scientific reports

Novelty and imitation within the brain: a Darwinian neurodynamic approach to combinatorial problems

Dániel Czégel[1,2,3,4,6]✉, Hamza Giaffar[5,6], Márton Csillag[1,6], Bálint Futó[1] & Eörs Szathmáry[1,2,3]✉

Efficient search in vast combinatorial spaces, such as those of possible action sequences, linguistic structures, or causal explanations, is an essential component of intelligence. Is there any computational domain that is flexible enough to provide solutions to such diverse problems and can be robustly implemented over neural substrates? Based on previous accounts, we propose that a Darwinian process, operating over sequential cycles of imperfect copying and selection of neural informational patterns, is a promising candidate. Here we implement imperfect information copying through one reservoir computing unit teaching another. Teacher and learner roles are assigned dynamically based on evaluation of the readout signal. We demonstrate that the emerging Darwinian population of readout activity patterns is capable of maintaining and continually improving upon existing solutions over rugged combinatorial reward landscapes. We also demonstrate the existence of a sharp error threshold, a neural noise level beyond which information accumulated by an evolutionary process cannot be maintained. We introduce a novel analysis method, neural phylogenies, that displays the unfolding of the neural-evolutionary process.

Life, with its "endless forms most beautiful"[1], is a result of a Darwinian evolutionary process operating over the enormous representational capacity of chemistry. Any Darwinian process is built on the principles of replicating units, hereditary variation, and selection[2]; beyond the ability to sustain these, a Darwinian process makes no further demands of the underlying substrate. Replication, hereditary variation and selection collectively operate at other levels of biological organization, for example in the mammalian immune system[3], and may well feed on other-than-chemical substrates. The brain's neural networks are another example of a system that produces seemingly endless beautiful forms, from neural activity to the mental states and actions that they support. The endeavor of *Darwinian neurodynamics* (DN) explores the possible ways in which (i) Darwinian dynamics might emerge as an effective high-level algorithmic mechanism from plasticity and activity dynamics of neural populations[4,5] (*how does it work?*) and (ii) this high level algorithmic mechanism fits into cognition (*what is it good for?*). This endeavour can be seen as part of a more general pursuit of a theory of high-dimensional adaptations that unifies our understanding of evolutionary and learning processes (see Table 1).

In any model of DN, the modeller is required to specify how the ingredients of Darwinian evolution, namely replication, selection, and the heritable variation of firing patterns, emerge from lower-level neural activity and synaptic plasticity rules. A related early idea, termed *neural Darwinism*, together with its more recent implementations, considers selective amplification of a pre-existing but possibly adaptive repertoire of computational units[27–31]. It is halfway between traditional neural adaptation mechanisms, implementing effective high-dimensional hill-climbing in representation space, and a bona fide evolutionary search, with multiple rounds of selection over heritable variation generated by high-fidelity replication[4,32]. Building on these selectionist ideas, the *neuronal replicator hypothesis*[33] explores possible ways to add the crucial missing ingredient, replication, using exclusively local neural plasticity and activity rules. Neural replicators proposed to date come in two flavours:

[1]Institute of Evolution, Centre for Ecological Research, Budapest, Hungary. [2]Department of Plant Systematics, Ecology and Theoretical Biology, Eötvös University, Budapest, Hungary. [3]Parmenides Foundation, Center for the Conceptual Foundations of Science, Pullach, Germany. [4]Beyond Center for Fundamental Concepts in Science, Arizona State University, Tempe, AZ, USA. [5]Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA. [6]These authors contributed equally: Dániel Czégel, Hamza Giaffar and Márton Csillag. ✉email: danielczegel@gmail.com; szathmary.eors@gmail.com

1. Learning in evolution
How does a Darwinian process over populations of non-neural systems give rise to algorithmic elements of learning?
Emergent neural network-like dynamics. Connection strengths between units adapt and store information about the environment; activation dynamics of units, parametrized by the adapted interaction strengths, compute[6]. Paradigmatic examples include Hebbian learning in gene regulatory networks[7] and in ecosystems[8]. Computation, on a developmental and ecological timescale, gives rise to autoassociative dynamics. As a result, developmental systems and ecosystems, according to this theory, store memory traces about past environments, making the system easily re-adapt to those environments once they recur. Note that local Hebbian dynamics is either a consequence of direct selection for efficiency on a population of networks, as in gene regulatory networks, or it is not selected for, as in ecosystems[9]
Emergent Bayesian dynamics. Type frequencies store and update information about the environment, in a way that it is isomorphic to the competition dynamics of statistical hypotheses in a Bayesian setting[10–12]. Structural-dynamical equivalences extend far beyond those of replicator dynamics and Bayesian update[13]. They include a mapping between multilevel selection and Bayesian inference in hierarchical models[14], between quasispecies dynamics and filtering in hidden Markov models[15], and between evolutionary-ecological competition dynamics and expectation-maximization optimization of mixture models[13]. In short, Darwinian dynamics can be construed as accumulating evidence for models of the environment that are entailed in the phenotype[13,16,17]. This provides an interesting perspective that connects the normative formulation of natural selection in the brain to Bayesian model selection (i.e., structure learning) under the Bayesian brain hypothesis[14,18–21]
Emergent sampling algorithms implemented by the evolution of finite populations[12]. Sampling non-trivial probability distributions is considered to be a more and more fundamental module of mental computations[22]. Adaptation to appropriate fitness landscapes by a finite population results in sampling a corresponding distribution, making use of the stochasticity provided by genetic drift. Two examples are the evolution of types in finite population models with a regime of strong selection and weak mutation giving rise to Markov Chain Monte Carlo dynamics[23] and the equivalence of the evolution of relative frequencies in the Wright-Fisher model and a fundamental approximative statistical inference algorithm, particle filtering[12]
Emergence of evolutionary novelties as insights in the course of problem solving. At a macroevolutionary timescale, evolution proceeds by long nearly-static periods punctuated by sudden "inventions" of novel phenotypic solutions; this has been conceptually and algoritmically linked to search for out-of-the-box solutions in insight problems, where a sudden conscious emergence of a right solution follows a long incubation period marked by "stasis in solution space"[24,25]
2. Evolution in learning
How does a Darwinian process emerge from local learning rules?
The low-level channel of information passing between the system and the environment is any mechanism that provide plasticity and adaptation at a lower algorithmic level; heritable variation and selection is a higher-level emergent property. Darwinian neurodynamics exemplifies this approach. As opposed to proposed mechanisms under the umbrella conventionally called neural Darwinism[26,27], Darwinian neurodynamics (DN) decouples the concept of population in the neural and evolutionary sense: the replicators are neural activity patterns, not anatomical structures[4]. In particular, a network composed of the same exact neurons and synapses might produce different activity patterns at different time instances and therefore give rise to different replicators in the evolutionary sense. Another fundamental difference between neural Darwinism and Darwinian neurodynamics is that the latter performs bona fide evolutionary search, in which multiple rounds of selection acts on heritable variation

Table 1. A unified view of evolutionary and learning processes, as part of a theory of high-dimensional adaptations.

(i) *combinatorial replicators* copy a sequence of low-informational states of neural activity (represented by e.g. bistable neurons), akin to template replication[33,34], and(ii) *holistic replicators*[35] copy one high-informational state, such as memory traces of autoassociative attractor networks[5]. Three broad categories of *neuronal* units of evolution have been suggested: (i) neural activity patterns[5,33], (ii) connectivity[34], and (iii) evolvable neuronal paths[30]. If neuronal replicators exist, their appearance marks a new way in which information is represented in evolutionary units and transmitted between different neuronal replicators: a filial major transition in evolution, as proposed in[36].

DN is similar to other approaches to understanding the computational functions of the mind and its neural correlates in the brain: efficient theoretical frameworks of adaptation, notably reinforcement learning and statistical learning, have been successfully guiding experimentation in neuroscience in recent years. As in the aforementioned theories, in DN, the neural correlates of crucial algorithmic elements, replication, selection and heritable variation, are a priori unknown, emphasizing the importance of hypothesis-driven search. When searching for these algorithmic ingredients of an emergent Darwinian process, three important conceptual issues arise recurrently.

1. *Unlimited hereditary potential* Long term evolution rests on the ability of novel solutions, generated via mutation, to replicate. If the set of solutions that can be copied with high-enough fidelity is limited, no efficient evolutionary search can take place. This requirement for high enough fidelity copying of a very large range of potential solutions is one that directly in silico models of evolution rarely has to be considered, as replication can be achieved simply with arbitrary precision. When the replication mechanism is itself explicitly modelled, the question of its capacity to replicate solutions must be addressed.
2. *Maintenance of information* Although the appearance of novel beneficial mutations is necessary, it is not sufficient for continuous evolutionary search; information must also be maintained. In particular, the information that is lost through deleterious mutation must (at least) be compensated by selection. This sets an upper limit on the mutation rate, depending on the selective advantage of the current solution over the surrounding ones.
3. *Representation of phenotype/solution* In evolutionary theory, phenotypes over which selection acts, are represented by genotypes that replicate. In neurobiology, computational states (e.g. solutions to a given problem) are represented by neural activity patterns. These mappings between genotype/neural activity and phenotype/solution might be highly complex. Representation is important for at least two different reasons. On one hand, it provides enormous adaptive potential: appropriate transformation of rich-enough representations might render simple algorithms surprisingly powerful. In evolution, in particular, this permits genotypic

> mutations to be channeled along some, potentially fit, phenotypic directions; this is one of the mechanisms upon which evolvability rests. On the other hand, we must separate the effect of representation and the algorithm operating on top of it; in particular, the evaluation of the efficiency of algorithms based on a few hand-picked and hand-fabricated representations of a given problem. This issue is unavoidable in DN, as we have at this point no detailed knowledge of how particular mental states are represented in human and animal brains. That is, the understanding of these relevant variables at the computational/behavioral, algorithmic and implementational level remains at present far from complete.

In this paper, we propose a model of Darwinian neurodynamics based on the emergent replication of *dynamic* patterns of neural activity. Building on the versatility and efficiency of the reservoir computing paradigm, we demonstrate that dynamic activity patterns can be effectively replicated via *training one recurrent network by another*. The two main components of our model that map both to building blocks of evolutionary dynamics and proposed mechanisms of mental search (e.g. in the combinatorial space of action sequences) are *evaluation* of proposed activity patterns and nearly perfect *copying* of them. The latter proceeds via a *teacher* network, i.e. the network that generated the fitter signal, training a *learner* network, i.e. associated to the less fit signal. We show that an ensemble of such recurrent networks, capable of training one another in parallel based on previous evaluation of their output signals, can give rise of an effective evolutionary search that *i)* maintains information about already discovered solutions and *ii)* improves on them occasionally through the appearance and spread of beneficial mutations.

By continually selecting and imperfectly replicating solutions that are better fit to the demands of the cognitive task at hand, and discarding those which don't meet the muster, DN can be seen as a process of stochastic parallel search with redistribution of resources. In the biosphere, this process of Natural Selection successfully operates in an enormously high dimensional (genomes range in size from $\sim$ 2kb to 10^8kb), combinatorial (each position can be one of four bases) and compositional (genomes are semantically and syntactically complex) space of genotypes. The solution spaces of problems in many cognitive domains share these characteristics to differing extents. Language is one such domain; indeed an evolutionary computational framework has previously been reported addressing aspects of this faculty[37], in which linguistic constructs (as units of evolution) can be coherently multiplied with hereditary variation under selection to achieve e.g. mutual comprehension in a population. Causal reasoning and action planning are two other examples of cognitive domains characterized by complex search spaces, where it is unclear if local gradient information is useful. To model these complex search spaces in the most generic manner possible, we focus on the capacity of neural-evolutionary dynamics to solve two classes of combinatorial search problems: the Travelling Salesman Problem (TSP) and the tunably rugged NK landscapes[38]. We demonstrate novel analysis methods that combine approaches from computational neuroscience and phylogenetics, and are generally applicable to any proposed DN architecture.

This endeavour of understanding the implementational, algorithmic and computational features of Darwinian neurodynamics is worthwhile for at least two reasons: (i) for its potential as a useful, though not necessarily ubiquitous, computational module in wet brains; a capacity, which, if it exists, has been selected for by biological natural selection, and (ii) for its potential as a computational module that is orthogonal to current paradigms, implemented in engineered systems such as spiking neural network hardware architectures. We do not expect that DN, as a computational module, is necessarily topmost in the computational hierarchy and therefore we expect behavioral correlates to potentially be indirect. More emphasis is placed here on discussing DN as a fundamental problem-solving neural machine as opposed to relating its solutions to possibly convoluted human behavioral patterns.

Building blocks of models of Darwinian neurodynamics

When proposing a DN architecture, one has to specify how necessary elements of a Darwinian evolutionary process, namely, replication and selection of units of evolution, and heritable variation over them, are neurally implemented. In the following, we show how these choices can be made by building up our model architecture, based on dynamic firing patterns as evolutionary units, which we refer to as *recurrent DN*, in a step-by-step manner.

Unit of evolution. The unit of evolution in recurrent DN is the output firing pattern of a recurrent computational unit, as illustrated in Fig. 1a. It is, in the current model, a one-dimensional time-dependent signal, usually interpreted as neuronal firing rate. Importantly, the unit of evolution is not the recurrent network itself, it is the computation it does. We use reservoir computing units to implement these recurrent computations. In reservoir computing units, learning takes place only through the linear readout synapses; the recurrent connections within the network are fixed. This makes the map from computer to computation highly degenerate: the same computation, i.e., output signal, can be represented (at least approximately) by an enormous number of different reservoirs. The reservoirs themselves form representational slots for the replicators, i.e., the output signals.

Replication. The crucial step in any DN architecture is the replication of firing patterns, utilizing local learning rules. Here, in recurrent DN, replication takes place by one recurrent network training another, as shown in Fig. 1b. More precisely, the training signal of reservoir A is the output of reservoir B, rendering, after some training period, the two output signals *approximately* the same. In other words, the output of reservoir B serves as a template for the output of reservoir A, resulting in the replication of the output of reservoir B, with some variation. We use the FORCE algorithm[39] to train reservoirs. Note that the FORCE algorithm is a supervised learning algorithm, which is in general inapplicable for computations where a supervised signal is not available;

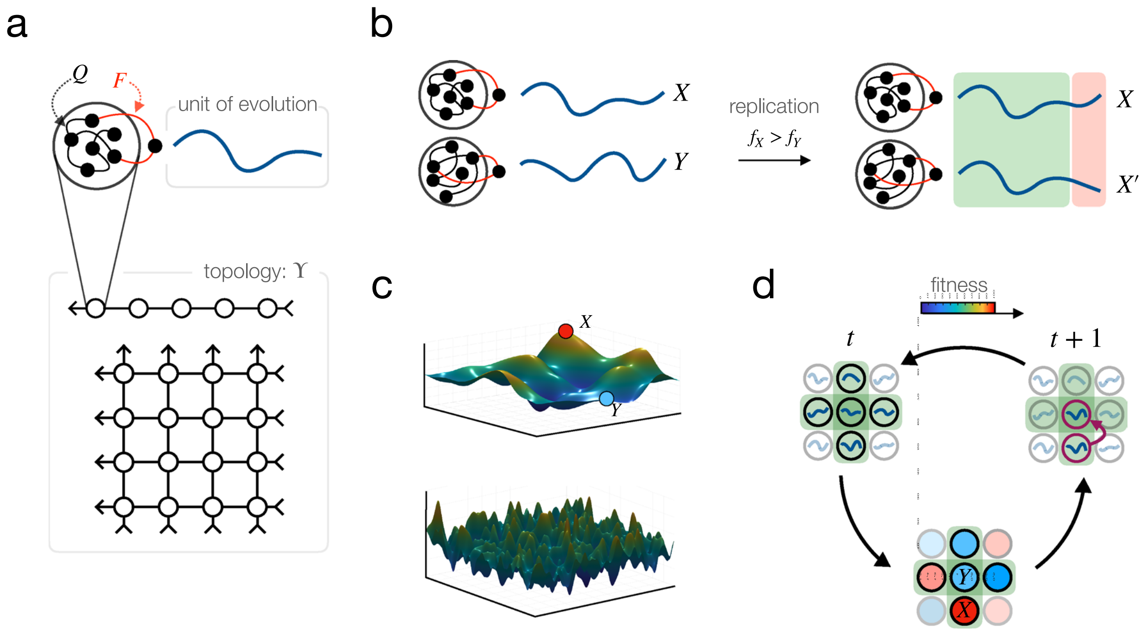

Figure 1. Schematic illustration of component processes of recurrent Darwinian neurodynamics. (**a**) Reservoir computers, arranged in a 1D array or a 2D lattice, host replicating signals. (**b**) Replication of output signals takes place by one reservoir training another. Reservoirs that output fitter signals train, those that output less fit signals learn. Training time and neural noise determine the fidelity of replication. (**c**) Illustration of ruggedness of fitness landscapes. More rugged landscapes, such as the one in the bottom, has more non-equivalent local maxima, making optimization more difficult. (**d**) Full dynamics. During one generation, each reservoir output Y is compared to those of its neighbours; the reservoir with the best output signal among the neighbours (here, X) replicates its signal to reservoir Y by training it.

an evolutionary process artificially creates supervised signals by displacing the low-dimensional informational bottleneck between the environment and the system from reservoir training to fitness computations.

Heritable variation. Heritable variation is a necessary for selection to act upon, and therefore, to improve solutions over time according to their fitness metric. Variation comes from two sources in our model. (i) It comes from imperfect training of one network by another via the FORCE algorithm, and (ii) it also comes from neural noise that we model as white noise added to the learned signal. Importantly, this variation is heritable if the resulting mutated signal can be learned by yet another reservoir. If the set of heritable signals is practically unlimited (i.e., much larger than the population size, the number of replicating signals), the system exhibits unlimited hereditary potential, upon which long-term evolution can take place.

Fitness landscape. Evolution, like reinforcement learning, necessarily channels all information about the environment through a one-dimensional bottleneck. It is this scalar-valued function, the fitness function, that must contain all task-related information if the architecture itself is meant to be of general purpose. It is not in the scope of this paper to discuss possible implementations of this evaluation signal in wet brains, but this similarity suggests that if Darwinian dynamics takes place in brains, the striatal reward system might be a possible candidate location of fitness assigment. In this paper, we intend to demonstrate the feasibility of an emergent Darwinian process over dynamic neural firing patterns in an as task-agnostic manner as possible. We therefore evaluate our recurrent DN architecture on general combinatorial landscapes with many non-equivalent local optima: on the Tavelling Salesman Problem and on NK-landscapes with different ruggedness. Figure 1c visualizes the idea of ruggedness of a continuous 2-dimensional landscape; in our simulations, we use high-dimensional discrete landscapes that are difficult to visualize.

The traveling salesman problem (TSP) is stated as: compute the shortest possible route that visits each city (from a given list of destinations) and returns to the origin city. Possible solutions can thus be represented as permutations of the cities. In this classic NP hard combinatorial optimization problem, the total path length l for a permutation is related to fitness f by any monotonously decreasing function.

NK-landscapes[40], originally modelling epistatic fitness interactions of genes, are simple non-deterministic landscapes that are able to model arbitrary level of ruggedness, set by a parameter K. They assign fitness f to binary sequences $x_1, x_2, \ldots x_N$ as

$$f(x_1, x_2, \ldots, x_N) = \sum_{i=1}^{m} f_i \tag{1}$$

where each of the m fitness-components f_i depend on exactly $K \leq N$ coordinates. The value of a fitness component for all of its 2^K possible arguments is drawn from the uniform distribution over [0, 1] independently. As an effect of the stochastic nature of NK landscape generation, realizations of landscapes with the same parameters N and K generally differ. N and K therefore define task *classes*, with N, the number of bits or "genes", parametrizing problem size and K parametrizing "ruggedness" or "frustration" in the system[41]. Note that NK landscapes include spin glass Hamiltonians as special case, corresponding to (a subset of) $K = 2$ landscapes. $K > 2$ landscapes, on the other hand, model higher order interactions between elements. However, even low levels of ruggedness (any $K > 1$) make finding even local optima difficult[42]. Overall, NK landscapes form a simple *and* flexible benchmark for combinatorial problems.

Selection. Selection in DN refers to the assignment of number of offspring to replicating firing rate patterns. It is done through evaluating the solutions represented by the replicating signals according to a scalar function, the fitness. In our architecture, reservoirs that output higher-fitness signals become *trainer* reservoirs, reservoirs outputting lower-fitness signals are assigned to be *learners*. The signal of trainer reservoirs reproduce, while the signal of learner reservoirs dies (i.e., disappears form the population of signals). The full evolutionary process over a two-dimensional sheet of reservoirs is illustrated on Fig. 1d.

Population structure defined by cortical topology. Fitness values determine the reproduction rate of signals, in terms of assigning the corresponding recurrent unit to be either teacher or learner. This assignment is dependent on comparing a subset of signals with each other. We define these competing subsets in terms of a connection topology of recurrent units: each recurrent unit (which is a neural network itself) is competing with its direct neighbours according to this pre-defined meta-topology. Special cases of this network meta-topology include a one-dimensional array of networks for illustration purposes (see phylogenetic trees of signals in section 3), and a two-dimensional grid, that might be identified with a cortical sheet of canonical microcircuits, see Fig. 1a. Although these meta-topologies induce a local competition dynamics, competing groups overlap and therefore high-fitness solutions are able to spread in the population globally. Population structure, defined here by this meta-topology of recurrent units, strongly shapes the course of evolutionary competition dynamics in general. If the evolutionary system as a whole is itself subject to optimization, e.g. by another evolutionary process at a higher level, we might expect to observe fine-tuned hyper-parameters in general, and a fine-tuned cortical topology in particular. Although optimizing cortical topology for various performance measures is out of the scope of this paper, we refer to evolutionary graph theory[43] as a systematic study of amplifying and diminishing effects of population structure on selection.

Representation. Notice that replication acts at the level of signals whereas selection acts at the level of solutions represented by the signals. This mapping between signal to solution determines the accessibility of solutions by evolution, just like the genotype-phenotype map in biology determines the accessibility of phenotypes. Solutions that are represented by many signals can be found with relative ease by evolutionary trajectories, whereas solutions that are represented by a tiny fraction of signal space or not at all are difficult to find. Statistical properties of the signal-solution map are therefore of fundamental importance. Since our goal here is to explore how well our recurrent DN arcitecture performs over rugged fitness surfaces with a large number of non-equivalent local optima, we map the signals to combinatorial solutions represented by (i) permutations, in case of the travelling salesman problem (TSP) fitness landscape, or by (ii) binary sequences, in case of the NK-landscape. These signal-solution maps are visualized in Fig. 2a; their statistical properties are shown in Fig. 2b–e.

Results

Having described the essential ingredients, we now focus on exploring the characteristics of recurrent Darwinian neurodynamics as a coherent computational framework. In all cases, we either use a one dimensional array, or a two-dimensional sheet of reservoirs as meta-topology (i.e., population structure from an evolutionary point of view), mirroring the possibility of representation by cortical microcircuits. First, we illustrate the evolution of the *best* firing pattern and the corresponding solution on a well-known combinatorial optimization problem, the traveling salesman problem (TSP), see Fig. 3a. We then turn to a more general (but less visual) set of combinatorial problems, optimization over NK landscapes. We introduce an analysis technique for tracking the evolution of firing patterns as a population, *neural phylogenies* (Fig. 3b,c), and demonstate the ability of recurrent DN to continuously improve upon currently best solutions through beneficial mutations. Next, we formulate fitness in terms of *information gain* in order to make evolution over different landscapes comparable, and we show the time evolution of fitness distribution of the population over different landscapes and initial populations (Fig. 3d). Finally, we demonstrate the existence of a sharp neural-evolutionary *error threshold*, a neural noise level above which information gained by evolution cannot be maintained (Fig. 4a–c), and contrast it with the effect of varying selection strength (Fig. 4d–f) that does not entail a sharp breakdown of the evolutionary process.

We emphasize here that there is nothing inherent in the computational units (reservoirs) or in the specific signal-solution map that makes the system suitable to search in combinatorial spaces. From the point of view of the computational task, these are arbitrary choices. This highlights our point: it is the emergent evolutionary dynamics over the output signals that is responsible for any higher-than-random performance on a combinatorial task.

Breeding traveling salesmen. Given a list of cities, the Traveling Salesman Problem (TSP), asks for the shortest length route that visits each city and returns to the origin. While simple to state, both the rapid growth of the number of possible routes (permutations of the city list) with the number of cities and the difficulty of factor-

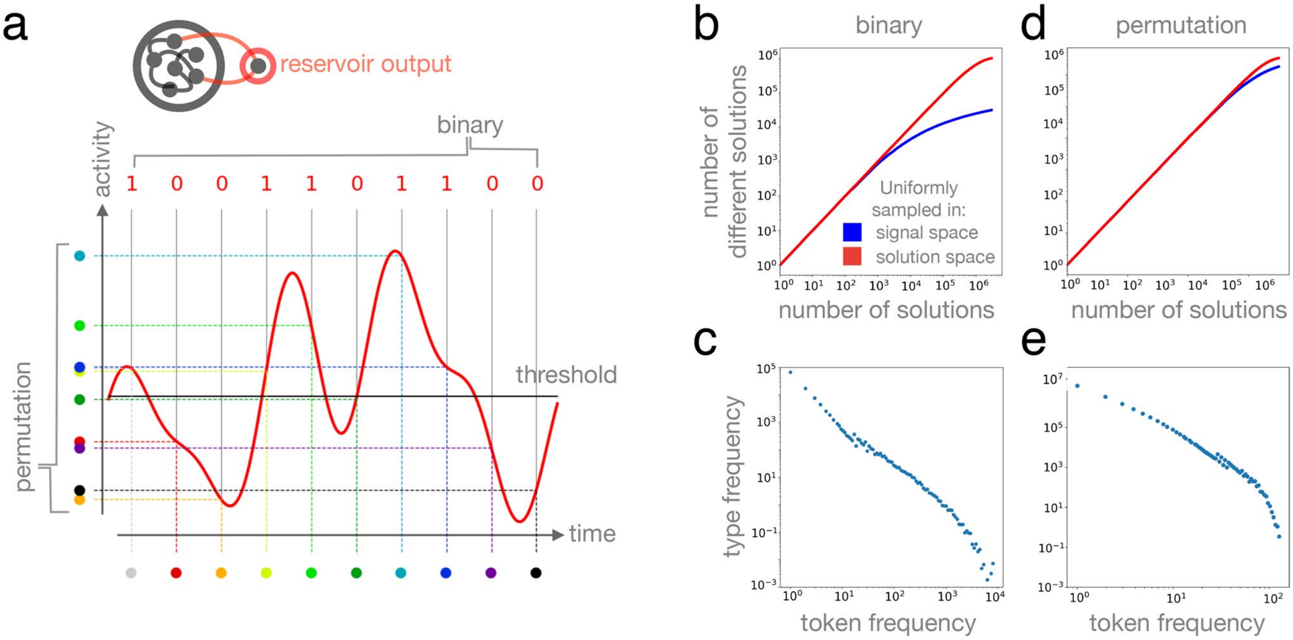

Figure 2. Signal to solution maps and their statistical properties. (**a**) Output signals of reservoirs, shown in red, are mapped to (i) binary sequences by thresholding signal amplitude (i.e., neural activity) at regular time intervals (top), or to permutations by ordering these signal amplitudes. (**b**–**e**) Statistical representation bias of signal to solution maps, showing that a few solutions are represented by many signals, while most solutions are represented by only a few signals. (**b**, **d**) Number of different solutions as a function of the number of sampled solutions. Samples are drawn uniformly in signal space (blue; see "Materials and methods" for details), or in solution space (red). While uniform sampling in solution space saturates exponentially, uniform sampling in signal space leads to a different saturation curve as an effect of some solutions being sampled many times while others are sampled less or not at all. (**c**, **e**) Number of different solutions (types, *y* axis) with a given sampled frequency (token frequency, *x* axis), mirroring representation bias.

ing the problem into simpler sub-problems renders the TSP NP hard. A wide range of algorithmic approaches have been proposed for the TSP, including those inspired by evolutionary dynamics. Here, we select the TSP as a classic example of a difficult combinatorial optimization problem to demonstrate the behaviour of this recurrent DN architecture. The goal of this illustration is not to compete with other specialized algorithms employing TSP specific heuristics, but to highlight the importance of representation and demonstrate the capacity of this DN system.

Figure 2a illustrates one possible encoding of a permutation in a time varying signal; here each coloured dot represents a city identity and the order along the *y* axis represents a route. Notice again that many other possible signals (genotypes) are consistent with a specific route (phenotype): this degeneracy may have important consequences for the *evolvability* of the representation (see "Discussion"). Figure 3a shows a toy example TSP with 10 cities (a solution space size of $\sim 2^{18}$) that is solved via evolutionary dynamics over a 1D array of reservoirs.

Although performance, in general, depends on the initial signals and the inner weights of reservoirs (determining the scope of signals that can be learned by each reservoir), the single run we visualize in Fig. 3a captures an essential feature of any evolutionary dynamics: the automatic reallocation of computational resources to improving upon currently best solutions. After pooling a diverse set of initial conditions, the dynamics automatically narrows down to local improvements around highly promising solutions, culminating, in this example, in the global optimum after 11 generations.

Phylogeny of firing patterns. Although tracking the best solution at all times already offers intuition behind our neuro-evolutionary dynamics, a higher resolution analysis can be given by following the fate of *all* signals as they replicate and mutate, leading to neural *phylogenies* (ancestry trees), visualized in Fig. 3b and c.

We twist conventional phylogenies in two ways. First, since the spatial arrangement of reservoirs (i.e., hosts of replicating signals) is fixed to be a 1D array or a 2D lattice, we simply use this additional information to fix the spatial position of replicators at all times to be those of their host reservoirs. As time (measured in the number of generations) passes vertically, a vertical line corresponds to replicators in the same host reservoir at subsequent generations. Second, in order to keep phylogenies simple, we visualize only the *fitness* of each signal at all times; since the fitness of different signals is likely to be different, this provides a good proxy for the actual phylogeny of signals.

Such neural phylogenies are especially useful in detecting the dynamics of relevant scales of competition. Initially, out of the sea of randomly chosen initial conditions, locally good signals emerge and start to spread. They then form growing homogeneous islands. Higher-fitness islands out compete lower-fitness ones, the relevant scale of competition increases. In the meantime, computations given by local fitness comparisons are allocated to compare higher and higher fitness solutions. Phylogenies also reveal the reservoirs' differential ability to learn;

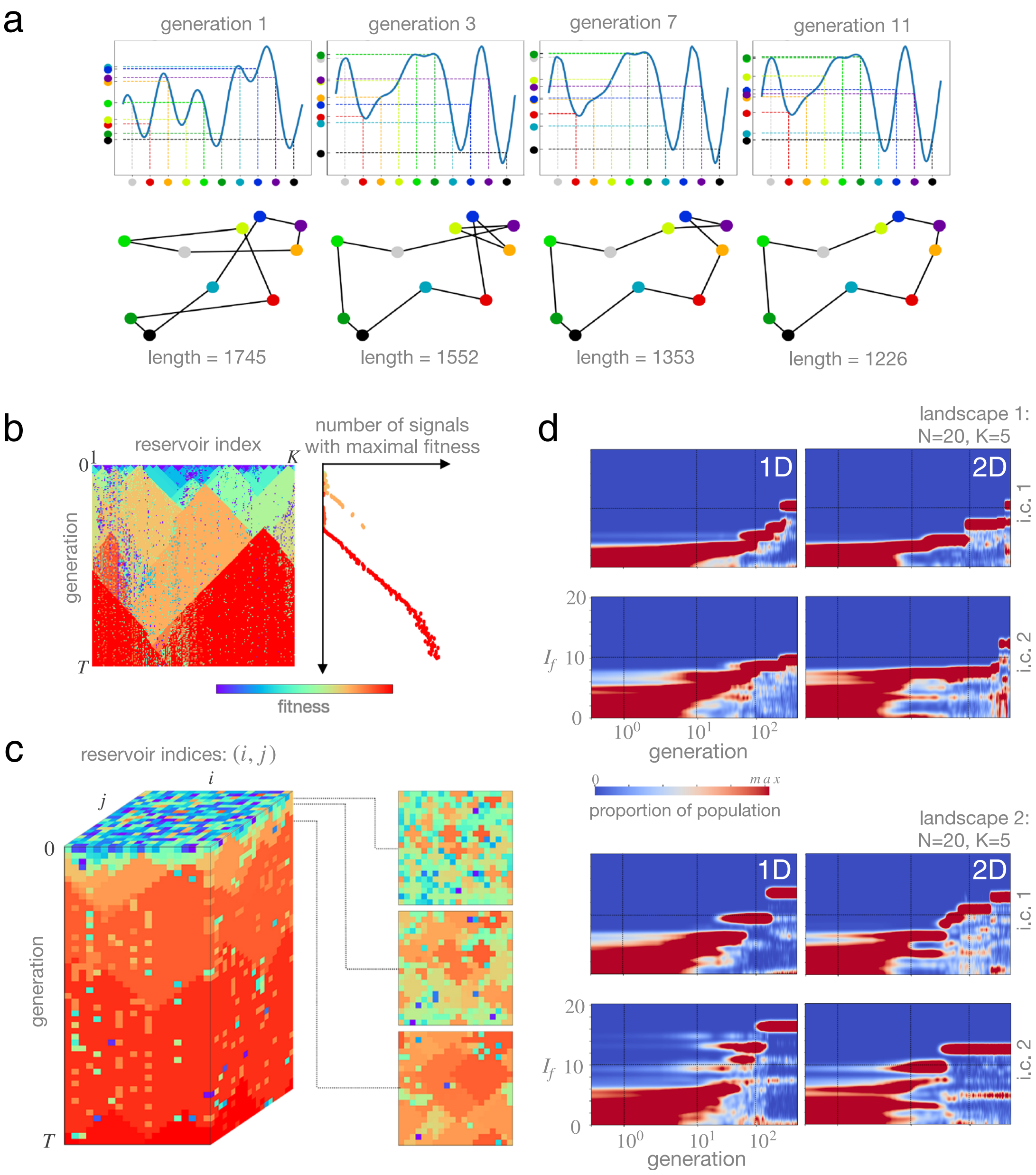

Figure 3. (**a**) Recurrent Darwinian neurodynamics over a 1D array of reservoirs solving the travelling salesman problem (TSP). Shorter paths in solution space correspond to higher fitness in signal space. Top: Highest-fitness signals of their generation. Bottom: Corresponding solutions, permutations representing a path over major Hungarian cities. (**b**) Phylogenies of neural firing patterns. A 1D array of reservoirs is shown horizontally; time evolution proceeds vertically downwards. Color corresponds to the fitness of signals outputted by their reservoirs, evaluated over an NK landscape with $N = 20$ and $K = 5$. Signals with higher fitness spread over local connections between reservoirs. Reservoirs differ in their ability to learn, in extreme cases, they form defects or local walls. (**c**) Phylogeny of firing patterns over a 2D sheet of reservoirs. Cross sections (right) illustrate local spread of high fitness firing patterns, leading to competing homogeneous islands. Sporadic reservoirs with diminished ability to learn do not hinder the process significantly, as opposed to the 1D case. (**d**) Evolutionary information gain I_f over time. In all cases, the population of firing patterns adapts to an NK landscape with $N = 20$ (hence the maximum of I_f is 20) and $K = 5$. We display three levels of between-run variability here. (i) dimensionality of the arrangement of reservoirs (columns); (ii) two realizations of an NK landscape with $N = 20$ and $K = 5$ (top & bottom); (iii) two different initial populations of firing patterns (rows in each block). Although there is a high level of between-run variability across all dimensions, the population of firing patterns keeps finding better and better solutions in all cases, suggesting the feasibility of this process as an efficient computational module.

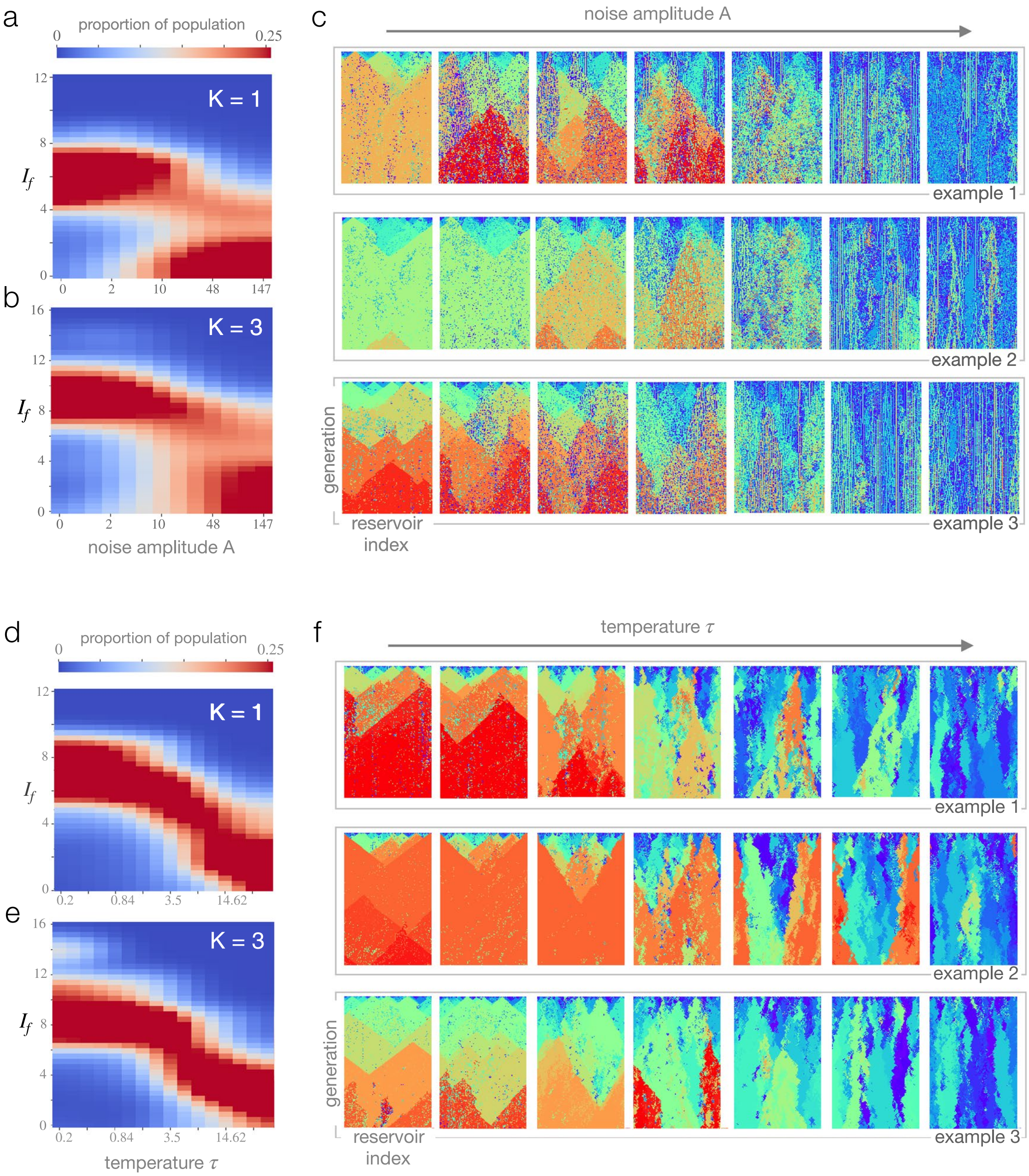

Figure 4. The effect of noisy copying (**a**–**c**) and stochastic selection (**d**–**f**). (**a**, **b**, **e**, **f**) Information gain distribution I_f of the 200th generation, as a function of neural noise amplitude A (**a**, **b**) and of selection temperature τ (**d**, **e**), over a NK landscape with size $N = 20$ and ruggedness $K = 1$ (**a**, **c**) and $K = 3$ (**b**, **d**). Each column shows an average over 5 runs (see "Materials and methods" for details). (**c**, **f**) Neural phylogenies of specific runs over an NK landscape with $N = 20$ and $K = 3$ as a function of increasing neural noise amplitude A (**c**) and selection temperature τ (**f**).

in extreme cases, particularly inflexible reservoirs form a local wall, preventing otherwise high-fitness signals as replicators to spread.

Information gain. Since fitness values f cannot be compared across landscapes, we transform fitness to *information gain* I_f, calculated as $I_f = -\log_2 q_f$, where q_f is the fraction of all types (here, binary sequences of length N) that have higher or equal fitness than f. If the fitness f of the current type is the median of all fitness values, then $I_f = 1$ bit; if f is at 75 percentile, $I_f = 2$ bits, and so on. Figure 3d shows the distribution of information gain I_f in the population as it evolves over NK landscapes with $N = 20$ and $K = 5$. We compare runs over three axes: different population of initial signals, different landscapes, and the dimensionality of the meta-topology of reservoirs (i.e., 1D array or 2D lattice). The high observed between-run variability makes it uninformative to average performance over multiple runs, we therefore present results of single runs. In all runs, information gain increases monotonously. It increases in jumps, corresponding to rare beneficial mutations that improve the currently best solutions. Between jumps, information is maintained: highest-fitness solutions do not disappear from the population unless a higher-fitness mutant appears. This is marked by relatively steady periods of information gain distributions. Finally, islands of lower-fitness solutions are sometimes also kept by the dynamics, as the existence of multiple relative stable modes of the information gain distribution suggests.

Error threshold for Darwinian neurodynamics. The *error threshold* in population genetics refers to a critical mutation rate above which populations diffuse out of fitness peaks to neighbouring genotypes[44]. Although these neighbouring genotypes have lower fitnesses, they are more numerous; lower mutation rate allows for (i) narrower or (ii) lower fitness peaks. Beyond the error threshold, locally good types are not maintained, and consequently, cannot be improved upon by beneficial mutations.

In analogy with population genetics, here we present an error threshold for Darwinian neurodynamics: a critical neural noise value above which the Darwinian evolutionary process over neural replicators breaks down. As shown in Fig. 4a and b, there is indeed a sharp transition from high to low fitness, and correspondingly, high to low information gain I_f as neural the noise level increases. This threshold might provide a useful definition to what *neural replication* means: an approximate copying of firing patterns on top of which a functional Darwinian process, below error threshold, can be built. For such processes, theoretical frameworks and arguments from evolutionary biology and population genetics might be efficiently transferred to understand neural dynamics at a computational and algorithmic level.

In particular, we point out the possibility of an *adaptive* mutation rate in the context of neural replicators, where adaptation can occur both within a single unfolding of the Darwinian neurodynamic process and over genetic evolutionary timescales. Such adaptive mutation rate could be an outcome of selection pressures coming from the outer (genetic) evolutionary loop, optimizing for a specific computational function of the within-brain Darwinian neurodynamic process. Although throughout this paper we consider this computational function to be combinatorial search, here we propose another possibility: that such an evolutionary mechanism implements particle filtering, performing statistical inference. These two computational functions are not exclusive: brains might adaptively control for the mutation rate and the timescale of the process, achieving a “crossover” from one computational domain to the other on demand. Furthermore, this might serve as an explanation to how high-fidelity neural replication appeared over genetic evolutionary timescales in the very first place: through a selection pressure perfecting statistical inference, implemented through particle filters. Such selection pressure might have acted over the computational function of statistical inference (at a high mutation rate regime) way before it co-opted for an efficient Darwinian optimization mechanism at a low mutation rate (below error threshold) regime.

The effect of selection strength. Selection strength is another fundamental concept in evolutionary dynamics. It refers to the magnitude of differential survival in a population in a given environment. In accordance with evolutionary game theory[45,46], we model selection strength by introducing a temperature parameter τ controlling for the stochasticity of the selection process. Zero temperature $\tau = 0$ implies a deterministic selection, i.e., higher-fitness individuals are always selected over less fit ones, whereas infinite temperature $\tau = \infty$ results in random selection: individuals are selected for replication independently of their fitness. Intermediate temperature values interpolate between these two extremes, with selection being correlated with fitness to a given degree. We introduce tunable selection strength through a temperature parameter to our model as follows. For each reservoir unit, we select its teacher unit from its neighborhood in two steps. First, we rank the focal unit and all its neighbors according to the fitness of their output. In case of a 1D and a 2D topology of reservoir units, this means comparing 3 and 5 outputs, respectively. This results in a rank r_i associated with each reservoir i in the neighborhood. Second, we select the teacher reservoir (which might be the same as the focal reservoir) to be reservoir i with probability $p_i \propto \exp(-r_i/\tau)$. Clearly, setting $\tau = 0$ reproduces deterministic selection of the reservoir with maximal fitness. However, it is a question whether increasing τ would result in a smooth deterioration of the evolutionary process or an abrupt one, as we see when neural noise amplitude is increased. We therefore analysed several actual runs through their phylogenetic tree (Fig. 4f) as well as the amount of information the evolutionary process can accumulate in the function of temperature τ (Fig. 4e,f). In contrast to the effect of an increase in neural noise, the effect of a gradual decrease of selection strength (i.e., increase in τ) is smooth. Populations can still maintain some information they accumulated over their evolutionary history even when selection is relatively weak, corresponding to high τ values. Overall this result points at the qualitatively different nature of mutation rate (resulting from neural noise) and selection strength as evolutionary parameters: one resulting in an abrupt, the other in a gradual breakdown of the process.

Discussion

Darwinian neurodynamics links two well-studied and experimentally rooted conceptual systems: neuroscience and evolution. Two principal aims of this endeavor are (i) to ascertain whether or not brains use principles of Darwinian evolution to select actions or representations, and if so, to understand how these principles are reified in neural populations; (ii) to assemble a conceptual bridge over which evolutionary principles can be transferred to neuromorphic learning architectures. These two goals might very well overlap. This paper approaches goals (i) from a reverse engineering perspective, asking how local learning in populations of neurons can sustain evolutionary dynamics at a higher computational level. We discuss the key theoretical issues involved in mapping Darwinian dynamics onto neuronal populations, anchoring our discussion in a simple computational paradigm based on the emergent replication of dynamic firing patterns. We point at the series of decisions one has to make when setting up a DN architecture, from the computational context, in this case combinatorial search problems, all the way down to algorithmic and implementational choices.

Biological evolution has produced spectacular solutions in geological time. Were a similar mechanism operational in the brain on the millisecond timescale, spectacular intellectual achievements would be directly comparable to solutions in evolution. In a remarkable twist an evolutionary mechanism might be even more efficient in the brain than in the wild, for two reasons: first, it can be complemented by other features of the nervous system that are not available for ordinary biological populations; second, even the basic operations such as copying would not be limited by the linearity of informational macromolecules in the neural mechanism. Time will tell whether the latter is an asset or not.

Natural selection can be thought of as an algorithmic process that dynamically reallocates computational resources from globally bad to globally good solutions. It does this in a simple and therefore universally robust way: good solutions are copied in place of bad ones. Many engineering methods are inspired by this idea, notably evolution strategies and estimation of distribution algorithms[47,48]: both iteratively propose new solutions based on the goodness of previously proposed ones. These, like all machine learning algorithms, trade off universality for efficiency in particular cases[49]. Evolution as a dynamic reallocation algorithm, however, is only useful if existing solutions are not washed away by the sea of mutations while searching for better solutions. In other words, although mutations generate the necessary variation upon which selection can act, there is an upper limit, given by the Eigen error threshold in simple combinatorial systems. Mutations generate variation through which the space of possible solutions is explored. If replication and selection act at different levels, connected by e.g. a genotype-phenotype or, in our case, a signal-solution map, then generated variation at the replication level is transformed at the selection level. This leads to a potentially highly non-uniform exploration of phenotypes/solutions. These non-uniformities can take two forms. *i*) Evolvability/facilitated variation: generated variation at the phenotype level is informed by the evolutionary past and proposes solutions that have a higher expected fitness than proposed solutions without this mechanism. *ii*) Limited accessibility of phenotypic regions: phenotypes cannot be represented with equal ease. This effect combines with fitness to determine evolutionary trajectories. The recurrent DN architecture introduced in this paper illustrates these conceptual issues in a specific context, but they are sufficiently general that they are likely relevant to any non-trivial model of DN.

The recurrent DN model exemplifies another core idea: the use of a population of *supervised* learning modules to solve *reinforcement* learning like problems, where information from the environment is only available through a scalar bottleneck, called reward or fitness. Our architecture minimally couples these two dynamics: the role of being a teacher or learner unit is determined via a simple comparison of reward/fitness; variation on which the Darwinian process feeds is introduced 'blindly' through necessarily imperfect supervised training. There is no need for gradient information or another computational module that is aware of the problem structure; the population of agents - in this case, neural firing patterns—explores both locally and globally via emergent Darwinian dynamics. We envisage that this recurrent architecture can be substantially extended and generalized along the lines of these core ideas to more realistic scenarios. One possibility involves adapting the reservoir structure to increase the range of signals that can successfully be learned[50]; recent work suggests that embedding strongly connected cell assemblies within reservoir networks can enable rapid learning of complex signals within a single presentation of the teaching signal[51]. Another important direction involves copying (noisy) higher-dimensional attractors instead of one-dimensional time-varying signals.

One might expect that brains, having evolved under tight energy and space constraints, would employ an evolutionary process, including the maintenance of a population of solutions, only if it provides a qualitative advantage in search problems over non-population based processes. What are these qualitative advantages? Here we provide a subjective account, emphasizing algorithmic features that are unique to evolution, and pointing at their relation to cognitive problem solving.

A key feature of evolution is the unrestrictedness of representations, and the opportunity for open-endedness this unrestrictedness offers[52,53]. Since there is no need for estimating how to improve on the current representation, e.g. by computing gradients or solving complicated credit assignment problems, in evolution, "anything goes". Anything that is, from which assembly instructions can be copied. The more the phenotypes (or candidate solutions, in the language of DN) are decoupled from the replicating substrate that holds the set of instructions for making the phenotype, the more freely the (infinite) phenotype space can be explored. In biology, phenotypes are constructed from genotypes by (probabilistic) developmental programs; in this sense the phenotype is decoupled from the genotype. These programs are sets of instructions that encode incremental and modular construction, they allow for arbitrary re-use and can modify existing modules, paving the royal road to open-endedness. This royal road may well have been stumbled upon by neural systems at some point in their convoluted evolutionary history, replete with complex problem solving. Indeed, probabilistic programs leading to compositional/hierarchical action sequences and the inference of these programs from sensory data offers a successful modeling framework for human cognition[54,55].

Biological evolution goes beyond this, however, by employing an array of clever tricks to boost search in (developmental) program space. Tricks that are not tied to the specificities of life as we know it. These include i) a variety of methods for maintaining a diverse set of solutions[56,57], ii) (re)combining already invented modules at different timescales[58–60], and iii) accelerating change by setting up arms races of various kinds[61,62]. We believe that exploring these substrate independent algorithmic tricks in the context of Darwinian neurodynamics potentially holds unexpected insights along the way.

Materials and methods

Reservoir computing model.

Unit of evolution. In this model of Darwinian neurodynamics, the unit of evolution is the output activity of a reservoir computer. A reservoir computer R consists of *(i)* a reservoir containing n_{neuron} neurons, and *(ii)* a readout consisting of n_{readout} neurons. The recurrent weights between neurons in the reservoir are contained in matrix $\boldsymbol{Q}$, and the feedforward weights from the reservoir to the readout in matrix $\boldsymbol{F}$ (a toy example is shown in Fig. 1a, where $n_{\text{neuron}} = 6$ and $n_{\text{readout}} = 1$). The readout layer is fed back to the reservoir via feedback weights contained in matrix $\boldsymbol{B}$. Only the readout weights $\boldsymbol{F}$ are trained; the recurrent and feedback weights remain fixed throughout training, as in[39]. Throughout all simulations, we set $n_{\text{readout}} = 1$. The output of reservoir R_i, i.e., the (discretized) time-varying activity of its readout neuron, is represented by the vector $\boldsymbol{y}^i$. These output signals are the units of evolution and are parameterized by their respective reservoirs. This parametrization is many-to-one: the same signal can be represented by large volumes of parameter space.

The matrix $\boldsymbol{Q}$ of recurrent weights is a sparse matrix with connection probability p and zero entries along its diagonal; $\boldsymbol{B}$ and $\boldsymbol{F}$ represent fully connected topologies. Weights are initialized as follows:

$$F_{ij} \sim \mathcal{U}(-1,\ 1) \tag{2}$$

$$Q_{ij}, B_{ij} \sim \mathcal{N}\left(0,\ \frac{g}{\sqrt{p * n_{\text{neuron}}}}\right) \tag{3}$$

where g is a scaling constant that controls the presence and/or strength of spontaneous chaotic activity in the network.

Reservoir dynamics are described by the following differential equation:

$$\frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x} + \boldsymbol{Q}\boldsymbol{r} + \boldsymbol{B}\boldsymbol{y} \tag{4}$$

where $\boldsymbol{x}$ is the n_{neuron}-dimensional vector of neuronal activities within the reservoir, $\boldsymbol{r} = \tanh(\boldsymbol{x})$ is the vector of neuronal firing rates, and $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{r}$ is the output of the reservoir (the activity of the readout neuron).

A number $N_{\text{reservoir}}$ of reservoir computers are connected to form a network in which the output of one reservoir may act as the input of another. Each node in this network is therefore a reservoir computing unit, R_i where $i = 1, ..., N_{\text{reservoir}}$, and a link between nodes i and j represents the idea that R_i and R_j may each either receive input from or send output activity to the other. In this sense one network may act as a *teacher* and the other a *learner*. Note that we describe a network of networks, where the higher level connectivity between units R_i is denoted $\boldsymbol{\Upsilon}$ and the lower level connectivity describes the recurrent, output and feedback weights (contained in matrices $\boldsymbol{Q}^i$, $\boldsymbol{F}^i$ and $\boldsymbol{B}^i$ respectively) in each R_i. For $\boldsymbol{\Upsilon}$, we consider one and two dimensional regular lattices where only neighbouring reservoir units are connected - in the 1D case, each node is connected to two others and in the 2D case, to four others. These lattices wrap around such that there are no boundaries.

Initialization. In a typical run, each reservoir unit R_i is initialized by training the output weights $\boldsymbol{F}^i$ such that R_i outputs a randomly assigned signal. This initialization signal, $\boldsymbol{y}^{i,0}$, is constructed by drawing n_{Fourier} sine coefficients a_m and n_{Fourier} cosine coefficients b_m from a uniform distribution in $[-0.5, 0.5]$, to obtain

$$\boldsymbol{y}^{i,0}(t) = \sum_{m=1}^{n_{\text{Fourier}}} a_m \sin(mt) + b_m \cos(mt) \tag{5}$$

This signal is then rescaled along the y axis to fix its range to a given value A. The FORCE algorithm[39] is then used to train output weights F^i.

Replication. In any given generation t, each R_i can either act as a *learner*, i.e. learn the output of another unit R_j, or as a *teacher*. Output weights of the learner unit are trained via FORCE such that the output activity pattern is, in effect, imperfectly copied from teacher to learner ($\boldsymbol{y}^{j,t} \quad \rightarrow \quad \boldsymbol{y}^{i,t+1}$).

The FORCE algorithm is a supervised learning method, therefore it is necessary to provide an explicit target signal. Here the output signal $\boldsymbol{y}^j$ acts as the target for R_i during training. The target signal is presented to the learner one single time at full length. Using the recursive least squares (RLS) algorithm, the learning rule is as follows:

$$\frac{d\boldsymbol{F}}{dt} = -\boldsymbol{e}\frac{\boldsymbol{P}\boldsymbol{r}}{1 + \boldsymbol{r}^T\boldsymbol{P}\boldsymbol{r}} \tag{6}$$

where the error vector, $\boldsymbol{e}$, is

$$\boldsymbol{e}(t) = \boldsymbol{y}^i(t) - \boldsymbol{y}^j(t) \tag{7}$$

and $\boldsymbol{P}$ is an $n_{\text{neuron}} * n_{\text{neuron}}$ matrix which is the running estimate of the inverse of the correlation matrix of neural activities plus a regularization term. It is updated by the following rule

$$\frac{d\boldsymbol{P}}{dt} = -\frac{\boldsymbol{P}\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{P}}{1 + \boldsymbol{r}^T\boldsymbol{P}\boldsymbol{r}} \tag{8}$$

The initial value of $\boldsymbol{P}$ is

Number of neurons	n_{neuron}	1000
Number of readout neurons	n_{readout}	1
Connection probability of sparse $\boldsymbol{Q}$ matrix	p	0.1
Chaocity parameter	g	1.5
Timestep	dt	0.1
Signal generation time	t_{signal}	300
Learning rate of FORCE algorithm	α	1
Training time	t_{train}	300
Evaluation time	$t_{\text{evaluation}}$	30
Number of sine and cosine coefficients of initial signal	n_{Fourier}	5

Table 2. Fixed parameters.

	Fig. 3a	**Fig. 3b**	**Fig. 3c**	**Fig. 3d**	**Fig. 4a**	**Fig. 4b**	**Fig. 4c**	**Fig. 4d**	**Fig. 4e**	**Fig. 4f**
Number of reservoirs $N_{\text{reservoir}}$	100	200	400	100	100	100	100	100	100	100
Number of generations T	100	200	100	500	200	200	200	200	200	200
N	None	20	20	20	20	20	20	20	20	20
K	None	5	3	5	1	3	3	1	3	3
Topology	2D	1D	2D	1D, 2D	1D	1D	1D	1D	1D	1D
Noise amplitude A	0	0	0	1.08	Multiple values in [0, 147]			0	0	0
Temperature τ	0	0	0	0	0	0	0	Multiple values in [0, 42.75]		

Table 3. Parameters that are varied.

$$\boldsymbol{P}(0) = \frac{\boldsymbol{I}_{n_{\text{neuron}}}}{\alpha} \tag{9}$$

where $\boldsymbol{I}_{n_{\text{neuron}}}$ is the identity matrix and α acts as the learning rate of the FORCE algorithm.

Selection. The direction of training between two reservoirs is determined by the relative fitness of the two output signals. Reservoir outputs are periodic signals; one period of the signal is taken for evaluation. Each signal/genotype is mapped to a solution/phenotype via a neural genotype-phenotype (GP) map. Here we consider two deterministic GP maps: (i) a signal vector, $\boldsymbol{y}^i$ is mapped to a binary string, $\boldsymbol{v}^i$, of length N, where i indexes the signals (see Fig. 2a above), and (ii) $\boldsymbol{y}^i$ to permutation vector $\boldsymbol{h}^i$ (see Fig. 2a along the activity axis), where the ordering is defined by the magnitude of the signal at bin edges. The fitness landscape is a map from the vector valued phenotype, either $\boldsymbol{v}^i$ or $\boldsymbol{h}^i$, to a scalar fitness, f^i. This mapping depends on the problem under consideration: the TSP and NK models are two examples described in this paper.

Variation. The copying of signals from one reservoir to another is typically imperfect: this process introduces 'mutations' into the copied signal, providing variation upon which selection can act. The fidelity of copying between reservoir units varies significantly among signals; unfortunately, this is not readily controlled with the FORCE algorithm. One approach to controlling the mutation rate involves injecting additional noise into the copied signal (see Fig. 4). In particular, here we add Gaussian noise $\eta(t)$ with various amplitudes A to the signal of the learner reservoir after training. The noise amplitude A is defined as

$$A = \int_0^{t_{\text{evaluation}}} \eta^2 \, dt. \tag{10}$$

Darwinian dynamics. A generation is defined as one round of signal copying; in this round each reservoir output signal is compared with its neighbouring reservoirs. In the 1D cases (Fig. 3a,b,d and Fig. 4a–f), $2N_{\text{reservoir}}$ comparisons are made in parallel in each generation as reservoir R_i is compared to R_{i-1} and R_{i+1}. In case of the 2D topology from Fig. 3c,d , this number is $4N_{\text{reservoir}}$, as each reservoir is compared to its neighbours on the square grid, 2–2 by each dimension. In deterministic selection (Fig. 3 and 4a–c) scenarios, the reservoir with the best signal is selected as a teacher reservoir; in case of stochastic selection (Fig. 4 d–f), first reservoirs are ranked according to the fitness of their output signal, where rank of reservoir i is denoted by r_i, and then reservoir i is selected as the teacher reservoir with probability

$$p_i = \frac{\exp(r_i/\tau)}{\sum_j \exp(r_j/\tau)} \tag{11}$$

where j iterates over the neighborhood as defined above and τ is a “temperature” parameter controlling for the stochasticity of the selection process.

This dynamics is run for a total of T generations.

Simulation parameters. Those parameters that were fixed throughout all simulations shown in the paper are described in Table 2, while the parameters that were varied between simulations are described in Table 3.

Python code for all simulations is available at https://github.com/csillagm/reservoir-dn.

Received: 12 February 2021; Accepted: 21 May 2021
Published online: 15 June 2021

References

1. Darwin, C. *On the Origin of Species, 1859* (Routledge, 2004).
2. Smith, J. M. *The Problems of Biology* (Springer, 1986).
3. Müller, V., De Boer, R. J., Bonhoeffer, S. & Szathmáry, E. An evolutionary perspective on the systems of adaptive immunity. *Biol. Rev.* **93**(1), 505–528 (2018).
4. Fernando, C. T., Szathmáry, E. & Husbands, P. Selectionist and evolutionary approaches to brain function: A critical appraisal. *Front. Comput. Neurosci.* **6**(24), 24 (2012).
5. Szilágyi, A., Zachar, I., Fedor, A., de Vladar, H. P. & Szathmáry, E. Breeding novel solutions in the brain: A model of Darwinian neurodynamics. *F1000 Res.* **5**, 2416 (2016).
6. Watson, R. A. *et al.* Evolutionary connectionism: Algorithmic principles underlying the evolution of biological organisation in evo-devo, evo-eco and evolutionary transitions. *Evol. Biol.* **43**(4), 553–581 (2016).
7. Kouvaris, K., Clune, J., Kounios, L., Brede, M. & Watson, R. A. How evolution learns to generalise: Using the principles of learning theory to understand the evolution of developmental organisation. *PLoS Comput. Biol.* **13**, 4 (2017).
8. Power, D. A. *et al.* What can ecosystems learn? Expanding evolutionary ecology with learning theory. *Biol. Direct* **10**(1), 69 (2015).
9. Watson, R. A. & Szathmáry, E. How can evolution learn?. *Trends Ecol. Evol.* **31**(2), 147–157 (2016).
10. Harper, M. The replicator equation as an inference dynamic. arXiv:0911.1763 (2009).
11. Shalizi, C. R. *et al.* Dynamics of bayesian updating with dependent data and misspecified models. *Electron. J. Stat.* **3**, 1039–1074 (2009).
12. Suchow, J. W., Bourgin, D. D. & Griffiths, T. L. Evolution in mind: Evolutionary dynamics, cognitive processes, and bayesian inference. *Trends Cogn. Sci.* **21**(7), 522–530 (2017).
13. Czégel, D., Giaffar, H., Zachar, I., Tenenbaum, J. B. & Szathmáry, E. Evolutionary implementation of Bayesian computations. *BioRxiv* **1**, 685842 (2020).
14. Czégel, D., Zachar, I. & Szathmáry, E. Multilevel selection as Bayesian inference, major transitions in individuality as structure learning. *R. Soc. Open Sci.* **6**, 8 (2019).
15. Akyıldız, Ö. D. A probabilistic interpretation of replicator-mutator dynamics. arXiv:1712.07879 (2017).
16. Campbell, J. O. Universal Darwinism as a process of Bayesian inference. *Front. Syst. Neurosci.* **10**, 49 (2016).
17. Frank, S. A. Natural selection v. how to read the fundamental equations of evolutionary change in terms of information theory. *J. Evol. Biol.* **25**(12), 2377–2396 (2012).
18. Knill, D. C. & Pouget, A. The bayesian brain: The role of uncertainty in neural coding and computation. *Trends Neurosci.* **27**(12), 712–719 (2004).
19. Doya, K., Ishii, S., Pouget, A. & Rao, R. P. N. *Bayesian Brain: Probabilistic Approaches to Neural Coding* (MIT Press, 2007).
20. Friston, K. The free-energy principle: A unified brain theory?. *Nat. Rev. Neurosci.* **11**(2), 127 (2010).
21. Rao, R. P. N. Decision making under uncertainty: A neural model based on partially observable markov decision processes. *Front. Comput. Neurosci.* **4**, 146 (2010).
22. Sanborn, A. N. & Chater, N. Bayesian brains without probabilities. *Trends Cogn. Sci.* **20**(12), 883–893 (2016).
23. Sella, G. & Hirsh, A. E. The application of statistical physics to evolutionary biology. *Proc. Natl. Acad. Sci. USA* **102**(27), 9541–9546 (2005).
24. Fedor, A. *et al.* Cognitive architecture with evolutionary dynamics solves insight problem. *Front. Psychol.* **8**, 427 (2017).
25. Öllinger, M., Volz, K. & Szathmáry, E. Editorial for the research topic: Insight and intuition-two sides of the same coin?. *Front. Psychol.* **9**, 689 (2018).
26. Gerald, M. *Edelman. Neural Darwinism. The Theory of Neuronal Group Selection* (Basic Books, 1987).
27. Edelman, G. M. Neural Darwinism: Selection and reentrant signaling in higher brain function. *Neuron* **10**(2), 115–125 (1993).
28. Smith, J. M. *The Problems of Biology* (Oxford University Press, 1986).
29. Changeux, J.-P., Courrège, P. & Danchin, A. A theory of the epigenesis of neuronal networks by selective stabilization of synapses. *Proc. Natl. Acad. Sci.* **70**(10), 2974–2978 (1973).
30. Fernando, C., Vasas, V., Szathmáry, E. & Husbands, P. Evolvable neuronal paths: A novel basis for information and search in the brain. *PLoS ONE* **6**, 8 (2011).
31. Auerbach, J., Fernando, C. & Floreano, D. Online extreme evolutionary learning machines. In *Artificial Life Conference Proceedings 14*, 465–472. (MIT Press, 2014).
32. Michod, R. E. Darwinian selection in the brain. *Evolution* **43**(3), 694–696 (1989).
33. Fernando, C., Goldstein, R. & Szathmáry, E. The neuronal replicator hypothesis. *Neural Comput.* **22**(11), 2809–2857 (2010).
34. Fernando, C., Karishma, K. K. & Szathmáry, E. Copying and evolution of neuronal topology. *PLoS ONE* **3**, 11 (2008).
35. Szathmáry, E. The evolution of replicators. *Philos. Trans. R. Soc. Lond. B* **355**(1403), 1669–1676 (2000).
36. Szathmáry, E. Toward major evolutionary transitions theory 2.0. *Proc. Natl. Acad. Sci.* **112**(33), 10104–10111 (2015).
37. Steels, L. & Szathmáry, E. The evolutionary dynamics of language. *Biosystems* **164**, 128–137 (2018).
38. Kauffman, S. A. & Weinberger, E. D. The nk model of rugged fitness landscapes and its application to maturation of the immune response. *J. Theor. Biol.* **141**(2), 211–245 (1989).
39. Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**(4), 544–557 (2009).
40. Kauffman, S. A. *et al. The Origins of Order: Self-organization and Selection in Evolution* (Oxford University Press, 1993).
41. Kauffman, S. & Levin, S. Towards a general theory of adaptive walks on rugged landscapes. *J. Theor. Biol.* **128**(1), 11–45 (1987).
42. Kaznatcheev, A. Computational complexity as an ultimate constraint on evolution. *Genetics* **212**(1), 245–265 (2019).
43. Lieberman, E., Hauert, C. & Nowak, M. A. Evolutionary dynamics on graphs. *Nature* **433**(7023), 312–316 (2005).
44. Eigen, M., McCaskill, J. & Schuster, P. The molecular quasi-species. *Adv. Chem. Phys.* **75**, 149–263 (1989).

45. Szabó, G. & Tőke, C. Evolutionary prisoners dilemma game on a square lattice. *Phys. Rev. E* **58**(1), 69 (1998).
46. Zhang, Y., Ning, H., Wang, J. & Xia, C. Coveting the successful neighbor promotes the cooperation for the spatial public goods game on two-layered lattices. *Chaos Solitons Fractals* **105**, 29–37 (2017).
47. Beyer, H.-G. & Schwefel, H.-P. Evolution strategies: A comprehensive introduction. *Nat. Comput.* **1**(1), 3–52 (2002).
48. Larrañaga, P. & Lozano, J. A. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation* Vol. 2 (Springer, 2001).
49. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997).
50. Recanatesi, S., Ocker, G. K., Buice, M. A. & Shea-Brown, E. Dimensionality in recurrent spiking networks: Global trends in activity and local origins in connectivity. *PLOS Comput. Biol.* **15**, 1–29 (2019).
51. Dasgupta, S., Kulvicius, T., Wörgötter, F. & Tetzlaff, C. The use of Hebbian cell assemblies for nonlinear computation. *Sci. Rep.* **5**, 10 (2015).
52. Stanley, K. O. Why open-endedness matters. *Artif. Life* **25**(3), 232–235 (2019).
53. de Vladar, H. P., Santos, M. & Szathmáry, E. Grand views of evolution. *Trends Ecol. Evol.* **32**(5), 324–334 (2017).
54. Lake, B. M., Salakhutdinov, R. & Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science* **350**(6266), 1332–1338 (2015).
55. Botvinick, M. M. Hierarchical reinforcement learning and decision making. *Curr. Opin. Neurobiol.* **22**(6), 956–962 (2012).
56. Pásztor, L., Botta-Dukát, Z., Magyar, G., Czárán, T. & Meszéna, G. *Theory-Based Ecology: A Darwinian Approach* (Oxford University Press, 2016).
57. Szilágyi, A. *et al.* Ecology and evolution in the rna world dynamics and stability of prebiotic replicator systems. *Life* **7**(4), 48 (2017).
58. Schlosser, G. & Wagner, G. P. *Modularity in Development and Evolution* (University of Chicago Press, 2004).
59. Livnat, A., Papadimitriou, C., Pippenger, N. & Feldman, M. W. Sex, mixability, and modularity. *Proc. Natl. Acad. Sci.* **107**(4), 1452–1457 (2010).
60. Smith, J. M. & Szathmáry, E. *The Major Transitions in Evolution* (Oxford University Press, 1997).
61. Van Valen, L. A new evolutionary law.. *Evol. Theor.* **1**, 1–30 (1973).
62. Crespi, B. J. Vicious circles: Positive feedback in major evolutionary and ecological transitions. *Trends Ecol. Evol.* **19**(12), 627–633 (2004).

Acknowledgements

We would like to thank Thomas Filk, Sergey Shuvaev, Sashank Pisupati, and András Szántó for useful discussions. We also thank Tollevin Williams for help producing the figures. This research was supported by the National Research, Development, and Innovation Office under 'Theory and solutions in the light of evolution' GINOP-2.3.2-15-2016-00057 (D.C., M.C., B.F., and E.S.), and 'Learning and evolution' KKP_19_129848 (D.C. and E.S.), and the Templeton World Charity Foundation under grant number TWCF0268 (D.C. and E.S.).

Author contributions

D.C., E.S. conceived the idea, D.C, H.G., M.C., B.F. designed and performed research, H.G., M.C., B.F. implemented and ran simulations, D.C. wrote the first draft of the manuscript, D.C., H.G., M.C., E.S. wrote the final version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to D.C. or E.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.