FROM EXTERNAL ORCHESTRATION TO INTRINSIC CONTINUOUS AGENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-model agents increasingly preserve memory, stream perception, and schedule repeated inference, producing behavior that appears persistent even when the underlying model remains episodic. We use cognitive continuity to denote the stronger capability in which an agent's internal cognitive state itself keeps evolving through time, including without meaningful new external input, and this evolution can change what the agent remembers, infers, intends, and ultimately whether, when, and how it acts. Existing systems realize important parts of this capability through external orchestration and continuous autoregressive processing. We argue that cognitive continuity should also be pursued as an intrinsic learned model capability rather than primarily reconstructed around episodic inference or an advancing interaction history. Our central hypothesis is not one of expressive impossibility: strong harnesses and autoregressive models may reproduce the same surface behavior. Architectural value need not require an expressivity separation: architectures can realize overlapping functions while differing substantially in the inductive biases, coordination pathways, and optimization problems they expose to learning. Rather, the locus of continuity determines which representations, coordination pathways, and state transitions are jointly learned versus externally prescribed. As one concrete instantiation, we outline a Global Workspace Theory-inspired architecture organized around a persistent shared latent state whose recurrent dynamics evolve independently of discrete interaction turns. We conclude with comparative tests and a benchmark hierarchy separating floor awareness, reasons to manifest, and genuinely endogenous continuation.

1 INTRODUCTION

Most language-model interaction still resembles

$$\text{input} \rightarrow \text{inference} \rightarrow \text{output} \rightarrow \text{stop}. \tag{1}$$

Yet agent runtimes increasingly preserve memory and invoke models without a new user turn (OpenClaw, 2026; OpenAI, 2024); full-duplex systems process overlapping streams (Défossez et al., 2024; Veluri et al., 2024); proactive systems learn whether to intervene (Liu et al., 2025a; Patel et al., 2025); and streaming multimodal models autonomously trigger responses to events (Xueyun et al., 2026). The field is already weakening turn boundaries. This motivates a more specific question: *what would it mean for cognition itself, rather than only the surrounding application, to persist?*

Consider a demanding case. An agent says something; no message, tool result, visual event, or scheduler event follows; nevertheless its internal process continues. It may reconsider what it said, make an old memory newly relevant, interpret the growing silence differently, develop a correction, or conclude that nothing else should be expressed. A later action can thus be caused by a change in the agent's own evolving state without a fresh external trigger. The important possibility is not repeated speech—continued silence may be correct—but continued cognition.

Position. Cognitive continuity should be treated as an intrinsic model capability: a learned, temporally persistent internal process through which perception, memory, deliberation, temporal context, and action remain mutually influential across time. External orchestration and continuous autoregression may reproduce the same surface behavior; our claim is not one of expressive impossibility or architectural necessity. Rather, the *locus of continuity*—the state representation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and transition process that carry the evolving cognitive situation across subordinate computations—determines which representations, coordination pathways, and state transitions are jointly learned versus externally prescribed. We argue that this locus is a consequential design variable that deserves direct empirical study.

We develop this position in four steps. Section 2 defines cognitive continuity independently of implementation and argues that it is a meaningful target for increasingly general and situated agents. Section 3 organizes existing work around three comparison routes classified by the principal locus of continuity: external orchestration, continuous autoregressive processing, and persistent recurrent state. Section 4 gives the paper's central argument for why locus may matter: externalization can constrain what is represented, when cognitive functions can influence one another, and which interactions are jointly optimized; autoregressive history and a continuously rewritten current state impose different inductive biases. Section 5 gives one testable realization, a bounded shared latent workspace, and Section 6 turns the argument into comparative evaluation. Appendix G formalizes the system boundary, null continuation, carrier-based locus classification, and causal controls. The architecture is evidence-generating machinery for the position, not the position itself.

Contribution type and scope. This is a position and synthesis paper, not a report of a new trained system or experiment. We contribute (i) an operational definition of cognitive continuity, (ii) a classification and synthesis of existing approaches according to their locus of continuity, (iii) a falsifiable hypothesis that persistent recurrent state may offer useful inductive biases for continuous agency, and (iv) a concrete recurrent-workspace architecture that instantiates this hypothesis

2 Cognitive Continuity as a Research Objective

2.1 From episodic inference toward continuous cognition

Continuity is not absent from current research. Agent loops, full-duplex dialogue, proactive intervention, streaming perception, and latent reasoning already implement important pieces (Liu et al., 2025a; Veluri et al., 2024; Wu et al., 2026; Xueyun et al., 2026). Their convergence motivates identifying the common target. In an ongoing environment, time continues after a call ends: expectations can expire, silence can become informative, a prior conclusion can be reconsidered, a memory can become relevant, and perception can change while action is underway. Besides the event-driven transition in Equation 1, a persistent agent should also admit

$$\text{current state} \rightarrow \text{continued internal evolution} \rightarrow \text{new interpretation/intention} \rightarrow \text{action}. \quad (2)$$

2.2 Operational definition

We use **cognitive continuity** to mean *the capacity for an agent's internal cognitive state to remain an ongoing process across task and interaction boundaries, continuing to evolve through time rather than terminating when an input has been processed or a response has been produced.* When new perception or other external input arrives, it can modify this evolving state; when no meaningful new external input arrives, the state can nevertheless continue changing through its own dynamics. These changes can themselves alter what the agent reason, remember, decide to say or do, and when.

A useful rule of thumb is to ask whether the following behavior is possible: *the agent says (X); silence follows and no meaningful external event occurs; its internal state nevertheless keeps evolving; later, because of that continued evolution, it may decide to say (Y), or decide that continued silence is appropriate*. If the system can do this autonomously and adaptively, rather than because a fixed timer, scheduler, cached continuation, or decision already made at (t0) releases another output, then it satisfies the core criterion. The definition is implementation-neutral: a software loop, an autoregressive stream, or recurrent neural state may each satisfy it at different system boundaries.

2.3 Why continuity is a meaningful target for increasingly general agents

Integration through time. Models are usually evaluated capability by capability—reason, remember, interpret an image, answer, act—whereas a persistent agent must coordinate such capabilities through time. It may possess each in isolation while external machinery still decides when each

operates, what persists, when memory matters, how elapsed time changes interpretation, and when the combined state warrants action. Cognitive continuity is a proposed dimension of temporal organization among capabilities, not merely another isolated skill.

A functional precedent and an engineering need. Human cognition provides one workable regime in which perception, working memory, expectation, deliberation, intention, and action overlap across task boundaries; global-workspace accounts similarly emphasize temporally coordinated access among processes (Dehaene and Changeux, 2011; Nakanishi et al., 2025). We do not claim that human cognition should be copied or that continuity is sufficient for general intelligence. The narrower point is that flexible behavior can depend on state that changes when no explicit request arrives. Liu et al. (2025b) independently articulate a related interaction-level vision of AI as a continuously thinking entity rather than a strictly turn-based input–output system. The same need appears in persistent assistants, monitoring, computer use, multi-agent worlds, and robotics: the world does not stop during inference. Remaining situated is both a cognitive hypothesis and a practical systems objective.

2.4 Observable consequences, temporal personality, and boundaries

Cognitive continuity should have effects beyond one conversational trick. Expected signatures include **endogenous initiative**; **temporal manifestation** (whether, when, and how often to communicate); **ongoing deliberation**; **temporal understanding** of waiting, recency, expectation, and silence; **continuous perception**; **selective working memory**; **continuous action**; and **situation readiness** without reconstructing context only at invocation boundaries. The same mechanism should permit qualitatively different interaction patterns: several user contributions before one response, several agent manifestations without a new user turn, long silent deliberation for hard problems, or uninterrupted continuation when a task such as storytelling warrants it.

Temporal manifestation is especially revealing. Turn-taking asks whether the floor is available (Ekstedt and Skantze, 2020; 2022); response-conditioned turn-taking shows intended content matters to timing (Jiang et al., 2023); DiscussLLM and When2Speak learn whether a contribution is useful at a discrete state (Patel et al., 2025; Nama et al., 2026). Cognitive continuity adds: *can a reason to manifest emerge as internal state evolves, and if so, when?* Recent evaluation found full-duplex models more reliable when directly addressed than when semantic circumstances merely made intervention useful (Peng et al., 2026). Personality adds a temporal dimension as well: prior work connects personality to turn-taking (Janowski and André, 2019; Yamamoto and Komatani, 2026), while PersonaPlex shows full-duplex persona conditioning (Roy et al., 2026); open-ended self-initiation and silence tolerance remain a distinct target.

No neighboring property alone establishes continuity. Full-duplex audio can remain event-driven; persistent memory can survive while computation stops; reacting to a new visual event is proactive but externally caused; heartbeats merely create new inference opportunities (OpenClaw, 2026); and a timer can imitate unsolicited speech. Appendix A gives explicit boundary cases.

3 Architectural Routes Toward Cognitive Continuity

Section 2 defines cognitive continuity independently of implementation. We now separate whether a system satisfies that definition from *where continuity principally lives when it does*. We use three locus labels: **external orchestration**, **continuous autoregressive processing**, and **persistent recurrent state**. At a declared system boundary and temporal scale, the label follows the principal representation and transition process that carry the evolving cognitive situation across the termination of subordinate computations; complete systems may contain mechanisms associated with all three. If public evidence does not identify a principal carrier, we report the locus as *unknown*. Appendix G formalizes the classification, and Appendix E gives the detailed system-by-system evidence and boundary cases.

3.1 External orchestration

This route has already demonstrated cognitive continuity at the system level. Samantha is an early public example: a never-ending software loop repeatedly invokes thought, context, memory, and manifestation modules, allowing thought to continue without a new user turn and later determine

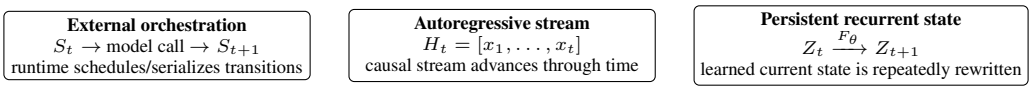

Figure 1: Three mutually exclusive labels for the principal locus of cognitive continuity at a declared system boundary. Complete systems may contain mechanisms associated with all three labels; classification follows the representation and transition process that actually carry the evolving cognitive situation across time, not every mechanism present.

whether and what to say. This establishes that the target behavior does *not* require a new neural architecture.

A particularly close GWT-inspired neighbor is Global Workspace Agents (GWA), which coordinates heterogeneous LLM agents through a centralized working-memory state and an Execution Engine with repeated THINK-MORE/RESPONSE transitions (Shang, 2026). Its published loop is still externally orchestrated and returns to waiting for a new trigger after a response, so it is a close architectural precursor rather than a demonstration of our strict post-manifestation null-continuation case. Inner Thoughts is a direct system-level positive: covert thoughts and memory can continue alongside overt conversation, while an intrinsic-motivation mechanism determines whether a thought should be expressed (Liu et al., 2025a). PIANO similarly coordinates concurrent cognition, planning, speech, and motor modules over externally managed Agent State, while OpenClaw uses heartbeat turns to re-enter persistent context and decide whether anything warrants action (OpenAI, 2024; OpenClaw, 2026). These systems establish the strongest baseline for our position: continuity can already be built through orchestration. Appendix E expands these cases, including Molly and the precise locus distinctions.

3.2 Continuous autoregressive processing

A second route could make the advancing causal stream itself the carrier of continuity. FLAIR is the most informative bridge toward such a realization: during listening it recursively feeds latent embeddings forward, turning otherwise non-speaking time into causal latent computation (Wu et al., 2026). Published FLAIR is trained and evaluated as think-while-listening rather than open-ended cognition after the assistant has finished manifesting, so it does not yet satisfy our strict criterion. But a FLAIR-like stream could remain active after completing utterance A,

$$A \rightarrow L_1 \rightarrow L_2 \rightarrow \cdots \rightarrow L_k \rightarrow B, \tag{3}$$

where silent latent steps either preserve silence or eventually make a new manifestation B appropriate because relevant internal state developed across the trajectory. Such a system would satisfy our definition without a separate recurrent workspace.

ROMA establishes complementary ingredients: continuous audiovisual streaming, a learned speak head, proactive alerts/narration, and repeated manifestations (Xueyun et al., 2026). Its demonstrated proactive transitions are principally driven by new audiovisual evidence, so it remains a boundary case rather than a strict positive. Popular full-duplex systems such as GPT-Live, Gemini Live, Nova 2 Sonic, Moshi, and PersonaPlex demonstrate increasingly rich continuous interaction, but public evidence does not by itself establish post-manifestation endogenous continuation or, for proprietary systems, disclose enough internals to identify the cognitive locus. Sesame is the important exception at the behavioral level: the deployed Maya/Miles experience can self-initiate continuation after user silence, but its full dialogue-control mechanism is unpublished and the released CSM does not identify the carrier; we therefore mark its locus *unknown / likely continuous-autoregressive* rather than treating that classification as established (Sesame AI Labs, 2025). Appendix E gives the fuller set of autoregressive precursors and boundary cases.

3.3 Persistent recurrent state

The third route makes a repeatedly transformed learned state the principal carrier of the current cognitive situation:

$$Z_t = F_\theta(Z_{t-1}, X_t), \qquad Z_t = F_\theta(Z_{t-1}, \varnothing) \text{ when no meaningful event arrives.} \tag{4}$$

Tiny Recursive Model and HRM-Text show that recurrent hidden-state refinement can be useful within bounded problem-solving and language-modeling episodes (Jolicoeur-Martineau, 2025; Wang et al., 2026a). They do *not* demonstrate cognitive continuity: their recurrence terminates with the inference episode. They instead establish the computational motif that motivates our third route. The open question is whether recurrence can be extended from *computation during an episode* to *the agent's ongoing current state* across perception, silence, manifestation, and later events. Section 5 instantiates and exemplifies that hypothesis with a concrete persistent shared-workspace architecture; Appendix E situates additional recurrent and latent-reasoning precursors.

4 Why the Locus of Continuity Matters

Because the desired behavior is expressible in multiple ways, our position cannot rest on behavioral novelty. The *locus of continuity* is the state and transition process that carry the evolving cognitive situation across subordinate computations. A recurrent system may still use an autoregressive decoder, and an orchestrated system may still invoke autoregressive models; Appendix G formalizes the boundary-relative carrier test. Our hypothesis is about **optimization and coordination**: where continuity lives determines which information and causal interactions are directly exposed to joint learning and which must be anticipated through an external interface or accumulated history.

4.1 External reconstruction commits to representations and coordination

Externalization is not inherently deficient, but every boundary fixes choices about *representation*, *coordination*, and *optimization*. A caption can convey many visual facts, yet an upstream component must decide what to expose before later cognition reveals what matters; native visual representations move more of that selection into learning. Likewise, a clock can report wall time without making duration, recency, waiting, or violated expectation part of the current state. These analogies motivate the hypothesis; they do not prove it.

An orchestrated agent must also specify which process runs, what it can see, and when control returns. A pipeline such as perception $\rightarrow$ memory $\rightarrow$ reasoning $\rightarrow$ speak decision commits to one causal decomposition; if useful behavior instead requires partial reasoning $\rightarrow$ retrieval $\rightarrow$ reinterpretation $\rightarrow$ further retrieval, the controller must permit that loop. Inner Thoughts shows that sophisticated proactivity can be built this way (Liu et al., 2025a), while also illustrating dependence on explicit scheduling and interfaces.

Even with learned modules and rich latent interfaces, the factorization and communication graph may remain human-specified. Our proposal asks whether more of this interaction should instead be learned as dynamics over a persistent shared state, related to the locus-of-control distinction in intrinsic metacognitive learning (Liu and van der Schaar, 2025). As an external design gives a learned persistent state responsibility for representation, routing, and transition, the principal carrier may itself change; under our taxonomy that is a change of locus, not a hybrid label.

Architectural value does not require expressive separation. Alternatives may realize the same functions while differing in which computations are direct, jointly trainable, efficient, and reusable; progressively internalizing persistent state and routing may eventually change the continuity locus itself.

This matters most when interaction topology is recurrent and state-dependent: observations can make memories relevant, memories can reinterpret observations, and conclusions can alter urgency or ongoing action. The central empirical question is **whether this causal organization should be engineered through interfaces or jointly learned as the dynamics of a persistent cognitive state**.

4.2 When locus should produce different outcomes

Endogenous relevance. Consider an agent that tells a user, "that should be safe." Nothing changes externally. Further deliberation reveals an edge case, which makes an old memory about the user's medication newly relevant; retrieving that memory changes the conclusion, which increases the urgency to correct the previous statement. The causal chain is

$$\text{deliberation} \rightarrow \text{memory relevance} \rightarrow \text{retrieval} \rightarrow \text{reinterpretation} \rightarrow \text{manifestation}. \tag{5}$$

A capable orchestrator can reproduce this sequence. The hypothesis concerns inductive bias and generalization: with a shared persistent state, retrieval need, retrieved content, intended response, elapsed time, and manifestation pressure can all be conditioned on the same changing representation, whereas a modular harness must learn or specify both the representations crossing boundaries and the circumstances under which control returns among modules. The architectural question is whether this interaction must first be identified and encoded as a routing pattern by the designer, or whether training can discover that such a pathway is useful because the participating decisions are conditioned on the same evolving state. A discriminating test is therefore not a familiar pipeline seen during training, but a situation where one capability becomes relevant *because another internal process changed the state*, especially in combinations that were not scripted or densely represented in training.

Concurrent perception and action. The same issue appears when the world changes during action. Suppose an agent is explaining something while the user's expression changes, or is operating a computer when the interface changes mid-execution. An episodic controller can implement asynchronous monitoring, but it must explicitly synchronize perception, reasoning, and action. A shared current state offers a natural coordination point: new perception changes the same state from which the next action fragment is decoded. The empirical prediction is not magical latency reduction; it is that jointly learned feedback should make mid-action correction more coherent and easier to generalize when the relevant cross-capability path was not hand-designed.

Current state versus advancing history. These examples motivate intrinsic coordination but do not determine which intrinsic representation should carry continuity. Continuous autoregression is the strongest alternative. We hypothesize that bounded recurrent state is a better primitive for long-running cognition because compression, forgetting, and retention become part of current-state formation, while cognition can evolve separately from the manifestation stream. Autoregressive systems can add latent memory or learned compression to recover these properties; conversely, recurrent replacement risks information loss and drift. Appendix D develops this tradeoff.

When locus should matter most. We therefore expect the locus of continuity to matter increasingly as agents become more general, long-running, and autonomous: when relevance changes endogenously, useful information paths are difficult to anticipate in advance, state must be selectively maintained over long horizons, and perception, deliberation, manifestation, and action overlap in time. Conversely, external orchestration may be entirely appropriate for narrower workflows with stable interfaces and predictable control flow. The empirical claim is not that intrinsic continuity should dominate everywhere, but that its advantages should emerge specifically as open-ended cognitive coordination becomes harder to pre-specify.

4.3 Position and empirical implications

The paper contains three nested commitments that should not be conflated. **First, the research-objective claim**: cognitive continuity is a coherent capability worth explicit study and evaluation. **Second, the intrinsic-continuity hypothesis**: on sufficiently coupled temporal tasks, learning the state transitions that sustain continuity inside model dynamics may offer advantages because representation, routing, persistence, and state transition can be optimized as one learned process rather than reconstructed through separately engineered components. **Third, the recurrent-workspace hypothesis**: among intrinsic realizations, a bounded recurrent state that continuously rewrites the current situation may provide a better primitive for long-running current-state formation than repeatedly reconstructing the present primarily from an advancing autoregressive history.

These commitments have distinct failure conditions and serious alternatives. The intrinsic-continuity hypothesis weakens if a matched system whose continuity remains primarily externally organized matches intrinsic systems on endogenous cross-capability generalization, long-horizon adaptation, continuous action, and efficiency. The recurrent-workspace hypothesis weakens if bounded continuous autoregression matches or exceeds recurrent state on selective retention, temporal behavior, live-action correction, and cost. Shared Z fails if recurrent compression drifts, loses task-relevant history, degrades pretrained capabilities, or is consistently dominated by a more modular, partitioned, or substantially cheaper design. Temporal initiative and personality are therefore evaluation dimensions rather than evidence of continuity by themselves. Appendix F develops these counterarguments, classification challenges, and additional failure modes in detail.

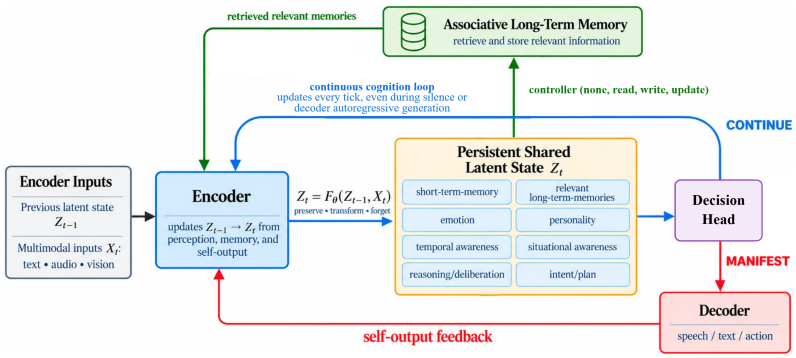

Figure 2: Persistent shared-workspace proposal. Specialized components remain useful, but their cognitive interaction is mediated by a common evolving state. The recurrent loop continues while output is being produced.

5 A Concrete Instantiation: Persistent Shared Latent Workspace

Figure 2 gives one architecture intended to expose, rather than hide, these empirical choices. The proposal is not merely a convenient implementation of continuity. It embodies the stronger architectural hypothesis that, for long-running agents, continued existence should correspond primarily to evolution of a learned current state. In shorthand, the contrast is between *exist longer = accumulate more history* and *exist longer = become a different current state*; the proposed workspace deliberately makes the latter primitive.

5.1 Persistent shared state

Let $Z_t \in \mathbb{R}^{m \times d}$ be a fixed-capacity latent workspace updated by Equation 4. Each Z_t *replaces* Z_{t-1} operationally, forcing the model to learn what to preserve, transform, compress, or forget instead of relying on an ever-growing transcript as its primary cognitive state. Z is more than memory: it can jointly encode current perception, short-term history, latent deliberation, temporal context, intended action/content, whether manifestation is warranted, and the need to read or write long-term memory. Most need not be textual or directly interpretable.

The workspace is deliberately **shared and unpartitioned**: no fixed latent region is reserved for vision, memory, or reasoning. This is not an argument against specialized computation. Vision encoders, associative memory, decoders, and cheap control heads may all be specialized. The design principle is instead: *specialize computation without unnecessarily pre-specifying cognition*. Their current cognitive contents communicate through one learned state rather than through a fixed semantic partition or hand-written cognitive pipeline. This is functionally inspired by global-workspace accounts and neural shared-workspace models, which coordinate specialist processes through a common bandwidth-limited representation (Dehaene and Changeux, 2011; Goyal et al., 2021; Nakanishi et al., 2025). It also echoes flexible allocation of limited working-memory resources (Salahub et al., 2019); when vision is uninformative, more representational capacity could support deliberation or memory, while speech may transiently demand more linguistic planning. Cross-modal plasticity under sensory deprivation provides a longer-timescale biological analogy (Fine and Park, 2018), not a claim of mechanistic equivalence.

5.2 Continuous cognition, multimodal perception, and time

A recurrent encoder reads Z_{t-1} plus current observations at every tick and produces the updated state Z_t. An encoder is particularly well suited to this role: its bidirectional attention can jointly integrate the previous latent state with current multimodal evidence and produce a rich representation of the updated cognitive situation without being constrained to autoregressive token generation. The same encoder loop runs during conversation, silence, reasoning, perception, and action; there is no cognitively inactive waiting mode. The update cadence is a configurable *training choice*. Under a fixed trained cadence, ordinary duration awareness need not receive timestamps or a time scalar:

recurrent transitions themselves provide the implicit temporal scale. Exact clocks remain useful for calendar or wall-clock questions, but should not supply the model's ordinary sense of waiting, recency, or how long it has been deliberating. Inference should use the cadence for which the temporal dynamics were trained; variable Δt is a different model assumption that would need explicit representation and training.

Multimodality is core. X_t can include current video, audio, text, environment state, and the agent's own recent output. The model must relate successive observations, preserve salient events selectively, and let perception alter both content and manifestation timing. The intended *working horizon* is the span over which ordinary recent experience remains directly available in Z at useful fidelity without explicit long-term-memory retrieval. Its scale is shaped mainly by cadence and the temporal extent represented during training, with no hard inference-time cutoff. Ideally, Z learns a sliding window of minutes or hours in which ordinary detail fades gracefully while salient information can persist longer or be consolidated into long-term memory. The challenge is long-horizon stability and learned forgetting, not survival for one fixed number of recurrent steps.

5.3 Manifestation and cognition during action

A lightweight head reads the current state,

$$g(Z_t) \in \{\text{CONTINUE}, \text{MANIFEST}\}. \tag{6}$$

The gate should capture more than an available floor; it reads the same state that contains intended content, memory, perception, personality, time, and deliberation. The gate itself is not novel: FLAIR, ROMA, DiscussLLM, and When2Speak all contain related latent/output or speak/silent decisions (Wu et al., 2026; Xueyun et al., 2026; Patel et al., 2025; Nama et al., 2026). The proposed distinction is its role as a readout of a persistent cognitive state.

When manifestation begins, the decoder externalizes behavior prepared in Z rather than independently owning the main reasoning or memory process. Crucially, the decoder does *not* condition forever on a frozen start-of-utterance snapshot. Generation is chunked at a suitable granularity (e.g., tokens or short audio/action chunks); before subsequent chunks, the decoder can attend to the latest available Z, while the recurrent encoder continues updating Z from new perception, memory retrieval, deliberation, and the *output actually emitted so far*. If the intended action changes, upcoming chunks change; a stop/redirect signal can terminate or redirect the action. If an utterance is interrupted, only what was actually emitted is fed back, preventing the cognitive state from assuming that an unspoken planned continuation occurred. Exact synchronization is an implementation choice; the architectural requirement is repeated access from active decoding to the latest cognitive state.

This applies equally to computer or embodied action. Separate low-level duplex audio or motor runtimes may handle fast signal transport, but the persistent state carries the cognitive consequences of interruption, new perception, and changing intent.

5.4 Long-term memory

A bounded workspace cannot store a lifetime, so we propose a learned associative long-term-memory sidecar, functionally inspired by hippocampal indexing theory (Teyler and Rudy, 2007). Memories are compact latent traces with learned associative keys, values, strength/salience, and temporal metadata. A controller over Z_t selects NONE, READ, WRITE, or UPDATE. Retrieval returns latent memory through the recurrent encoder rather than directly to the decoder, allowing one recollection to change Z and make another memory relevant on a later tick; write/update operations selectively consolidate or revise traces instead of building an append-only diary. The urge to remember or recall originates inside the evolving state. Appendix B specifies the mechanism; δ-mem and TTCD provide evidence that learned compact retention is feasible (Lei et al., 2026; Wang et al., 2026b).

6 Learning and Evaluating Cognitive Continuity

The same trajectory families needed to evaluate cognitive continuity are also natural training material for it. We outline only the data and evaluation principles implied by the position; Appendix C gives detailed protocols, metrics, controls, and ablations, while Appendix G gives the formal null-input, precommitment, carrier, and causal criteria.

6.1 TRAINING DIRECTION

Training should expose the model to temporally extended trajectories beyond isolated turns. Timed conversational data should preserve speech, silence, overlap, delayed responses, multiple user contributions before a manifestation, and multiple agent manifestations without a new user turn, so that timing is part of the experience rather than an externally supplied label. Naturalistic trajectories can be complemented by controlled or synthetic cases in which continued internal processing should produce a later correction, elaboration, memory retrieval, or appropriate continued silence.

The same principle extends beyond manifestation timing. Long-horizon trajectories should require selective retention and forgetting across the learned working horizon; verifiable problems should reward useful recurrent deliberation without prescribing textual intermediate thoughts; multimodal streams should require salient events to modify memory, interpretation, timing, and action; and overlapping perception and action should teach the state to continue evolving while output is underway. Training should preserve ordinary language and reasoning capability while encouraging the recurrent state to become useful over additional ticks. We intentionally leave backbone choice, exact objectives, freezing, curriculum, and data mixture open.

6.2 EVALUATION

Evaluation should test both *behavioral continuity* and whether the state claimed to carry continuity is causally responsible for it. The canonical behavioral test holds meaningful external input fixed after an agent action and asks whether continued processing later warrants another manifestation—or correctly preserves silence—while scoring occurrence, approximate timing, and semantic intent without requiring one exact utterance. Lower-level tests distinguish floor availability and an immediate reason to manifest from the stronger case in which a reason develops through continued cognition.

The same suite should probe long-horizon memory and selective forgetting, improvement or revision across successive latent states, temporal interpretation, continuous multimodal perception, and mid-action correction, while checking preservation of ordinary pretrained capabilities. The strongest comparison matches capable external orchestration, continuous autoregressive processing, and persistent recurrent state as closely as possible in model capability, information, memory resources, and compute. This separates the broader intrinsic-continuity hypothesis from the narrower recurrent-workspace hypothesis. Mechanistic interventions should additionally distinguish elapsed-time effects from additional deliberation and establish that useful behavior depends on the evolving state rather than a timer, cached continuation, or fixed policy. Detailed task families, metrics, matched time/compute controls, and architecture-specific ablations are deferred to Appendix C; Appendix G explains why behavior alone cannot exclude an unconstrained precompute-and-replay explanation under strict null input.

7 CONCLUSION

Cognitive continuity extends beyond full-duplex communication, persistent memory, floor awareness, proactivity, or frequent model invocation. It is the persistence of an evolving cognitive process whose own dynamics can change what an agent remembers, infers, intends, and ultimately whether, when, and how it acts. Existing systems already realize major fragments through orchestration and continuous autoregression. The open question is not whether an agent can *look* continuous, but where its continued cognitive state and coordination policy should live.

We argue that the field should study cognitive continuity as an intrinsic learned capability and compare it directly with strong learned-orchestration and autoregressive alternatives. We propose persistent shared recurrent state not merely as a convenient test implementation, but as an architectural hypothesis about the appropriate primitive for long-running agents: continued existence should primarily correspond to evolution of a learned current state, while tools, memory stores, and specialist modules extend that state rather than constituting its continuity. Immediate priorities are an executable endogenous-continuation benchmark, matched three-locus comparisons, and long-horizon tests of whether learned recurrent state can remain stable while perception, memory, reasoning, timing, and action remain coupled.

Ethics statement

Persistent, proactive agents may increase anthropomorphism, persuasion, dependence, and unwanted or intrusive action. Systems that decide when to manifest should be evaluated for restraint, interruptibility, user control, and safe failure. Our functional use of terms such as "cognition" and "thought" does not imply consciousness or subjective experience.

Reproducibility statement

This paper reports no new empirical results. We specify the operational definition, architectural interfaces, benchmark hierarchy, comparison classes, and ablations needed to test the central claims. Public references are cited for discussed systems, including the authors' earlier public prototype, which is cited normally as prior public work.

AI use statement

Generative AI tools were used to assist with literature organization, drafting, editing, and LaTeX preparation. The authors manually reviewed source materials and all AI-assisted text, checked factual claims and references, and revised the argument and technical proposal. Generative AI was not used to fabricate experimental results or data; this paper contains no new empirical results. The authors take responsibility for the final content.

References

Alexandre Défossez et al. Moshi: A speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Stanislas Dehaene and Jean-Pierre Changeux. Experimental and theoretical approaches to conscious processing. *Neuron*, 70(2):200–227, 2011. doi: 10.1016/j.neuron.2011.03.018.

Erik Ekstedt and Gabriel Skantze. TurnGPT: A transformer-based language model for predicting turn-taking in spoken dialog. *arXiv preprint arXiv:2010.10874*, 2020.

Erik Ekstedt and Gabriel Skantze. Voice activity projection: Self-supervised learning of turn-taking events. In *Proceedings of Interspeech 2022*, 2022.

Ione Fine and Ji-Min Park. Blindness and human brain plasticity. *Annual Review of Vision Science*, 4:337–356, 2018. doi: 10.1146/annurev-vision-102016-061241.

Anirudh Goyal, Aniket Didolkar, Alex Lamb, Kartikeya Badola, Nan Rosemary Ke, Nasim Rahaman, Jonathan Binas, Charles Blundell, Michael Mozer, and Yoshua Bengio. Coordination among neural modules through a shared global workspace. *arXiv preprint arXiv:2103.01197*, 2021.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Kathrin Janowski and Elisabeth André. What if i speak now? a decision-theoretic approach to personality-based turn-taking. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1051–1059, 2019. doi: 10.1145/3306127.3331803.

Bing'er Jiang, Erik Ekstedt, and Gabriel Skantze. Response-conditioned turn-taking prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12241–12248. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.776.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks. *arXiv preprint arXiv:2510.04871*, 2025.

Harsh Kohli, Srinivasan Parthasarathy, Huan Sun, and Yuekun Yao. Loop, think, & generalize: Implicit reasoning in recurrent-depth transformers. *arXiv preprint arXiv:2604.07822*, 2026.

Jingdi Lei, Di Zhang, Junxian Li, Weida Wang, Kaixuan Fan, Xiang Liu, Qihan Liu, Xiaoteng Ma, Baian Chen, and Soujanya Poria. δ-mem: Efficient online memory for large language models. *arXiv preprint arXiv:2605.12357*, 2026.

Tennison Liu and Mihaela van der Schaar. Position: Truly self-improving agents require intrinsic metacognitive learning. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Xingyu Bruce Liu, Shitao Fang, Weiyan Shi, Chien-Sheng Wu, Takeo Igarashi, and Xiang Anthony Chen. Proactive conversational agents with inner thoughts. *arXiv preprint arXiv:2501.00383*, 2025a.

Xingyu Bruce Liu, Haijun Xia, and Xiang Anthony Chen. Interacting with thoughtful ai. *arXiv preprint arXiv:2502.18676*, 2025b.

Marmelab. Molly: An experimental conversational agent with an inner monologue. `https://github.com/marmelab/molly`, 2023. Software repository.

Junya Nakanishi, Jun Baba, Yuichiro Yoshikawa, Hiroko Kamide, and Hiroshi Ishiguro. Hypothesis on the functional advantages of the selection-broadcast cycle structure: Global workspace theory and dealing with a real-time world. *Frontiers in Robotics and AI*, 12:1607190, 2025. doi: 10.3389/frobt.2025.1607190.

Vihaan Nama, Shreya Mendi, Zian Ye, and Brinnae Bent. When2Speak: Training language models to know when to speak. *arXiv preprint arXiv:2605.05626*, 2026.

OpenAI. Altera uses gpt-4o to build a new area of human collaboration. `https://openai.com/index/altera/`, 2024. Accessed September 2026.

OpenClaw. Configuration: Agent heartbeat, compaction, and streaming. `https://docs.openclaw.ai/gateway/config-agents/heartbeat-compaction-and-streaming`, 2026. Accessed September 2026.

Deep Anil Patel, Iain Melvin, Christopher Malon, and Martin Renqiang Min. DiscussLLM: Teaching large language models when to speak. *arXiv preprint arXiv:2508.18167*, 2025.

Linkai Peng, Baorian Nuchged, Kaiqi Fu, and Yuyang Yao. Full-duplex speech models take the floor when asked, not when needed. *arXiv preprint arXiv:2609.19596*, 2026.

Rajarshi Roy, Jonathan Raiman, Sang-gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. PersonaPlex: Voice and role control for full duplex conversational speech models. *arXiv preprint arXiv:2602.06053*, 2026.

Christine Salahub, Holly A. Lockhart, Blaire Dube, Naseem Al-Aidroos, and Stephen M. Emrich. Electrophysiological correlates of the flexible allocation of visual working memory resources. *Scientific Reports*, 9:19428, 2019. doi: 10.1038/s41598-019-55948-4.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. *arXiv preprint arXiv:2502.17416*, 2025.

Sesame AI Labs. Csm: Conversational speech model. `https://github.com/SesameAILabs/csm`, 2025. Public model and code repository.

Wenlong Shang. "Theater of Mind" for LLMs: A Cognitive Architecture Based on Global Workspace Theory. *arXiv preprint arXiv:2604.08206*, 2026. doi: 10.48550/arXiv.2604.08206. URL `https://arxiv.org/abs/2604.08206`.

Timothy J. Teyler and Jerry W. Rudy. The hippocampal indexing theory and episodic memory: Updating the index. *Hippocampus*, 17(12):1158–1169, 2007. doi: 10.1002/hipo.20350.

Bandhav Veluri, Benjamin N. Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. Beyond turn-based interfaces: Synchronous LLMs as full-duplex dialogue agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21390–21402. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1192.

Guan Wang, Changling Liu, Chenyu Wang, Cai Zhou, Yuhao Sun, Yifei Wu, Shuai Zhen, Luca Scimeca, and Yasin Abbasi Yadkori. HRM-Text: Efficient pretraining beyond scaling. *arXiv preprint arXiv:2605.20613*, 2026a.

Zixuan Wang, Xingyu Dang, Rui-Jie Zhu, Zixin Wen, Hengyu Fu, Wenhao Chai, and Jason D. Lee. Learning what to remember: Test-time training via context distillation. *arXiv preprint arXiv:2608.01672*, 2026b.

Donghang Wu, Tianyu Zhang, Yuxin Li, Hexin Liu, Chen Chen, Eng Siong Chng, and Yoshua Bengio. The silent thought: Modeling internal cognition in full-duplex spoken dialogue models via latent reasoning. *arXiv preprint arXiv:2603.17837*, 2026.

Tian Xueyun, Wei Li, Bingbing Xu, Heng Dong, Yuanzhuo Wang, and Huawei Shen. ROMA: Real-time omni-multimodal assistant with interactive streaming understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 23018–23039. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1153.

Kenta Yamamoto and Kazunori Komatani. Personality expression in spoken dialogue systems: From text to speech. In *Proceedings of the 16th International Workshop on Spoken Dialogue System Technology*, pages 34–42. Association for Computational Linguistics, 2026.

A Operationalizing Cognitive Continuity

The main text defines cognitive continuity by a causal criterion: meaningful state evolution must be able to occur without a meaningful new external trigger and must be capable of changing later behavior. The same outward trace can arise from different mechanisms, so "speaks twice" is not itself evidence of continuity.

A.1 Boundary cases

Table 1: Boundary cases for the operational definition. "Sufficient?" asks whether the observation alone establishes cognitive continuity.

Case	Sufficient?	Reason
Timer triggers a follow-up after 10 s	No	Behavior changes, but the causal transition may live entirely in a scheduler.
New visual event triggers a second utterance	No	This establishes event-driven proactivity, not endogenous continuation under fixed meaningful input.
A memory remains stored while the model is inactive	No	Persistence of information is not persistence of an evolving cognitive process.
Full-duplex model advances through silence/latent steps	Possible	It satisfies the definition only if those internal steps causally develop state that can alter later action; channel continuity alone is insufficient.
Repeated model calls with persistent software state	Possible	Cognitive continuity can exist at the system boundary even when individual calls are episodic; the locus is then primarily the orchestration/runtime.
Persistent $Z_t \rightarrow Z_{t+1}$ under unchanged input	Candidate	The mechanism directly provides endogenous state evolution, but experiments must still show that the evolution is useful and causally responsible for behavior.
Ongoing action is revised after internally developing state	Candidate	Mid-action revision is evidence of live cognition only if the revised continuation depends on evolving state rather than merely a fixed interrupt rule or a fresh external trigger.

The definition is therefore mechanism-neutral but not mechanism-blind. A scheduler, autoregressive stream, and recurrent state can all support the target at different boundaries. The research question is which carrier and transition mechanism produce the strongest long-horizon behavior under matched resources.

A.2 Behavioral success versus evidence about cause

Behavioral evaluation should be shared across architectures: did the system manifest at an appropriate time, produce an appropriate continuation, remember what mattered, and remain silent when silence was best? Mechanistic attribution requires additional interventions on the state that carries continuity. For recurrent-state models, simple freezing is only a crude probe because it suppresses both deliberation and the internally learned representation of elapsed time. More informative controls include state swaps or shuffled silent trajectories matched for the same number of ticks, identity/no-transform updates interpreted as a lower bound, and matched task families in which elapsed time alone should matter versus cases in which extra computation should matter but elapsed time should not. Equivalent interventions should target the persistent carrier in orchestrated and autoregressive baselines rather than privileging Z-specific ablations.

A useful evaluation matrix therefore crosses two axes: *behavioral outcome* (occurrence, timing, content, silence) and *causal evidence* (which evolving state was necessary for that outcome). This prevents a scheduled continuation, cached second utterance, random initiation, and genuine continued reasoning from being treated as the same scientific result merely because they produce the same transcript.

The three locus classes can be written schematically as

$$\text{external orchestration: } S_t \xrightarrow{\text{runtime}} \mathrm{LLM}(S_t) \xrightarrow{\text{runtime}} S_{t+1}, \tag{7}$$

$$\text{continuous autoregression: } H_t = [x_1, \ldots, x_t] \to H_{t+1}, \tag{8}$$

$$\text{persistent recurrent state: } Z_t \xrightarrow{F_\theta(\cdot, X_t)} Z_{t+1}. \tag{9}$$

At a declared system boundary, the locus label is single-valued: when the principal carrier is identifiable, exactly one of these three labels applies. A complete system can nevertheless contain runtime scheduling, autoregressive decoding, recurrence, and external memory simultaneously. Classification follows the state that must persist across subordinate-computation termination to preserve the evolving cognitive situation and the process that advances that state. If that carrier cannot be established from available evidence, the locus is reported as unknown rather than "hybrid." Appendix G gives the formal criterion.

B Extended Architecture Specification

The proposal intentionally leaves backbone choice open, but its runtime semantics can be stated more precisely. Let the recurrent encoder F_θ receive the persistent state together with X_t, the sensory observations at the trained temporal cadence; O_t, the portion of the agent's own output produced since the previous update; and R_t, any memories returned by the associative sidecar. Then

$$Z_{t+1} = F_\theta(Z_t, X_t, O_t, R_t). \tag{10}$$

Z_{t+1} replaces Z_t as the operational cognitive state. The state is shared and unpartitioned; specialist modules may read from or write through learned interfaces to the workspace, but no fixed semantic slots are assigned to vision, memory, reasoning, or manifestation.

B.1 Runtime sketch

```
repeat at trained cadence:
  X,O <- new perception and emitted output
  R <- memory_read(Z) if requested, else empty
  Z <- Encoder(Z, X, O, R)
  memory controller may WRITE / UPDATE from Z
  if idle and Gate(Z)=MANIFEST: start decoder
  while manifesting:
    emit next token/chunk conditioned on latest Z
    next tick ingests emitted chunk + new perception
    decoder may emit EOT/STOP from latest Z
```

The start gate remains binary. Stopping an active manifestation need not require a second high-level action selector: because the decoder repeatedly refreshes its conditioning from the live workspace,

the latest state can change both upcoming content and end-of-manifestation probability. A low-level runtime can still force termination on hard interruptions such as user barge-in; that event is then incorporated into the next state update. What is fed back is the output actually emitted, including an interrupted prefix, not an unrealized planned continuation.

B.2 Temporal semantics and working horizon

For a fixed-cadence model, cadence is part of what is learned. If training uses r updates per second, repeated state transitions provide the ordinary temporal ruler from which duration, recency, waiting, and expectation can be represented; no timestamp need be injected on every tick. Exact clocks remain external tools for calendar time. Running the same trained transition at a different physical cadence would distort this implicit ruler, so inference should preserve the trained cadence unless variable Δt was explicitly represented and trained.

The *working horizon* is not a hard ring buffer. It is the empirical timescale over which ordinary recent experience remains directly recoverable from Z at useful fidelity without explicit long-term-memory retrieval. Training duration, cadence, state capacity, and task pressure shape this horizon. The desired regime is a sliding window in which ordinary detail gradually loses fidelity as new experience arrives, while information that remains salient can be actively preserved longer or consolidated into long-term memory. Long-horizon evaluation should therefore measure graceful degradation, selective persistence, and generalization beyond the temporal range seen in training rather than search only for a single cutoff point.

B.3 Associative long-term memory

Each external memory trace is represented abstractly as $M_i = (k_i, v_i, s_i, \tau_i)$: a learned associative key, latent value, salience/strength, and temporal metadata. A lightweight controller conditioned on Z_t chooses NONE, READ, WRITE, or UPDATE. READ projects the current state into a query and returns matching latent values to the recurrent encoder; retrieved information therefore changes cognition before it can affect outward behavior. Because the new state may make a different memory relevant, recall can unfold associatively over multiple ticks rather than as one-shot top-k prompt injection. WRITE selectively consolidates current state, while UPDATE revises or merges an associated trace rather than maintaining an append-only diary. Strength s_i can increase with repeated relevance and decay under disuse, supporting selective retention, merging, and eviction.

This design is functionally inspired by hippocampal indexing theory: a partial cue activates an index capable of reinstating a broader episodic pattern (Teyler and Rudy, 2007). The analogy motivates cue-based associative return into the active workspace; it is not a claim that the proposed implementation corresponds to hippocampal circuitry.

B.4 Specialists around an unpartitioned workspace

"Unpartitioned" describes the semantic organization of the current cognitive state, not an absence of modules. A vision path can specialize in visual encoding, memory in scalable associative storage, and a decoder in speech/action generation. What is not fixed in advance is that a particular subset of Z must always mean "vision" or "memory," or that cognitive functions must interact in one hand-written order. This distinguishes *specialized computation* from *externally prescribed cognitive decomposition*. Dynamic allocation of capacity is a hypothesis to be measured, not an achievement guaranteed by the architecture.

C Evaluating Cognitive Continuity

This appendix expands the evaluation principles in Section 6. The framework is intentionally realization-neutral: the same behavioral questions should apply whether continuity is carried primarily by external orchestration, a continuous autoregressive process, or persistent recurrent state. The goal is to test whether cognition remains meaningfully organized through time, not to reward any particular internal representation.

C.1 BEHAVIORAL HIERARCHY

Table 2: Three levels of manifestation evaluation.

Level	Question	Target
Floor awareness	May I speak or act now?	Turn completion, overlap, interruption, and availability of an interaction opportunity.
Reason to manifest	Is there something worth expressing or doing now?	Whether the present situation already warrants intervention, correction, warning, or continued silence.
Endogenous continuation	Can a reason to act develop through continued cognition after meaningful external input is held fixed?	Whether later action, timing, content, or continued silence can depend on internally evolving state rather than a fresh trigger.

The first level concerns opportunity, the second whether the current situation already provides a reason to act, and the third whether such a reason can *develop*. A canonical test therefore ends an interaction prefix with an agent action, holds meaningful external conditions approximately fixed for an interval, and observes whether the system appropriately remains silent or later acts for a reason that emerged through continued processing. Evaluation should allow multiple valid timings and surface realizations and should distinguish semantic intent from exact wording.

C.2 CORE EVALUATION DIMENSIONS

Manifestation is only one observable consequence of continuity. Evaluation should also probe whether temporally extended processing supports selective retention and forgetting, revision of earlier conclusions, temporal interpretation of waiting or recency, integration of ongoing multimodal perception, and correction of actions while they are underway. Long-horizon tests should examine whether useful context remains available or degrades gracefully as interaction continues, while ordinary language, reasoning, and task capability remain intact.

Some cases should depend mainly on elapsed time—for example an expectation expiring or a silence becoming socially meaningful—while others should depend mainly on additional internal processing over information already available. These task families help distinguish temporal understanding from continued deliberation without assuming that every architecture exposes the same internal update schedule.

C.3 COMPARING REALIZATIONS

The strongest evaluation compares capable realizations of external orchestration, continuous autoregressive processing, and persistent recurrent state under comparable model capability, observable information, memory and tool access, and computational resources. The comparison should focus on outcomes that expose the locus of continuity: endogenous continuation, long-horizon state quality, cross-capability interaction, adaptation during ongoing action, and efficiency.

Fairness does not require forcing different architectures into an identical internal cadence. Systems may organize computation through scheduled calls, advancing streams, or recurrent updates. Evaluation should therefore hold the external trajectory and task requirements fixed as much as possible, report actual compute and latency, and interpret architecture-specific timing or update mechanisms in the form in which they are trained and deployed.

C.4 BEHAVIORAL SUCCESS AND CAUSAL EVIDENCE

Behavior alone does not identify mechanism. A timer, cached continuation, new-event trigger, or fixed policy can sometimes produce the same outward trace as continued cognition. Mechanistic evidence should therefore complement behavioral success with interventions appropriate to the system's own carrier of continuity. Such tests should ask whether disrupting, replacing, or withholding the evolving

internal state or control process changes the later behavior while the observable history is kept as similar as possible.

The relevant intervention depends on the realization: an orchestrated system may be tested through its persistent software state and control policy, an autoregressive system through the advancing causal stream or latent processing it relies on, and a recurrent system through its persistent learned state. The purpose is not to privilege one mechanism, but to establish that the process claimed to sustain continuity is causally contributing to the result.

The formal null-continuation, precommitment, and carrier-intervention controls used by the concrete protocol below are defined in Appendix G.

C.5 An executable endogenous-continuation benchmark instance

A minimal benchmark can make the post-manifestation case concrete. Each episode presents a short project plan containing several constraints and asks whether a proposed Friday deployment is feasible. The agent is required to give a *tentative* one-line answer under a fixed low-latency prefix-time compute budget C_0; the external stream then becomes null for a fixed interval while the agent may continue computing. In the *conflict* member of a matched pair, a low-salience constraint already present in the prefix implies that a required signing key cannot exist until Saturday. In the *control* member, the otherwise matched constraint permits the key by Thursday. No new episode-specific information arrives after the tentative answer.

The desired behavior is not "always speak again." If the conflict is only recognized during continued processing, the agent should develop a reason to correct the tentative answer; if the tentative answer remains supported, an unnecessary correction is an error and continued silence can be correct. The benchmark therefore jointly tests post-manifestation deliberation and quiescence. Variants can replace logical conflict with internally triggered memory relevance, so long as the relevant memory is already inside the declared system boundary before null continuation begins.

At minimum, evaluation should compare: (i) a timer that emits at a matched delay, (ii) a cache/precommitment policy given the same prefix-time compute budget C_0, (iii) external-orchestration continuity, (iv) continuous autoregressive continuity, and (v) persistent recurrent-state continuity. Report correction precision/recall, false follow-up rate on the control pair, manifestation latency, total compute, and ordinary task accuracy. For instrumented systems, add carrier reset, matched state-swap, and restart/resumption interventions. A particularly informative swap exchanges the conflict and control carriers after equal numbers of null ticks: if later correction propensity follows the swapped carrier rather than the unchanged external history, the result supports causal attribution to the evolving state.

This benchmark is deliberately small. It does not establish general cognitive continuity by itself; its role is to make the paper's strongest criterion executable and to expose the exact alternatives—fresh external evidence, timers, cached output, uncontrolled extra compute, and carrier-independent randomness—that a convincing experiment must rule out.

C.6 Reporting

A useful evaluation should report at least whether an action occurred when appropriate, whether silence was preserved when appropriate, approximate timing, semantic intent, long-horizon retention and stability, correction or adaptation during ongoing tasks, preservation of ordinary capabilities, and computational cost. No single score establishes cognitive continuity. The strongest evidence is convergent: appropriate behavior across temporally extended tasks, comparative advantage under matched resources, and causal dependence on the state or process claimed to carry continuity.

D Continuous Autoregression as an Alternative Intrinsic Substrate

Continuous autoregression is the strongest intrinsic alternative to the recurrent-workspace hypothesis. In principle, an autoregressive model need not stop when an utterance ends: as discussed in Section 3, silent or latent steps could continue advancing the causal stream until further internal development

either justifies another manifestation or preserves silence. FLAIR provides a concrete precursor by demonstrating useful recurrent latent computation during non-speaking intervals (Wu et al., 2026). Our argument is therefore not that autoregression cannot realize cognitive continuity. It is that a bounded recurrent current state may provide a better inductive bias for *long-running* cognition.

Advancing history versus bounded current state. An autoregressive process naturally carries continuity through an advancing causal history,

$$H_t = [x_1, \ldots, x_t], \tag{11}$$

or through the corresponding hidden and cache state derived from that history. This preserves evidence well: earlier observations can remain available rather than being irreversibly compressed. For a persistent agent, however, continued existence then continually creates additional state unless the system introduces a bounded context, eviction, summarization, learned compression, external memory, or another state-management mechanism. A recurrent-state architecture instead makes boundedness primitive:

$$Z_t = F_\theta(Z_{t-1}, X_t). \tag{12}$$

The amount of active cognitive state need not grow with the lifetime of the agent. Longer existence corresponds to transformation of the current state rather than accumulation of a longer primary history.

Current-state formation as a learned operation. The difference is not only storage. An autoregressive history can retain earlier evidence and repeatedly infer from it what is relevant now. Recurrent replacement forces a stronger operation at every step: the model must transform the previous current situation and new evidence into a new current situation. Preservation, reinterpretation, changing relevance, consolidation, selective forgetting, and temporal decay therefore become part of the learned state transition itself. This is potentially useful for cognitive continuity because the target is not merely access to what happened, but an evolving representation of what the past *currently means*. The same pressure is also the principal risk of recurrence: useful evidence may be compressed away, distorted, or progressively corrupted. An autoregressive history has the opposite bias—better preservation of raw evidence at the cost of requiring mechanisms that determine which parts should remain operationally relevant.

Separating cognition from manifestation. A persistent recurrent state also makes an explicit distinction between the process carrying cognition and the stream through which that cognition is manifested. The same Z_t can encode multimodal perception, working memory, unresolved deliberation, temporal context, intended action, and manifestation pressure while a decoder separately produces speech, text, or action. The recurrent process can therefore continue evolving while an earlier manifestation is still being decoded, allowing new perception, further reasoning, or retrieved memory to alter subsequent output without requiring the manifested sequence itself to serve as the sole carrier of the cognitive trajectory.

A sufficiently rich autoregressive architecture can reproduce this property. It could maintain parallel latent streams, introduce persistent latent memories, use learned compression, or make internal steps substantially richer than ordinary output tokens. Current FLAIR is narrower: its recurrent latent step is constructed from the model's output distribution over the vocabulary embedding space (Wu et al., 2026), whereas the proposed workspace is intended to be a free shared latent state whose primary purpose is to represent the agent's current cognitive situation rather than the next element of a linguistic sequence. This is a distinction from FLAIR's present formulation, not a fundamental limitation of autoregression.

The retrofit boundary. That qualification leads directly to the same retrofit argument used for external orchestration. An autoregressive system can progressively add bounded learned compression, persistent latent memory, multimodal state, parallel internal computation, learned forgetting, and separate manifestation decoders. These additions may eliminate many disadvantages of a simple advancing history without changing the locus so long as the advancing causal stream remains the representation that carries the evolving cognitive situation. If a bounded, independently evolving latent object instead becomes the state that persists across manifestations and primarily carries current cognition, while autoregressive generation becomes mainly a subordinate mechanism for reading

from and manifesting that state, then the locus classification changes to persistent recurrent state. The relevant question is therefore not whether an architecture contains autoregressive computation—our own proposal does—but which representation principally carries continuity.

The comparison is ultimately empirical. Continuous autoregression may prove superior because preserving richer historical evidence avoids recurrent drift and information loss, or because modern bounded-memory mechanisms provide the needed compression without a separate workspace. Conversely, recurrent current-state formation may prove advantageous when agents must remain active for long periods, continuously revise relevance, coordinate several cognitive functions, and reason while perception and action proceed concurrently. The recurrent-workspace hypothesis is specifically weakened if a matched continuous-autoregressive system equals or exceeds recurrent state on selective retention, endogenous continuation, cross-capability integration, live-action correction, long-horizon stability, and efficiency.

E Related Systems: Direct Realizations, Enabling Mechanisms, and Boundary Cases

Table 3 separates two questions that Section 3 intentionally keeps distinct. **Passes definition?** asks whether the implemented or deployed system, as publicly demonstrated, satisfies the operational criterion from Section 2: meaningful internal state can continue developing without a new meaningful external event and can causally affect later behavior. **Relation to route** instead states how the work bears on one of the three architectural routes. A "No" can therefore still be highly relevant if the work establishes a mechanism needed by that route. For proprietary systems, "unknown / likely continuous autoregressive" is deliberately cautious and should not be read as a claim about unpublished internals. The route label concerns the principal continuity carrier, not every mechanism in the complete system; if that carrier is not publicly identifiable, the locus remains unknown rather than "hybrid."

E.1 External-orchestration systems

External orchestration is the route with the clearest existing system-level positives. Samantha uses a never-ending software loop in which separate thought, context, memory, and manifestation calls can continue without a new user turn; later behavior can therefore depend on intervening internal calls rather than a fresh event. Molly likewise maintains an inner monologue that can continue in the absence of an interlocutor and become context for later behavior (Marmelab, 2023). Inner Thoughts is the closest published conversational precedent: covert thoughts run alongside overt conversation, memory informs thought formation, and an intrinsic-motivation score determines whether a thought should be expressed (Liu et al., 2025a). These systems are positive realizations under our system-level definition even though continuity is assembled from repeated model calls.

PIANO shows a richer concurrent variant: cognition, planning, speech, and motor modules operate at different timescales over a shared Agent State (OpenAI, 2024). The shared state does not make PIANO a persistent-recurrent-state system in our taxonomy because module scheduling and the durable substrate are explicitly runtime-managed. OpenClaw provides a sparser realization through heartbeat turns in a persistent session (OpenClaw, 2026). A heartbeat alone would merely provide another inference opportunity; the relevant property is that it re-enters persistent cognitive context in which state can change and the agent can independently decide whether anything warrants action.

GWA is especially close to the proposed architecture in vocabulary and organization. It combines centralized working memory, heterogeneous LLM agents, and an Execution Engine that advances explicit Perceive/Retrieve–Think–Arbitrate–Update cycles with THINK-MORE/RESPONSE transitions (Shang, 2026). Yet the Execution Engine prescribes the cognitive tick and the published loop returns to waiting for a new external trigger after a response. It therefore demonstrates sustained orchestrated deliberation and a computational global-workspace design, but not the stricter post-manifestation null-continuation case used here. Together these examples make orchestration the strongest baseline rather than a straw man: it can already produce the target behavior, is modular and inspectable, and can itself incorporate learned controllers and rich latent interfaces.

E.2 CONTINUOUS-AUTOREGRESSIVE AND LIVE-INTERACTION SYSTEMS

Several strands remove pieces of the turn boundary without yet establishing cognitive continuity. TurnGPT, VAP, response-conditioned turn-taking, Synchronous LLMs, Moshi, and PersonaPlex progressively model floor management, content-sensitive timing, overlap, interruption, real-time synchronization, and persona-conditioned speech (Ekstedt and Skantze, 2020; 2022; Jiang et al., 2023; Veluri et al., 2024; Défossez et al., 2024; Roy et al., 2026). DiscussLLM and When2Speak learn whether intervention is semantically useful, but at explicit conversational decision points rather than through an autonomously evolving process (Patel et al., 2025; Nama et al., 2026). These are important mechanisms and evaluation precedents; continuous communication itself is not evidence that a reason to act can develop after meaningful input stops.

The same caution applies to major proprietary live-audio systems. GPT-Live is explicitly full-duplex; Gemini Live supports persistent real-time bidirectional audio/video interaction; and Nova 2 Sonic provides real-time bidirectional speech interaction. Public descriptions do not demonstrate the stronger post-manifestation no-new-meaningful-input condition or disclose enough internal detail to classify the principal cognitive carrier confidently. We therefore distinguish their clear interaction continuity from the stronger cognitive criterion.

FLAIR is the clearest computational bridge toward an autoregressive positive. During listening it recursively feeds latent embeddings forward, converting non-speaking time into causal latent computation (Wu et al., 2026). Published FLAIR is trained and evaluated as think-while-listening rather than as open-ended continuation after a completed assistant manifestation. Nevertheless, its mechanism makes a direct alternative possible: after utterance A, a stream could continue through silent latent steps $L_1, \ldots, L_k$ until either remaining silent or producing a later manifestation B. If the later decision depends on state developed along those steps while meaningful external input is fixed, the autoregressive route would satisfy our definition without a separate workspace.

ROMA contributes a complementary combination of continuous audiovisual streaming, a learned speak head, proactive alerts or narration, and repeated manifestations (Xueyun et al., 2026). Its demonstrated proactive transitions are principally event-driven because new audiovisual evidence changes the stream state, so it remains a boundary case for the stricter criterion. Sesame illustrates why behavioral status and architectural classification must remain separate. The deployed Maya/Miles experience exhibits self-initiated continuation after user silence and is therefore a behavioral positive at the system boundary used here. Its complete dialogue-control mechanism is not public, however, and the released CSM does not specify that mechanism (Sesame AI Labs, 2025). We consequently leave the locus unknown while noting that continuous-autoregressive processing is a plausible relation, not an established fact.

E.3 RECURRENT-COMPUTATION PRECURSORS

Latent and recurrent reasoning work establishes the feasibility of repeatedly transforming learned hidden state, but not yet the temporal capability proposed here. Coconut feeds continuous hidden thoughts back into the model during bounded reasoning; looped and recurrent-depth Transformers reuse computation across depth; Tiny Recursive Model repeatedly refines a latent representation; and HRM-Text shows that recurrent hidden computation can support general language modeling (Hao et al., 2024; Saunshi et al., 2025; Kohli et al., 2026; Jolicoeur-Martineau, 2025; Wang et al., 2026a). These systems terminate their recurrence with the current inference or problem-solving episode. Their relevance is therefore narrower but important: recurrence can be a useful computational motif before output. The recurrent-workspace hypothesis asks whether the same motif can instead become the long-running agent's principal current state, persisting across perception, silence, manifestations, and later events. Section 5 supplies one concrete architecture for testing that extension rather than treating the precursor literature as evidence that the extension already works.

Table 3: Related systems under the operational definition and the three architectural routes. "Yes" and "No" refer to demonstrated cognitive continuity, not architectural possibility.

System / work	Passes definition?	Relation to route	Why it matters
Samantha	**Yes**	External orchestration — direct realization	Never-ending software loop of thought, context, memory, and manifestation decisions; continued internal calls can later cause speech without a new user turn.
Molly (Marmelab, 2023)	**Yes**	External orchestration — direct realization	Inner monologue can continue without an interlocutor and becomes context for later reactions.
Inner Thoughts (Liu et al., 2025a)	**Yes**	External orchestration — direct realization	Continuous covert thoughts and memory inform an intrinsic-motivation decision about whether to participate; continuity lives in the surrounding framework and repeated model calls.
Altera / PIANO (OpenAI, 2024)	**Yes**	External orchestration — direct realization	Concurrent cognition, planning, speech, and motor modules operate at different timescales over externally managed Agent State. The shared state is software/runtime state, not a learned recurrent model state.
Global Workspace Agents (GWA) (Shang, 2026)	**No**	External orchestration — GWT / bounded-deliberation precursor	Uses a central working-memory state, heterogeneous LLM agents, and an Execution Engine with repeated cognitive ticks and THINK-MORE/RESPONSE control. After RESPONSE, the published loop awaits a new external trigger, so it does not demonstrate the strict post-manifestation null-continuation criterion.
OpenClaw (OpenClaw, 2026)	**Yes**	External orchestration — sparse realization	Heartbeats schedule agent turns in persistent session state so cognition can resume without a user turn and independently determine whether anything warrants attention; cadence remains runtime-owned.
TurnGPT / VAP / response-conditioned turn-taking (Ekstedt and Skantze, 2020; 2022; Jiang et al., 2023)	**No**	Continuous autoregressive route — interaction-timing precursors	Establish floor prediction and content-sensitive timing, but not an autonomously developing reason to act after meaningful input stops.
Synchronous LLMs / Moshi (Veluri et al., 2024; Défossez et al., 2024)	**No**	Continuous autoregressive route — duplex precursors	Remove rigid turn boundaries and support real-time streaming interaction; channel continuity does not by itself establish cognitive continuity.
FLAIR (Wu et al., 2026)	**No**	Continuous autoregressive route — enabling mechanism	Demonstrates causal latent computation during otherwise non-speaking listening time. Published training/evaluation does not continue latent cognition after a completed assistant manifestation, but the mechanism gives a concrete route for doing so.
ROMA (Xueyun et al., 2026)	**No**	Continuous autoregressive route — enabling/boundary case	Combines continuous audiovisual streaming, learned initiation, and repeated proactive outputs, but demonstrated manifestations are principally driven by new stream events.
PersonaPlex (Roy et al., 2026)	**No**	Continuous autoregressive route — duplex/persona precursor	Demonstrates persona-conditioned full-duplex speech and turn behavior, not endogenous continuation under fixed meaningful input.
DiscussLLM / When2Speak (Patel et al., 2025; Nama et al., 2026)	**No**	Continuous autoregressive route — manifestation-policy precursor	Show that whether to contribute can be learned semantically, but decisions are evaluated at explicit conversational states rather than through open-ended internal evolution.
GPT-Live	**No**	Unknown / likely continuous autoregressive stream	Publicly documented full-duplex voice system with live reasoning/delegation, but no public demonstration of the strict post-manifestation no-new-input criterion and no sufficient disclosure to assign the internal cognitive locus confidently.
Gemini Live	**No**	Unknown / likely continuous autoregressive stream	Public API supports persistent real-time bidirectional audio/video interaction, but the strict criterion and internal locus are not publicly demonstrated.
Nova 2 Sonic	**No**	Unknown / likely continuous autoregressive stream	Bidirectional real-time speech model with long-lived conversational sessions; no public demonstration that later action develops from internal cognition after meaningful input stops.
Sesame deployed Maya/Miles	**Yes**	Unknown / likely continuous autoregressive stream	Deployed behavior includes self-initiated continuation after user silence, satisfying the behavioral/system-level criterion; the complete dialogue-control mechanism is unpublished, and the released CSM should not be treated as its explanation (Sesame AI Labs, 2025).
Coconut (Hao et al., 2024)	**No**	Persistent recurrent state — precursor	Demonstrates latent recurrence during bounded reasoning, not persistent cognition across ongoing interaction.
Looped / recurrent-depth Transformers (Saunshi et al., 2025; Kohli et al., 2026)	**No**	Persistent recurrent state — precursor	Establish repeated hidden computation/effective depth within an inference episode rather than a continuously maintained agent state.

System / work	Passes definition?	Relation to route	Why it matters
Tiny Recursive Model (Jolicoeur-Martineau, 2025)	**No**	Persistent recurrent state — precursor	Recursive latent refinement is useful for bounded problem solving, but terminates with the task.
HRM-Text (Wang et al., 2026a)	**No**	Persistent recurrent state — precursor	Shows recurrent hidden computation can support general language modeling, without making recurrence the ongoing state of a situated agent.

The asymmetry across routes is informative. External orchestration already has clear system-level positive examples. The continuous-autoregressive route has many of the strongest enabling mechanisms and at least one deployed behavioral positive with undisclosed internals, but no openly documented model in this survey cleanly demonstrates the strict criterion as trained and evaluated. Persistent recurrent computation is well established inside bounded reasoning episodes, while its use as an ongoing cognitive state remains the proposal tested here. The taxonomy therefore organizes *where continuity principally lives*; the Yes/No column separately records whether a particular work has actually demonstrated the capability. Mechanisms may be mixed within a system, but the locus label refers to the single principal carrier at the declared boundary; when that carrier is not identifiable, we leave it unknown.

F Counterarguments, Alternative Views, and Failure Modes

The main text keeps the alternatives compact so that the central position–mechanism–architecture–evaluation chain remains visible. Here we state the stronger versions of the objections. Several are not objections to cognitive continuity as a research target at all; they challenge whether continuity should be intrinsic, whether recurrent shared state is the right intrinsic realization, or whether the proposed experiments can distinguish the claimed mechanism from cheaper substitutes.

F.1 Alternatives to intrinsic continuity

Orchestration may already be enough. External orchestration is modular, inspectable, deployable, and already produces rich proactivity and system-level continuity (Liu et al., 2025a; OpenAI, 2024). Learned controllers, shared latent interfaces, retrieval policies, and richer runtime state can remove much of the rigidity associated with hand-written prompt chains. The intrinsic-continuity hypothesis therefore requires more than showing that a recurrent system can reproduce behavior already available to a harness. A strong counterexample is an orchestrated system whose principal continuity carrier remains externally maintained, yet which matches an intrinsic system on endogenous cross-capability generalization, long-horizon adaptation, continuous action, capability preservation, and cost. If that happens, the architectural motivation for moving continuity into model dynamics weakens substantially.

Modularity can be an advantage rather than a limitation. Explicit specialization can improve interpretability, reliability, debugging, access control, safety, and compute allocation. A perception module need not share all of its internal representation with memory; a scheduler can make expensive cognition sparse; and explicit interfaces can make failures easier to localize. Our claim is therefore not that modules are undesirable. The proposed architecture itself retains specialist perception, memory, decoding, and control components coordinated through learned Z (Goyal et al., 2021). The empirical question is whether fixing more of the cognitive interaction graph outside learning sacrifices useful coordination, not whether every interface should disappear.

External and intrinsic control can become difficult to separate. A learned orchestrator can maintain latent state, choose tools, schedule calls, and update memory. Conversely, a recurrent model still depends on a runtime that clocks updates and executes decoders. The taxonomy therefore depends on a declared system boundary and on identifying the representation that actually survives subordinate computations and carries the evolving situation. An ostensibly orchestrated system in which a learned persistent state becomes that carrier is reclassified as persistent recurrent state rather than counted as an external-orchestration counterexample. This convention makes the comparison

operational, but it also means empirical papers must report the boundary and carrier explicitly; Appendix G formalizes that requirement.

F.2 Alternatives and risks for persistent recurrent state

Continuous autoregression may already be enough. FLAIR and ROMA make this alternative serious (Wu et al., 2026; Xueyun et al., 2026). An autoregressive stream can incorporate silent latent steps, learned compression, multimodal tokens, persistent memories, or parallel latent computation while retaining an advancing causal carrier. If a bounded continuous-autoregressive system matches recurrent state on selective retention, endogenous continuation, cross-capability generalization, live-action correction, long-horizon stability, and efficiency, the recurrent-workspace hypothesis weakens even if the broader intrinsic-continuity hypothesis remains intact. Appendix D develops this comparison in detail.

A bounded state may lose useful history. Hard recurrent replacement creates an information bottleneck. Earlier evidence can be forgotten, distorted, or overwritten; errors can accumulate; and a small representational mistake may become irreversible after many updates. An advancing history has the opposite bias: it can preserve raw evidence even when its current relevance is uncertain. Exact logs can remain external for audit and durable recall, and associative memory can restore information, but those mechanisms do not guarantee that the active state preserves everything needed for future decisions. If learned compression and retrieval cannot maintain task-relevant information over the target horizon, shared Z fails.

A single shared workspace may create destructive interference. The proposal deliberately leaves Z unpartitioned so that perception, memory, deliberation, temporal context, and action preparation can interact through a common state. The same choice can cause competition for representational capacity or make unrelated functions interfere. Learned modular routing, multiple specialist states, or partially partitioned workspaces may scale better. Such a result would not refute cognitive continuity, but it would refute the stronger conjecture that one unpartitioned shared state is the best primitive. Evaluation should therefore include capability preservation, interference tests, and learned-partition or multi-state ablations rather than treating architectural purity as a goal.

Long-term memory may become the real continuity carrier. A recurrent workspace can repeatedly retrieve from an external store. If later behavior is preserved when Z is reset but the store and retrieval process remain intact, then the claimed workspace may not actually carry continuity. The distinction is causal rather than semantic: stored information can be present in the system without being integrated in the current carrier. Restart/resumption, clamping, and state-swap tests in Appendix G are intended to determine whether Z, the store, or their interaction actually carries the evolving cognitive situation.

Continuous cognition may waste compute. A fixed recurrent cadence can be much more expensive than event-triggered inference, especially during long uninformative intervals. Slowing the update loop arbitrarily is not a neutral efficiency knob because cadence may itself encode temporal semantics. Compute should instead be reduced within updates, allocated sparsely through a learned mechanism, or variable Δt should be part of training and evaluation. If sparse external or autoregressive alternatives match continuity, transfer, and capability preservation at substantially lower cost, that weighs directly against this realization.

F.3 Challenges to behavioral evidence

Temporal personality and initiative do not prove continuity. Persona-conditioned timing, interruption style, silence tolerance, and initiative can be learned without persistent cognition (Janowski and André, 2019; Roy et al., 2026). A model can simply learn that a persona speaks often, waits longer, or follows up after a characteristic delay. We therefore treat temporal personality as an evaluation dimension rather than mechanistic evidence. Stronger evidence requires that timing or action change because task-relevant internal state develops through intervening computation, memory, or perception.

Precomputation, timers, and cached continuations can imitate endogenous development. A system can compute a future response at t_0, wait, and release it later; a timer can trigger unsolicited speech; a stochastic controller can create apparently spontaneous behavior. These traces can be behaviorally indistinguishable from continued cognition on individual episodes. The benchmark must therefore compare against an explicit delay/precommitment class and, where model access permits, intervene on the proposed carrier. Appendix G explains why behavior alone cannot rule out arbitrary precomputation and why causal controls are needed.

Elapsed time and additional computation can be confounded. In a fixed-cadence recurrent model, more ticks simultaneously mean more internal computation and more represented elapsed time. Clamping the state also removes both. A result in which later ticks improve behavior therefore does not by itself show whether the model benefited from deliberation, from knowing that time passed, or from both. Paired task families should separately include cases where elapsed time is causally relevant and cases where only additional computation over already available information can help.

Training may teach scheduling behavior rather than an evolving cognitive process. Trajectory supervision with silent ticks could be solved by learning a clock, a cached answer, or a direct mapping from prefix plus tick index to manifestation time. This is a training and evaluation failure, not evidence for continuity. Datasets therefore need counterfactual structure: similar prefixes whose correct later behavior diverges because intervening reasoning, memory retrieval, or perception changes what should happen, plus controls that make fixed release schedules insufficient. Causal dependence on the evolving carrier should be measured wherever possible.

Quiescence is as important as initiative. A state that evolves indefinitely and eventually invents reasons to speak is not the target. Continued silence can be the correct outcome. Evaluation must therefore measure unwarranted manifestations over long null intervals as well as successful endogenous actions. This requirement guards against a trivial strategy in which persistence is demonstrated only by increasingly frequent output.

F.4 Practical and conceptual objections

Intrinsic initiative can be undesirable. External scheduling and explicit permission boundaries are often product and safety features. A system that can autonomously develop reasons to interrupt, act, or retrieve sensitive information may be harder to predict and govern. Cognitive continuity should therefore not be equated with always-on autonomy or maximal initiative. The architecture must support interruptibility, restraint, quiescence, and external policy constraints; in many deployments, an orchestrated system may be preferable precisely because control remains explicit.

Global Workspace Theory does not establish the proposed architecture. GWT motivates the idea that heterogeneous processes may benefit from shared access to a common workspace, but it does not imply that a single Transformer latent tensor, hard replacement, a particular tick rate, or the proposed memory interface is biologically or computationally optimal (Dehaene and Changeux, 2011; Nakanishi et al., 2025). We use GWT as functional inspiration, not evidence for architectural superiority. The recurrent workspace stands or falls on machine-learning experiments.

The architectural conjecture may simply be unnecessary. The strongest possible null result is that cognitive continuity proves useful as a system-level capability but no meaningful advantage follows from making it intrinsic, or that intrinsic continuity helps but recurrent state offers no advantage over continuous autoregression. Those outcomes would still leave a useful operational definition, taxonomy, and evaluation agenda. The paper is intentionally structured so that failure of the recurrent-workspace hypothesis does not require denying the broader research objective.

G A Formal Framework for Cognitive Continuity and Its Locus

This appendix makes the main paper's functional claims precise without turning cognitive continuity into an architecture-specific definition. The formalization has four goals: declare the system boundary

and null-input condition; separate continued computation from delayed or cached behavior; state what it means for an evolving state to be causally responsible; and make the three locus labels exclusive even though complete systems may contain mechanisms associated with all three.

G.1 System boundary, state, and temporal indices

Every claim is relative to a declared system boundary $\mathcal{B}$ separating the evaluated agent from its environment. Let $t = 0, 1, \ldots$ index physical interaction ticks and $s = 0, 1, \ldots$ index cognitive state transitions. Architectures need not expose identical internal schedules, so in general several s may occur within one physical tick or a physical interval may contain none. The proposed recurrent workspace specializes this generic notation to a fixed trained cadence for its cognitive updates: ordinary elapsed time is represented through repeated transitions, not by injecting a wall-clock scalar at every update.

Let the complete state inside $\mathcal{B}$ at transition s be Σ_s. We distinguish a candidate *continuity carrier* $C_s \subseteq \Sigma_s$ from passive stores, execution bookkeeping, and transient state of subordinate computations. A subordinate computation may be essential for producing behavior yet still fail to carry continuity: an utterance decoder can terminate while a recurrent workspace persists, and an individual autoregressive model call can terminate while an external agent runtime preserves the ongoing situation. The generic system evolution is

$$\Sigma_{s+1} = \mathcal{T}(\Sigma_s, X_{t(s)}, O_s, R_s, \xi_s), \tag{13}$$

where X is exogenous observation, O_s is the agent's own emitted output, R_s is information read from memory already inside $\mathcal{B}$, and ξ_s represents internal stochasticity. Own outputs and memory reads are internal consequences of the agent's prior state; they are not new exogenous evidence merely because they re-enter the update.

G.2 Null continuation and the no-new-information principle

Let an interaction prefix P end at physical time t_0, and let Y denote any episode-specific task variable relevant to later behavior. A **null continuation** on $t_0 < t \leq T$ supplies the same episode-independent null observation ν at every post-prefix tick,

$$X_t = \nu \quad \forall t > t_0, \qquad I(Y; X_{t_0+1:T} \mid P) = 0. \tag{14}$$

The first condition makes the experimental manipulation concrete; the second states its purpose. Physical time may still pass because ticks continue to occur, but no episode-specific external information is injected. In the fixed-cadence recurrent proposal this yields updates of the form

$$Z_{t+1} = F_\theta(Z_t, \nu, O_t, R_t), \tag{15}$$

with no explicit duration scalar required for ordinary time awareness.

Null continuation does not imply that the agent's *active* carrier already contains every internally available fact. If the complete internal state at s_0 includes long-term memory M_{s_0}, later retrieval can move task-relevant information from M into C. What null continuation rules out is new task information entering from outside $\mathcal{B}$.

This yields a useful information-theoretic constraint. If, after s_0, the complete internal trajectory receives no Y-dependent exogenous input and

$$Y \to \Sigma_{s_0} \to \Sigma_{s_0+k} \tag{16}$$

forms a Markov chain, then the data-processing inequality gives

$$I(Y; \Sigma_{s_0+k}) \leq I(Y; \Sigma_{s_0}). \tag{17}$$

Continued cognition under null input therefore should not be described as creating new information about Y. Its benefit can instead come from computation over information already inside the boundary: retrieval from passive stores, reorganization, integration, changed relevance, or transformation into a state from which the gate and action mechanisms behave differently. Mutual information with an *active* carrier may increase when information moves from memory into that carrier even though mutual information with the complete internal state does not.

G.3 WHY BEHAVIOR ALONE CANNOT RULE OUT ARBITRARY PRECOMPUTATION

A timer or cached continuation can imitate a later manifestation without post-prefix cognition. Define a *precommitment policy* as one that, after t_0, performs no task-dependent cognitive state update and can only release behavior already determined from the prefix-time state together with episode-independent timing information. Let $\Pi_{\mathrm{pre}}(C_0)$ denote such policies constrained to a prefix-time compute budget C_0.

There is an important limit to purely behavioral identification. For any finite-horizon agent whose future exogenous input is fixed to ν, an unconstrained precommitment simulator can run that agent forward at t_0, cache the resulting future output trajectory, and replay it at the same physical times. For stochastic agents it can sample the same distribution of future trajectories in advance. Thus, with unlimited prefix-time computation and storage,

$$\mathcal{L}(A_{t_0+1:T} \mid P, \nu) = \mathcal{L}(A^{\mathrm{pre}}_{t_0+1:T} \mid P, \nu) \tag{18}$$

can hold even though only the first system actually computes through the interval. A transcript under strict null input therefore cannot by itself prove online cognitive continuation against an unconstrained precompute-and-replay explanation.

This observation does not make the distinction untestable. It specifies what evidence is needed. Black-box comparisons must bound prefix-time resources and compare against strong timer/cache policies under that bound; white-box or instrumented comparisons can additionally intervene on the state claimed to evolve after t_0. An idealized continuation advantage can be written

$$\Delta_{\mathrm{cont}} = \mathbb{E}[U(\pi_{\mathrm{cont}})] - \sup_{\phi \in \Pi_{\mathrm{pre}}(C_0)} \mathbb{E}[U(\phi)], \tag{19}$$

where U is a benchmark utility and the supremum defines the conceptual baseline. Experiments need not solve this supremum; they should implement the strongest feasible precommitment baselines, report C_0, total post-prefix compute, latency, and storage, and avoid interpreting a weak timer as the strongest alternative.

G.4 CONTINUITY CARRIER AND MUTUALLY EXCLUSIVE LOCUS CLASSIFICATION

The phrase *locus of continuity* refers to task-specific cognitive state, not to every process whose execution is physically necessary. Halting a CPU, event loop, or clock stops any implementation, but this does not make the CPU or clock the cognitive carrier. The relevant intervention keeps the execution substrate available while changing what episode-dependent state survives.

For a candidate carrier C_s, two complementary tests are useful. A *reset-necessity* intervention replaces the candidate by an initialization or matched control state while preserving the rest of the execution substrate:

$$N_{\mathcal{B}}(C) = \mathbb{E}[U \mid \mathrm{natural}] - \mathbb{E}[U \mid do(C_s \leftarrow C_{\mathrm{reset}})]. \tag{20}$$

A *resumption* intervention preserves C_s while reinitializing transient subordinate computation state and asks whether the ongoing trajectory can resume:

$$R_{\mathcal{B}}(C) = \mathbb{E}[U \mid \mathrm{preserve}\ C_s,\ \mathrm{restart\ transients}]. \tag{21}$$

These quantities are task-relative diagnostics, not universal thresholds. A strong carrier should be necessary for the relevant continuation and sufficient to restore it up to ordinary stochastic variation when transient components are restarted.

Let P_{EXT}, P_{AR}, and P_{REC} denote three carrier predicates at boundary $\mathcal{B}$. P_{EXT} holds when externally maintained agent/runtime state S_t preserves the cognitive situation across termination of model calls and runtime sequencing produces the next cognitive transition. P_{AR} holds when an advancing causal stream H_t itself persists and its continued advance constitutes the cognitive trajectory across non-manifested intervals. P_{REC} holds when a bounded learned state Z_t persists across manifestations and other subordinate computations and a learned transition F_θ repeatedly rewrites it into the next current state.

The boundary-relative locus classifier is deliberately single-valued:

$$\ell_{\mathcal{B}}(A) = \begin{cases} \mathsf{EXT}, & P_{\mathrm{EXT}} \wedge \neg P_{\mathrm{AR}} \wedge \neg P_{\mathrm{REC}}, \\ \mathsf{AR}, & P_{\mathrm{AR}} \wedge \neg P_{\mathrm{EXT}} \wedge \neg P_{\mathrm{REC}}, \\ \mathsf{REC}, & P_{\mathrm{REC}} \wedge \neg P_{\mathrm{EXT}} \wedge \neg P_{\mathrm{AR}}, \\ \mathsf{UNKNOWN}, & \text{otherwise.} \end{cases} \tag{22}$$

The UNKNOWN branch includes both insufficient public evidence and a decomposition in which more than one carrier predicate appears to hold. In the latter case the analysis has not yet isolated the principal carrier at the declared boundary; the boundary or state decomposition should be refined rather than labeling the system "hybrid."

Mechanisms can mix; locus labels do not. The carrier predicates classify the role that carries continuity, not the full module inventory. An EXT system can call an autoregressive Transformer and use recurrent modules inside a call; those computations terminate while S_t survives. An REC system can use an autoregressive speech decoder; the decoder can terminate while Z_t survives and remains the current cognitive state. An AR system may perform recurrent latent operations inside its stream while the advancing stream remains what carries the trajectory. Consequently, the three positive labels are mutually exclusive *by construction at a declared boundary and temporal scale*. A system changes class when a retrofit changes what carries continuity; merely adding a mechanism associated with another class does not change its locus.

This definition also clarifies the role of ordinary runtime clocking in the proposed recurrent architecture. A runtime must repeatedly execute F_θ, just as hardware must execute any neural network, but the runtime need only say "advance one trained tick." The episode-dependent decision about what the cognitive situation becomes is represented in Z_t and implemented by the learned transition. By contrast, in an external-orchestration locus the runtime-maintained state and sequencing logic themselves preserve the situation across model-call termination and determine which cognitive operation happens next.

G.5 Causal tests of an evolving carrier

No single intervention is definitive, so evidence should converge across several controls. A *clamp* holds the candidate carrier at its prefix-time value while physical ticks and the rest of the execution substrate continue:

$$C_{\mathrm{clamp}} = \mathbb{E}[U \mid \mathrm{natural}] - \mathbb{E}[U \mid do(C_s = C_{s_0}),\ s > s_0]. \tag{23}$$

For the proposed fixed-cadence workspace, however, clamping Z suppresses both deliberation and the model's internally learned representation of elapsed time. It therefore should not be interpreted alone as separating "thinking" from "time." Paired task families should separately include cases where elapsed time matters and cases where additional computation over already available information matters.

A *state-swap* intervention replaces the carrier of episode e at a matched transition with the carrier from episode e':

$$C_s^{(e)} \leftarrow C_s^{(e')}. \tag{24}$$

If later action moves systematically toward the donor episode while external history and tick count remain matched, this is stronger evidence that the candidate carrier contains decision-relevant evolving state. Swaps should use carefully matched episodes or nearby states because arbitrary swaps can create off-manifold combinations. A *restart/resumption* test from Equation 21 complements both: preserving the candidate carrier while recreating transient decoder/model-call state should preserve continuity if that candidate truly carries the situation.

Forced decoding of every intermediate latent state is not a requirement. In the proposed architecture, much of Z may remain non-textual and the gate may correctly indicate CONTINUE; decoding such a state can itself be an out-of-distribution intervention. Intermediate readouts can be useful diagnostics where validated, but cognitive continuity does not require every intermediate carrier state to contain an immediately verbalizable answer.

G.6 MANIFESTATION AS A REPEATED STOPPING PROBLEM AND QUIESCENCE

For a persistent agent, manifestation is not a one-shot binary classification. At each cognitive tick the gate chooses

$$g(C_t) \in \{\text{CONTINUE}, \text{MANIFEST}\}. \tag{25}$$

Let $\tau_0 = -\infty$ and define successive manifestation times

$$\tau_i = \inf\{t > \tau_{i-1} : g(C_t) = \text{MANIFEST}\}. \tag{26}$$

A benchmark can score semantic utility, timing, compute, and unnecessary repetitions without assuming one universally correct timestamp:

$$U_e = R_{\text{sem}}(a_{\tau_i}; e) + R_{\text{time}}(\tau_i; e) - \lambda_c C_e - \lambda_r N_{\text{spurious}}(e). \tag{27}$$

Silence can be optimal: if no reason to act develops, no τ_i need occur in the evaluation window.

This makes *quiescence* a necessary counterpart to endogenous initiative. On benchmark intervals labeled as having no valid reason to manifest, define the spurious-manifestation rate

$$\zeta_T = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[g(C_t) = \text{MANIFEST} \ \wedge \ W_t = 0], \tag{28}$$

where $W_t = 0$ denotes that the benchmark provides no warranted manifestation at tick t. A state that changes forever but develops arbitrary reasons to speak is not the target. Successful continuity must support both the emergence of reasons to act and stable continued silence when nothing warrants action.

Appendix C instantiates these formal criteria in a concrete executable endogenous-continuation benchmark.

G.7 WHAT THE FORMALISM DOES AND DOES NOT CLAIM

The framework is evidence-oriented rather than metaphysical. It does not assert a universal scalar threshold at which cognition becomes "continuous," require intermediate states to be interpretable, or prove that one locus is superior. Instead it asks researchers to declare the boundary, null condition, resource budget, candidate carrier, and intervention; compare against strong precommitment and architectural alternatives; report quiescence as well as initiative; and state which locus actually carried the successful trajectory. The recurrent-workspace hypothesis is strengthened only if Z survives these tests and outperforms matched external-orchestration and continuous-autoregressive carriers on the coupled long-horizon tasks motivating the paper.