NEAT-DNFs: A NeuroEvolutionary Framework for Evolving Dynamic Neural Field Architectures

João G. Cunha
Universidade do Minho, Center Algoritmi
Guimarães, Portugal
id10787@uminho.pt

Wolfram Erlhagen
Universidade do Minho, Centre of Mathematics
Guimarães, Portugal
wolfram.erlhagen@math.uminho.pt

Raymond H. Cuijpers
Eindhoven University of Technology, Human-Technology Interaction Group
Eindhoven, Netherlands
r.h.cuijpers@tue.nl

Estela Bicho
Universidade do Minho, Centre Algoritmi
Guimarães, Portugal
estela.bicho@dei.uminho.pt

Abstract

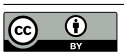
Dynamic Neural Fields (DNFs) offer a biologically grounded framework for modelling continuous-time neural dynamics underlying a wide range of cognitive and sensorimotor functions. Despite their expressive power, DNF architectures are typically designed and tuned manually, as their nonlinear dynamics and kernel-based interactions make systematic design and parameterisation difficult. As a result, constructing functional DNF models requires expert knowledge and iterative trial-and-error. We introduce NEAT-DNFs, a neuroevolutionary framework that extends NeuroEvolution of Augmenting Topologies (NEAT) to the automated synthesis of DNF architectures. In NEAT-DNFs, neural fields and inter-field interactions are encoded as evolvable genes, enabling the joint evolution of intrinsic parameters and architectural topology. This allows continuous-time neural dynamics to be discovered autonomously, without manual tuning. We evaluate NEAT-DNFs on a hierarchy of benchmark tasks of increasing complexity. Initial tasks target core Dynamic Field Theory (DFT) mechanisms, including detection, working memory, and selection. We then consider more demanding problems that require the emergence of intermediate neural fields, thereby testing the framework's capacity for structural innovation. Across all tasks, NEAT-DNFs reliably evolves compact and stable architectures, increasing structural complexity only when necessary. This work bridges neuroevolution and DFT, establishing a foundation for the automated design of biologically inspired neural field models.

CCS Concepts

• **Computing methodologies** → **Randomized search**; *Bio-inspired approaches.*

Keywords

NEAT, Neuroevolution, Dynamic Field Theory, Dynamic Neural Fields, Evolutionary Neural Networks, Structural Evolution



GECCO '26, San Jose, Costa Rica

ACM ISBN 979-8-4007-2487-9/2026/07
https://doi.org/10.1145/3795095.3805169

ACM Reference Format:
João G. Cunha, Wolfram Erlhagen, Raymond H. Cuijpers, and Estela Bicho. 2026. NEAT-DNFs: A NeuroEvolutionary Framework for Evolving Dynamic Neural Field Architectures. In *Genetic and Evolutionary Computation Conference (GECCO '26), July 13–17, 2026, San Jose, Costa Rica.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3795095.3805169

1 Introduction

By representing neural activity as spatially distributed activation fields shaped by nonlinear interaction kernels, Dynamic Neural Fields (DNFs) provide a principled and interpretable account of population-level neural dynamics [8, 23]. These properties have made DNF architectures attractive for modelling cognitive processes and sensorimotor control in embodied agents [5, 7, 20, 27, 29].

Despite their expressive power, DNF systems are typically handcrafted: designers specify both field parameters (e.g., time constants, resting levels, kernel profiles) and architecture (the number of fields and their couplings). Because DNF dynamics are highly nonlinear and sensitive to parameter changes, this process is time-consuming, hard to transfer across tasks, and biased by prior assumptions [13], limiting scalability and broader adoption.

Evolutionary computation offers a natural alternative. By searching directly in the space of dynamical systems, evolutionary methods can discover parameter configurations and structures that give rise to desired behaviours without requiring explicit design [11]. Prior work has shown that evolutionary strategies can optimise DNF parameters [10, 13, 19], but these approaches assume a fixed topology, leaving the more fundamental problem of architectural discovery unaddressed.

In parallel, neuroevolution has produced methods that evolve both parameters and topology in Artificial Neural Networks (ANNs) [4, 6, 17]. Among these, NeuroEvolution of Augmenting Topologies (NEAT) enables incremental complexification while protecting innovation through speciation [25], and has been widely validated for discrete-time, weight-based networks [18]. Extending NEAT to DNFs, however, is non-trivial: continuous-time NEAT variants for Continuous Time Recurrent Neural Networks (CTRNNs) exist, but they still evolve nodes and scalar synaptic weights [16, 18], whereas DNFs are spatially extended, kernel-coupled dynamical systems whose computation arises from interaction kernels over continuous feature dimensions. Consequently, genome encodings, mutation

operators, and compatibility metrics must be reformulated in terms of fields and kernels, not neurons and weights.

In this paper, we introduce NEAT-DNFs, a neuroevolutionary framework that extends NEAT to the evolution of DNF architectures. Genomes encode both intrinsic field dynamics and kernel-defined inter-field couplings, enabling joint optimisation of spatiotemporal dynamics and topology via parametric mutation and structural operators that add fields and interactions. We evaluate whether NEAT-DNFs can (i) tune intra-field dynamics, (ii) discover functional inter-field transformations, and (iii) introduce hidden fields only when intermediate processing is required. These questions are tested on a progression of tasks with increasing dynamical and architectural demands, chosen to reflect core DFT mechanisms (detection, working memory, and selection instabilities) as well as two established paradigms, Delayed Match-to-Sample (DMTS) [15] and Inhibition of Return (IOR) [12]. By analysing evolved architectures and their resulting dynamics, we assess both behavioural performance and how task-driven topological complexity emerges.

The remainder of this paper is organised as follows. Section 2 reviews background on NEAT and DNFs. Section 3 presents NEAT-DNFs, including its encoding and evolutionary operators. Section 4 describes the evaluation tasks, and Section 5 reports the results. Sections 6 and 7 discuss the implications and limitations of the approach and conclude with directions for future work.

2 Background

2.1 NeuroEvolution and NEAT

Neuroevolution applies evolutionary algorithms to the optimisation of neural systems, historically focusing on ANNs. Early work often evolved connection weights under fixed topologies, which can limit scalability by constraining representational capacity and imposing designer biases [11, 28].

NEAT introduced a constructive alternative by evolving both parameters and topology simultaneously [25]. It begins with minimal networks and incrementally complexifies them through mutations that add nodes and connections. Historical markings enable meaningful crossover between differing topologies, while speciation protects novel structures during optimisation. This combination has made NEAT a foundational method in neuroevolution and a basis for many subsequent variants [18].

Later extensions adapted NEAT ideas to broader representations, including indirect encodings and substrate-based approaches (e.g., HyperNEAT) [24] and spatially organised field structures [14]. However, these methods are still primarily formulated for point-neuron networks in which computation arises from scalar synaptic weights (in discrete or continuous time). They therefore do not directly transfer to neural field models, where behaviour depends on continuous feature dimensions and interaction kernels rather than scalar weights between neurons.

2.2 Dynamic Field Theory and Dynamic Neural Fields

DFT provides a mathematical framework for modelling neural population dynamics as continuous-time dynamical systems defined over continuous feature spaces. Rather than representing computation through discrete neuron activations, DFT characterises neural processing in terms of the evolution of spatially distributed activation fields whose behaviour emerges from local excitation, lateral inhibition, and external input [8, 23].

At the core of DFT are DNFs, which describe the temporal evolution of activation over a continuous dimension. An activation field $u(x,t)$, defined over a spatial or feature dimension x, evolves according to a nonlinear differential equation [2]:

$$\tau \dot{u}(x,t) = -u(x,t) + h + s(x,t) + \int k_{uu}(x-x')g(u(x',t))dx' + q\xi(x,t) \quad (1)$$

Here $u(x,t)$ represents the average membrane potential at time t at neural position x, where x belongs to a continuous subset of $\mathbb{R}$ [7]. The resting level $h(<0)$ sets the baseline activation of the field, while $s(x,t)$ represents time-dependent external input originating either from sensory stimulation or from other neural fields. The time constant τ governs the intrinsic temporal dynamics of the field, determining how rapidly activation evolves in response to input.

Nonlinearity arises through the output function $g(u)$ (implemented as a sigmoid with steepness $\beta = 5.0$), which ensures that only sufficiently activated neural populations contribute to recurrent interactions. The interaction kernel $k_{uu}(x,x')$ defines the strength and spatial extent of coupling between neural populations at positions x and x'. Finally, ξ represents Gaussian white noise, with amplitude q, which captures stochastic neural variability.

The interaction kernel largely determines a field's dynamical regime. In this work, we consider two standard kernel parameterisations used in DFT: a Gaussian profile and a Mexican-hat profile implemented as a difference of Gaussians [23], both with global inhibition. Commonly, Gaussian kernels with global inhibition are chosen to promote competitive selection, whereas Mexican-hat kernels are used to support multi-peak dynamics and sustained activation [9, 26, 27]. These associations, however, are not hard constraints: both kernel types can realise selection, multi-peak, or sustained regimes depending on parameter settings (e.g., excitation/inhibition balance and the strength of global inhibition). Because NEAT-DNFs explores a broad parameter space, we treat the kernel type as a structural choice while allowing evolution to determine the resulting regime through parameter optimisation.

We define the Gaussian kernel with global inhibition as:

$$k_{uu}(x-x') = A\exp\left(-\frac{|x-x'|^2}{\sigma^2}\right) - A_{\text{glob}} \quad (2)$$

where A and σ control the amplitude and spatial spread of local excitation, and A_{glob} provides approximately uniform inhibition. Increasing A_{glob} suppresses competing activations and can enforce winner-take-all selection.

We parameterise the Mexican-hat kernel as a difference of Gaussians with global inhibition:

$$k_{uu}(x-x') = A_{\text{exc}}\exp\left(-\frac{|x-x'|^2}{\sigma_{\text{exc}}^2}\right) - A_{\text{inh}}\exp\left(-\frac{|x-x'|^2}{\sigma_{\text{inh}}^2}\right) - A_{\text{glob}}, \quad (3)$$

where local excitation ($A_{\text{exc}}, \sigma_{\text{exc}}$) is balanced by broader lateral inhibition ($A_{\text{inh}}, \sigma_{\text{inh}}$). Depending on the excitation-inhibition balance and global inhibition, this type can stabilise one or more activation peaks after input removal, but can also yield competitive selection. Overall, smooth variation of these parameters enables transitions between dynamical regimes, making kernel parameters natural targets for evolutionary optimisation.

Complex architectures are constructed by coupling multiple neural fields. When a field u projects to a field v, the output from u provides spatially structured input to v via convolution with an interaction kernel [29]:

$$s_v(x,t) = \int k_{uv}(x - x')g(u(x',t))dx' \tag{4}$$

Here $s_v(x,t)$ represents the input to field v, and $k_{uv}(x - x')$ defines how activation at position x' in field u influences position x in field v, [9, 26, 27]. As with self-interactions, inter-field kernels may adopt Gaussian or Mexican-hat profiles (Equations 2 and 3).

2.3 Prior Evolution of Dynamic Neural Fields

Several studies have explored evolutionary and stochastic optimisation for tuning parameters of neural field models [3, 10, 13, 19, 21]. Collectively, these works provide important proof-of-concept results, demonstrating that evolutionary optimisation can be used to parametrise DNFs.

However, their scope is fundamentally limited. The architectures considered consist of a single neural field or a small, manually specified set of coupled fields, and evolution operates exclusively on parameter values. The structure of the system itself remains fixed and hand-designed. From an evolutionary computation perspective, this is a critical limitation, as many cognitive and control functions emerge not only from parameter values but from the interaction among different neural populations; this gap motivates the development of NEAT-DNFs.

3 Algorithm Description

As in NEAT, evolution proceeds through a population of genomes that undergo mutation, crossover, and selection, with innovation protected via historical markings and speciation. However, the representation of individuals, the mutations, and the compatibility distance metric are redesigned to operate directly on DNFs.

Each solution in NEAT-DNFs encodes a DNF-based architecture composed of multiple interacting neural fields. Evolution begins from an ultra-minimal configuration, consisting of just randomly parameterised input and output fields with no inter-field connections. From this starting point, evolution incrementally complexifies architectures by introducing new interactions and intermediate fields only when they provide a fitness advantage.

3.1 Genome Encoding

The genome in NEAT-DNFs consists of two gene types: *field genes* and *interaction genes*. Together, they encode both the intrinsic parameters of individual neural fields and how they are connected, as shown in Figure 1.

3.1.1 Field Genes. Each field gene represents a single DNF and encodes: (i) a unique identifier; (ii) a categorical role (input, hidden, or output); (iii) intrinsic field parameters, including the time constant τ, resting level h, and the type and parameters of the self-interaction kernel. Input fields receive external stimuli, output fields generate the model's observable responses, and hidden fields are internally evolved units that mediate processing and have no direct interaction with the external environment. Field genes, therefore, define the *intra-field dynamics* that govern how activation evolves over time and space. Since the choice of kernel type itself is evolvable, through evolution, fields may exhibit qualitatively different computational regimes, such as winner-take-all selection, sustained working memory, or single/multi-peak representations

3.1.2 Interaction Genes. Interaction genes encode the connections between pairs of fields. Each interaction gene specifies: (i) the source and target field identifiers; (ii) the type and parameters of the self-interaction kernel; (iii) an enabled/disabled flag; and (iv) a historical innovation number. When enabled, an interaction gene induces a convolution-based projection from the source field to the target field, shaping how activity propagates between fields. Interaction kernels may be excitatory or inhibitory and may adopt Gaussian or Mexican-hat profiles. As with field genes, kernel parameters are subject to evolutionary optimisation, allowing evolution to discover both the strength and spatial structure of inter-field communication.

3.2 Evolutionary Operators

NEAT-DNFs supports both parametric and structural mutations, allowing evolution to refine existing internal field dynamics while progressively expanding architectural complexity as needed.

3.2.1 Parametric Mutations. Parametric mutations operate on existing genes without modifying the underlying topology. For **field genes**, these mutations adjust intrinsic dynamical properties, including the resting level h, time constant τ, and the parameters of the self-interaction kernel. Evolution may also switch the kernel type between Gaussian and Mexican-hat profiles. Kernel parameter values are incremented or decremented, enabling the gradual refinement of field behaviour.

For **interaction genes**, parametric mutations similarly modify the parameters of inter-field interaction kernels. These include adjustments to kernel amplitudes and widths, changes in kernel type, and inversions of interaction polarity between excitatory and inhibitory influences.

3.2.2 Structural Mutations. Three different structural mutations modify the topology of the architecture. One operator *adds a new interaction* between two previously unconnected fields, creating an additional pathway for activation flow with randomly initialised kernel parameters. This mutation establishes connectivity between fields without altering the number of fields and is critical for signal propagation.

A second structural operator *introduces a new hidden field* along an existing connection. Rather than adding a field in isolation, this mutation replaces an existing interaction with: an input connection from the original source field to the new field, followed by an output connection from the new field to the original target. The parameters of the outgoing interaction inherit those of the original connection, while the incoming interaction is initialised with pre-defined parameters. This was designed as an attempt to limit

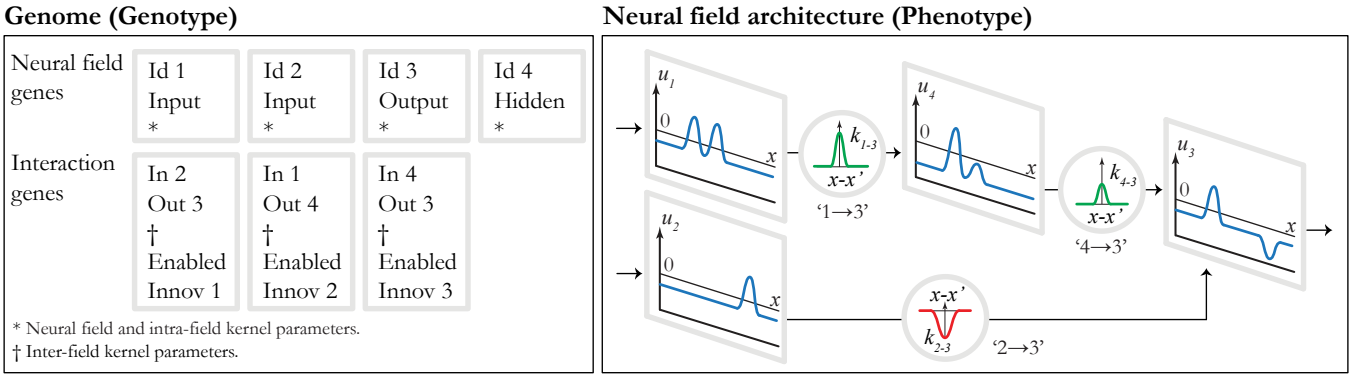


Figure 1: Genotype-to-phenotype mapping in NEAT-DNFs. The left panel displays the genomic representation (genotype), which contains four field genes (two input, one hidden, and one output) and three interaction genes that define connections between fields. The right panel illustrates the corresponding decoded DNF-based architecture (phenotype).

the disruptive effects on existing functionality, allowing the new field to be gradually integrated and optimised through subsequent parametric mutations.

In addition to these additive operators, the system supports a *toggle mutation* that enables or disables existing interaction genes. Disabling an interaction removes its influence from the phenotype without deleting the underlying genetic material. This reversible mutation allows evolution to explore alternative information flow configurations, silence pathways, and later reactivate them if selection pressures change.

3.3 Historical Markings and Crossover

As genomes diverge through structural mutations, recombination becomes non-trivial due to topological mismatches. NEAT-DNFs adopts NEAT's historical marking mechanism to address this problem. Each structural innovation is assigned a globally unique innovation number at the time of its creation, which is inherited by offspring.

During crossover, genomes are aligned according to innovation numbers. Matching genes are inherited randomly from either parent, while disjoint and excess genes are inherited from the fitter parent. Disabled genes remain disabled in the offspring, preserving the integrity of previously silenced pathways while allowing reactivation through future mutations.

3.4 Speciation and Compatibility Distance

Structural mutations can introduce new fields or interaction pathways whose benefit only emerges after several generations of parametric refinement. To protect such innovations from premature loss, NEAT-DNFs incorporates a speciation mechanism that extends NEAT's original compatibility-based approach to kernel-defined neural interactions.

Individuals are partitioned into species based on a compatibility distance δ, that captures both architectural and functional differences. Two genomes are considered similar if they share a comparable topological structure and exhibit similar spatial interaction profiles. The compatibility distance is defined as:

$$\delta = \frac{c_e E}{N} + \frac{c_d D}{N} + c_k \overline{K} \tag{5}$$

where E and D denote the number of excess and disjoint genes, respectively, and N is the size of the larger genome, used to normalise for architectural complexity (with $N = 1$ for genomes with less than 5 genes). The coefficients c_e, c_d and c_k control the relative influence of structural and functional differences. $\overline{K}$ is a kernel difference term that aggregates parametric differences across matching interaction genes.

In standard NEAT, functional similarity between connections is assessed through scalar weight differences. This assumption does not hold for DNF-based models, where interactions are defined by spatial kernels whose effects depend on multiple parameters. In NEAT-DNFs, functional similarity is therefore evaluated at the level of interaction kernels rather than individual weights. Thus, $\overline{K}$ is defined by:

$$\overline{K} = c_a \sum_{i \in M} |a_i^1 - a_i^2| + c_w \sum_{i \in M} |w_i^1 - w_i^2| \tag{6}$$

where M is the set of *matching genes*, a_i^1, a_i^2, w_i^1, and w_i^2 denote the amplitude and width parameters of the corresponding kernels in the two genomes. The coefficients c_a and c_w weight the contribution of amplitude and spatial extent differences. For Gaussian kernels, these parameters correspond directly to the kernel amplitude and spread; for Mexican-hat kernels, they refer to the excitatory component's amplitude and width.

As in NEAT, fitness sharing is applied within species to prevent dominance by a single lineage and to maintain diversity. Reproductive opportunities are allocated in proportion to (shared) species fitness.

3.5 Stagnation Management

To improve search efficiency and mitigate evolutionary stagnation, NEAT-DNFs incorporates several regulatory mechanisms that modulate reproductive opportunities based on fitness performance. At the population level, if global fitness plateaus for ten generations, reproduction is restricted to the two highest-performing species

to redirect evolutionary pressure toward the most promising architectural subtypes. Within individual species, any species failing to demonstrate fitness improvements over seven generations is classified as stagnant and excluded from further reproduction. Furthermore, the framework implements a conditional elitism mechanism where species containing more than five members grant their highest-performing "champion" guaranteed passage to the subsequent generation without modification.

3.6 Fitness Evaluation

Each genome is evaluated by simulating the corresponding DNF architecture in response to task-specific input patterns. External stimuli are applied to input fields as spatially localised Gaussian inputs:

$$s_{\text{input}}(x,t) = \sum_i A \exp\left(-\frac{(x-p_i)^2}{2\sigma^2}\right) \tag{7}$$

where p_i denotes the stimulus position, A its amplitude, and σ its width. In all experiments reported in this paper, Gaussian stimuli use fixed values of $A = 20$ and $\sigma = 5$. These values were chosen to ensure reliable supra-threshold activation across the range of resting levels explored during evolution, while keeping the input spatially localised enough to evaluate position-specific field responses.

The resulting spatiotemporal activity patterns across neural fields are evaluated using a set of task-dependent *behavioural fitness measures* f_b, which are combined into a single scalar fitness score:

$$f = \sum_{N_b} w_b f_b \tag{8}$$

where w_b specifies the relative importance of behaviour b.

Fitness evaluation is based on observable spatiotemporal properties of field dynamics. Across all tasks considered in this work, successful behaviour can be characterised in terms of a small number of qualitative criteria: whether an activation peak exists, where it is located, how strong and spatially extended it is, or whether the field remains near its resting state. Accordingly, we define a small set of reusable behaviour fitness functions that can be combined in different ways to form task-specific fitness functions.

To evaluate whether a field forms activation peaks with desired properties, we compute a fitness score based on (i) the number of supra-threshold peaks, and for the peak closest to the target, (ii) its position, (iii) amplitude, and (iv) width:

$$\begin{aligned} f_{\text{bump}} = &\frac{0.45}{1+|n_{\text{actual}} - n_{\text{target}}|} + \frac{0.30}{1+|p_{\text{closest}} - p_{\text{target}}|} \\ &+ \frac{0.15}{1+|a_{\text{closest}} - a_{\text{target}}|} + \frac{0.10}{1+|w_{\text{closest}} - w_{\text{target}}|} \end{aligned} \tag{9}$$

For tasks in which a field is expected to return to baseline, we assess proximity to the resting level using:

$$f_{resting} = \frac{1}{1+|u_{\max} - h|} \tag{10}$$

Where $u_{\max}$ represents the maximum activation value across the field.

Finally, to explicitly penalise undesired supra-threshold activation, we define:

$$f_{no\ peaks} = \begin{cases} 1, & u_{\max} < 0, \\ \exp(-10\, u_{\max}), & u_{\max} \geq 0. \end{cases} \tag{11}$$

This function assigns maximal fitness when no activation peaks are present and applies an exponential penalty when positive activations occur.

Task-specific fitness functions are constructed by selecting and weighting appropriate subsets of these behavioural fitness measures, depending on the requirements of the task.

3.7 Implementation and Parameter Settings

All experiments use the same evolutionary and simulation settings; parameters were not tuned for any specific task. We hold these settings constant across tasks to isolate the effect of task constraints. Values were selected experimentally, and performance is robust to moderate variations. To support reproducibility and adoption, the implementation of NEAT-DNFs and the complete hyperparameter specification are publicly available at `Jgocunha/neat-dnfs`.

4 Experimental Evaluation

Our evaluation follows the modelling philosophy of DFT and therefore does not target standard pattern-recognition benchmarks. Instead, we use a hierarchy of tasks where behaviour must emerge from continuous-time neural dynamics.

4.1 Core Dynamics

The first set of experiments primarily evaluates whether NEAT-DNFs can evolve intrinsic neural field parameters to reproduce fundamental dynamic mechanisms central to DFT, which constitute the foundational building blocks of more complex cognitive architectures.

4.1.1 Detection Instability. This task evaluates whether transient external input induces a localised activation peak that propagates from the input field to the output field, followed by a return to baseline after stimulus removal. A Gaussian stimulus ($p = 50$) is applied to the input field for $\Delta t_{stim} = 200$ms, followed by a "stimulus-free" period of $\Delta t_{no\ stim} = 400$ms. Successful solutions must satisfy four equally weighted criteria: (i) formation of an activation peak in the input field during stimulation; (ii) formation of a corresponding peak in the output field; (iii) decay of input field activity to resting level after stimulus removal; and (iv) decay of output field activity to resting level after stimulus removal.

4.1.2 Memory Instability. The memory task extends the detection scenario by requiring the output field to maintain a stable activation peak in the absence of input, corresponding to working memory. The stimulus protocol is identical to the detection task, with a stimulus duration of $\Delta t_{stim} = 200$ ms, followed by a post-stimulus period of $\Delta t_{no\ stim} = 600$ ms. The input field is required to return to resting level after stimulus removal, while the output field must maintain a stable activation peak. Fitness evaluation differs from the detection task only in the requirement for self-sustained output-field activation.

4.1.3 Selection Instability. The selection task evaluates the emergence of competitive, winner-take-all dynamics. Two spatially distinct stimuli ($p_1 = 25$, $p_2 = 75$) are presented simultaneously to the input field for $\Delta t_{stim} = 200$ ms. During stimulation, the input field must represent both activation peaks, while the output field must stabilise a single peak at one of the stimulus locations. After stimulus removal, both fields are required to return to resting level after a $\Delta t_{no\ stim} = 600$ ms period.

4.2 Compositional Tasks Requiring Architectural Innovation

While the core dynamics tasks primarily test parameter optimisation, the central contribution of NEAT-DNFs lies in its ability to evolve architecture. The following tasks cannot be solved by a two-field architecture alone and require the introduction of at least one intermediate neural field that is capable of memorising previously seen stimuli, in order to bias future decision-making.

4.2.1 Delayed Match-to-Sample. The DMTS task is defined on a one-dimensional circular feature space (colour hue), discretised into $N = 360$ points. Each trial consists of three phases: (i) *sample phase* ($\Delta t_{sample} = 200$ ms): a single colour/hue stimulus is presented; (ii) *delay phase* ($\Delta t_{delay} = 600$ ms): no external input is applied; and (iii) test phase ($\Delta t_{test} = 200$ ms): two colours (match and non-match) are presented.

Fitness is the mean of six equally weighted criteria assessing input and output field dynamics across the three task phases. During the sample phase, both input and output fields must form an activation peak at the sample location. During the delay phase, both fields must return to baseline while information about the sample is maintained internally. During the test phase, the output field must selectively respond only to the matching stimulus location.

4.2.2 Inhibition of Return. For the Inhibition of Return (IOR) task, the spatial domain is discretised with $N = 100$. A single cue stimulus is presented either on the left or right half of the domain for $\Delta t_{cue} = 200$ ms, followed by a stimulus-free interval of $\Delta t_{delay} = 400$ ms. In the test phase, $\Delta t_{test} = 200$ ms, both left and right cues are presented, and the new cue should be detected first.

Fitness is evaluated using five equally weighted criteria. During the cue phase, both input and output fields must form an activation peak at the cued location. After cue removal, activity must decay in both, but an internal inhibitory trace must persist. Upon presentation of both cues, the new one should be detected first.

5 Results

All reported results are based on 100 independent evolutionary runs per task, using a fitness threshold of 0.95 across all behavioural fitness components to determine successful convergence. All tasks achieved a 100% success rate. Table 1 summarises convergence behaviour and emergent architectural properties of the evolved solutions across all tasks.

5.1 Quantitative Evolutionary Performance

Across the core dynamics tasks (detection, memory, and selection instabilities), NEAT-DNFs converges reliably and fast. Detection and memory reach the fitness threshold in a median of 1 and 2 generations, respectively, indicating that these regimes are typically discovered with minimal search. Selection converges more slowly (median 9 generations), consistent with its stronger dynamical requirements. The selection task requires the input field to exhibit multi-peak dynamics while the output field implements winner-take-all competition to select a single location. Because both stimuli are presented simultaneously at equal amplitude, successful solutions must realise a narrowly tuned competitive regime. Overall, rapid convergence on these foundational mechanisms provides a stable substrate for subsequent structural complexification, enabling later tasks to build on robust core dynamics.

Table 1: Convergence and emergent architectural complexity across tasks. Convergence is reported as the median number of generations required to reach the fitness threshold. Architectural complexity is reported as mean ± standard deviation of the number of hidden fields and enabled inter-field connections.

Task	Median Gen.	Hidden Fields (mean ± sd)	Enabled Connections (mean ± sd)
Detection	1	0.00 ± 0.00	1.00 ± 0.00
Memory	2	0.00 ± 0.10	1.00 ± 0.00
Selection	9	0.23 ± 0.51	1.41 ± 1.09
DMTS	44	1.29 ± 0.50	3.57 ± 1.27
IOR	57.5	1.19 ± 0.39	3.43 ± 0.96

For the DMTS and IOR tasks, convergence occurs over substantially longer evolutionary timescales while remaining reliable. Both tasks require structural innovation in the form of an additional memory field, along with finely tuned couplings from this field to the output. In DMTS, the memory-to-output coupling needs to provide an excitatory bias that supports the delayed matching (median 44 generations). In IOR, the corresponding coupling must be inhibitory to produce the delayed suppression of the previously cued location (57.5 generations).

5.2 Emergence of Topological Complexity

A central claim of this work is that NEAT-DNFs introduces structural complexity only when functionally necessary. The architectural statistics in Table 1 strongly support this claim.

For all core dynamics tasks, successful solutions almost exclusively retained minimal two-field architectures, consisting only of the initial input and output fields. Occasional runs introduced additional structure, but such cases were rare and did not confer performance advantages.

In contrast, tasks requiring an internal representation of previously presented stimuli consistently induce architectural growth. In DMTS, successful solutions exhibit 1.29 ± 0.50 hidden fields and 3.57 ± 1.27 enabled connections on average. These additional fields enable the system to maintain stimulus information during the delay period and to bias subsequent selection during the test phase, functionality that cannot be achieved by a two-field architecture alone. A similar pattern is observed for IOR, where evolved architectures contain 1.19 ± 0.39 hidden fields and 3.43 ± 0.96 enabled connections on average. Here, intermediate fields support delayed

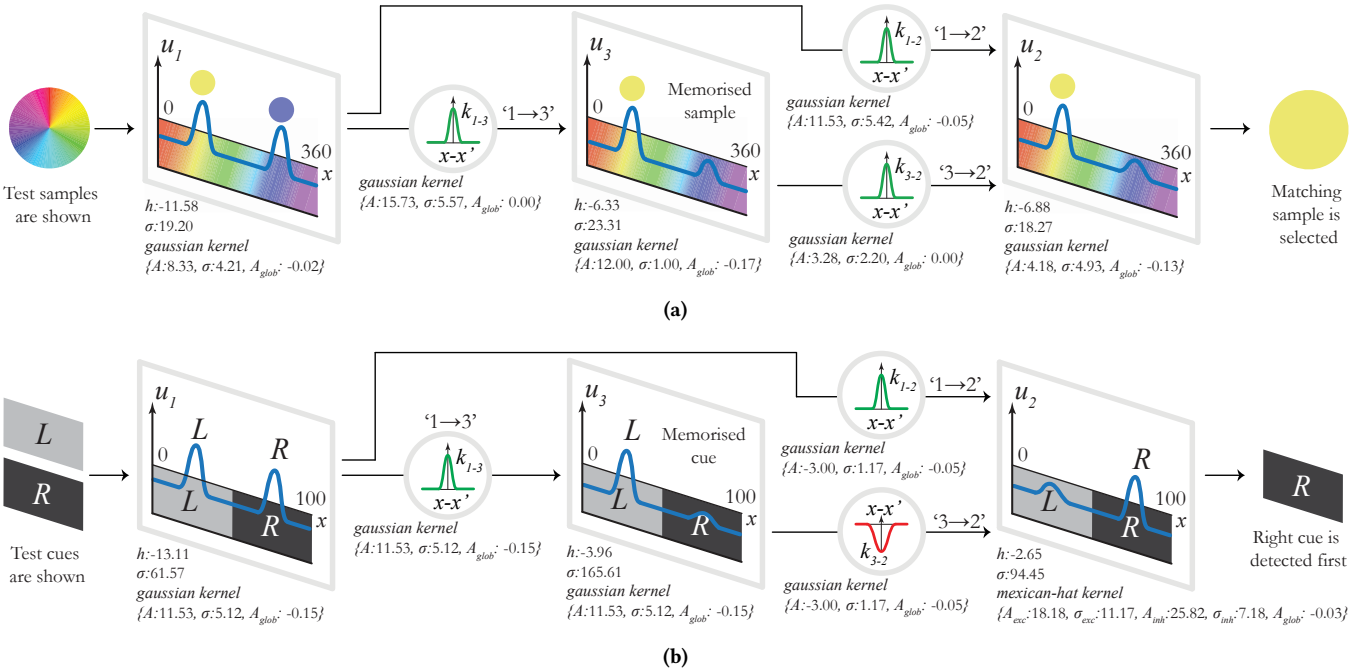


Figure 2: Representative highest-fitness minimal-topology phenotypes for the (a) DMTS and (b) IOR tasks. Both solutions consist of a three-field architecture in which an intermediate self-sustained field maintains a memory of the previously seen sample or cue. Through an excitatory (DMTS) or inhibitory (IOR) coupling to the output field, this memory either positively or negatively biases future decisions.

inhibitory influences on the output field, reducing responsiveness to previously attended locations and biasing selection toward novel alternatives.

For both the DMTS and IOR tasks, the evolved architectures are near-optimal, as the minimal viable solution requires only a single hidden field and three inter-field connections.

5.3 Evolved Field Dynamics

To complement the quantitative results above, we analysed the internal field dynamics of representative high-fitness solutions for the two most complex tasks, DMTS and IOR. Figure 2 describes the evolved minimal topologies for both tasks, while Figure 3 shows the corresponding field dynamics across task phases. Visualisations of the phenotypes and spatiotemporal field dynamics for all tasks are provided in the accompanying video ▶ youtu.be/tgNbhQQRmbM.

5.3.1 Delayed Match-to-Sample. Figure 2a illustrates a representative evolved solution for the DMTS task. The majority of successful solutions converge to a minimal three-field, three-connection architecture comprising an input field, an output field, and an added hidden field that exhibits working-memory dynamics.

The corresponding field dynamics are shown in Figure 3a. During the sample phase, the input field forms an activation peak at the sample location (yellow hue $p_{stim} = 60$) and drives activation to both the output and the working-memory fields through evolved excitatory couplings, producing peaks in both fields. After stimulus removal (delay phase), activity in the input field decays to resting level, while the working-memory field maintains a self-sustained activation peak, encoding an internal representation of the yellow sample. This activation propagates to the output field via the evolved coupling, pre-shaping it into a subthreshold bump at the remembered location. During the test phase, two competing stimuli (yellow and purple hue $p_{stim} = 60, 270$) are presented and detected by the input field. Selection then emerges in the output field, which stabilises a single activation peak at the location consistent with the previously seen sample.

5.3.2 Inhibition of Return. Figure 2b shows a representative evolved solution for the IOR task. As in DMTS, successful solutions typically exhibit a three-field architecture, with the third field acting as a working-memory. The key difference lies in the evolved inhibitory projection from the memory field to the output field.

Field dynamics for the representative solution are shown in Figure 3b. During the initial cue phase, a stimulus at one spatial location (left side $p_{stim} = 20$) is detected by the input field, and activation propagates to both the output field and the memory field, producing peaks at the cued location. After cue removal (delay), the memory field maintains a self-sustained peak, preserving a trace of the previously attended cue.

Depending on the duration of the delay, the output field undergoes either a period of *facilitation*, in which an existing slowly decaying pre-shape from the input field lingers, so that re-presentation of the same stimulus would lead to faster activation. At later times, inhibitory input from the memory field suppresses activity at the previously cued location, producing a "negative pre-shape". During the test phase, when competing stimuli are presented (left and right

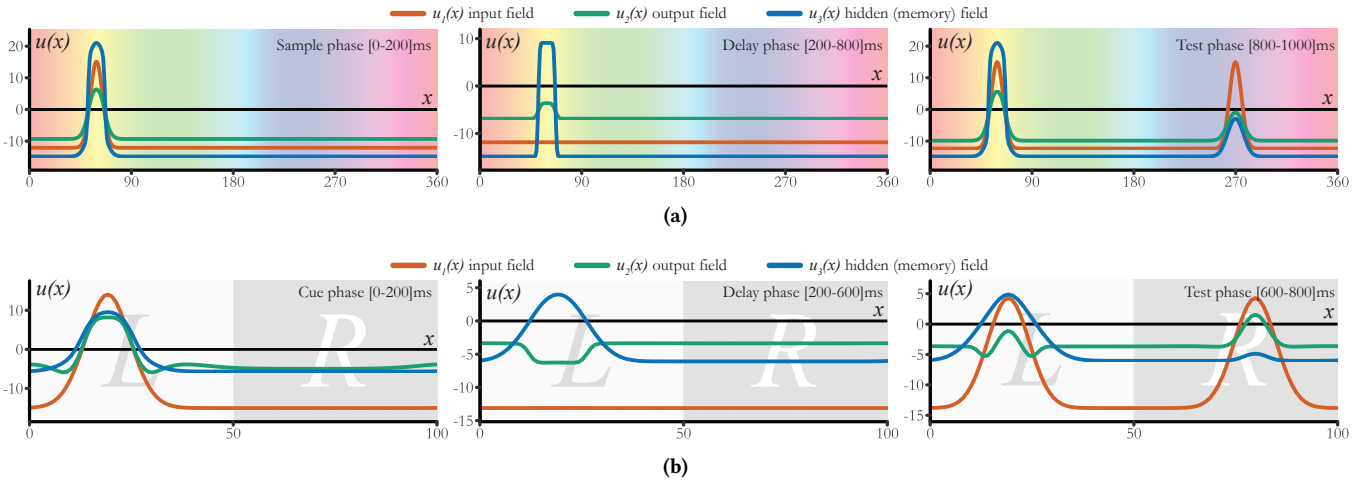


Figure 3: Representative highest-fitness minimal-topology field dynamics for the (a) DMTS and (b) IOR tasks. (a) DMTS: activation is stored as a self-sustained trace during the delay and biases the output to select the matching test stimulus. (b) IOR: a sustained activation produces localised inhibition at the cued location, shifting output selection toward the novel location.

cues, $p_{stim} = 20, 80$), this localised suppression delays re-selection of the cued location and biases the output field toward the novel location, reproducing inhibition-of-return behaviour.

6 Discussion

The results in this paper show that NEAT-DNFs can evolve both DNF parameters and topology to solve a range of behavioural tasks. By extending NEAT-style complexification to continuous-time, kernel-based interactions, the framework can autonomously generate foundational DFT mechanisms —detection, working memory, and selection — without manual design. Core behaviours emerge in compact two-field solutions, while tasks that genuinely require intermediate processing (DMTS and IOR) reliably discover additional hidden fields. Importantly, the resulting architectures remain interpretable: when extra structure appears, it tends to take on clear functional roles (e.g., working-memory), rather than adding arbitrary complexity.

There are also clear limitations. First, the approach still depends on hand-designed evaluation criteria: fitness functions encode what counts as the "right" spatiotemporal behaviour, which requires domain knowledge and can shape what evolution discovers. Second, while single-objective optimisation is sufficient for the benchmarks here, scaling to richer tasks will likely benefit from multi-objective formulations. Multi-objective NEAT variants have been shown to be effective in broader settings, and incorporating a similar approach into NEAT-DNFs is a natural next step [1, 22].

Finally, the scalability of NEAT-DNFs to higher-dimensional fields and larger, multi-layer DFT architectures remains open. The tasks in this study required limited structural innovation; the more demanding test is whether evolution can reliably discover multi-field architecture for embodied agents. A direct next step is to evaluate NEAT-DNFs in closed-loop sensorimotor control problems where DFT has been successful, and to study whether the same task-driven emergence of structure holds under real-time constraints and dynamic environments.

7 Conclusion

We introduced NEAT-DNFs, a neuroevolutionary framework that extends NeuroEvolution of Augmenting Topologies (NEAT) to the automated synthesis of Dynamic Neural Field (DNF) architectures. By encoding neural fields and kernel-defined inter-field couplings as evolvable genes, NEAT-DNFs jointly optimises intrinsic field dynamics and architectural topology, enabling the autonomous discovery of interpretable Dynamic Field Theory (DFT)-style computational mechanisms without manual tuning.

Across a hierarchy of tasks, NEAT-DNFs reliably evolves compact solutions and increases structural complexity only when required by task demands. In particular, for tasks that require internal state (DMTS and IOR), evolution consistently discovers intermediate memory fields and functionally appropriate couplings that bias subsequent decisions through excitatory pre-shaping or delayed inhibition. These results establish NEAT-DNFs as a step toward scalable, interpretable evolutionary synthesis of biologically grounded dynamical controllers.

Acknowledgments

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Unit Project Scope UID/00319/2025 - Centro ALGORITMI (ALGORITMI/UM) and PhD Fellowship under grant 2022.13948.BD.

References

[1] Omer Abramovich and Amiram Moshaiov. 2016. Multi-objective topology and weight evolution of neuro-controllers. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, Vancouver, BC, Canada, 670–677. doi:10.1109/CEC.2016.7743857

[2] Shun-ichi Amari. 1977. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics* 27, 2 (1977), 77–87. doi:10.1007/BF00337259

[3] Derrik E. Asher, Nicolas Oros, and Jeffrey L. Krichmar. 2015. The Importance of Lateral Connections in the Parietal Cortex for Generating Motor Plans. *PLOS ONE* 10, 8 (Aug. 2015), e0134669. doi:10.1371/journal.pone.0134669

[4] Alejandro Baldominos, Yago Saez, and Pedro Isasi. 2020. On the automated, evolutionary design of neural networks: past, present, and future. *Neural Computing*

and Applications 32, 2 (Jan. 2020), 519–545. doi:10.1007/s00521-019-04160-6

[5] Dimitrios Chatziparaschis, Shan Zhong, Vasileios Christopoulos, and Konstantinos Karydis. 2024. Adaptive Environment-Aware Robotic Arm Reaching Based on a Bio-Inspired Neurodynamical Computational Framework. In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. IEEE, Pasadena, CA, USA, 510–515. doi:10.1109/RO-MAN60168.2024.10731226

[6] Shifei Ding, Hui Li, Chunyang Su, Junzhao Yu, and Fengxiang Jin. 2013. Evolutionary artificial neural networks: a review. *Artificial Intelligence Review* 39, 3 (March 2013), 251–260. doi:10.1007/s10462-011-9270-6

[7] Wolfram Erlhagen and Estela Bicho. 2006. The dynamic neural field approach to cognitive robotics. *Journal of Neural Engineering* 3, 3 (Sept. 2006), R36–R54. doi:10.1088/1741-2560/3/3/R02

[8] Wolfram Erlhagen and Estela Bicho. 2014. A Dynamic Neural Field Approach to Natural and Efficient Human-Robot Collaboration. In *Neural Fields*, Stephen Coombes, Peter Beim Graben, Roland Potthast, and James Wright (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 341–365. doi:10.1007/978-3-642-54593-1_13

[9] Flora Ferreira, Weronika Wojtak, Emanuel Sousa, Luis Louro, Estela Bicho, and Wolfram Erlhagen. 2021. Rapid Learning of Complex Sequences With Time Constraints: A Dynamic Neural Field Model. *IEEE Transactions on Cognitive and Developmental Systems* 13, 4 (Dec. 2021), 853–864. doi:10.1109/TCDS.2020.2991789

[10] Jérémy Fix. 2013. Template based black-box optimization of dynamic neural fields. *Neural Networks* 46 (Oct. 2013), 40–49. doi:10.1016/j.neunet.2013.04.008

[11] Dario Floreano, Peter Dürr, and Claudio Mattiussi. 2008. Neuroevolution: from architectures to learning. *Evolutionary Intelligence* 1, 1 (March 2008), 47–62. doi:10.1007/s12065-007-0002-4

[12] Jorge Ibáñez-Gijón and David M. Jacobs. 2012. Decision, Sensation, and Habituation: A Multi-Layer Dynamic Field Model for Inhibition of Return. *PLoS ONE* 7, 3 (March 2012), e33169. doi:10.1371/journal.pone.0033169

[13] Christian Igel, Wolfram Erlhagen, and Dirk Jancke. 2001. Optimization of dynamic neural fields. *Neurocomputing* 36, 1-4 (Feb. 2001), 225–233. doi:10.1016/S0925-2312(00)00328-3

[14] Benjamin Inden, Yaochu Jin, Robert Haschke, and Helge Ritter. 2010. NEATfields: evolution of neural fields. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, Portland Oregon USA, 645–646. doi:10.1145/1830483.1830601

[15] Jeffrey S. Johnson, John P. Spencer, Steven J. Luck, and Gregor Schöner. 2009. A Dynamic Neural Field Model of Visual Working Memory and Change Detection. *Psychological Science* 20, 5 (May 2009), 568–577. doi:10.1111/j.1467-9280.2009.02329.x

[16] Cesar Gomes Miguel, Carolina Feher Da Silva, and Marcio Lobo Netto. 2008. Structural and Parametric Evolution of Continuous-Time Recurrent Neural Networks. In *2008 10th Brazilian Symposium on Neural Networks*. IEEE, Salvador, Brazil, 177–182. doi:10.1109/SBRN.2008.12

[17] Jean-Baptiste Mouret, Stéphane Doncieux, and Benoît Girard. 2010. Importing the computational neuroscience toolbox into neuro-evolution-application to basal ganglia. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, Portland Oregon USA, 587–594. doi:10.1145/1830483.1830592

[18] Evgenia Papavasileiou, Jan Cornelis, and Bart Jansen. 2021. A Systematic Literature Review of the Successors of "NeuroEvolution of Augmenting Topologies". *Evolutionary Computation* 29, 1 (March 2021), 1–73. doi:10.1162/evco_a_00282

[19] Jean-Charles Quinton. 2010. Exploring and optimizing dynamic neural fields parameters using Genetic Algorithms. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Barcelona, Spain, 1–7. doi:10.1109/IJCNN.2010.5596293

[20] Yulia Sandamirskaya. 2014. Dynamic neural fields as a step toward cognitive neuromorphic architectures. *Frontiers in Neuroscience* 7 (2014). doi:10.3389/fnins.2013.00276

[21] Stefan Schneider, Christian Igel, Christian Klaes, Hubert R. Dinse, and Jan C. Wiemer. 2004. Evolutionary Adaptation of Nonlinear Dynamical Systems in Computational Neuroscience. *Genetic Programming and Evolvable Machines* 5, 2 (June 2004), 215–227. doi:10.1023/B:GENP.0000023689.70987.6a

[22] Jacob Schrum and Risto Miikkulainen. 2016. Solving Multiple Isolated, Interleaved, and Blended Tasks through Modular Neuroevolution. *Evolutionary Computation* 24, 3 (Sept. 2016), 459–490. doi:10.1162/EVCO_a_00181

[23] Gregor Schöner, John Spencer, and DFT Research Group. 2015. *Dynamic Thinking: A Primer on Dynamic Field Theory*. Oxford University Press, Oxford. doi:10.1093/acprof:oso/9780199300563.001.0001

[24] Kenneth O. Stanley, David B. D'Ambrosio, and Jason Gauci. 2009. A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks. *Artificial Life* 15, 2 (April 2009), 185–212. doi:10.1162/artl.2009.15.2.15202

[25] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation* 10, 2 (June 2002), 99–127. doi:10.1162/106365602320169811

[26] Weronika Wojtak, Flora Ferreira, Wolfram Erlhagen, and Estela Bicho. 2015. Learning joint representations for order and timing of perceptual-motor sequences: A dynamic neural field approach. In *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Killarney, Ireland, 1–7. doi:10.1109/IJCNN.2015.7280717

[27] Weronika Wojtak, Flora Ferreira, Luís Louro, Estela Bicho, and Wolfram Erlhagen. 2023. Adaptive timing in a dynamic field architecture for natural human–robot interactions. *Cognitive Systems Research* 82 (Dec. 2023), 101148. doi:10.1016/j.cogsys.2023.101148

[28] Xin Yao. 1999. Evolving artificial neural networks. *Proc. IEEE* 87, 9 (Sept. 1999), 1423–1447. doi:10.1109/5.784219

[29] Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis, and Gregor Schoner. 2011. Dynamic Neural Fields as Building Blocks of a Cortex-Inspired Architecture for Robotic Scene Representation. *IEEE Transactions on Autonomous Mental Development* 3, 1 (March 2011), 74–91. doi:10.1109/TAMD.2011.2109714

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009