible. This critical property also holds for the continuous, explicitly compositional vector encodings provided by TPRs, because the same vector that encodes lock appears in the vectors that encode the binding lock:*R* and the binding lock:*L*. TPRs successfully disentangle the *what* and the *where* information carried by compositional structure; these very different types of information are thoroughly fused in traditional noncompositional neural models.

All types of compositional structures comprise a set of filler:role bindings (Newell 1980): TPRs provide a fully general method for neurally encoding compositional

structure. Further, the mathematical operations that underlie TPRs enable systematic, compositional generalization, so that TPRs do not just encode compositional structure but specifically do so in a way that facilitates the behavioral benefits of compositional structure. In fact, mathematical results show that with inputs and outputs encoded as TPRs, neural networks can be designed to precisely compute many complex compositional-structure processing functions—including the types of functions shown in Figure 2 (Smolensky 2012). Thus, the intricate structure-sensitive processing that empowers symbolic computing (Fodor and Pylyshyn 1988) can be performed by networks using TPR encodings. In cognitive science, theories of language based on TPRs have given rise to new computational architectures for grammars; these have significantly advanced linguistic theories explaining the compositional structure of language (Pater 2009; Prince and Smolensky 1997). Networks processing TPRs can enforce the strong kind of constraints on compositional processes that are imposed by grammars.

Thus NECST does indeed provide an answer to the Central Paradox: it provably allows neural computing to carry out important types of compositional-structure processing previously only computable by discrete symbolic computing. But NECST goes beyond symbolic computing, reaping the benefits of the Continuity Principle. TPRs provide a new type of encoding: continuous compositional structure (Goldrick, Putnam, and Schwarz, 2016; Smolensky, Goldrick, and Mathis, 2014). Returning to the example of the ordered pair structure, halfway between the two vectors encoding the discrete roles L and R is a vector which encodes a continuous role; this blend of the L and R roles can be bound to a filler such as t to produce the TPR for a binding we can visualize as $[\boxed{\text{t}}]$. Here, t fills a weak, half-strength role that spans both the left and right halves of the pair: t is half-present in both places at once. Moreover, TPRs enable a single role like L to be filled with a blend of symbols, each simultaneously present (or active) to a continuously variable degree. Such continuous structures may seem odd, but recent work in linguistics (e.g., Rosen, 2019; Zimmermann, 2019) has shown how they make possible theories that cover bodies of data that no single theory using discrete structure can. For example, basic speech sounds can fill continuous roles: the t sound appearing in the middle of the French phrase petit ami (literally, 'small friend') simultaneously resides both at the end of the first and the beginning of the second word (Smolensky, Rosen, and Goldrick, 2020): exactly a case of $[\boxed{\text{t}}]$. Such continuous compositional structure can explain observed complex sound patterns that can be explained by no single theory—even a probabilistic one— that forces such sounds into discrete roles.

PROGRESS FROM EARLY NECST AI: 2G NEUROCOMPOSITIONAL COMPUTING

To illustrate 2G neurocomputing, we present the NECSTransformer (Schlag et al. 2019). Recall that the original Transformer constructs its own implicit data-flow graph for each input symbol sequence, for each layer; this determines, for a given symbol, the extent to which other symbols are consulted for information to determine the vector that encodes the given symbol at that layer. In the NECSTransformer, this vector has explicit compositional structure: it is the TPR for a continuous structure composed from a given number—n_r—of parts determined by n_r roles, roles which vary across symbols, layers, and inputs, and result from deep learning. To create its TPR at a given layer, each symbol generates n_r vectors to encode its roles, and computes a filler for each of these roles. Each filler is a weighted sum of encodings of other symbols in the same layer.

This model generates a graph at each layer, but goes beyond the plain Transformer by labeling each link with a vector that encodes a relation: a relation that holds between the symbol that sends a filler and the symbol receiving it, which binds it to the corresponding role vector it has generated. Crucially, this relationship is no longer merely implicit in the information flow: it is inserted explicitly into the TPR activation vector that encodes the symbol, making it information that is fed directly to subsequent layers for further processing. For example, in a math-problem-solving application, the TPR created to encode a digit in a divisor includes a role interpretable as "denominator," bound to a filler sent from a numerator digit.

Interpreting structural relations invented through deep learning

The explicit encoding of relations computed by the NECSTransformer enables it to learn general abstract graph structures; in fact, each NECSTransformer model uses deep learning to invent its own type of compositional structure to optimally perform its task. On math problem solving, when encoding a ratio of ratios, the invented structures implicitly exploit the inference rule in Figure 2B, assigning the same role to the denominator-of-a-denominator as to a numerator-of-a-numerator (respectively v and x in Figure 2B) (Schlag et al. 2019).

Two other NECST models that process English invent structures that are partially interpretable in grammatical terms, although no information is given to the models about grammar in general or the structure of English in
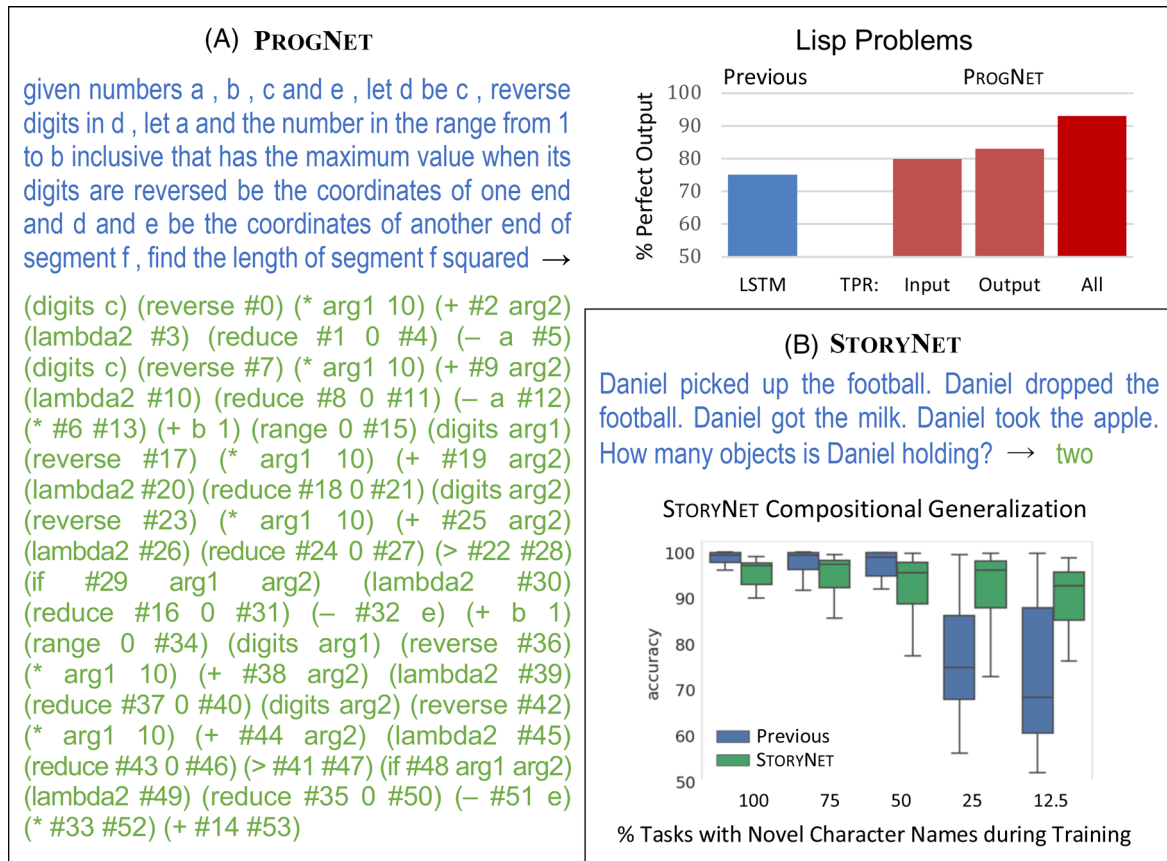
FIGURE 5 NECST models showing improved compositional generalization over previous models from designed, symbolically motivated inductive biases—PROGNET: Chen et al. (2020); STORYNET: Schlag and Schmidhuber (2018)

particular. In a model that learns to answer questions about Wikipedia articles, QANET, the roles learned for TPR encodings of English questions partially align with recognizable linguistic structural properties, at levels ranging from small features of words (such as PLURAL), to general semantic categories like PREDICATE (expressed by verbs and adjectives) to an invented type of multiword sequence that could be called a *wh*-RESTRICTOR-PHRASE, like *what famous event in history* (Palangi et al. 2018). Another model, CAPTIONNET, learns to generate image captions, in the process inventing roles that are highly predictive of the sequence of parts of speech (noun, adjective, etc.) in the captions it generates (Huang et al. 2018; 2019).

Improved compositional generalization

STORYNET (Schlag and Schmidhuber 2018) is a NECST model that answers questions about short synthetic narratives (Weston et al. 2015), such as the one shown in Figure 5B. To test compositional generalization, during training certain character names were allowed to appear in only some fraction of the multiple subtasks in this dataset; the model was then tested on all subtasks using these

withheld character names. The smaller the fraction of subtasks in which a given name was seen during training, the more compositional generalization is required, and as the plot shows, the more STORYNET decisively dominates the previous state-of-the-art system.

On two highly challenging natural tasks involving much more than simple compositionality, NECST models achieved new state-of-the-art performance. One is CAPTIONNET: image captioning requires learning an extremely complex relationship between the two very different, approximately compositional, natures of images and sentences. Another is a NECSTransformer model that generates text summaries (Jiang et al. 2021), SUMMA-RYNET. Without being told to do so, the model invents structure that parcels syntax—sentence structure—into roles, and semantics—meaning—into fillers.

BENEFITS OF IMPROVED COMPREHENSIBILITY FROM 2G NEUROCOMPOSITIONAL COMPUTING

In addition to improved compositional generalization, closer alignment with the type of computing

underlying human cognition renders NECST AI systems more human-comprehensible, as already illustrated by QANET and CAPTIONNET. Because 2G neurocompositional AI systems are significantly less opaque than previous neural models, they deliver several important benefits for overcoming serious problems facing the development of more explainable, trustworthy, and controllable AI systems (Adadi and Berrada 2018).

Informing deep structure learning

Centuries of successful analysis of cognition as compositional-structure processing have given cognitive science and traditional symbolic AI powerful symbolic theories for computing cognitive functions, but the computational chasm separating symbolic compositional processing from traditional noncompositional neural computing has rendered this wealth of insight largely unusable for constructively informing the development of contemporary, deep-learning-based AI. Neurocompositional computing opens the door to using such theories to inform—that is, improve the inductive bias of—neural learning.

An example is provided by the NECST-generation AI model, PROGNET (Figure 5A), which learns to take a problem stated in English and produce a computer program that solves the problem; one version solves math word problems, another, Lisp-programming problems (Chen et al. 2020). Like CAPTIONNET and SUMMARYNET above, PROGNET solves a task requiring far more than identifying compositional structure. Computing the linguistic subparts of the English problem statement (e.g., in Figure 5A, the number in the range from 1 to b inclusive that has the maximum value when its digits are reversed) is an important first step, but translating such English expressions into the corresponding sequence of Lisp commands requires rich knowledge of how the syntax and semantics of English expressions correspond to the syntax and semantics of Lisp operations, and knowledge of how composition of the English subparts relates to composition of the Lisp subparts.

Previous models for such program-generation tasks produce the output program one symbol at a time, but NECST enables models that take advantage of the structure inherent in the programs: a sequence of commands, each a tuple consisting of an operation and the arguments it operates upon, such as (add, x, 5) or (append, list1, list2). PROGNET is designed to bias learning towards producing not a sequence of individual symbols, but a sequence of TPRs, each decodable as such a tuple of symbols comprising a command. Informing the model in this way about the structure of its outputs allowed it to set a new state of

the art for both problem-solving tasks, generating Lisp programs as long as 55 commands perfectly. And the similarity structure of the learned vector encodings of symbols can be partially interpreted: for example, arithmetic operators cluster together, as do geometric functions like area, and operations for processing character sequences.

A second example, already introduced, is STORYNET (Schlag and Schmidhuber 2018), which was provided with valuable biases for compositional learning (recall Figure 5B). This network learned to encode abstractions of entities and relations as vectors and to use built-in operations to bind them together into a TPR that explicitly encodes a continuous graph structure capturing knowledge of the events in the narrative. The learning bias provided to this network was a set of useful built-in operations for updating, as each sentence of the narrative arrives, the vector encoding the continuous knowledge graph, and for sequentially extracting information from the graph to answer questions. The model successfully learned how to use these operations.

Diagnosing errors

Recall that the NECST model QANET, in the service of learning to answer questions about Wikipedia articles, invents a number of roles that can be interpreted grammatically. In addition, some learned fillers can be interpreted semantically (Figure 6B). The word Who has several meanings, and which filler is assigned to a particular instance of Who correlates with which meaning is appropriate for the specific context of that instance. In the articles' many mentions of the TV character named Dr. Who, the model sometimes assigns Who the filler that would be appropriate for a question word, as in Who died? (although QANET made this error at a considerably lower rate than a state-of-the-art probabilistic symbolic system designed for analyzing English sentences, Manning 2017). When the model made this internal filler error, it was five times more likely to produce an incorrect output—an output that would have been appropriate had Who actually been a question word rather than a character name (Palangi et al. 2018).

Controlling output

The incomprehensibility of contemporary black-box noncompositional neural AI models severely limits our ability to control them. This lack of control is problematic; for example, it makes it challenging to address the socially toxic biases the outputs of these models often display (Bender et al. 2021). In contrast, the relative
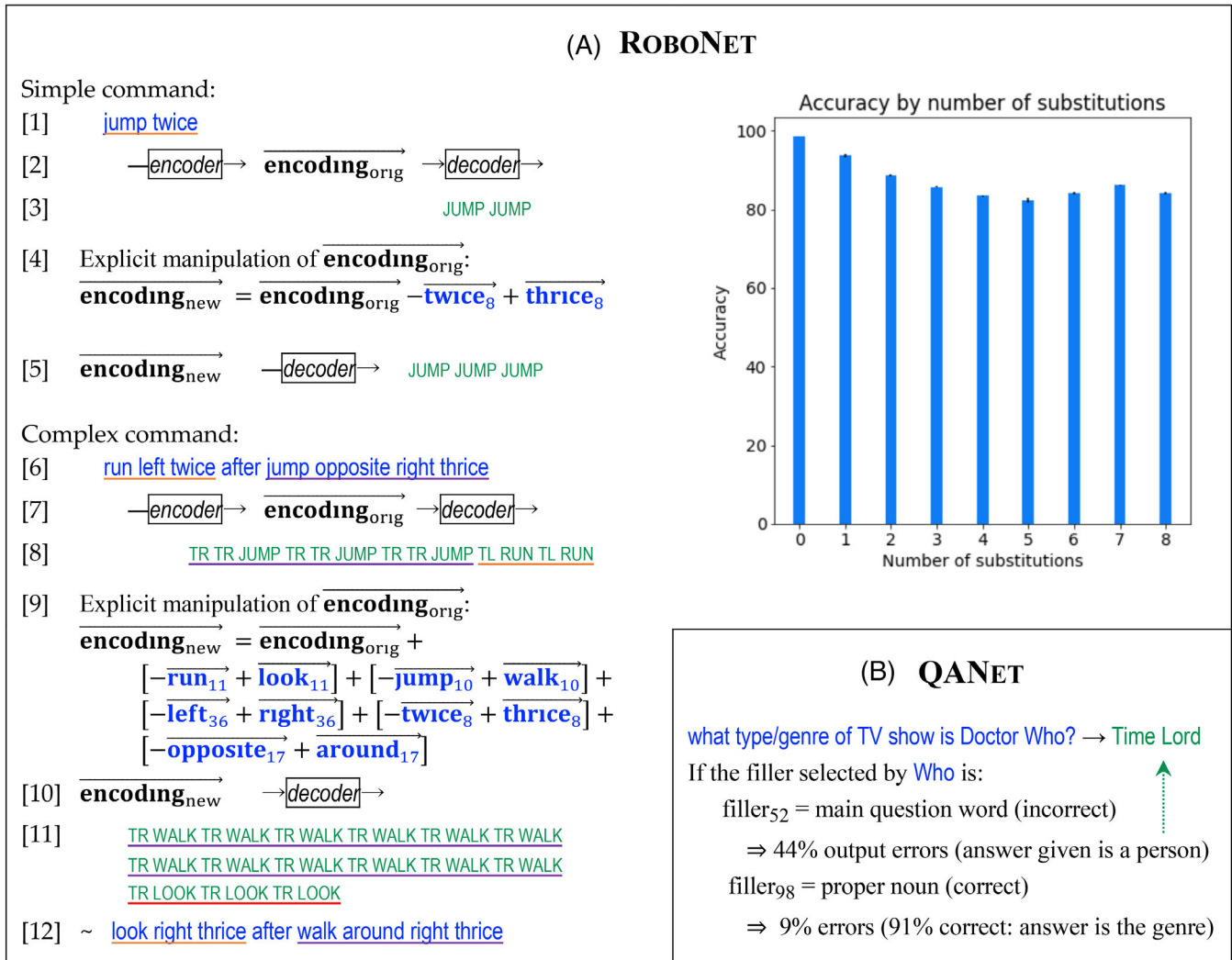
(A) RoboNet

Simple command:

[1]   jump twice

[2] $\longrightarrow$ encoder $\longrightarrow$ $\mathbf{encoding}_{orig}$ $\rightarrow$ decoder $\rightarrow$

[3]   JUMP JUMP

[4]   Explicit manipulation of $\overrightarrow{\mathbf{encoding}_{orig}}$:

$$\overrightarrow{\mathbf{encoding}_{new}} = \overrightarrow{\mathbf{encoding}_{orig}} - \overrightarrow{\mathbf{twice}_8} + \overrightarrow{\mathbf{thrice}_8}$$

[5] $\overrightarrow{\mathbf{encoding}_{new}}$ $\longrightarrow$ decoder $\longrightarrow$ JUMP JUMP JUMP

Complex command:

[6]   run left twice after jump opposite right thrice

[7] $\longrightarrow$ encoder $\longrightarrow$ $\overrightarrow{\mathbf{encoding}_{orig}}$ $\rightarrow$ decoder $\rightarrow$

[8]   TR TR JUMP TR TR JUMP TR TR JUMP TL RUN TL RUN

[9]   Explicit manipulation of $\overrightarrow{\mathbf{encoding}_{orig}}$:

$$\overrightarrow{\mathbf{encoding}_{new}} = \overrightarrow{\mathbf{encoding}_{orig}} +$$
$$[-\overrightarrow{\mathbf{run}_{11}} + \overrightarrow{\mathbf{look}_{11}}] + [-\overrightarrow{\mathbf{jump}_{10}} + \overrightarrow{\mathbf{walk}_{10}}] +$$
$$[-\overrightarrow{\mathbf{left}_{36}} + \overrightarrow{\mathbf{right}_{36}}] + [-\overrightarrow{\mathbf{twice}_8} + \overrightarrow{\mathbf{thrice}_8}] +$$
$$[-\overrightarrow{\mathbf{opposite}_{17}} + \overrightarrow{\mathbf{around}_{17}}]$$

[10] $\overrightarrow{\mathbf{encoding}_{new}}$ $\rightarrow$ decoder $\rightarrow$

[11]   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

[12]   ~   look right thrice after walk around right thrice

Accuracy by number of substitutions

(B) QANet

what type/genre of TV show is Doctor Who? $\rightarrow$ Time Lord

If the filler selected by Who is:

     filler_{52} = main question word (incorrect)

         $\Rightarrow$ 44% output errors (answer given is a person)

     filler_{98} = proper noun (correct)

         $\Rightarrow$ 9% errors (91% correct: answer is the genre)

FIGURE 6 Example NECST analyses demonstrating new capabilities enabled by greater comprehensibility: controlling output and diagnosing errors—RoboNet: Soulos et al. (2020); QANET: Palangi et al. (2018)

comprehensibility of NECST encoding vectors can enable direct intervention—precision surgery—on internal activation patterns in order to control the output of a network.

Our illustration of this introduces a discovery revealed by NECST analysis techniques about the internal encodings learned by standard neural networks (McCoy et al. 2019). (For a demo, see http://rtmccoy.com/tpdn/tpr_demo.html.) One case study concerned a simple hypothetical robot: given an input description of a maneuver in highly simplified English, the task of the network (which we dub RoboNet) is to produce an appropriate output sequence of basic commands to carry out that maneuver (Lake and Baroni 2018).

When RoboNet is given extensive training data that widely exemplifies the compositional possibilities inherent in its highly compositional artificial task, it achieves good compositional generalization. A NECST-based analysis revealed that in this case, the key internal encoding can

be extremely well-approximated by a TPR encoding of the input, with appropriate role assignments to input words and appropriate vector encodings of these words and their roles (Soulos et al. 2020). As depicted in Figure 6A, this understanding of RoboNet's learned encodings as TPRs of approximately discrete structures allows us to write a closed-form expression for the learned vector [2] that internally encodes any given input [1], which in turn tells us exactly how to take the internal encoding of an input such as jump twice and directly modify it into the encoding that would be produced by jump thrice: we subtract the vector hidden in this encoding that encodes twice and add the vector that encodes thrice [4]—in the process, changing the activation level of every neuron by a precisely determined amount. Now the output of the model changes from jump jump [3] to jump jump jump [5]; that is, the alterations that we have made to the model's internal encodings produce exactly the behavioral alteration we intended to

make, illustrating how the interpretable structure of TPRs facilitates control of a model's behavior.

Considerably more complex modifications, and extended sequences of successive modifications, can also be successfully effected in this way. The eight-word input in [6] produces the encoding vector [7] and output [8]. We can change this encoding to match that of a virtual input that differs from it in five words [12] by taking five steps of subtracting the encoding of an existing word and adding in the encoding of a new virtual word [9]. (TL, TR denote Turn-Left and Turn-Right; $\overrightarrow{twice}_8$ is the vector encoding of twice:8, the binding of filler twice to role 8.) The resulting encoding produces the output [11] that correctly corresponds to the virtual new input [12]: the original 13 output commands have been replaced by 30 new ones. The plot shows how, even after many such successive substitutions, the internal modifications specified by our NECST analysis continue to control behavior with high accuracy.

CURRENT LIMITATIONS: THE ROAD TO 3G NEUROCOMPOSITIONAL COMPUTING

The deployment of TPRs in AI has already led to significant progress, but the gains are clearly still short of those expected from full compositional abilities. Why is that? The work to date is only the first step towards fully achieving NECST computing, falling short in several respects. Identifying these shortcomings, and proposing techniques for addressing them, provides a road to 3G neurocompositional computing.

First, while the TPRs of 2G systems successfully disentangle the *what* and *where* aspects of compositional structure, there is nothing to explicitly bias the model to learn processing that takes advantage of this disentanglement to promote compositional generalization. Building such a bias into the network structure will be a key component of 3G neurocompositional computing.

Second, the mathematics of TPRs allows them to embed inside one another just as [lock able] can be embedded within [un [lock able]]. The corresponding neural operations are beyond 2G NECST computing, but are now in development, and will be integral to 3G.

Third, the transformative advances in grammatical theory derived from NECST center on the innovation that grammatical expressions are encoded in activation vectors that optimally satisfy grammatical constraints, which are encoded in the strengths of connections joining neurons (Prince and Smolensky, 1993/2004, 1997). To go beyond the relatively modest gains in natural language processing

observed in 2G neurocompositional computing, incorporating such optimization-based processing into 3G neurocompositional computing may set the stage for major breakthroughs in language processing, as it did in linguistic theory. An initial step in this direction shows how NECST can strengthen inference ability by enabling the meaning of symbols denoting particular entities and relations to adapt continuously to—to be optimized for—their structural context within a knowledge graph of facts (Lalisse and Smolensky 2019).

The research program reviewed here aspires to a reconvergence of AI and cognitive science through a unified theory of the computing underlying both human and machine intelligence. Recognizing the importance of such a synergy is not new: it was already envisioned nearly two centuries ago by Ada Lovelace, who saw that in the general purpose computer, "not only the mental and the material, but the theoretical and the practical, are brought into more intimate and effective connexion with each other" (Bowden 1953, Note A, p. 369). It is our hope that the development of neurocompositional computing will ultimately contribute to the realization of her remarkably prescient vision of intelligent machines.

CONFLICT OF INTEREST

Paul Smolensky, Roland Fernandez and Jianfeng Gao declare a financial interest in Microsoft Corporation. The remaining authors declare there is no conflict of interest.

ORCID

Paul Smolensky https://orcid.org/0000-0003-2420-182X

ENDNOTE

[1] The tensor product yields encodings that are a special kind of vector: a tensor, which we can take to be a set of real numbers that can be visualized as forming a multidimensional grid. For the tensor product of a filler vector $\mathbf{f} = (f_1, f_2, \cdots)$ and a role vector $\mathbf{r} = (r_1, r_2, \cdots)$, this is a 2-D grid; the number in the ith row and jth column is simply $f_i r_j$.

REFERENCES

Adadi, A., and M. Berrada. 2018. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)." *IEEE Access* 6: 52138–60.

Andreas, J., M. Rohrbach, T. Darrell, and D. Klein. 2016. "Neural Module Networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 39–48.

Baroni, M. 2020. "Linguistic Generalization and Compositionality in Modern Artificial Neural Networks." *Philosophical Transactions of the Royal Society B: Biological Sciences* 375(1791): 20190307.

Bender, E. M., T. Gebru, A. McMillan-Major, and S. Shmitchell. 2021. "On the Dangers of Stochastic Parrots: Can Language Models be Too Big?" In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 610–23.

Bowden, B. V. 1953. *Faster Than Thought: A Symposium on Digital Computing Machines*. London: Pitman Publishing, Inc.

Chen, K., Q. Huang, H. Palangi, P. Smolensky, K. D. Forbus, and J. Gao. 2020. "Mapping Natural-Language Problems to Formal-Language Solutions Using Structured Neural Representations." In *Proceedings of the International Conference on Machine Learning*, 1566–75. PMLR.

Churchland, P. S., and T. J. Sejnowski. 2016. *The Computational Brain*. Cambridge, MA: MIT Press.

Eliasmith, C. 2013. *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford: Oxford University Press.

Fodor, J. A., and Z. W. Pylyshyn. 1988. "Connectionism and Cognitive Architecture: A Critical Analysis." *Cognition* 28(1–2): 3–71.

Fukushima, K., and S. Miyake. 1982. "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition." In *Competition and Cooperation in Neural Nets*, 267–85, edited by S. Amari and M. A. Arbib. New York: Springer.

Garcez, A. d., M. Gori, L. L. S. Lamb, M. Spranger, and S. N. Tran. 2019. "Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning." *Journal of Applied Logics* 6(4): 611–32.

W Gayler, R. 2003. "Vector Symbolic Architectures Answer Jackendoff's Challenges for Cognitive Neuroscience." In *Proceedings of the Joint International Conference on Cognitive Science*.

Goldrick, M., M. Putnam, and L. Schwarz. 2016. "Coactivation in Bilingual Grammars: A Computational Account of Code Mixing." *Bilingualism: Language and Cognition* 19(5): 857–76.

Grosz, B. J., and P. Stone. 2018. "A Century-Long Commitment to Assessing Artificial Intelligence and Its Impact on Society." *Communications of the Association for Computing Machinery* 61(12): 68–73.

Henderson, J. 2020. "The Unstoppable Rise of Computational Linguistics in Deep Learning." In *Proceedings of the Association for Computational Linguistics*.

Hendler, J. A., A. Tate, and M. Drummond. 1990. "AI Planning: Systems and Techniques." *AI Magazine* 11(2): 61.

Hinton, G. E., J. L. McClelland, and D. E. Rumelhart. 1986. "Distributed Representations." In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, edited by D. E. Rumelhart, J. L. McClelland and The PDP Research Group, Volume 1, pp. 77–109. Cambridge, MA: MIT Press.

Hinzen, W., E. Machery, and M. Werning, eds. 2012. *The Oxford Handbook of Compositionality*. Oxford: Oxford University Press.

Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9(8): 1735–80.

Hofstadter, D. R. 1979. *Gödel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books.

Hornik, K., M. Stinchcombe, and H. White. 1989. "Multilayer Feedforward Networks are Universal Approximators." *Neural Networks* 2(5): 359–66.

Huang, Q., L. Deng, D. Wu, C. Liu, and X. He. 2019. "Attentive Tensor Product Learning." In *Proceedings of the American Association for Artificial Intelligence*, Volume 33, 1344–51.

Huang, Q., P. Smolensky, X. He, L. Deng, and D. Wu. 2018. "Tensor Product Generation Networks for Deep NLP Modeling." In *Proceedings of the North American Association for Computational Linguistics*.

Janssen, T. M. V. 2012. "Compositionality: Its Historical Context." In *The Oxford Handbook of Compositionality*, edited by W. Hinzen, E. Machery and M. Werning. Oxford: Oxford University Press.

Jiang, Y., A. Celikyilmaz, P. Smolensky, P. Soulos, S. Rao, H. Palangi, R. Fernandez, C. Smith, M. Bansal, and J. Gao. 2021. "Enriching Transformers with Structured Tensor-Product Representations for Abstractive Summarization." In *North American Chapter of the Association for Computational Linguistics*, 4780–93.

Joshi, A. K. 1985. "Tree Adjoining Grammars: How Much Context-Sensitivity is Required to Provide Reasonable Structural Descriptions?" In *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, *Studies in Natural Language Processing*, edited by D. R. Dowty, L. Karttunen, and A. M. Zwicky, 206–50. Cambridge, UK: Cambridge University Press.

Kahneman, D. 2011. *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux.

Kim, N., and T. Linzen. 2020, November. "COGS: A Compositional Generalization Challenge Based on Semantic Interpretation." In *Proceedings of the Empirical Methods in Natural Language Processing*.

Kiparsky, P., and J. F. Staal. 1969. "Syntactic and Semantic Relations in Pāṇini." *Foundations of Language* 5: 83–117.

Lake, B. M., and M. Baroni. 2018. "Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks." In *Proceedings of the International Conference on Machine Learning*, 2879–88.

Lalisse, M. R., and P. Smolensky. 2019. "Augmenting Compositional Models for Knowledge Base Completion Using Gradient Representations." In *Proceedings of the Society for Computation in Linguistics*, 257–66.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient-Based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86(11): 2278–24.

Manning, C. D. 2017. "Stanford Parser." nlp.stanford.edu/software/lex-parser.shtml.

Manning, C. D., K. Clark, J. Hewitt, U. Khandelwal, and O. Levy. 2020. "Emergent Linguistic Structure in Artificial Neural Networks Trained by Self-Supervision." *Proceedings of the National Academy of Sciences* 117(48): 30046–54.

Marcus, G. 2001. *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. Cambridge, MA: MIT Press.

Marcus, G., and E. Davis. 2019. *Rebooting AI: Building Artificial Intelligence We Can Trust*. New York: Pantheon.

McCoy, R. T., T. Linzen, E. Dunbar, and P. Smolensky. 2019. "RNNs Implicitly Implement Tensor Product Representations." In *Proceedings of the International Conference on Learning Representations*.

Minsky, M. 1991. "Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy." *AI Magazine* 12(2): 34–51.

Mitchell, J., and M. Lapata. 2010. "Composition in Distributional Models of Semantics." *Cognitive Science* 34(8): 1388–29.

Newell, A. 1980. "Physical Symbol Systems." *Cognitive Science* 4(2): 135–83.

Palangi, H., P. Smolensky, X. He, and L. Deng. 2018. "Question-Answering with Grammatically-Interpretable Representations." In *Proceedings of the American Association for Artificial Intelligence*, 5350–57.

Pater, J. 2009. "Weighted Constraints in Generative Linguistics." *Cognitive Science* 33(6): 999–1035.

Prince, A. S., and P. Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Malden, MA: Blackwell Publishers.

Prince, A. S., and P. Smolensky. 1997. "Optimality: From Neural Networks to Universal Grammar." *Science* 275(5306): 1604–10.

Qiu, X., T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. 2020. "Pre-Trained Models for Natural Language Processing: A Survey." *Science China Technological Sciences* 63: 1–26.

R Rosen, E. 2019. "Learning Complex Inflectional Paradigms Through Blended Gradient Inputs." In *Proceedings of the Society for Computation in Linguistics*, 102–12.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. "Learning Internal Representations by Error Propagation." In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, edited by D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Volume 1, 318–62. Cambridge, MA: MIT Press.

Schlag, I., and J. Schmidhuber. 2018. "Learning to Reason with Third Order Tensor Products." In *Proceedings of the Advances in Neural Information Processing Systems*, 9981–93.

Schlag, I., P. Smolensky, R. Fernandez, N. Jojic, J. Schmidhuber, and J. Gao. 2019. "Enhancing the Transformer with Explicit Relational Encoding for Math Problem Solving." In *Proceedings of the NeurIPS Workshop on Context and Composition in Biological and Artificial Neural Systems*. arXiv:1910.06611v2.

Sejnowski, T. J. 2018. *The Deep Learning Revolution*. Cambridge, MA: MIT press.

Smolensky, P. 1988. "On the Proper Treatment of Connectionism." *Behavioral and Brain Sciences* 11(1): 1–23.

Smolensky, P. 1990. "Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems." *Artificial Intelligence* 46(1–2): 159–216.

Smolensky, P. 2012. "Symbolic Functions from Neural Computation." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370(1971): 3543–69.

Smolensky, P., and M. Goldrick. 2016. "Gradient Symbolic Representations in Grammar: The Case of French Liaison." *Rutgers Optimality Archive* 1552.

Smolensky, P., M. Goldrick, and D. Mathis. 2014. "Optimization and Quantization in Gradient Symbol Systems: A Framework for Integrating the Continuous and the Discrete in Cognition." *Cognitive Science* 38(6): 1102–38.

Smolensky, P., and G. Legendre. 2006. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. Volume 1: Cognitive Architecture; volume 2: Linguistic and Philosophical Implications*. MIT Press.

Smolensky, P., R. T. McCoy, R. Fernandez, M. Goldrick, and J. Gao. 2022. "Neurocompositional Computing in Human and Machine Intelligence: A Tutorial." Microsoft Technical Report MSR-TR-2022-5. https://www.microsoft.com/en-us/research/publication/neurocompositional-computing-in-human-and-machine-intelligence-a-tutorial/.

Smolensky, P., E. Rosen, and M. Goldrick. 2020. "Learning a Gradient Grammar of French Liaison." In *Proceedings of the Annual Meeting on Phonology*, Volume 8.

Soulos, P., R. T. McCoy, T. Linzen, and P. 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*, 238–54.

Szabó, Z. 2012. "The Case for Compositionality." In *The Oxford Handbook of Compositionality*, edited by W. Hinzen, E. Machery and M. Werning, 64–80. Oxford: Oxford University Press.

Thomas, R. 2020. "How AI is Driving the New Industrial Revolution." *Forbes*.

Turek, M. 2016. "Explainable Artificial Intelligence (XAI)." DARPA. www.darpa.mil/program/explainable-artificial-intelligence.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. "Attention is All You Need." In *Proceedings of the Advances in Neural Information Processing Systems*, 5998–6008.

Wang, B., R. Shin, X. Liu, O. Polozov, and M. Richardson. 2020, July. "RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-To-SQL Parsers." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7567–78. Association for Computational Linguistics.

Weston, J., A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. 2015. "Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks." *arXiv preprint arXiv:1502.05698*.

Zimmermann, E. 2019. "Gradient Symbolic Representations and the Typology of Ghost Segments." In *Proceedings of the Annual Meeting on Phonology*, Volume 7.

AUTHOR BIOGRAPHIES

Paul Smolensky is Krieger-Eisenhower Professor of Cognitive Science at Johns Hopkins University and Partner Researcher in the Deep Learning group at Microsoft Research Redmond. He was a founding member of the Parallel Distributed Processing Research Group at UCSD, which was key to the neural networks revolution of the 1980s, and is the a recipient of the David E. Rumelhart Prize for Outstanding Contributions to the Formal Analysis of Human Cognition.

Tom McCoy recently completed his PhD thesis in the Department of Cognitive Science at Johns Hopkins, and is now a postdoctoral researcher in the Department of Computer Science at Princeton. He studies how people and computational models learn languages and represent linguistic structure.

Roland Fernandez is a Principal Researcher in the Deep Learning group at Microsoft Research. His interests include brain-inspired neural systems of learning and reasoning using systematic and compositional knowledge structure.

Matt Goldrick is a Professor in the Department of Linguistics and (by courtesy) Psychology at Northwestern University. His research draws on behavioral experiments, acoustic analysis, and computational modeling to develop theories of the cognitive and neural mechanisms underlying spoken language in mono- and multilingual speakers.

Jianfeng Gao is a Distinguished Scientist and Vice President of Microsoft. He is the head of the Deep Learning group at Microsoft Research, leading the development of AI systems for natural language processing, Web search, vision language understanding, dialog, and business applications. He is an Affiliate Professor of Computer Science & Engineering at University of Washington, an IEEE Fellow, and a Distinguished Member of ACM.