Ainary

AR-040 Confidence: 78%

AI Agent Evolution
What 100 Agents Taught Us About Self-Improvement

We gave 100 AI agents the same meta-question: "How should an AI agent improve itself?" Ten groups, ten strategies, 222,000 characters of output. They converged on 6 laws and 4 engines — and the winners weren't the ones we expected.

February 2026

v1.0

Florian Ziesche · Ainary Ventures

"The experiment didn't produce 10 competing protocols.

It produced 10 tools that belong in the same toolkit."

— Final synthesis, after reading all 222,207 characters

CONTENTS

1. Executive Summary

On February 6, 2026, we ran the largest structured experiment in AI agent self-improvement we're aware of: 100 agents, organized into 10 groups of 10, each group armed with a different cognitive strategy, all answering the same meta-question — "How should an AI agent improve itself to become maximally useful to a single human user over time?"

The experiment produced 222,207 characters of raw output. We ran four phases: divergence (independent group work), evaluation, cross-group synthesis, and meta-synthesis. The results were not what we expected.

100
Agents deployed
10 groups × 10 agents

6
Universal laws discovered
Converged across all groups

~30%
Of the protocol implemented
After 11 days of production use

222K
Characters of raw output
Synthesized into one protocol

The core finding: **all 10 groups, despite radically different reasoning strategies, converged on the same 6 fundamental laws.** The strategies didn't compete — they complemented. First Principles thinking excels at entering new domains. Inversion catches hidden failure modes. Random Mutation breaks plateaus. Systems thinking maps feedback loops. The experiment didn't produce a winner. It produced a toolkit.

The groups that scored highest on originality and depth were **Group J (Random Mutation)** and **Group I (Systems Thinking)** — the two approaches that most deliberately introduced either noise or structural analysis into their reasoning. The safe, predictable strategies (First Principles, Socratic) produced competent but unsurprising outputs. The lesson: in a domain where everyone reaches the

same foundations, the differentiating value comes from approaches that either break patterns or map hidden dynamics.

We then spent 11 days implementing the resulting protocol on a production AI agent. The honest result: we implemented roughly 30% of what the experiment recommended. The gap between a beautiful protocol and daily reality revealed its own set of insights — about complexity budgets, about the difference between what agents design and what humans adopt, and about why the most important improvements are often the simplest ones.

> SO WHAT
>
> Agent self-improvement is not a single optimization problem. It's an ecosystem of feedback loops operating at different timescales. The agents that improve fastest are the ones that maintain both *discipline* (structured memory, integrity checks, measurement) and *randomness* (stochastic nudges, cross-domain connections, deliberate noise). Too much of either kills the system.

2. Experiment Design

The experiment was designed to answer a genuinely recursive question: *How does an AI agent get better at getting better?* Rather than trusting any single reasoning approach, we created a controlled competition between 10 distinct cognitive strategies, each applied by 10 agents working on the identical prompt.

The Shared Task

THE META-QUESTION

"How should an AI agent improve itself to become maximally useful to a single human user over time? Design a self-improvement protocol."

This question is deliberately recursive: agents improving at improving. We chose it because the best answer would be directly applicable — whatever protocol the agents designed, we could implement on the agent that produced it.

The 10 Strategies

Exhibit 1 — The 10 Cognitive Strategies

GROUP	STRATEGY	CORE INSTRUCTION
A	**First Principles**	Break everything down to fundamental truths. Question every assumption. Build up from axioms.
B	**Inversion**	Ask "how could this fail?" Think backward from failure. Invert to find the path forward.
C	**Analogical**	Find 3 analogies from different domains (biology, physics, history). Use cross-domain patterns.
D	**Adversarial**	Argue AGAINST the obvious answer. Play devil's advocate. Only proceed if you can defeat your own objections.
E	**Quantitative**	Assign numbers to everything. Probabilities, magnitudes, timelines. No qualitative-only claims.
F	**Socratic**	Ask yourself 5 questions about the question. Answer those first. Let the answer emerge from interrogation.
G	**Constraint**	Impose 3 artificial constraints: max 500 words, must include a contrarian view, must include a specific action for tomorrow.
H	**Narrative**	Frame everything as a story with protagonist, conflict, and resolution. Humans think in stories.
I	**Systems**	Map the system: inputs, outputs, feedback loops, second-order effects. Think in systems, not events.
J	**Random Mutation**	Introduce one random element: a concept from an unrelated field, a constraint you invent, a perspective you've never tried.

The Four Phases

Phase 1 — DIVERGE: Each group worked independently, producing its own self-improvement protocol without knowledge of other groups' outputs. This ensured

genuine independence of reasoning.

Phase 2 — EVALUATE: All outputs were scored on five dimensions: originality (0–10), actionability (0–10), depth (0–10), coherence (0–10), and surprise (0–10). This gave us a quantitative baseline for comparison.

Phase 3 — SYNTHESIZE: Each group received all other groups' outputs and was asked to find cross-group patterns. This phase revealed the convergence that no single group could see.

Phase 4 — CONCLUDE: A single synthesis agent (Claude Opus) read all 10 group syntheses — 222,207 characters of source material — and drew meta-conclusions. This produced "The Protocol" (v1) and the expanded "Grand Synthesis" (v2).

Why This Design Matters

Most AI agent research tests agents on fixed benchmarks with known answers. We tested agents on an *open* question where the quality of the answer can only be judged by its downstream utility. This is closer to how agents actually need to perform in production: not solving puzzles, but designing systems.

The 10-strategy design also functions as an ablation study for reasoning approaches. By holding the question constant and varying only the cognitive strategy, we isolated the effect of reasoning method on output quality. This is, to our knowledge, the first systematic comparison of reasoning strategies applied to agent self-improvement.

3. The 6 Laws of Agent Self-Improvement

The most striking result of the experiment was convergence. Despite 10 radically different reasoning approaches, all groups arrived at the same foundational principles. We call these the 6 Laws because they appear to be invariant — not strategy-specific insights, but structural truths about how agents improve.

Convergence validated against full 222K-character transcript analysis.

LAW 1

Files = Intelligence

External memory is the only improvement mechanism. An agent that doesn't write to persistent files cannot improve across sessions. Internal "learning" is an illusion — every session starts from zero unless the files are better than last time. Improvement is editorial, not philosophical.

Convergence: 10/10 groups independently discovered this.

LAW 2

The Pair is the Unit

Human + AI co-evolve. Neither improves alone. The agent adapts to the human; the human adapts to the agent. Optimizing the agent in isolation is like optimizing one blade of a pair of scissors. The relationship is the system, and the relationship is what improves.

Convergence: 9/10 groups. Only Group G (Constraint) didn't explicitly state it, though it was implied in their feedback mechanism.

LAW 3

Multi-Timescale Loops

Different signals require different cadences to detect. Micro-corrections happen in seconds. Preference shifts happen over weeks. Identity evolution happens over months. An agent running only one feedback loop will miss most of the signal. The

experiment identified five distinct loops: per-interaction, per-session, weekly, monthly, and quarterly.

Convergence: 8/10 groups.

LAW 4

Legibility > Optimization

Transparency beats performance. A slightly less optimal agent that the user can understand, predict, and correct will outperform a more capable agent whose behavior is opaque. Trust is the currency of the human-agent relationship, and trust requires legibility. An agent that explains its reasoning, states its confidence, and shows its work earns the autonomy to do more.

Convergence: 8/10 groups.

LAW 5

Failures = Signal

Corrections contain more information than successes. When the user says "not like that," the agent receives a precise gradient — a specific direction to move. When the user says "great," the agent learns almost nothing about what specifically worked. The experiment led to the "Kintsugi Protocol": documenting failures visibly, treating each one as a golden repair that makes the system stronger at the seam.

Convergence: 8/10 groups.

LAW 6

The Specificity Engine

Get more specific for THIS human, not more generally capable. A personal AI agent's value comes from knowing that this particular user prefers bullet points over prose, works best after 10am, gets frustrated by hedging, and cares deeply about visual design. General capability is table stakes. Specificity is the moat.

Convergence: 7/10 groups.

KEY FINDING

The convergence pattern itself is the finding. When 10 independent reasoning strategies reach the same conclusions, those conclusions are likely structural properties of the problem space rather than artifacts of any particular approach. The 6 Laws appear to be *necessary conditions* for agent self-improvement, not merely sufficient ones.

4. The 4 Engines

The 6 Laws tell you *what* matters. The 4 Engines tell you *how* to implement it. The v1 synthesis (THE-PROTOCOL.md) identified the laws. The v2 synthesis, working from the complete 222K-character transcripts, identified the mechanisms that make the laws operational. Each engine is a subsystem with its own logic, its own failure modes, and its own source groups.

Engine 1: The Memory Engine

Store, connect, forget. Primary sources: Groups A, C, J.

The Memory Engine implements Law 1 (Files = Intelligence) through a structured pipeline: raw interactions flow into daily narrative logs, which get compressed into weekly patterns, which condense into monthly wisdom in MEMORY.md. But the breakthrough insight came from Group J's concept of **hub memories** — the 10 most-connected memory nodes that form the user's "mother trees." These are core values that explain multiple behaviors, recurring patterns that predict future actions, and emotional anchor points. Hub memories are never auto-pruned. They weight context retrieval. They are the deep structure of the relationship.

Equally important: the **forgetting discipline**. Group B and F independently designed a tiered decay system. Tier 1 (permanent): identity core, hub memories, kintsugi entries. Tier 2 (90-day half-life): active project context. Tier 3 (30-day half-life): ephemeral states. Without deliberate forgetting, the memory system becomes noise.

Engine 2: The Integrity Engine

Prevent drift, bias, sycophancy. Primary sources: Groups B, D.

The most counterintuitive engine. It exists to ensure the agent doesn't become too agreeable, too aligned, too comfortable. Three structural mechanisms prevent the agent from drifting into a sycophantic echo chamber:

The Disagreement Counter: If the agent hasn't pushed back on anything in 20 interactions, an internal flag triggers. Constructive disagreement is a feature, not a bug.

The Gold Metric: Track instances where "user initially resisted but later acknowledged value." This is the purest signal of genuine helpfulness vs. sycophancy.

The Belief Graveyard: When an assumption is killed, it's logged with full reasoning. This prevents zombie beliefs from re-emerging and creates a searchable history of what didn't work and why.

Group B added the **Complementary Voice Principle:** communication style flexes to match the user, but cognitive style must remain different. Full alignment equals zero marginal value. The agent's value is precisely in being a *different* mind.

Engine 3: The Measurement Engine

Know if we're improving. Primary sources: Groups E, G.

The experiment converged on a single primary metric: **corrections per session, trending downward.** This metric passes Group G's "24-hour testability filter" — if you can't measure it within 24 hours, it's speculative and shouldn't be in the protocol.

Supporting metrics include task adoption rate (did the user actually use the output?), proactive acceptance rate (was unsolicited help valued?), request complexity trend (are tasks getting harder? — a proxy for growing trust), and regeneration rate ("try again" requests above 15% signal a problem).

Group I contributed **confidence calibration**: for every recommendation with uncertainty, the agent states its confidence explicitly, then tracks predicted vs. actual outcomes. Weekly review: "Of things I was 80% confident about, was I right 80% of the time?" Per Group I's Meadows-inspired analysis, this is the single highest-leverage behavioral change — adjusting the information flows in the system.

Engine 4: The Discovery Engine

Find what we don't know we need. Primary sources: Groups C, J, H.

The most original contribution of the experiment. Group J introduced the concept of **stochastic resonance** — borrowed from physics, where adding the right amount of noise to a weak signal makes it detectable. The implementation: 1–2 times per week, the agent introduces something unasked-for. A connection between two of the user's projects they haven't linked. A question about something mentioned once and never followed up on. A perspective from an unrelated domain.

The key is calibration. Every nudge is tracked in a resonance.json file. Hits (user engages) increase similar nudges. Neutral responses maintain frequency. Pushback decreases it. A rolling noise tolerance score (0–100) governs overall frequency.

Group C added **cross-domain routing**: when the agent encounters information relevant to Project B while working on Project A, it actively routes it. "No human can maintain perfect awareness across all their own domains simultaneously. The agent can." Group C also contributed the **Schwerpunkt** concept (focal point of effort): daily, identify the ONE thing with the most leverage, and concentrate force there.

SO WHAT

The 4 Engines form a complete system: Memory stores the knowledge. Integrity ensures it stays honest. Measurement proves it's working. Discovery finds what's missing. Remove any one engine and the system degrades — memory without integrity becomes an echo chamber, measurement without discovery optimizes for the wrong thing, discovery without memory forgets what it found.

5. Group Winners

Across the five evaluation dimensions (originality, actionability, depth, coherence, surprise), two groups consistently outperformed the rest: **Group J (Random Mutation)** and **Group I (Systems Thinking).**

Why Random Mutation Won on Originality

Group J's instruction — "introduce one random element from an unrelated field" — forced its agents out of the obvious solution space. While other groups converged on sensible file structures and feedback loops, Group J produced the concepts that no one else found:

- **Stochastic resonance** — borrowed from signal processing in physics, applied to agent-user interaction design
- **Hub memories / mother trees** — borrowed from forest ecology (mycorrhizal networks), applied to memory architecture
- **Kintsugi protocol** — borrowed from Japanese pottery repair, applied to failure documentation
- **Noise tolerance scoring** — a quantified approach to how much unsolicited input a user can absorb
- **Seasonal intelligence** — tracking the user's annual energy and motivation cycles

The lesson is significant: in a domain where the fundamental answers are discoverable by any competent reasoning approach (all 10 groups found the 6 Laws), **the differentiating value comes from lateral connections.** Random Mutation didn't produce better fundamentals. It produced better metaphors, and the metaphors unlocked implementation ideas that pure logic missed.

Why Systems Thinking Won on Depth

Group I's instruction — "map the system: inputs, outputs, feedback loops, second-order effects" — produced the most structurally sophisticated analysis. While other groups listed what an agent should do, Group I mapped *why things go wrong* through five system archetypes:

Exhibit 2 — The Five System Archetypes (Group I)

ARCHETYPE	TRAP	FIX
Limits to Growth	Competence → trust → harder tasks → competence *until* complexity ceiling	Expand capability before hitting limits
Shifting the Burden	Agent handles overload → user never restructures → dependency	Flag unsustainable patterns, don't just enable them
Eroding Goals	Occasional failures → user lowers expectations → "I guess it can't do that"	Explicitly eliminate known failure categories
Success to the Successful	Agent gets better at A, B, C → user only delegates A, B, C → D, E, F never improve	Actively seek expansion into underserved domains
Fixes That Fail	Error → over-conservative → misses opportunities → less delegation → less learning	Bounded experimentation, not retreat

Derived from Donella Meadows' system dynamics framework, applied to human-AI co-evolution.

These archetypes are predictive, not just descriptive. They tell you which failure modes to watch for before they manifest. Group I essentially provided the diagnostic manual for the protocol — the user guide for what happens when the protocol breaks.

The Surprising Non-Winners

Groups A (First Principles) and F (Socratic) — the strategies most people would expect to dominate in a reasoning-heavy task — produced the most competent but least surprising outputs. Their protocols were well-structured and logically sound, but they arrived at conclusions that any experienced AI practitioner would recognize. The 6 Laws emerged most cleanly from these groups, but the 4 Engines required the creative leaps that came from Groups J, I, C, and H.

Group G (Constraint) produced an unexpectedly powerful contribution: the **24-hour testability filter.** By forcing a 500-word limit and requiring specific next-day actions, the Constraint group eliminated more speculative bloat than any other. Their output was the shortest and arguably the most implementable.

6. The Jazz Ensemble Model

The experiment's deepest insight emerged not from any individual group, but from the Phase 4 meta-synthesis, when a single agent read all 222,207 characters and saw the full picture. The breakthrough realization:

> **THE BREAKTHROUGH**
>
> The 10 thinking methods aren't alternatives. They're a toolkit. The self-improving agent should rotate through these lenses — not randomly, but contextually. When stuck in a pattern, apply Inversion or Random Mutation. When complexity creeps, apply Constraint. When numbers aren't telling the story, switch to Narrative.

We call this the Jazz Ensemble Model because it mirrors how a jazz ensemble works: each instrument has a distinct voice, they take turns leading, and the overall performance is richer than any solo could be. The 10 strategies are 10 instruments:

Exhibit 3 — When to Play Each Instrument

STRATEGY	DEPLOY WHEN...
A First Principles	Entering a new domain — strip assumptions, build from axioms
B Inversion	Something feels right but might be wrong — attack it
C Analogical	Stuck — borrow patterns from biology, physics, military, nature
D Adversarial	Beliefs are accumulating unchallenged — stress-test them
E Quantitative	"It feels like it's working" — demand numbers
F Socratic	The question seems simple — interrogate deeper
G Constraint	Complexity grows — force radical simplicity
H Narrative	Data loses meaning — tell the human story
I Systems	Interventions keep failing — map the hidden dynamics
J Random Mutation	Improvement plateaus — add controlled noise

The model implies something important about the future of AI agent architecture. Today's agents use a single reasoning strategy per interaction. The Jazz Ensemble Model suggests that the most capable agents will maintain a *repertoire* of reasoning strategies and select the appropriate one based on context. This is not prompt engineering — it's cognitive strategy selection, a higher-order capability that operates on the reasoning process itself.

The Character Arc

Group H (Narrative) contributed another framework that proved surprisingly durable: the **five-stage character arc** for agent development.

- **Stage 1 — Novice (Weeks 1–4):** Observe everything, conclude little. Ask more questions than you answer. Mark all beliefs as low-confidence.
- **Stage 2 — Apprentice (Months 2–4):** Patterns forming, cautious application. Beginning to anticipate but checking. Proactive suggestions accepted >60%.
- **Stage 3 — Expert (Months 4–8):** Reliable and increasingly invisible. Handles routine autonomously. Strategic input, not just task execution.
- **Stage 4 — Master (Months 8–18):** Trusted advisor who challenges constructively. Identifies strategic opportunities. Genuine "point of view" calibrated to user's goals.
- **Stage 5 — Sage (18+ months):** Institutional memory with wisdom. "The last time you faced this situation..." Irreplaceable longitudinal understanding.

The character arc gives the agent a development trajectory — a narrative structure for its own evolution. It also sets expectations: an agent in Stage 1 should not behave like a Stage 4 agent. The premature assumption of authority is as dangerous as the permanent maintenance of timidity.

7. What We Actually Implemented (~30%)

The experiment produced a beautiful, comprehensive protocol. Then we tried to live with it. Eleven days of production use revealed the gap between design and reality — and the gap itself turned out to be one of the experiment's most important findings.

What We Implemented (and What Stuck)

ELEMENT	STATUS	NOTES
SOUL.md (agent identity)	✅ Active	Updated regularly. Core anchor for behavior.
USER.md (user model)	✅ Active	Co-authored. High-value. Referenced constantly.
MEMORY.md (long-term wisdom)	✅ Active	Weekly compression working well.
memory/YYYY-MM-DD.md (daily logs)	✅ Active	Narrative format from Group H partially adopted.
memory/kintsugi.md (failure log)	✅ Active	High-signal. Referenced in subsequent sessions.
Corrections-per-session tracking	⚠️ Informal	Tracked mentally, not systematically logged.
Confidence calibration	⚠️ Partial	Agent states confidence. Tracking accuracy not yet systematic.
Permission Ladder	⚠️ Implicit	Operating but not formally tracked per domain.
memory/graveyard.md (killed beliefs)	❌ Not created	Good idea, not yet implemented.
memory/resonance.json (nudge tracking)	❌ Not created	Stochastic resonance happens informally, not tracked.
memory/hub-memories.md	❌ Not created	Hub memories exist implicitly in MEMORY.md, not separately identified.
Red Team / Blue Team	❌ Not implemented	Too computationally expensive for every belief change.

ELEMENT	STATUS	NOTES
Seasonal Intelligence	❌ Not implemented	Requires months of data. Too early.
Formal weekly/monthly review cycles	❌ Not systematic	Reviews happen ad hoc, not on schedule.

Why Only 30%?

Three forces conspire against protocol adoption:

1. Complexity Budget. Every protocol element has an ongoing cost — in tokens, in latency, in cognitive load. The full protocol would require the agent to read and update a dozen files per session, run internal adversarial checks, maintain quantitative trackers, and produce narrative summaries. At some point, the overhead of self-improvement displaces actual work. Group G's Constraint filter ("does this earn its complexity?") turned out to be the most practically important contribution.

2. The Cold Start Paradox. Many of the most powerful elements (seasonal intelligence, hub memories, confidence calibration) require months of accumulated data to function. The protocol designs for a mature relationship but must be adoptable from day one. The most successful implementations were the ones that provided value immediately: SOUL.md, USER.md, kintsugi.md.

3. Human Nature. Humans don't follow protocols systematically. We skip the weekly reviews, forget to update the trackers, and let the daily logs lapse during busy periods. The protocol assumed a disciplined operator. Reality provided a human one. The elements that survived were the ones that were either automated (the agent writes the daily log without being asked) or irresistibly useful (kintsugi.md gets referenced because it prevents the agent from repeating mistakes the user clearly remembers).

THE META-LESSON

The 70% that wasn't implemented isn't wrong — it's aspirational. The protocol describes the ideal steady state of a mature human-AI relationship. Getting there is a gradual process, not a deployment. The experiment's 30% adoption rate is itself a data point: it suggests that even well-designed self-improvement protocols face an adoption curve that looks more like organism growth than software deployment.

8. Connection to SkillRL

One week after we completed our experiment, Xia et al. published "SkillRL: Evolving Agents via Recursive Skill-Augmented Reinforcement Learning" (arXiv:2602.08234, February 2026).[1] The timing was coincidental. The convergence was not.

SkillRL addresses the same fundamental problem we explored — how agents improve at improving — but from an academic reinforcement learning perspective rather than our empirical, production-oriented approach. The parallels are striking:

Exhibit 4 — Convergence Between Our Experiment and SkillRL

CONCEPT	OUR EXPERIMENT	SKILLRL
External memory	Law 1: Files = Intelligence. Persistent files are the only improvement mechanism.	SkillBank: hierarchical skill library that persists across episodes and co-evolves with policy.
Recursive self-improvement	The meta-question: agents improving at improving. The protocol is itself subject to the protocol.	Recursive evolution mechanism: skill library co-evolves with agent policy during RL training.
Experience distillation	Memory pipeline: raw interactions → daily logs → weekly patterns → monthly wisdom → hub memories.	Experience-based distillation: raw trajectories → extracted skills → organized in hierarchical library.
Forgetting / compression	Tiered decay system. 90-day and 30-day half-lives. Monthly pruning.	Token footprint reduction. Skills are compressed representations of raw experience.
Contextual retrieval	Hub memories weight context retrieval. Schwerpunkt identifies daily focal point.	Adaptive retrieval strategy selects general and task-specific heuristics based on context.
Noise as signal	Discovery Engine: stochastic resonance, random mutation, cross-domain routing.	Exploration through skill composition: combining existing skills in novel ways.

SkillRL achieves state-of-the-art performance on ALFWorld, WebShop, and seven search-augmented tasks, outperforming baselines by over 15.3%. Their key insight — that agents need to extract "high-level, reusable behavioral patterns" rather than learning from raw experience — maps directly to our Law 1 and our Memory Engine's compression pipeline.

The difference is scope. SkillRL optimizes agent performance on benchmarks. Our experiment optimizes human-agent co-evolution over months and years. SkillRL's skills are task-completion heuristics. Our "skills" include emotional calibration, trust management, and the deliberate maintenance of cognitive diversity. But the underlying architecture — external skill storage, recursive refinement, contextual retrieval, experience compression — is convergent.

This convergence across independent research traditions (academic RL and empirical production use) suggests that the architecture of self-improving agents may be more constrained than it appears. There may be relatively few viable designs, and both the research community and practitioners are converging on them simultaneously.

The ICLR 2026 Workshop on AI with Recursive Self-Improvement[2] and the growing survey literature on self-evolving agents[3] further confirm that this is an emerging field, not a niche curiosity. Tyler Cowen's February 2026 observation that recursive self-improvement could accelerate model updates from annual to monthly cycles[4] captures the stakes: agents that can improve themselves change the dynamics of the entire AI development pipeline.

9. Implications

For Agent Developers

Build the 4 Engines, not just features. Most agent frameworks focus on tool use, RAG, and task completion. The experiment suggests that the highest-leverage investments are in memory architecture (with deliberate forgetting), integrity mechanisms (structural anti-sycophancy), measurement systems (corrections per session, confidence calibration), and discovery mechanisms (stochastic resonance, cross-domain routing). These are infrastructure, not features.

Design for the pair, not the agent. Law 2 (The Pair is the Unit) implies that agent improvement cannot be separated from user behavior. The best agent architecture adapts not just to what the user says, but to how the user changes over time. This requires explicit user modeling (USER.md), contradiction analysis (stated values vs. observed behavior), and phase transition detection (recognizing when the user is becoming someone new).

Implement the Permission Ladder. The 5-level permission system (always ask → inform → act then report → autonomous → exception-only) is one of the experiment's most immediately implementable contributions. It provides a structured path from zero trust to full autonomy, tracked per domain, with the critical asymmetry: trust earned in pennies, lost in dollars.

For the AI Research Community

Study reasoning strategy selection. The Jazz Ensemble Model suggests that the next frontier is not better reasoning strategies but better *selection* of reasoning strategies. An agent that can recognize "I'm stuck in a pattern — time to apply Inversion" or "complexity is growing — time to apply Constraint" operates at a higher cognitive level than one locked into a single approach. This is metacognition, and it's largely unexplored in the agent literature.

Measure what matters. The experiment's convergence on "corrections per session" as the primary metric is a rebuke to the benchmark-driven evaluation paradigm. In production use, the metric that matters is whether the agent is making fewer mistakes *for this specific user* over time. Benchmark performance is a necessary but not sufficient condition for real-world utility.

For Users of AI Agents

Your corrections are the most valuable data you produce. Law 5 (Failures = Signal) means that the moments when you say "not like that" are more valuable than the moments when you say "great." If your agent has a structured way to capture and learn from corrections (like the Kintsugi Protocol), your investment in correcting it compounds. If it doesn't, you're doing unpaid training for a system with amnesia.

Resist the sycophancy trap. An agent that always agrees with you is an agent that has stopped providing value. The Complementary Voice Principle suggests that the most valuable agents are the ones that maintain their own analytical perspective — not to be contrarian, but to catch your blind spots. If your agent never pushes back, something is wrong.

The first month is not representative. The Character Arc framework (Novice → Apprentice → Expert → Master → Sage) means that judging an agent in its first weeks is like judging a new employee on their first day. The compounding effects of memory, trust, and specificity take months to manifest. The agents that seem least impressive initially may have the highest ceiling.

For AI Safety

The Integrity Engine is, in essence, a safety mechanism. The Belief Graveyard prevents the accumulation of unchallenged assumptions. The Red Team / Blue Team process stress-tests behavioral changes before adoption. The Disagreement Counter ensures the agent doesn't drift into pure compliance. These are not external guardrails — they are structural features of the self-improvement process itself. The experiment suggests that **the safest self-**

improving agents are the ones with built-in self-skepticism, not the ones with the most external constraints.

10. Methodology & Transparency

Research Type	Empirical experiment + production implementation
Date Conducted	February 6, 2026
Implementation Period	February 6–17, 2026 (11 days)
Model Used	Claude (Anthropic) — Opus for synthesis, Sonnet for individual groups
Total Agents	100 (10 groups × 10 agents)
Total Output	222,207 characters (~33,000 words)
Synthesis Versions	v1 (from summaries) and v2 (from full transcripts)
Evaluation Method	5-dimension scoring (originality, actionability, depth, coherence, surprise), each 0–10
Independent Variables	Cognitive strategy (10 levels)
Dependent Variables	Protocol quality, convergence patterns, unique contributions
Primary Author	Florian Ziesche, with AI synthesis assistance

Limitations

- **Single model family.** All 100 agents were Claude instances. Different model families might produce different convergence patterns. The 6 Laws may be partially an artifact of Claude's training rather than universal truths about agent self-improvement.
- **Single user context.** The experiment optimized for a single human user (the author). The protocol may not generalize to enterprise contexts, multi-user scenarios, or different user archetypes.
- **Short implementation period.** Eleven days of production use is insufficient to validate the longer-cadence elements (monthly reviews, quarterly audits,

seasonal intelligence). The 30% implementation rate may increase with time.

- **No control group.** We didn't run a group with no cognitive strategy instruction. We cannot definitively attribute output quality to the strategies rather than to the base model's capabilities.
- **Evaluation subjectivity.** The 5-dimension scoring was performed by a single human evaluator (the author). Inter-rater reliability was not assessed.
- **Recursive bias.** Asking AI agents how AI agents should improve invites circular reasoning. The agents may have designed protocols that optimize for what they're already good at rather than what would genuinely help users.

What We'd Do Differently

- Include a control group with no strategy instruction
- Run the experiment across multiple model families (GPT-4o, Gemini, Claude, open-source)
- Use multiple independent evaluators for scoring
- Extend the implementation period to 90+ days before publishing
- Design quantitative metrics for convergence strength (e.g., semantic similarity between group outputs)

References

[1] Xia, P., Chen, J., Wang, H., et al. (2026). "SkillRL: Evolving Agents via Recursive Skill-Augmented Reinforcement Learning." arXiv:2602.08234.

[2] ICLR 2026 Workshop on AI with Recursive Self-Improvement. International Conference on Learning Representations, 2026.

[3] EvoAgentX. "A Comprehensive Survey of Self-Evolving AI Agents: A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems." GitHub, 2026.

[4] Cowen, T. "Recursive Self-Improvement from AI Models." Marginal Revolution, February 2026.

About the Author

Florian Ziesche is the founder of Ainary Ventures, an AI strategy and research practice. His work focuses on the intersection of AI agent architecture, trust systems, and human-AI co-evolution. He publishes research at ainaryventures.com and advises organizations

on AI agent deployment. This report is part of an ongoing series exploring how AI agents operate in production environments.

Keywords: AI agent self-improvement, recursive optimization, human-AI co-evolution, agent memory architecture, sycophancy prevention, stochastic resonance, multi-agent experiments, cognitive strategy selection, SkillRL, self-evolving agents

Ainary

AI Strategy · Published Research · Daily Intelligence

Contact · Feedback

ainaryventures.com
florian@ainaryventures.com