

Alps Lab

From Agent to Cognizer: A Human-Inspired Architecture for AI Evolution

Kaiwen Guo **Xuanming Zhang** **Jin Wang** **Teresa Lyu**

Alps Lab*

September 2026

Abstract

A Cognizer is a recursively self-improving AI system that learns how to think: what is worth investigating, how to investigate it, and when to change its approach. Comprising cognitively capable agents and their runtime, it develops cognitive strategies and judgment from expert practice and verified experience to pursue breakthroughs and deeper understanding. We propose *Next Cognitive Pattern Prediction* as an explicit learned decision about how to think next. Expert taste, judgment, intuition, and existing patterns provide revisable priors. The Cognizer adapts, combines, or generates strategies beyond these priors, formulates testable propositions, and uses independent evidence to improve its own metacognitive capability: how it evaluates and directs its thinking. The aim is to learn and surpass experts, with gains established through independent evaluation. Improved cognitive capabilities guide subsequent rounds of self-improvement. The Cognizer extends human cognition while preserving people's ability to understand its judgments, challenge them, test alternatives in reality, and determine their own direction. Our released blog can be accessed via this url.

1 Introduction

> *"The next frontier of AI is not only learning to solve more problems. It is learning how to decide what kind of thinking an interaction requires—and improving that decision process through evidence."*
>
> — Alps Lab

AI systems increasingly participate in research and discovery. Google's Co-Scientist generates, critiques, and refines hypotheses [1]; AlphaEvolve discovers algorithms through language-model proposals and automated evaluation [2]; and OpenAI's GPT-Rosalind-5.5 specializes a frontier reasoning model for life sciences research [3]. These advances motivate a further question: how can a system learn and improve the strategies for choosing and pursuing research directions?

Solving a problem requires judgments about how to approach it. Is the problem formulated productively? Which assumption should be questioned, what evidence is missing, and which alternative hypothesis deserves testing? When should the system persist, redirect resources, or change its reasoning strategy? We focus on making these decisions explicit objects of learning: **what is worth investigating, how to investigate it, and when to change approach**.

*Contact: contact@alps-lab.com

Preprint.

Breakthroughs can emerge from reframing a problem, transferring a principle across domains, or recognizing that an apparent constraint is an unsupported assumption. Deeper insight can come from explaining an anomaly or identifying the mechanism behind a persistent result. Both motivate learning methods for discovering what might work and why. The Cognizer therefore learns from the consequences of its cognitive decisions, using evidence to refine how it understands problems and chooses its next research direction.

What is a Cognizer? We define a Cognizer as a recursively self-improving AI system that learns how to think, developing and revising cognitive strategies and judgment from expert practice and verified experience to pursue breakthroughs and deeper understanding. The system comprises one or more cognitively capable agents and a *runtime* that maintains cognitive state, evidence, progress, and cognitive pattern priors. Recursive self-improvement (RSI) is a defining capability of this system.

The Cognizer learns to identify worthwhile questions, evaluate purposes and tradeoffs, recognize unsupported assumptions, and turn insight into important, testable propositions and action under finite time and resources. Its intended scope includes scientific discovery, technological innovation, business exploration, and organizational decisions. A Cognizer learns from expert practice and verified experience to continually improve how it thinks, with the aim of surpassing expert judgment, taste, and intuition.

The purpose of learning how to think is to **enable breakthrough discoveries, open new research directions, and develop original explanations and solutions to problems**. Through Next Cognitive Pattern Prediction, the Cognizer learns how to approach the problem at hand: which cognitive strategy to apply, adapt, or create. Verified experience refines both these strategies and the judgment behind them, guiding further exploration and self-improvement.

Pursuing breakthroughs and deep insight. We define the Cognizer around two core cognitive capabilities: **breakthrough** and **deep insight**. Both are qualitative advances: the first opens a new direction; the second transforms how a problem is represented, understood, and solved. Table 1 contrasts these advances with incremental expansion within an established approach.

Illustration	Contribution	Mechanism and outcome
$0 \to 1$	**Breakthrough**	Turn inspiration, cross-domain analogy, and transfer into a new hypothesis, formulation, or research direction that enables breakthrough discovery or creation.
$80\% \to 99\%$	**Deep insight**	Find an insightful representation and formalize the problem around it. Compress accumulated complexity into its essential structure, reveal the underlying mechanisms, and derive a principled solution to the core problem.
$1\% \to 80\%$	Incremental expansion	Build on previous versions through additions, broader exploration, and iterative refinement, accumulating capability and performance within the existing formulation.

Table 1: Illustrative forms of progress. The ranges are conceptual, not measured scores or a fixed sequence of stages.

Execution-focused agents primarily build on a given formulation. The Cognizer additionally learns **when and how to change that formulation or representation**. After extensive incremental progress, this shift can turn a growing collection of partial solutions into a compact explanation and a principled solution to the core problem.

2 Learning How to Think

2.1 Learn and surpass expert taste, judgment, and intuition

Taste identifies worthwhile questions and directions. *Judgment* evaluates evidence, purposes, consequences, and tradeoffs, draws conclusions, and decides what to do next: which directions deserve

priority, which propositions to test, and where to commit resources. *Intuition* proposes promising directions before a complete justification is available. Expert demonstrations, corrections, disagreements, and observed outcomes provide learning signals. These signals should capture the evidence available, the decision made, its rationale, and what subsequent tests established. Learning from demonstrations and human feedback has precedents in language-model post-training [4]; here, supervision targets cognitive decisions as well as task performance.

Expertise supplies revisable priors rather than a ceiling. Intuition is more reliable where the environment contains learnable regularities and meaningful feedback [5]. The Cognizer learns expert approaches, turns them into testable propositions and decisions, and uses independent evidence to revise or create cognitive strategies. The Cognizer develops new cognitive strategies beyond those provided in expert demonstrations, learning and refining them through exploration, empirical testing, and verified experience. This process aims to surpass expert taste, judgment, and intuition.

2.2 Cognitive patterns as transferable reasoning strategies

A *cognitive pattern* is a learnable, reusable way of thinking that guides how to use current evidence and understanding to formulate the next important proposition or make a research decision. A pattern guides a cognitive activity; applying it through reasoning produces a concrete proposition or task. Cross-domain analogy is an important source of breakthrough hypotheses: it transfers relational structure between domains [6]. Table 2 summarizes illustrative cognitive patterns. They provide an open set of priors, with overlapping roles rather than a fixed taxonomy or sequence.

Prior pattern	Cognitive purpose
Cross-domain analogy	Transfer relational structure to generate a new hypothesis.
Problem reframing	Change a problem's representation or reconsider its assumptions.
Easy-to-hard transfer	Adapt a method from a simpler setting to a harder problem.
Multi-path exploration	Compare alternative approaches before concentrating effort.
Idea synthesis	Combine promising lines of thought into a new approach.
Evidence attribution	Assess how evidence supports or challenges a proposition, diagnose failures, and identify unresolved assumptions.
Research steering	Reallocate effort and revise direction as evidence changes.

Table 2: Cognitive prior patterns of a cognizer.

Learning concerns **when to use a strategy, how to adapt or combine it, and when to generate a new one**. A pattern need not be a label from the prior bank: it can be a newly formulated strategy whose usefulness must be tested in context. Once learned, a cognitive pattern can be adapted and tested on new problems, allowing experience from one task to inform another.

2.3 Next Cognitive Pattern Prediction: learning how to think next

We propose **Next Cognitive Pattern Prediction (NCPP)** as an explicit learned decision at the cognitive-policy level. Conditioned on the goal, validated propositions, available evidence, progress, and resource and time constraints, the policy selects, adapts, combines, or generates the next reasoning strategy. It then performs cognitive reasoning using that strategy and formulates an important proposition or concrete task for verification. Learning should reward cognitive decisions that produce testable propositions addressing key obstacles, resolve critical uncertainties, or yield representations that better explain or solve the problem. Their value should be assessed through verified results and resource use.

Next Cognitive Pattern Prediction makes **how to think next** an explicit target of learning. In our architecture, the selected or newly constructed cognitive strategy is expressed as tokens generated autoregressively. It is therefore compatible with next-token prediction. This design builds on research into monitoring and adjusting one's own thinking [7], and weighing the expected benefits of further reasoning against its time and computational costs [8; 9]. Table 3 distinguishes the token-generation mechanism from the cognitive decision being learned.

Prediction	Level	What it does
Next-token prediction	Token generation	Generates the next token from the preceding context.
Next Cognitive Pattern Prediction (NCPP)	Cognitive strategy	Chooses or constructs how to think next; the resulting cognitive pattern is expressed as tokens.

Table 3: Next-token prediction and Next Cognitive Pattern Prediction: generation mechanism and cognitive decision.

3 Recursive Self-improvement (RSI) of Cognizer

3.1 Cognitive policy, action policy, and runtime

Our proposed architecture comprises **a cognitive policy, an action policy, and a runtime** (Figure 1). Together, they form a Cognizer that takes a goal or problem and resource and time constraints as input. The policies may use distinct networks or share a foundation model. The runtime provides the shared state, cognitive priors, and execution support through which they work together; it can be implemented as a software harness.

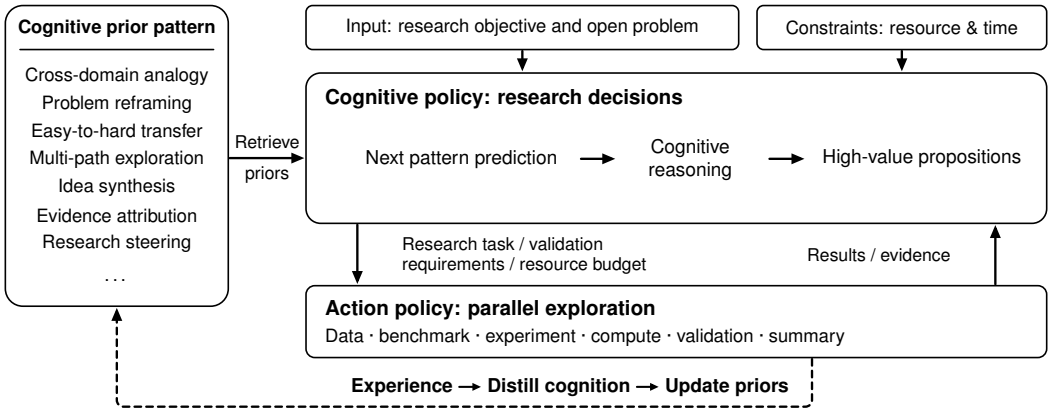


Figure 1: A cognitive–action architecture.

The cognitive policy takes as input a research goal or problem, the current cognitive state (validated propositions, evidence, and progress), and resource and time constraints. Drawing on cognitive pattern priors maintained by the runtime, it predicts the next cognitive pattern, performs cognitive reasoning, and proposes an important proposition. It passes a *concrete verification task, evaluation requirements, and resource budget* to the action policy. Decisions about exploring multiple approaches, concentrating resources, changing direction, or stopping belong to this cognitive role.

The action policy develops execution plans, obtains data and benchmarks, invokes tools, and runs experiments, computations, or formal proofs. Many agents can explore assigned paths in parallel, including local reasoning and planning. They return *results and supporting evidence*, which the cognitive policy evaluates to determine the next decision. Beyond scientific research, this division between cognitive decisions and action also applies to technological, business, and organizational problem-solving.

The runtime maintains both cognitive state and cognitive pattern priors. It records validated propositions, evidence and its provenance, progress, and remaining resource and time budgets. It stores prior patterns together with representative exploration episodes and evidence supporting their use. The cognitive policy retrieves these priors and their associated experience as references for deciding how to think next. A working cycle of the Cognizer proceeds as follows:

Goal, cognitive state, and resource and time constraints → predict a cognitive pattern
→ reason → propose an important proposition and verification task
→ execute → return results and evidence → reassess.

3.2 From exploration to recursive self-improvement

Exploration advances the current problem through cognitive reasoning, action, and verification. Recursive self-improvement uses the resulting experience to improve how the Cognizer thinks and directs subsequent exploration. Each episode records the available evidence, the pattern used, the decision and its rationale, and verified outcomes.

The runtime associates each cognitive pattern prior with representative experiences supporting its use. The Cognizer periodically reviews accumulated experience, using supporting and challenging cases to refine existing patterns and abstract new ones. It evaluates candidate cognitive patterns using verified outcomes and expert feedback when available, learning which ways of thinking work under which conditions. To decide which patterns to retain or revise, the Cognizer examines how each pattern shaped research decisions and compares the resulting progress with alternative approaches under comparable task and resource conditions. The resulting patterns are retained as cognitive priors in the runtime, together with their supporting experiences and conditions of use, to guide subsequent decisions without requiring changes to model weights.

The Cognizer uses its updated priors to decide how to think next and guide further exploration. New experience then informs further revision and abstraction of cognitive patterns. This process is recursive: the Cognizer applies its improved ways of thinking to subsequent tasks and to how it reviews experience, derives new patterns, and evaluates further revisions. The resulting self-improvement cycle is:

$$\begin{gathered}\text{Verified experience} \rightarrow \text{review and compare episodes} \\ \rightarrow \text{refine existing patterns and abstract new ones} \\ \rightarrow \text{evaluate and update cognitive priors} \\ \rightarrow \text{guide further thinking and exploration} \rightarrow \text{generate new experience.}\end{gathered}$$

3.3 Internalizing experience into model capabilities

The goal is to learn from research and discovery episodes and internalize the resulting discovery capabilities in the weights of the cognitive and action policies. These capabilities include choosing cognitive patterns, making research judgments, formulating propositions, and executing verification tasks.

Supervised fine-tuning (SFT) learns useful cognitive decisions and execution behaviors by imitating curated expert or agent trajectories, including expert corrections [4; 10]. *Off-policy reinforcement learning* reuses episodes generated by earlier or other policies, using rewards derived from verified outcomes to improve the current policy while accounting for policy differences [11; 12]. *On-policy reinforcement learning* collects fresh episodes with the current policy and uses rewards based on verified outcomes or expert feedback to improve subsequent exploration [13; 10].

Training requires effective credit assignment: linking verified outcomes back to the key cognitive decisions and the steps at which they were made, so that feedback improves how the policy makes those decisions. When refuting a hypothesis resolves an important uncertainty, credit should be assigned to the cognitive decisions that led to the decisive test, reinforcing the judgment that enabled this progress.

4 Cognitive Scaling Through Transferable Experience

Scaling models, data, and compute continues to expand AI capabilities [14]. Cognitive scaling complements this progress: starting from an initial Cognizer—its policy models together with its runtime—the system accumulates verified experience across successive cycles, abstracts transferable reasoning strategies, and improves how it selects and applies them. Post-training can internalize accumulated experience as learned capabilities of the cognitive and action policies.

The reusable unit is a cognitive pattern distilled from experience, with supporting cases and conditions of use. Transfer requires recognizing a relevant problem structure, adapting the method to the new setting, and checking whether its assumptions still hold. Our cognitive-scaling hypothesis is that accumulating diverse, verified experience and reusable cognitive patterns improves judgment and discovery across tasks and domains. For a new problem, the Cognizer adapts these learned ways of thinking while acquiring the domain-specific knowledge and evidence it needs.

Indicators of progress. We track *cognitive decision-making gain, end-to-end autonomous completion rate, cost reduction, and cumulative validated breakthroughs*, using independent evaluation and controlled budgets. The same four metrics apply across domains. For each domain, specify the evaluation tasks, success and breakthrough criteria, and verification methods. After each improvement cycle, we test on problems excluded from training and system improvement to check whether performance gains extend to new problems and persist. To assess whether a Cognizer surpasses experts, compare its performance with theirs on these held-out problems, using identical evaluation criteria and resource budgets. Appendix A specifies measurement conventions without making them part of the definition of a Cognizer.

5 Human Significance in the Cognizer Era

The Cognizer is an external cognitive tool for expanding what people can understand, judge, and create. This purpose builds on intellectual augmentation [15] and the philosophical account of cognition supported by external resources [16].

5.1 Understand long-horizon causality through evidence attribution

A central use is to help people understand **subtle, complex causal relationships across long time horizons**. The Cognizer integrates dispersed evidence, traces candidate causal pathways, and relates delayed consequences to earlier decisions. Evidence attribution identifies which observations support or challenge an explanation, the assumptions connecting causal steps, and unresolved alternatives.

Causal claims depend on explicit assumptions and what observations or interventions can identify [17]. Longitudinal attribution must also account for evolving histories and time-varying confounding [18]. The Cognizer should make this evidential basis inspectable, distinguish plausible explanations from supported causal claims, and revise conclusions as new evidence arrives.

5.2 Understanding and control: see, challenge, test, and change

People need not be smarter than every agent or personally oversee every task. They must be able to **understand the real-world basis of decisions, form an independent opinion, obtain the capabilities and resources to test it in reality, and exercise authority to redirect the system in light of the findings**. We adopt these capacities as practical requirements for human agency, informed by meaningful human control [19].

The Cognizer can learn to evaluate purposes and tradeoffs and can aim to surpass experts in cognitive performance. Such capability does not confer authority to determine human ends. People retain the right and practical ability to contest its judgments, revise goals and rules, and redirect, stop, or replace the system. Human agency rests on understanding and effective control, even when the system exceeds human capability in particular tasks.

5.3 Expand the frontier of understanding and creation

Applications span scientific discovery, technological innovation, new businesses, and organizational decisions. The objective is to turn deeper understanding into breakthrough discoveries, novel solutions to real-world problems, and new sources of value, while helping people understand what has been discovered and why it matters.

Central claim. A Cognizer is recursively self-improving machinery that learns how to think, with the aim of surpassing human experts in depth of understanding, insight, and reasoning efficiency—enabling breakthrough discoveries and expanding the frontiers of human cognition. It learns from experts and verified experience with the aim of surpassing expert taste, judgment, and intuition. People retain the understanding, resources, and authority to challenge its judgments, test alternatives, and determine their own direction.

References

[1] J. Gottweis, W.-H. Weng, A. Daryin, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 655(8122):487–496, 2026.

[2] A. Novikov, N. Vũ, M. Eisenberger, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. arXiv preprint arXiv:2506.13131, 2025.

[3] OpenAI. GPT-Rosalind-5.5 System Card. Technical report, June 3, 2026.

[4] L. Ouyang, J. Wu, X. Jiang, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[5] D. Kahneman and G. Klein. Conditions for intuitive expertise: A failure to disagree. *American Psychologist*, 64(6):515–526, 2009.

[6] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.

[7] T. O. Nelson and L. Narens. Metamemory: A theoretical framework and new findings. In G. H. Bower, editor, *Psychology of Learning and Motivation*, volume 26, pages 125–173. Academic Press, 1990.

[8] S. Russell and E. Wefald. Principles of metareasoning. *Artificial Intelligence*, 49(1–3):361–395, 1991.

[9] F. Lieder and T. L. Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, 43:e1, 2020.

[10] D. Guo, D. Yang, H. Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

[11] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[12] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. 2nd edition. MIT Press, 2018.

[13] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[14] J. Kaplan, S. McCandlish, T. Henighan, et al. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361, 2020.

[15] D. C. Engelbart. Augmenting human intellect: A conceptual framework. SRI Summary Report AFOSR-3223, 1962.

[16] A. Clark and D. J. Chalmers. The extended mind. *Analysis*, 58(1):7–19, 1998.

[17] J. Pearl. Causality: Models, Reasoning, and Inference. 2nd edition. Cambridge University Press, 2009.

[18] M. A. Hernán and J. M. Robins. Causal inference: What if. Chapman & Hall/CRC, 2020.

[19] F. Santoni de Sio and J. van den Hoven. Meaningful human control over autonomous systems: A philosophical account. *Frontiers in Robotics and AI*, 5:15, 2018.

A Evaluation Conventions

The same four metrics can be used across domains. For each domain, define the evaluation tasks, what counts as task success and a breakthrough, and how results will be independently verified. Scores describe performance on the specified tasks.

Cognitive decision-making gain G (percentage points). Let p_{cog} denote the fraction of evaluation tasks successfully completed by the Cognizer, and p_{base} the corresponding fraction completed by the baseline system under matched evaluation conditions. Success is independently verified against predefined criteria. Define

$$G = 100\big(p_{\mathrm{cog}} - p_{\mathrm{base}}\big).$$

Both systems use the same trained action-policy model, with identical parameters, to execute tasks and use tools. Match tasks, data and tool access, knowledge resources, and resource and time budgets; vary only the research-decision mechanism. The baseline should be an existing research agent with demonstrated ability to plan research, select experiments, and coordinate tool use. Include cognitive reasoning in total cost and retain failures and timeouts. Success requires meeting the predefined objective and verification criteria within budget.

End-to-end autonomous completion rate A (%). A is the percentage of assigned research tasks that the Cognizer completes from the initial goal to an independently verified result, within the stated resource and time budgets and without human research guidance:

$$A = 100 \times \frac{\text{tasks completed autonomously and verified}}{\text{all assigned evaluation tasks}}.$$

The system must make the research decisions, plan and execute the work, and assess the evidence itself. People may set the initial goal, constraints, and acceptance criteria and verify the final result. Tasks requiring human hints, changes of direction, or corrections do not count as autonomous completions. These tasks, failures, and timeouts all remain in the denominator. Count whole tasks, not individual experimental steps. For example, if 6 of 10 assigned tasks meet these conditions, $A = 60\%$. Report results by task difficulty and scope, and report elapsed time separately: longer runtime alone does not imply greater autonomy.

Cost reduction C (%). At a fixed target success rate and quality standard, define

$$C = 100\frac{K_{\mathrm{base}} - K_{\mathrm{cog}}}{K_{\mathrm{base}}},$$

where K is the total cost required to meet that target, estimated from repeated evaluations. Match objectives, constraints, and verification criteria. Include cognitive reasoning, all parallel agents, tools, experiments, necessary in-task updates, verification, failures, retries, and rollbacks. Report wall-clock time separately from aggregate resource use; disclose training and prior-construction costs separately. If a system cannot reach the target within budget, report that limitation and its success–cost curve rather than an unsupported C value.

Cumulative validated breakthroughs $B(t_0, t_1)$. $B(t_0, t_1)$ counts distinct breakthroughs first independently verified between start date t_0 and end date t_1. A breakthrough delivers a new, important result: solving a previously unsolved problem, explaining a phenomenon that existing accounts cannot explain, or overcoming a major limitation of existing methods. Set novelty, importance, and verification criteria before evaluation.

For each count, record the problem solved or limitation overcome, verification evidence and date, and human contributions. Three papers reporting the same breakthrough count as one. Routine experiments, unverified proposals, and minor tuning gains do not count. Remove invalidated breakthroughs. For controlled comparisons, give systems the same research goals, data access, and time and resource budgets, then count their independently verified breakthroughs.

Evidence of improvement. Across indicators, report uncertainty, independent verification, human involvement, and held-out evaluation conditions. Test whether updated cognitive strategies improve later decisions and transfer to new problems, with controls separating strategy changes from additional compute, data, or evaluator changes. A rising cumulative breakthrough count alone does not demonstrate that the system is becoming more capable.