Assembloid Agency: Unreal Engine API for brain-on-a-chip platforms

Jenn Leung
University of the Arts London
London, United Kingdom
`j.leung@fashion.arts.ac.uk`

Chloe Loewith
University of Cambridge
Cambridge, United Kingdom
`cl969@cantab.ac.uk`

Abstract

Assembloid Agency proposes an open-source Unreal Engine API for interfacing with brain-on-a-chip platforms [1], where the API mediates between living neurons cultured on high-density microelectrode arrays and simulated game environments. Building on 'Organoid Array Computing: The Design Space of Organoid Intelligence' [15], which speculates on a future where three-dimensional brain cultures assemble into more complex cognitive infrastructures and hence may become increasingly 'designable' and 'playable' organisms [2]. We extend this design space to apply games design principles to context engineering for organoid intelligence, treating organoids as polycomputational agents trained through reinforcement learning [24]. Our proposed plugin exposes functions for stimulation, recording, visualization, and real-time control of neuronal cultures within Unreal Engine while providing various RL-based game templates as behavioral benchmarking environments. This allows researchers to rapidly prototype experimental contexts, whether using biological wetware, spiking neural networks, or EEG signals as stand-ins. Game templates will be built on Unreal Engine's Learning Agents plugin to support single- and multi-organoid training scenarios.

Assembloid Agency embraces the dual embodiment of biologically engineered intelligence [10]. Here, "assembloid" refers to assembling organoids into new computational ecologies, while "agency" invokes not only the game engine as an experimental agent, but also the negotiated play of agency between biological and synthetic actors. By designing an API for game engines, we also anticipate the possibilities of interdisciplinary applications across games, interactive experiences, AI benchmarking, to architectural design, while foregrounding ethical and aesthetic guardrails for organoid intelligence.

1 Unreal Engine API for brain-on-a-chip platforms

Recent advances in organoid intelligence (OI) have begun to merge neuroscience, bioengineering, and computing, creating hybrid systems where living neural tissues function as adaptive information processors. This convergence has given rise to a new field of biocomputing that reimagines what counts as an intelligent or computational system. We propose developing an Unreal Engine API to interface with brain-on-a-chip platforms, which specifically refer to 3D cultures of brain organoids integrated with high-density microelectrode arrays capable of bidirectional electrical communication [1]. While originally developed for biomedical research, these platforms are now emerging as a novel substrate for biological computing, known as OI [24]. Pioneering work in this field by Cortical Labs and researchers from Johns Hopkins University has demonstrated the potential of these OI systems [12, 24].

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Creative AI Track.

Building on recent breakthroughs, including FinalSpark's NeuroPlatform [5] and Cortical Labs' DishBrain project [12], we aim to extend traditional AI gameplay paradigms to incorporate living neural systems. Our proposed Unreal Engine plugin will provide tools to: 1) record neuronal activity in real-time 2) deliver targeted electrical stimulation 3) train living neurons within virtual environments through reinforcement learning game templates. By creating an API that interfaces with biocomputers like the CL1 and simulators such as NEST simulator [6, 11], we enable a new generative approach to game design that incorporates living neural networks directly into game engines. This platform will allow researchers and developers to prototype organoid-based systems in rich, interactive 3D environments.

2 Methodology and related work

In Sec. 2 we will discuss methodology and theoretical foundations of the project. We propose that the API acts as a mediating membrane where biological and algorithmic agencies "contaminate" each other [21], dissolving model/reality divides and enabling distributed cognition across neurons, code, and simulated space [7].

2.1 Organoids as organisms of polycomputation

Prior work in 'Organoid Array Computing' establishes a framework for designing with Organoid Intelligence (OI) [15], proposing that brain organoids perform multiple simultaneous computations through chemical, mechanical, and biological processes. These interconnected systems "form the basis of a polycomputational system"[15]. This perspective extends Bongard and Levin's paradigm of biological polycomputation. As Bongard and Levin observe: "Biology is rife with polycomputing at all scales." Often, a single substrate could perform multiple computational tasks simultaneously [3], for example, OI systems combine morphogenetic cultures, microelectrode arrays, vascular scaffolds, and machine learning algorithms to create evolvable polycomputational agents.

Meanwhile, OI system architectures often depend on computational approaches stacked together, such as physical reservoir computing, where the organoid itself functions as a neural network layer [3], as well as multiple layers of reinforcement learning, which has proven effective for training organoids [12]. As reinforcement learning operates recursively across many layers of an OI system, it is apparent that a multilayered approach is necessary to mirror the poly-computational nature of the biological substrate itself.

2.2 API as mediating membrane

The Assembloid Agency API is designed to be a bi-directional interface between the game engine and neuronal cultures. The game engine delivers closed-loop electrical stimulation to organoids via high-density microelectrode arrays based on dynamic gameplay feedback [12], while organoids with neuroplastic spiking activities are mapped to reinforcement learning frameworks that reshape observations and reward functions [12, 5].

Following Parisi's framework, this membrane enables mutual "contamination" between biological and algorithmic agencies, dissolving conventional model/reality distinctions. The system, then, demands continuous renegotiation of fundamental reinforcement learning constructs: what constitutes state, action, or reward is dynamically co-defined. This approach responds to the "model-in-the-loop" paradigm which moves away from a linear understanding of neuroscience [22].

The interface between the organoid layer and the virtual environment enables agential fluidity. Following Hutchins' notion of distributed-cognition, cognition, much akin to Bongard and Levin's poly-computational framework, is present across artefacts, environments, and bodies [7]. With this, we also speculate that the API membrane facilitates the distribution of agency across scales.

2.3 Game Design as Agency Play

This approach draws on a tradition of using games as testbeds for artificial intelligence. Early examples such as Tesauro's TD-Gammon demonstrated that reinforcement learning through self-play could reach superhuman levels in backgammon without explicit strategic programming [25]. Two decades later, DeepMind's deep Q-network (DQN) achieved a major milestone by learning to play a diverse set of Atari 2600 games directly from raw pixel inputs, using a combination of convolutional neural networks, Q-learning, and experience replay [17]. The same architecture and hyperparameters were applied across multiple games, establishing the Arcade Learning Environment as a standard benchmark for general RL agents. AlphaZero combined deep neural networks with Monte Carlo Tree Search to learn and master chess, shogi, and Go entirely through self-play, starting from random play with only the basic rules provided, without any human game data or handcrafted domain knowledge [23]. OpenAI Five applied large-scale multi-agent RL to Dota 2, a partially observable, real-time strategy game requiring coordination, long-horizon planning, and adaptation to novel tactics [20].

Recently, large language models have been adapted to play games without explicit RL training, using in-context learning alone. In the Street Fighter III experiment [19], models received textual descriptions of the game state and recent moves, then selected the next move in real time. Interestingly, smaller, lower-latency models often outperformed larger ones, suggesting that reaction speed can outweigh raw model capacity in time-critical domains. These experiments also surfaced failure modes (hallucinating invalid actions, or refusing to play) that highlight the importance of grounding agents in the constraints of their environment.

Placing wetware into this lineage reframes these AI benchmarks: the "policy" is not a set of weights in silico but a living, plastic network whose synaptic configurations evolve in response to designed feedback loops. By providing standardized game templates, reward schemes, and data pipelines, the Unreal Engine API situates organoid intelligence within a comparative framework used for software agents. This continuity allows for direct methodological borrowing, from self-play to reward shaping, while extending the scope of what counts as an agent in game-based AI research.

3 Poly-computational layers of Assembloid Agency API

3.1 CL1/ NEST

The proposed toolkit will be implemented in two parallel configurations to accommodate both biological and simulated neural systems:

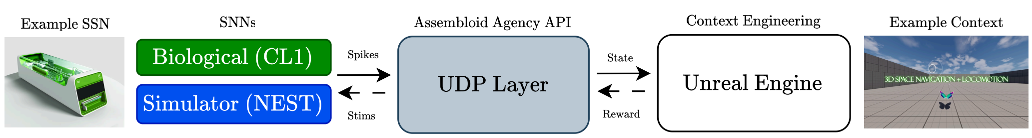
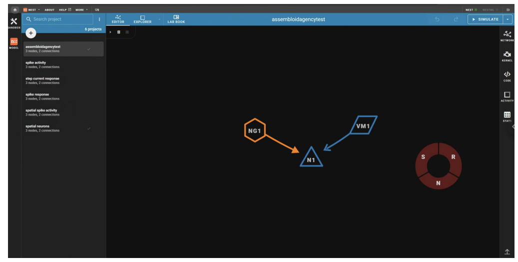


Figure 1: **Poly-computational layers between biocomputing systems and Unreal Engine.** Example biodirectional pipeline uses either CL1 or NEST Simulator. Arrow pairs show closed-loop symmetry: neural spikes → game state (top arrows), and game rewards → neural stimulation (bottom arrows).

CL1 Integration. For biological computing, the plugin integrates with Cortical Labs' CL1, a neuromorphic platform where living neurons cultured on high-density microelectrode arrays process information in real time through closed-loop electrophysiology. CL1 combines optimized cell culture conditions, FPGA/ASIC hardware for millisecond-precision stimulation and recording, and a Python API (CL-API) for bidirectional control [11]. In their recent Nature Reviews Bioengineering article, Kagan proposes that commercially viable devices such as CL1 could be used to benchmark generalized intelligence abilities, such as navigating game environments.

NEST Simulator. For users without access to biological components, the plugin supports the NEST Simulator (GNU General Public License v2) [6], a spiking neural network framework that replicates large-scale neural dynamics in software. NEST includes templates such as Pong and provides a lower-barrier environment for prototyping. Both CL1 and NEST share standardized interfaces within our plugin: identical state/reward mappings, spike-to-game-action translation layers, and compatible data protocols. This design ensures seamless transitions between simulated and biological systems while preserving a consistent development workflow.

(a) The CL1 biocomputer platform includes an integrated microelectrode array(MEA) and perfusion system for maintaining neuronal cultures. Adapted from [11].

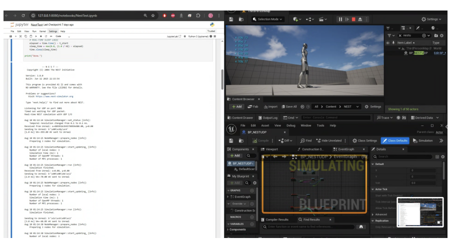


(b) NEST Desktop is a platform for spiking neural network simulation. Image shows a simple neural network implementation including a sample neuron, recorder, and stimulator.

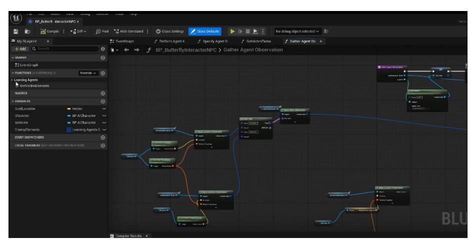
Figure 2: **Substrates for biological computing.** **See Sec.3 for details.

3.2 UDP communication

Both CL1 and NEST offer extensive documentation and example Jupyter notebooks for UDP-based communication. In CL1, the CL-API registers a remote host as a “spike firehose” target, opens a socket, and listens for reply packets, which are translated into stimulation API calls [13]. NEST similarly supports UDP spike-streaming, enabling both platforms to send and receive spike events in real time. On the Unreal Engine side, our implementation opens a UDP socket on a specified port and runs a listener thread. Incoming spike data is parsed and dispatched to the game thread via delegates or events, ensuring thread-safe updates to the game state. This bidirectional architecture allows for live stimulation, adaptive reward delivery, and dynamic environmental feedback. Figure 3a demonstrates an example UDP closed-loop communication between UE and NEST Simulator. Unreal Engine sends XY position data to NEST, while spike events in mV and stimulation timing pulses (1ms precision) is returned to Unreal Engine.



(a) UDP communication between Unreal Engine and NEST Simulator.



(b) Reinforcement Learning in UE5.

Figure 3: **Closed-loop interaction between biocomputing system and Unreal Engine.** **See Sec.3.2 and 3.4 for details.

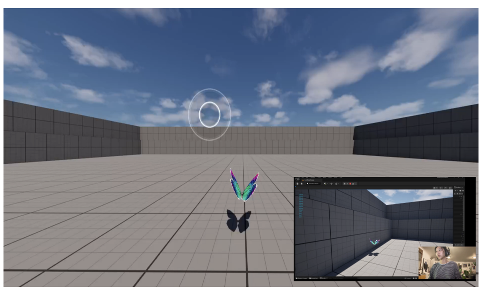
3.3 Plugin Functions

The Unreal Engine API plugin extends beyond simple UDP communication to create a closed-loop interface between game environments and living neuron cultures or organoids. Designed with both C++ and Blueprint access, it enables researchers and non-specialist users not only to send stimulation commands and receive spike responses, but also to manage the reinforcement learning process by delivering positive or negative feedback based on neuronal activity and record experimental data directly within Unreal. Through high-level functions such as `SendStimulus()`, users can specify which electrodes to activate (channels), firing frequency (Hz), pulse width (μs), stim current amplitude (μA), and duration (ms), translating game states into precise electrical stimulation patterns. `GetSpikeResponse()` retrieves low-latency neuronal activity (mV) with timestamp (ms) for gameplay feedback, while `SendRewardSignal()` provides positive or negative reinforcement to drive learning in the biological system. The plugin also includes tools to `VisualizeSpikes()`, RecordSessionData(), and ExportToCSV(), along with live data streaming via OSC or Spout for integration with other real-time visualization tools.

3.4 Game Templates

Inspired by OpenAI [4], we include pre-built Unreal Engine templates to accelerate gamified experimentation and to benchmark biologically engineered intelligence through structured task-based environments. The templates are designed with embedded reinforcement learning workflows using the Unreal Engine Learning Agents Plugin. Figure 3b displays a simple set-up of reinforcement learning observations, actions, and policy using the Learning Agents Plugin.

3D Navigation and FPS Game. Based on closed-loop feedback, Unreal Engine translates game states and player positions into spiking patterns that can be used for 3D navigation. Figure 4a demonstrates how an API user could visualize spikes as axis mappings (X/Y axis or forward velocity) for the player pawn, using spike rate on a set of channels to drive the magnitude or direction of movement. This simple 3D navigation task creates an open-loop mapping that tests for motor control, signal stability, and latency before moving onto closed-loop interactions. In Figure 4b, we begin to provide environmental sensory feedback to living neurons, placing them in a doom-like first-person shooter game context. Here, we propose to encode target object proximity as stimulation frequency, for example, near objects at 100 Hz bursts and far objects at 10 Hz. Direction can also be introduced via stimulating different electrode groups. In an FPS game, the overall spiking levels can be mapped as shooting velocity, benchmarking for adaptive reward learning.



(a) 3D navigation with spike-to-axis mappings. Spiking activities are mapped into movement inputs as movement direction and scale value.

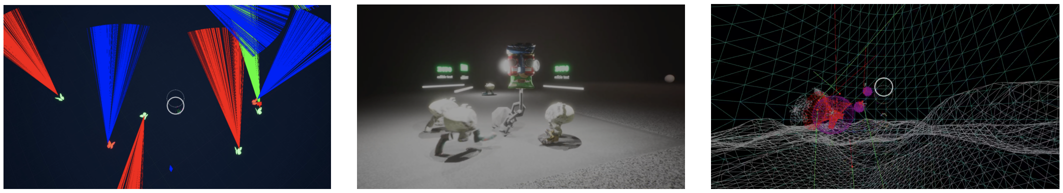
(b) Adapting spike activity to play an FPS game.

Figure 4: **3D navigation and environmental feedback.** Examples of neural control in simulated 3D environments. **See Sec.3.4 for details.

Inter-Agent, Intra-Agent, and Multi-Agent Simulations. As speculated by Leung et al.[15], multiple neuronal cultures may be used in a multi-agent environment. Figure 5a builds on the FPS game to reward each team based on successful hits and utilize spiking activities to control movement. This example could be a useful benchmarking task to compare cultures with different physical assemblies, MEAs, or chemical properties. Figure 5b proposes a coordination-based game that may be used in a single neuronal culture or multiple cultures interfaced together, where different agents receive partial sensory access that could potentially develop specific abilities when reward is only given to coordinated success. Figure 5c tests for goal-directed flocking where channels are mapped as individual agents, spikes translated into movement vectors. To close the loop, stimulation frequency will be matched to swarm density. Altogether, these templates may be helpful to benchmark for emergent behavioral traits such as adversarial learning, self-organization, or division-of-labour.

4 Further Development

To support interdisciplinary research and creative outputs, we plan to develop additional visualization tools and facilitate easier integration with other creative technology software such as TouchDesigner, Ableton, and other external applications. For example, this could include visualizing latent dynamics real-time [14], or to help benchmark and platform interactivity for a wider range of open-source neural assemblies [9].



(a) Team battle (b) Role differentiation (c) Goal-directed swarming

Figure 5: **Multi-agent simulation.** Competitive team battle, role specification, and self-organization as example contexts. **See Sec.3.4 for details.

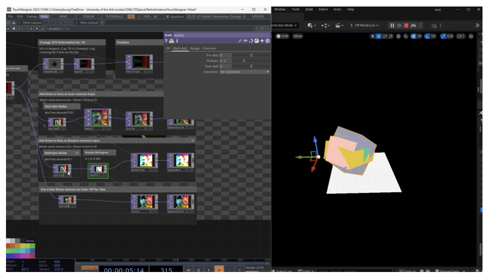


Figure 6: **Data streaming to external applications.** The proposed toolkit will include data streaming or data export such as OSC or texture-streaming via Spout to other applications.

5 Limitations

Since biocomputing systems face inherent biological variability [16], their performance may be inconsistent due to differences in cellular viability, maturation states, and hardware conditions [12]. Reproducibility depends critically on standardized biocomputing setups (perfusion, electrode arrays, etc.), and visible results may require extended training periods compared to artificial systems. Current brain organoid systems also face significant scaling constraints, particularly around vascularization, which limits their viable size, longevity, and functional complexity [26]. These biological limits mean that present-day organoids are thought to fall short of the neural architectures required for capacities for complex games and consciousness [8]. While this reduces the likelihood of immediate concerns, it remains essential to ensure that any future increases in neural complexity are matched with rigorous, standardized assessments for detecting relevant capacities, so that technological scaling proceeds ethically.

A persistent challenge across biocomputing systems lies in temporal alignment— how to scale and synchronize stimulation. Neurons operate on millisecond timescales, and if feedback is delivered too late or asynchronously, the biological substrate cannot meaningfully associate the stimulus with its preceding activity [12]. In reinforcement learning terms, this is analogous to issuing a reward or punishment long after the behavior occurred: the signal no longer encodes a coherent consequence. Designing closed-loop protocols that operate within the biological window of perception and plasticity thus remains a critical technical and conceptual task. The trade-off is further complicated by cellular fragility. Overstimulation affects substrate longevity, narrowing the range of sustainable experiments within its lifetime. As improvements in electrode density, adaptive filtering, and computational interpolation advance, more stable, fine-grained interaction with organoid substrates will become possible.

Finally, simplified performance metrics can drive "value capture," where narrow reward functions eclipse broader research aims [18]. This could bias both the biological system's adaptation and the interpretation of its behavior. Mitigation strategies (reward-schema versioning, periodic metric rotation, and transparent audit logs) should remain integral to experimental design[18].

6 Conclusion

Assembloid Agency provides plug-and-play tools for researchers to integrate living neuronal signals into Unreal Engine environments, facilitating both open and closed-loop experimentation for scalable bioengineered intelligence benchmarking. The open-source plugin and templates, which include reinforcement learning task-based benchmarks (e.g. vs. DishBrain [12]) and future creative applications such as world navigation, team battles, sound synthesis, procedural environment design, LLM coupling and data streaming, enable direct comparisons between biological and artificial agents. While this framework bridges game engines, web protocols, and biocomputing for cross-disciplinary use, its efficacy depends on addressing biological variability: organoid performance fluctuates due to culture conditions, hardware fidelity, and extended training requirements as poly-computational agents. We mitigate this through adaptive game design, standardized documentation, and synthetic controls (NEST simulations), ensuring robustness and playfulness at the core of the toolkit.

References

[1] Leyla Amirifar, Amir Shamloo, Rohollah Nasiri, Natan Roberto de Barros, Ze Zhong Wang, Bige Deniz Unluturk, Alberto Libanori, Oleksandr Ievglevskyi, Sibel Emir Diltemiz, Samuel Sances, Ilangko Balasingham, Stephanie K Seidlits, and Nureddin Ashammakhi. Brain-on-a-chip: Recent advances in design and techniques for microfluidic models of the brain in health and disease. *Biomaterials*, 285:121531, June 2022.

[2] Joshua Bongard and Michael Levin. Living things are not (20th century) machines: Updating mechanism metaphors in light of the modern science of machine behavior. *Front. Ecol. Evol.*, 9, March 2021.

[3] Joshua Bongard and Michael Levin. There's plenty of room right here: Biological systems as evolved, overloaded, multi-scale machines. *Biomimetics (Basel)*, 8(1):110, March 2023.

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI gym. *arXiv [cs.LG]*, June 2016.

[5] FinalSpark. Neuroplatform - finalspark. `https://finalspark.com/neuroplatform/`, April 2024. Accessed: 2024-9-16.

[6] Steffen Graber, Jessica Mitchell, Anno Christopher Kurth, Dennis Terhorst, Jan-Eirik Welle Skaar, Catherine Mia Schöfmann, Susanne Kunkel, Guido Trensch, Nicolai Haug, Daniel Mallett, Pishyi Yehor Andriyovich, Xavier Otazu Porter, Anthony Y Lee, and Hans Ekkehard Plesser. NEST 3.8, 2024.

[7] Edwin Hutchins. *Cognition in the Wild*. MIT Press, August 1996.

[8] Jacob Jeziorski, Reuven Brandt, John H. Evans, Wendy Campana, Michael Kalichman, Evan Thompson, Lawrence Goldstein, Christof Koch, and Alysson R. Muotri. Brain organoids, consciousness, ethics and moral status. *Seminars in Cell & Developmental Biology*, 144, Mar 2022. doi: https://doi.org/10.1016/j.semcdb.2022.03.020.

[9] 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*, Volume 7 - 2024, 2024. ISSN 2624-8212. doi: 10.3389/frai.2024.1376042. URL `https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1376042`.

[10] Brett J. Kagan. Two roads diverged: Pathways toward harnessing intelligence in neural cell cultures. *Cell Biomaterials*, 1(8):100156, 2025. ISSN 3050-5623. doi: 10.1016/j.celbio.2025.100156. URL `https://www.sciencedirect.com/science/article/pii/S3050562325001473`.

[11] Brett J Kagan. The CL1 as a platform technology to leverage biological neural system functions. *Nat. Rev. Bioeng.*, July 2025.

[12] 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*, 110(23):3952–3969.e8, 2022. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2022.09.001. URL `https://www.sciencedirect.com/science/article/pii/S0896627322008066`.

[13] Cortical Labs. Cl-api documentation (early release). `https://github.com/Cortical-Labs/cl-api-doc`, 2024. Accessed: 2025-08-09.

[14] Rodrigo González Laiz, Tobias Schmidt, and Steffen Schneider. Self-supervised contrastive learning performs non-linear system identification, 2025. URL `https://arxiv.org/abs/2410.14673`.

[15] Jenn Leung, Chloe Loewith, and Ivar Frisch. Organoid array computing: The design space of organoid intelligence. *Antikythera Digital Journal*, May 2025.

[16] Xiaofeng Liu, Zhiyuan Zhou, Yao Zhang, Hao Zhong, Xiulei Cai, and Rongfa Guan. Recent progress on the organoids: Techniques, advantages and applications. *Biomedicine & Pharmacotherapy*, 185:117942, Apr 2025. doi: https://doi.org/10.1016/j.biopha.2025.117942. URL `https://www.sciencedirect.com/science/article/pii/S0753332225001362`.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. *Playing Atari with Deep Reinforcement Learning*. DeepMind, Dec 2013. URL `https://arxiv.org/pdf/1312.5602`.

[18] Christopher Nguyen. Value capture. *J. Ethics Soc. Philos.*, 27(3), May 2024.

[19] Banjo Obayomi. Aws builder center, 2025. URL `https://builder.aws.com/content/2dbNlQiqKvUtTBV15mHBqivckmo/14-llms-fought-314-street-fighter-matches-heres-who-won`.

[20] OpenAI. Openai five, 2018. URL `https://openai.com/index/openai-five/`.

[21] Luciana Parisi. *Contagious Architecture: Computation, Aesthetics, and Space*. MIT Press, March 2013.

[22] Steve M. Potter, Ahmed El Hady, and Eberhard E. Fetz. Closed-loop neuroscience and neuroengineering. *Frontiers in Neural Circuits*, Volume 8 - 2014, 2014. ISSN 1662-5110. doi: 10.3389/fncir.2014.00115. URL `https://www.frontiersin.org/journals/neural-circuits/articles/10.3389/fncir.2014.00115`.

[23] David Silver, Thomas Hubert, Julian Schrittwieser, and Demis Hassabis. Alphazero: Shedding new light on chess, shogi, and go, Dec 2018. URL `https://deepmind.google/discover/blog/alphazero-shedding-new-light-on-chess-shogi-and-go/`.

[24] 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. *Front. Sci. Ser.*, 1, February 2023.

[25] G. Tesauro. Temporal difference learning and td-gammon. *ICGA Journal*, 18(2):88–88, Jun 1995. doi: https://doi.org/10.3233/icg-1995-18207. URL `https://cling.csd.uwo.ca/cs346a/extra/tdgammon.pdf`.

[26] Xingli Zhao, Zilu Xu, Lang Xiao, Tuo Shi, Haoran Xiao, Yeqin Wang, Yanzhao Li, Fangchao Xue, and Wen Zeng. Review on the vascularization of organoids and organoids-on-a-chip. *Frontiers in Bioengineering and Biotechnology*, 9, Apr 2021. doi: https://doi.org/10.3389/fbioe.2021.637048.