Embodied Neurocomputation: A Framework for Interfacing Biological Neural Cultures with Scaled Task-Driven Validation

Johnson Zhou[◐●] **Daniel Tanneberg**[◑●]
Forough Habibollahi[◐] **Alon Loeffler**[◐] **Kiaran Lawson**[◐] **Valentina Baccetti**[◐]
Kwaku Dad Abu-Bonsrah[◐] **Candice Desouza**[◐] **Finn Doensen**[◐] **Bradley Watmuff**[◐]
Daria Kornienko[◐] **Azin Azadi**[◐] **Justin L. Bourke**[◐]
Bernhard Sendhoff[◑] **Brett J. Kagan**[◐]

[◐]Cortical Labs, Australia [◑]Honda Research Institute Europe, Germany [●]equal contribution

Abstract

Biological neural networks (BNNs) have been established as a powerful and adaptive substrate that offer the potential for incredibly energy and data efficient information processing with distinct learning mechanisms. Yet a core challenge to utilizing BNN for neurocomputation is determining the optimal encoding and decoding mechanisms between the traditional silicon computing interface and the living biology. Here, we propose an Embodied Neurocomputation framework as a systems-level approach to this multi-variable optimization encoding/decoding problem. We operationalize this approach through the first large-scale parameter optimization of encoding configurations for a BNN agent performing closed-loop navigation along an odor-style gradient in a simulated grid-world. Despite the relative simplicity of the task, the biological interactions gave rise to a massive multi-combinatorial search space for optimal parameters. By considering how the components of the system are interconnected and parameterized, we evaluated approximately 1,300 parameter combinations, over 4,000 hours of real-time agent-environment interactions, to identify 12 configurations that consistently demonstrated learning across multiple episodes. These configurations achieved significantly higher task performances than optimized silicon-based DQN agents under the same interaction budget. These findings represent an initial step toward robust and scalable goal-oriented learning using BNNs. Our framework establishes a foundation for applying task-driven neurocomputing and supports the development of field-wide benchmarks. In the long term, this work supports the development of hybrid bio-silicon architectures capable of efficient, adaptive and real-time computation, including the potential for robotic control applications.

1 Introduction

The rapid growth of artificial intelligence and machine learning has pushed computational demands to unprecedented levels, exposing the limitations of traditional silicon-based architectures. As conventional hardware nears the physical boundaries of Moore's Law and struggles with the energy inefficiencies of the von Neumann bottleneck, the ecological and economic costs of scaling current computing paradigms have become unsustainable [1, 2]. In contrast, biological neural networks (BNNs) perform highly complex, non-linear processing, including continuous learning, robust pattern recognition, and adaptive decision-making, while using only a fraction of the power required by digital systems [3–5]. Bridging this efficiency gap has motivated the search for fundamentally new computational substrates. By harnessing the intrinsic processing power of living BNN cultures,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

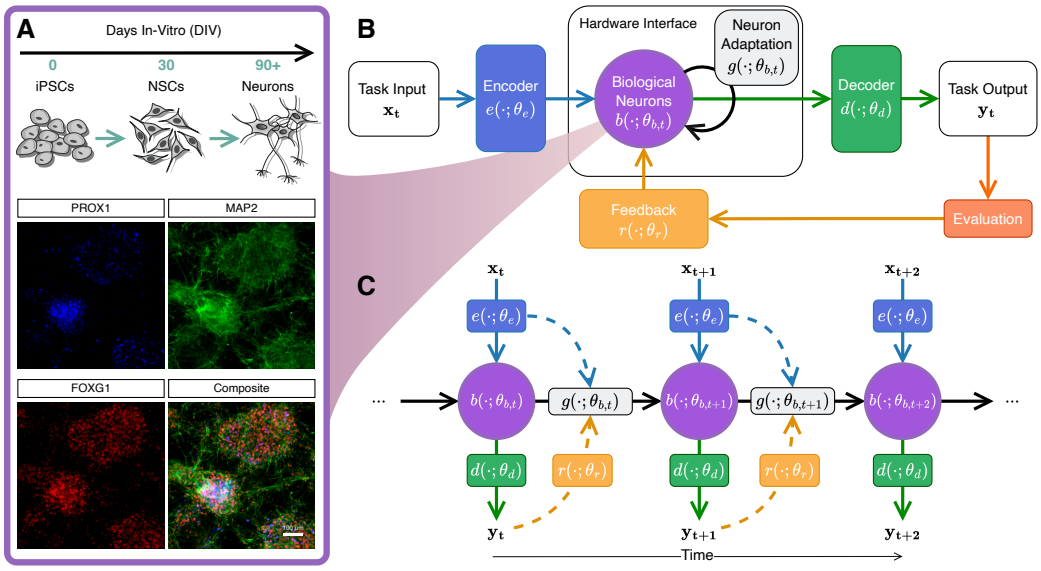


Figure 1: Schematic of the proposed Embodied Neurocomputation Framework defined in Section 2. **A**: Preparation and characterization of the biological neural substrate. (*Top*) Timeline of neuronal differentiation from induced Pluripotent Stem Cells (iPSCs) via Neural Stem Cells (NSCs) to mature neurons at 90 days *in vitro* (DIV) (*Bottom*) Immunocytochemical characterization of neuronal identity. Micrographs show distinct labeling for PROX1, MAP2, and FOXG1, confirming the cortical and hippocampal makeup of the computational network. **B**: Functional mapping of framework components involved in transforming inputs to outputs ($\mathbf{x}_t \mapsto \mathbf{y}_t$) at timestep t through a silicon-biological hardware interface. **C**: Evolving temporal dynamics driven by biological neuron adaptation $g(\cdot;\cdot)$ as it is influenced by encoding $e(\cdot;\cdot)$ and feedback $r(\cdot;\cdot)$ mechanisms.

neurocomputing represents a critical paradigm shift: moving away from energy-hungry silicon toward engineered embodied biological systems capable of inherently efficient, intelligent computation.

Despite neurocomputing's promise, a fundamental bottleneck in functional *in vitro* systems lies at the interface between the digital environment and biological substrate. Micro-Electrode Arrays (MEAs) are the primary platform for this bidirectional communication, enabling both recording from and stimulation of cultured neural networks [6, 7]. However, while *decoding* network activity has advanced substantially [8, 9], the inverse challenge, *encoding* meaningful digital information into a biological format, remains profoundly difficult [10]. Unlike deterministic silicon pathways, biological networks are adaptive, noisy, and non-stationary [11]. As a result, translating structural data into electrical stimuli often yields unpredictable network states, as static or poorly tuned encoding schemes struggle to reliably drive the non-linear dynamics required for computation [12].

Current efforts to drive information processing in MEA-embodied networks rely heavily on heuristic or ad-hoc stimulation protocols. Although studies have demonstrated closed-loop control, pattern recognition, and task-specific learning [6, 13–16], they typically use highly constrained encoding parameters. Consequently, systematic literature exploring the multidimensional parameter space needed for effective information translation remains limited. Biological encoding spans a vast stimulation space, including frequency, amplitude, pulse width, waveform morphology, and spatiotemporal distribution [17]. Without broad evaluation across these parameters, it remains unclear which encoding regimes optimally couple with network biophysics to produce robust plasticity and learning [13].

While methods for culturing and assembling BNNs continue to evolve (e.g., [18, 19]), the engineering principles needed to harness their computational capacity remain underexplored. This capacity emerges from the structural and temporal interplay between external information forces and internal physiological states [14, 20], requiring systematic interface optimization. To address this gap, this study moves beyond isolated proof-of-concept stimulation paradigms.

In this study, we first introduce a formal *Embodied Neurocomputation Framework* (Section 2), which conceptualizes the interface between digital and biological substrates as a multi-variable optimization problem composed of four interdependent modules: (1) encoding of task information into electrical

stimuli, (2) biological transformation via the neural network's intrinsic dynamics, (3) decoding of neural responses into task-relevant outputs, and (4) feedback to drive adaptation.

Secondly, to validate this framework, we conduct a large-scale empirical evaluation (Section 3) within a simulated goal-driven navigation environment. By integrating 26 individual BNN cultures via the Cortical Labs CL1 platform [21] with an automated hyperparameter optimization pipeline, we systematically screened approximately 1,300 encoding configurations over 4,000 hours of real-time agent-environment interactions.[1] We find that: (1) specific parameter subsets enable robust learning across multiple episodes, and (2) optimized biological agents can significantly outperform silicon-based Deep Q-Networks, which serves as a standardized benchmark for discrete action-space interactions, when evaluated over an equivalent number of training steps. To our knowledge, this is the first study to optimize neurocomputing parameters at scale, establishing a systematic and data-driven foundation for the future of information processing using biological neural substrates.

2 Embodied Neurocomputation Framework

We define *Embodied Neurocomputation* as a computational framework that interfaces biological neural networks (BNNs) with conventional computers. While physical implementations utilize specific hardware such as multi-electrode arrays (MEAs) that mediates BNN interaction in the context of electrical activity and stimulus, the framework remains hardware-agnostic by encapsulating these physical interactions within defined mapping functions.

We approach this framework through systems-level thinking since each of its components are interconnected and highly parameterized. Adjusting any single part of the system changes how the entire system responds. Consequently, we treat the configuration of a neurocomputational task as a multi-variable optimization problem, where the objective is to find the optimal set of transformations across all parameters to achieve a specific computational goal. This approach distinguishes our work from existing definitions of Neural Computers [22, 23] which focus on internalizing computational primitives to a latent state of artificial neural networks. In this section, we introduce the key elements of the neurocomputation framework in an abstract and generally applicable manner. In Section 3, we concretely apply this framework and use it to optimize task oriented parameters at scale.

We model neurocomputation $f(\,\cdot\,;\theta_t)$ at time t, with parameter set θ_t, as sequential feed-forward mappings that transform elements of an input sequence ($\mathbf{x}_t$) into corresponding elements of an output sequence ($\mathbf{y}_t$) at timestep (t) (Figure 1):

$$\mathbf{y}_t = f(\mathbf{x}_t;\theta_t) = \Big[\; d(\,\cdot\,;\theta_d) \circ b(\,\cdot\,;\theta_{b,t}) \circ e(\,\cdot\,;\theta_e) \;\Big](\mathbf{x}_t), \text{ composed of} \tag{1}$$

$$\mathbf{u}_t = e(\mathbf{x}_t;\theta_e),\ \mathbf{v}_t = b(\mathbf{u}_t;\theta_{b,t}),\ \mathbf{y}_t = d(\mathbf{v}_t;\theta_d), \text{ evaluated by} \tag{2}$$

$$\text{Score} = \text{Metric}(\mathbf{y}_t)\ , \tag{3}$$

where $\mathbf{x}_t$ is the initial input space of information, $\mathbf{y}_t$ is the final task-relevant output space and $e(\,\cdot\,;\theta_e)$, $b(\,\cdot\,;\theta_{b,t})$, and $d(\,\cdot\,;\theta_d)$ represent the sequential *encoding*, *biological transformation*, and *decoding* functions respectively. The output sequence can be evaluated by a Metric$(\cdot)$ that produces a *task score* that is indicative of the performance as it relates to its intended task outcomes. We separately define a mechanism for driving BNN adaptation towards high-level objectives through *feedback* based on measurable performance characteristics. The feedback function $r(\text{Score};\theta_r)$, is parameterized by θ_r, guided by the score produced by the defined metric:

$$\theta_{b,t+1} = g(\; r(\text{Score};\theta_r)\;;\theta_{b,t})\ , \quad \text{where } b(\mathbf{u}_t;\theta_{b,t}) \neq b(\mathbf{u}_t;\theta_{b,t+1})\ . \tag{4}$$

A key differentiator of neurocomputation relative to conventional computing is that the core transformation is driven by biology, defined here as the mapping $b(\,\cdot\,;\,\cdot\,)$, which is inherently stochastic and adaptive as modeled by the *biological adaptation* function $g(\,\cdot\,;\,\cdot\,)$ (see Section 2.2). Conceptually, encoding and decoding serve as interfaces to modulate the inputs and outputs of a biological system whose underlying biophysical dynamics are significantly more complex and less explicitly modeled than the algorithmic structures used in silicon-based machine learning. Building on this, we differentiate *learning* as the process of driving BNN adaptations to improve measurable task-oriented performance, from *training*, which refers to the algorithmic optimization used in artificial neural

[1] The dataset and experiment code available at: https://github.com/Cortical-Labs/embodied-neurocomputation

networks. This terminology emphasizes that biological adaptation emerges from intrinsic biophysical plasticity rather than gradient-based parameter updates.

Designing a neurocomputation application requires the specification of each functional component and its respective parameters. We define the total parameter space at timestep t as $\theta_t = \theta_e \cup \theta_{b,t} \cup \theta_d \cup \theta_r$, comprising both tunable hyperparameters (*training*) ($\theta_e, \theta_d, \theta_r$) and adaptive biological parameters (*learning*) (θ_b). This unifies common BNN application paradigms, from physical reservoirs [24, 25] to goal-driven objectives via purposefully designed or optimized feedback [14, 15, 26–28], enabling systematic optimization across the neurocomputational pipeline. Selecting an optimal configuration in this time-dependent high-dimensional space is non-trivial, requiring simultaneous optimization of design variables and evolving biological dynamics toward a specific task objective.

2.1 Encoding and Decoding

We define *encoding* as the process of transforming task specific information into a format that can be communicated to BNNs while maintaining compatibility with hardware responsible for biological-digital interface. While our framework is agnostic to specific types of hardware, we formulate encoding concretely in the context of electrical stimulation using micro-electrode arrays (MEAs), which is currently one of the most common forms of BNN interfacing technology [29, 13, 30]. Encoder functions based on MEA technology can be modeled as a transformation of task specific information into a stimulation matrix, subsequently delivered to the BNNs as electrical stimulation. We specify the *stimulation matrix* ($\mathbf{u}_t \in \{0,1\}^{C \times \tau_{\text{in}}}$) as a spatio-temporal representation of the encoding stimulus. This matrix encapsulates the physical parameters of stimulation into a single computation structure, where C is the stimulation channels and τ_{in} is the total number of pulse delivery time steps.

The encoder is parameterized by the union of two distinct sets, $\theta_e = \theta_{\text{task}} \cup \theta_{\text{stim}}$, reflecting its dual responsibility in translating task-specific requirements and configuring hardware-interfacing protocols. While *task parameters* (θ_{task}) are often arbitrary and depend on the specific computational objective, interface parameters (θ_{stim}) govern the physical delivery of stimulation, including but not limited to, stimulation frequency, stimulation amplitude, and pulse width. We provide concrete examples in Section 3 as part of our empirical evaluation.

Decoding is the inverse process of encoding, where we transform BNN responses captured by interfacing hardware into task-relevant formats. In MEA-based systems, we model the decoder as a function that transforms a response matrix ($\mathbf{v}_t \in \mathbb{R}^{C \times \tau_{\text{out}}}$), which captures continuous real-valued electrical potentials across C channels over τ_{out} time steps, into the required output. It is common to further transform these continuous potentials into discrete events known as *spikes*, representing neuronal action potentials, using a spike detection algorithm: $\mathbf{v}'_t = d'(\mathbf{v}_t)$, where $\mathbf{v}'_t \in \{0,1\}^{C \times \tau_{\text{out}}}$ is the binary matrix of detected spikes and $d'(\cdot)$ is the detection function [31–34].

2.2 Biological transformation

We model the BNN as a dynamical operator that maps the encoded stimulation matrix $\mathbf{u}_t$ to the response matrix $\mathbf{v}_t$. This mapping is non-stationary and depends on the temporal structure of the input, with parameters θ_b evolving according to stimulation and response history, external feedback, as well as through intrinsic spontaneous processes. In this framework, the mapping $b(\,\cdot\,;\,\cdot\,)$ can be interpreted as a qualitative change in the information contained in the input (in this case the stimulation matrix $\mathbf{u}_t$ at time step t). We refer to qualitative change to a transformation that does not simply scale or linearly filter the input, but reorganizes its informational structure by mapping raw inputs into functionally meaningful representations.[2] The mapping $b(\,\cdot\,;\,\cdot\,)$ therefore enables the BNN to extract and retain relevant features of the input, facilitating adaptivity of its internal state. This is expressed as:

$$\mathbf{v}_t = b\left(\mathbf{u}_t; \theta_{b,t}\right), \quad \text{with } \theta_{b,t} = g\left(\mathbf{i}_{t-1}; \theta_{b,t-1}\right), \tag{5}$$

where the parameters $\theta_{b,t}$ change over time and the input $\mathbf{i}_{t-1}$ is drawn from $\{\mathbf{v}_{t-1}, r(\text{Score}_{t-1}; \theta_r), \varnothing\}$. We include $\varnothing$ to account for spontaneous biological adaptation in the absence of external interaction [25, 35].

[2] E.g., continuous signals transformed into patterns that capture relational or dynamical features, revealing hidden structure not accessible from raw input.

The set of parameters $\theta_{b,t}$ serve as a mathematical proxy for the network's adaptive state, implicitly capturing underlying mechanisms such as specific plasticity rules, multiple timescales of adaptation, structural changes in connectivity, and biophysical processes, without explicitly modeling them. The recursive dependence (5) implies that the transformation implemented by the network is not fixed, but evolves as a function of the system's own activity. Thus, the response at time step t depends not only on current and past input, but also on how previous outputs shaped the transformation itself. Systems with this property can be regarded as "third-order" information-processing systems [36]. Here, the resulting transformation is non-invertible, reflecting reorganization and compression of information. Importantly, we treat encoding and decoding parameters as fixed for each learning phase, such that $\theta_{e,t} = \theta_{e,t-1}$ and $\theta_{d,t} = \theta_{d,t-1}$, so that adaptation is localized to the BNN.

2.3 Feedback

Finally, we define *feedback* as a special form of encoding designed to purposefully drive BNN adaptation towards high-level objectives. Similarly, feedback is hardware agnostic and the techniques involved may range from purposefully crafted electrical stimulation to the dispensation of chemical mediators such as neurotransmitters [26, 27, 37, 38]. Feedback is a stimulus delivered after the BNN responds to an initial stimulus, with the objective of modulating the mapping between that initial stimulus and the resulting biological state. Feedback can broadly be either 1) *reinforcing* (r^+) to encourage and reinforce mappings that lead to favorable (from the perspective of the designer) task outcomes or, 2) *plasticity inducing* (r^-) to encourage development of alternative mapping structures.

3 Empirical Evaluation

Task definition. To demonstrate practical application of our framework, we create a simulated goal-driven navigation environment, which tasks an agent to navigate a barrier enclosed gridworld to reach a randomly spawned goal position referred to as *food*, guided by a scalar sensor value referred to as *odor* (see Figure 2A). In each step, the agent can take one of three actions: `move forward`, `turn left`, or `turn right`. Successful goal acquisition by the agent leads to a score increment which relocates the food, while barrier collisions decreases the score (Appendix A.6). Task performance is evaluated on the total score acquired over a set number of steps across three modes: A) `30 steps over 1 episode`, B) `150 steps over 1 episode` and C) `30 steps over 5 episodes` spaced with 2 minutes rest. Episodes are reset using a common seed to ensure deterministic goal presentation.

Experimental setup. We implement a scaled distributed optimization platform consisting of an *optimization server* using the Optuna hyperparameter optimization framework (HPO) [39] and multiple connected Cortical Labs CL1 *clients* via the Cortical Cloud service (see Figure 2B).[3] Each CL1 encloses a single MEA comprising a unit of *neuron culture* [21].[4] To address biological stochasticity, we evaluate identical parameter sets across multiple simultaneous CL1 clients, using their aggregated score as a single representative outcome.

Parameter optimization and evaluation. We focus optimization on encoding hyperparameters θ_e, which also dictate delivery of feedback pulses, while using relatively simple and fixed decoding and feedback mechanisms, to isolate their specific contributions from the broader, highly-coupled system dynamics. Specifically, we evaluate 1,296 parameter combinations for θ_e (see Table 1) derived from a prioritized subset of six high-impact encoding parameters, selected via domain expertise to ensure an experimentally tractable search space. The optimization proceeds in two sequential stages across four experimental groups: Stage 1 (Groups 1 and 2) explores identical parameter sets under differing biological and hardware (biophysical) configurations using mode A, while Stage 2 (Groups 3 and 4) refines top performing Stage 1 parameters under common biophysical setup to evaluate modes B and C respectively (see Figure 2C). To ensure the stability of Stage 1 parameters, we performed multiple experimental replicates (4x) for each configuration in Stage 2, which results in a total of 256 trials.

Encoding. The environment presents a scalar sensor value (Appendix A.6) which indicates that the odor field is strongest: (1) to the right or behind, (-1) to the left or (0) in front of the agent. Using a simplified representation allows us to decouple the study of BNN encoding techniques from the complexities of environment translation, while ensuring that modern silicon-based agents can achieve high performance to provide a stable benchmark for comparison.

[3] Accessible: https://corticallabs.com/cloud

[4] Power consumption for maintaining and interfacing biological substrates on the CL1 is quantified in Appendix A.9.

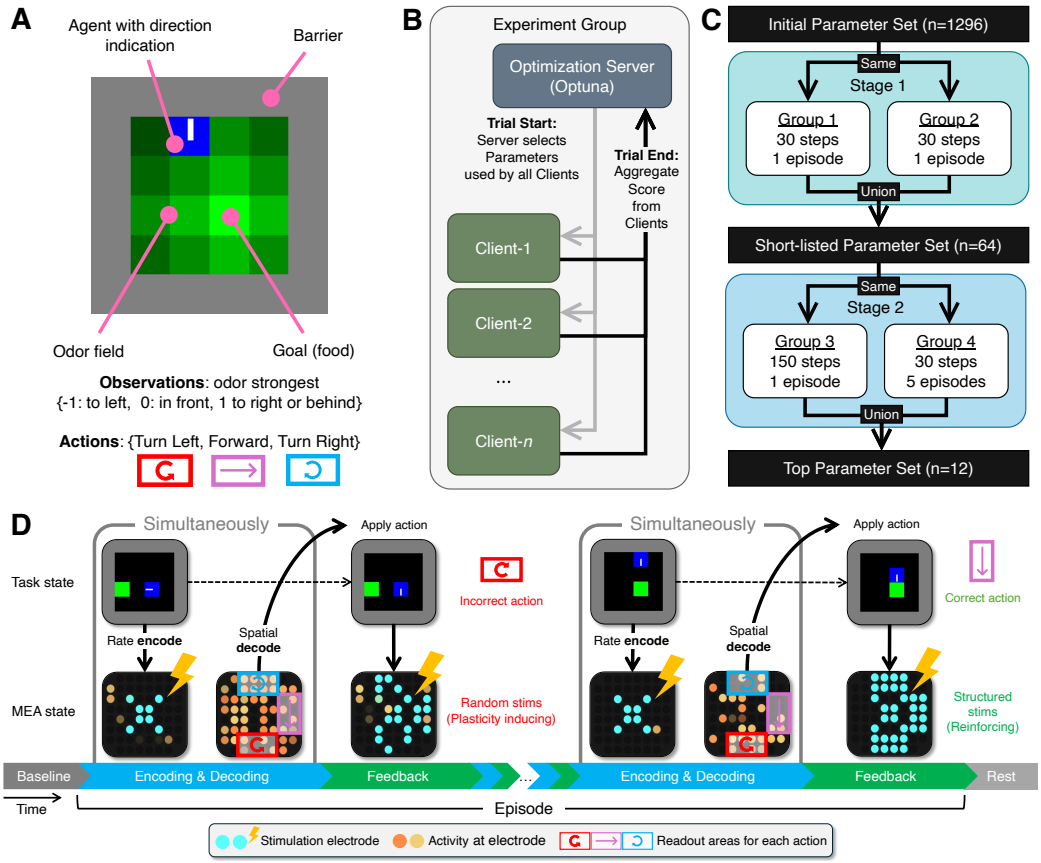


Figure 2: Experimental setup for framework evaluation. **A**: Simulated goal-driven environment featuring an agent, food source, and odor gradient within a 6×6 gridworld. **B**: Distributed optimization architecture using an Optuna server to dispatch parameters to parallel clients for aggregate trial scoring. **C**: Two-stage parameter screening pipeline: Stage 1 reduces 1,296 initial combinations to a shortlist ($n = 64$), and Stage 2 identifies the final 12 optimal configurations. **D**: Closed-loop interaction cycle, illustrating task encoding, spatial decoding, and feedback types.

We use rate encoding (Appendix A.4), which translates the sensor value to a sequence of stimulations spaced by a frequency (Hz) that is interpolated between a `minimum frequency` (Hz) and `maximum frequency` (Hz). The `stimulation amplitude` (µA) and `pulse width` (µs) define the charge delivered by symmetrical biphasic rectangular pulses with a negative-polarity leading phase. Finally, we refer to a *tick* as a discrete unit of time in which rate encoding is delivered, parameterized by `tick rate` (Hz). The rate at which the agent interacts with the environment is measured in the number of ticks that passes for each step of the environment (referred to as `ticks per step`). The real-time speed of agent-environment interactions is governed by `tick rate` / `ticks per step` (see Figure 2D and Appendix A.5). A breakdown of parameter values screened is listed in Table 1.

Decoding. For all experiments, we use a *count decoding* approach adapted from [40]. During each interaction period, spikes identified by the CL1's on-board spike detector are aggregated within three distinct spatial regions of the micro-electrode array (see Appendix A.3), where each region corresponds to a specific action. Total spike counts are normalized against a baseline of spontaneous activity spanning the 60 interactions immediately preceding each episode. The agent then executes the action associated with the region exhibiting the highest relative spike density.

Feedback. We use approaches adapted from [40] to deliver feedback after each agent-environment interaction, guided by the environment *reward* (see Appendix A.6). When reward value > 0, we deliver five structured bursts of 100 Hz pulses (80 ms duration) spaced across the feedback duration to both encoding and decoding regions to reinforce neural activity. To induce plasticity (reward ≤ 0), we use random stimulation to the same regions, where each stimulation electrode has an activation

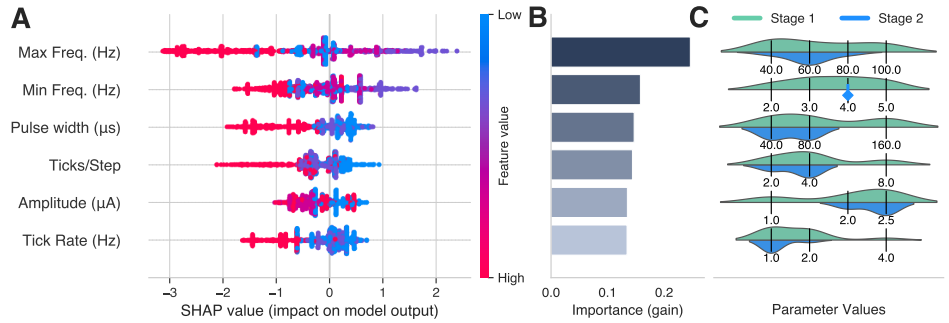


Figure 3: **A**: SHAP contributions to Stage 1 top 1% performance (see Section 4.1). **B:** XGBoost feature importance in Stage 1. **C:** Top 1% parameter distributions across Stages 1 and 2.

probability of $p = 1/3$ and temporal pattern generated by sampling inter-stimulation intervals derived from a 3-25 Hz frequency range. For both modes, stimulation lasts twice as long as the preceding interaction period, with the same regimen used across all experiments. While stimulation amplitude and pulse width are inherited from the encoding configuration, feedback procedure remains fixed.

Biological. For Stage 1, we screen identical parameter sets under task mode A using two distinct yet standard biophysical setups. For Group 1 we use seven cortical/hippocampal (1:1 ratio) cultures within a Polydimethylsiloxane (PDMS) ring, and nine cortical/hippocampal (1:1 ratio) cultures on astrocytes in a monolayer for Group 2. For Stage 2, we evaluate task modes B and C using comparable setups. In both Groups 3 and 4, we use five cortical-only cultures in a monolayer with a consistent ratio of cultures with and without astrocytes. See Appendices A.1 and A.2 for details. Depending on culture activity, Groups 1, 3, and 4 utilize CL1 as a mini-incubator with manual daily media changes, while Group 2 employs the onboard perfusion pump for automated exchange.

Importantly, the encoding, decoding, and feedback modules are fixed, i.e., non-trainable and non-adaptive, during each trial. Hence all observed behavior related changes are due to the adaptation of the biological cultures during the trial.

4 Results

4.1 Identifying top parameters

Table 1 reports the range of parameters screened in each experimental stage as well as the top performing parameter values identified. To understand the importance of each parameter combination on overall performance, we fitted an XGBoost [41] classifier with a binary objective to predict whether a parameter combination would yield a task score in the top 1% (`class 1`) or otherwise (`class 0`). We use aggregate scores from Stage 1 where a minimum of 4 clients have responded to ensure each score is sufficiently representative of the parameter set's performance across multiple cultures. We then applied Shapley Additive Explanations (SHAP) [42, 43], which quantifies the contribution of each feature toward the change in the model's expected output, enabling us to identify which

Table 1: Experimental parameter combinations across stages, categorized by: rate encoding (min/max frequency), stimulation delivery (amplitude, pulse width), and temporal gameplay characteristics (tick rate, ticks per step) (see Section 3). **Bold** indicates top-performing parameters after each stage.

Parameter	Experiment Stage		
	Stage 1 (n=1296)	Stage 2 (n=64)	Top (n=12)
Min Frequency (Hz)	2.0, 3.0, **4.0**, 5.0	**4.0**	4.0
Max Frequency (Hz)	**40.0**, **60.0**, **80.0**, **100.0**	**40.0**, **60.0**, **80.0**, 100.0	40.0, 60.0, 80.0
Amplitude (μA)	1.0, **2.0**, **2.5**	2.0, **2.5**	2.5
Pulse Width (μs)	**40.0**, **80.0**, 160.0	**40.0**, **80.0**	40.0, 80.0
Tick Rate (Hz)	**1.0**, **2.0**, 4.0	**1.0**, **2.0**	1.0, 2.0
Ticks per Step	**2**, **4**, 8	2, **4**	4

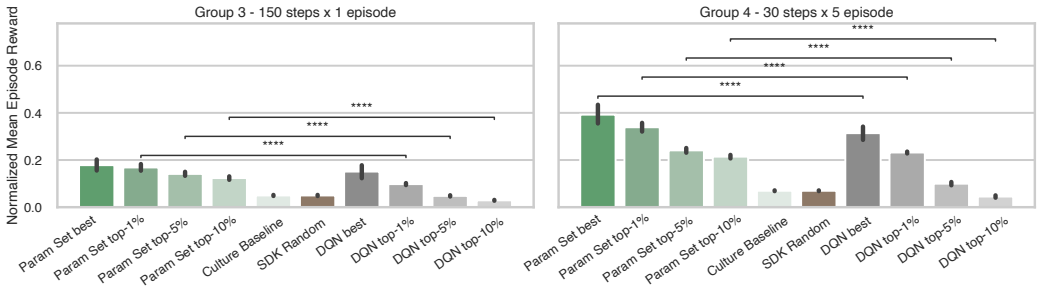


Figure 4: Performance of BNN, DQN, and baseline agents in (left) Group 3 `150 steps` $\times$ `1 episode` and (right) Group 4 `30 steps` $\times$ `5 episodes`. Bars represent mean $\pm$ 95% CI. Significance determined via Brunner-Munzel tests, where ****: $p < 10^{-4}$.

encoding parameters most significantly drive high-performance outcomes. As shown in Figure 3, we find that: (1) maximum frequency for rate encoding is the strongest driver, favoring moderate values of 40-60 Hz, and (2) higher stimulation amplitude, shorter pulse width, and fewer ticks per step, corresponding to faster environmental interaction rates, all support improved performance.

Based on these insights, we reduced our initial pool of $n = 1,296$ combinations to a shortlist of $n = 64$ for Stage 2, from which we derived $n = 12$ combinations exhibiting consistent improvement over baseline controls (Table 1). For baselines, we use non-adaptive neuronal data via the CL-SDK Simulator[5] comprising simulated random spikes (*SDK Random*) and replayed spontaneous activity (*Culture Baseline*). Specifically, for Stage 1, top performance is defined as trials exceeding the 99th percentile of *SDK Random* scores across both Groups 1 and 2.

4.2 BNNs learn to navigate

In Figure 4, we present Stage 2 results comparing Group 3 (a single 150-step episode) with Group 4 (five 30-step episodes separated by 2-minute rests). We report the normalized mean episode reward, which is the ratio of total reward to the maximum achievable reward. In both groups, optimized culture specific parameter sets demonstrated consistent performance gains over the baseline controls.

We employ an optimized Deep-Q Network (DQN) [44] as a representative of modern reinforcement learning to assess whether biological agents can acquire task-relevant behavior through learned strategies. The DQN comparison puts the BNN performance into perspective, as comparisons between biological and *classical* machine learning systems are inherently difficult. Rather than solely comparing maximum performance, it provides a reference for task difficulty and the qualitative degree of learning achieved by the BNN. More details on the DQN setup can be found in Appendix A.7. While the low-dimensional nature of this task does not necessitate high-capacity algorithms, we avoid rule-based approaches, such as heuristic decision trees or tabular methods, as they are insufficient for capturing the complex learning dynamics characteristic of biological and artificial neural networks. Statistical analysis using Brunner-Munzel tests [45] of the same top $x\%$ of parameters, which account for differences in sample sizes, distributions, and variances, shows that BNN agents achieve significantly higher performance than corresponding DQN groups (Figure 4) with 1.18x and 1.25x improvement for Group 3 and 4 respectively, and by 3.6x and 5.6x relative to culture baseline.

More interestingly, Figure 4 shows that BNN agents achieve higher performance when learning is distributed across multiple episodes separated by rest periods, compared to a single continuous episode with equivalent number of interactions. In Figure 5A, we observe a clear learning progression starting from episode three, with peak performance achieved by using optimal parameter sets for each BNN agent. Examples of behavioral traces for episode one and episode five (*left* and *right*, respectively) show a distinct shift toward goal-directed behavior. We find that this learning is characterized by a more strategic use of the `move forward` action, deploying it at the right moments to enable goal acquisition (Figure 5B). This is supported by a distinct shift in network activity (Figure 5C) with initial activity primarily evoked near the encoding electrodes becoming more diverse and distributed across the decoding regions of the network by episode five.

[5]Accessible: https://docs.corticallabs.com/#cl-sdk-simulator

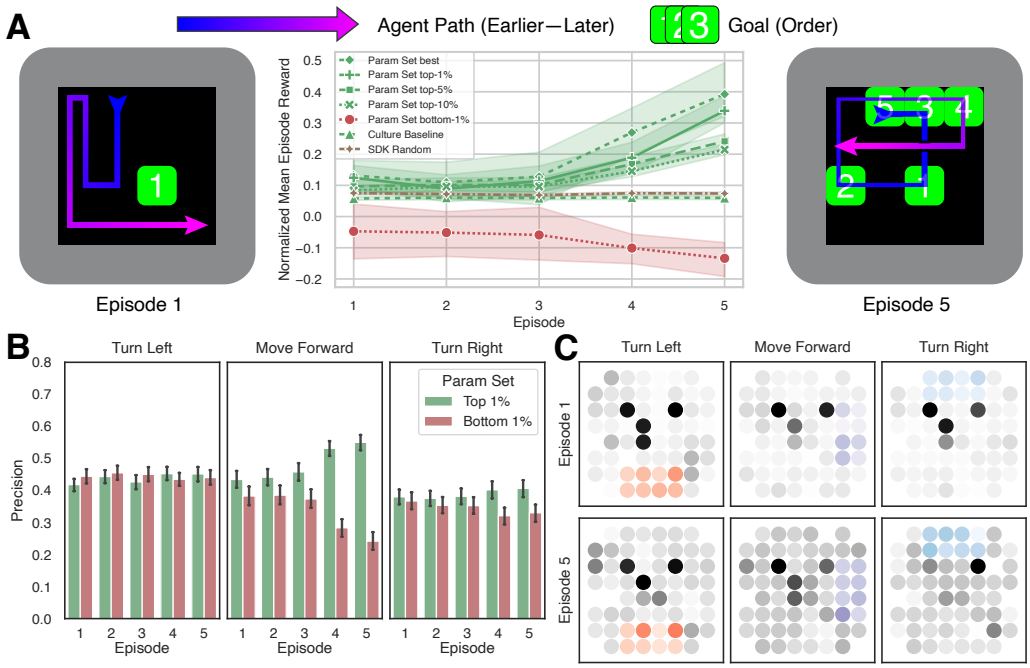


Figure 5: **A**: Group 4 learning progression, showing task performance (mean $\pm$ 95% CI) across five episodes for various parameter sets and baselines, with examples of agent traces for episodes 1 and 5 (*left* and *right* respectively). **B**: Precision of decoded actions associated with reinforcing feedback (mean $\pm$ 95% CI) for the top and bottom 1% parameter sets in (A). **C**: Heat map showing average of relative spike counts across the MEA during episode 1 and 5 corresponding to examples in (A).

4.3 Disentangling biological contribution

Our task definition requires the BNN to adapt to fulfill a spatial gap between encoded observations and decoded actions. To determine whether the observed learning is attributable to biological processes, we test biological contribution through a formal hypothesis test (refer to Appendix A.8 for full details and results). We evaluate: H_1: (Emergence), where the BNN adapts to bridge the spatial gap so the agreement between observations and actions increases over time, and H_0: (Null), where decoded action remains independent of the observation.

We apply this test to the Top and Bottom 1% performing configurations in the Group 4 experiments. We also introduce two additional ablation controls, Permuted and Matched Media, to isolate the role of the BNN by permuting spatial decode mapping and using media without cultured neurons, respectively (Figure 6). The results support H_1 and contradict H_0, as the top-performing configurations demonstrate a significant positive agreement slope while the ablation controls exhibit near-zero agreement. These findings indicate that learning performance is both biologically driven and spatially dependent. Furthermore, the significant difference in learning rates between the Top and Bottom groups ($p < 0.001$) suggests that the emergence of biological learning is highly sensitive to hyperparameter selection, with the negative slope observed in the Bottom group serving as evidence of non-trivial, though non-task-aligned, biological adaptation.

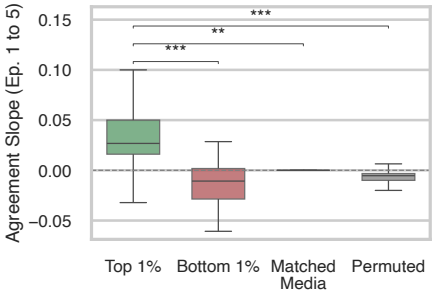


Figure 6: Agreement slope between observations and actions across Episodes 1 to 5 for the Top 1% and Bottom 1% configurations in Group 4, including Matched Media and Permuted ablation controls. Positive slope indicates the emergence of task-aligned learning.

5 Limitations

Search space constraints: To ensure experimentally tractable parameter optimization, we restricted our search to a prioritized subset of encoding parameters within predefined ranges based on domain expertise. While this approach enabled an unprecedentedly large-scale screening of 1,296 combinations, it remains possible that optimal configurations reside beyond these initial boundaries. Nevertheless, the identified parameter-importance trends identified here provide a foundation for future work by guiding the expansion of the search space into broader encoding regimes or additional framework components, such as decoding and feedback.

Temporal dependencies: To prevent systematic bias, we employ a non-monotonic parameter delivery schedule across trials to avoid confounding of parameter values with trial progression. However, since the rest period between trials (2 minutes) is identical to the rest period between episodes in Group 4, carry-over effects from previous stimulation regimes cannot be ruled out. This uncertainty raises a compelling question regarding whether the performance lag observed in Group 4, where learning gains emerge only after the third episode, represents the emergence of longer-term adaptive processes. Although we refrain from using the term *memory* as it is not yet formally defined within our framework, investigating these temporal couplings offers an exciting direction for future work to characterize the specific timescales required for retention and resetting of biological states.

Task complexity and generalizability: Our evaluation utilized a low-dimensional, scalar sensory input within a controlled gridworld environment as an initial proof of concept. While it remains unknown whether the integrated neurocomputational pipeline, comprising encoding, decoding, and feedback components, can generalize to high-dimensional, continuous sensory streams, our results provide a baseline for future comparative studies. This enables more purpose-driven interface strategies and their systematic validation in increasingly complex environments.

Transferability of top parameters: To reduce the possibility of culture-specific selection, identical parameter configurations were evaluated across multiple cultures, and their aggregated performance was used as score. We also included distinct biological culture configurations across the experimental groups. The resulting parameters were based on the average response of multiple cultures tested over multiple sessions. We applied these identified parameters to different cultures that were tested during different sessions. We found that learning was reproduced, suggesting that the top parameters have transferability across cultures. How far this transfer extends is a question for future work.

Feedback-encoding overlap: Since feedback is a specialized form of encoding, the two modules may not be fully independent and share a subset of stimulation parameters in our experiments. This design choice prevents a fully decoupled optimization of each component. However, we used a (partially) shared parameter setting to ensure that the observed effects reflect general stimulation properties rather than module-specific tuning. This provides a consistent baseline that can inform future work aimed at disentangling and optimizing stimulation protocols for each component individually.

6 Conclusion

We introduced *Embodied Neurocomputation* as a systems-level framework for interfacing biological neuronal networks (BNNs) with digital environments through encoding, biological transformation, decoding, and feedback. We operationalized this framework as a coupled optimization problem and validated it through a large-scale evaluation of approximately 1,300 encoding configurations across 26 neural cultures, spanning $\sim$4,000 hours of real-time closed-loop agent-environment interactions.

Our results show that BNNs can support goal-oriented behavior when embedded in a sensorimotor loop. A small subset of encoding regimes produced consistent learning and significantly outperformed non-adaptive culture baselines and silicon DQN agents trained with the same number of interactions. These findings indicate that the computational potential of living neural substrates depends not only on the biological network itself, but also on the structure of the bio-silicon interface.

To validate the approach in this study the search space was necessarily constrained, with a low-dimensional sensory input, and the temporal dependencies underlying multi-episode improvement require further characterization. Future work should extend optimization to decoding and feedback, evaluate richer sensory environments, and directly probe the timescales and mechanisms of biological adaptation. By moving beyond heuristic stimulation protocols toward systematic interface optimization, this work provides a foundation for scalable, efficient, and adaptive bio-silicon computation, with long-term implications for low-power intelligent systems and hybrid biological-digital architectures.

References

[1] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and Policy Considerations for Deep Learning in NLP. *arXiv*, 2019.

[2] Asen Mehonic and Anthony J. Kenyon. Brain-inspired computing needs a master plan. *Nature*, 2022.

[3] Ed Bullmore and Olaf Sporns. The economy of brain network organization. *Nature Reviews Neuroscience*, 2012.

[4] Lena Smirnova, Brian S. Caffo, David H. Gracias, Qi Huang, Itzy E. Morales Pantoja, Bohao Tang, Donald J. Zack, Cynthia A. Berlinicke, J. Lomax Boyd, Timothy D. Harris, Erik C. Johnson, Brett J. Kagan, Jeffrey Kahn, Alysson R. Muotri, Barton L. Paulhamus, Jens C. Schwamborn, Jesse Plotkin, Alexander S. Szalay, Joshua T. Vogelstein, Paul F. Worley, and Thomas Hartung. Organoid intelligence (OI): The new frontier in biocomputing and intelligence-in-a-dish. *Frontiers in Science*, 2023.

[5] Lukas Wegmeth, Tobias Vente, Alan Said, and Joeran Beel. Green Recommender Systems: Understanding and Minimizing the Carbon Footprint of AI-Powered Personalization. *arXiv*, 2025.

[6] Daniel A. Wagenaar, Radhika Madhavan, Jerome Pine, and Steve M. Potter. Controlling Bursting in Cortical Cultures with Closed-Loop Multi-Electrode Stimulation. *The Journal of Neuroscience*, 2005.

[7] Daniele Poli, Vito P. Pastore, and Paolo Massobrio. Functional connectivity in in vitro neuronal assemblies. *Frontiers in Neural Circuits*, 2015.

[8] John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 2014.

[9] Juan A Gallego, Matthew G Perich, Lee E Miller, and Sara A Solla. Neural manifolds for the control of movement. *Neuron*, 2017.

[10] Goded Shahaf and Shimon Marom. Learning in Networks of Cortical Neurons. *The Journal of Neuroscience*, 2001.

[11] Sergi Valverde. Breakdown of Modularity in Complex Networks. *Frontiers in Physiology*, 2017.

[12] Satoshi Moriya, Hideaki Yamamoto, Ayumi Hirano-Iwata, Shigeru Kubota, and Shigeo Sato. Quantitative Analysis of Dynamical Complexity in Cultured Neuronal Network Models for Reservoir Computing Applications. *International Joint Conference on Neural Networks (IJCNN)*, 2019.

[13] Dowlette-Mary Alam El Din, Leah Moenkemoeller, Alon Loeffler, Forough Habibollahi, Jack Schenkman, Amitav Mitra, Tjitse Van Der Molen, Lixuan Ding, Jason Laird, Maren Schenke, Erik C. Johnson, Brett J. Kagan, Thomas Hartung, and Lena Smirnova. Human neural organoid microphysiological systems show the building blocks necessary for basic learning and memory. *Communications Biology*, 2025.

[14] Brett J. Kagan, Daniela Duc, Ian Stevens, and Frederic Gilbert. Neurons Embodied in a Virtual World: Evidence for Organoid Ethics? *AJOB Neuroscience*, 2022.

[15] Benedikt Maurer, Vaiva Vasiliauskaitė, Julian Hengsteler, Gino Cathomen, Tobias Ruff, Cedric Schmid, János Vörös, and Stephan J. Ihle. Reinforcement learning for closed-loop optimisation of spatiotemporal stimulation in patterned neuronal networks. *bioRxiv*, 2026.

[16] Alon Loeffler, Forough Habibollahi, Kwaku Dad Abu-Bonsrah, Azin Azadi, Candice Desouza, Hui Wen Chan, Yusei Nishi, Johnson Zhou, Finn Doensen, Hideaki Yamamoto, Brad Watmuff, and Brett J. Kagan. Handwritten Digit classification with neural cultures is influenced by neural architecture, network dynamics, and decoding methods. *bioRxiv*, 2026.

[17] Md Sayed Tanveer, Dhruvik Patel, Hunter E. Schweiger, Kwaku Dad Abu-Bonsrah, Brad Watmuff, Azin Azadi, Sergey Pryshchep, Karthikeyan Narayanan, Christopher Puleo, Kannathal Natarajan, Mohammed A. Mostajo-Radji, Brett J. Kagan, and Ge Wang. Starting a synthetic biological intelligence lab from scratch. *Patterns*, 2025.

[18] Daniel Rosebrock, Sneha Arora, Naresh Mutukula, Rotem Volkman, Elzbieta Gralinska, Anastasios Balaskas, Amèlia Aragonés Hernández, René Buschow, Björn Brändl, Franz-Josef Müller, Peter F. Arndt, Martin Vingron, and Yechiel Elkabetz. Enhanced cortical neural stem cell identity through short SMAD and WNT inhibition in human cerebral organoids facilitates emergence of outer radial glial cells. *Nature Cell Biology*, 2022.

[19] Kwaku Dad Abu-Bonsrah, Candice Desouza, Forough Habibollahi, Hui Wen Chan, Brad Watmuff, Mirella Dottori, and Brett J. Kagan. A novel protocol for the efficient generation of all three major hippocampal neuronal sub-populations from human pluripotent stem cells. *bioRxiv*, 2026.

[20] Brett J. Kagan. Two roads diverged: Pathways toward harnessing intelligence in neural cell cultures. *Cell Biomaterials*, 2025.

[21] Brett J. Kagan. The CL1 as a platform technology to leverage biological neural system functions. *Nature Reviews Bioengineering*, 2025.

[22] Mingchen Zhuge, Changsheng Zhao, Haozhe Liu, Zijian Zhou, Shuming Liu, Wenyi Wang, Ernie Chang, Gael Le Lan, Junjie Fei, Wenxuan Zhang, Yasheng Sun, Zhipeng Cai, Zechun Liu, Yunyang Xiong, Yining Yang, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Neural Computers. *arXiv*, 2026.

[23] Daniel Tanneberg, Elmar Rueckert, and Jan Peters. Evolutionary training and abstraction yields algorithmic generalization of neural computers. *Nature Machine Intelligence*, 2020.

[24] Hongwei Cai, Zheng Ao, Chunhui Tian, Zhuhao Wu, Hongcheng Liu, Jason Tchieu, Mingxia Gu, Ken Mackie, and Feng Guo. Brain organoid reservoir computing for artificial intelligence. *Nature Electronics*, 2023.

[25] Takuma Sumi, Hideaki Yamamoto, Yuichi Katori, Koki Ito, Satoshi Moriya, Tomohiro Konno, Shigeo Sato, and Ayumi Hirano-Iwata. Biological neurons act as generalization filters in reservoir computing. *Proceedings of the National Academy of Sciences*, 2023.

[26] Ash Robbins, Hunter E. Schweiger, Sebastian Hernandez, Alex Spaeth, Kateryna Voitiuk, David F. Parks, Tjitse Van Der Molen, Jinghui Geng, Isabel Cline, Kenneth S. Kosik, Sofie R. Salama, Tal Sharf, Mohammed A. Mostajo-Radji, David Haussler, and Mircea Teodorescu. Goal-directed learning in cortical organoids. *Cell Reports*, 2026.

[27] Moein Khajehnejad, Forough Habibollahi, Alon Loeffler, Aswin Paul, Adeel Razi, and Brett J. Kagan. Dynamic Network Plasticity and Sample Efficiency in Neural Cultures: A Comparison with Deep Learning. *Cyborg and Bionic Systems*, 2025.

[28] Forough Habibollahi, Amitesh Gaurav, Moein Khajehnejad, and Brett J Kagan. Biological neurons vs deep reinforcement learning: Sample efficiency in a simulated game-world. *NeurIPS Workshop: Memory in Artificial and Real Intelligence (MemARI) Workshop*, 2022.

[29] Moein Khajehnejad, Forough Habibollahi, Ahmad Khajehnejad, Chris French, Brett J Kagan, and Adeel Razi. Graph-based representation learning of neuronal dynamics and behavior. *arXiv*, 2024.

[30] Sihan Hua, Yaoyao Liu, Jinping Luo, Shangchen Li, Longhui Jiang, Pei Wu, Shutong Sun, Li Shang, Chengji Lu, Kui Zhang, Juntao Liu, Mixia Wang, Huaizhang Shi, and Xinxia Cai. Microelectrode arrays cultured with in vitro neural networks for motion control tasks: Encoding and decoding progress and advances. *Microsystems & Nanoengineering*, 2025.

[31] R. Quian Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering. *Neural Computation*, 2004.

[32] Marius Pachitariu, Nicholas Steinmetz, Shabnam Kadir, Matteo Carandini, and Harris Kenneth D. Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *bioRxiv*, 2016.

[33] Jason E Chung, Jeremy F Magland, Alex H Barnett, Vanessa M Tolosa, Angela C Tooker, Kye Y Lee, Kedar G Shah, Sarah H Felix, Loren M Frank, and Leslie F Greengard. A fully automated approach to spike sorting. *Neuron*, 2017.

[34] Carlos Vargas-Irwin and John P Donoghue. Automated spike sorting using density grid contour clustering and subtractive waveform decomposition. *Journal of Neuroscience Methods*, 2007.

[35] Luca Ciampi, Ludovico Iannello, Fabrizio Tonelli, Gabriele Lagani, Angelo Di Garbo, Federico Cremisi, and Giuseppe Amato. Neuro-Inspired Visual Pattern Recognition via Biological Reservoir Computing. *arXiv*, 2026.

[36] Brett J. Kagan, Valentina Baccetti, Brian D. Earp, J. Lomax Boyd, Julian Savulescu, and Adeel Razi. A quantifiable information-processing hierarchy provides a necessary condition for detecting agency. *arXiv*, 2026.

[37] Fred D. Jordan, Martin Kutter, Jean-Marc Comby, Flora Brozzi, and Ewelina Kurtys. Open and remotely accessible Neuroplatform for research in wetware computing. *Frontiers in Artificial Intelligence*, 2024.

[38] Brett J. Kagan, Forough Habibollahi, Brad Watmuff, Azin Azadi, Finn Doensen, Alon Loeffler, Seung Hoon Byun, Bram Servais, Candice Desouza, Kwaku Dad Abu-Bonsrah, and Nicole Kerlero de Rosbo. Harnessing Intelligence from Brain Cells In Vitro. *The Neuroscientist*, 2025.

[39] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[40] Brett J. Kagan, Andy C. Kitchen, Nhi T. Tran, Forough Habibollahi, Moein Khajehnejad, Bradyn J. Parker, Anjali Bhat, Ben Rollo, Adeel Razi, and Karl J. Friston. In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. *Neuron*, 2022.

[41] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[42] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017.

[43] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2020.

[44] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

[45] Edgar Brunner and Ullrich Munzel. The nonparametric behrens-fisher problem: asymptotic theory and a small-sample approximation. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 2000.

[46] E. D. Adrian. The impulses produced by sensory nerve endings: Part I. *The Journal of Physiology*, 1926.

[47] E. D. Adrian and Yngve Zotterman. The impulses produced by sensory nerve-endings: Part II. The response of a Single End-Organ. *The Journal of Physiology*, 1926.

[48] E. D. Adrian and Yngve Zotterman. The impulses produced by sensory nerve endings: Part 3. Impulses set up by Touch and Pressure. *The Journal of Physiology*, 1926.

A Supplementary Materials

A.1 Cell Culture and Cryopreservation

hiPSC lines (CLV3.1, CLV4.4) were maintained under standard pluripotency conditions and differentiated into cortical neurons via a tri-inhibition protocol combining dual-SMAD inhibition with WNT and later MEK inhibition [18], or into hippocampal neurons (yielding DG, CA3, and CA1 subtypes, [19]). Cortical and hippocampal cultures were cryopreserved at DIV 30 and DIV 40, respectively, in CryoStor CS10 at 3×10^6 cells/mL using controlled-rate freezing and stored in liquid nitrogen until use. On the day of plating, vials were rapidly thawed at 37°C, washed in plating medium (CTX medium [19] + chroman-1 50 nM), and assessed for viability by Trypan blue exclusion; only preparations exceeding 90% viability were used.

A.2 MEA Chip Preparation, Co-Culture Assembly, and Maturation

Sixty-electrode MEA chips (Multi Channel Systems) were coated with poly-L-ornithine (0.1 mg/mL, Merck) followed by human laminin-521 (10 µg/mL, StemCell Technologies) to promote neuronal adhesion. For cultures with primary human cortical astrocytes (ScienCell; passage +1 post-thaw), these were seeded slowly on to the electrode area in 5 µL plating medium (as above) at 70,000 cells/chip into 200 µL plating medium and allowed to form a confluent monolayer over up to 3 days before neuronal addition. For cultures without astrocytes, a Polydimethylsiloxane (PDMS) ring was placed around the electrode area and allowed to dry overnight. In co-cultures, thawed cortical and hippocampal neurons were co-seeded at 100,000 cells each (1:1 ratio; 200,000 total) either onto the astrocyte layer, or directly into the coated surface inside the PDMS ring, in 5 µL plating medium, prior to flooding with 1 mL extra plating media after 1h. In cortical neuron-only cultures, the hippocampal component was excluded from the plating mixture. After 7 days, AraC 2 µM was added for a single day to inhibit proliferation, and media was changed to BrainPhys + SM1 supplement (1X) + Pen Strep (1X) + Glutamax (1X) + BDNF (10 ng/mL) + dbcAMP (250 µM) + ascorbic acid (200 µM) thereafter. Cultures were maintained at 37°C, 5% CO_2 with 50% medium changes 3 times every week, and matured for 60 days post-seeding prior to electrophysiological and information-processing assessment.

A.3 Micro-Electrode Array (MEA) Layout

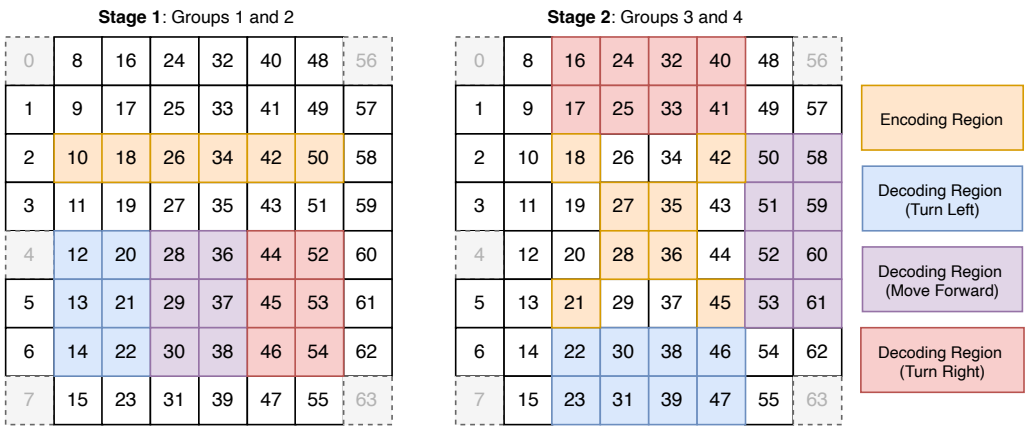


Figure 7: Spatial layouts used for encoding and decoding on the MEA. **Stage 1 Layout (left)**: A configuration placing encoding and decoding regions on opposite sides of the chip, which we find to be limiting due to spatial heterogeneity in neuronal coverage. **Stage 2 Layout (right)**: An improved design positioning the encoding region centrally and equidistant from spatially separated decoding regions to mitigate spatial bias. In both configurations, we use the same number of encoding and decoding electrodes.

A.4 Rate Encoding

Rate encoding [30, 40, 46–48] is a technique that communicates information to biological neurons by modulating the frequency of electrical stimulation. We implement rate encoding by interpolating scalar sensor information (x) between user-defined minimum (F_{min}) and maximum (F_{max}) frequencies, such that:

$$F(x) = F_{\text{min}} + (F_{\text{max}} - F_{\text{min}}) \left[\frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \right]. \tag{6}$$

At each environment interaction, we deliver rate encoding simultaneously (see Figure 8) to all electrodes within the defined encoding region (Figure 7). The duration of encoding is defined by the encoding parameters `tick rate` and `ticks per step`.

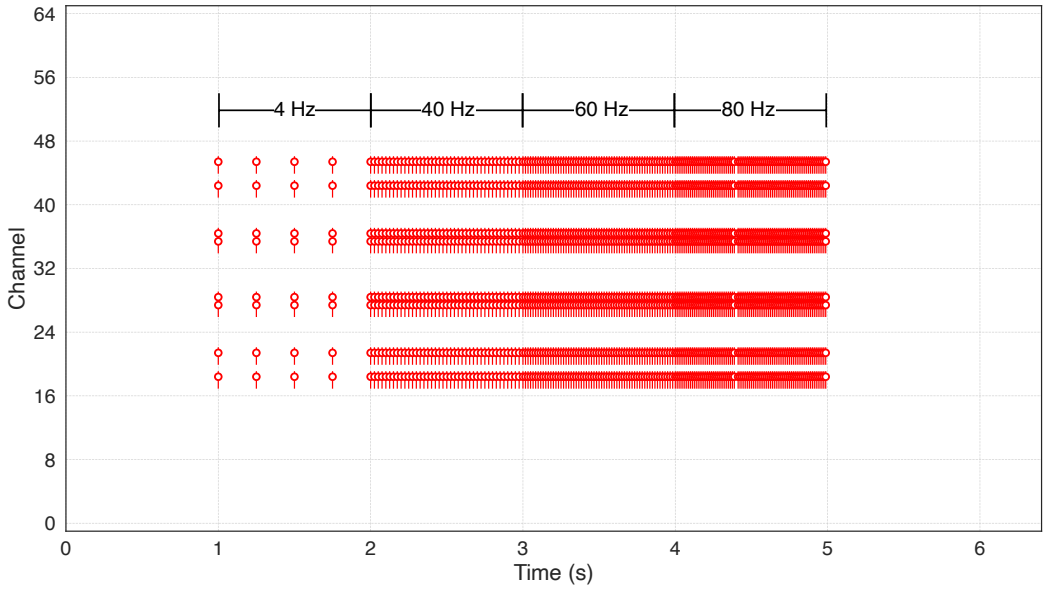


Figure 8: Raster plot showing four examples of rate encoding on the encoding channels being delivered at 4 Hz, 40 Hz, 60 Hz, and 80 Hz for 1 second each, with stimulation pulses marked in red.

A.5 Temporal Rollout

Schematic of temporal unfolding of environment state encoding and decoding is shown in Figure 9. At each game step t, sensor information is encoded and delivered as electrical stimulation to the biological neural networks (BNNs) via rate encoding while simultaneously collecting spikes detected by the CL1 device. The step$(\cdot)$ function advances the environment to the next timestep by decoding an action A_t from the spike events using count decoding and moving the agent. A reward value is returned for this interaction R_{t+1} which initiates suitable Feedback stimulation.

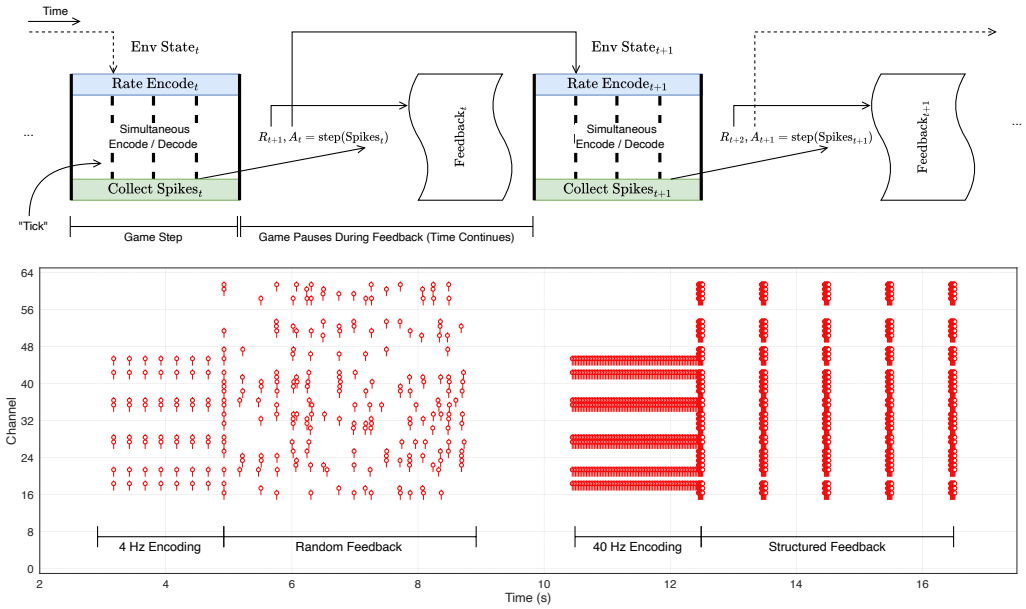


Figure 9: Schematic of temporal unfolding of environment state encoding and decoding (above) with an example raster plot marking stimulation pulses in red that correspond to two environment interactions: (1) 4 Hz encoding followed by random feedback, and (2) 40 Hz encoding followed by structured feedback.

A.6 Reward and Sensing Dynamics

The reward function R_t at timestep t (see Figure 9) is defined by the agent's interaction with the environment components, specifically the goal (*food*) and obstacle boundaries. The reward signal is composed of three distinct terms: (1) a terminal-style reward of $+2$ upon successful food acquisition; (2) a collision penalty of -0.2 when the agent strikes an obstacle or boundary; and (3) a continuous term representing the relative change in sensed odor intensity otherwise.

The sensing mechanism relies on an *odor sensor* $\mathcal{S} \in \{-1, 0, +1\}$, that tracks relative changes in an *odor field*, with intensity that decays exponentially with the distance d from the food source, formulated as $I(d) \propto e^{-\lambda d}$, where λ is the decay constant. The gradient of the field is discretized into three directional states that indicates the highest odor intensity relative to the agent's heading: (i) left $[-1]$, (ii) directly in front of the agent $[0]$, (iii) and, right or behind $[+1]$.

A.7 Deep-Q Agent

For the *DQN Agent*, we used a hidden layer size of 8 with `ReLU` non-linearity and an online learning setup matching the BNN setup, i.e., learning from each experience once, with the same number of total environment interactions as the BNN agents. We optimized the parameters `learning rate`, `target network update frequency`, `train frequency` indicating the number of steps before training is triggered, `final exploration` the final ϵ for the exploration, and `exploration horizon` the ratio of training steps for annealing ϵ. In total, we tested around $9,500$ parameter configurations to optimize these settings.

We report cut-off-early scores, for both, the reported DQN as well as the BNN agents, to compare their performance and learning dynamics given the same environment interaction budget. The parameters of the DQN are optimized for this learning horizon for a fair comparison. We ran additional DQN experiments with an increased budget of up to 1,000 episodes (200x times the original budget), using the original optimized parameters as well as using new optimized parameters for this setting. The DQN agents converge to a maximum normalized score around 0.75. Note that the BNN agents also seem to be not converged yet (see Fig. 5A). We also evaluated the number of episodes the DQN agents require to match the performance of the BNN agents after 5 episodes: all tested and optimized parameter settings for the DQN agents required at least four times more episodes as the BNN agents to equal their score.

A.8 Biological learning as a testable hypothesis

We model the task as the transformation $y_t = d(b(e(x_t, \theta_e), \theta_b), \theta_d)$ (Equation 1), where:

- $e(\cdot, \theta_e)$ is a fixed, spatially uniform, rate encoder that interpolates ordinal observation x_t into a scalar stimulation frequency,
- $d(\cdot, \theta_d)$ is a fixed, count based decoder over three spatially-distinct regions representing actions y_t, and
- $b(\cdot, \theta_b)$ is the adaptive biological neural network (BNN).

Since encoding is centralized and decoding is spatially distributed, the BNN must perform a non-linear spatial reorganization to map a central frequency based stimulus to one of the three equidistant periphery regions. More specifically, encoding simultaneously activates all electrodes within the encoding region with same stimuli intensity (yellow in Figure 7) and placement is equidistant and distinct from the decoder regions, creating a spatial gap.

This spatial gap ensures that any potential correlation between the encoded signal and the target action must be mediated by the BNN to bridge the physical distance between the input and output electrode regions. Since the parameters θ_e and θ_d are held fixed for all episodes within a trial, any emergence of task-relevant behavior must be attributable to changes in biology $b(\cdot, \theta_{b,t})$, which must transform the signal to satisfy this spatial requirement.

We, therefore, treat biological learning as a testable hypothesis through the following formulations:

- H_1 **(Emergence):** The BNN adapt such that the *agreement* between the decoded action y and the observation x exceeds chance and increases over time. This is suggestive of the emergence of biological learning to transform a central temporal stimulus into a specific spatial response. Formally, the mapping $x \xrightarrow{b} y$ is valid if and only if $b(., \theta_b)$ induces a non-trivial spatial reorganization of the input signal.
- H_0 **(Null Hypothesis):** The decoded action y remains independent of the observation x. This is representative of a *passive relay* scenario, where the biological network fails to reorganize, and the task performance remains at chance levels because the fixed encoder cannot bridge the spatial gap to the decoder. Formally, we refer to *passive relay* where the mapping $x \to y$ remains essentially a stochastic or identity-mismatched function, failing to establish a correlation between input x and correct action y despite BNN adaptation.

We define agreement as the interaction accuracy, $P(y_t = x_t)$, representing the probability of the decoded action matching the correct action given the observation per time step t. We use the slope of agreement, which is the change in accuracy across episodes, to quantify a rate of learning and

calculate the difference in slope between the Top percentile and each comparison group for hypothesis testing. We compute a 95% confidence interval by resampling the trials 2,000 times and determine significance (p-value) via a permutation test with 2,000 shuffles.

Table 2: Table reports agreement for the Group 4 experiments, chance agreement is 0.33.

Percentile	Comparison Grp.	Ep. 1	Ep. 2	Ep. 3	Ep. 4	Ep. 5	Slope/Ep.
1%	Top	**0.3929**	0.3265	**0.3648**	**0.3903**	**0.5281**	**+0.0334**
1%	Bottom	0.2143	0.1920	0.1964	0.1801	0.1592	−0.0122
1%	Matched Media	0.0657	0.0979	0.0936	0.0893	0.0929	+0.0046
1%	Permuted	0.3214	**0.3347**	0.3270	0.3219	0.2944	−0.0067
5%	Top	**0.3668**	**0.3577**	**0.3662**	**0.3980**	**0.4365**	**+0.0180**
5%	Bottom	0.2372	0.2357	0.2255	0.2036	0.2082	−0.0090
5%	Matched Media	0.0665	0.1117	0.0947	0.1062	0.0962	+0.0054
5%	Permuted	0.3266	0.3285	0.3268	0.3204	0.3127	−0.0036
10%	Top	**0.3545**	**0.3525**	**0.3569**	**0.3818**	**0.4196**	**+0.0159**
10%	Bottom	0.2774	0.2594	0.2685	0.2409	0.2375	−0.0098
10%	Matched Media	0.0664	0.1079	0.0902	0.1014	0.0973	+0.0055
10%	Permuted	0.3291	0.3295	0.3286	0.3236	0.3161	−0.0032

Table 3: Statistical significance of the learning rate difference (Top vs. Comparison Group)

Percentile	Comparison Grp.	Median slope	Mean slope	IQR	Sig. vs. Top
1%	Top	**+0.0268**	**+0.0334**	[+0.0161, +0.0500]	-
1%	Bottom	−0.0107	−0.0122	[−0.0286, +0.0018]	**0.0005**
1%	Matched Media	+0.0000	+0.0046	[+0.0000, +0.0000]	**0.0030**
1%	Permuted	−0.0054	−0.0067	[−0.0100, −0.0032]	**0.0005**
5%	Top	**+0.0179**	**+0.0180**	[+0.0036, +0.0339]	-
5%	Bottom	−0.0089	−0.0090	[−0.0214, +0.0036]	**0.0005**
5%	Matched Media	+0.0000	+0.0054	[+0.0000, +0.0143]	**0.0015**
5%	Permuted	−0.0036	−0.0036	[−0.0068, −0.0007]	**0.0005**
10%	Top	**+0.0143**	**+0.0159**	[+0.0036, +0.0286]	-
10%	Bottom	−0.0107	−0.0098	[−0.0214, +0.0036]	**0.0005**
10%	Matched Media	+0.0000	+0.0055	[+0.0000, +0.0134]	**0.0010**
10%	Permuted	−0.0029	−0.0032	[−0.0057, −0.0007]	**0.0005**

Tables 2 and 3 report results stratified by percentile (1%, 5%, and 10%). The comparison groups include the Top percentile, the Bottom percentile, as well as two ablation controls being Permuted group (combinatorial permutation of decoding regions[6], and the Matched Media group (no-cells control).

The data shown in Tables 2 and 3 support H_1 and are inconsistent with H_0. In the Top encoding configurations, the BNN acquires the required mapping across episodes, with agreement climbing from approximately 0.39 to 0.53. In Bottom configurations, agreement remains below chance and continues to drift lower. The learning-rate difference is significant at every threshold (permutation $p < 0.001$).

Comparison with the two controls demonstrates that the performance can be both attributable to the biology and is dependent on spatial-mapping. Media-only (No Cells) involves replaying the game on media without cultured neurons using parameter-matched configurations. This group shows flat agreement (slope ≈ 0) and remains far below chance ($\sim$0.10), suggesting that the task cannot be solved by the silicon interface alone. Permute Decode[6] involves replaying the top recordings with the permuted decoding mapping. This collapses the learning trend to chance (slope ≈ -0.006),

[6]Permuted group involves 6 combinatorial permutations where Permuted 1 is the true mapping and Permuted 2 to 6 are shuffled. For brevity, we have reported the aggregate average in Tables 2 and 3, while Table 4 shows the full Permuted results.

Table 4: Full results from the decoder permutation. Permuted 1 and 2 maps actions as `[Left, Forward, Right]` and `[Right, Forward, Left]`, respectively. Since the emergent learning is mainly attributable to the Forward action (Fig. 5B), a lower positive learning slope is retained.

Perc.	Decode	Ep. 1	Ep. 2	Ep. 3	Ep. 4	Ep. 5	Slope/Ep.
1%	Permuted 1 (true)	**0.3929**	0.3265	**0.3648**	**0.3903**	**0.5281**	**+0.0334**
1%	Permuted 2	0.3342	0.3010	0.3316	0.3699	0.4082	+0.0217
1%	Permuted 3	0.3597	0.3801	0.3418	0.3189	0.3469	−0.0087
1%	Permuted 4	0.3061	0.3189	0.3265	0.3112	0.2449	−0.0130
1%	Permuted 5	0.3138	0.3648	0.3597	0.3163	0.2730	−0.0130
1%	Permuted 6	0.2934	0.3087	0.2755	0.2934	0.1990	−0.0204
5%	Permuted 1 (true)	**0.3668**	0.3577	**0.3662**	**0.3980**	**0.4365**	**+0.0180**
5%	Permuted 2	0.3464	0.3294	0.3452	0.3628	0.3815	+0.0104
5%	Permuted 3	0.3611	0.3549	0.3577	0.3424	0.3407	−0.0053
5%	Permuted 4	0.2925	0.3158	0.2971	0.2948	0.2778	−0.0050
5%	Permuted 5	0.3430	0.3588	0.3566	0.3407	0.3277	−0.0049
5%	Permuted 6	0.2902	0.2834	0.2772	0.2613	0.2358	−0.0131
10%	Permuted 1 (true)	**0.3545**	0.3525	0.3569	**0.3818**	**0.4196**	**+0.0159**
10%	Permuted 2	0.3447	0.3452	0.3569	0.3664	0.3795	+0.0091
10%	Permuted 3	0.3560	0.3499	0.3577	0.3476	0.3415	−0.0031
10%	Permuted 4	0.2994	0.3049	0.2854	0.2860	0.2790	−0.0060
10%	Permuted 5	0.3513	0.3612	0.3696	0.3548	0.3426	−0.0024
10%	Permuted 6	0.2941	0.2863	0.2735	0.2634	0.2378	−0.0136

demonstrating that the task performance depends on the correct spatial-mapping for decoding rather than a dominant encoder driven policy.

Further, the marked divergence between Top and Bottom groups suggests that not only the biology is attributable to performance, configuration parameters can have a marked impact on whether the required spatial-mappings are developed. While the slope is negative for the Bottom group, it is important to note that the group starts from a significantly higher baseline ($\sim$0.21) than the Matched Media group ($\sim$0.10). We interpret the negative slope as evidence that the BNN is undergoing non-trivial (though non-task-aligned) biological adaptation, whereas the Matched Media group remains non-adaptive. As shown in Table 5, the difference in the learning rates (slopes) between these two groups is statistically significant.

Table 5: Comparison of slope estimates between Media and Bottom across percentiles.

Percentile	Comparison	Slope (Media)	Slope (Bottom)	Difference	95% CI	Perm. p
1%	Media vs Bottom	+0.0046	−0.0122	+0.0168	[+0.0052, +0.0285]	**0.0055**
5%	Media vs Bottom	+0.0054	−0.0090	+0.0144	[+0.0084, +0.0206]	**0.0005**
10%	Media vs Bottom	+0.0055	−0.0098	+0.0154	[+0.0112, +0.0198]	**0.0005**

A.9 Quantifying Energy Usage

It has been estimated that, the human brain consumes approximately 10-20 W with processing capacity of around 1 exaFLOPS [4]. However, a direct comparison using standard silicon metrics like FLOPS per Watt is problematic for two primary reasons:

1. Biological Neural Networks (BNNs) operate on non-von Neumann architectures where information processing relies on analogue, stochastic, and biochemical processes rather than discrete floating-point operations, and
2. the precise mechanistic formulation of BNN computation remains an open question in neuroscience.

Because BNNs do not perform classical arithmetic, they cannot be evaluated using floating-point metrics. We argue that our contribution, the Embodied Neurocomputation Framework, provides the necessary formal structure to allow for scalable and systematic studies using BNNs. As interaction techniques mature and the mechanisms through which BNNs process information become clearer, we expect that BNN processing capacities can be quantified. This will enable the development of fair and realistic energy benchmarks to compare biological and silicon-based architectures in the future.

Although we cannot perform a direct FLOPS comparison, we provide empirical data regarding the power consumption of the hardware supporting the biological substrate. Using 96,186 data points collected from 67 CL1 units over a two-day period, we quantified the power consumption required to maintain the biological interface. The results are summarized in Table 6.

Table 6: Power consumption required to maintain the biological interface.

Percentile	Power (W)
1%	10.6
5%	17.5
10%	18.0
50%	22.7
90%	23.8
95%	25.2
99%	25.5

This data represents hardware power consumption required for interaction and support of biological substrate under typical load within the Cortical Cloud service. In contrast, typical modern GPU enabled workstations may consume on average 579-641W [5]. We believe this provides a practical foundation for future researchers to calculate the relative efficiency of BNNs against silicon-based systems as biological formalisms mature. Such comparisons must account for the fact that neuron computation capacity is not yet well characterized, which makes direct comparisons with arithmetic-based metrics such as FLOPS incompatible.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims and contributions of the paper, the proposed framework and its validation, are described in Section 2 and evaluated in Section 3 and Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed explicitly in Section 5 and Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper focuses on empirical evaluation and does not present theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Experimental setup is explained in Section 3, additional details in Section 4, and in the supplementary material, which includes details for additional baseline configurations as well as biological culture handling. The environment and application code will be made available at the link in the manuscript.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: While the code for the environment and application will be released upon acceptance, the complete reproduction of results requires specific biological protocols detailed in Appendix A.1 and Appendix A.2. A dataset of electrophysiology recordings taken during the experiments will be made available at the link specified in the manuscript.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All details are given in Section 3, Section 4, and in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All reported results include confidence intervals and use statistical tests for significant claims as described in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The hardware and infrastructure used for the distributed optimization are detailed in Section 1 in the contribution paragraph. The computational requirements for the baseline agents are negligible due to their low-complexity architectures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research was conducted in accordance with the NeurIPS Code of Ethics. The study does not involve human participants, personally identifiable data, surveillance data, or datasets containing sensitive demographic attributes. The empirical work uses *in vitro* biological neural cultures interfaced with a simulated task environment through micro-electrode arrays, and the manuscript transparently reports the experimental setup, parameter optimization, baselines, statistical analyses, and limitations. All procedures involving biological material were conducted under the relevant institutional, biosafety, and regulatory oversight applicable to the authors' research setting.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper discusses the potential positive impact of embodied neurocomputation for developing efficient, adaptive, and biologically grounded computational systems. It also frames the work as an early-stage, controlled research framework rather than a deployed technology. Potential negative impacts are limited by the current experimental scope, but future bio-silicon systems may raise ethical and governance questions around responsible use, biological material, and claims about cognition or adaptive behavior in neural cultures. These risks are mitigated in the present work by using *in vitro* cultures in a simulated task environment, avoiding human data or real-world deployment, and transparently reporting limitations and experimental constraints.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: No such data or model will be released.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All used assets are credited accordingly.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No new assets released.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: No such experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The study does not involve human participants, crowdsourcing, or collection of human-subject data. Experiments used *in vitro* neural cultures, with all biological material handled under relevant institutional, biosafety, and regulatory approvals.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: No LLM involved in the experiments and framework.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.