Bootstrapping Poverty of the Stimulus Through Random Projections

Xavier Hinaut

Bootstrapping Poverty of the Stimulus Through Random Projections

February 2026

Xavier Hinaut

Inria Center of the University of Bordeaux, Labri, IMN, France.

Chapter 1

Bootstrapping Poverty of the Stimulus Through Random Projections

Contents

Language involves several levels of abstraction, from small sound units like phonemes to contextual sentence-level understanding. Large Language Models (LLMs) have shown an impressive ability to predict human brain recordings. For instance, while a subject is listening to a book chapter from Harry Potter, LLMs can predict parts of brain imaging activity (recorded by functional Magnetic Resonance Imaging or Electroencephalography) at the phoneme or word level. These striking results are likely due to their hierarchical architectures and massive training data. Despite these feats, they differ significantly from how our brains work and provide little insight into the brain's language processing. We will see how simple Recurrent Neural Networks like Reservoir Computing can model language acquisition from limited and ambiguous contextual data better than LSTMs.

My work includes modeling language comprehension, language acquisition from a robotic perspective, sensorimotor models, and extended models of Reservoir Computing to model working memory and hierarchical processing. I propose creating a

new generation of neural-based computational models of language processing and production, utilizing biologically plausible learning mechanisms that rely on recurrent neural networks. This approach involves developing novel sensorimotor mechanisms to account for the shaping of action-perception, building hierarchical models from the sensorimotor to the sentence level, and embodying these models in robots.

I aim to model general hierarchical sensorimotor processes; thus, our models are not only relevant to language or vocal learning, but are interesting for a larger set of sensorimotor tasks.

1.1 Introduction and Insights

Let's make an experiment. I call it the "Repeat after me" or "Would I be able to reproduce these sounds?" challenge. Open the radio or TV on a random channel, listen to the first sentence you hear, wait five seconds, and try to repeat it entirely. You probably managed to repeat nearly the same sentence, at least it had the same meaning. Now switch to a channel speaking in a totally unkown language to you, and do the same. You will probably need to make an arbitrary guess of when a sentence begins and ends. In the end, you may be able to imitate a few syllables, but not much more... Why is it so difficult? Because you already struggle to imitate all the different syllables separately and cannot represent them clearly in your mind. Moreover, you are not able to properly distinguish the different words in the sentence, thus you have no chance to access a glimpse of the sentence' meaning. Unless you help yourself by guessing part of it from the context of the TV broadcast. After this unsuccessful trial, you take a walk outside. You listen to the birds singing, and then you think again about this challenge: "Would I be able to reproduce these sounds"? You can try to wisle a bit, but you know you will need quite some time to learn it (if you ever manage to do so).

Maybe, this is probably not very friendly to start with an impossible challenge. Let me comfort you for a moment... after all, your brain remains significantly more capable than present-day Artificial Intelligence (AI) while handling such challengin tasks. Indeed, your brain is able to catch a lot of information *on the fly*! This may not sound impressive at start, but most wellknown AI systems today (e.g. ChatGPT, Gemini, ...) can't do it. As you can see on Figure 1.1, your brain has to first process sounds into phonemes, into words, into sentences, and so on, while understanding the context in the utterance. It's extraordinary to pull this off in real time!

Why is it difficult for AI? First, it is still extremely hard for AI systems to do these kind of operations while adapting to the current context. When using an AI to transcribe a meeting about a particular technical topic, you'll notice that many words (more or less similar in sound, but quite different in meaning) are being recognized instead of the ones you actually said. This is because speech is often ambiguous without context, but that doesn't matter for our brains because we are able to cope with it most of the time. In case we didn't understood one word we

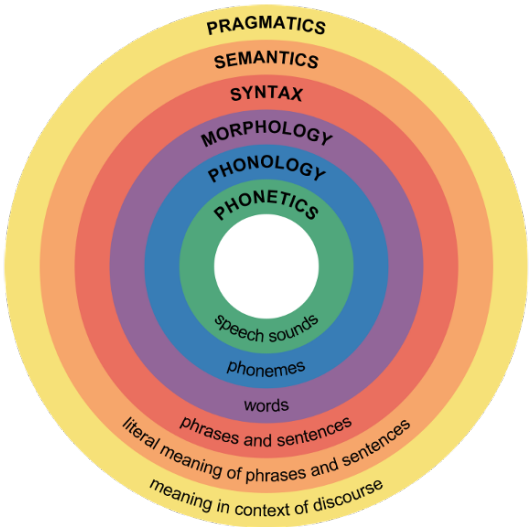

Figure 1.1: Abstraction levels that your brain needs to go through to understand an utterance (starting from the center). Image from Wikimedia.

are able to interrupt the speaker to make it repeat the word we didn't understood quickly to let him continue. Second, understanding a sentence before it ends and interrupting on the fly is also very hard for AIs, mostly because they are made to build full parse of sentences or are trained on well formed written texts[1]. More generally, most AI systems are not built to be able to answer partially at *any time*[2]: they often need to wait until the end of an utterance to start process and generate the adequate answer. This is a fundamental difference with how your brain is built: if it detects an imminent danger, your brain is able to activate your muscles (to *fight, flight, or freeze*) in few hundreads milliseconds even before you may be conscious of what is exactly going on. Hence, *on the fly* is the default mode of our brains. AI machines doesn't process and produce language as we do, because they are not built to *fight, flight, or freeze*.

Another key reason why AI and brains differ is the amount of data used for learning. Nowadays, the most impressive quantity of data is used for Large Language Models (LLM): e.g. Llama 3 was trained 15 trillion tokens (15 followed by twelve zeros) [Grattafiori *et al.* 2024]. A token could be a word, a part of word or sometimes a character: there are the input and output elementary units that LLM process and produce. In average it corresponds to 0.75 of a word [OpenAI 2026]. Assuming a human is reading 250 words a minute in average, one would need more than 100.000 years to read these 15 trillions tokens, not counting the essential breaks

[1]Some AI labs like Kyutai (Paris, France) working specifically on these challenges are making great progress in the last years, but they don't take inspiration from the brain to do so.

[2]In computer science, an anytime algorithm (also reffered as anytime process) is an algorithm designed to return a valid solution to a problem even if it's interrupted before it finishes running. Most algorithms in AI are not built that way.

a human needs. These numbers seem inconceivable to the human mind. On the contrary, children are able to speak fluently within just a few years—despite being exposed to far less data than LLMs—and they acquire a deeper, more intuitive understanding of the real world. The aim of this chapter is to explore potential cognitive and developmental models that could help explain the underlying mechanisms of this rapid acquisition. This remarkable efficiency in learning remains poorly understood. However, one of the answers may lie in another key difference between AI and human babies: unlike AI, children learn in a developmental way, progressing through distinct stages [Thelen & Smith 1994] (see Fig. 1.2). Some early models had been already developed [Oudeyer *et al.* 2007], but in Machine Learning (ML) in general such learning by successive stages is reffered to as *curriculum learning*. In practice, curriculum learning is often implemented as a scheduling of training samples from easy to difficult [Bengio *et al.* 2009]. This is markedly different from developmental learning in children, which involves changes in cognitive and perceptual abilities, and in internal representations, rather than only changes in the order of presented data. In children, each developmental stage builds on previously acquired abilities: each new stage is *bootstrapped* from a set of prerequisite skills. This bootstrapping mechanism is crucial because it strongly constrains the space and actions that must be explored at each stage, thereby drastically reducing the effective learning complexity. This progressive accumulation of competencies provides a powerful source of sample efficiency and explains how complex behaviors and representations can emerge from limited experience. In contrast, most current AI/ML systems lack such explicit mechanisms for developmental bootstrapping, and therefore cannot leverage previously acquired internal structures as prerequisites for the acquisition of more advanced skills.

We are not yet done with the differences between our brains and AIs. The fact that LLMs require so much data can be seen as a consequence of the remaining differences. First, they learn by predicting the next token (∼0.75 word) in billions of sentences. They started to generate qualitative enough language (that enable human-level discussions) only after they were trained on such amount of data. This is because they need to store huge amount of statistics on language to be able to generate these qualitative interactions that seem natural to humans. But why storing complex statistics? Because, when answering, to select the next token to generate, they need to produce probabilities for all possible tokens[3]. In other words, in order to produce natural sentences with the contrain of generating it token by token – because that's how they were trained – they don't have the choice but to very precisely imitate the statistics of our language. Of course, humans produce also sentences by articulated chunks of phonemes, but they have a global idea of the sentence they want to produce instead of computing statistical relations at the

[3]More precisely, the selection of the next token is done through a stochastic (i.e. random) process, which favor most probable tokens, but doesn't take always the most probable, otherwise the sentences would not look natural, and the LLMs would be often stuck in loops such as "the cat that sees the mouse eaten by the cat that sees the mouse eaten ...". The full process is repeated once again for each token.

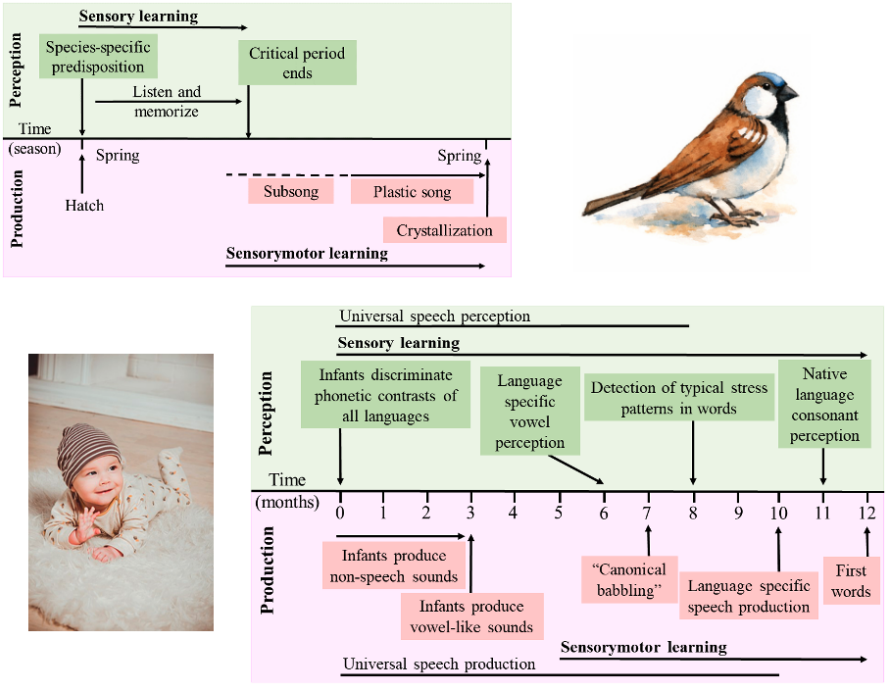

Figure 1.2: There are common vocal developmental stages for both songbirds and humans, here shown for the first year. In green we see the perceptual stages and in pink the production stages. Both songbirds and humans have critical periods for sensory learning, and a babbling phases (called subsong for songbirds. This means there are probably common mechanisms that lead to these common developmental stages. Modified from our review on vocal sensorimotor learning models [Pagliarini *et al.* 2021]. Inspired from [Kuhl 2004] and [Doupe & Kuhl 1999]. Baby image: Wikimedia. Bird image: generated with AI.

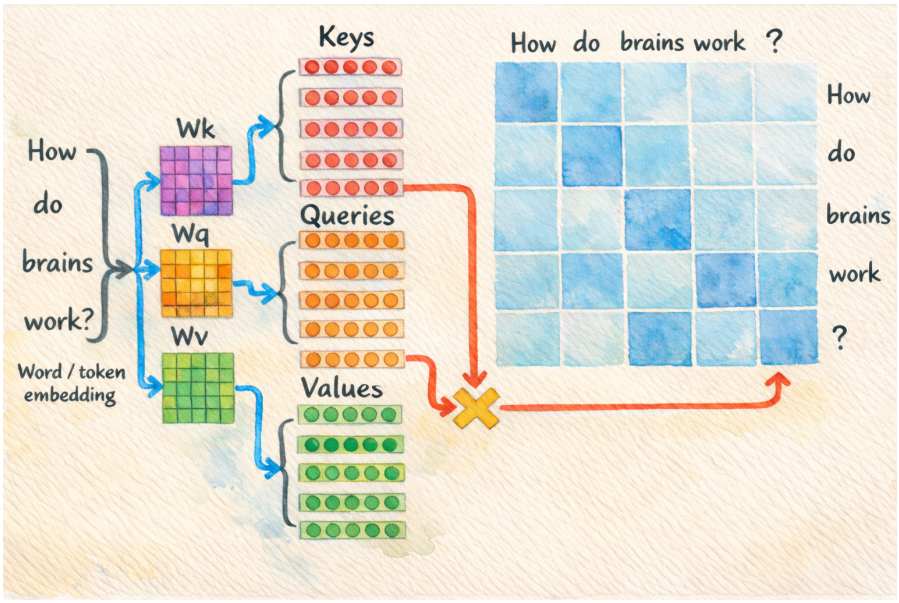

Figure 1.3: Attention mechanism in Transformers/LLMs needs to process all words at all times. For each word in a sentence (and more generally in its context window) it needs to compute how this word is "attending" to all other words. This is represented by a matrix, meaning that for n words in the context window, $n * n = n^2$ operations will be performed at each time. Image generated with AI.

sub-word level.

Secondly, our brain process one word after another, whereas LLMs process all tokens of a given discussion at once, and they have to do it again for each token they produce! This is due to the *attention mechanism* [Vaswani *et al.* 2017] (see Fig. 1.3): in order to predict the next token, they have to read all previous tokens in their *context window*... all the time. This sounds technical but it has a huge impact. For instance, if you ask for a summary of what Hermion did in one of Harry Potter book chapter (giving a copy of this chapter book along with your question), a LLM will need to read all the chapter again and again for each token it is generating! In no way this is similar to the way our brain is working, and this represents a huge amount of computations and energy consumption. This is one of the reason why our brain is way more efficient than LLMs and only consumes about 20 watts (similar to two energy-saving light bulbs).

To put it in perspectives of theories in cognitive sciences on the predictive brain, I do not believe the brain is learning by predicting the next input as LLMs do. First because, in order to process *on the fly* the brain is probably doing something else[4] [Christiansen & Chater 2016, Christiansen *et al.* 2016] than just trying to precisely predict the next input. Secondly, the fact that LLMs require massive amounts of data suggests that next token prediction is not an efficient learning strategy. By contrast brains likely relies on fundamentally more efficient learning mechanism since they learn from very limited data in comparison.

Building on these ideas, in this chapter I introduce models of human language acquisition and processing by focusing on important brain features that we have discussed above. Our models have several aims: (1) process inputs *on the fly*, (2) learn with as little data as possible, (3) use biologically plausible algorithms, (4) learn in interaction with the world in order to learn within a sensori-motor loop enabling an agent to imitate and correct itself autonomously. An advantage of such features is that they ease the implementation of such models in robots in order to perform experiments on language grounding and embodiement.

[4]Christiansen Chater propose that the brain is in the *Now or Never Bottleneck* problem [Christiansen & Chater 2016] when processing a stimulus (e.g. an utterance): it is forced to extract the necessary information as soon as possible, otherwise the information will be lost. Thus, the rich perceptual input needs to be recoded as it arrives in order to capture the key elements of the sensory information [Christiansen *et al.* 2016]. These compressed (or *chunked*) representations are abstractions of inputs (filtering out the details) rather than predictions encoding all the fluctuations of fast incoming inputs. Memory limitations also apply to these recoded representations; hence the brain needs to chunk the compressed representations into *"multiple levels of representation of increasing abstraction in perception, and decreasing levels of abstraction in action"* [Christiansen *et al.* 2016]. Therefore, each sequence of chunks at one level will be encoded as a single chunk to a higher level. In summary, they suggest the brain must implement a hierarchical "Chunk and Pass" mechanism to solve the "Now or Never Bottleneck" problem.

1.2 Need for bio-plausible models of language processing and acquisition

In this section, we argue that current language models, despite their performance, fail to capture key biological principles of language processing and acquisition, motivating the need for grounded, hierarchical, and action–perception–based neurocomputational architectures.

For less than a decade, deep learning networks have been shifting the limits of benchmark performance in natural language processing (NLP) (e.g. [Devlin *et al.* 2018, Brown *et al.* 2020, Achiam *et al.* 2023]), motivating the creation of new types of evaluation frameworks such as the ARC-AGI challenge [Chollet 2019]. These breakthroughs have been enabled by new kinds of representations and mechanisms (e.g. context-dependent word embeddings [Devlin *et al.* 2018], attention [Vaswani *et al.* 2017], etc.), which are integrated into increasingly complex architectures [Jiang *et al.* 2024], and, for some models, combined with reinforcement learning from human feedback (RLHF) [Hejna *et al.* 2023].

However, the brain must parse and learn from incoming stimuli incrementally; it cannot unfold time in the way required by the back-propagation through time (BPTT) algorithm. As a result, we still lack the key neuronal mechanisms needed to properly model the hierarchies of functions involved in language perception and production. Other models of language processing that aim to reproduce brain dynamics, such as Event-Related Potentials (ERPs) [Brouwer & Hoeks 2013] or functional Magnetic Resonance Imaging (fMRI) [Garagnani *et al.* 2008], have been developed. Nevertheless, these models often lack explanatory power regarding the causes of the observed dynamics, that is, what is computed, why it is computed, and for what purpose. We therefore need more biologically plausible learning mechanisms that can also provide causal explanations of the experimental data being modelled.

There is converging evidence that language production and comprehension are not separate processes within a modular mind; rather, they are interwoven, and this interweaving enables individuals to predict both themselves and others [Pickering & Garrod 2013]. The interweaving of action and perception is crucial because it allows an agent (or a baby) to learn from its own actions, for instance by learning the perceptual consequences (e.g. the sounds heard) of its own behaviours (e.g. vocal productions) during babbling [Schwartz *et al.* 2012]. In this way, the agent learns in a self-supervised manner rather than relying solely on supervised learning, which, by contrast, presupposes non-biological teacher signals explicitly designed by the modeller. However, we still lack neuronal models that causally explain how these perceptuo-motor units emerge and are shaped over development. More specifically, the existence of sensorimotor (i.e. mirror) neurons at abstract representational levels (often referred to as action-perception circuits [Pulvermüller & Fadiga 2010]), together with the perceptuo-motor shaping of sensorimotor gestures [Schwartz *et al.* 2012], suggests the presence of similar action-

perception mechanisms implemented at different hierarchical levels. How can we move towards such hierarchical architectures grounded in action-perception mechanisms?

Taken together, these observations suggest that biologically plausible models of language must integrate hierarchical action–perception mechanisms with grounded semantic learning. More generally, this concern can be framed within the Symbol Grounding Problem, originally formulated by Harnad [Harnad 1990], which highlights the requirement that systems manipulating symbols must ultimately "anchor" their meanings in "raw" perceptual experience. Importantly, a language processing model requires a mechanism to acquire the semantics of (symbolic) perceptuo-motor gestures as well as of more abstract representations; otherwise, it would be limited to morphosyntactic and prosodic features of language. These symbolic gesturesneed to be grounded in the mental concepts they represent, namely the signified. Several theories and robotic experiments provide examples of how symbols may be grounded or how they may emerge [Taniguchi *et al.* 2016]. These are important conceptual questions for AI (Artificial Intelligence) in robotics, and, from a neuroscientific perspective, for understanding how the brain solves these problems. However, current neurocomputational models that aim to explain brain processes generally lack such grounding. Robotics can play an important role in this respect, as it enables the grounding of semantics through direct interaction with the physical world and with humans. To this end, mechanisms that start from raw sensory inputs and raw motor commands are required, allowing plausible representations to emerge through development rather than relying on arbitrary or engineered ones (e.g. word embeddings).

More recently, the field has also *delved* [Juzek & Ward 2024][5] into the development of large language models (LLMs), which provide new methodological tools. For instance, models such as *Centaur* [Binz *et al.* 2024] enable the modelling of human behaviour in cognitive science experiments, while several toolboxes allow the analysis, or controlled activation, of complex concepts in LLMs [Templeton *et al.* 2024, Paulo *et al.* 2024, Lindsey *et al.* 2025]. In parallel, we are actively exploring architectures that aim to bridge the gap between neurocomputational models and LLMs, for example by proposing recurrent–transformer hybrids that require less training data [Bendi-Ouis & Hinaut 2024, Bendi-Ouis & Hinaut 2026]. However, this line of research is still under development and therefore falls outside the scope of the present chapter. In future work, we aim to integrate this approach with the one presented in Section 1.4.

1.3 Random projections are cheap and yet powerfull

In the previous sections, we have shown that modern AI systems, such as large language models, do not provide reliable models of human learning or brain processes.

[5]The use of the verb *delved* here deliberately echoes the title of [Juzek & Ward 2024]: "Why Does ChatGPT 'Delve' So Much? Exploring the Sources of Lexical Overrepresentation in Large Language Models".

In light of the challenges discussed above, you may be wondering which architectures could bring us closer to possible solutions. Although a definitive solution is still out of reach, there are promising directions to explore. We believe that one particularly promising direction lies in a specific class of neural architectures known as *Reservoir Computing*. Together with colleagues, we have already conducted several computational experiments that demonstrate the strong potential of this approach, while remaining well aligned with the challenges discussed above.

In short, Reservoir Computing is a paradigm to train Recurrent Neural Networks (RNN) without training all connections. RNNs are used to process time series or sequences. They have cycles (i.e. recurrent connections) between neurons inside their internal layer, which enable them to store contextual information and to have some memory of past inputs. Conceptually, they can be contrasted to Feed-Forward Networks (FNN), which do not have such cycles and therefore no recurrent internal state. For instance, LLMs are based on complex and huge FNN architectures (e.g. about 675 billion parameters[6] for Mistral 3 [Mistral AI 2025]). As a consequence, they do not rely on recurrent connections to store past information; instead, they handle context through attention over previously processed tokens (and their cached internal representations), rather than through cycles as in RNNs.

Before explaining how the reservoir computing paradigm emerged, let me first share a striking experience from the first edition of the Hack1robo hackathon[7] that we organised in 2023. At the beginning of the weekend, one team set themselves the challenge of generating melodies using a completely random RNN. That is, without training it at all. In less than two days, using the ReservoirPy library [Trouvain *et al.* 2026], they managed to build random RNNs capable of producing what they called "proto-melodies". They developed a graphical interface and connected it to a MIDI keyboard, which allowed users to control several meta-parameters[8] that altered the dynamics of the random network and, as a result, transformed the generated melodies. One of these meta-parameters, called the spectral radius, makes it possible to control the network dynamics, and one could clearly hear the effect of regular versus more chaotic regimes on the melodies. In this way, they had created a kind of creative DJ interface based solely on the random projections implemented in the network itself. This experience made a strong impression on me, because it was the first time I could directly see and "hear" the activity of a reservoir in real time. Since their final presentation was also particularly entertaining, they won the first prize of the hackathon. A couple of year later, they continued this project by creating a startup. In a nutshell, random networks can generate fascinating dynamics, and reservoir computing exploits the natural dynamics of random networks.

[6] A parameter can be approximated as a weighted connection between two neurons.

[7] A contraction of *marathon* and *hack*; a hackathon is an event, usually lasting a few days, during which teams work on challenging projects.

[8] Meta-parameters modify several parameters at the same time.

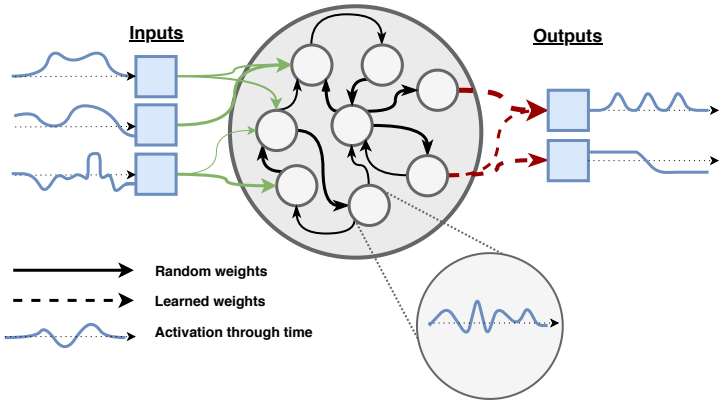

Figure 1.4: The Reservoir Computing (RC) paradigm to train Recurrent Neural Networks (RNNs). Input and recurrent weights are fixed and random while output weights are trained. Time series provided as input generate a non-linear combination of dynamics inside the *reservoir* (the recurrent part in the middle). The output layer linearly *reads out* some of these dynamical combinations: it makes a weighted sum of reservoir states. Image from [Juven & Hinaut 2020].

1.3.1 Reservoir Computing insights

To begin, let us introduce reservoir computing through a simple example shown in Figure 1.4. Inputs are fed into a recurrent layer of neurons, called the *reservoir*. The reservoir states combine the incoming inputs with their previous states through recurrent connections. These reservoir states are then transmitted to an output layer, called the *read-out*. The input and recurrent connections are typically fixed and randomly generated. In practice, only the connections of the output layer are trained, in a supervised manner, using a variant of linear regression. We will later describe in more detail how this mechanism works. For now, let us first turn to the context in which this approach emerged.

An intuition for why this paradigm is called Reservoir Computing is as follows. The terms "reservoir" for the recurrent layer and "read-out" for the output layer originate from the fact that many combinations of the inputs are generated within the recurrent layer (through random projections). The "reservoir" is thus literally a reservoir of nonlinear computations (hence "reservoir computing"). From this reservoir, one linearly decodes (i.e., "reads out") the combinations that are useful for the task to be solved. Reservoirs can be implemented on a wide range of physical substrates [Tanaka *et al.* 2019] (e.g., electronic, photonic, mechanical, chemical and even with bacteria [Ahavi *et al.* 2026a, Ahavi *et al.* 2026b]).

A convenient way to understand how reservoir computing operates is to view it as a temporal Support Vector Machine (SVM) [Verstraeten 2009]. SVMs work as follows: as illustrated in Figure 1.5, suppose that you want to separate blue dots from red dots, but that in the original two-dimensional space they cannot be separated

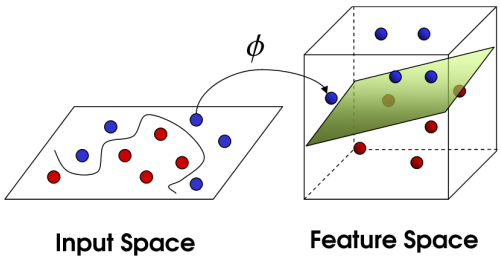

Figure 1.5: Projection of inputs in a higher dimensional space. Image from Wikimedia.

by a line. With a SVM [Vapnik 1999], these inputs (i.e., the dots) are projected into a higher-dimensional space. In this high-dimensional space, it becomes possible to find a hyperplane (the analogue of a *line* in higher dimensions) that separates the blue dots from the red dots. In practice, this corresponds to learning a linear decision function in the projected space. This principle corresponds to the so-called *kernel trick* in SVMs. Different kernel functions can be used with SVMs; in reservoir computing, this kernel is random.

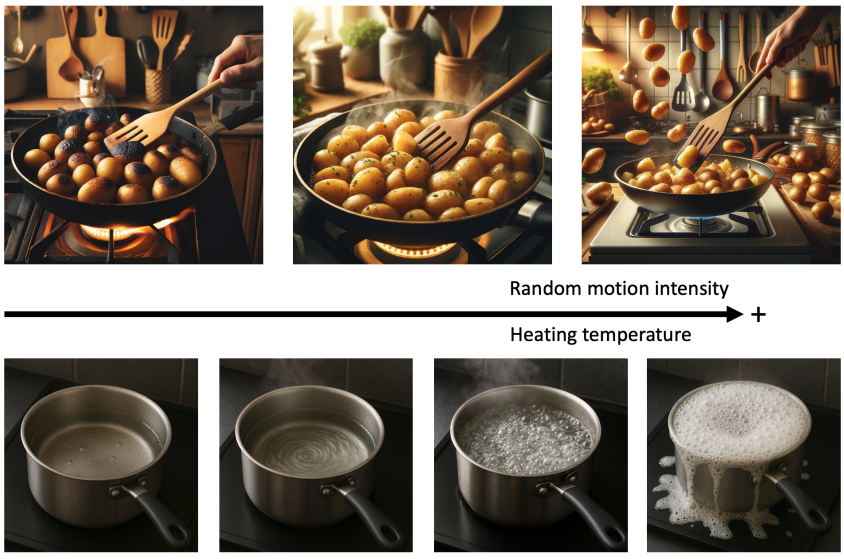

Figure 1.6: Intuition about the effect of the spectral radius hyperparameter. Increasing the spectral radius in a reservoir is loosely analogous to increasing the strength of the internal dynamics of the reservoir. For instance, when the temperature of heated water is increased, the water dynamics progressively intensify, until a point at which the water spills out of the pot. In general, the optimal spectral radius lies at the *edge of chaos* (or *edge of stability*, i.e., between stable and chaotic regimes). Images generated with AI.

But you may wonder why random projections should work at all. Let us consider an example from everyday life. Imagine that you are cooking potatoes in a frying pan with a wooden spatula. Making a few random stirring gestures from time to time is sufficient to cook them properly, as illustrated in the upper middle image of Figure 1.6. However, if one day you are too focused on the notifications on your phone, you will probably make movements that are too shallow or too infrequent, and the potatoes will burn (upper left). On the contrary, on days when you have drunk too much coffee, you will probably make overly vigorous stirring motions and throw some potatoes out of the frying pan (upper right). These different stirring dynamics are comparable to the effect of the spectral radius hyperparameter on the dynamics of the reservoir. In Figure 1.7, we can see the effect of different spectral radius on the dynamics of the reservoir.

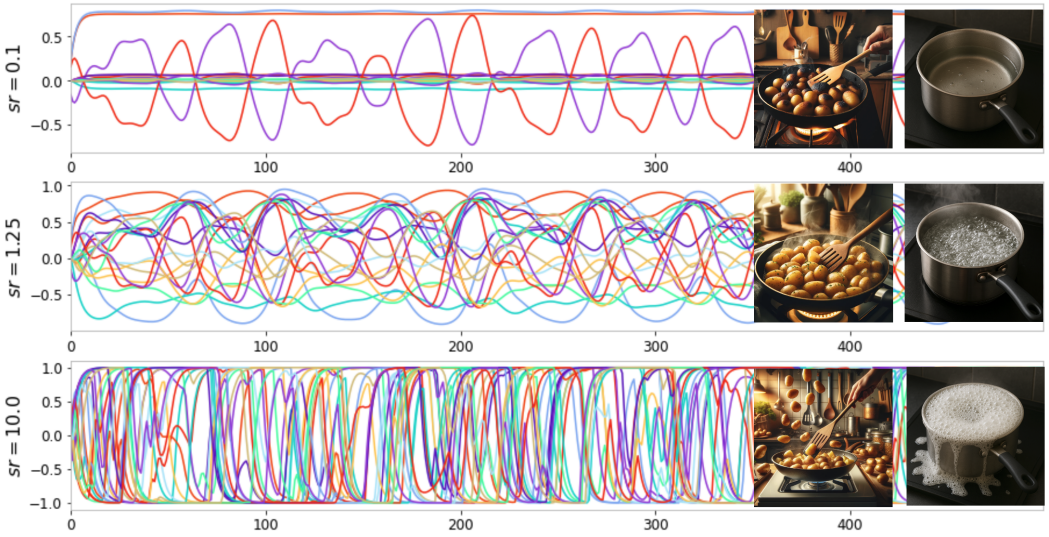

Figure 1.7: Comparison of the effect of various spectral radius (sr) values on the dynamics of the reservoir. From a reservoir of 100 neurons, a subset of 20 are plotted. Each colored curve represents the activity of one of these neurons across time. (top) regular dynamics for $sr = 0.1$. (middle) edge of chaos dynamics. (bottom) chaotic dynamics.

1.3.2 Context and Biology of Reservoir Computing

Reservoir Computing has emerged on several occasions in the literature. It is often stated that RC emerged independently in 1995 in two laboratories from the computational neuroscience community [Buonomano & Merzenich 1995, Dominey 1995]. However, it can be argued that related forms had already appeared earlier (see the references collected by Herbert Jaeger on Scholarpedia[9] [Jaeger 2007]). RC

[9] http://www.scholarpedia.org/article/Echo_state_network

therefore appeared only a few years after the well-known **SRN!** (**SRN!**) (Simple Recurrent Network) introduced by Elman in 1990. RC re-emerged in the early 2000s with the Echo State Network (ESN) proposed by Jaeger [Jaeger 2001] and with the Liquid State Machine (LSM) introduced by Wolfgang Maass and colleagues [Maass *et al.* 2002]. Subsequently, an RC community began to take shape: the machine learning community focused more on ESNs, whereas the computational neuroscience community mainly emphasized LSMs[10]. This development was probably reinforced by the strong performance reported by Jaeger on chaotic time series prediction [Jaeger & Haas 2004].

Reservoirs are of interest for neuroscience because they can be seen as "a canonical computational unit" [Haeusler & Maass 2007]; they can also be used to model "a cortical column", which computational neuroscientists often consider as a generic unit of computation. Since 1995 [Dominey 1995], my former PhD supervisor, Peter Dominey, has used this framework to model the cortico-basal network, with the reservoir playing the role of the (prefrontal) cortex and the output layer playing the role of the striatum (the input of the basal ganglia from the cortex). Dominey [Dominey *et al.* 1995] showed that, even with random networks (which were not yet called reservoirs), it was possible to observe neuronal activation patterns similar to those reported in studies of sequence processing in the monkey prefrontal cortex [Barone & Joseph 1989]. RC developed much more rapidly in the machine learning community from the 2000s onward, but in the 2010s it also became increasingly popular among experimental neuroscientists. Neuroscientists started to adopt the idea of high-dimensional and nonlinear representations that can be decoded by a linear classifier. This provided a new way to interpret electrophysiological recordings from monkeys [Machens *et al.* 2010, Rigotti *et al.* 2013, Enel *et al.* 2016]: the goal was no longer to identify specific sequential patterns in neural activity (as in [Barone & Joseph 1989]), but rather to determine, through linear decoding, whether particular information was present in the neural population activity.

1.4 Towards Modelling Cross Situational Learning

Now that we have gained insights and intuitions about Reservoir Computing, we investigate how this framework can help us understand how children acquire language from co-occurrences and noisy supervision. As discussed earlier, another key challenge is to explain how the brain can rapidly learn a language from very limited data, in contrast with Large Language Models.

1.4.1 Introduction

Infants rapidly acquire word-referent mappings by exploiting statistical regularities across situations [Yu & Smith 2007, Smith & Yu 2008]. A large body

[10] Even if ESNs or equivalent (rate-coded RNNs) are also often used in computational neuroscience nowadays.

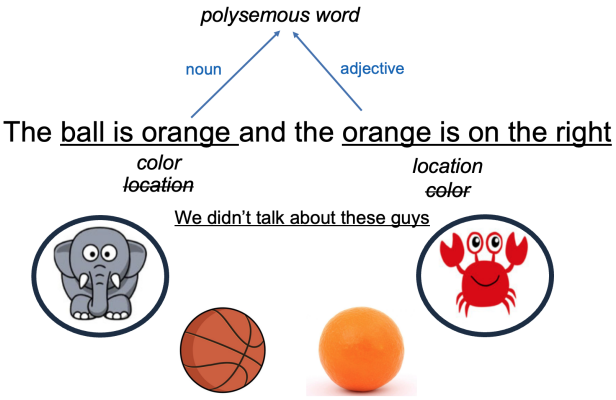

Figure 1.8: Full sentence Cross-Situational Learning paradigm. From a sentence, the model must identify the correct objects and features. It learns the appropriate mapping between words and their meanings solely from the co-occurrences of words within the sentence. The scene is rich in objects and features, yet each sentence describes only a subset of this scene. Consequently, the mapping cannot be established in a single exposure; it must instead be acquired across multiple situations. In particular, certain words such as "orange" cannot be reliably learned from a single exposure, even when presented in isolation. Because "orange" is polysemous, only the sentential context enables the model to disambiguate its meaning.

of experimental and modeling work in language acquisition aims to understand how children learn language by observing their environment and interacting with others [Chen & Mooney 2008, Tomasello 2009, Thomason *et al.* 2018, Vanzo *et al.* 2020, Roembke *et al.* 2023]. Even before one year of age, infants are able to segment words from continuous speech using statistical learning mechanisms [Saffran *et al.* 1996, Yu & Smith 2007]. However, mapping words to meanings remains highly ambiguous, as utterances typically co-occur with rich scenes containing multiple potential referents. For example, when hearing the phrase "The blue glass is on the left.", a learner must determine how words relate to visual concepts, such as distinguishing a blue glass from a green one (see Fig. 1.8).

To resolve such ambiguity, children rely on repeated exposure to words across different contexts, progressively accumulating evidence about their meanings. This process is commonly referred to as cross-situational learning (CSL) [Taniguchi *et al.* 2017, Juven & Hinaut 2020, Warren *et al.* 2020, Dinh & Hinaut 2020, Variengien & Hinaut 2020, Roembke *et al.* 2023]. In the CSL paradigm, learners are exposed to multiple words and multiple potential referents at each learning episode, making it impossible to infer correct mappings from a single observation. Instead, word meanings emerge gradually from statistical regularities across situations.

In parallel, research on grounded language learning [Cangelosi 2010] has investigated how artificial agents and robots can associate lin-

guistic symbols with objects, attributes, and actions in the physical world [Chen & Mooney 2011, Matuszek *et al.* 2013, Tellex *et al.* 2011, Thomason *et al.* 2016, Beinborn *et al.* 2018]. Several computational models have explored CSL by tracking co-occurrences between word forms and perceptual referents to account for early word learning [Taniguchi *et al.* 2017, Roesler *et al.* 2018]. These approaches have provided valuable insights into how meaning can emerge from ambiguous input, often under weak or noisy supervision.

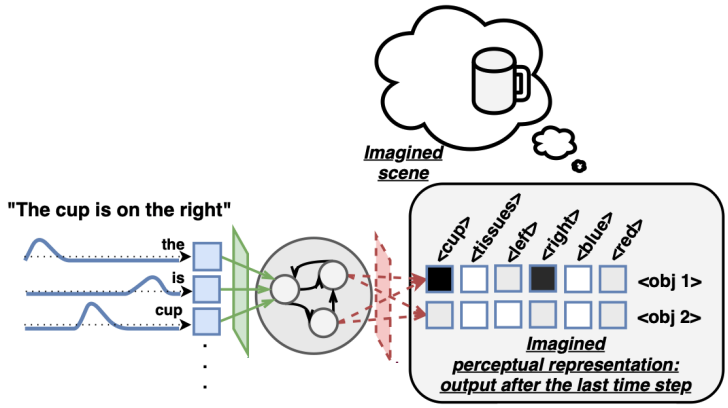

Figure 1.9: CSL reservoir model after training. The reservoir processes the sentence word by word and correctly activates the outputs (in black) corresponding to the scene, i.e. the features *cup* and *right* of object 1, and no feature for the other object. Some outputs (in gray) may exhibit subthreshold activity but are disregarded in the final interpretation of the sentence. Image modified from [Juven & Hinaut 2020].

At the same time, many robotic implementations of language grounding focus on isolated words or on short, command-like utterances, which differ from the richer and more continuous nature of everyday language. This motivates the study of alternative computational frameworks that can learn from full sentences under cross-situational ambiguity, using limited data and simple neural architectures, in conditions closer to those faced by human learners.

1.4.2 Online Learning at Sentence Level

To address the challenges outlined above, we conducted modelling experiments [Juven & Hinaut 2020, Variengien & Hinaut 2020, Oota *et al.* 2022] using a learning agent that can represent either a child or a robot (see Figure 1.10). This shared formulation allows the same model to be studied both in abstract simulations and in experiments with a physical robot. Such a setup is particularly valuable for investigating language learning in embodied and interactive contexts, where linguistic symbols must be grounded in perception and action. Moreover, it ensures continuity with our previous work on robot language grounding. [Hinaut *et al.* 2014, Hinaut *et al.* 2015a, Hinaut *et al.* 2015b, Twiefel *et al.* 2016a, Hinaut *et al.* 2016,

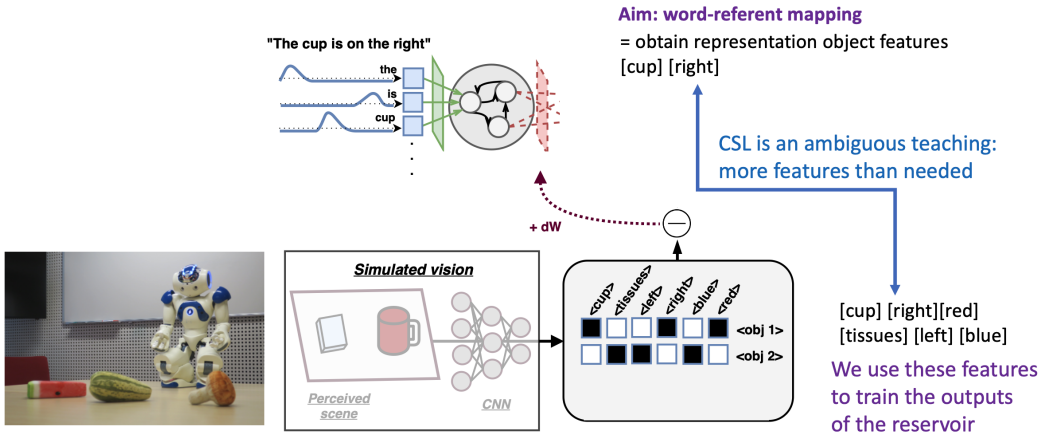

Figure 1.10: Cross-situational learning framework during model training. From a sentence, the model must learn the correct correspondence between words and their referents (objects, features, or locations). This mapping is inferred solely from the co-occurrences of words within the context of the sentence. Words are never presented in isolation to facilitate their association with the corresponding referents, and the model does not have prior knowledge of the meaning of any word at the beginning of training. (1 - middle bottom) The visual scene observed by the agent is transformed into a simple one-hot encoding (i.e. localist representation): each row corresponds to an object, and each column to one of its features. (2 - middle top) This symbolic representation of features (i.e. the *teaching signal*) is used to train the readout layer of the reservoir. (3 - right) As there are generally more features in the teaching signal than words in the sentence, the resulting supervision is ambiguous. Image adapted from [Juven & Hinaut 2020].

Twiefel *et al.* 2016b].

In Figure 1.8, we present the Cross-Situational Learning (CSL) task that the model is trained to perform. Instead of focusing on isolated word-referent mappings, as in several previous modelling approaches, we consider learning directly from full sentences. In this setting, the model must infer the correspondence between words and their referents (objects, features, or locations) solely from co-occurrence statistics within sentential contexts. Words are never presented in isolation, and the model starts with no prior knowledge of their meanings.

In order to control for the complexity of the corpus and include some particular features (explained below), we are randomly generating the object scenes and corresponding possible sentences. The former can describe 1 or 2 object. The grammar is described in Figure 1.12 for the default dataset of size 4 objects. The task is challenging for several reasons: (1) each sentence describes only a subset of the visual scene, which may contain multiple objects and features; (2) different words may share the same referent (e.g., "middle" and "center"); (3) presence of the polysemous word "orange" which can be disambiguated (i.e. *noun* or *adjective*) only from the context; (4) the models are trained with only 1000 sentences, which is small given the large number of possible combinations of objects, colors, and spatial relations ($> 40,000$).

We trained the neural networks as follows. We used reservoirs with 1000 recurrent units. They were trained using the RLS FORCE learning algorithm [Sussillo & Abbott 2009], using the ReservoirPy library[11] [Trouvain *et al.* 2020, Trouvain *et al.* 2026]. In order to have a point of comparison, we another commonly used RNN: a LSTM (Long-Short Term Memory) network [Hochreiter & Schmidhuber 1997]. LSTMs are still considered state-of-the-art for various machine learning tasks given its relative simplicity compared to Transformers, and probably one of the best alternative to Reservoir Computing for small datasets[12]. In order to compare reservoirs with LSTM with approximately the same number of trainable units, we took LSTM of recurrent size 20 units[13]. In order to picture the lack of generalization of LSTMs with such small training dataset, we also compared to LSTMs with 40 and 80 units. The hyperparameters of reservoirs and LSTMs were optimized on the default corpus – with an object vocabulary size of 4. Note that training reservoirs took much less time computation time in comparison to LSTMs (x10 times faster for the 4-object experiment), which among other things needed several epochs of training (from 15 to 50 depending on LSTM size); the reservoirs were only seeing the dataset once (i.e. 1 epoch). For more details please refer to [Variengien & Hinaut 2020].

In Figure 1.9, we can see how the CSL reservoir model works after being trained (on the default 4-object dataset). It processes a 1-object sentence word by word

[11] ReservoirPy repository: `https://github.com/reservoirpy/reservoirpy`

[12] Gated Recurrent Unit (GRU) network could also be considered as a comparable alternative.

[13] In LSTMs all the connections between units are trained: e.g. just in the recurrent layer there are $20 * 20 = 400$ trainable parameters. Whereas, in reservoir computing only the output layer is trained.

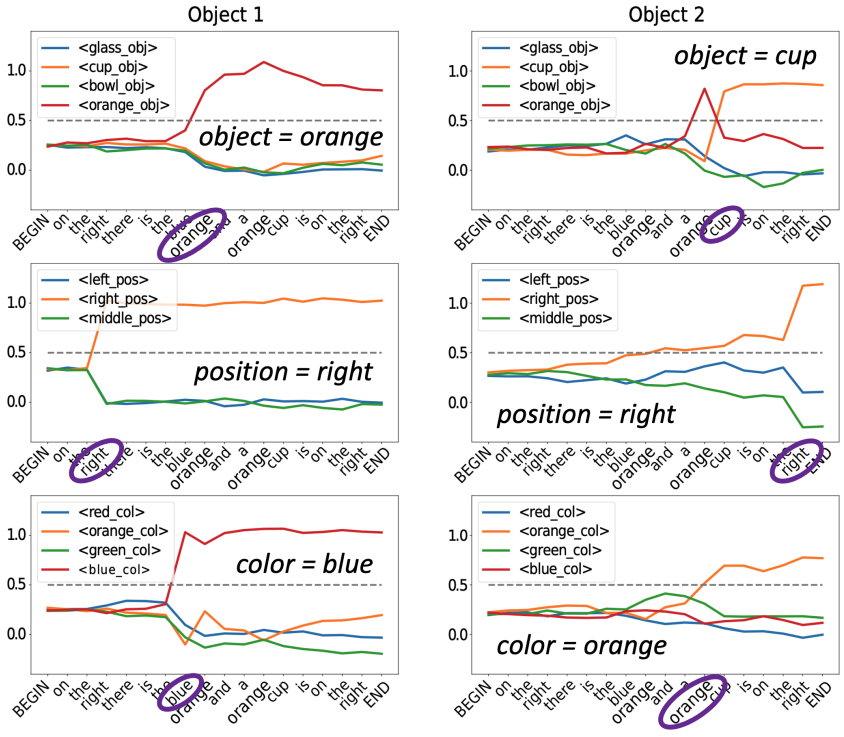

Figure 1.11: Reservoir outputs obtained after training for a 2-object sentence. The reservoir processed the following sentence word by word: "On the right there is the blue orange and a[n] orange cup is on the right" and correctly activates the corresponding outputs. Some outputs may exhibit subthreshold activity but are disregarded in the final interpretation of the sentence. In particular, we can see that the correct outputs are progressively found shortly after the onset of each significant word. This is an emerging property not an engineered feature due to specific training, as we didn't give teacher signal on particular word onsets. Moreover, (top left) we can see a particularly interesting emerging property due to the polysemous nature of the word *orange*: for the second object, first recognized as a noun and just after as an adjective. Image modified from [Variengien & Hinaut 2020].

OBJ → **cup** | **bowl** | **orange** | **glass**
COL → **red** | **orange** |**blue**| **green**
POS → **left** | **middle** | **right**
THE → **a** | **the**
THIS → (**this** | **that**)
SENTENCE-1-OBJ → THIS **is** THE (COL)? OBJ
| THE OBJ (**on the** POS)? **is** COL
| THE (COL)? OBJ **is on the** POS
| **there is** THE (COL)? OBJ **on the** POS
| **on the** POS (**there**)? **is** THE (COL)? OBJ
SENTENCE → SENTENCE-1-OBJ
| SENTENCE-1-OBJ **and** SENTENCE-1-OBJ

Figure 1.12: Grammar used to generate the default corpus with vocabulary size of 4 objects. The grammar can generates sentences describing one or two objects depending on the scenario. Note that the polysemous word "orange" is used both as a noun and as an adjective. The total number of different sentences that could be generated is 473344 ($= 688^2$). We randomly sample 1000 sentences from it: 300 *1-object* sentences and 700 *2-object* sentences. For extended experiments, the corpus was extended by varying the number of objects from 4 to 50.

and correctly activates the outputs corresponding to the scene, i.e. some features for object 1, and no feature for the object 2. In Figure 1.11, we can see the reservoir outputs obtained after training for a 2-object sentence. The reservoir processed the following sentence word by word: "On the right there is the blue orange and a[n] orange cup is on the right" and correctly activates the corresponding outputs. In particular, we can see that the correct outputs are progressively found shortly after the onset of each significant word. This is an emerging property not an engineered feature due to specific training, as we didn't give teacher signal on particular word onsets. Moreover, we see that the polysemoud word *orange* can be recognized as noun or adjective depending on the context, and that this recognition activity can be updated quickly, showing partial reinterpretation of the sentence.

In order to create different levels of difficulty we generated corpora of different complexity, the easiest containing only 4 different objects and the harder one containing 50 different objects (without changing the training size of 1000 sentences). To evaluate the generalization abilities of the models we created two measures: the *valid error* and the *exact error* which are explained in Figure 1.13. Surprisingly, reservoirs demonstrate robust generalization when increasing the vocabulary size: the error grows slowly compared to an LSTM of fixed size.

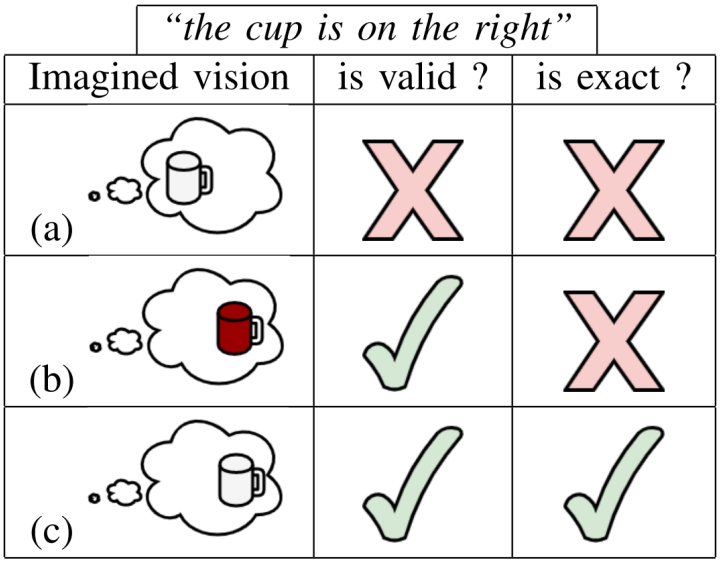

Figure 1.13: Evaluations of different imagined scenes leading to two kinds of error measured. (a) is not valid or exact because the cup is not on the right. (b) is not exact because the sentence does not mention the cup color. Image from [Juven & Hinaut 2020].

1.4.3 Perspectives

Some further work [Variengien & Hinaut 2020] on this generalisation difference capabilities have shown that the internal states of LSTM seem very poor in terms of diversity of states compared to reservoirs. LSTMs seem to encode precise information in a fractal way, making extraction and generalisation of such information difficult. On the contrary, the internal states of the reservoir are already rich at start, and don't change during training, which seems to facilitate generalization with few examples. Moreover, we have shown that while changing slightly the encoding of outputs, we manage to make the CSL reservoir model generalise from 1-object sentence exposition only during training, to 2-object sentence during testing [Dinh & Hinaut 2020]. Overall, this makes us think that reservoir computing is an interesting candidate to bootstrap generalization from little data, and thus a good model for developmental processes in general, not only for language.

In subsequent work, we also applied this model on two robot grounding datasets of higher complexity and the reservoir kept its ability to generalise [Oota *et al.* 2022]. Importantly, we have already shown that this reservoir computing approach to language processing is not limited to these particular setup and work in a variety of conditions: (1) words can be coded using various *word embedding* representations [Oota *et al.* 2022], there do not need to be one-hot-encoded; (2) it is not limited to a particular language, as we provided a proof of concept on 15 occidental and asian languages [Hinaut & Twiefel 2019], (3) can be

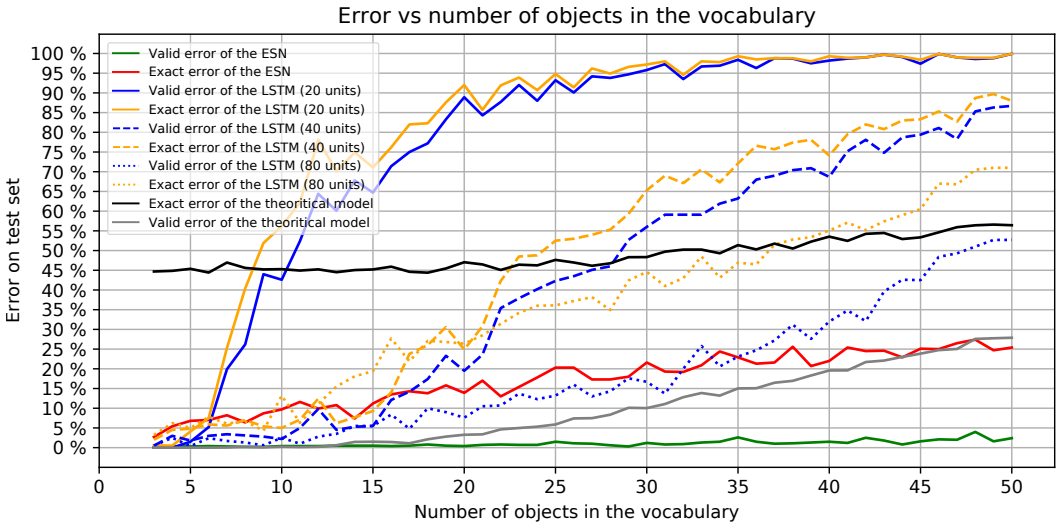

Figure 1.14: Performance comparison of 5 models (1 reservoir + 3 LSTMs + theoretical) for datasets with a vocabulary size (i.e. nr. of objects) from 4 to 50. We can see that the small LSTM (20 units) is not able to keep good performances when the number of objects starts to increase. The medium LSTM (40 units) is able to obtained good performances until 15 objects, but then the error degrades quickly. The bigger LSTM (80 units) limits the increasing of the error compared to the other LSTM, but is suprisingly less performing for ∼15-objects corpora. On the contrary, the reservoir is suprinsingly able to keep an error below the theoretical model and all the LSTMs, and the trend shows that it would stay below theoretical model wathever the vocabulary size. Image from [Variengien & Hinaut 2020].

trained as bilingual model [Hinaut *et al.* 2015a] and even to cope with *code-switched* sentences [Detraz & Hinaut 2019][14] (4) it is not limited at the word level itself, as we tested it at the phoneme level representation and it works as well [Dinh & Hinaut 2020], (5) can use a variety of (output) semantic representations, like predicate [Hinaut & Dominey 2013][15] or simple graph-like structures [Hinaut *et al.* 2016] that are usefull in robotics because they represents commands (*move(ball, left)*); (6) can be integrated in a hierarchical reservoir architecture, where various sequential reservoirs are trained to go from speech to phoneme, to word, to Part-of-Speech, to Semantic Role Labelling [Pedrelli & Hinaut 2020, Pedrelli & Hinaut 2022]: this shows that symbolic recognition can be done at several levels of hierarchy in an online fashion.

Going deeper in the analysis of the model presented in the previous subsection, we made several analyses the internal activation of reservoirs and LSTMs [Variengien & Hinaut 2020]. We showed that LSTMs learn to create relatively sparse representations, in particular internal unit activity change for few words only, as some kind of "mixed selectivity" [Rigotti *et al.* 2013]. On the contrary, reservoir activity showed a highly diverse and distributed activity. In a context where one agent has to learn a task from scratch with a small dataset, it seems that the reservoir representations based on random projections are very useful. This suggests that these random projections help to bootstrap generalization quickly. This seems to be confirmed by two other recent experiments. First, we showed [Léger *et al.* 2024] that adding a reservoir for preprocessing the inputs given to a state-of-the-art reinforcement learning algorithm could reduce the training time (which is usually in millions of training steps even for simple sensorimotor tasks like OpenAI Gym ones) [Brockman *et al.* 2016]. Second, in a prediction of COVID-19 hospitalization with only about 400 days of data with 400 features, RC is the best performing method compared several others [Ferté *et al.* 2024]. Thus, RC seems to have the ability to create good abstraction of inputs with little data.

More generally, in order to observe the emergence of interesting properties of human cognition, the following features appear to be the most compelling for making substantial progress in the development of neuro-plausible models: (1) seek for generic neural models that seeking for models able to *code-swtich* [Detraz & Hinaut 2019], that is, having the flexibility to switch language within a sentence; (2) seek for sensorimotor representations (e.g. commons in articulatory representations in bilingual models [Moore *et al.* 2025]) and self-supervision thanks to the sensorimotor action-perception loop [Schwartz *et al.* 2012]; (3) seek for new ways of generalization with little data, embrassing the *less is more* paradigm [Cohen *et al.* 2025]. Surprisingly, although bilingual models were already developped since a while [Frank 2021, Moore *et al.* 2025], models experimenting code-switching seem to have emerge only recently [Detraz & Hinaut 2019,

[14] Code-switching refers to the alternation between two or more languages within a single conversation, sentence, or utterance.

[15] Originally the reservoir parser language model, called *ResPars*, was using predicates as output representation.

Tsoukala *et al.* 2021, Frank *et al.* 2022]. Whereas, this ability to flexibility stwitch within a sentence is related to *cognitive control*, another ability of humans which is explored in cognitive models [O'Reilly *et al.* 2010], an important cognitive function that current AI systems lack for genuine flexibility.

1.5 Discussion

1.5.1 Conclusion

Grounded language acquisition relates to how linguistic symbols acquire meaning through interaction with the environment, a central issue in both infant language development and robotic learning [Chen & Mooney 2008, Thomason *et al.* 2018, Juven & Hinaut 2020, Vanzo *et al.* 2020]. A key challenge in this context is learning under ambiguity, as utterances typically co-occur with rich perceptual scenes containing multiple potential referents.

In this chapter, we investigated this problem using a cross-situational learning setup in which models must learn from full sentences under weak and noisy supervision. Rather than focusing on isolated words or short, command-like utterances, as is common in many robotic grounding approaches [Chen & Mooney 2011, Matuszek *et al.* 2013, Tellex *et al.* 2011], we considered sentence-level learning in ambiguous contexts, in closer alignment with developmental observations.

The proposed modeling framework emphasizes simple sequence-based architectures, incremental processing, and dynamic memory, allowing efficient learning from limited data. By relying on co-occurrence statistics rather than explicit disambiguating cues, the models capture key aspects of the learning conditions faced by human learners, who must infer meaning without direct supervision.

While more complex neural architectures can achieve strong performance on large-scale language tasks, they typically rely on extensive data and pretraining. In contrast, the approach explored here highlights how lightweight and biologically inspired models can generalize in low-data regimes, offering complementary insights into the mechanisms that may support early language acquisition.

Finally, the present study deliberately focuses on textual input and symbolic scene representations in order to isolate core learning mechanisms. Extending this framework to richer perceptual modalities, such as vision and speech, represents an important direction for future work and a necessary step toward more comprehensive models of grounded language learning.

1.5.2 Why Reservoir Computing fits cross-situational learning

Beyond the empirical results presented in this chapter, several intrinsic properties of Reservoir Computing help explain its suitability for cross-situational learning under weak and noisy supervision.

Learning from limited data. Reservoir Computing can be interpreted as a form of high-dimensional temporal projection followed by a simple linear read-

out, an analogy sometimes drawn with kernel methods or support vector machines. In such settings, learning does not require dense coverage of the input space, but rather sufficient information near decision boundaries. This property helps explain why reservoir-based models can generalize from relatively small datasets, as observed in this chapter and in previous work on audio and language processing tasks [Trouvain & Hinaut 2021, Variengien & Hinaut 2020, Oota *et al.* 2022]. This characteristic is particularly relevant for modeling language acquisition, where learners must infer structure from sparse and noisy observations.

Random projections as a generic computational principle. A central ingredient of Reservoir Computing is the use of random projections to map sequential input into a high-dimensional dynamical space. Random projections are known to preserve the geometric structure of data while enabling efficient dimensionality expansion, as formalized by results such as the Johnson–Lindenstrauss lemma [Johnson *et al.* 1984]. Similar principles underlie methods such as compressive sensing, where sparse signals can be recovered from a limited number of incoherent measurements [Duarte *et al.* 2006]. From this perspective, random projections can be seen as a generic and low-cost computational mechanism for structuring information and promoting abstraction, one that biological systems may plausibly exploit.

Bootstrapping Poverty of the Stimulus Through Random Projections The structuring and abstracting properties of Reservoir Computing could be interesting to consider in debates surronding the *Poverty of the Stimulus* Hypothesis [Chomsky 1965]. In a rational analysis of acquisition, what looks like "poverty of the stimulus" can often be alleviated by domain-general inductive biases that make learning from sparse data feasible [Prefors *et al.* 2006]. Consistently, recent evidence from neural language models challenges the claim that innate syntax is the only route to generalization, while suggesting that human-like data efficiency requires inductive biases beyond those tested so far [Yang *et al.* 2026]. From this viewpoint, it would be interesting to examine whether such inductive biases could be instantiated as particular reservoir dynamics – with hyperparameters fixed a priori or slowly modulated by changes in gene regulation – without committing to explicitly innate linguistic constraints.

Reservoirs as substrates for computation. Beyond abstract models, Reservoir Computing highlights the idea that useful computation can emerge from a wide variety of physical substrates [Tanaka *et al.* 2019, Ahavi *et al.* 2026b]. Instead of carefully optimized networks, reservoirs rely on the intrinsic dynamics of complex systems, with learning confined to a simple readout layer. This view suggests that equivalents of random projections and rich dynamical transformations may be readily available in physical, chemical, or biological systems. From a practical standpoint, this also points toward more computationally efficient and potentially energy-efficient alternatives to training deep networks with gradient-based methods.

Taken together, these properties help explain why Reservoir Computing provides a compelling framework for studying (grounded) language acquisition from limited data. Its combination of simplicity, expressive dynamics, and biological plausibility

makes it well suited for investigating how structured linguistic representations can emerge from sequential input under realistic learning constraints.

1.6 Appendix: Some Reservoir Formalism

There can be different kinds of units in a reservoir: spiking or non-spiking (average firing rate) neurons. There are different kinds of equations for both. I will just present the most popular version of reservoir: the Echo State Network (ESN). One of the general ways to define ESN is by using the leaky version. The state transition of the leaky-ESN is computed as follows:

$$x(t) = (1-\alpha)x(t-1) + \alpha\,\mathbf{tanh}(W_{in}u(t) + Wx(t-1)) \tag{1.1}$$

where $u(t) \in R^{N_U}$ is the input vector at time t, $x(t) \in R^{N_R}$ is the reservoir state, $W_{in} \in R^{N_R \times N_U}$ is the input matrix, $W \in R^{N_R \times N_R}$ is the recurrent matrix, $\alpha \in [0,1]$ is the leaking rate – more often called the *leak-rate* – and **tanh** is the element-wise hyperbolic tangent. N_U is the number of input units and N_R the number of units in the reservoir. The leak-rate is equivalent to the inverse of a time constant, it is a simplification of writing:

$$\alpha = \frac{dt}{\tau} \tag{1.2}$$

with τ the time constant of neurons and dt the time step discretisation (which equals 1 by default)[16].

The values of matrix W are randomly initialized, for instance using a uniform distribution and then rescaled. This rescaling of W is done in order to obtain a spectral radius[17] ρ equal to the one set by the user as HP[18]. The values in matrix W_{in} are randomly initialized, for instance from a uniform distribution and then rescaled in order to have an *input scaling* of σ, which is the one set by the user as hyperparameter. Usually, W and W_{in} matrices are sparse: my recommendation is to use a percentage of non-zero connection of about $10-20\%$, but the influence of the sparseness on the performance is often weak. A sparse reservoir enables faster computations.

The output of the ESN is computed as follows:

$$y(t) = W_{out}[1; x(t)] \tag{1.3}$$

where $y(t) \in R^{N_Y}$ is the output at time t, W_{out} is the output matrix, and $[.;.]$ stands for the concatenation of two vectors. N_Y is the number of output (read-out) units.

The output weights are learned using an equivalent of linear regression. The most common practice is to use a regularized version, like the ridge regression:

$$W_{\text{out}} = YX^T(XX^T + \beta I)^{-1} \tag{1.4}$$

[16] We showed in [Hinaut & Dominey 2013] that changing dt does not affect much the performance on a language task as soon as the sampling rate of inputs are changed accordingly.

[17] The spectral radius is the maximum absolute eigenvalue of the matrix W.

[18] A hyperparameter is a parameter that need to be predefined and which is not optimized by the learning algorithm.

where X is the concatenation of the reservoir activities at all time steps with a bias vector at 1, each row corresponding to a time step. Similarly, Y is the concatenation of desired outputs and β is the regularization parameter (often called *ridge* parameter).

A few more details. The spectral radius ρ controls the internal dynamics: more stable dynamics will be obtained for low values and more chaotic ones with high values. I will not talk about the ESP as it is a theoretical recommendation from Jaeger [Jaeger 2001] (derived from principles of linear networks) but not a rule that should be followed blindly. In practice spectral radii higher than one should be always tried when exploring hyperparameters because an ESN is a non-linear system that depends on its inputs. Especially, in the case of the leaky ESN where the *effective spectral radius* is different from the one defined by the user [Jaeger *et al.* 2007][19].

You can find a tutorial to explore the hyperparameters of reservoirs in the GitHub repository of our *ReservoirPy* library[20]. We illustrate plots to show how the internal dynamics of the network change with respect to the changes of hyperparameters such as the spectral radius, the input scaling and the leak-rate.

In most our studies, we are using ESNs as defined by Jaeger [Jaeger 2001, Jaeger *et al.* 2007], where the state of each recurrent unit also corresponds to its output (i.e. the activation function applies to the states directly). One may argue that it is less biologically plausible, but it has the advantage of having bounded states which prevents the states to take infinite values – which would stop the program because *NAN* values are encountered. Of course bounded states are obtained with a bounded activation function: e.g. *tanh*. This is one of the reasons why we use Jaeger's definition of ESNs. To my knowledge, they seem to be the most used type of reservoir since two decades. Another reason is that it enables to compare our models with many other published papers. For a detailed explanation of the various version of ESNs, David Verstraeten provides a clear explanation in his PhD thesis [Verstraeten 2009].

[19] I have unpublished results showing that one can have very high values of spectral radius (e.g. a million) that still work for a given task as soon as one also decrease the leak-rate. Hyperparameters such as the spectral radius, the leak-rate and the input scaling are linked, that is why we suggest to fix at least one of them when doing hyperparameter exploration [Hinaut & Trouvain 2021].

[20] https://github.com/reservoirpy/reservoirpy/blob/master/tutorials/4-Understand_and_optimize_hyperparameters.ipynb

Bibliography

[Achiam *et al.* 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat*et al. GPT-4 technical report*. arXiv preprint arXiv:2303.08774, 2023. (Cited on page 8.)

[Ahavi *et al.* 2026a] Paul Ahavi, Thi-Ngoc-An Hoang, Philippe Meyer, Sylvie Berthier, Federica Fiorini, Florence Castelli, Olivier Epaulard, Audrey Le Gouellec and Jean-Loup Faulon. *Living Bacterial Reservoir Computers for Information Processing and Sensing*. bioRxiv, 2026. (Cited on page 11.)

[Ahavi *et al.* 2026b] Paul Ahavi, Audrey Le Gouellec and Jean-Loup Faulon. *Microbial computing: Review and perspectives*. Biotechnology Advances, vol. 87, page 108766, 2026. (Cited on pages 11 and 25.)

[Barone & Joseph 1989] P Barone and J-P Joseph. *Prefrontal cortex and spatial sequencing in macaque monkey*. Experimental brain research, vol. 78, no. 3, pages 447–464, 1989. (Cited on page 14.)

[Beinborn *et al.* 2018] Lisa Beinborn, Teresa Botschen and Iryna Gurevych. *Multimodal Grounding for Language Processing*. In Proceedings of the 27th International Conference on Computational Linguistics, pages 2325–2339, 2018. (Cited on page 16.)

[Bendi-Ouis & Hinaut 2024] Yannis Bendi-Ouis and Xavier Hinaut. *Recurrent Attention Network*. In Bernstein Conference 2024, 2024. (Cited on page 9.)

[Bendi-Ouis & Hinaut 2026] Yannis Bendi-Ouis and Xavier Hinaut. *Echo State Transformer: Attention Over Finite Memories*, 2026. (Cited on page 9.)

[Bengio *et al.* 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert and Jason Weston. *Curriculum learning*. In Proceedings of the 26th annual international conference on machine learning, pages 41–48, 2009. (Cited on page 4.)

[Binz *et al.* 2024] Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K Eckstein, Noémi Éltető*et al. Centaur: a foundation model of human cognition*. arXiv preprint arXiv:2410.20268, 2024. (Cited on page 9.)

[Brockman *et al.* 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang and Wojciech Zaremba. *OpenAI Gym*, 2016. (Cited on page 23.)

[Brouwer & Hoeks 2013] Harm Brouwer and John C. J. Hoeks. *A time and place for language comprehension: mapping the N400 and the P600 to a minimal*

cortical network. Frontiers in Human Neuroscience, vol. 7, 2013. (Cited on page 8.)

[Brown *et al.* 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell*et al. Language models are few-shot learners*. Advances in neural information processing systems, vol. 33, pages 1877–1901, 2020. (Cited on page 8.)

[Buonomano & Merzenich 1995] Dean V Buonomano and Michael M Merzenich. *Temporal information transformed into a spatial code by a neural network with realistic properties*. Science, vol. 267, no. 5200, pages 1028–1030, 1995. (Cited on page 13.)

[Cangelosi 2010] Angelo Cangelosi. *Grounding language in action and perception: From cognitive agents to humanoid robots*. Physics of life reviews, vol. 7, no. 2, pages 139–151, 2010. (Cited on page 15.)

[Chen & Mooney 2008] David L Chen and Raymond J Mooney. *Learning to sportscast: a test of grounded language acquisition*. In Proceedings of the 25th International Conference on Machine Learning, pages 128–135, 2008. (Cited on pages 15 and 24.)

[Chen & Mooney 2011] David Chen and Raymond Mooney. *Learning to interpret natural language navigation instructions from observations*. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 25, 2011. (Cited on pages 16 and 24.)

[Chollet 2019] François Chollet. *On the measure of intelligence*. arXiv preprint arXiv:1911.01547, 2019. (Cited on page 8.)

[Chomsky 1965] Noam Chomsky. Aspects of the theory of syntax. MIT press, 1965. (Cited on page 25.)

[Christiansen & Chater 2016] Morten H Christiansen and Nick Chater. *The now-or-never bottleneck: A fundamental constraint on language*. Behavioral and brain sciences, vol. 39, 2016. (Cited on page 7.)

[Christiansen *et al.* 2016] Morten H Christiansen, Nick Chater and Peter W Culicover. Creating language: Integrating evolution, acquisition, and processing. MIT Press, 2016. (Cited on page 7.)

[Cohen *et al.* 2025] Laura Cohen, Xavier Hinaut, Lilyana Petrova, Alexandre Pitti, Syd Reynal and Ichiro Tsuda. *Less is More: some Computational Principles based on Parcimony, and Limitations of Natural Intelligence*. arXiv preprint arXiv:2506.07060, 2025. (Cited on page 23.)

[Detraz & Hinaut 2019] Pauline Detraz and Xavier Hinaut. *A Reservoir Model for Intra-Sentential Code-Switching Comprehension in French and English*. In CogSci'19-41st Annual Meeting of the Cognitive Science Society, 2019. (Cited on pages 23 and 24.)

[Devlin *et al.* 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. CoRR, vol. abs/1810.04805, 2018. (Cited on page 8.)

[Dinh & Hinaut 2020] Thanh Trung Dinh and Xavier Hinaut. *Language Acquisition with Echo State Networks: Towards Unsupervised Learning*. In ICDL 2020 - IEEE International Conference on Development and Learning, Valparaiso / Virtual, Chile, October 2020. (Cited on pages 15, 21 and 23.)

[Dominey *et al.* 1995] Peter Dominey, Michael Arbib and Jean-Paul Joseph. *A model of corticostriatal plasticity for learning oculomotor associations and sequences*. Journal of cognitive neuroscience, vol. 7, no. 3, pages 311–336, 1995. (Cited on page 14.)

[Dominey 1995] Peter F Dominey. *Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning*. Biological cybernetics, vol. 73, no. 3, pages 265–274, 1995. (Cited on pages 13 and 14.)

[Doupe & Kuhl 1999] Allison J Doupe and Patricia K Kuhl. *Birdsong and human speech: common themes and mechanisms*. Annual review of neuroscience, vol. 22, no. 1, pages 567–631, 1999. (Cited on page 5.)

[Duarte *et al.* 2006] Marco F Duarte, Michael B Wakin, Dror Baron and Richard G Baraniuk. *Universal distributed sensing via random projections*. In Proceedings of the 5th international conference on Information processing in sensor networks, pages 177–185, 2006. (Cited on page 25.)

[Enel *et al.* 2016] Pierre Enel, Emmanuel Procyk, René Quilodran and Peter Ford Dominey. *Reservoir computing properties of neural dynamics in prefrontal cortex*. PLoS computational biology, vol. 12, no. 6, page e1004967, 2016. (Cited on page 14.)

[Ferté *et al.* 2024] Thomas Ferté, Dan Dutartre, Boris P Hejblum, Romain Griffier, Vianney Jouhet, Rodolphe Thiébaut, Pierrick Legrand and Xavier Hinaut. *Reservoir computing for short high-dimensional time series: an application to SARS-CoV-2 hospitalization forecast*. Proceedings of Machine Learning Research, 2024. (Cited on page 23.)

[Frank *et al.* 2022] Stefan Frank, Xavier Hinaut, Edith Kaan, Yung Han Khoe, Lin Chen, Irene Elisabeth Winther and Yevgen Matusevych. *Bilingual Sentence Processing: when Models Meet Experiments*. In Proceedings of the Annual Meeting of the Cognitive Science Society, volume 44, 2022. (Cited on page 24.)

[Frank 2021] Stefan L Frank. *Toward computational models of multilingual sentence processing.* Language Learning, vol. 71, no. S1, pages 193–218, 2021. (Cited on page 23.)

[Garagnani *et al.* 2008] Max Garagnani, Thomas Wennekers and Friedemann Pulvermüller. *A neuroanatomically grounded Hebbian-learning model of attention-language interactions in the human brain.* European Journal of Neuroscience, vol. 27, no. 2, pages 492–513, January 2008. (Cited on page 8.)

[Grattafiori *et al.* 2024] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan*et al. The llama 3 herd of models.* arXiv preprint arXiv:2407.21783, 2024. (Cited on page 3.)

[Haeusler & Maass 2007] Stefan Haeusler and Wolfgang Maass. *A statistical analysis of information-processing properties of lamina-specific cortical microcircuit models.* Cerebral cortex, vol. 17, no. 1, pages 149–162, 2007. (Cited on page 14.)

[Harnad 1990] Stevan Harnad. *The symbol grounding problem.* Physica D: Nonlinear Phenomena, vol. 42, no. 1-3, pages 335–346, 1990. (Cited on page 9.)

[Hejna *et al.* 2023] Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W Bradley Knox and Dorsa Sadigh. *Contrastive preference learning: learning from human feedback without rl.* arXiv preprint arXiv:2310.13639, 2023. (Cited on page 8.)

[Hinaut & Dominey 2013] Xavier Hinaut and Peter Ford Dominey. *Real-Time Parallel Processing of Grammatical Structure in the Fronto-Striatal System: A Recurrent Network Simulation Study Using Reservoir Computing.* PLoS ONE, vol. 8, no. 2, page e52946, February 2013. (Cited on pages 23 and 26.)

[Hinaut & Trouvain 2021] Xavier Hinaut and Nathan Trouvain. *Which Hype for my New Task? Hints and Random Search for Reservoir Computing Hyperparameters.* In ICANN 2021 - 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 2021. (Cited on page 27.)

[Hinaut & Twiefel 2019] Xavier Hinaut and Johannes Twiefel. *Teach your robot your language! trainable neural parser for modeling human sentence processing: Examples for 15 languages.* IEEE Transactions on Cognitive and Developmental Systems, vol. 12, no. 2, pages 179–188, 2019. (Cited on page 21.)

[Hinaut *et al.* 2014] Xavier Hinaut, Maxime Petit, Gregoire Pointeau and Peter F. Dominey. *Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks.* Frontiers in Neurorobotics, vol. 8, May 2014. (Cited on page 18.)

[Hinaut *et al.* 2015a] Xavier Hinaut, Johannes Twiefel, Maxime Petit, Peter Dominey and Stefan Wermter. *A recurrent neural network for multiple language acquisition: Starting with english and french.* In Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches (CoCo 2015), 2015. (Cited on pages 18 and 23.)

[Hinaut *et al.* 2015b] Xavier Hinaut, Johannes Twiefel, M Borghetti Soares, Pablo Barros, Luiza Mici and Stefan Wermter. *Humanoidly speaking–learning about the world and language with a humanoid friendly robot.* In International Joint Conference on Artificial Intelligence Video competition, 2015. (Cited on page 18.)

[Hinaut *et al.* 2016] Xavier Hinaut, Johannes Twiefel and Stefan Wermter. *Recurrent neural network for syntax learning with flexible predicates for robotic architectures.* In 2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), pages 150–151. IEEE, 2016. (Cited on pages 18 and 23.)

[Hochreiter & Schmidhuber 1997] Sepp Hochreiter and Jürgen Schmidhuber. *Long Short-Term Memory.* Neural Computation, vol. 9, no. 8, pages 1735–1780, November 1997. (Cited on page 18.)

[Jaeger & Haas 2004] Herbert Jaeger and Harald Haas. *Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication.* Science, vol. 304, no. 5667, pages 78–80, 2004. (Cited on page 14.)

[Jaeger *et al.* 2007] Herbert Jaeger, Mantas Lukoševičius, Dan Popovici and Udo Siewert. *Optimization and applications of echo state networks with leaky-integrator neurons.* Neural networks, vol. 20, no. 3, pages 335–352, 2007. (Cited on page 27.)

[Jaeger 2001] H Jaeger. *The "echo state" approach to analysing and training recurrent neural networks.* Bonn, Germany: GMD Technical Report, vol. 148, no. 34, 2001. (Cited on pages 14 and 27.)

[Jaeger 2007] Herbert Jaeger. *Echo state network.* Scholarpedia, vol. 2, no. 9, page 2330, 2007. (Cited on page 13.)

[Jiang *et al.* 2024] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand*et al. Mixtral of experts.* arXiv preprint arXiv:2401.04088, 2024. (Cited on page 8.)

[Johnson *et al.* 1984] William B Johnson, Joram Lindenstrauss*et al. Extensions of Lipschitz mappings into a Hilbert space.* Contemporary mathematics, vol. 26, no. 189-206, page 1, 1984. (Cited on page 25.)

[Juven & Hinaut 2020] Alexis Juven and Xavier Hinaut. *Cross-Situational Learning with Reservoir Computing for Language Acquisition Modelling.* In 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, July 2020. (Cited on pages 11, 15, 16, 17, 21 and 24.)

[Juzek & Ward 2024] Tom S Juzek and Zina B Ward. *Why Does ChatGPT" Delve" So Much? Exploring the Sources of Lexical Overrepresentation in Large Language Models.* arXiv preprint arXiv:2412.11385, 2024. (Cited on page 9.)

[Kuhl 2004] Patricia K Kuhl. *Early language acquisition: Cracking the speech code.* Nature reviews neuroscience, vol. 5, no. 11, pages 831–843, 2004. (Cited on page 5.)

[Léger *et al.* 2024] Corentin Léger, Gautier Hamon, Eleni Nisioti, Xavier Hinaut and Clément Moulin-Frier. *Evolving reservoirs for meta reinforcement learning.* In International Conference on the Applications of Evolutionary Computation (Part of EvoStar), pages 36–60. Springer, 2024. (Cited on page 23.)

[Lindsey *et al.* 2025] Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah and Joshua Batson. *On the Biology of a Large Language Model.* Transformer Circuits Thread, 2025. (Cited on page 9.)

[Maass *et al.* 2002] Wolfgang Maass, Thomas Natschläger and Henry Markram. *Real-time computing without stable states: A new framework for neural computation based on perturbations.* Neural computation, vol. 14, no. 11, pages 2531–2560, 2002. (Cited on page 14.)

[Machens *et al.* 2010] Christian K Machens, Ranulfo Romo and Carlos D Brody. *Functional, but not anatomical, separation of "what" and "when" in prefrontal cortex.* Journal of Neuroscience, vol. 30, no. 1, pages 350–360, 2010. (Cited on page 14.)

[Matuszek *et al.* 2013] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer and Dieter Fox. *Learning to parse natural language commands to a robot control system.* In Experimental Robotics, pages 403–415. Springer, 2013. (Cited on pages 16 and 24.)

[Mistral AI 2025] Mistral AI. *Introducing Mistral 3.* `https://mistral.ai/news/mistral-3`, 2025. Accessed: 2026-01-12. (Cited on page 10.)

[Moore *et al.* 2025] Charlotte Moore, Peter W Donhauser, Denise Klein and Krista Byers-Heinlein. *Efficient neural encoding as revealed by bilingualism.* Pro-

ceedings of the National Academy of Sciences, vol. 122, no. 34, page e2513768122, 2025. (Cited on page 23.)

[Oota *et al.* 2022] Subba Reddy Oota, Frédéric Alexandre and Xavier Hinaut. Cross-Situational Learning Towards Robot Grounding. HAL preprint, April 2022. (Cited on pages 16, 21 and 25.)

[OpenAI 2026] OpenAI. *What are tokens and how to count them*, 2026. Accessed on January 2026. (Cited on page 3.)

[Oudeyer *et al.* 2007] Pierre-Yves Oudeyer, Frdric Kaplan and Verena V Hafner. *Intrinsic motivation systems for autonomous mental development*. IEEE transactions on evolutionary computation, vol. 11, no. 2, pages 265–286, 2007. (Cited on page 4.)

[O'Reilly *et al.* 2010] Randall C O'Reilly, Seth A Herd and Wolfgang M Pauli. *Computational models of cognitive control*. Current opinion in neurobiology, vol. 20, no. 2, pages 257–261, 2010. (Cited on page 24.)

[Pagliarini *et al.* 2021] Silvia Pagliarini, Arthur Leblois and Xavier Hinaut. *Vocal Imitation in Sensorimotor Learning Models: A Comparative Review*. IEEE Transactions on Cognitive and Developmental Systems, vol. 13, no. 2, pages 326–342, June 2021. (Cited on page 5.)

[Paulo *et al.* 2024] Gonçalo Paulo, Alex Mallen, Caden Juang and Nora Belrose. *Automatically interpreting millions of features in large language models*. arXiv preprint arXiv:2410.13928, 2024. (Cited on page 9.)

[Pedrelli & Hinaut 2020] Luca Pedrelli and Xavier Hinaut. *Hierarchical-Task Reservoir for Anytime POS Tagging from Continuous Speech*. In 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, July 2020. (Cited on page 23.)

[Pedrelli & Hinaut 2022] Luca Pedrelli and Xavier Hinaut. *Hierarchical-Task Reservoir for Online Semantic Analysis From Continuous Speech*. IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 6, pages 2654–2663, June 2022. (Cited on page 23.)

[Pickering & Garrod 2013] Martin J. Pickering and Simon Garrod. *An integrated theory of language production and comprehension*. Behavioral and Brain Sciences, vol. 36, no. 4, pages 329–347, June 2013. (Cited on page 8.)

[Prefors *et al.* 2006] Amy Prefors, Terry Regier and Joshua B Tenenbaum. *Poverty of the stimulus? A rational approach*. In Proceedings of the Annual Meeting of the Cognitive Science Society, volume 28, 2006. (Cited on page 25.)

[Pulvermüller & Fadiga 2010] Friedemann Pulvermüller and Luciano Fadiga. *Active perception: sensorimotor circuits as a cortical basis for language*. Nature

Reviews Neuroscience, vol. 11, no. 5, pages 351–360, April 2010. (Cited on page 8.)

[Rigotti *et al.* 2013] Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller and Stefano Fusi. *The importance of mixed selectivity in complex cognitive tasks.* Nature, vol. 497, no. 7451, pages 585–590, 2013. (Cited on pages 14 and 23.)

[Roembke *et al.* 2023] Tanja C Roembke, Matilde E Simonetti, Iring Koch and Andrea M Philipp. *What have we learned from 15 years of research on cross-situational word learning? A focused review.* Frontiers in Psychology, vol. 14, 2023. (Cited on page 15.)

[Roesler *et al.* 2018] Oliver Roesler, Amir Aly, Tadahiro Taniguchi and Yoshikatsu Hayashi. *A probabilistic framework for comparing syntactic and semantic grounding of synonyms through cross-situational learning.* In ICRA-2018 Workshop on" Representing a Complex World: Perception, Inference, and Learning for Joint Semantic, Geometric, and Physical Understanding", 2018. (Cited on page 16.)

[Saffran *et al.* 1996] Jenny R Saffran, Richard N Aslin and Elissa L Newport. *Statistical learning by 8-month-old infants.* Science, pages 1926–1928, 1996. (Cited on page 15.)

[Schwartz *et al.* 2012] Jean-Luc Schwartz, Anahita Basirat, Lucie Ménard and Marc Sato. *The Perception-for-Action-Control Theory (PACT): A perceptuo-motor theory of speech perception.* Journal of Neurolinguistics, vol. 25, no. 5, pages 336–354, September 2012. (Cited on pages 8 and 23.)

[Smith & Yu 2008] Linda Smith and Chen Yu. *Infants rapidly learn word-referent mappings via cross-situational statistics.* Cognition, vol. 106, no. 3, pages 1558–1568, 2008. (Cited on page 14.)

[Sussillo & Abbott 2009] David Sussillo and L.F. Abbott. *Generating Coherent Patterns of Activity from Chaotic Neural Networks.* Neuron, vol. 63, no. 4, pages 544–557, August 2009. (Cited on page 18.)

[Tanaka *et al.* 2019] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano and Akira Hirose. *Recent advances in physical reservoir computing: A review.* Neural Networks, vol. 115, pages 100–123, 2019. (Cited on pages 11 and 25.)

[Taniguchi *et al.* 2016] Tadahiro Taniguchi, Takayuki Nagai, Tomoaki Nakamura, Naoto Iwahashi, Tetsuya Ogata and Hideki Asoh. *Symbol emergence in robotics: a survey.* Advanced Robotics, vol. 30, no. 11-12, pages 706–728, April 2016. (Cited on page 9.)

[Taniguchi *et al.* 2017] Akira Taniguchi, Tadahiro Taniguchi and Angelo Cangelosi. *Cross-situational learning with Bayesian generative models for multimodal category and word learning in robots.* Frontiers in Neurorobotics, vol. 11, page 66, 2017. (Cited on pages 15 and 16.)

[Tellex *et al.* 2011] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew Walter, Ashis Banerjee, Seth Teller and Nicholas Roy. *Understanding natural language commands for robotic navigation and mobile manipulation.* In Proceedings of the AAAI Conference on Artificial Intelligence, volume 25, 2011. (Cited on pages 16 and 24.)

[Templeton *et al.* 2024] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah and Tom Henighan. *Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet.* Transformer Circuits Thread, 2024. (Cited on page 9.)

[Thelen & Smith 1994] Esther Thelen and Linda B Smith. A dynamic systems approach to the development of cognition and action. MIT press, 1994. (Cited on page 4.)

[Thomason *et al.* 2016] Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone and Raymond J Mooney. *Learning Multi-Modal Grounded Linguistic Semantics by Playing" I Spy".* In IJCAI, pages 3477–3483, 2016. (Cited on page 16.)

[Thomason *et al.* 2018] Jesse Thomason, Jivko Sinapov, Raymond Mooney and Peter Stone. *Guiding exploratory behaviors for multi-modal grounding of linguistic descriptions.* In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018. (Cited on pages 15 and 24.)

[Tomasello 2009] Michael Tomasello. Constructing a language. Harvard university press, 2009. (Cited on page 15.)

[Trouvain & Hinaut 2021] Nathan Trouvain and Xavier Hinaut. *Canary Song Decoder: Transduction and Implicit Segmentation with ESNs and LTSMs.* In ICANN 2021 - 30th International Conference on Artificial Neural Networks, volume 12895 of *Farkaš I., Masulli P., Otte S., Wermter S. (eds) Artificial Neural Networks and Machine Learning – ICANN 2021. Lecture Notes in Computer Science*, pages 71–82, Bratislava, Slovakia, September 2021. Springer, Cham. (Cited on page 25.)

[Trouvain *et al.* 2020] Nathan Trouvain, Luca Pedrelli, Thanh Trung Dinh and Xavier Hinaut. *Reservoirpy: an efficient and user-friendly library to design echo state networks.* In International Conference on Artificial Neural Networks, pages 494–505. Springer, 2020. (Cited on page 18.)

[Trouvain *et al.* 2026] Nathan Trouvain, Paul Bernard and Xavier Hinaut. reservoirpy: A Simple and Flexible Reservoir Computing Tool in Python. preprint, 2026. (Cited on pages 10 and 18.)

[Tsoukala *et al.* 2021] Chara Tsoukala, Mirjam Broersma, Antal Van Den Bosch and Stefan L Frank. *Simulating code-switching using a neural network model of bilingual sentence production.* Computational Brain & Behavior, vol. 4, no. 1, pages 87–100, 2021. (Cited on page 24.)

[Twiefel *et al.* 2016a] Johannes Twiefel, Xavier Hinaut, Marcelo Borghetti, Erik Strahl and Stefan Wermter. *Using natural language feedback in a neuro-inspired integrated multimodal robotic architecture.* In 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 52–57. IEEE, 2016. (Cited on page 18.)

[Twiefel *et al.* 2016b] Johannes Twiefel, Xavier Hinaut and Stefan Wermter. *Semantic role labelling for robot instructions using echo state networks.* In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2016. (Cited on page 18.)

[Vanzo *et al.* 2020] Andrea Vanzo, Danilo Croce, Emanuele Bastianelli, Roberto Basili and Daniele Nardi. *Grounded language interpretation of robotic commands through structured learning.* Artificial Intelligence, vol. 278, page 103181, 2020. (Cited on pages 15 and 24.)

[Vapnik 1999] Vladimir Vapnik. The nature of statistical learning theory. Springer science & business media, 1999. (Cited on page 12.)

[Variengien & Hinaut 2020] Alexandre Variengien and Xavier Hinaut. *A journey in ESN and LSTM visualisations on a language task.* arXiv preprint arXiv:2012.01748, 2020. (Cited on pages 15, 16, 18, 19, 21, 22, 23 and 25.)

[Vaswani *et al.* 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser and Illia Polosukhin. *Attention is All you Need.* In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. (Cited on pages 7 and 8.)

[Verstraeten 2009] David Verstraeten. *Reservoir Computing: computation with dynamical systems.* PhD thesis, Ghent University, 2009. (Cited on pages 11 and 27.)

[Warren *et al.* 2020] David E Warren, Tanja C Roembke, Natalie V Covington, Bob McMurray and Melissa C Duff. *Cross-situational statistical learning of new words despite bilateral hippocampal damage and severe amnesia.* Frontiers in Human Neuroscience, vol. 13, page 448, 2020. (Cited on page 15.)

[Yang *et al.* 2026] Xiulin Yang, Arianna Bisazza, Nathan Schneider and Ethan Gotlieb Wilcox. *A Unified Assessment of the Poverty of the Stimulus Argument for Neural Language Models*. arXiv preprint arXiv:2602.09992, 2026. (Cited on page 25.)

[Yu & Smith 2007] Chen Yu and Linda B Smith. *Rapid word learning under uncertainty via cross-situational statistics*. Psychological science, vol. 18, no. 5, pages 414–420, 2007. (Cited on pages 14 and 15.)