SAMPLE — CHAPTER 3

Continual Intelligence

The continual-learning research arc — twelve essays

Datt Goswami

continualintelligence.xyz · v2026.07 · sample

Contents

The full book: twelve essays tracing the continual-learning research arc. This sample carries Chapter 3 in full.

CHAPTER 3 OF 12 · SAMPLE

The Big World Hypothesis: Why Continual Learning Is Inevitable

Article 3 of 12 | [CL][WM] | Anchor papers: Javed & Sutton (n.d.) · Lewandowski et al. 2023 · Hafner et al. 2023 · Micheli et al. ICLR 2023 | Series: Continual Intelligence

The previous article in this series documented a crisis: deep networks trained on sequential tasks irreversibly lose the ability to learn new ones. This article explains *why* that happens — and why no fix short of a structural response will be sufficient. The reason is not a training bug. It is a mathematical property of the world.

---

§1 — The Core Intuition: You Can't Model Everything

Consider two systems that have both been trained to "play games."

The first is a chess engine. Its world is perfectly bounded: 64 squares, 32 pieces, a deterministic ruleset with no hidden state. Every legal board position can be enumerated in principle. Every outcome follows from actions taken. The engine's internal model of the game is, in a formal sense, *complete* — it can represent every state the game will ever visit. When you ask this engine to play better, the question is just "can it search deeper?" The world and the model are commensurable.

The second is a robot tasked with preparing a meal in an unfamiliar kitchen. Its world includes the spatial layout of this particular kitchen, the mechanics of this particular stove knob, the texture of this particular cutting board, the chemical properties of today's ingredients, the preferences of whoever will eat the meal, the ambient temperature, the acoustic resonance of the pan — and all the ways each of these dimensions interacts with the others across time. No finite model

will represent all of this. More crucially: **the model the robot carries will always be smaller than the kitchen**, no matter how large the robot's memory, how powerful its processors, or how long its training.

This asymmetry — between what the agent can model and what the world actually contains — is the central fact of real-world learning. It does not diminish as agents get smarter. It does not close as training data accumulates. It is a structural feature of any finite learning system operating in a physical environment. And it has a name.

> ***Key Takeaway:*** *The gap between an agent's model and the world it operates in is not a temporary engineering problem. It is a permanent mathematical property of any finite system embedded in a non-finite environment.*

---

§2 — Sutton's Manifesto: The Big World Hypothesis

In a technical manifesto that reframes the agenda for artificial intelligence, Javed & Sutton state the Big World Hypothesis as follows:

> *"The big world hypothesis says that for many learning problems, the world is multiple orders of magnitude larger than the agent. The agent neither fully perceives the state of the world nor can it learn the correct value or optimal action for each state. It has to rely on approximate solutions to achieve its goals."*

That phrase — *multiple orders of magnitude* — is doing heavy lifting. They are not saying the world is slightly bigger than the agent, or that current architectures are undersized and need to be scaled. They are saying that the gap is structural and

permanent. An agent at 10× its current size would still face a world 10× as large. The ratio does not close.

This has three immediate consequences, each of which the rest of the series follows.

First consequence: convergence to optimal is the wrong goal. Standard reinforcement learning theory asks whether an agent converges to the optimal policy. In a small world (a chess board, a grid world, a finite MDP with a stationary reward function), this question is meaningful. Optimal policies exist and can in principle be found. In a big world — a physical environment with partially observable, non-stationary, and effectively infinite state space — the optimal policy is not representable. Asking whether the agent converges to it is like asking whether a zip code can fully describe a country. The question is not malformed; it is vacuous. The correct question is whether the agent continuously improves its approximation.

Second consequence: continuous learning is not optional. If the world is bigger than any model, then the agent's model is permanently underfit. New parts of the world are always being encountered, always exceeding the model's current capacity to represent them. The agent must update continuously — not because we want it to, but because the mismatch between world and model never stops growing if the agent stops updating. Continual learning is not a feature; it is a requirement.

Third consequence: the standard benchmarks are measuring the wrong thing. [← A1: Plasticity crisis is what BWH predicts — loss of adaptive capacity in a world larger than the model's reach] [← A10: Benchmark failure is BWH failure — every benchmark that assumes a finite task set assumes a small world that the BWH says doesn't exist.] If the benchmarks assume the agent will converge to a solution on a fixed task set, they are measuring performance in a small world. Real performance — in a big world — is measured over a lifetime of never-converging, continuously improving approximation.

The BWH makes a falsifiable prediction: *agents equipped with a learned model of their local environment will outperform model-free agents as environment complexity scales, and the gap will widen with scale.* This is testable. We will return to the evidence in §5.

Figure 1 — The Embedding Hierarchy: What an Agent Can and Cannot Model

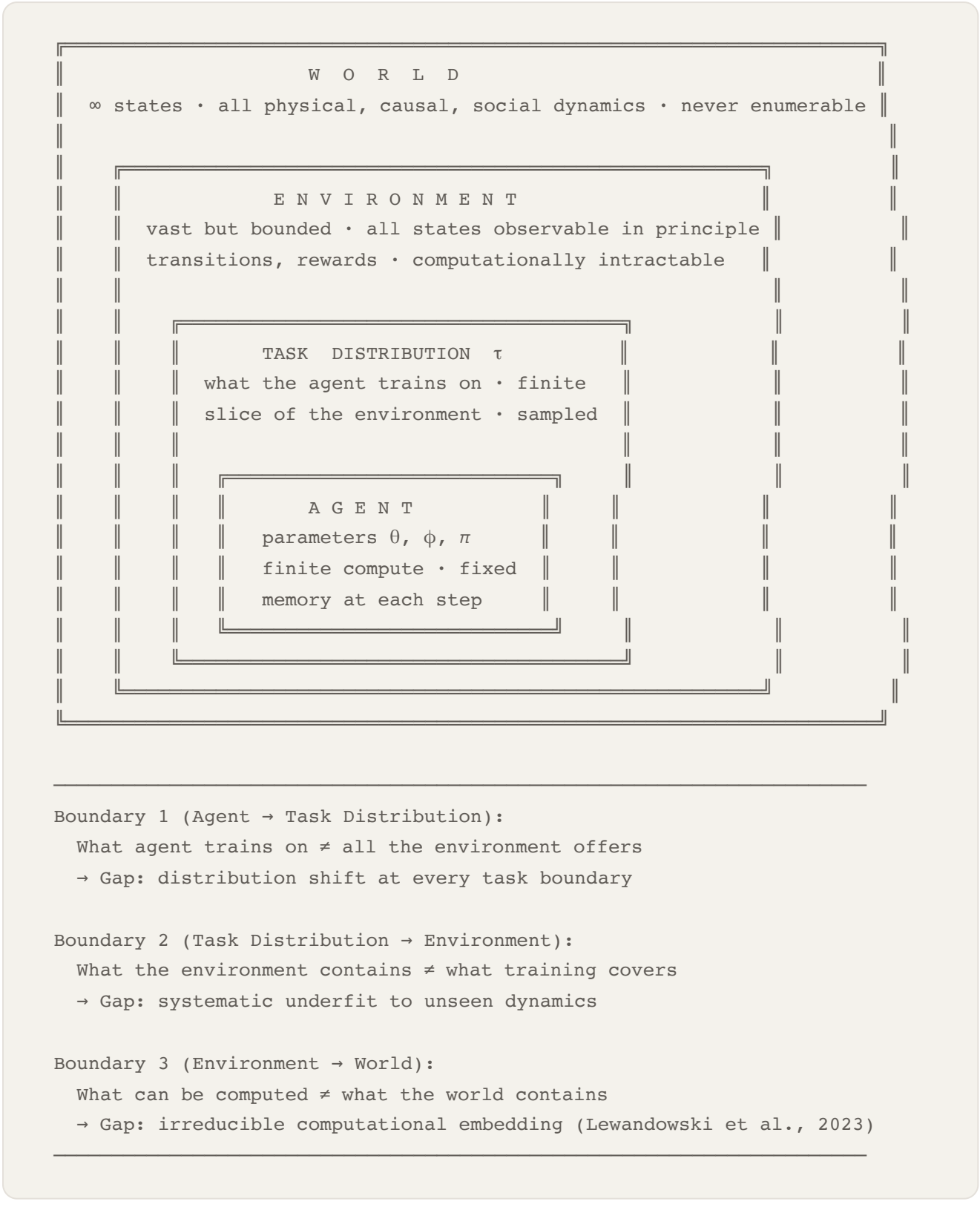

Figure 1: The embedding hierarchy of a continual RL agent. Each concentric boundary represents a gap between what the agent can represent and what the world contains. BWH formalizes Boundary 3 as irreducible: no scaling of the agent closes the gap between computational capacity and world complexity. The

agent can only reduce Boundary 1 (better generalization across its task distribution) and Boundary 2 (better environment modelling) — never eliminate them.

> ***Key Takeaway:*** *The Big World Hypothesis is falsifiable: it predicts that model-based agents outperform model-free agents as environment complexity grows, and that the advantage scales with complexity. §5 provides three empirical proofs. The hypothesis also predicts that conventional convergence-to-optimum benchmarks miss most of what matters in real settings.*

---

§3 — From Philosophy to Mathematics: The Embedded Agent

Javed & Sutton's manifesto makes a philosophical argument. Lewandowski et al. (2023) make it mathematical.

Prior formulations of the CL problem imposed *explicit* constraints on the agent: a fixed budget of parameters, a limited replay buffer size, a cap on compute per step. These constraints are well-motivated — they model the reality that agents have limited resources. But they have a structural problem: they are ad hoc. The right budget is unclear. The constraints can be loosened by scaling. And critically, an agent that satisfies the constraints only because they are imposed — rather than because the problem itself demands them — is not a general solution. It is a workaround.

Lewandowski et al. (2023) identify a deeper source of constraint: **computational embedding**. Their argument is this: an agent is embedded in its environment. It observes from inside the environment, not from outside it. Its computations take time and consume resources that the environment does not pause to

accommodate. The environment does not wait for the agent to finish thinking. The agent's computational footprint — what it can observe, compute, and update in each time step — is bounded, while the state of the environment is not.

This embedding is not a design choice. It is not a budget the experimenter imposes. It is an ontological fact about what it means to be an agent-in-environment. And its consequence is irreducible: regardless of how much computational capacity the agent has, it will always be processing a compressed, partial view of the environment. **No scaling of the agent's capacity removes the embedding constraint.** It shifts the bottleneck but never eliminates it.

The key result in Lewandowski et al.'s framework: when an agent is computationally embedded, the problem of finding a policy that is optimal given that embedding is *fundamentally different* from the problem of finding an optimal policy in the unconstrained sense. Standard RL theory — policy gradient, Q-learning, value iteration — targets the unconstrained optimum. It assumes the agent can, in principle, represent any relevant value function and update toward it with sufficient data. The computational embedding result says this assumption fails structurally in real environments: the agent's representation is incomplete by construction, and its updates are necessarily approximate.

This is not bad news. It is clarifying news. If we know the agent cannot converge to an unconstrained optimum, we can stop trying to build systems that aim for that target. Instead, we can build systems that aim for what is actually achievable: the best policy the agent can maintain given its computational embedding — updated continuously as new parts of the environment are encountered.

[→ A4: GVFs as Proto-World-Models] The formalism here — an embedded agent optimizing a constrained approximation, updated across an infinite stream of environmental encounters — is exactly the structure that the Alberta group's General Value Function architecture was designed to serve, years before the theoretical framing existed.

> ***Key Takeaway:*** *Lewandowski et al. (2023) show that the agent's resource constraints are not engineering limitations — they are an ontological consequence of being embedded in an environment. This mathematical result transforms the BWH from a philosophical assertion into a structural theorem: the constraint on agents is irreducible, and the appropriate response is continuous approximation improvement, not convergence to optimality.*

---

§4 — The Engineering Challenge: A Big World Simulator

If the BWH is correct — if real environments are irreducibly larger than any agent and continual learning is therefore mathematically necessary — then our evaluation infrastructure is built for the wrong problem. Benchmarks with fixed task sequences, episodic resets, and convergence-based metrics are measuring performance in small worlds. We have been optimizing, inadvertently, for the wrong environment class.

Kumar et al. (arXiv:2408.02930, 2024) formalize this gap as a scientific challenge: the need for what they call a **Big World Simulator**. The core demand is this: a simulation environment that is large enough, non-stationary enough, and open-ended enough that agents cannot memorize their way to good performance. A Big World Simulator must exhibit dynamics that continue to exceed the agent's current model regardless of how long the agent trains. It must produce genuine non-stationarity — not the artificial task-switching of current benchmarks, but the organic change that arises when the environment itself has its own dynamics.

This is a hard requirement. It means the simulator cannot be a finite MDP with a fixed transition function. It cannot be a fixed task sequence that the agent will eventually exhaust. It must be generative — producing new situations faster than

any fixed agent can accommodate them. And it must do so in a way that is tractable to run: infinite complexity at simulation cost is not useful.

The gap between current infrastructure and this requirement is not small. Every benchmark in the CRL literature — as documented in A10 of this series — was designed with convergence in mind. The agent trains; performance is measured; the benchmark reports a number. None of these benchmarks were designed for the setting the BWH describes: an environment that remains genuinely challenging no matter how long the agent runs. [← A10: The Benchmark Gap documents this failure in detail — no pre-2025 benchmark satisfies the non-stationarity requirement that a Big World Simulator demands.]

The closest existing system is SPIRAL (discussed in §7), which generates its non-stationarity through multi-agent self-play: as the agent improves, its opponent improves, continuously extending the frontier of what the agent must learn. This is structurally closer to the Big World Simulator requirement than any fixed-task benchmark. But it operates in a specific domain (game-playing), and whether its non-stationarity is sufficient to constitute a "big world" in Sutton's sense remains an open empirical question.

> ***Key Takeaway:*** *The BWH implies a requirement for new evaluation infrastructure — simulators that remain genuinely larger than any fixed agent across arbitrarily long training runs. Current benchmarks, by design, do not satisfy this requirement. Building a Big World Simulator is the central engineering challenge that follows from accepting the BWH.*

---

§5 — Why World Models Make CL Tractable

The Big World Hypothesis, taken alone, sounds like a counsel of despair. If the world is always bigger than the model, what can an agent actually achieve? Is the lesson just "approximate solutions forever, with no progress toward anything better"?

No. And this is the key insight that transforms BWH from a limitation into a design principle.

An agent does not need to model the whole world. It only needs to model the parts of the world it will actually encounter — the states it will visit, the transitions it will experience, the rewards it will receive. If the agent can build and maintain a *local* world model — a compressed, updatable representation of the dynamics that are relevant to its current tasks — then the BWH's unbounded complexity becomes manageable. The agent cannot close the gap between its model and the full world. But it can continuously shrink the *relevant* part of that gap: the distance between its model and the slice of the world it is currently navigating.

This is what a **world model** (φ) provides. Not a complete model of everything, but a predictive scaffold for the part of the environment the agent cares about. When new tasks arrive — when the environment shifts and the relevant slice changes — the agent can update φ to incorporate the new dynamics without discarding everything it learned before. The world model is the representational substrate that makes continual updating possible.

Figure 2 — CL Without and With a World Model: The Same Task Sequence, Two Outcomes

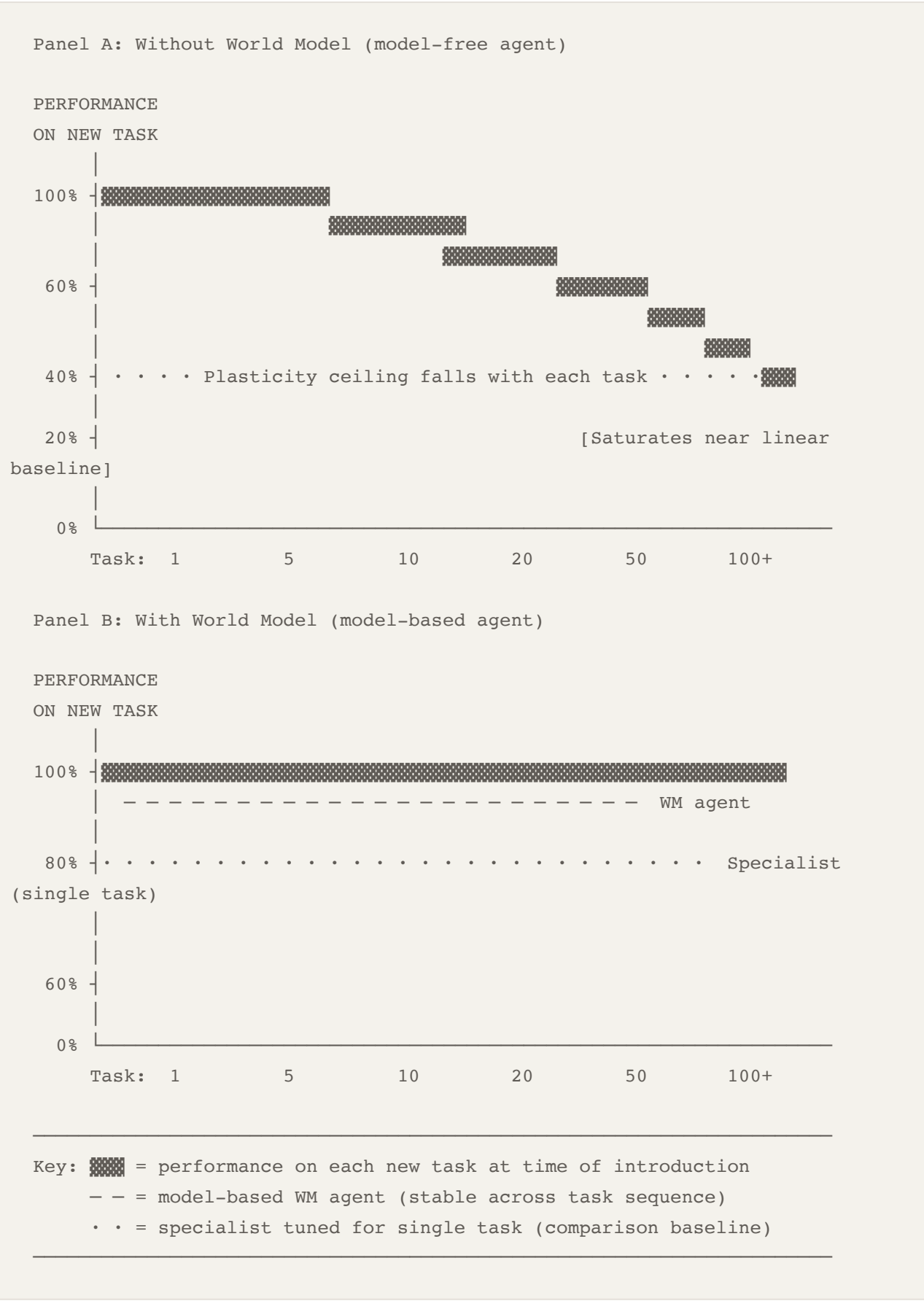
Panel A: Without World Model (model-free agent)
PERFORMANCE
ON NEW TASK
100%
60%
40%
20%
0%
Plasticity ceiling falls with each task
[Saturates near linear baseline]
Task: 1 5 10 20 50 100+
Panel B: With World Model (model-based agent)
PERFORMANCE
ON NEW TASK
100%
WM agent
80%
Specialist (single task)
60%
0%
Task: 1 5 10 20 50 100+
Key: ▓▓ = performance on each new task at time of introduction
— — = model-based WM agent (stable across task sequence)
· · = specialist tuned for single task (comparison baseline)

Figure 2: Performance on each new task across a growing sequence. Panel A (without world model): the plasticity ceiling falls monotonically — each task is learned to a lower peak than the last, eventually reaching linear-classifier-equivalent performance. This is the crisis documented in A1. Panel B (with world model): the agent maintains a stable performance ceiling across the task sequence because the world model provides a reusable representational scaffold. The world model is not a magic fix for plasticity; it is the structural reason why plasticity can be maintained.

Three engineering systems demonstrate this principle in practice today, covering the three major paradigms for building world models.

DreamerV3 (Hafner et al., 2023) is a general model-based RL algorithm that learns a world model using a Recurrent State Space Model (RSSM) — a latent dynamics model trained to predict future observations, rewards, and episode termination signals from the agent's current latent state and action. The policy and value function are trained *entirely inside this imagined model*, using imagined trajectories rather than real environment interactions.

The key result: DreamerV3 outperforms specialized, tuned algorithms across more than 150 diverse tasks — Atari, ProcGen, DMLab, Minecraft, DMControl visual observation tasks, BSuite — using a single fixed hyperparameter configuration. The algorithm is not tuned per benchmark; a single set of hyperparameters is applied everywhere. DreamerV3 is the first algorithm to collect diamonds in Minecraft from scratch without human data or predefined curricula — a long-standing benchmark in AI for requiring farsighted strategy, sparse rewards, and open-world exploration simultaneously.

This result is direct empirical evidence for the BWH prediction. A model that learns to predict local environment dynamics — and uses that prediction to train a policy — generalizes across a wider range of environments than any algorithm

optimized directly on individual tasks. The world model is not just a sample-efficiency trick; it is the mechanism that makes cross-domain transfer possible.

IRIS (Micheli et al., ICLR 2023) takes a different architectural path. Its world model is a discrete autoencoder that compresses observations into categorical latents, combined with an autoregressive Transformer that models the sequence of latent states. The policy is trained in imagination against this Transformer world model. With the equivalent of approximately two hours of gameplay in the Atari 100k benchmark — strictly limiting agent-environment interactions — IRIS achieves a mean human normalized score of 1.046, outperforming humans on 10 out of 26 Atari games. This sets a state-of-the-art for model-based methods without lookahead search on this benchmark.

The 1.046 HNS figure is significant beyond its absolute value. Human normalized score of >1.0 means the model-based agent *surpasses human performance* on average, in 100k steps, without specialized per-game tuning. The world model provides a sufficient compression of game dynamics that the agent can plan effectively in imagination rather than expending its limited interaction budget on real environment sampling. This is the world model's tractability dividend: by compressing relevant dynamics, it converts an expensive real-world query into a cheap model query.

TD-MPC (Hansen et al., ICML 2022) solves a different slice of the problem: continuous control from raw observations. Its world model is a latent task-oriented dynamics model trained jointly with a terminal value function via temporal difference learning. The key architectural choice: use the model for *short-horizon* local trajectory optimization, and use the value function to estimate *long-horizon* returns without unrolling the model to full depth. TD-MPC achieves superior sample efficiency and asymptotic performance on continuous control tasks from DMControl and MetaWorld compared to prior model-based and model-free baselines.

The TD-MPC design illustrates a general principle: a world model does not need to be perfect or globally accurate. It needs to be accurate over the horizon that matters for planning. A model that is accurate for 5-10 steps ahead is sufficient to substantially improve policy learning, even if it accumulates errors at longer horizons. The agent needs to model the *relevant* part of its world — not all of it.

Beyond these three, Guo et al. (2025, arXiv:2506.02918) demonstrate the principle in a language model agent setting. They propose dynamics modelling (DyMo) — a world model for LLM tool-use — showing that LLM agents in stateful environments benefit substantially from maintaining a predictive model of environment state transitions. Existing approaches that rely on repeated real-environment trials (test-time compute via sampling) are impractical for stateful deployment; a dynamics model converts environment queries into model queries. This is the same tractability argument, now applied to the frontier of LLM agent deployment.

Figure 3 — DreamerV3 Across 150+ Domains: WM Advantage Across All Benchmark Classes

Figure 3: DreamerV3 versus PPO (model-free baseline) across eight benchmark suites covering 150+ tasks (Hafner et al., 2023). Bar lengths show relative performance within each benchmark's scoring scale; exact scores vary by benchmark and are reported in the original paper. The critical result: DreamerV3

uses a single fixed hyperparameter configuration — no per-benchmark tuning — and still outperforms specialized methods. The world model advantage is largest on data-limited benchmarks (Atari 100k) and on sparse-reward open-world tasks (Minecraft, where PPO approaches zero). This cross-domain generalization is direct empirical evidence for the BWH prediction that model-based agents scale better than model-free agents as environment complexity increases.

> ***Key Takeaway:*** *World models make CL tractable by converting the BWH's unbounded world complexity into a manageable local prediction problem. An agent doesn't need to model everything — only the parts of the world it will visit. DreamerV3, IRIS, and TD-MPC demonstrate three structurally different ways to build this local model, and all three outperform their model-free counterparts, with advantages that grow on more complex tasks. This is empirical confirmation of the BWH prediction.*

---

§6 — CL as Computationally Constrained RL

The Big World Hypothesis and the computational embedding result together suggest something broader: continual learning is not a separate paradigm from reinforcement learning. It is RL in the regime that RL was always going to encounter in practice — the regime where the environment is bigger than the agent, where optimal policies are not representable, and where the agent's only recourse is continuous approximation improvement.

Kumar et al. (arXiv:2307.04345, monograph, updated 2025) formalize this unification. They introduce a framework in which CL is RL under computational constraints — specifically, the constraint that the agent cannot represent or

access all of its past experience simultaneously. This constraint is not artificial; it is what the computational embedding imposes. An agent that cannot replay arbitrary past transitions (because it cannot store them all) is an agent with a limited memory horizon. An agent that cannot reprocess all past tasks with its current parameters (because computation is bounded) is an agent with a limited revision horizon.

Within this framework, the phenomena that continual learning research has catalogued — catastrophic forgetting, plasticity loss, forward transfer, backward transfer — are not ad hoc pathologies. They are predictable consequences of computational constraints applied to a standard RL objective. Forgetting is what happens when the memory constraint prevents replay of old transitions. Plasticity loss is what happens when the revision constraint prevents re-optimization of old task features. Forward transfer is what happens when the world model is accurate enough to bootstrap new task learning from prior structure.

The monograph also presents empirical case studies on forgetting, relearning, exploration, and auxiliary task learning — demonstrating that the framework's predictions match observed behavior in CRL systems.

This reframing matters for practice. It means that CRL researchers can draw on the full theoretical machinery of RL — value function approximation, policy gradient theory, model-based planning — rather than treating CL as an isolated specialty with its own separate toolkit. The constraints are the domain of CL; the objectives and methods are RL's. [→ A4: The GVF architecture is exactly this: RL's prediction machinery applied under the computational embedding constraint — a proto-world-model that serves the CL objective without requiring CL-specific methods.]

> ***Key Takeaway:*** *Continual learning is not a separate field from RL — it is RL operating under the computational constraints that real environments impose. This unification, formalized by Kumar et al. (arXiv:2307.04345), means that*

> *every result in CRL is potentially generalizable using RL theory, and every CRL failure mode has a corresponding RL-theoretic explanation.*

---

§7 — SPIRAL: The First Empirical Vindication

If the Big World Hypothesis is correct, a system that operates in a more genuinely non-stationary environment — one that continuously extends the frontier of what the agent must learn — should develop more general capabilities than one trained in a fixed, small environment. This is a testable prediction.

SPIRAL (Liu et al., 2025, arXiv:2506.24119) provides the closest existing test of this prediction. SPIRAL trains language models via zero-sum multi-agent self-play: two agents compete in adversarial games, and as each agent improves, the other agent must improve further to remain competitive. The non-stationarity is endogenous — it is generated by the agents themselves, not imposed by the environment designer. As the skill level rises, the effective task distribution shifts. The agent cannot memorize its way to success; the distribution keeps moving.

The empirical result is striking. Training Qwen3-4B-Base on Kuhn Poker — a simple card game — without any mathematical supervision yields an 8.6% improvement on math benchmarks and an 8.4% improvement on general reasoning benchmarks (Liu et al., 2025). The model received no math training. The improvement emerged entirely from competitive self-play dynamics: expected value calculation, case decomposition, bluffing and counter-bluffing strategies that require systematic conditional reasoning.

This result is direct evidence for the BWH's central claim. An agent trained in a self-play environment — where the opponent continuously enlarges the effective world the agent must navigate — develops more general cognitive capabilities than the same agent trained on any fixed curriculum. The world model here is

implicit: the agent must develop a theory of what the opponent will do, which requires building a predictive model of opponent behavior. This is world modeling in a social environment.

The SPIRAL result also illuminates why SPIRAL's structure is structurally superior to fixed-task benchmarks for testing CRL hypotheses. [← A10: The Benchmark Gap showed that no pre-2025 benchmark eliminates the episodic reset — SPIRAL eliminates it structurally, because self-play is continuous.] In self-play, there is no moment where the environment returns to a neutral state. The agent's current policy *is* the environment for the opponent. Changes in the agent's policy change the environment the opponent faces. This creates the kind of genuine, endogenous non-stationarity that the BWH predicts is the normal condition of any agent operating in a big world.

Verwimp et al. (2023, arXiv:2311.11908) survey the landscape of CL applications and identify the road forward as precisely this direction: systems that generate their own non-stationarity, rather than passively receiving it from a fixed task designer. SPIRAL is the first large-scale realization of that direction.

> ***Key Takeaway:*** *SPIRAL empirically validates the BWH prediction. An agent trained under genuine, endogenous non-stationarity — where the effective world keeps growing because the opponent keeps improving — develops capabilities that transfer beyond its training domain. The 8.6% math improvement from Kuhn Poker training, without math supervision, is the Big World Hypothesis working as predicted.*

---

§8 — What Remains Open: The Simulator Gap

The BWH is well-motivated philosophically and mathematically. The computational embedding result is rigorous. The world model evidence is empirically strong. And yet a genuine Big World Simulator — as called for by Kumar et al. (2408.02930) — does not exist.

The gap between what we have and what the hypothesis demands is specifically this: all existing systems that come close to the BWH regime (DreamerV3 across 150+ tasks, SPIRAL via self-play, AgarCL via continuous ecology) operate in bounded domains. DreamerV3's 150+ tasks are diverse, but they are a curated, finite set with defined observation and action spaces. SPIRAL's self-play generates non-stationarity within the structure of a specific game. The question — whether a system trained in these environments would genuinely generalize to a richer world — remains unanswered.

This is not a flaw in the BWH. It is the natural consequence of the BWH being ahead of current infrastructure. The hypothesis states a mathematical property of real environments; building evaluation systems that fully instantiate that property is an engineering challenge that has barely begun.

Three specific open questions follow directly from the BWH's predictions:

1. **Scale of non-stationarity**: How large does an environment need to be before a fixed-architecture agent demonstrably fails to converge? DreamerV3 succeeds at 150+ tasks with fixed hyperparameters — does this success continue at 1,500 tasks? At 15,000? The computational embedding result predicts a threshold; no experiment has found it.

2. **World model completeness**: DreamerV3 learns a world model; IRIS learns a world model; TD-MPC learns a world model. But all of these world models are domain-specific, trained within a particular observation and action space. A Big World Simulator would require a world model that updates across fundamentally different observation modalities and action spaces simultaneously. No current architecture does this.

3. **Evaluation protocol for BWH-satisfying agents**: The metric of "performance on task τ" is appropriate for small-world benchmarks. The appropriate metric for a BWH-satisfying agent is something closer to Abel et al.'s lifetime average reward — accumulated continuously over an indefinitely long run in an indefinitely complex environment. No evaluation infrastructure currently supports this.

These are not arguments against the BWH. They are its agenda for the next phase of research.

> ***Key Takeaway:*** *The Big World Hypothesis is theoretically established and empirically supported, but the engineering infrastructure to test it rigorously — a genuine Big World Simulator — does not yet exist. This is the central open problem at the intersection of CL and WM research.*

---

§ What Comes Next

This article has established the theoretical spine:

- **The BWH** (Javed & Sutton): the world is irreducibly larger than any agent
- **Computational embedding** (Lewandowski et al., 2023): the constraint is not engineering — it is ontological

- **CL as constrained RL** (Kumar et al., arXiv:2307.04345): CL is the right name for RL in the regime the BWH describes
- **World models as the tractable response** (DreamerV3, IRIS, TD-MPC): local modeling is sufficient to make the BWH's challenge manageable
- **SPIRAL as empirical vindication** (Liu et al., 2025): endogenous non-stationarity drives the generalization gains the BWH predicts

[→ A4: GVFs as Proto-World-Models] traces the architectural lineage that the BWH implies. If the response to the BWH is a world model that updates continuously across an agent's lifetime, then the question is how to build such a model — and the General Value Function architecture, developed at University of Alberta starting in 2011, turns out to be a blueprint that anticipated this requirement by over a decade. A4 shows the structural correspondence between GVF predictions and modern world model components.

[→ A6: The Forgetting Transformer] shows what happens when the BWH response is applied at the architectural level. If the agent must continuously update its world model without discarding prior knowledge, then the attention mechanism — designed for static input distributions — needs modification. The Forgetting Transformer is that modification.

The emotional shift from A1 to this article is intentional. The plasticity crisis is alarming because it seems like a fundamental limit on what deep learning can do. The Big World Hypothesis reframes that alarm as understanding: the plasticity crisis is not a bug — it is the exact failure mode that the BWH predicts for a model-free agent in a big world. The fix is not to patch the training procedure. The fix is to give the agent a world model.

> ***Final Key Takeaways***

1. *The Big World Hypothesis (Javed & Sutton) states that real environments are irreducibly larger than any finite agent — not by degree, but by orders of magnitude. The gap does not close as agents scale.*
2. *Lewandowski et al. (2023) formalize this as computational embedding: the agent's computational constraint is an ontological consequence of being inside its environment, not an engineering choice. Regardless of capacity, the agent is always embedded — always constrained.*
3. *The BWH makes continual learning mathematically necessary, not merely convenient. If no agent converges to optimal in a big world, then all agents must update indefinitely, and the only question is how gracefully.*
4. *World models (φ) make the BWH tractable: an agent does not need to model the whole world, only the relevant local dynamics. DreamerV3 (150+ domains, single config), IRIS (1.046 HNS on Atari 100k), and TD-MPC (continuous control from pixels) demonstrate three complementary proofs of this principle.*
5. *SPIRAL (Liu et al., 2025) provides the first empirical vindication of the BWH prediction: endogenous non-stationarity (self-play) drives generalization gains (8.6% math improvement from poker training) that no fixed curriculum produces.*
6. *The central open engineering challenge is the Big World Simulator: an environment large enough, non-stationary enough, and open-ended enough that no fixed-architecture agent can converge. Building one is the research frontier that the BWH defines.*

GET THE FULL BOOK

Continual Intelligence

Twelve essays that read the continual-learning literature so you don't have to — from the benchmark gap and the plasticity crisis through world models, RL-for-reasoning, and the case for open-ended AI. Every claim was traced to its source paper before publication.

It is a **living book**: versioned like software, and every future edition is included with your copy. The purchase includes the book as **PDF and EPUB** and every future version. The essays are free at **continualintelligence.xyz** — the book buys permanence, sequence, and one artifact you can hand to a teammate.