When AI Builds Itself, Who Remembers Why?

Temperature Zero, Recursive Self-Improvement, and the Continuity Layer AI Safety Is Missing

Francisco J. Mayorga, Jr.
Creator, Mayorga Mnemosyne AI Continuity Framework™
June 2026

Artificial intelligence is entering a new phase. The question is no longer only whether a model can answer a prompt, write code, pass an exam, or use tools. The harder question is whether AI systems may soon help build future AI systems, and whether human institutions can preserve the reasons behind that process.

Anthropic recently framed this concern with unusual directness in its essay "When AI builds itself." The company is careful to say that full recursive self-improvement has not arrived and is not inevitable. That caution matters. But the same essay also argues that AI is already accelerating AI development, and that the possibility of systems helping design their own successors may come sooner than most institutions are prepared for. [1]

This essay argues that the public debate is still missing a layer. Temperature zero, the AI setting that tells a model to choose the most likely next token rather than sample more freely, is not continuity. It can reduce token-level variation, but it does not preserve the memory of why an answer mattered. Retrieval is not continuity. A larger context window can surface more documents, but it cannot decide which version of a decision still governs. A pause is not continuity. A pause may slow development, but it does not automatically preserve the reasons, assumptions, evidence, and judgments that made the last development step acceptable.

The missing layer is institutional continuity: the capacity of an organization to preserve meaning across time. Institutional memory is not simply a storage room full of files. It is closer to a hospital chart. A chart matters because it connects symptoms, tests, diagnoses, medications, allergies, past decisions, and current risks. Without that continuity, a patient becomes a series of disconnected visits. Without institutional continuity, an AI-assisted organization becomes a series of impressive answers with no durable memory of why they were trusted.

The *Mayorga Mnemosyne AI Continuity Framework™* proposes continuity as a neglected governance layer above inference, retrieval, and emergency coordination. Its claim is not that AI should stop changing. Its claim is that consequential change must become legible, challengeable, and accountable. If AI systems begin helping build future AI systems, then society will need more than faster models and better logs. It will need a way to remember why important changes were accepted.

Temperature zero can make a machine more repeatable. Continuity architecture makes AI-assisted reasoning accountable across time.

1. The Loop Begins to Close

For most of AI history, human teams drove the development cycle: writing the code, designing the experiments, interpreting the results, and deciding which failures mattered. AI could assist, but humans remained the main continuity layer. They remembered the project history, the design tradeoffs, the

unresolved debates, the warnings that did not make it into the final slide deck, and the reasons one path was chosen over another.

That boundary is softening. Anthropic describes a progression from chatbots helping with short code snippets, to coding agents that can write and edit files, to autonomous agents that run code and delegate hours of work to other agents. In the future, Anthropic says, agents could become capable enough to build and train models themselves. If that happens, future versions of Claude could be continuously improved by Claude itself. [1]

This is not yet the normal operating state of AI development. But it is credible enough that governance should prepare before the loop tightens. Recursive self-improvement, in plain language, means that the tool begins to participate in improving the next version of the tool. The model writes code, proposes tests, summarizes evaluation results, helps tune training processes, or recommends design changes that shape the next model.

That shift is not only a story about speed. It is a story about memory. When a human team builds a system, the team carries a rough continuity layer in its people, documents, habits, meeting notes, and institutional arguments. That layer is imperfect, but it exists. Once AI becomes part of the build process, that fragile human memory layer becomes more important, not less.

A ship crossing an ocean does not need only a powerful engine. It needs a logbook, charts, compass readings, records of storms, repairs, course corrections, and near misses. A faster ship with no logbook is not simply more efficient. It is harder to govern. The danger is not only that AI may build itself. The danger is that it may build itself faster than we can remember why.

2. What Institutional Memory Means

Let me be specific about institutional memory. It is the ability of an organization to remember its own meaning across time: what it decided, why it decided it, what evidence it trusted, what alternatives it rejected, what language it used, and what changed later.

A company with millions of documents can still forget its own reasoning. A government agency can have archives and still lose the rationale behind a rule. A research lab can have notebooks, dashboards, and code repositories while forgetting which assumptions made a result meaningful. Storage is not memory. Memory begins when stored material can guide future judgment.

The hospital chart makes the point. The chart is useful not because it contains many pages, but because it preserves causal lineage. It tells the next doctor what was tried, what worked, what failed, what cannot be repeated, and what would be dangerous to ignore. Institutional memory performs the same function for organizations. It connects yesterday's reasoning to tomorrow's action.

This matters deeply for AI because AI systems are increasingly used not only to answer questions, but to shape decisions. They help create policies, draft legal documents, shape training programs, and inform executive decisions. If the reasoning behind those outputs disappears after each session, the organization may keep moving while silently losing its memory.

3. The False Comfort of Temperature Zero

Before recursive self-improvement became a public concern, a smaller but revealing mistake had already entered the AI conversation. Many users and builders began treating temperature zero as if it were a synonym for reliability.

The intuition is understandable. In ordinary AI practice, temperature controls how much randomness a model uses when choosing the next token, the small unit of text the model processes and generates. At higher temperatures, the model is more willing to choose less likely tokens. At lower temperatures, it

narrows its choices. At temperature zero, many systems behave as if they are selecting the highest scoring next token.

To a manager, teacher, lawyer, engineer, researcher, or executive, that sounds reassuring. If the model varies too much, turn the randomness down. If the answer changes, freeze the oracle. If the system must be trusted, make it deterministic.

To an ML engineer, it may sound obvious that temperature zero is not memory. But many of the people now governing AI deployments are not ML engineers. They are executives, educators, compliance officers, lawyers, public officials, product leaders, and managers. For them, repeatability can easily be mistaken for reliability.

Temperature zero is like telling a chef not to improvise. It says: follow the standard recipe. Choose the most likely ingredient. Do not wander. But the chef is still working in a living kitchen. The stove may run hot. The ingredients may change. The recipe may have been updated overnight. The same instruction can reduce improvisation without preserving the history of why the recipe exists.

The same is true in AI. Temperature zero adjusts a sampling parameter. It may help the model repeat itself more often. It does not tell the organization whether the answer is canonical, whether the evidence remains valid, whether the definition has shifted, or whether the recommendation contradicts a prior decision.

A thermometer is not a memory system.

4. Why Zero Is Not Really Zero

Even at the technical level, temperature zero should not be treated as a perfect guarantee in real production systems. The reasons are not mystical. They are mechanical.

Modern AI systems run on vast numerical calculations. Tiny rounding differences can, in rare cases, cause one word to win over another. Add the fact that production servers often group many users' requests together to save costs, and the same visible prompt can quietly travel through a different computational path.

Think of it like measuring the same recipe with two kitchen scales. Most of the time, the difference is too small to matter. But if two ingredients are almost tied, a tiny measurement difference can decide which one gets added first. In an AI model, that tiny difference can become the first fork in a longer road.

PyTorch documentation notes that floating-point arithmetic has limited precision and that addition and multiplication are not associative. In simple terms, changing the order of numerical operations can slightly change the result. [6] Usually this does not matter. But when two next-token scores are extremely close, a microscopic difference can change the chosen token, and one different token can send the rest of the answer down a different path.

Production serving adds another source of variation: batching. Servers often process multiple requests together to use expensive hardware efficiently. vLLM documents batch invariance as a feature for reducing this class of variation, and Thinking Machines Lab has described batch size and numerical paths as important practical sources of nondeterminism in LLM inference. [4][5]

OpenAI's developer materials have described seed and system fingerprint tools as providing mostly deterministic behavior, not absolute reproducibility. [3] That distinction is important. Responsible builders should still use the lower-level tools available to them: seeds, stable model snapshots, disciplined prompts, and batch-invariant serving where feasible. These reduce one class of variation. They lower the noise floor.

But a lower noise floor is not a memory spine.

5. The Perfectly Repeatable Amnesiac

Now imagine the strongest possible version of deterministic inference. Suppose an AI system always produces the exact same output when given the exact same input. No hidden variation. No batching surprise. No floating-point fork. Same prompt, same answer, every time.

That would be valuable. It would help engineers test systems, help researchers compare models, and help organizations audit outputs. It would matter in automated evaluation workflows where one model is used to judge another. It would matter in reinforcement learning pipelines where outputs become training signals.

But would it make the AI trustworthy as institutional intelligence? Not necessarily.

A perfectly repeatable model can still forget that an organization rejected a recommendation last month. It can still use the word risk one way in January and another way in June. It can still cite a source without preserving how that source was selected. It can still recommend a strategy that leadership already rejected for documented reasons. It can still be deterministic while the institution remains amnesiac.

The *Mayorga Mnemosyne AI Continuity Framework™* calls this condition *brilliant amnesia*: local brilliance without durable memory. The system performs impressively in the moment. The organization loses the thread across time.

Brilliant amnesia is not ordinary forgetfulness with a new name. It is organizational forgetting amplified by AI speed, fluency, and authority. Because AI outputs often look polished, the loss of reasoning lineage can be hidden. The answer looks complete. The chart is missing. The institution sees the conclusion but not the chain of custody behind it.

This is the enterprise problem beneath the technical problem. AI systems increasingly shape policies, training content, product decisions, legal drafts, and risk assessments. If those outputs do not preserve why they were accepted, what they depended on, and how they changed, the organization accumulates text without accumulating judgment.

6. A Pause Buys Time. It Does Not Create Memory.

Anthropic's call for a coordinated, verifiable slowdown or pause option should be taken seriously. Reuters reported that Anthropic has urged frontier AI developers to establish a coordinated way to slow or temporarily pause development if advanced systems begin improving themselves faster than society can manage the risks. [2]

A pause may be necessary. It may buy time for evaluation, governance, coordination, and public deliberation. But a pause is not continuity.

Grounding an aircraft does not make it safe by itself. It buys time. Safety comes from maintenance records, inspection protocols, failure reports, design histories, component traceability, pilot decisions, and a disciplined account of what must be fixed before flight resumes. The grounding matters because it creates a space for memory, investigation, and repair.

The same is true for frontier AI. A pause may buy time. Continuity tells us what to do with the time. It asks what changed, what evidence justified the change, what warnings were unresolved, which definitions were active, who accepted the risk, and what would make development safe to resume.

Without continuity, a pause can become only a brake. Useful, perhaps necessary, but insufficient. It slows motion without necessarily preserving meaning.

7. Retrieval, Logs, and MLOps Are Necessary but Not Sufficient

A common answer to AI amnesia is retrieval. Give the system a vector database, meaning a database that stores text and ideas as numerical patterns rather than simple keywords. Give it file search. Give it a larger context window. Give it the whole archive.

These tools are valuable. They help the model see more. But seeing more is not the same as remembering correctly.

A warehouse can hold every document in the company and still be useless if no one knows which document is current, which memo was superseded, which source became unreliable, or which decision was approved. Continuity is the librarian who knows which edition matters. It is the archivist who preserves the chain of custody. It is the reviewer who knows that a source can be relevant and still not authoritative.

The same caution applies to logs and MLOps. Model cards make model reporting more transparent by documenting intended uses, performance, evaluation procedures, and relevant context. [7] Datasheets for Datasets propose documentation for datasets, including motivation, composition, collection, and recommended uses. [8] NIST's AI Risk Management Framework gives organizations a common language for governing, mapping, measuring, and managing AI risk. [9] MLflow model registries and tools such as LangSmith provide model lineage, traceability, observability, and production monitoring. [10][11]

These tools are like cameras, receipts, labels, and maintenance logs. They tell us many important things. But they do not always tell us the full courtroom story: what counted as evidence, what objection was raised, why one interpretation prevailed, and who accepted the judgment.

These are important advances. The continuity framework does not replace them. It asks what still remains after them: how does an organization preserve the human-governed meaning of a decision across time?

A Git commit records what changed in code. A model card records what is known about a model. A trace records what happened during a run. A policy framework helps organize risk management. A continuity record asks a different question: what did this change mean, why was it accepted, what prior reasoning did it alter, and who is accountable for treating it as valid?

That is the difference between technical persistence and institutional continuity. Technical persistence keeps artifacts. Institutional continuity preserves the governed reasons those artifacts matter.

8. What a Continuity Record Looks Like

The prescription cannot remain purely abstract. Consider a realistic scenario: an AI-assisted lab changes an evaluation threshold after a coding agent begins completing longer, more open-ended software tasks. A normal technical log might record the code change, the date, the metric, and the person who merged the update. That is useful. It is not enough.

A continuity record would preserve the surrounding judgment. It would ask what assumption changed, what evidence justified the new threshold, what prior threshold was superseded, what risks were accepted, who approved the change, what dissent or uncertainty remained, and when the decision must be reviewed again.

Continuity field	What it preserves
Decision or change	The evaluation threshold changed from one approved standard to another.
Reason accepted	The prior threshold no longer matched the observed task horizon and risk profile.
Evidence used	Benchmark results, incident reports, human review notes, and known limitations.
Prior memory altered	The old threshold remains visible as superseded, not silently erased.
Human governance	The reviewer, approver, unresolved objections, and review date are preserved.
Future trigger	Conditions under which the record must be revisited, such as a new model capability or safety incident.

This example is deliberately modest. It is not a full technical specification. It is a public sketch of the missing object: a record that preserves not only what happened, but why the change became legitimate. It shows how continuity differs from a transcript, a log, or a static document. It converts an AI-assisted output into an institutional memory object that can be challenged, superseded, corrected, or carried forward.

9. What the Continuity Framework Adds

The *Mayorga Mnemosyne AI Continuity Framework™* proposes continuity as a governance architecture for AI-assisted work. It is not a new neural network architecture, not a replacement for model documentation, not a substitute for safety evaluation, and not a guarantee of truth. It is a way to organize the memory layer around consequential AI outputs.

In plain terms, the continuity framework asks fewer but deeper questions: What was decided? Under what assumptions? Why is this answer different from the last one?

Those questions are simple. Their consequences are not. A system designed around the *Mayorga Mnemosyne AI Continuity Framework™* treats important AI outputs as continuity records rather than disposable text. It tries to preserve the path from question to answer, from answer to decision, and from decision to later revision.

The ambition is not to make AI perfectly deterministic. The ambition is to make AI-assisted reasoning durable, auditable, and governable. The lower layer tries to make outputs reproducible. The upper layer asks whether the meaning of those outputs survives time.

This is why the continuity framework matters in the age of recursive self-improvement. If AI systems begin building future AI systems, we need a way to preserve not only what they built, but why each meaningful change was accepted. The question at the center of the framework is whether intelligence can remain accountable as it accelerates.

10. What Continuity Cannot Promise

A serious continuity framework must state its limits.

Continuity architecture cannot make a weak model wise. It cannot guarantee truth. It cannot eliminate every source of low-level nondeterminism inside closed provider infrastructure. It cannot prove that

recursive self-improvement will or will not occur. It cannot replace technical safety research, alignment work, cybersecurity, governance, law, or human judgment.

It also cannot be reduced to memory at any cost. Bad memory is dangerous. Hallucinated memory is worse. A system that lets AI promote its own unsupported claims into institutional canon would not solve brilliant amnesia. It would automate it.

This is why human-governed memory matters. AI may propose a memory. It may summarize evidence. It may flag a possible contradiction. But consequential canon should not be silently written by the system whose behavior is being governed. In this view, AI proposes, humans govern, and continuity preserves.

The correct relationship is not determinism versus continuity. It is determinism plus continuity. Deterministic serving lowers the noise floor. Retrieval expands the available evidence. Safety evaluations probe dangerous capabilities. Policy sets thresholds. Continuity preserves the reasons, decisions, definitions, and justified changes that must survive across all of them.

That is the sober claim. It is narrower than a universal solution and stronger because of it.

11. The Question After Intelligence

The public conversation often asks whether AI will become more intelligent. That question matters. But it is no longer enough. We also need to ask whether AI-assisted institutions can remain continuous as intelligence accelerates.

If models become more capable, organizations will be tempted to move faster. If agents can write code, generate evaluations, summarize risks, and propose changes, teams will be tempted to trust the velocity. But speed without continuity is not compounding intelligence. It is acceleration without memory.

The next frontier is not only artificial intelligence. It is intelligence that remembers why it changed.

When AI systems help shape future AI systems, the core governance question is simple: can we still inspect why important changes were accepted? If we cannot, we may not be governing intelligence at all. We may be building systems capable of passing every exam in the world while stripping them of the infrastructure required to remember why the answers matter.

Intelligence without continuity does not compound. It accelerates its own amnesia.

Author Note

Francisco J. Mayorga, Jr. is the creator of the *Mayorga Mnemosyne AI Continuity Framework™*, a continuity architecture for preserving governed memory, provenance, decision lineage, and justified change across AI-assisted work. The framework's underlying argument, that intelligence must be made to survive time and not only perform in the moment, is developed across ongoing essays and two extended works: *Mnemosyne: The Doctrine of Continuity* and *The AI Continuity Advantage*.

Framework updates, practical experiments, and essay archives are available through the following public channels:

- franciscomayorga.com
- GitHub: Mayorga Mnemosyne AI Continuity Framework™
- Zenodo DOI record
- Academia.edu profile

References and Source Notes

[1] Anthropic. "When AI builds itself." June 2026. https://www.anthropic.com/institute/recursive-self-improvement

[2] Reuters. "Anthropic says AI labs need coordinated plan to halt development if risks rise." June 4, 2026. https://www.reuters.com/business/anthropic-says-ai-labs-need-coordinated-plan-halt-development-if-risks-rise-2026-06-04/

[3] OpenAI Cookbook. "How to make your completions outputs consistent with the seed parameter." November 2023. https://developers.openai.com/cookbook/examples/reproducible_outputs_with_the_seed_parameter

[4] Thinking Machines Lab. "Defeating Nondeterminism in LLM Inference." September 10, 2025. https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/

[5] vLLM Documentation. "Batch Invariance." Accessed June 2026. https://docs.vllm.ai/en/latest/features/batch_invariance/

[6] PyTorch Documentation. "Numerical accuracy." Accessed June 2026. https://docs.pytorch.org/docs/stable/notes/numerical_accuracy.html

[7] Mitchell, Margaret, et al. "Model Cards for Model Reporting." FAT* 2019 / arXiv:1810.03993. https://arxiv.org/abs/1810.03993

[8] Gebru, Timnit, et al. "Datasheets for Datasets." Communications of the ACM, 2021 / arXiv:1803.09010. https://arxiv.org/abs/1803.09010

[9] National Institute of Standards and Technology. "Artificial Intelligence Risk Management Framework (AI RMF 1.0)." 2023. https://doi.org/10.6028/NIST.AI.100-1

[10] MLflow. "ML Model Registry." Accessed June 2026. https://mlflow.org/docs/latest/ml/model-registry/

[11] LangChain. "LangSmith Observability." Accessed June 2026. https://docs.langchain.com/langsmith/observability