

# A Mind Cannot Be Smeared Across Time

Michael Timothy Bennett

School of Computing, The Australian National University  
Canberra, ACT, Australia  
m@michaeltimothybennett.com

## Abstract

Whether machines can be conscious depends not only on what they compute, but *when* they compute it. Most deployed artificial systems realise their functions via sequential or time-multiplexed updates, yet a moment of conscious experience feels unified and simultaneous. I prove that this difference matters. I augment Stack Theory with algebraic laws relating within time-window constraint satisfaction to conjunction. I introduce a temporal semantics over windowed trajectories  $\tau_{\Delta}$  and prove that existential temporal realisation  $\diamond_{\Delta}$  does not preserve conjunction. A system can realise all the ingredients of experience across time without ever instantiating the experienced conjunction itself. I then distinguish two postulates, Chord and Arpeggio. Chord is the position that conscious unity requires *objective co-instantiation* of the grounded conjunction within the window, like a musical chord. Arpeggio only needs the ingredients to *occur* within window, like a melody. I formalise concurrency-capacity to measure what is needed to satisfy co-instantiation. Finally, I review neurophysiological evidence suggesting that consciousness depends on phase synchrony and effective connectivity, and that loss of consciousness is associated with its breakdown. Under Chord, software consciousness on strictly sequential substrates is impossible for contents whose grounding requires two or more simultaneous contributors. The hardware matters.

**Supplementary material** — <https://github.com/ViscousLemming/Technical-Appendices>

## Introduction

A central open question in machine consciousness is whether *behaviourally* and *functionally* matched systems must be matched in *phenomenal realisation*. Stack Theory approaches this by formalising cognitive processes in terms of abstraction layers (Bennett 2025b, 2023a; Bennett, Welsh, and Ciaunica 2024). Various aspects of incumbent theories of consciousness are formalised (Seth and Bayne 2022; Block 1995; Chalmers 1995; Rosenthal 1986; Solms 2021; Baars 1988), organised into a hierarchy of causal identities (Bennett 2023a), and derived as a necessary consequence of a generalisation-optimal learning theory that makes valence primary (Bennett 2025a,c, 2023b, 2022). Those necessary ingredients of consciousness are beyond the scope of this

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

paper. Here I am concerned with a question of sufficiency. Stack Theory shows how the behaviour of two systems can *appear* identical at one level of abstraction, while being different in the levels below. Conscious experience can be purely functional, but still depend on more than surface level function—not just on *what* is computed, but *when* and *how*. The ingredients of what looks like a single experienced moment at a high level of abstraction may be smeared across objective micro-time at a lower level, such that no objective time-slice contains the full grounded conjunction. If the ingredients of a subjective moment need not synchronise in objective time, then current hardware suffices to host machine consciousness—and a “liquid-brained” swarm intelligence (Solé, Moses, and Forrest 2019) like an ant colony can be conscious. But if synchronised broadcast and causal co-dependence matter, then far fewer systems qualify. This question of architectural sufficiency is the *Temporal Gap*.

The Temporal Gap was proposed in earlier work (Bennett 2025b), but was not rigorously formalised. Here I add a *Stack-Time Semantics* module to Stack Theory, including induced layer time, window environments, temporal lifting operators, and algebraic laws relating “within-window” satisfaction to conjunction. Using this, I refine the Temporal Gap into a testable architectural question about synchrony and concurrency.

**Relation to existing consciousness theories.** Beyond Stack Theory, many scientific theories emphasise some form of *integration* or *unity* across time and causal organisation: global workspace theory (Baars 1988; Dehaene and Naccache 2001), integrated information (Tononi 2004; Oizumi, Albantakis, and Tononi 2014), higher-order representation (Rosenthal 1986), attention-schema modelling (Webb and Graziano 2015), predictive-processing and active-inference (Friston 2010), and more general analyses of time and consciousness (James 1890; Varela 1999). The aim here is not to adjudicate among theories, but to extract a common formal issue: if phenomenal content is modelled as a *conjunction* of grounded ingredients, what does it mean for that conjunction to be realised *within the temporal extent of a subjective moment*?

## Results

1. **Compositional grounding.** Any higher-layer statement can be grounded compositionally to the base layer preserving truth conditions (Theorem 1).
2. **Subjective vs objective time.** Higher-layer ticks are maximal objective-time blocks of constant encoding, and can be arbitrarily sparse (Proposition 1).
3. **Temporal lifting algebra.** The core non-commutation result underpinning the Temporal Gap (Theorem 3).
4. **Chord vs Arpeggio postulates.** Both postulates stated in terms of a higher-layer moment statement and its base-layer grounding (Definitions 16 and 17).
5. **Concurrency capacity threshold.** Architectures with limited simultaneous contribution can satisfy ingredient-wise occurrence while forbidding co-instantiation beyond capacity (Theorem 4).
6. **Evidence supporting Chord.** Co-instantiation requirements are connected to evidence linking consciousness to synchrony and effective connectivity (Section ).

## Stack Theory Primitives

I follow the core notation of Stack Theory (Bennett 2025d,b).

**Definition 1** (Environment and programs). An *environment* is a nonempty set  $\Phi$  of mutually exclusive states. A *program* is a set  $p \subseteq \Phi$ . A program  $p$  is true at  $\phi \in \Phi$  iff  $\phi \in p$ . Let  $P := 2^\Phi$  be the set of all programs. A *vocabulary* is any finite  $\mathfrak{v} \subseteq P$ .

**Intuition.** An environment is the set of possible states. A program is a set of states, each state representing a point in time and a smallest possible change in an environment, equating time with change (Bennett 2025b). A program is true at a state when the state is in that set. Conjunction is set intersection.

**Definition 2** (Statements, truth sets, extensions). The *language* generated by  $\mathfrak{v}$  is

$$\mathcal{L}_{\mathfrak{v}} := \left\{ l \subseteq \mathfrak{v} : l \text{ is finite and } \bigcap_{p \in l} p \neq \emptyset \right\}.$$

Elements  $l \in \mathcal{L}_{\mathfrak{v}}$  are (conjunctive) *statements*. The *truth set* of  $l$  is

$$\mathsf{T}(l) := \bigcap_{p \in l} p \subseteq \Phi.$$

I say  $l$  is true at  $\phi$  iff  $\phi \in \mathsf{T}(l)$ . By convention,  $\mathsf{T}(\emptyset) = \Phi$ .

A *completion* of  $l$  is any  $l' \in \mathcal{L}_{\mathfrak{v}}$  with  $l \subseteq l'$ . Define the *extension* of  $l$  by  $\text{Ext}(l) := \{l' \in \mathcal{L}_{\mathfrak{v}} : l \subseteq l'\}$ . For  $X \subseteq \mathcal{L}_{\mathfrak{v}}$ , write  $\text{Ext}(X) := \bigcup_{x \in X} \text{Ext}(x)$ .

**Intuition.** A statement is a finite bundle of programs that must all be true at once. A completion adds more programs, making the statement more specific. The extension collects all completions that still include the original statement.

**Definition 3** (Tasks and policies). A *task* is a pair  $\alpha = \langle I_\alpha, O_\alpha \rangle$  where  $I_\alpha \subseteq \mathcal{L}_{\mathfrak{v}}$  is a set of admissible input statements and  $O_\alpha \subseteq \text{Ext}(I_\alpha)$  is a set of correct output statements. A *policy* is any statement  $\pi \in \mathcal{L}_{\mathfrak{v}}$ . A policy is *correct* for  $\alpha$  iff

$$\text{Ext}(I_\alpha) \cap \text{Ext}(\pi) = O_\alpha.$$

Let  $\Pi_\alpha$  denote the set of correct policies for  $\alpha$ .

**Intuition.** A task specifies which inputs are allowed and which outputs count as correct. A policy is correct when it selects exactly the correct outputs for the allowed inputs.

## Grounding Stack Theory

**Definition 4** (Abtractor). Fix a base vocabulary  $\mathfrak{v} \subseteq P$  and a policy  $\pi \in \mathcal{L}_{\mathfrak{v}}$ . The induced *abtractor* is the function

$$\mathfrak{f}(\mathfrak{v}, \pi) := \left\{ q \in P : \exists o \in \text{Ext}(\pi) \text{ s.t. } \bigcap_{r \in o} r = q \right\}.$$

Thus  $\mathfrak{f}(\mathfrak{v}, \pi)$  is a new vocabulary consisting of truth sets of completions compatible with  $\pi$ .

**Intuition.** The abtractor takes completions of the policy and turns each into one set of states. The collection of these sets forms the vocabulary of the next layer—a compression that maps completions to equivalent programs.

**Definition 5** (Stack). A *stack of depth  $m$*  is a sequence of uninstantiated task functions  $\langle \lambda^0, \lambda^1, \dots, \lambda^m \rangle$  such that for each  $i < m$ ,  $\lambda^{i+1}$  is a restriction of  $\lambda^i$  (written  $\lambda^{i+1} \sqsubset \lambda^i$ ). Each  $\lambda^i$  assigns to each vocabulary  $\mathfrak{v}$  a task  $\lambda^i(\mathfrak{v})$ . The notation  $\Pi_{\lambda^i(\mathfrak{v})}$  denotes the set of correct policies for that task. A *stack state* is a pair of sequences

$$\langle \pi^0, \dots, \pi^{m-1} \rangle \quad \text{and} \quad \langle \mathfrak{v}^0, \dots, \mathfrak{v}^m \rangle$$

satisfying, for each  $i < m$ ,

$$\mathfrak{v}^{i+1} = \mathfrak{f}(\mathfrak{v}^i, \pi^i) \quad \text{and} \quad \pi^i \in \Pi_{\lambda^i(\mathfrak{v}^i)}.$$

**Intuition.** At each layer you pick a policy that solves the task for that layer. That policy also defines the next vocabulary, the higher level description of what is going on.

## Grounding a Higher-Layer into Lower Layers

**Lemma 1** (Grounding through one abtractor). Let  $\mathfrak{v} \subseteq P$  and  $\pi \in \mathcal{L}_{\mathfrak{v}}$ , and set  $\mathfrak{v}' = \mathfrak{f}(\mathfrak{v}, \pi)$ . Then for every program  $p \in \mathfrak{v}'$  there exists a completion  $o_p \in \text{Ext}(\pi)$  such that  $\mathsf{T}(o_p) = p$ . Moreover, for any statement  $l \in \mathcal{L}_{\mathfrak{v}'}$  there exists a statement  $\text{ground}_\pi(l) \in \mathcal{L}_{\mathfrak{v}}$  with  $\mathsf{T}(l) = \mathsf{T}(\text{ground}_\pi(l))$ .

*Proof.* See Supplementary Material. Each macro-ingredient has a lower-layer completion with the same truth set; replacing ingredients and intersecting preserves truth conditions.  $\square$

## Compositional Grounding

**Definition 6** (Compositional grounding across a stack). Fix a stack with vocabularies  $\mathfrak{v}^0, \dots, \mathfrak{v}^m$  and policies  $\pi^0, \dots, \pi^{m-1}$ . For a statement  $l^m \in \mathcal{L}_{\mathfrak{v}^m}$ , define grounded statements  $g^m, \dots, g^0$  recursively by

$$g^m := l^m, \\ g^i := \text{ground}_{\pi^i}(g^{i+1}) \in \mathcal{L}_{\mathfrak{v}^i} \quad (i = m-1, \dots, 0).$$

Define the *compositional grounding* map  $\text{Ground}_{0 \leftarrow m}$  by  $\text{Ground}_{0 \leftarrow m}(l^m) := g^0$ .

**Intuition.** Grounding works like compiling a high level statement into lower level pieces. Replace each high level ingredient with a lower level statement that means the same thing. Repeating this down the stack gives a base level conjunction true in exactly the same cases.

**Theorem 1** (Compositional grounding preserves truth conditions). *For any stack and any statement  $l^m \in \mathcal{L}_{\mathfrak{v}^m}$ ,*

$$\mathbb{T}(\text{Ground}_{0 \leftarrow m}(l^m)) = \mathbb{T}(l^m).$$

*Proof.* Apply Lemma 1 iteratively:  $\mathbb{T}(g^i) = \mathbb{T}(g^{i+1})$  for each  $i < m$ , hence  $\mathbb{T}(g^0) = \mathbb{T}(g^m) = \mathbb{T}(l^m)$ .  $\square$

## Stack-Time Semantics Module

This section defines induced layer time, window semantics, and temporal lifting operators as a reusable addendum to Stack Theory.

### Objective Time and Layer Time as a Quotient

**Definition 7** (Objective time and trajectories). Objective time is represented by a discrete index  $t \in \mathbb{N}$ . An *objective trajectory* is a function  $\tau : \mathbb{N} \rightarrow \Phi$  such that  $\tau(t+1) \neq \tau(t)$  for all  $t \in \mathbb{N}$ .

**Intuition.** Objective time is a step counter for objective change. If nothing changes then there is no next objective index. Two events are simultaneous when they occur at the same objective index.

**Definition 8** (Encoding map). For a vocabulary  $\mathfrak{v} \subseteq P$  and a state  $\phi \in \Phi$ , define  $\text{Enc}_{\mathfrak{v}}(\phi) := \{p \in \mathfrak{v} : \phi \in p\} \subseteq \mathfrak{v}$ .

**Intuition.** Given a state and a vocabulary, the encoding collects all programs in that vocabulary true at that state—the most detailed true statement in that vocabulary.

**Lemma 2** (Encoding yields the maximal true statement). *For any  $\mathfrak{v}$  and  $\phi \in \Phi$ ,  $\text{Enc}_{\mathfrak{v}}(\phi) \in \mathcal{L}_{\mathfrak{v}}$  and  $\phi \in \mathbb{T}(\text{Enc}_{\mathfrak{v}}(\phi))$ . Moreover, if  $l \in \mathcal{L}_{\mathfrak{v}}$  is true at  $\phi$ , then  $l \subseteq \text{Enc}_{\mathfrak{v}}(\phi)$ .*

*Proof.* See Supplementary Material.  $\square$

**Definition 9** (Induced layer trajectory and layer time). Given an objective trajectory  $\tau$  and a vocabulary  $\mathfrak{v}$ , define the induced layer trajectory  $\tau^{(\mathfrak{v})}(t) := \text{Enc}_{\mathfrak{v}}(\tau(t)) \in \mathcal{L}_{\mathfrak{v}}$ . Define the *layer-tick map*  $\kappa_{\mathfrak{v}} : \mathbb{N} \rightarrow \mathbb{N}$  by  $\kappa_{\mathfrak{v}}(0) = 0$  and, for  $t > 0$ ,

$$\kappa_{\mathfrak{v}}(t) := \begin{cases} \kappa_{\mathfrak{v}}(t-1) & \text{if } \tau^{(\mathfrak{v})}(t) = \tau^{(\mathfrak{v})}(t-1), \\ \kappa_{\mathfrak{v}}(t-1) + 1 & \text{otherwise.} \end{cases}$$

Define  $t \approx_{\mathfrak{v}} t'$  iff  $\kappa_{\mathfrak{v}}(t) = \kappa_{\mathfrak{v}}(t')$ . The *layer time* is  $\mathbb{N}/\approx_{\mathfrak{v}}$ : each equivalence class is a maximal contiguous block of objective times on which  $\tau^{(\mathfrak{v})}$  is constant.

**Intuition.** For each objective time step, write down the encoding. If it stays the same for several steps, treat them as one higher level moment. Layer time advances only when something visible at that level changes.

**Proposition 1** (Layer time can be arbitrarily sparse). *For any  $N \in \mathbb{N}$  there exist  $\Phi, \mathfrak{v}, \tau$  such that the induced trajectory  $\tau^{(\mathfrak{v})}$  changes at most once in the first  $N$  objective steps.*

*Proof.* See Supplementary Material. Choose a vocabulary that cannot distinguish many distinct objective microstates; the macrostate stays constant.  $\square$

### Time and Lifting Operators

A subjective moment is commonly modelled as having nonzero temporal extent (the “specious present”) (James 1890; Varela 1999). I introduce temporal operators to formalise satisfaction *within* such a window.

**Definition 10** (Ingredient-wise vs co-instantiated satisfaction). Fix a windowing map  $W$ , an objective trajectory  $\tau$ , and a window index  $t$ . For a program  $p \subseteq \Phi$ , define

$$\text{Occur}_W(p, \tau, t) \iff \exists u \in W(t) (\tau(u) \in p).$$

For a statement  $l \in \mathcal{L}_{\mathfrak{v}}$ , define

$$\text{Occur}_W(l, \tau, t) \iff \forall p \in l \text{ Occur}_W(p, \tau, t).$$

Define co-instantiation of the conjunction by

$$\text{CoInst}_W(l, \tau, t) \iff \exists u \in W(t) (\tau(u) \in \mathbb{T}(l)).$$

**Intuition.** Ingredient-wise means every ingredient becomes true at least once somewhere in the window. Co-instantiated means there is a time step in the window where all ingredients are true together.

*Remark 1* (Quantifier structure). For conjunctive  $l$ , ingredient-wise satisfaction is  $\forall p \in l \exists u \in W(t)$ , while co-instantiation is  $\exists u \in W(t) \forall p \in l$ . The Temporal Gap amounts to whether conscious unity requires ingredient-wise satisfaction or co-instantiation.

**Definition 11** (Window environments and window trajectories). Fix  $\Delta \in \mathbb{N}$ . Define the *window environment*  $\Phi^\Delta := \Phi^{\{0,1,\dots,\Delta\}} \cong \Phi^{\Delta+1}$  whose elements are windows  $\sigma = (\phi_0, \dots, \phi_\Delta)$ . Given an objective trajectory  $\tau$ , define the induced window trajectory  $\tau_\Delta(t) := (\tau(t), \tau(t+1), \dots, \tau(t+\Delta)) \in \Phi^\Delta$ .

**Intuition.** Treat a whole window of  $\Delta+1$  consecutive states as a single new state.

**Definition 12** (Temporal lifts of programs). For a program  $p \subseteq \Phi$ , define the lifted programs on  $\Phi^\Delta$ :

$$\begin{aligned} \diamond_{\Delta} p &:= \{ \sigma \in \Phi^\Delta : \exists k \leq \Delta \text{ s.t. } \sigma(k) \in p \}, \\ \square_{\Delta} p &:= \{ \sigma \in \Phi^\Delta : \forall k \leq \Delta, \sigma(k) \in p \}. \end{aligned}$$

**Intuition.**  $\diamond_{\Delta} p$  means  $p$  is true at least once in the window.  $\square_{\Delta} p$  means  $p$  is true at every step in the window.

**Definition 13** (Lifted statements). For a statement  $l \in \mathcal{L}_{\mathfrak{v}}$ , define

$$\begin{aligned} \diamond_{\Delta} l &:= \{ \diamond_{\Delta} p : p \in l \}, \\ \square_{\Delta} l &:= \{ \square_{\Delta} p : p \in l \}. \end{aligned}$$

Then  $\mathbb{T}(\diamond_{\Delta} l) = \bigcap_{p \in l} \diamond_{\Delta} p$  and similarly for  $\square$ .

**Intuition.** To lift a statement, lift each ingredient and require all lifted ingredients to hold. With  $\diamond_{\Delta}$  this means every ingredient shows up somewhere in the window.

*Remark 2* (Temporal semantics as an abstraction layer). The window environment  $\Phi^\Delta$  and the lifted vocabulary  $\{\diamond_\Delta p : p \in \mathfrak{v}\}$  define a new abstraction layer in the native Stack-Theory sense, bridging “over time” constraints and pointwise semantics. The compilation of history-dependent (pathwise) abstractors into pointwise abstractors on augmented state is detailed in the Supplementary Material.

## The Algebraic Core

**Theorem 2** (Universal lift commutes with conjunction). *For any statement  $l \in \mathcal{L}_\mathfrak{v}$  and any  $\Delta \in \mathbb{N}$ ,  $\square_\Delta T(l) = T(\square_\Delta l)$ .*

*Proof.* See Supplementary Material.  $\square_\Delta$  is “for all positions”; since  $\forall$  distributes over conjunction,  $\square_\Delta$  commutes with  $\wedge$ .  $\square$

**Theorem 3** (Existential lift does not commute with conjunction). *For any statement  $l \in \mathcal{L}_\mathfrak{v}$  and any  $\Delta \in \mathbb{N}$ ,*

$$\diamond_\Delta T(l) \subseteq T(\diamond_\Delta l).$$

*Moreover, for  $\Delta \geq 1$  there exist environments and statements for which the inclusion is strict.*

*Proof.* If  $\sigma \in \diamond_\Delta T(l)$  then for some  $k$  I have  $\sigma(k) \in T(l) = \bigcap_{p \in l} p$ . Hence  $\sigma \in \diamond_\Delta p$  for every  $p \in l$  and therefore  $\sigma \in \bigcap_{p \in l} \diamond_\Delta p = T(\diamond_\Delta l)$ .

For strictness when  $\Delta \geq 1$ , let  $\Phi = \{a, b, c\}$  and  $\mathfrak{v} = \{p, q\}$  with  $p = \{a, c\}$  and  $q = \{b, c\}$ . Then  $l = \{p, q\} \in \mathcal{L}_\mathfrak{v}$  because  $p \cap q = \{c\} \neq \emptyset$ . Take  $\Delta = 1$  and window  $\sigma = (a, b)$ . Then  $\sigma \in \diamond_1 p$  and  $\sigma \in \diamond_1 q$ , so  $\sigma \in T(\diamond_1 l)$ . But  $\sigma \notin \diamond_1 T(l) = \diamond_1(p \cap q)$  because neither  $a$  nor  $b$  lies in  $p \cap q = \{c\}$ .  $\square$

**Proof intuition.**  $\diamond_\Delta$  is “there exists a position”. Since  $\exists$  does not distribute over conjunction, ingredients may occur at different positions without co-occurring.

*Remark 3* (Degenerate equality cases). If  $\Delta = 0$  or  $|l| \leq 1$ , then  $\diamond_\Delta T(l) = T(\diamond_\Delta l)$ . Strict separation requires  $\Delta \geq 1$  and at least two ingredients. A sufficient condition for commutation along a trajectory (within-window persistence) is given in the Supplementary Material.

*Remark 4* (Reading the theorem back into window semantics). In the sliding-window case, Definition 10 corresponds to

$$\begin{aligned} \text{Occur}_{W_\Delta}(l, \tau, t) &\iff \tau_\Delta(t) \in T(\diamond_\Delta l), \\ \text{CoInst}_{W_\Delta}(l, \tau, t) &\iff \tau_\Delta(t) \in \diamond_\Delta T(l). \end{aligned}$$

Theorem 3 is exactly the claim that ingredient-wise satisfaction is strictly weaker than co-instantiation in general.

## Chord and Arpeggio Postulates

The above algebra does not decide which window semantics corresponds to phenomenality. I now state Chord and Arpeggio in terms of occurrence and co-instantiation, with a higher-layer moment statement and its base-layer grounding.

**Assumption 1** (Subjective moments live at some layer). Fix a stack state with vocabularies  $\mathfrak{v}^0, \dots, \mathfrak{v}^m$ . Assume a candidate subjective moment is represented by a conjunctive statement  $l^m \in \mathcal{L}_{\mathfrak{v}^m}$ .

**Definition 14** (Phenomenal realisation predicate). Let  $\text{PhenReal}(l^m, \tau, t)$  mean “moment statement  $l^m$  is phenomenally realised at (higher-layer) time  $t$  along objective trajectory  $\tau$ .” I treat  $\text{PhenReal}$  as primitive, constrained by Chord/Arpeggio.

**Intuition.**  $\text{PhenReal}$  names the claim that moment  $l^m$  is experienced at subjective time  $t$  along run  $\tau$ . Here  $l^m$  can include first, second, and third order selves and causal identities for objects and properties (Bennett 2025b).

**Definition 15** (Base grounding of a moment statement). Let  $g^0 := \text{Ground}_{0 \leftarrow m}(l^m)$  be the base-layer grounding of  $l^m$  (Definition 6). By Theorem 1,  $T(g^0) = T(l^m)$ . I call  $g^0$  the *grounded conjunction* of the moment statement.

**Definition 16** (Chord (objective co-instantiation required)). Fix a windowing map for the phenomenal layer, typically the sliding window  $W_\Delta$  of horizon  $\Delta$  (Section ). Chord postulates that for all  $l^m, \tau, t$ ,

$$\begin{aligned} \text{PhenReal}(l^m, \tau, t) &\Rightarrow \text{CoInst}_{W_\Delta}(g^0, \tau, t) \\ &\text{(equivalently } \tau_\Delta(t) \in \diamond_\Delta T(g^0)\text{)}. \end{aligned}$$

If the moment is experienced, then within the corresponding subjective window there exists an objective time-slice at which the entire grounded conjunction holds.

**Intuition.** Chord says: if the moment is experienced, then within the window there is at least one objective time step where all grounded ingredients are true together.

**Definition 17** (Arpeggio (objective smearing permitted)). Fix the same windowing map  $W_\Delta$  as in Definition 16. Arpeggio postulates the following.

1. **Ingredient-wise necessity.** For all  $l^m, \tau, t$ ,

$$\begin{aligned} \text{PhenReal}(l^m, \tau, t) &\Rightarrow \text{Occur}_{W_\Delta}(g^0, \tau, t) \\ &\text{(equivalently } \tau_\Delta(t) \in T(\diamond_\Delta g^0)\text{)}. \end{aligned}$$

2. **Smearing permitted.** There exist  $l^m, \tau, t$  such that

$$\begin{aligned} \text{PhenReal}(l^m, \tau, t) \wedge \text{Occur}_{W_\Delta}(g^0, \tau, t) \\ \wedge \neg \text{CoInst}_{W_\Delta}(g^0, \tau, t). \end{aligned}$$

This allows a moment to be experienced without any instant realising the full grounded conjunction.

**Intuition.** Arpeggio says: if the moment is experienced, each grounded ingredient must occur somewhere in the window, but there need not be a single time step where they are all true together.

## Architectural Consequences

A polycomputer simultaneously executes multiple functions using the same parts or physical substrate (Bongard and Levin 2023). A liquid brain such as an ant colony computes through movement rather than persistent structure (Solé, Moses, and Forrest 2019). In contrast, a “solid brain” has persistent structure supporting, e.g., a bio-electric abstraction layer with synchronous broadcast (Bennett 2025b). I now translate “single-thread sequential emulation” versus

“synchronous polycomputational realisation” into a minimal architectural invariant bounding *simultaneous causal contribution*. The Temporal Gap asks whether that bound is consciousness-relevant (and whether conscious unity requires causal exchange between co-instantiated ingredients, explored in (Bennett 2026)).

### Concurrency Capacity as an Invariant

**Definition 18** (Contributor model). Fix  $n \geq 2$  potential contributors and write  $[n] := \{1, \dots, n\}$ . Define a base environment  $\Phi := X \times \mathcal{P}([n])$ , where  $x \in X$  is a “data” component and  $A \subseteq [n]$  is the set of currently *active* contributors. For each  $i \in [n]$  define  $p_i := \{(x, A) \in \Phi : i \in A\}$ . Let  $l_n := \{p_1, \dots, p_n\}$  and note  $T(l_n) = \{(x, [n]) : x \in X\}$ .

**Intuition.** The contributor model records which contributors are active at each time step. A conjunction mentioning many contributors can only be true when all are active simultaneously.

**Definition 19** (Concurrency capacity). An architecture has *concurrency capacity*  $c \in \{1, \dots, n\}$  if its admissible trajectories satisfy  $|A_t| \leq c$  for all objective times  $t$  (writing  $\tau(t) = (x_t, A_t)$ ). The case  $c = 1$  formalises a sequential single-thread model;  $c \geq n$  formalises synchronous poly-computing at the relevant scale.

**Intuition.** Capacity  $c$  is the maximum number of contributors active at one time step. If the grounded content needs more than  $c$  active together, co-instantiation cannot hold.

**Theorem 4** (Synchronous threshold for co-instantiation). Fix  $n \geq 2$  and content statement  $l_n$ . If an architecture has concurrency capacity  $c < n$ , then along every admissible trajectory  $\tau$  and for every horizon  $\Delta$ ,  $\tau_\Delta(t) \notin \diamond_\Delta T(l_n)$  for all  $t$ . However, if every contributor is activated at least once within some horizon- $\Delta$  window, then  $\tau_\Delta(t) \in T(\diamond_\Delta l_n)$ . Thus the architecture admits ingredient-wise satisfaction but forbids objective co-instantiation.

*Proof.* If  $c < n$ , then  $|A_t| \leq c < n$  at every  $t$ , so  $A_t \neq [n]$  and  $\tau(t) \notin T(l_n)$  for all  $t$ . No window contains an element of  $T(l_n)$ , so  $\diamond_\Delta T(l_n)$  is never satisfied. For the second claim, if each  $i$  appears in  $A_{t+k_i}$  for some  $k_i \leq \Delta$ , then  $\sigma = \tau_\Delta(t)$  satisfies  $\sigma \in \diamond_\Delta p_i$  for all  $i$ , hence  $\sigma \in T(\diamond_\Delta l_n)$ .  $\square$

**Proof intuition.** If at most  $c$  contributors can be active, an  $n$ -way conjunction with  $n > c$  is never true at an instant, though each ingredient can appear somewhere in a window.

**Corollary 1** (Single-thread CPU vs synchronous polycomputer). Let a moment statement  $l^m$  have base grounding  $g^0$  whose ingredients include a conjunctive substatement equivalent to  $l_n$  for some  $n \geq 2$ . Fix the phenomenal windowing map  $W_\Delta$  used in Chord/Arpeggio.

1. Under Chord, any sequential-update architecture with capacity  $c < n$  cannot phenomenally realise  $l^m$  (Theorem 4).
2. Under Arpeggio, the same sequential architecture remains a candidate, because  $T(\diamond_\Delta g^0)$  may hold without  $\diamond_\Delta T(g^0)$ .

3. Any synchronous architecture with capacity  $c \geq n$  can satisfy  $\diamond_\Delta T(g^0)$  by activating the required contributors simultaneously, hence can host the moment under both postulates.

**Remark 5** (Connection to valence-first Stack Theory). My valence-first application of Stack Theory treats certain phenomenal configurations as synchronised, causally efficacious conjunctions over many ingredients (Bennett 2025b). The capacity model isolates one minimal implementation parameter—simultaneous contribution—controlling whether such conjunctions can be instantiated at any objective instant versus only “smeared” across a window.

### Evidence for Chord

Chord and Arpeggio predict different empirical signatures *within* a short subjective window. Either consciousness requires window-local coordinated co-engagement, or it can tolerate mere ingredient-wise activation distributed across the window. Several classic paradigms are more naturally described by the coordination view. In masking, conscious perception correlates with transient *long-range* gamma phase synchrony across cortical areas, even when local activity is similar across seen versus unseen stimuli (Melloni et al. 2007). In non-REM sleep, TMS-EEG responses become strong yet *local*, failing to propagate, consistent with breakdown of effective connectivity when consciousness fades (Massimini et al. 2005). Perturbational measures such as PCI track level of consciousness across wake/sleep/anesthesia and disorders of consciousness, consistent with a requirement for integrated, differentiated dynamics (Casali et al. 2013). These findings do not *prove* Chord, but they make the existential permissiveness of Arpeggio less compelling. Conscious level covaries with the presence of a temporally coordinated, system-level regime. I therefore treat Chord as the default working hypothesis; this step remains defeasible.

### Attribution and Implications

The Temporal Gap is the question of whether consciousness is affected by the *gap between ingredient coverage and co-instantiation* for a grounded conjunction.

Fix a grounded conjunction  $g^0$  and an objective trajectory  $\tau$ . Define the minimal horizon needed for ingredient coverage versus co-instantiation:

$$w_{\text{ing}}(g^0, \tau) := \inf\{\Delta \in \mathbb{N} : \exists t \tau_\Delta(t) \in T(\diamond_\Delta g^0)\},$$

$$w_{\text{co}}(g^0, \tau) := \inf\{\Delta \in \mathbb{N} : \exists t \tau_\Delta(t) \in \diamond_\Delta T(g^0)\}.$$

The strict Temporal Gap pattern is  $w_{\text{ing}} < w_{\text{co}}$  (or  $w_{\text{co}} = \infty$ ). These are *instrumentable* when base-layer ingredients are defined in terms of architectural state variables. This kind of architectural metric aligns with the view that behaviour alone may be insufficient for consciousness assessment and that concrete *indicator properties* should be checked against candidate theories (Butlin et al. 2023).

A concrete test follows: construct paired systems that are behaviourally matched at a reporting layer but differ in base-layer concurrency capacity (one synchronous with  $c \geq n$ , one sequential with small  $c$ ). Select candidate contents whose

groundings include large conjunctions, and measure  $w_{\text{ing}}$  and  $w_{\text{co}}$ . Chord predicts that phenomenality tracks bounded  $w_{\text{co}}$ ; Arpeggio predicts it may track  $w_{\text{ing}}$  alone.

If Chord is even *plausible*, behavioural equivalence between a synchronous system and a sequential emulation does not settle phenomenal equivalence. This creates an ethically salient zone of reasonable disagreement about moral status, particularly when a system could have valenced experience. A precautionary stance toward sentience under uncertainty has been defended in nearby contexts (Birch 2024). One concrete recommendation is to treat large Temporal-Gap regimes ( $w_{\text{co}} \gg w_{\text{ing}}$  or  $w_{\text{co}} = \infty$ ) as a *high-uncertainty* region for attribution, motivating conservative deployment policies until the Temporal Gap is better resolved.

## Conclusion

I provided a Stack-Theory-compatible Stack-Time Semantics addendum and used it to make the Temporal Gap into a measurable algebraic and architectural question. Compositional grounding (Theorem 1) ensures any higher-layer phenomenal statement has a base-layer grounded conjunction with identical truth conditions. Induced layer time partitions objective time into maximal constant macro-blocks and can be arbitrarily sparse (Proposition 1), so subjective moments need not correspond to single objective instants. The Temporal Gap becomes a question of the non-commutation pattern

$$\diamond_{\Delta} T(l) \subseteq T(\diamond_{\Delta} l),$$

with strict inclusion in general for  $\Delta \geq 1$  and multi-ingredient statements (Theorem 3). This yields a clean separation between ingredient-wise window occurrence and objective co-instantiation.

Chord and Arpeggio are competing postulates about which notion constrains phenomenality. Section notes that conscious level covaries with temporally coordinated dynamics, tentatively favouring Chord.

A simple architectural invariant (concurrency capacity) can force the strict Temporal Gap pattern for large conjunctions (Theorem 4). Under Chord, software consciousness on a *strictly sequential* substrate is impossible for any content whose grounding requires two or more simultaneous contributors. Real machines are not perfectly sequential at the physical level, so the practical question becomes which physical concurrency scale is relevant for grounding. Consciousness according to Stack Theory requires a “tapestry of valence” supported by a solid-brained polycomputing architecture in which each co-instantiated ingredient simultaneously acts upon the others, all linked and synchronised (a condition formally included in Chord in (Bennett 2026)). Extreme concurrency capacity may be needed, and still concurrency is only one aspect of this. The central conclusion remains that consciousness attribution is hardware-sensitive, and surface-level behavioural equivalence alone cannot wash out the Temporal Gap.

## References

Baars, B. J. 1988. *A Cognitive Theory of Consciousness*. Cambridge: Cambridge University Press. ISBN 0521301335.

Bennett, M. T. 2022. Computable Artificial General Intelligence. *Under Review*.

Bennett, M. T. 2023a. Emergent Causality and the Foundation of Consciousness. In *16th International Conference on Artificial General Intelligence*, Lecture Notes in Computer Science, 52–61. Springer. ISBN 978-3-031-33469-6. OUCI metadata page: <https://ouci.dntb.gov.ua/en/works/7BoXJMW4/>.

Bennett, M. T. 2023b. The Optimal Choice of Hypothesis Is the Weakest, Not the Shortest. In *16th International Conference on Artificial General Intelligence*, Lecture Notes in Computer Science, 42–51. Springer.

Bennett, M. T. 2025a. A Formal Theory of Optimal Learning with Experimental Results. *IJCAI*, 10967–10968.

Bennett, M. T. 2025b. *How To Build Conscious Machines*. Ph.D. thesis, The Australian National University.

Bennett, M. T. 2025c. Optimal Policy Is Weakest Policy. In *Artificial General Intelligence*, volume 16057 of *Lecture Notes in Computer Science*, 43–56. Springer.

Bennett, M. T. 2025d. Technical Appendices. Archived release on Zenodo. Source repository: <https://github.com/ViscousLemming/Technical-Appendices>.

Bennett, M. T. 2026. Spacetime Bounds on Consciousness.

Bennett, M. T.; Welsh, S.; and Ciaunica, A. 2024. *Why Is Anything Conscious?* Preprint.

Birch, J. 2024. *The Edge of Sentience: Risk and Precaution in Humans, Other Animals, and AI*. Oxford University Press. ISBN 9780191966729.

Block, N. 1995. On a confusion about a function of consciousness. *Behavioral and Brain Sciences*, 18(2): 227–247.

Bongard, J.; and Levin, M. 2023. There’s Plenty of Room Right Here: Biological Systems as Evolved, Overloaded, Multi-Scale Machines. *Biomimetics*, 8(1).

Butlin, P.; Long, R.; Elmoznino, E.; Bengio, Y.; Birch, J.; Constant, A.; Deane, G.; Fleming, S. M.; Frith, C.; Ji, X.; Kanai, R.; Klein, C.; Lindsay, G.; Michel, M.; Mudrik, L.; Peters, M. A. K.; Schwitzgebel, E.; Simon, J.; and VanRullen, R. 2023. Consciousness in Artificial Intelligence: Insights from the Science of Consciousness. arXiv:2308.08708.

Casali, A. G.; Gosseries, O.; Rosanova, M.; Boly, M.; Sarasso, S.; Casali, K. R.; Casarotto, S.; Bruno, M.-A.; Laureys, S.; Tononi, G.; and Massimini, M. 2013. A theoretically based index of consciousness independent of sensory processing and behavior. *Science Translational Medicine*, 5(198): 198ra105.

Chalmers, D. J. 1995. Facing up to the problem of consciousness. *Journal of Consciousness Studies*, 2(3): 200–219.

Dehaene, S.; and Naccache, L. 2001. Towards a cognitive neuroscience of consciousness: basic evidence and a workspace framework. *Cognition*, 79(1-2): 1–37.

Friston, K. 2010. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2): 127–138.

James, W. 1890. *The Principles of Psychology*. New York: Henry Holt and Company.

Massimini, M.; Ferrarelli, F.; Huber, R.; Esser, S. K.; Singh, H.; and Tononi, G. 2005. Breakdown of cortical effective connectivity during sleep. *Science*, 309(5744): 2228–2232.

Melloni, L.; Molina, C.; Pena, M.; Torres, D.; Singer, W.; and Rodriguez, E. 2007. Synchronization of neural activity across cortical areas correlates with conscious perception. *Journal of Neuroscience*, 27(11): 2858–2865.

Oizumi, M.; Albantakis, L.; and Tononi, G. 2014. From the phenomenology to the mechanisms of consciousness: Integrated Information Theory 3.0. *PLOS Computational Biology*, 10(5): e1003588.

Rosenthal, D. M. 1986. Two concepts of consciousness. *Philosophical Studies*, 49(3): 329–359.

Seth, A. K.; and Bayne, T. 2022. Theories of consciousness. *Nature Reviews Neuroscience*, 23(7): 439–452.

Solms, M. 2021. *The Hidden Spring: A Journey to the Source of Consciousness*. London: Profile Books.

Solé, R.; Moses, M.; and Forrest, S. 2019. Liquid brains, solid brains. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1774): 20190040.

Tononi, G. 2004. An information integration theory of consciousness. *BMC Neuroscience*, 5(1): 42.

Varela, F. J. 1999. The specious present: a neurophenomenology of time consciousness. In Petitot, J.; Varela, F. J.; Pachoud, B.; and Roy, J.-M., eds., *Naturalizing Phenomenology: Issues in Contemporary Phenomenology and Cognitive Science*, 266–314. Stanford, CA: Stanford University Press.

Webb, T. W.; and Graziano, M. S. A. 2015. The attention schema theory: a mechanistic account of subjective awareness. *Frontiers in Psychology*, 6: 500.