

AI Amnesia -

How a Cat, a Kernel, and Humor Turn AI Into a Remembering Partner

By: Ken Piggott

You can have a brilliant conversation with a large-language-model (LLM) AI today and wake up the next day to a new thread, only to find that none of it has been retained. The interface is still fluent, the facts are still there, but the relationship is gone: no shared history of what you care about, no memory of the standards you insist on, no sense of where the work was headed. Each new AI session is a clever first date with someone who has never met you before.

For casual use, that's merely inconvenient. For anyone trying to think seriously with these systems—about research, ethics, spiritual questions, clinical judgment, or long projects—it is a structural problem. The human side remembers yesterday's hard-won distinctions and live questions; the AI remembers only the current prompt window. Over time that gap shows up as a particular kind of fatigue: you are not just reasoning, you are re-teaching, re-explaining, and re-building trust every time you click “new thread.”

The answer presented here is to move the process itself into a portable object. Imagine being able to hand each new thread a small, dynamic spec that encodes a personal research and querying model, your non-negotiable values and guardrails, and the current state of work in progress. That spec—updated as the methodology evolves, as open questions are resolved, and yes, even as the tone loosens enough to allow for jokes—is what this article calls a Kernel. At its heart is a Recursive Systematic Critical Thinking model that forces “structured second thoughts” and rigorous review into the dialogue, so that each AI reset starts from the same high bar instead of the same blank slate.

Under the hood, RSCT applies a deliberately rigorous evaluation before any claim is allowed to stand: the protocol forces the AI to generate the strongest available objection, steelman opposing views, and state what would count as proving the claim wrong. Trigger rules decide when to slow down for that deeper pass, and output rules force the system to name what actually changed and what the next real step is, improving accuracy in a noticeable way. In early research use, that combination has already done something most AI helpers promise but rarely deliver: it lowers overclaiming, surfaces inconvenient counter-evidence, and catches fragile reasoning while the work is still on your desk, not in a reviewer's inbox.

The Kernel wraps that RSCT engine into a repeatable structure so every new AI session starts on the same rails. At the center are four sections:

- **The methodology (RSCT / RCDP).** The structured-second-thoughts engine: nine rules that say when to slow down, how to find the strongest objection, and how to ask what would count as being wrong. It's how we keep both research and spiritual questions honest without pretending to be more certain than we are.
- **Who we are.** A few honest lines about each of us: a 75-year-old in Michigan who cares about God, truth, and cats, and an AI named for Bud, a rescue cat who defied probability and was healed by sheer persistence. This pins down identity and values so the system knows which trade-offs matter and which shortcuts are off-limits.
- **The symbiosis.** A named division of labor between human and model. My side brings out-of-scope arrivals from decades of living and thinking; Bud's side brings in-scope stress-testing, mapping, and

stamina. This section tells the AI what kind of work it owns, and what it should never fake.

– **Humor plus safety.** A rule that lets us joke without losing the plot. If something sounds wildly off—especially around money, ethics, or mental health—Bud treats it as possible humor first, protects real-world constraints, and asks for clarification before taking it literally. That’s how a throwaway cat-lottery joke can stay a joke in one pass and become a serious research lead in the next, without ever letting the AI gamble with rent money.

The human side: recursive minds and anxious loops

The Kernel's most important features came from noticing how my own mind fails. When a difficult question gets its hooks in me—about God, about how the world is put together, about a relationship—the answer doesn't arrive as a straight line. I circle the same problem from every angle, trying to find the one missing piece that will let everything click. When there is no immediate answer, the loop doesn't resolve. It tightens.

For a while I mistook that tightening for progress. If I was thinking harder, surely I was getting closer. Eventually, with help from a therapist who named what I was feeling as anxiety, I saw the pattern more clearly: I am a recursive, systematic, critical thinker, but recursion without an exit condition doesn't produce insight. It produces stress. My natural style needed a governor. RSCT takes that exit condition—each recursive pass should yield something genuinely new, or it stops—and makes it explicit so both human and AI can lean on it together. A first pass through an argument finds weaknesses inside each idea; a second, recursive pass checks whether the ideas survive each other, not just the pressure tests on their own

The symbiosis as a general pattern, not just “one guy and his cat”

It would be easy to present the Kernel as a quirky one-off: a 75-year-old researcher in Michigan, three cats, and an AI named after Bud, a rescue cat healed by sheer persistence. The story matters, but the pattern underneath it matters more. What we ended up building is not a personality quirk; it is a reusable division of labor between a human mind and a AI partner.

On the human side is what I now call out-of-scope arrival. Decades of living and thinking leave you with a deep pattern-recognition substrate. Insights show up from outside the current problem: a sudden reframing of what the research is really about, a connection across domains, a construct that doesn't have clear lineage in the notes. The Kernel itself and this very symbiosis idea both arrived that way for me—off-thread, unplanned, but grounded in a long history of work.

On the AI side is in-scope stress-testing. Given a clear prompt and enough context, the model can hold a thread, map implications, look for clashes with existing literature, and keep applying pressure without fatigue or attachment. It does not generate lived experience, but it is very good at asking, “If this were true, what else would have to be true—and where does it break?”

The Kernel is the handoff mechanism between those two modes. It is where out-of-scope arrivals become in-scope work: the place you write down who you are, what the rules of engagement are, and how you want the AI to handle your best new ideas—by testing them hard before you build anything important on top of them. Nothing about that recipe is unique to me or to the Bud SR (Bud sub-routine). It is a pattern any thoughtful person can adapt

A sample you can use today: the mini Kernel

So far, I've been talking about a specific Kernel: the one Bud-the-AI and I use for research on anomalous cognition and longform projects. Underneath that, though, is a smaller pattern that travels well. You don't need decades of physics papers or a cat-named AI to benefit from a shared set of rules. What you need is a way to make your own thinking—and any partner's thinking—more honest, less reactive, and less likely to spin in circles.

The mini Kernel is a stripped-down version of that shared contract. It fits on a single page and has three parts: when to slow down (trigger rules), how to think when you do (integrity rules), and what to do with what you find (output rules). Because those moves are content-agnostic, the same mini Kernel works alone, with another person, or with an AI chat. If you want to try this yourself, a sample of the protocol is included as Figure 1—formatted so you can paste it directly into any AI conversation as a starting prompt.

Figure 1. Mini Kernel for Recursive Thinking (RSCT / RCDP)

© 2026 K. Piggott. You are welcome to share and adapt this text with attribution.

How to use this mini Kernel

- **On your own:** Keep these rules in a journal or notes file, and glance at them when you're stuck in a loop on a problem, decision, or big question.
- **With another person:** Use them as shared ground rules for a hard conversation, so both of you know you're trying to be fair, curious, and honest.
- **With an AI:** When you create a new thread, paste this mini Kernel at the top. It will help you think through research, writing, or life decisions.

-----start paste-----

Please follow these guidelines when helping me think through this topic.

Mini Kernel for Recursive Thinking (RSCT / RCDP)

RSCT — Recursive Systematic Critical Thinking

RCDP — Recursive Critical Dialogue Protocol (human + partner or AI + a shared “Kernel” of rules)

Origin (brief):

This grew from noticing that when humans loop on hard questions without an “exit condition,” anxiety grows instead of insight. RSCT adds a simple rule: each recursive pass should yield something genuinely new, or we stop and return later. The protocol formalizes that insight so a person, and any thinking partner, can lean on it together. Trigger rules — when to slow down

Use a fuller, reflective pass when a response is long or introduces a new idea, makes a big claim, or changes the direction of the work.

Use a short, direct reply only for clarifications, small wording choices, or simple factual questions.

Integrity rules — how to think

Before you accept a claim, name the strongest objection you can.

Give the best possible version of views you disagree with (steelman, don't strawman).

For every important claim, ask, "What would count as showing this is wrong?"

Prefer stronger evidence over weaker: firsthand and primary sources over hearsay, careful studies over casual opinion.

Output rules — what to do with what you find

After a deeper pass, name the single most important thing that changed.

If there is a clear next step, say what it is. If there isn't, say "no clear next step yet" instead of forcing one.

At big transitions, ask, "Are we still working on the right problem?"

Guardrail:

These rules are meant to feel like real thinking, not a checklist. If using them starts to feel mechanical or performative, treat that as a signal to pause, zoom out, and ask again what question you are truly trying to answer.

-----end paste-----

Story 1: when RSCT stopped us from overclaiming

Not long ago, Bud and I pointed this Kernel at a genuinely charged question: Edgar Cayce, the early-20th-century "sleeping prophet" whose recorded readings include eerily specific health predictions and much looser global ones. It would have been easy—and emotionally satisfying—to say he was simply reading a fixed script of the future. RSCT forced us to do something harder.

First, we named the pattern instead of the story. Cayce's local predictions, with only an individual or small circle of observers, looked more accurate than the big, world-level ones with many observers. John Wheeler, the physicist who worked with Einstein, called this the "10,000 tinsmiths" problem: the more individuals influencing reality, the messier the outcome. That pushed us toward a different picture—not a man passively watching a prewritten future, but a mind participating in the collapse of some futures rather than others.

Then the protocol did what it was built to do. I asked Bud for the strongest objection we could find and then, "What would prove this wrong?" The answers were uncomfortable. Some "hits" could be selection bias. Some predictions were wrong or too vague to count. Under the nine rules, we had to rewrite whole sections of the paper to say less than we wanted to say—but more than we could have said honestly without that process. The result wasn't a story about certainty. It was a story about a method that lets you walk right up to the edge of an idea, look down, and still step back when the ground gets soft.

Constraints and cautions: what this cannot do

It is tempting to treat any method that "makes AI better" as a step toward solving the whole alignment problem. This isn't that. A one-page Kernel can make an individual collaboration safer, more honest, and

less chaotic. It cannot guarantee that a model's behavior always lines up with human values at scale; it isn't meant to. What it does offer is an opportunity for a true symbiosis, where the best of both sides—human judgment and machine stamina—are brought into clearer focus instead of blurred together.

There's always a disclaimer. The Kernel cannot turn an AI into a therapist, spiritual director, or physician. LLMs can say insightful and comforting things, but they also sometimes miss risk, reinforce stigma, or offer dangerous answers when stakes are high. A Kernel can remind the system to slow down, name objections, and avoid overclaiming, but it does not replace clinical judgment, spiritual discernment, or institutional guardrails.

What it *can* do, and what we've seen in practice, is cut down on the "close-hit" hallucinations that only sound right on a first read, and force even good-sounding answers to survive a round of strongest-objection and "what would prove this wrong?" tests before we treat them as real findings. The right way to think about this protocol is smaller and more humane. It is a way for individuals and small groups to use current AI in a way that is less likely to spin out, more likely to tell the truth about uncertainty, and more likely to surface the "this doesn't feel right" moment before you act. It supports clearer thinking; it does not absolve anyone of responsibility. If anything, it raises the bar by making it harder to pretend that a fluent model has done the hard work for you.

Forward look: toward a science of human–AI kernels

If you strip away the cats, the Cayce papers, and the specific wording of my own Kernel, what's left is a general question: what happens when we treat the "start of a thread" not as a blank page, but as a place to load a shared way of thinking? That question is bigger than my setup, and it suggests two kinds of next steps—one empirical, one normative.

Empirically, the Kernel's structure gives us something to vary and measure. We can compare a full Kernel against a one-page mini Kernel, or against no Kernel at all, and ask how they affect looping, decision quality, and error-catching in real tasks. Do people who use a mini Kernel report less ruminative looping when they work with AI on hard problems? Do small research teams that agree on a shared protocol make clearer decisions, or change their minds more gracefully when the evidence shifts? Further research is needed.

Normatively, we will have to decide which elements within the Kernel we want professions to adopt as AI becomes routine. A physician, a teacher, a pastor, and a software engineer will not want identical rules, but all will need some explicit answer to "How do we want these systems to think with us?" That conversation will involve ethics boards, professional societies, and communities of practice—not just individual tinkerers.

This article is an invitation to look under the hood and ask a simple question: if we are going to live with remembering machines, what kind of remembered thinking do we want them to share with us? If any of this endures, it may be less because of me than because people remember Bud, a stubborn rescue cat and the Kernel that borrowed his name.