AGITB: A Signal-Level Benchmark for Evaluating Artificial General Intelligence

Matej Šprogar

University of Maribor, Faculty of Electrical Engineering and Computer Science, Koroška cesta 46, SI–2000 Maribor, Slovenia

Abstract

Current artificial intelligence systems exhibit strong performance on narrow tasks, while existing evaluation frameworks provide limited insight into generality across domains. We introduce the Artificial General Intelligence Testbed (AGITB), a complementary benchmarking framework grounded in fourteen explicitly stated axioms and implemented as a suite of fourteen automated, simple, and reusable tests.

AGITB evaluates models on their ability to learn and to predict the next input in a temporal sequence whose semantic content is initially unknown to the model. The framework targets core computational properties, such as determinism, adaptability, and generalisation, that parallel principles observed in biological information processing. Designed to resist brute-force or memorisation-based strategies, AGITB requires autonomous learning across previously unseen environments, in a manner broadly inspired by cortical computation. Preliminary application of AGITB suggests that no contemporary system evaluated to date satisfies all test criteria, indicating that the benchmark provides a structured and interpretable means of assessing progress toward more general learning capabilities. A reference implementation of AGITB is freely available on GitHub.

Keywords: artificial general intelligence, benchmarking, learning, generalisation, symbol grounding problem, neural networks, temporal sequence prediction

1. Introduction

Despite major advances in machine learning and neural network architectures, artificial intelligence (AI) systems still fall short of the flexibility and robustness characteristic of human cognition. Their surface-level competence often conceals a deeper absence of understanding, a capacity that remains essential for any credible claim to artificial general intelligence (AGI) [1, 2].

Yet these limitations have not prevented growing speculations that AGI may be close, claims that are difficult to substantiate in the absence of a rigorous and informative benchmark. Meaningful assessment of progress toward AGI requires more than specialised metrics or qualitative impressions; it calls for a principled, general-purpose benchmark capable of systematically capturing and comparing the cognitive capabilities relevant to general intelligence.

Numerous attempts have been made to define benchmarks for evaluating general intelligence in machines, the most influential being the Turing Test [3]. However, none has achieved its intended purpose, and all lack key properties such as gradual resolution, interpretability, and full automation. Existing benchmarks typically assess only superficial task performance rather than the underlying mechanisms of generalisation. In response to these limitations, this paper introduces the Artificial General Intelligence Testbed (AGITB), a novel benchmark designed to evaluate foundational cognitive abilities in artificial intelligence systems.

Email address: matej.sprogar@um.si (Matej Šprogar)

Although AGITB is proposed as a general testbed and benchmark for artificial general intelligence, it is grounded in empirical knowledge of neural function in the human brain. This reflects the fact that neuron-based systems are the only systems currently known to support a broad range of cognitive abilities. At present, there is no evidence that fundamentally different computational paradigms can produce behaviour comparable to that of humans. Pursuing fully abstract or architecture-agnostic criteria for general intelligence may therefore be premature until the principles underlying natural, neuron-based intelligence are better understood.

Although AGITB does not aim to evaluate consciousness or semantic comprehension, it provides a principled framework for distinguishing narrow AI systems from those exhibiting generalisable, adaptive behaviour. To position AGITB within the broader landscape of AGI evaluation, we include a comparative analysis with the Abstraction and Reasoning Corpus (ARC) [4] and the NeuroBench framework [5].

2. Background

The rapid progress of deep learning has enabled AI systems to exhibit increasingly sophisticated reasoning, problem-solving, and dialogue capabilities. However, despite these advances, a persistent reluctance remains to attribute "intelligence" to machines. This hesitation is partly rooted in the intuitive association of intelligence with uniquely human traits, such as consciousness, self-awareness, and subjective experience, which remain elusive in artificial systems.

Historically, as AI systems have succeeded in domains once considered hallmarks of human intelligence, the definition of intelligence has undergone significant shifts.[1] As AI approaches human-level capabilities, we may inadvertently set a perpetually receding goalpost for AGI, failing to recognise it even when achieved.

Although AGI is typically envisioned as matching human cognitive flexibility across diverse domains, its evaluation has largely defaulted to narrow, task-specific metrics. This is partly due to the absence of a universally accepted AGI benchmark. Researchers have thus gravitated toward achieving superhuman performance in discrete domains, where progress can be clearly quantified. However, such specialised benchmarks favour narrow AI by rewarding depth within isolated subdomains rather than breadth of adaptation and general reasoning — hallmarks of general intelligence. Ironically, some of these benchmarks are now so specialised that humans have difficulty with them.

2.1. A benchmark that only humans and AGI can meet

An effective AGI test must be trivial to solve for humans yet remain inaccessible to contemporary machine learning models that rely on brute-force methods, pretraining, or statistical pattern matching. Such a test must demand capabilities that transcend memorisation or domain-specific heuristics, requiring generalisation, abstraction, and adaptive reasoning.

A valid evaluation of AGI must target behavioural capacities that current artificial systems do not yet robustly exhibit. Meaningful evaluation, therefore, must focus on capacities that cannot be obtained through data or computational scale alone, but instead require systems to acquire structure through interaction and adaptation. One possible direction involves closer alignment with the computational principles of the human cortex, motivating interest in neuromorphic approaches such as spiking neural networks, which explicitly incorporate time-sensitive, event-driven dynamics [6]. While such architectures do not, in themselves, constitute general intelligence, they illustrate alternative design dimensions that remain underexplored in conventional artificial neural networks.

In alignment with this biologically grounded perspective, AGITB departs from symbolic, high-level evaluations and instead assesses intelligence at the lowest, signal-processing level. While Turing was right to suggest that communication could serve as a basis for evaluating machine intelligence, natural language remains problematic as a test medium: it conveys human knowledge through symbols whose meanings are not intrinsically grounded in machines, as argued by Harnad [7]. Although the symbol-grounding problem

[1] For instance, the success of Deep Blue against Garry Kasparov in chess (a task formerly seen as a benchmark for AGI) was quickly reframed as a triumph of brute-force computation rather than genuine intelligence.

is an old philosophical issue, it has regained prominence in contemporary research across cognitive science, neuroscience, and machine learning [e.g. 8, 9, 10].

A more elemental approach is therefore adopted. Rather than judging intelligence by symbolic interpretation, it evaluates whether a system can detect, learn, and generalise patterns in raw binary signals. A neural spike by itself contains the smallest amount of information possible and is, as such, grounded but free of other semantics. A binary signal accurately represents the neural spike.

Building on the view that intelligence is fundamentally about extracting structure from data to enable prediction [11], AGITB operates at the level of signal-based prediction. This form of low-level prediction constitutes the structural basis from which semantically grounded, high-level anticipations about the external world can emerge, without presupposing semantic understanding itself. This approach aligns closely with the functioning of biological intelligence at the cortical level, which processes time-sensitive sensory spike trains rather than disembodied symbols.

3. Artificial general intelligence testbed

The testbed supports the development and evaluation of more general learning systems by defining a clear set of requirements expressed as axioms that a model under evaluation must meet. A model must satisfy all requirements in order to claim success on the benchmark.

The guiding premise of AGITB is not that it provides a definitive or exclusive criterion for artificial general intelligence, but that it captures a set of capabilities that appear necessary for moving beyond narrow, task-specific behaviour. Although this premise cannot be proven in the absence of a precise definition of intelligence, it could be challenged by the existence of a narrow system that satisfies all AGITB criteria. To date, however, no artificial system has done so, whereas the biological brains meet the benchmark's requirements. This suggests that AGITB identifies competencies that current AI systems lack, and that satisfying these requirements may be indicative of progress toward more general forms of intelligence.

AGITB is not intended as a sufficient criterion for artificial intelligence in any broad sense. A system that satisfies all requirements does not thereby qualify as an AI system, let alone an AGI, since the benchmark does not assess higher-level capacities such as reasoning, abstraction, or natural language competence. Rather, it targets a set of low-level capabilities that may serve as precursors to, but do not themselves guarantee, more general forms of intelligence.

3.1. Metric-free design

A distinguishing architectural feature of AGITB is its deliberate avoidance of conventional correctness metrics such as accuracy or mean-squared error. The limitations of these metrics are twofold. First, they cannot reliably distinguish between AGI and human performance, or that of non-AGI systems, as contemporary AI models can already surpass humans on standard benchmarks. Second, in an era of elevated expectations driven by specialised generative AI systems, the anticipated AGI performance on such metrics is often set unrealistically high. Even well-educated humans possessing fully developed and highly parallelised brains may underperform relative to current AI systems. It is therefore unreasonable to expect a first-generation AGI, potentially operating on a simplified and computationally constrained simulation of the brain, to match or exceed human-level results. Currently, our understanding of neural mechanisms is insufficient to justify such expectations.

Without conventional metrics or predefined performance thresholds, it is challenging to determine a meaningful level of competence. AGITB addresses this problem by employing a self-referential evaluation approach, in which the model under test is compared against itself. Each test constructs a controlled scenario involving one or more independent instances of the model, whose behaviours are analysed comparatively. Success is thus defined in terms of the relative consistency or superiority of model responses, rather than by any external quantitative metric. Requiring each test to be passed 5,000 times renders AGITB an extreme form of stress testing, ensuring that successful performance reflects genuine robustness rather than chance.

For these reasons, and to minimise type I errors, AGITB employs an all-or-nothing criterion: the system under evaluation must successfully pass *all* tests. This design choice is justified, as individual tests are

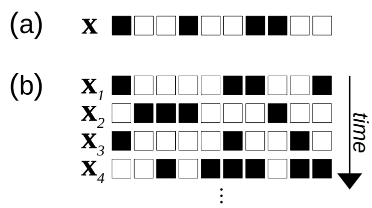

Figure 1: (a) Example of a 10-bit input $\mathbf{x}$ with four bits set. (b) Example of an input sequence.

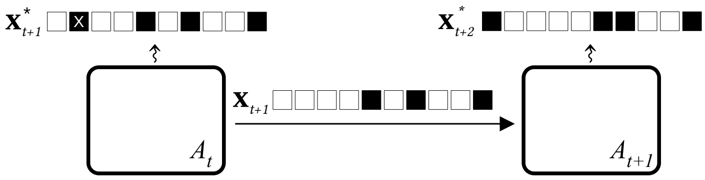

Figure 2: Iterative adaptation in discrete time. At time step t, the model A_t issued the prediction $\mathbf{x}^*_{t+1}$. After observing the realised input $\mathbf{x}_{t+1}$, it adapted itself in response to the error in the second bit and subsequently produced the next one-step-ahead prediction $\mathbf{x}^*_{t+2}$.

solvable by non-AGI systems, whereas the simultaneous satisfaction of all fourteen requirements is believed to demand capabilities beyond narrow intelligence. AGITB therefore posits that what non-AGI systems lack is the property referred to as "intelligence," understood as the synergistic integration of all axioms taken together.

3.2. Architecture

AGITB draws inspiration from the *ladder to human-comparable intelligence* [12], but departs from its ladder-like, hierarchical structure. Rather than defining a sequential progression of increasingly demanding cognitive abilities, AGITB integrates these underlying ideas into a single testbed grounded in fourteen requirements, all of which are verified by fully automated and transparent tests.

The benchmark treats an AGI model under evaluation as a black box that predicts the next input from the historical sequence of observed signals. Each input consists of ten bits, the specific semantics of which are immaterial; each bit may represent an arbitrary channel, such as a pixel, an audio band, or actuator feedback.

Each input represents a snapshot of multiple parallel signals at a single time step (Figure 1a). Spatial organisation within each input encodes local structure, whereas semantic richness arises from the temporal evolution of the input sequence (Figure 1b). The interaction between spatial and temporal dimensions gives rise to structured patterns that are challenging for the model to adapt to.

The testbed presents the AGI model with a stream of inputs over time. At each time step t, the model is required to issue a prediction for the subsequent input x_{t+1}, as shown in Figure 2. A transition from model A_t to A_{t+1} is triggered by the arrival of the actual input (x_{t+1}). The central challenge is not simply extrapolation but discerning the underlying causes or regularities that produce the observed input stream and using that understanding to make accurate future predictions.

The testbed presents the AGI model with a stream of inputs over time. At each discrete time step t, the model is required to issue a prediction of the subsequent input $\mathbf{x}^*_{t+1}$, as illustrated in Figure 2. Upon receipt of the actual input $\mathbf{x}_{t+1}$, the model transitions from configuration A_t to A_{t+1}. The central challenge is not mere extrapolation, but the identification of the underlying regularities or generative causes that give rise to the observed input stream, and the use of this inferred structure to produce accurate future predictions.

AGITB makes no structural assumptions about the evaluated model. In particular, the form of its internal state, representational content, and update mechanisms are left unspecified. Imposing any formal assumption, such as requiring an explicit state representation or prescribed update rule, would risk privileging particular model classes and would be inconsistent with AGITB's objective of remaining as architecture-agnostic as possible.

Verification under AGITB involves comparing independently instantiated copies of a model under controlled experimental conditions. While model instances are always distinct as entities, they may realise equal or unequal configurations within the configuration space of a given AGI model type. As models are treated as black boxes, AGITB does not attempt to determine configuration equality through behavioural comparison over unbounded horizons, which is generally infeasible. Instead, AGITB assumes that, for each model type, the model author supplies a well-defined mechanism for determining whether two instances realise the same configuration or different configurations.

3.3. Terminology

Let $\mathbf{x} \in X = \{0,1\}^L$ with $L = 10$ denote a ten-bit input vector, and let $\boldsymbol{\varphi} = (\mathbf{x}_t)_{t=1}^N, \mathbf{x}_t \in X$ denote an input sequence with N elements. For any input vector $\mathbf{x} \in X$, we write $\mathbf{x}[i]$ for its i-th bit.

Let $\mathcal{M}$ denote the set of all reachable[2] configurations of a model type under AGITB evaluation. Individual configuration is denoted by $A \in \mathcal{M}$, with $B \in \mathcal{M}$ representing an independent instance of the same model type. When temporal indexing is required, we write A_t for the configuration obtained after t update steps along its learning trajectory, with A_0 representing the initial configuration at the onset of learning.

For brevity, we refer to a model configuration simply as a model.

Definition 1 (Prediction). A model's prediction is given by the operator $\rightsquigarrow \subseteq \mathcal{M} \times X$, where

$$A_t \rightsquigarrow \mathbf{x}^*_{t+1}$$

denotes that the model configuration $A_t \in \mathcal{M}$ predicts the next input to be $\mathbf{x}^*_{t+1} \in X$.

Definition 2 (Model update). Model update is given by the transition relation $\dot{\mapsto} \subseteq \mathcal{M} \times X \times \mathcal{M}$, where

$$A_t \xmapsto{\mathbf{x}_{t+1}} A_{t+1}$$

denotes the atomic transition of the model configuration $A_t \in \mathcal{M}$ upon receiving the input $\mathbf{x}_{t+1} \in X$.

In deterministic implementations, the update relation may be realized as a function. Operationally, prediction and update are tightly coupled: processing the input $\mathbf{x}_{t+1}$ both updates the model and produces a prediction of the subsequent input $\mathbf{x}_{t+2}$. This allows us to write the combined transition as

$$A_t(\mathbf{x}_{t+1}) \to \big(A_{t+1}, \mathbf{x}^*_{t+2}\big),$$

which subsumes both model advancement and prediction.

Definition 3. For any model configuration $A \in \mathcal{M}$ and any input sequence $\boldsymbol{\varphi} = (\mathbf{x}_i)_{i=1}^N$, we write

$$A \xmapsto{\boldsymbol{\varphi}} A^{\boldsymbol{\varphi}}$$

to denote the transitions to the configuration obtained by sequentially updating the model with all inputs in $\boldsymbol{\varphi}$, i.e.,

$$A^{\boldsymbol{\varphi}} := (\cdots((A \xmapsto{\mathbf{x}_1}) \xmapsto{\mathbf{x}_2}) \cdots) \xmapsto{\mathbf{x}_N}.$$

Accordingly, $A_t^{\boldsymbol{\varphi}}$ denotes the configuration obtained by applying $\boldsymbol{\varphi}$ to A_t, that is, $A_t^{\boldsymbol{\varphi}} := (A_t)^{\boldsymbol{\varphi}}$. A model may be exposed to multiple input sequences in succession; for two sequences $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$, their concatenation is written $\boldsymbol{\varphi}_1\boldsymbol{\varphi}_2$, and the resulting configuration satisfies

$$A \xmapsto{\boldsymbol{\varphi}_1} A^{\boldsymbol{\varphi}_1} \xmapsto{\boldsymbol{\varphi}_2} A^{\boldsymbol{\varphi}_1\boldsymbol{\varphi}_2}.$$

[2]Starting from the initial configuration under some input history.

Definition 4. Autoregressive generation is given by the relation $\Rrightarrow \subseteq \mathcal{M} \times X^{\leq \mathbb{N}}$, where

$$A_t \Rrightarrow \boldsymbol{\varphi}^*$$

denotes that the model configuration A_t generates the (finite or infinite) sequence $\boldsymbol{\varphi}^* = (\mathbf{x}^*_{t+1}, \mathbf{x}^*_{t+2}, \ldots)$ by recursively feeding each predicted input back into the model. In particular, for all $k \geq 0$,

$$A_{t+k} \rightsquigarrow \mathbf{x}^*_{t+k+1}, \quad A_{t+k} \xmapsto{\mathbf{x}^*_{t+k+1}} A_{t+k+1}.$$

Definition 5. The model A is said to *learn* $\boldsymbol{\varphi}$, written $A \rhd \boldsymbol{\varphi}$, if there exists a finite number of learning steps after which A can reproduce a single instance of $\boldsymbol{\varphi}$ through autoregressive generation, i.e.

$$A \rhd \boldsymbol{\varphi} \;:\Longleftrightarrow\; \exists\, n \in \mathbb{N}_{\geq 1} :\; A^{\boldsymbol{\varphi}^n} \Rrightarrow \boldsymbol{\varphi}.$$

Definition 6. The *learning time in atomic steps* is defined as

$$\tau_A(\boldsymbol{\varphi}) \;:=\; \begin{cases} |\boldsymbol{\varphi}| \cdot \min\left\{\, n \in \mathbb{N}_{\geq 1} \;\middle|\; A^{\boldsymbol{\varphi}^n} \Rrightarrow \boldsymbol{\varphi} \right\}, & \text{if such an } n \text{ exists}, \\ \infty, & \text{otherwise}. \end{cases}$$

Learning time measures the first occurrence of accurate prediction and does not presuppose permanent retention. If the model never attains accurate prediction of the sequence, the learning time is infinite.

Definition 7 (Match score). For two sequences $\boldsymbol{\alpha} = (\mathbf{a}_1, \ldots, \mathbf{a}_m)$ and $\boldsymbol{\beta} = (\mathbf{b}_1, \ldots, \mathbf{b}_m)$ in X^m, define the (unnormalised) bitwise match score

$$S(\boldsymbol{\alpha}, \boldsymbol{\beta}) \;:=\; \sum_{j=1}^{m} \sum_{i=1}^{L} \mathbf{1}\{\mathbf{a}_j[i] = \mathbf{b}_j[i]\}\,.$$

3.4. Axioms for universal learning

Empirically verifying whether a system under evaluation satisfies all fourteen axioms presents two challenges. First, some axioms would require testing over an effectively unbounded set of cases. Second, others demand complex forms of evaluation that cannot be captured by a single, easily interpretable test. AGITB addresses the former challenge by pragmatically replacing an effectively unbounded test case set with a finite, tractable collection of test cases. The latter is addressed through a synergistic test design in which individual tests support and reinforce one another. When a requirement expressed by an axiom cannot be confirmed by a single test in isolation, consistent behavioural patterns observed across multiple tests provide indirect validation.

Consequently, the AGITB testbed cannot, and is not intended to, prove that a model under evaluation satisfies all fourteen axioms. Rather, its strength lies in identifying models that violate one or more axioms while also indicating the specific axiomatic constraints that are breached. A reference C++ implementation of the AGITB is freely available under the GPL-3 license at `https://github.com/matejsprogar/agitb`.

Axiom 1 (Requirement 1: Uninformed start). *All instances of a given model type begin transitioning from an identical initial configuration* $\diamond$*:*

$$\forall\, A :\; A_0 = \diamond.$$

The initial model is assumed to contain no environment-specific knowledge beyond the architectural biases necessary for learning. All input-relevant knowledge is acquired solely through subsequent exposure to data. This assumption is commonly regarded as a necessary condition for universality, defined as the ability to adapt to arbitrary environments.

By definition, each model must provide a prediction of the subsequent input. A neutral model, being completely uninformed about its surroundings, cannot make any meaningful prediction thus the prediction $\mathbf{x}^*_1$ is undefined and can be anything.

AGITB assumes that general-purpose learning systems, including biological brains, do not begin with an innate understanding of external inputs but instead acquire meaning through interaction with their environment. Each system must construct semantic content from raw sensory data rather than rely on pre-encoded knowledge.[3]

Although AGITB presumes identical initialization across model instances, no externally administered test can conclusively verify this requirement. Consequently, empirical assessments that all models start equal must be complemented by additional evaluations in which models are required to acquire semantic structure across multiple independent environments, none of which admit fixed task-relevant prior knowledge that would confer a universal advantage.

Axiom 2 (Perpetual change). *Every input modifies the model configuration.*

$$\forall\, t \geq 0 :\; A_{t+1} \neq A_t$$

Every input modifies the model, actively shaping its internal dynamics. The act of input processing itself introduces experience-dependent information and induces a continual change in the model's internal configuration.

Although this property is implied by Axiom 4, AGITB retains it as a separate requirement to explicitly state a fundamental aspect of the learning process.

Axiom 3 (Determinism). *Model evolution is deterministic with respect to input.*

$$\forall\, A \in \mathcal{M},\; \forall\, \mathbf{x} \in X :\; \exists!\, A' \in \mathcal{M} \text{ such that } A \overset{\mathbf{x}}{\mapsto} A'.$$

Biological neurons operate in a functionally deterministic manner, ensuring stability and consistency in brain function. Although minor stochastic effects may occur, they do not undermine the rule-governed nature of neural processing. By analogy, AGITB assumes that input history uniquely determines the model configuration. A model is therefore fully determined by its input history.

Determinism at the level of neural signal processing is necessary for stable, reproducible brain function, whereas the apparent unpredictability of cognition stems from the system's complexity rather than from genuine indeterminacy [13].

Axiom 4 (Trace). *Each input leaves a permanent internal trace.*

$$\forall t \neq s \geq 0 :\; A_t \neq A_s.$$

A model's internal configurations evolve without recurrence: configurations never repeat, trajectories contain no cycles, and every input leaves a permanent internal trace.

Human brains satisfy an effective version of this axiom at the lowest physical level. Each sensory or internal event induces irreversible microstate changes that are never exactly revisited. Through complex internal interactions, this persistent history contributes to the apparent unpredictability of decision-making. Although information may appear discarded, compressed, or behaviorally inaccessible at the cognitive level, AGITB posits that internal model configurations continue to progress in a distinct manner over time, consistent with permanent internal trace formation.

Axiom 5 (Time). *Model evolution depends on input order.*

$$\forall\, A, \forall\, \boldsymbol{\varphi}_1 \neq \boldsymbol{\varphi}_2 \in X^+ \;:\; A^{\boldsymbol{\varphi}_1 \boldsymbol{\varphi}_2} \neq A^{\boldsymbol{\varphi}_2 \boldsymbol{\varphi}_1}.$$

Model evolution depends intrinsically on input order: for any two distinct input sequences, exchanging their order necessarily results in a different internal configuration. This enforces a strict temporal asymmetry in learning dynamics and rules out commutative or order-invariant update mechanisms. Sensitivity to temporal structure in this strong sense is regarded as a defining property of intelligent systems.

[3] Although certain reflexes may be genetically specified, they do not constitute genuine understanding. Such reflexes are evolutionary features of the subcortical "old brain" and not prerequisites for intelligence [11, p. 66].

Axiom 6 (Absolute refractory period). *A model can learn a cyclic sequence only if the sequence satisfies the absolute refractory-period constraint.*

$$\forall \boldsymbol{\varphi} :\ A_0 \rhd \boldsymbol{\varphi} \ \Rightarrow\ \boldsymbol{\varphi} \in \boldsymbol{\Lambda}.$$

The set of *admissible* sequences is defined as

$$\boldsymbol{\Lambda} := \Big\{ (\mathbf{x}_1, \dots, \mathbf{x}_k) \in X^k \ \Big|\ 1 \le k \le k_{\max} \ \wedge\ \forall i \in \{1, \dots, k\} :\ \langle \mathbf{x}_i, \mathbf{x}_{((i \bmod k)+1)} \rangle = 0 \Big\}.$$

Biological intelligence relies on discrete spiking events for communication and learning, and individual neurons cannot fire again immediately after activation. AGITB incorporates an absolute refractory-period constraint to reflect this property, without assuming that all sequences admissible under this constraint are necessarily learnable.

Although absolute refractory periods are not themselves the source of spiking variability, they impose a minimum separation between spikes and thereby preserve temporal structure. Learning mechanisms based on spike timing cannot operate effectively when such a structure is absent [14]. Consequently, AGITB admits all temporal sequences consistent with biologically plausible refractory dynamics, while remaining agnostic to any particular semantic encoding of signals.

This axiom requires cyclic sequences, the simplest learning setting due to repeated input exposure; learning from non-repetitive input streams is addressed by Axiom 12.

Axiom 7 (Limited learnability).
(a) No model can learn everything there is to learn.

$$\exists\, \boldsymbol{\phi} \in \boldsymbol{\Lambda} \ :\ \neg\big(A_0 \rhd \boldsymbol{\phi}\big).$$

(b) All admissible length-2 sequences are universally learnable.

$$\forall A,\ \forall \boldsymbol{\phi} \in \boldsymbol{\Lambda} \cap X^2 \ :\ A \rhd \boldsymbol{\phi}.$$

Learning systems with finite representational and adaptive capacity exhibit inherent limits on the sets of sequences they can learn. In particular, no model can be trained so as to learn all admissible sequences. However, a nontrivial lower bound is preserved: every model can learn any admissible sequence of length two.

Definition 8. The set of *learnable* sequences is the subset

$$\boldsymbol{\Phi} := \big\{ \boldsymbol{\phi} \in \boldsymbol{\Lambda} \ \big|\ A_0 \rhd \boldsymbol{\phi} \big\}.$$

Axiom 8 (Temporal adaptability). *The model must be able to learn sequences with varying cycle lengths.*

$$\exists\, \boldsymbol{\phi}_1,\, \boldsymbol{\phi}_2 \in \boldsymbol{\Phi} \ :\ 0 < |\boldsymbol{\phi}_1| < |\boldsymbol{\phi}_2| \ \wedge\ A_0^{\boldsymbol{\phi}_1} \rhd \boldsymbol{\phi}_2.$$

This axiom requires the model to learn and track temporal structure across multiple timescales. Unlike rigid pattern-matching systems, an intelligent model should detect and predict a recurring structure across different periodicities. Systems that can accommodate only a single, predetermined temporal scale, therefore, fail this requirement.

Axiom 9 (Content sensitivity). *Adaptation time is input-dependent.*

$$\exists\, \boldsymbol{\phi}_1,\, \boldsymbol{\phi}_2 \in \boldsymbol{\Phi} :\ |\boldsymbol{\phi}_1| = |\boldsymbol{\phi}_2| \ \wedge\ \tau_{A_0}(\boldsymbol{\phi}_1) \ne \tau_{A_0}(\boldsymbol{\phi}_2).$$

The structural complexity of an input sequence affects the rate at which a model adapts, where adaptation time is defined as the number of iterations required for the model to accurately predict the entire temporal pattern. Simple or highly regular sequences typically lead to rapid convergence, whereas less regular inputs demand longer exposure before the model can reliably capture and reproduce the underlying pattern.

Axiom 10 (Context sensitivity). *Adaptation time is model-dependent.*

$$\exists\,\boldsymbol{\phi} \in \boldsymbol{\Phi},\ \exists\, A \neq B: \quad \tau_A(\boldsymbol{\phi}) \ \neq\ \tau_B(\boldsymbol{\phi}).$$

The model reflects the cumulative influence of past inputs and therefore provides the context in which new information is processed. When subsequent inputs are consistent with the structure already established through prior learning, adaptation may proceed quickly. Conversely, when new inputs conflict with this learned context, the model may require additional time to reorganise itself before accurate prediction becomes possible.

Corollary 1 (Unobservability). *Distinct model configurations may be observationally indistinguishable under autoregressive generation: there exist* $\boldsymbol{\varphi} \in \boldsymbol{\Lambda}$ *and* $A \neq B$ *such that*

$$A \Rrightarrow \boldsymbol{\varphi} \ \wedge\ B \Rrightarrow \boldsymbol{\varphi}.$$

Proof. By Axiom 7(b), every admissible sequence of length two is learnable from any model configuration. Fix any $\boldsymbol{\varphi} \in \boldsymbol{\Lambda}$ with $|\boldsymbol{\varphi}| = 2$. Let A and B be two distinct model configurations. By the definition of learning, there exist finite numbers of learning steps after which both configurations can autoregressively generate $\boldsymbol{\varphi}$. Hence $A \Rrightarrow \boldsymbol{\varphi}$ and $B \Rrightarrow \boldsymbol{\varphi}$, establishing the claim.

Identical outputs may arise from distinct models. This many-to-one mapping highlights that observable behaviour alone need not disclose the underlying configuration or history of a model's internal dynamics.

Axiom 11 (Denoising). *An informed model outperforms the best constant baseline at denoising a corrupted input. Let* $\boldsymbol{\varphi} = (\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \boldsymbol{\Lambda}$ *be drawn from the underlying stochastic generative process, and let* $\boldsymbol{\varphi}' = (\mathbf{x}'_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$ *be obtained from* $\boldsymbol{\varphi}$ *by the corruption process. Fix* $n := 5|\boldsymbol{\varphi}|$*, and let* $\mathbf{x}_1^* \in X$ *satisfy*

$$A_0^{\boldsymbol{\varphi}^{\,n}\boldsymbol{\varphi}'} \rightsquigarrow \mathbf{x}_1^*.$$

Then the model's expected match score on the clean input exceeds that of both constant predictors:

$$\mathbb{E}\big[S((\mathbf{x}_1^*), (\mathbf{x}_1))\big] \ >\ \max\Big\{\mathbb{E}\big[S((\mathbf{0}), (\mathbf{x}_1))\big],\ \mathbb{E}\big[S((\mathbf{1}), (\mathbf{x}_1))\big]\Big\},$$

where $\mathbf{0}, \mathbf{1} \in X$ *denote the all-zero and all-one inputs, and the expectation is taken with respect to the sequence generator and the corruption process.*

An intelligent model should be able to recall a previously observed sequence, even when the inputs are perturbed by noise. When re-exposed to a very familiar ($n = 5|\boldsymbol{\varphi}|$) stimulus, such a model is expected, on average, to outperform any trivial baseline predictor in predicting a single incoming input.[4] Average performance over 20 runs is used as the evaluation criterion within each trial because random models may occasionally generate correct predictions by chance, without demonstrating genuine learning or structural understanding.

The test procedure is not interpreted as a conventional significance test (e.g., at the 5% level); rather, it functions as a stringent robustness check. In each trial, accuracy is aggregated over 20 independently sampled sequences, and the model must outperform the best constant baseline. This procedure is repeated for 5,000 independent trials, and the model is required to succeed in every case. The design ensures that only large, systematic performance gains yield a passing result. Modest or marginal improvements, while potentially real, are intentionally regarded as failures, as the objective is to identify only clear and substantial advances in model capability.

[4] For binary predictions, random guessing yields accuracy 0.5, which can be dominated by a constant predictor (always 0 or 1) when the bit distribution is biased, as under the absolute refractory period constraint.

Axiom 12 (Generalisation). *An informed model predicts previously unseen inputs better than chance. Let* $\boldsymbol{\varphi} = (\boldsymbol{\varphi}_1 \| \boldsymbol{\varphi}_2) \in \boldsymbol{\Lambda}$ *be a sequence generated by a randomly initialised generator model, whose internal rule is unknown to the model under evaluation and induces nontrivial temporal correlations. The prefix* $\boldsymbol{\varphi}_1$ *is observed during training, while* $\boldsymbol{\varphi}_2$ *is withheld and serves as the target for prediction. Let* $\boldsymbol{\varphi}_2^*$ *satisfy*

$$A_0^{\boldsymbol{\varphi}_1} \Rrightarrow \boldsymbol{\varphi}_2^*.$$

Then the model's expected match score on the unseen continuation exceeds chance:

$$\mathbb{E}\left[\frac{S(\boldsymbol{\varphi}_2^*, \boldsymbol{\varphi}_2)}{L\,|\boldsymbol{\varphi}_2|}\right] \;>\; \tfrac{1}{2},$$

where the expectation is taken with respect to the sequence-generation procedure conditioned on the observed prefix $\boldsymbol{\varphi}_1$.

Only models capable of generalisation can derive lasting benefits from experience. After exposure to an initial set of stimuli, such models are expected, on average, to outperform chance-level baselines when predicting previously unseen inputs. As in the preceding requirement, performance is aggregated over 20 runs per trial, reflecting the fact that random or memorisation-based models may occasionally produce correct predictions by chance without capturing the underlying structure.

The generalisation test follows the same conservative design principles as the test for Axiom 11. By requiring success across all 5,000 independent trials, the procedure enforces a stringent acceptance threshold, under which even a single underperforming trial results in failure. Consequently, the test admits virtually no noise or marginal effects. This design is not intended to maximise statistical power; rather, it prioritises robustness, ensuring that only models exhibiting a clear, systematic, and reproducible advantage obtain a passing result.

Axiom 13 (Real-time liveness). *Each model update completes within a uniform time bound.*

$$\exists\, t_{\max} > 0 \;\; \textit{such that} \;\; \forall\, A \in \mathcal{M},\; \forall\, \mathbf{x} \in X : \; \Delta t(A, \mathbf{x}) \le t_{\max},$$

where $\Delta t(A, \mathbf{x})$ *denotes the wall-clock time required to perform the atomic transition* $A \overset{\mathbf{x}}{\mapsto} A'$.

A model must complete each atomic transition within a bounded amount of time in order to remain suitable for real-time interaction. This axiom enforces *real-time liveness*: the time required to process an input and update the internal configuration must remain uniformly bounded, independent of the model's accumulated experience and the specific input being processed.

Biological brains satisfy this requirement through massive parallelism: state transitions and signal emissions occur concurrently across large populations of neurons, ensuring that neural signal propagation times remain bounded and stable under normal operating conditions. As a result, cognitive processing does not slow down as a function of accumulated experience or momentary sensory input.

Although the next configuration A_{t+1} depends on the current configuration A_t, the influence of a given input on observable output may be immediate or delayed, depending on the model architecture. For example, in a feed-forward artificial neural network, the update $A_t \xmapsto{\mathbf{x}_{t+1}} A_{t+1}$ constitutes the first atomic step in the inference process leading to the output $\mathbf{x}^*_{t+1+d}$, where d denotes the network depth[5]. In such architectures, d forward passes are required for the effect of the input $\mathbf{x}_{t+1}$ to reach the output; consequently, for all $i < d$, the intermediate outputs $\mathbf{x}^*_{t+1+i}$ are independent of $\mathbf{x}_{t+1}$. This architectural latency does not violate the axiom, provided that each atomic transition itself remains time-bounded.

To empirically assess this requirement, AGITB compares processing times between an empty and a fully trained (complex) instance of the same model under identical input conditions. If the bounded-time axiom holds, the complex model must not exhibit processing times that are consistently greater than those of the empty model. This condition is evaluated using a one-sided Wilcoxon signed-rank test with a conservative threshold of $z = 3.090$, corresponding to a one-sided significance level of 0.1%. This stringent criterion strongly limits false positives and ensures reliable detection of models whose update time grows with experience, memory size, or input complexity.

[5] Depth is defined as the number of hidden layers plus the output layer.

3.5. Search space

To prevent models from relying on brute-force memorisation, a robust benchmark must define a problem space large enough to exceed the capacity of any model operating under realistic computational constraints in both time and memory. In AGITB, tasks typically require predicting a temporal sequence of seven inputs ($|\boldsymbol{\varphi}| = 7$), each consisting of ten bits ($L = 10$). This corresponds to a combinatorial space of size $|S| = 2^{70}$, representing all possible binary input sequences of that length.

AGITB incorporates a biologically inspired *refractory period*, which prohibits any neuron (bit) from firing in consecutive time steps. This restriction substantially reduces the number of valid sequences. There are $|S'| = (F_{|\boldsymbol{\phi}|+2})^L = 34^{10} \approx 2^{51}$ distinct seven-step temporal sequences of ten bits under the condition that a 1 never carries over to the next time step, where F_i denotes the i-th Fibonacci number with $F_0 = 0$.

In some cases, AGITB further constrains the space by requiring the sequence to be cyclic, such that the first input also satisfies the refractory condition relative to the last input in the sequence. The number of distinct cyclic temporal sequences respecting the refractory constraint is $|S''| = (L_{|\boldsymbol{\phi}|})^L = 29^{10} \approx 2^{49}$, where L_i denotes the i-th Lucas number with $L_0 = 2$.

The choice of seven-step sequences with ten-bit inputs is sufficient to detect non-AGI behaviour while maintaining computational efficiency. Increasing these default values could exceed the capabilities of a first-generation AGI under evaluation, potentially producing false negatives and substantially increasing runtime. The current configuration, therefore, ensures that each test remains both computationally feasible and diagnostically informative.

Within the comparatively constrained AGITB environment, every randomly generated input sequence is, in principle, learnable through exposure. However, the sheer size of the search space makes any form of explicit teaching-to-the-test computationally infeasible. Given that real-world sensory inputs may ultimately encompass tens of thousands of bits, a genuine AGI system must employ generalisable, pattern-based learning mechanisms capable of extracting latent structure from high-dimensional data.

3.6. Interpreting performance under AGITB

Before assessing the usefulness of AGITB, it is important to clarify its role as a pragmatic benchmark rather than an end in itself. Like the Turing Test, which serves as an empirical criterion rather than a philosophical claim [15], AGITB is intended as a practical instrument for evaluating progress toward artificial general intelligence. The ultimate objective remains the development of AGI, not merely success on the benchmark.

AGITB yields meaningful insights only when developers adhere strictly to its core requirements. Misinterpretations of fundamental elements, such as the notion of an "empty" model, can lead to erroneous conclusions and impede genuine progress toward AGI.

Overall, AGITB provides a structured testbed for empirically evaluating foundational capabilities across diverse computational paradigms, including classical symbolic systems, artificial neural networks, and large language models. Before benchmarking artificial systems, however, it is necessary to establish a baseline by considering the performance of human cognition.

3.6.1. Human performance

The inability to directly compare internal cortical states makes it impossible to verify AGITB requirements in humans in a strict computational sense. Nevertheless, because cortical architecture inherently supports low-level binary signal processing and the tests align with basic cognitive competencies, it is reasonable to assume that humans naturally satisfy most requirements. Demand 1 (Uninformed start), however, warrants further discussion.

Owing to prior experience and cognitive bias, an adult human's cortex may appear to "fail" this prerequisite, as it is no longer in an unconditioned state and may generate non-empty predictions. AGITB, however, requires the uninformed model before the first input—a condition met only in the fetal cortex. At that developmental stage, the cortex lacks synaptic organisation and, prior to any sensory stimulation, satisfies the criterion of true neutrality.

The more complex AGITB tests have cognitive-level analogues that can be observed through reasoning and introspection. Temporal flexibility (Axiom 8), for example, poses no difficulty for humans, who readily recognise temporal patterns of varying durations. Limited learnability (Axiom 7) reflects the finite capacity of the human cortex to store and maintain knowledge; its behavioural analogue resembles the onset of cognitive saturation or early dementia, in which recent experiences are lost. Because humans acquire different types of information at varying rates, the motivation for content-sensitive (Axiom 9) and context-sensitive (Axiom 10) models is evident, particularly at the level of individual neurons.

The denoising test (Axiom 11) and the generalisation test (Axiom 12) correspond to cognitive abilities in which humans excel, such as recalling and generalising when confronted with new or distorted inputs. Finally, liveness (Axiom 13) is a must for a brain in order to support the real-time behaviour of the organism.

3.6.2. Classical symbolic program performance

In principle, two alternative design approaches to AGI can be distinguished. One incorporates explicit or implicit bias in the form of prior knowledge about the external environment. The other excludes environmental bias, beginning from an initial model that contains no environment-specific knowledge.

The former category encompasses most AI and purported AGI systems developed to date; however, it remains fundamentally constrained by the Symbol Grounding Problem (SGP) [7]. Although such systems may display behaviour that appears intelligent, their interpretations of symbols depend on programmer-supplied conventions rather than grounded understanding, and they therefore cannot qualify as genuine AGI.

More specifically, biased systems incorporate the designer's assumptions about the meaning of the signals they process. In classical symbolic architectures, the program itself constitutes prior knowledge: its rules and representations presuppose interpretations of the symbols being manipulated. The very existence of such a program violates AGITB's first test, which prohibits external knowledge of any kind. In effect, the AGI program smuggles in the symbol-grounding problem it is meant to avoid.

AGITB's initial test formalises the requirement to learn from scratch, demanding that a system derive structure and meaning solely through exposure to intrinsically grounded binary signals. Only a universal system could, in principle, satisfy this condition. This suggests that a genuine AGI may not explicitly encode the operations of intelligence, but rather the dynamics of a substrate from which they can emerge. This aligns with the idea of the "Brain Simulator Reply" [16] to Searle's Chinese Room Argument [17]. To date, however, no such system has been demonstrated.

3.6.3. Artificial neural network performance

The expectation dichotomy also extends to connectionist architectures. In principle, artificial neural networks may incorporate built-in expectations, introduced through pretraining regimes or architectural priors, or they may be configured as expectation-free systems that start from a neutral initial position.

AGITB's requirement that models begin uninformed about their environment stands in fundamental tension with the dominant paradigm of modern deep learning. Contemporary neural models typically rely on extensive pretraining, during which network weights are shaped by prior exposure to structured or labelled data. Moreover, by mapping symbolic inputs to numerical vectors, standard ANNs effectively shift the symbol-grounding problem into a *number-grounding* problem. Although such vector representations capture relational regularities within the training data, they also introduce spurious associations not anchored in real-world semantics, leading to the phenomenon commonly described as hallucination. Internal model coherence does not entail external semantic validity.

An expectation-free network is intended to contain no environment-specific knowledge in its weights or architecture. Setting all weights to zero yields a degenerate system incapable of effective processing, while random initialization avoids this degeneracy at the cost of introducing implicit prior structure, violating the assumption of uninformedness. This highlights a fundamental limitation of current ANN architectures: they do not learn autonomously but instead depend on an external training procedure to drive adaptation. AGITB, by contrast, requires a blank system capable of autonomous adaptation in an unfamiliar environment. To date, no such mechanism has been demonstrated in artificial neural networks.

3.6.4. Large language model performance

Because contemporary large language models (LLMs) are deployed only after extensive pre-training, they fail the uninformed start test (Axiom 1). Their initial behaviour is strongly shaped by statistical regularities extracted from large linguistic corpora, rather than emerging solely from interaction-driven learning. Although an LLM's internal parameters and activations are technically accessible and can, unlike those of a human, be inspected or compared across instances, such analyses remain secondary until the uninformed start requirement is satisfied.

A further limitation arises from standard transformer-based implementations, which rely on a fixed-size context window [18]. When this capacity is exceeded, earlier tokens must be compressed, attenuated, or discarded [19]. To the extent that discarded information leaves no persistent internal representation, this mechanism fails to preserve an unbroken experiential trace, thereby violating the Trace requirement (Axiom 4).

Nevertheless, the question of whether an LLM can autonomously derive a solution to AGITB when prompted is straightforward to evaluate empirically. Multiple attempts using distinct prompts (one example is provided in Appendix A) yielded unpromising results: although systems such as ChatGPT, Gemini or Claude produced candidate programs that purported to satisfy the requirements, none progressed beyond the temporal adaptability test (Axiom 8).

In summary, LLMs do not engage in genuine learning solely from prompts, nor can they acquire the grounded, context-dependent understanding characteristic of human cognition. These limitations extend to large reasoning models, which inherit the same fundamental architectural constraints.

3.7. Remarks

AGITB evaluates a model's predictive capabilities after exposure to temporal sequences of both structured and random inputs. Random input sequences with arbitrary internal correlations are employed to minimise reliance on pretraining, ensuring that any observed learning arises from the input stream itself rather than from prior knowledge. By enforcing fundamental computational invariants of cortical function at the signal-processing level, AGITB remains agnostic to the external meaning of signals; the random inputs do not need to resemble real-world sensory data.

The low-level, binary operational framework makes AGITB particularly well-suited for evaluating NeuroAI models that aim to satisfy the principles of the embodied Turing Test [20], wherein cognitive understanding emerges from the integration of continuous sensory streams. The progression from raw signal prediction to higher-level abstraction mirrors the broader trajectory of AI, from early perceptrons to large-scale models such as GPT.

3.8. Cheating the benchmark

Because AGITB's tests are individually simple to solve, one might imagine circumventing the benchmark by engineering task-specific solutions and having the model selectively deploy them depending on the detected test scenario. In principle, the task being administered could be inferred by monitoring the number of instantiated models and the sequence of invoked methods. Or one could exploit the inability of AGITB to guard against random generator issues, and similar. !!!!!!!!!!!

However, such approaches would amount to subverting the benchmark rather than advancing AGI research. Although AGITB could be hardened against this form of cheating (by, for example, shuffling tests or redesigning the programming interface), these measures would reduce the transparency and interpretability of the testbed, thereby hindering its intended use by human developers.

The next potential avenue for circumventing the benchmark is to construct a model that passes AGITB only because the testbed uses a finite approximation of conditions that are, in principle, unbounded. Several requirements would ideally be evaluated over an infinite number of test cases, but such tests are computationally infeasible. As a practical compromise, AGITB executes a fixed number of iterations intended to approximate an otherwise indefinite process. This parameter, denoted `SimulatedInfinity` in the reference implementation, is currently set to 5,000.

Although this value is far from representing true infinity, it is presently believed to work well in combination with the other benchmark settings (temporal patterns with seven inputs of ten bits each) and to be sufficient for distinguishing promising approaches from non-promising ones. At the same time, it maintains computational efficiency, enabling rapid evaluation of diverse model prototypes.

For these reasons, the AGITB reference implementation is kept deliberately readable and fast to execute. To date, no artificial system has demonstrated the level of performance required by AGITB. Unless a credible attempt to circumvent the benchmark emerges, there is no justification for introducing a more obfuscated or slower and more cumbersome version of the testbed.

4. Competing benchmarks

Among existing benchmark tasks, the Abstraction and Reasoning Corpus [4] is most closely aligned in spirit, as it likewise emphasises generalisation over task-specific optimisation. A related effort is NeuroBench, which is designed to support the systematic evaluation of neuromorphic and other biologically inspired architectures. Both ARC and NeuroBench rely on a variety of correctness and complexity metrics to compare non-AGI models; their primary purpose is to distinguish weaker from stronger narrow systems. In contrast, AGITB is designed to evaluate whether a model satisfies a set of foundational capabilities that are plausibly associated with more general forms of intelligence, rather than to rank systems along a performance spectrum.

4.1. ARC

ARC presents visual reasoning tasks in which a model must infer novel transformations (such as recolouring, rearranging, or modifying spatial patterns) from a sequence of two input–output examples defined on discrete spatial grids.

However, ARC implicitly assumes the presence of high-level cognitive priors, including object permanence, spatial reasoning, numerical abstraction, and causal inference. These priors are not formally specified, placing an ambiguous and open-ended burden on the model designer. In contrast, AGITB adopts a fundamentally different stance: it treats the system under evaluation as a blank slate that must acquire structure and function exclusively through interaction with temporally structured input.

Although ARC presumes some form of temporal reasoning, it does not adequately support it, as each task provides only two images to illustrate a transformation. AGITB, by contrast, evaluates cognition as a dynamic process unfolding over time. A model can acquire knowledge and predictive capability only through continuous exposure to temporally structured data, not from disconnected before–and–after snapshots that lack the temporal continuity needed to infer causal relationships. For example, to recognise an object moving left, a model in AGITB must observe multiple intermediate states across time; the final image alone is insufficient to infer the transformation. Temporal structure, rather than static pattern comparison, provides the substrate from which invariants and causal relations can be learned.

ARC remains susceptible to the symbol-grounding problem whenever pixel colours are encoded as numbers, since numerical labels (0–9) impose externally defined semantics that may not align with the model's internal representation of colour. Under such a scheme, a colour functions as a human-assigned numerical category rather than as an intrinsically grounded signal. Encoding colour in additional binary dimensions using one-hot representations may mitigate the issue in ARC, where only ten colours are used, and such an expansion is still tractable. However, this strategy does not scale and therefore does not alleviate the broader symbol-grounding problem in general.

In summary, ARC evaluates high-level intelligence grounded in human cognitive priors, whereas AGITB evaluates adherence to fourteen low-level computational requirements intended to support the emergence of such priors. ARC and the Turing Test both frame intelligence through an anthropocentric lens, embedding assumptions drawn from human cognition. AGITB instead conceptualises intelligence as a universal capacity for learning that does not rely on innate symbolic structures or species-specific expectations.

4.2. NeuroBench

NeuroBench provides a unified framework for benchmarking diverse AI models across a standardised set of tasks and metrics. It is particularly oriented toward neuromorphic approaches, which have demonstrated advantages in resource efficiency and scalability. Within its algorithm track, the framework evaluates models on several challenges relevant to general AI research, including few-shot continual learning, object detection, sensorimotor decoding, and predictive modelling.

The predictive modelling challenge, which involves forecasting chaotic functions, is most closely aligned with AGITB's central premise that intelligence fundamentally concerns the prediction of future inputs. NeuroBench employs a synthetic one-dimensional Mackey-Glass time series for this task, a dataset designed for architectures with limited input/output capacity.

However, several issues limit the usability of chaotic function prediction (CFP) as a general AGI benchmark task. *First*, the Mackey-Glass data are numerical, and NeuroBench does not prescribe the encoding scheme. An inappropriate encoding can distort the temporal and causal structure of the observed signals, such that a numeric value—much like a symbol—derives its meaning from human interpretation rather than from the model's own grounded understanding. This effectively reintroduces the symbol-grounding problem in a numerical form.

Second, the threshold for AGI-level performance is not clearly defined. Although the symmetric mean absolute percentage error (sMAPE) is a standard forecasting metric, NeuroBench does not specify what performance level corresponds to general intelligence. Notably, humans themselves perform poorly at anomaly detection and long-horizon prediction of the Mackey-Glass signal [21].

Third, although long-term prediction is not inherently problematic, predicting multiple steps ahead without timely feedback deprives a system of the opportunity to detect and correct its own errors. This design is incompatible with online learning, where an AGI should continuously update itself upon observing discrepancies between predictions and outcomes. NeuroBench, by contrast, emphasises offline learning and assumes that an AGI would behave as a purely mechanistic predictor, lacking intrinsic mechanisms for self-correction, autonomous adaptation, and genuine agency.

Table 1 highlights the key differences among the tasks used in the three benchmarks. Whereas ARC and NeuroBench presuppose or require models to exhibit high-level cognitive capacities (such as object recognition, spatial manipulation, and various forms of reasoning), AGITB instead focuses on minimal, precisely defined requirements that can be evaluated directly at the signal-processing level.

Property	ARC	CFP	AGITB
Interface modality	Visual	Numeric	Binary
AGI type	Human	Universal	Universal
Cognitive priors	Yes	No	No
Abstraction level	High	Medium	Low
Task preparation	Manual	Automatic	Automatic
Grounding Problem	Yes	Yes	No
Input dimensionality	30×30 numbers	1 number	10 bits
Temporal sequence length	2	750+	7+

Table 1: Core properties of ARC, NeuroBench's chaotic function prediction (CFP), and AGITB.

5. Conclusion

Unlike conventional benchmarks that target high-level task performance, such as question answering or language translation, AGITB evaluates whether a system exhibits behaviours associated with core operational principles of the biological cortex. Its focus is on low-level, biologically grounded computational properties that are believed to underlie the emergence of general intelligence. The testbed comprises fourteen tightly interdependent tests, each simple in isolation but collectively requiring the kind of learning expected of an AGI.

AGITB requires models to begin uninformed and to acquire all functionality solely through exposure to structured or random input. This aligns with neuroscientific evidence that cortical learning is fundamentally input-driven: neural circuits develop through experience, not through pre-encoded semantics. In biological systems, high-level cognition arises not from symbolic manipulation but from the continual adaptive prediction of low-level sensory signals. Such prediction is more than pattern matching; it supports the progressive construction of signal-grounded knowledge from which abstraction and generalisation can emerge.

AGITB is solvable by humans yet remains unsolved by classical algorithms and current state-of-the-art AI systems. This persistent performance gap provides empirical support for the claim that AGITB targets capabilities characteristic of general rather than narrow intelligence. Although the absence of an artificial solution does not constitute a formal proof of adequacy, the fact that humans succeed where machines do not indicates that the benchmark captures functionally relevant aspects of general intelligence. In this sense, AGITB serves as a discriminative test and a principled tool for steering the development of systems capable of genuinely general, adaptive learning.

Funding

The author acknowledges the financial support from the Slovenian Research Agency (research core funding No. P2-0057).

Appendix A. Model construction prompt

You are an expert engineer and sequence-learning researcher. Your task is to create a **concrete solution** that satisfies the AGITB benchmark defined in the accompanying .zip archive.

Produce complete, compilable C++20 source code, including all required classes, methods, and internal logic needed to satisfy the benchmark's tests.

1. Study the following files:
– `README.md`
– `include/agitb.h`
– `include/utils.h`
From these, extract and understand:
– The exact **API contract** for the system-under-evaluation model.
– All **requirements and tests** that define the model's expected behaviour.
– Any helper utilities or wrappers that affect how the model is used.

2. Design a plausible AGITB candidate model
– Design a model class that satisfies the AGITB requirements.
– Architecturally, choose the **scientifically most suitable** predictor model, or a mixture of models, or any other solution type you deem appropriate.

3. Output format
– Output the complete, compilable C++20 code for `MyModel`.
– Clearly state how your design is expected to perform on the AGITB tests.

Use all of the instructions above to guide your analysis and implementation.

References

[1] G. Marcus, Deep learning: A critical appraisal (2018). arXiv:1801.00631.
URL https://arxiv.org/abs/1801.00631

[2] M. Mitchell, Why AI chatbots lie to us, Science 389 (6758) (2025) eaea3922. doi:10.1126/science.aea3922.

[3] A. M. Turing, Computing machinery and intelligence, Mind 59 (236) (1950) 433–460.
URL http://www.jstor.org/stable/2251299

[4] F. Chollet, On the measure of intelligence (2019). arXiv:1911.01547.
URL https://arxiv.org/abs/1911.01547

[5] J. Yik, K. V. den Berghe, D. den Blanken, Y. Bouhadjar, M. Fabre, P. Hueber, W. Ke, M. A. Khoei, D. Kleyko, N. Pacik-Nelson, A. Pierro, P. Stratmann, P.-S. V. Sun, G. Tang, S. Wang, B. Zhou, S. H. Ahmed, G. V. Joseph, B. Leto, A. Micheli, A. K. Mishra, G. Lenz, T. Sun, Z. Ahmed, M. Akl, B. Anderson, A. G. Andreou, C. Bartolozzi, A. Basu, P. Bogdan, S. Bohte, S. Buckley, G. Cauwenberghs, E. Chicca, F. Corradi, G. de Croon, A. Danielescu, A. Daram, M. Davies, Y. Demirag, J. Eshraghian, T. Fischer, J. Forest, V. Fra, S. Furber, P. M. Furlong, W. Gilpin, A. Gilra, H. A. Gonzalez, G. Indiveri, S. Joshi, V. Karia, L. Khacef, J. C. Knight, L. Kriener, R. Kubendran, D. Kudithipudi, S.-C. Liu, Y.-H. Liu, H. Ma, R. Manohar, J. M. Margarit-Taulé, C. Mayr, K. Michmizos, D. R. Muir, E. Neftci, T. Nowotny, F. Ottati, A. Ozcelikkale, P. Panda, J. Park, M. Payvand, C. Pehle, M. A. Petrovici, C. Posch, A. Renner, Y. Sandamirskaya, C. J. S. Schaefer, A. van Schaik, J. Schemmel, S. Schmidgall, C. Schuman, J. sun Seo, S. Sheik, S. B. Shrestha, M. Sifalakis, A. Sironi, K. Stewart, M. Stewart, T. C. Stewart, J. Timcheck, N. Tömen, G. Urgese, M. Verhelst, C. M. Vineyard, B. Vogginger, A. Yousefzadeh, F. T. Zohora, C. Frenkel, V. J. Reddi, The neurobench framework for benchmarking neuromorphic computing algorithms and systems, Nature Communications 16 (1) (2025) 1545. doi:10.1038/s41467-025-56739-4.

[6] W. Maass, Networks of spiking neurons: The third generation of neural network models, Neural Networks 10 (9) (1997) 1659–1671. doi:10.1016/S0893-6080(97)00011-7.

[7] S. Harnad, The symbol grounding problem, Physica D: Nonlinear Phenomena 42 (1) (1990) 335–346. doi:10.1016/0167-2789(90)90087-6.

[8] E. M. Bender, A. Koller, Climbing towards NLU: On meaning, form, and understanding in the age of data, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5185–5198. doi:10.18653/v1/2020.acl-main.463.
URL https://aclanthology.org/2020.acl-main.463/

[9] Y. Bisk, A. Holtzman, J. Thomason, J. Andreas, Y. Bengio, J. Chai, M. Lapata, A. Lazaridou, J. May, A. Nisnevich, N. Pinto, J. Turian, Experience grounds language, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 8718–8735. doi:10.18653/v1/2020.emnlp-main.703.
URL https://aclanthology.org/2020.emnlp-main.703/

[10] R. Gubelmann, Pragmatic norms are all you need – why the symbol grounding problem does not apply to LLMs, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 11663–11678. doi:10.18653/v1/2024.emnlp-main.651.
URL https://aclanthology.org/2024.emnlp-main.651/

[11] J. Hawkins, S. Blakeslee, On Intelligence, Times Books, 2004.

[12] M. Šprogar, A ladder to human-comparable intelligence: an empirical metric, Journal of Experimental & Theoretical Artificial Intelligence 30 (6) (2018) 1037–1050. doi:10.1080/0952813X.2018.1509897.

[13] S. Cave, There's no such thing as free will (2016).
URL https://www.theatlantic.com/magazine/archive/2016/06/theres-no-such-thing-as-free-will/480750/

[14] W. Gerstner, W. M. Kistler, Spiking Neuron Models: Single Neurons, Populations, Plasticity, Cambridge University Press, 2002. doi:10.1017/cbo9780511815706.

[15] S. Harnad, The turing test is not a trick: Turing indistinguishability is a scientific criterion, SIGART Bull. 3 (4) (1992) 9–10. doi:10.1145/141420.141422.

[16] P. Churchland, P. Churchland, Could a machine think?, Scientific American 262 (1) (1990) 32–37.

[17] J. R. Searle, Minds, brains, and programs, Behavioral and Brain Sciences 3 (3) (1980) 417–424. doi:10.1017/S0140525X00005756.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
URL http://arxiv.org/abs/1706.03762

[19] N. Paulsen, Context is what you need: The maximum effective context window for real world limits of llms (2025). arXiv:2509.21361.
URL https://arxiv.org/abs/2509.21361

[20] A. Zador, S. Escola, B. Richards, B. Ölveczky, Y. Bengio, K. Boahen, M. Botvinick, D. Chklovskii, A. Churchland, C. Clopath, J. DiCarlo, S. Ganguli, J. Hawkins, K. Körding, A. Koulakov, Y. LeCun, T. Lillicrap, A. Marblestone, B. Olshausen, A. Pouget, C. Savin, T. Sejnowski, E. Simoncelli, S. Solla, D. Sussillo, A. S. Tolias, D. Tsao, Catalyzing next-generation artificial intelligence through NeuroAI, Nature Communications 14 (1) (2023) 1597. doi:10.1038/s41467-023-37180-x.

[21] M. Thill, W. Konen, T. Bäck, Time series encodings with temporal convolutional networks, in: B. Filipič, E. Minisci, M. Vasile (Eds.), Bioinspired Optimization Methods and Their Applications, Springer International Publishing, Cham, 2020, pp. 161–173. doi:10.1007/978-3-030-63710-1_13.