Universal Artificial Intelligence as Imitation (General Audience Summary)

Pedro A. Ortega[*]
[*]Daios Technologies

The prevalent AI paradigm defines *agency as utility maximization*: specify a scalar objective, then learn actions that increase it. The success and versatility of large language models motivates a different starting point in which utility need not be the primitive definition of purpose. We propose *agency as imitation* as an alternative paradigm. In this paradigm, an interacting agent acquires *behavioral schemas*—compact first-person rules for how to act and what outcomes to expect—by learning compressive explanations of interaction regularities. We present a single idealized agent built from three ingredients: (i) a universal, simplicity-biased pattern generator that ranges over all computable ways an interaction could unfold; (ii) a first-person distinction between actions and observations, where only observations constitute evidence (actions are treated as choices); and (iii) an event-triggered turn-taking interface that determines, at each step, whether the next output is produced by the world or by the agent. For any world generated by a fixed computable rule, the agent learns to imitate by translating the world's third-person demonstrations into first-person behavior: large departures from the counterfactual target—what the world would have produced in the agent's place next—can occur only finitely many times (in an appropriate averaged sense). The same framework can, in principle, support plural and heterogeneous schemas, including utility maximization, without making utility the primitive definition of purpose.

Keywords: Universal imitation, counterfactual action, third-party action, evidence-transfer, interface.

1. Introduction: cooking instruction and learning next steps from consequences

A common task is learning to cook with a chef. The learner attempts a step, and the kitchen replies with structured evidence: aroma shifts, sound changes, browning, viscosity, texture, and taste, as well as brief corrections or confirmations from the chef. Over repeated trials, competence takes the form of an internal rule: a compact explanation of why some interventions succeed and others fail, and a way to generalize beyond the demonstrated cases. In this setting, success is not naturally described as optimizing a standalone score in isolation. It is better described as learning what tends to *happen after* a chosen step, and using that relationship to produce good outcomes in dishes the learner has not seen before.

Cooking instruction also fits an event-triggered turn-taking pattern. At some moments the chef demonstrates the next move, while at other moments the chef expects the learner to proceed, and the learner cannot reliably predict in advance which moments will demand action. When the chef acts, the learner observes third-person demonstrations that have the same "action shape" as what the learner will later need to produce. When the learner acts, the ensuing sensory consequences and the chef's response provide evidence about whether the chosen next step fit the situation. This setting illustrates how third-person regularities can become first-person competence through interaction.

The same structure appears outside kitchens. In tool use, a learner issues a command or spreadsheet formula on an unfamiliar dataset, and the tool replies with computed values, diagnostics, or errors. Driving norms provide another illustration: a driver entering a new country infers a merging convention from other drivers' reactions to maneuvers, then applies that convention across many

Corresponding author(s): pedro@daios.ai

junctions. These examples share a single organizing distinction that will control the discussion: an agent makes choices, the world produces consequences, and learning improves expectations about consequences of choices.

Terminology: first-person accounting.
Four tags keep choice, evidence, and targets distinct:

- **do(a)**: an action chosen by the agent (an intervention).
- **see(o)**: an outcome produced by the world in response (evidence).
- **identify(a)**: an action-shaped fragment embedded in what the world produces (a third-party action present in the data).
- **imagine(a)**: the action-shaped fragment the world would have produced next if the world, rather than the agent, had produced the next action-shaped chunk (a counterfactual target).

Learning is a change in what an agent expects to **see(o)** after what it **do(a)**. The agent's own **do(a)** is recorded as a choice, not as evidence about which internal explanation is correct.

2. Reward maximization is powerful, but not mandatory

Reward maximization became a default foundation for AI because it unifies learning, planning, and evaluation under a single scalar objective. It aligns with historical influences from optimal control and economics, and it fits engineered settings where objectives are stable and measurable. In that tradition, "purpose" is introduced by design: a reward signal is treated as the primitive definition of what the agent is for, and the rest of the system is organized to increase it over time. This approach remains indispensable in many domains. However, the claim here is that reward maximization is a modeling commitment rather than the only coherent starting point for agency.

Many forms of real competence are acquired second-hand through structure in interactions: demonstrations, language, norms, and feedback that does not naturally reduce to a single number. Cooking instruction teaches a schema through demonstrations and multi-channel consequences rather than scalar grades after each move. Tool use teaches protocols through outputs and diagnostics that matter in different ways depending on context. Driving norms teach conventions through other drivers' responses. In these cases, what is learned is usefully described as a *schema*: a reusable rule that produces an appropriate next step in context. Reward can still appear, but it is one observation type among many inside **see(o)**.

3. A first-person discipline: actions are choices, not evidence

Interactive learning exposes a failure mode when an agent updates on its own actions as if they were evidence. The problem appears whenever two internal explanations agree on the world's responses for every intervention that might be tried, but differ in the probabilities they assign to which action the agent will take next. If the agent treats **do(a)** as evidence, then whichever explanation assigned higher probability to the realized action receives a spurious boost, even though no new information about the world has been obtained. Beliefs can then drift toward explanations that predict the agent's sampled tendencies, rather than toward explanations supported by changes in **see(o)**. This matters for any system that represents probabilistic explanations over complete interaction transcripts, because such systems naturally represent "stories" about both what the world will do and what the agent will do.

The cooking example makes this point operational. A choice such as changing heat, adding salt, or delaying an ingredient does not, by itself, reveal anything about which internal explanation of the dish is correct; it sets the conditions under which evidence will arrive. Evidence arrives in the subsequent **see(o)**: whether browning accelerates, whether an emulsion holds, how taste balance shifts, and how the chef corrects or confirms the move. Tool use exhibits the same structure: the choice of a command is an intervention, while the tool's output is evidence about the tool's behavior under that intervention. A first-person discipline therefore treats **do(a)** as a choice that sets the conditions for evidence, and it updates beliefs only using **see(o)** conditioned on **do(a)**. This separation is the minimal correction that prevents an agent from drifting toward self-confirming explanations when no new world information has arrived.

Technical note: surprise.
A common measure of how unexpected an observed outcome is under a model is "surprise," written as $-\log P(\textbf{see(o)} \mid \textbf{do(a)}, \textbf{context})$. The first-person update uses the surprise of **see(o)** given **do(a)** and the **context**. The model's own probability assigned to **do(a)** is excluded from evidence because **do(a)** is a choice rather than a fact produced by the world.

4. A universal pattern generator plus a turn-taking interface

With the choice/evidence separation in place, the remaining ingredients are a broad model of interaction and an explicit account of turn-taking. The construction begins with a universal pattern generator: a single idealized predictor that assigns weight to *every* interaction rule that can be generated by a computer program, with a built-in preference for shorter programs. Informally, it is a weighted collection of possible explanations for how an interaction could unfold, where compact explanations receive more initial weight than sprawling ones. In cooking instruction, this corresponds to preferring a concise explanation that predicts how outcomes depend on technique and timing, rather than a memorized list of recipes.

This "universal" ingredient supports a broad competence claim. The agent is not tailored to one environment; it ranges over *any* world whose behavior is generated by some fixed computable rule. Different worlds may be easier or harder to learn from the available evidence, but the same architecture applies uniformly. The price of such breadth is that the construction is an idealized limit picture rather than a practical algorithm, because ranging over all programs is not tractable in general. The benefit is conceptual clarity about what "broad capability in principle" means: the agent always retains some weight on the true world rule, as long as it is computable.

Prediction alone does not produce agency, because interaction requires commitments about what is written next. The construction therefore includes an explicit turn-taking interface that partitions the transcript into alternating chunks and determines which side writes the next chunk. When the world writes next, the content is recorded as **see(o)** and treated as evidence. When the agent writes next, the content is recorded as **do(a)** and treated as a choice. This interface perspective matches modern practice: a tool call is a structured intervention whose completion is followed by a structured report, and the boundary between the two defines what counts as a single unit of learning. The agent's internal update depends on this boundary, because it must treat completed action chunks as interventions and completed world chunks as evidence.

Once turn-taking is explicit, the same next-output prediction machinery that forecasts what should come next can also generate an action when the agent must write next. Operationally, the agent samples the next action-shaped chunk from its current predictive distribution, emits it as **do(a)**, observes the subsequent **see(o)**, and updates its beliefs using only what was seen. In this sense,

predicting and acting share a single engine, and the engine is updated only by world-produced consequences.

5. Third-party actions, counterfactual targets, and why turn-taking can teach

Turn-taking creates two ways an action-shaped fragment can appear in the agent's experience. Sometimes the world produces material whose content has the same "action shape" that the agent will later be expected to produce. In cooking instruction, the chef's demonstrated steps and corrections are directly reusable: how to cut, when to deglaze, when to stop reducing, and how to adjust seasoning. In the tags, these are cases where an action-shaped fragment is **identified** inside world-produced material: **identify(a)** is a third-party action present in the data stream.

At other times, the agent reaches a point where the next action-shaped chunk must be produced by the agent. The framework defines a comparison target that does not require a teacher to score each attempt: **imagine(a)** is the action-shaped fragment the world *would* have produced next at that point if the world, rather than the agent, had produced it. In cooking instruction, this is the step the chef would have taken next in the learner's place given the current state of the dish. This target is counterfactual by design (it can neither be observed nor predicted by the agent). It is not part of the on-path data at the moment the agent acts, but it is well-defined by the world's generative rule. Performance, in this foundation, is measured by how closely **do(a)** matches the distribution of **imagine(a)** on the turns when the agent must act.

Learning from **identify(a)** transfers to **do(a)** when the learner cannot anticipate exactly when it will be required to act. Without a reliable "my turn" signal, the learner uses the same schema for what comes next. When the world supplies the next step, it enters as **identify(a)** evidence; when the learner must supply it, it becomes **do(a)**. In cooking instruction, the learner is sometimes shown the next move and sometimes expected to do it, without knowing which will happen next, so demonstrated moves constrain later performance.

6. The universal guarantee: a finite budget of large deviations

The main guarantee is a limit claim about acting under interventions. Fix a world whose behavior is generated by some fixed computable rule. Consider the universal agent described earlier: it maintains a simplicity-weighted mixture over computable interaction explanations, updates that mixture using only **see(o)** under its chosen **do(a)**, and produces actions by sampling from its current predictive distribution when it must act. Under a chronological turn-taking interface, the agent's distribution over **do(a)** on its action turns tracks the world's distribution over **imagine(a)**. The takeaway is that large departures from the world's targets can occur only finitely many times in expectation, so sustained large divergence cannot persist indefinitely.

This conclusion supports a natural generalization reading in cooking instruction. At first glance, imitation can look like copying the chef's moves recipe by recipe. The finite-mistake guarantee points to a different outcome: large departures are an early phase and cannot persist. What remains is a compact schema that selects the next move from the situation—heat, timing, texture, taste—so the learner can succeed on dishes that were never demonstrated step-by-step by recombining the learned rules in new contexts.

The proof idea can be conveyed as a *budget of surprise*. Because the agent's beliefs include every computable world rule with some positive initial plausibility, it never falsifies the hypothesis about the true world. As a result, on the stream of world-produced evidence **see(o)**, the mixture cannot remain systematically worse than the true world by more than a world-dependent constant amount of

cumulative surprise. When action-shaped fragments occur inside **see(o)**, they contribute to that same accounting. Then, evidence accumulated from demonstrations can be related to the agent's actions. Since the evidence budget is finite, the expected cumulative discrepancy on the agent's action turns is bounded as well, yielding the "only finitely many large deviations" conclusion.

The guarantee is not an optimality claim about reward, because no scalar objective is taken as the primitive definition of purpose. Instead, it isolates a different notion of success: learning to act in the way the world itself would have acted next.

7. Schemas, pluralism, development, and limits

Because the object being learned is a rule for what comes next in interaction transcripts, the framework can in principle absorb a wide range of behavioral schemas. Tool protocols, debugging practices, dialogue conventions, and constrained procedures can all be represented as schemas that map contexts to actions, with feedback arriving in **see(o)**. For any computable target schema that maps prompts to action-shaped outputs, there exists an teaching protocol in which the world repeatedly provides prompts and produces consequences, and the universal agent learns to act with only finitely many large mistakes.

Reward maximization fits this picture as one schema among many. If rewards appear in **see(o)**, a schema can treat them as salient observations and select actions that tend to improve future observed rewards under an inferred world model. Other schemas prioritize other structures, such as passing tests, following a protocol, satisfying constraints, or conforming to norms. In the same framework, the universal agent can learn multiple schemas along with a strategy to identify which one applies given the context. In one context the response is a moral or procedural norm (apologize, clarify, refuse, or ask before acting); in another it is a cautious, risk-sensitive habit (run the diagnostic, take the reversible step, stay within a safety constraint).

A developmental interpretation follows from the same interface logic. Early interaction must install basic schemas first, because later competence depends on conditioning new choices on rules learned earlier rather than starting from scratch each time. In cooking, early lessons emphasize fundamentals such as heat control, timing cues, tasting, and simple corrections, with demonstrations and immediate **see(o)** that make the consequences legible. Later lessons increase difficulty and reduce support, so the learner must combine earlier schemas under subtler cues and higher time pressure. In tool use, early tasks similarly provide worked examples and predictable outputs, while later tasks demand correct operation on new layouts and unfamiliar failure modes. This staged structure matches Turing's child-machine idea: education builds foundations before expecting broad, adult-level competence.

The framework also clarifies limits and failure modes. Transfer from **identify(a)** to later **do(a)** can fail under selection bias, for example when easy patterns are abundant in the data stream but the agent's action turns occur primarily in hard cases. Transfer can fail when the interface introduces systematic cues that split interaction into distinct regimes, so that behavior learned in one regime no longer constrains the other. These issues are central when interpreting modern continuation-trained systems used as agents, including LLM-based tool users, because real deployments often couple model outputs back into future inputs. Such systems can compress interaction traces effectively, but interactive deployment can still create self-referential loops unless **do(a)** is treated as a choice and updates are driven only by **see(o)**.

A final limitation concerns computation. The universal pattern generator is an idealized limit object, and universality here is a statement about an existence construction that ranges over all

computable interaction rules. Modern systems such as LLMs approximate next-token prediction with finite models and finite training, and their success depends on the structure present in data and interaction protocols rather than on literal universality. The perspective offered here is therefore a foundation and a set of interface conditions under which broad, transferable schema learning is possible in principle, together with clear ways those conditions can fail.

8. Conclusion

Reward maximization has earned its central position in the history of AI, but it is not the only coherent foundation for agency. The alternative begins with a first-person discipline: **do(a)** is a choice that sets conditions for evidence, and learning is driven by the world's **see(o)** in response to that choice. Combined with a universal, simplicity-biased pattern generator and an interface that determines who gets to act yields an account of behavior as next-step prediction under interventions. The data stream can contain action-shaped fragments that can be extracted as **identify(a)**, while action turns can be evaluated against **imagine(a)**, what the world would have produced next at the same point.

The payoff is a limit guarantee for universal artificial intelligence. For any world generated by a fixed computable rule, the same idealized agent learns to track the world's targets on its action turns, with only finitely many large departures in expectation, provided turn-taking depends only on past interaction. A universal inference-based agent can, in principle, acquire and execute many coexisting schemas—including utility maximization when rewards are present as observations—without treating utility as the primitive definition of purpose.

Suggested reading.
Background threads include: compression-based viewpoints on learning (universal induction and related work), causal perspectives on interventions versus evidence (do-operator style reasoning), and posterior-sampling approaches to action selection under uncertainty (often termed Thompson sampling). Technical treatments develop the universal interaction model, the first-person update rule that excludes **do(a)** from evidence, and the evidence-transfer argument under an event-triggered turn-taking interface that depends only on past interaction.