Universal Artificial Intelligence as Imitation

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

Modern AI often defines agency as reward maximization: specify an objective, then learn to optimize it through interaction. This paper argues for an alternative foundation in which agency is inference: purposeful behavior emerges from learning compact generative explanations of how outcomes depend on chosen interventions. We extend universal induction to interactions by placing a Solomonoff universal mixture over computable generators of complete action–observation histories, with a crucial epistemic rule: actions are interventions, not evidence, so beliefs update only through the world's responses to what the agent does. The resulting posterior over generators is a first-person belief state, and behavior follows from sampling from the posterior predictive over actions (probability matching). To connect "what happens" to "what the agent should do," we formalize a counterfactual target action: the action the world would have emitted in the agent's place. We prove a finite cumulative divergence bound between the agent's actions and these counterfactual actions, implying only finitely many large deviations (and finitely many mismatches for deterministic targets). We also show that the agent can be taught to behave like any computable function and in particular to learn arbitrary computable behavioral schemas—including reward maximization. In this view, rewards are one kind of observation among many—alongside demonstrations, language, tool outputs, and feedback—rather than the primitive definition of purpose.

Keywords: Solomonoff induction, universal imitation, causal interventions, adaptive control.

1. Introduction

Reinforcement learning has become the default language for "purposeful" behavior in machine learning: an agent is modeled as maximizing expected cumulative reward [15, 41]. This is a powerful formalization, but it is also a historically contingent modeling choice. The word *reinforcement* comes from psychology's operant-conditioning picture of behavior shaped by consequences, while the modern mathematical template inherits the optimal-control and economics view that "purpose" is the optimization of a specified objective [1, 33]. Because this template is so successful, we rarely ask a prior question: *must* purposeful behavior be characterized as reward maximization?

A competing intuition is both older and more general: much of intelligent behavior is acquired *second-hand*. Animals and humans do not learn only from first-person trial and error; they learn by absorbing patterns in what others do and say. In the broadest sense, *imitation* is the uptake of a *pattern*—a regularity in behavior, language, demonstration, analogy, or metaphor—and its conversion into a first-person capacity to act. This is not merely "copying actions" but schema acquisition: the learner internalizes a generative rule that can be applied in new situations, often without ever observing an explicit reward. A child learns how to apologize, how to take turns, and what counts as "rude"; a novice learns laboratory practice; a driver learns norms of merging. These are not fixed objectives revealed by scalar feedback. They are behavioral schemas and justifications acquired on the fly from observed structure.

Corresponding author(s): pedro@daios.ai

Imitation as continuation. In imitation-style settings, the learner often receives third-person demonstrations but no explicit reward or teacher corrections. In such cases, learning is best read as *continuation under a schema*: after observing a structured prefix (examples, demonstrations, instructions), the learner must itself produce the next action that extends the demonstrated regularity. This is why here we will represent each hypothesis as a *joint interaction generator* (inducing both an action channel and an observation channel): imitation requires hypotheses that can generate behavior, not only predict world responses.

Seen this way, imitation is inseparable from compression. To imitate a pattern is to find a short program that generates it: a reusable explanation that captures what is essential and discards what is incidental. Language is a particularly powerful carrier of such patterns: a verbal description can specify a behavior never before encountered; an analogy can transfer a schema from one domain to another; a metaphor can compress a complex policy into a memorable rule. In all these cases, the learner acquires not only actions but also the *reasons* that make the actions make sense in context. Purposeful behavior, on this view, is grounded in learned generative structure rather than in a pre-declared reward signal.

The obstacle is that third-person patterns do not automatically translate into first-person competence. The reason is causal. From a first-person perspective, *actions are choices* and *observations are evidence*. In third-person data, however, the demonstrator's "actions" are themselves observations for the learner: they are entangled with the demonstrator's information and latent intentions. Treating observed actions as if they were first-person interventions produces a characteristic error: updating beliefs as if self-generated actions were evidence about which hypothesis is true ("learning from one's own actions") or, in imitation, predicting consequences using $P(Y \mid X)$ when what one needs is $P(Y \mid \mathrm{do}(X))$ [27, 28, 30, 32]. Translating a pattern from "what *they* do" to "what happens when *I* do it" requires an explicit intervention/evidence asymmetry.

This paper develops a universal account of interactive learning that makes this asymmetry primary and ties it directly to compression. Solomonoff induction constructs a universal predictor by mixing over all computable hypotheses with weights given by description length [20, 22, 38, 39]. To extend this universality to agents, the object of prediction must be interaction histories, not passive observation strings. Therefore, we construct a Solomonoff-style universal semimeasure directly on interactions. This yields a single universal joint model over alternating action–observation sequences: a universal compressor of interaction patterns.

A joint universal model is still not a first-person agent. The agent–environment distinction is enforced epistemically by the belief update rule: only observations contribute evidence, while actions do *not* [27, 28]. Thus, hypothesis weights are updated only by observation likelihoods, while action probabilities are excluded from evidence. The resulting *intervention posterior* is a universal first-person belief state: it is the agent's compressed explanation of "how the world responds when I do things." Behavior then follows by mixing (or probability matching) hypotheses' induced action channels under that posterior. In this sense, purposeful behavior can be grounded in universal inference under interventions—with reward maximization appearing as one optional way of selecting among schemas, rather than as the defining semantic core.

Contributions.

- **Universal induction for interaction:** We extend Solomonoff's mixture idea from passive strings to *action–observation transcripts*, by mixing over computable programs that generate complete interactive histories [22, 38, 39].
- **First-person learning rule:** We give an explicit belief update in which the agent's own actions

are treated as *choices* rather than evidence. Beliefs update only through the world's responses to what the agent does, avoiding "learning from one's own actions" [27, 28, 30].

- **Behavior from beliefs:** We derive a simple control rule: at each step, act by mixing (or sampling) the action tendencies implied by the currently plausible programs [27, 28].
- **Interface and continuation targets:** We formalize variable-length turns with a computable gate that defines token boundaries, and use it to define what the world *would have written* at a would-be action slot, and when the same continuation is instead revealed by the world [32].
- **Universal imitation guarantee:** In a protocol where such continuations are sometimes revealed and sometimes demanded, we prove a finite cumulative divergence bound to any computable target continuation on the realized history; hence large deviations occur only finitely often, and for deterministic targets the expected number of mismatches is finite [20, 22, 38, 39].

2. Setup and the Universal Prior

We start by defining the agent's prior beliefs over possible generators of a single-stream substrate. Following Solomonoff [38, 39], we avoid committing to a parametric family and instead mix over computable generators, with an inductive bias toward shorter descriptions. Here we work with generators presented by explicit *one-step rules* $\nu(\cdot \mid \cdot)$, because the interaction setting will require conditioning on, and intervening into, these conditional rules directly.

Throughout, we will use the following shorthand notation for sequences: $x_{n:m} = x_n, x_{n+1}, \ldots, x_m$ for $n \leq m$, $x_{\leq n} = x_{1:n}$, and others such as $x_{<t}$ are defined in the obvious way. We also underline symbols to glue them together, so $\underline{ao}_{\leq t} = a_1, o_1, \ldots, a_n, o_n$. Given an alphabet Σ, Σ^∞ and Σ^* denote the infinite sequences and finite strings over the alphabet.

Single-stream substrate. We propose a single infinite symbol stream

$$x_{1:\infty} \in \Sigma^\infty,$$

over a fixed finite *base* alphabet Σ (for instance, $\Sigma = \{0, 1\}$). This stream will serve as a substrate for interactions.

Lower-semicomputable semimeasures (kernel-induced). Let $\mathcal{M}$ be the class of semimeasures on Σ^* specified by *one-step rules* as follows. We specify $\nu \in \mathcal{M}$ by assigning, for every finite prefix $h \in \Sigma^*$ (including the empty string ϵ) and symbol $x \in \Sigma$, a nonnegative number $\nu(x \mid h)$ such that

$$\sum_{x \in \Sigma} \nu(x \mid h) \;\leq\; 1 \qquad \text{for all } h \in \Sigma^*.$$

(Any deficit is the chance the generator stops after outputting x.) Given these one-step rules, define the mass of any finite prefix by the chain rule: for $x_{1:n} \in \Sigma^n$ define

$$\nu(x_{1:n}) := \prod_{t=1}^{n} \nu(x_t \mid x_{<t}),$$

where $x_{<t} := x_1 \cdots x_{t-1}$. Finally, *lower-semicomputable* means: there is a program that, given (h, x), outputs an increasing sequence of rationals converging to $\nu(x \mid h)$, uniformly over inputs. (This "kernel-first" viewpoint is also the natural setting for Levin's conditional a priori probability, which is defined directly on such conditional rules rather than by taking ratios of prefix masses [20].)

Stop convention (completing missing mass). Whenever a one-step rule assigns total mass < 1 over its next-symbol or next-token set, we treat the deficit as an explicit extra outcome $\perp$ ("stop"). If a rule $\nu(\cdot \mid h)$ ranges over a countable set Z and $\sum_{z \in Z} \nu(z \mid h) \leq 1$, define the completed rule on $Z \cup \{\perp\}$ by $\bar{\nu}(z \mid h) := \nu(z \mid h)$ for $z \in Z$ and $\bar{\nu}(\perp \mid h) := 1 - \sum_{z \in Z} \nu(z \mid h)$.

KL and TV (countable case). For probability distributions P, Q on a countable set Z, define

$$D_{\mathrm{KL}}(P \| Q) \;:=\; \sum_{z \in Z} P(z)\, \log \frac{P(z)}{Q(z)},$$

with the conventions $0 \log(0/q) = 0$ and $D_{\mathrm{KL}}(P \| Q) = \infty$ if $P(z) > 0$ but $Q(z) = 0$ for some z. Define total variation distance by

$$\mathrm{TV}(P, Q) \;:=\; \sup_{E \subseteq Z} |P(E) - Q(E)| \;=\; \frac{1}{2} \sum_{z \in Z} |P(z) - Q(z)|.$$

When Q comes from a semimeasure rule (total mass ≤ 1), TV and D_{KL} are understood between the completed rules $\bar{P}, \bar{Q}$ on $Z \cup \{\perp\}$ (and a measure assigns probability 0 to $\perp$). Throughout, log is the natural logarithm.

The universal prior. Fix an effective listing $(\nu_p)_{p \in \mathbb{N}}$ of $\mathcal{M}$ (so every $\nu \in \mathcal{M}$ appears at least once), and fix positive weights $(w(p))_{p \in \mathbb{N}}$ with $\sum_{p=1}^{\infty} w(p) \leq 1$. We interpret $w(p)$ as the agent's *universal prior* weight on generator p, with a built-in bias toward shorter descriptions (for example, choose a prefix-free code for indices p and set $w(p) = 2^{-|p|}$) [22]. Define the *universal semimeasure* on substrate prefixes by

$$M(x) := \sum_{p=1}^{\infty} w(p)\, \nu_p(x), \qquad x \in \Sigma^*.$$

Then M is itself a lower-semicomputable semimeasure on Σ^*, and it *dominates* every $\nu \in \mathcal{M}$: for each ν there exists a constant $c_\nu > 0$ such that

$$M(x) \;\geq\; c_\nu\, \nu(x) \qquad \text{for all } x \in \Sigma^*.$$

This dominance property is the sense in which M is *universal*: up to a fixed constant factor, it assigns at least as much mass to every substrate prefix as any computable generator in the class [22]. Note that this universality is relative to $\mathcal{M}$ as defined above: $\mathcal{M}$ is strictly smaller than the full class of lower-semicomputable semimeasures on Σ^*, because a lower-semicomputable prefix-mass function need not admit uniformly lower-semicomputable one-step conditionals [20].

3. Interaction Systems

We now make the interaction protocol explicit. The guiding intuition is simple: the agent and the world write into a shared substrate stream, but *who* gets to write next is decided by an interface rule. From the agent's first-person perspective, what it writes are *choices* (and therefore not evidence), while what the world writes are *outcomes* (and therefore evidence).

Definition 1 (Interaction system)**.** An *interaction system* is a tuple $(\Sigma, \Gamma, \pi, \mu)$ where:

- Σ is a fixed finite base alphabet and the substrate is a single stream $x_{1:\infty} \in \Sigma^{\infty}$.

- Γ is a *gate* that produces a binary process $\gamma_1, \gamma_2, \ldots$ indexed by substrate position, where $\gamma_k = 1$ means the agent writes the next substrate symbol at position k and $\gamma_k = 0$ means the world writes it.
- π and μ are (semi)measures on Σ^* (typically lower-semicomputable semimeasures) used as symbol-level generators for the agent and world, respectively, when they hold the gate. Each comes with a chronological one-step decomposition in the sense of Section 2: at each substrate position, the next symbol is drawn from a conditional distribution given the written prefix so far (and the already-emitted prefix of the current block).

Gating and token boundaries. The gate Γ is a *chronological* (non-anticipating) and *computable* rule for producing a binary process $\gamma_1, \gamma_2, \ldots \in \{0, 1\}$ indexed by substrate position. We fix $\gamma_1 = 1$ so the agent writes first, and for each $k \geq 1$ we sample

$$\gamma_{k+1} \sim \Gamma(\cdot \mid \gamma_{\leq k}, x_{\leq k}),$$

so the next gate value may depend on the entire written prefix (and past gate values) but never on future symbols. While γ_k remains constant, the same side continues emitting symbols into the shared stream. Crucially, a *token boundary* occurs exactly when γ switches value; this partitions the substrate into maximal constant-γ blocks. Reading these blocks in order yields an alternating sequence of interface-level tokens

$$a_1, o_1, a_2, o_2, \ldots \qquad (a_t, o_t) \in \mathcal{A} \times \mathcal{O},$$

where a_t is the t-th agent-written block and o_t is the subsequent world-written block. This is how the sets $\mathcal{A}$ and $\mathcal{O}$ are (implicitly) defined.

Interface-level tokens. The gate Γ induces the segmentation into blocks; the sets $\mathcal{A}$ and $\mathcal{O}$ are therefore *interface-level* token sets rather than intrinsic properties of Σ^*. In particular, action/observation tokens need not be self-delimiting when viewed as raw substrings of Σ^* in isolation: the block boundaries are provided by the gate, not by an assumed (prefix-free) code for $\mathcal{A}$ or $\mathcal{O}$. In other words, to uniquely decode actions and observations from the symbol transcript, we need the gating transcript.

First-person accounting in block generation. The key discipline is that each side treats its *own* completed tokens as interventions and the other side's completed tokens as evidence. We write $\hat{a}_t$ for an agent-produced action token (an intervention from the agent's view) and $\hat{o}_t$ for a world-produced observation token (an intervention from the world's view), where $\hat{z} := \mathrm{do}(Z = z)$ is bookkeeping for "this token was chosen by the side recording it."[32]

Let k lie inside the current block, let τ be the substrate index of the most recent token boundary, and let $x_{\tau:k}$ denote the prefix of the current block emitted so far. Then the next symbol is drawn from the generator that currently holds the gate. Writing the completed token history up to the current interaction round as $\underline{ao}_{<t}$, we sample

$$x_{k+1} \sim \begin{cases} \mu\big(x_{k+1} \mid \underline{a\hat{o}}_{<t} a_t,\ x_{\tau:k}\big) & \text{if } \gamma_k = 0 \text{ (world writes)}, \\ \pi\big(x_{k+1} \mid \underline{\hat{a}o}_{<t},\ x_{\tau:k}\big) & \text{if } \gamma_k = 1 \text{ (agent writes)}. \end{cases} \tag{1}$$

The only subtlety is the hat bookkeeping: the writer's *completed* past tokens are treated as interventions by that writer, and as evidence by the other side.

The timing matters. While a token is being emitted, both sides condition on the partially written block prefix $x_{\tau:k}$. Only after the block terminates (a token boundary occurs) do we *retroactively* mark the

completed block as an intervention for its writer. Thus, from the agent's perspective, completed agent blocks become $\hat{a}_t$ and completed world blocks become o_t; from the world's perspective, completed world blocks become $\hat{o}_t$ and completed agent blocks become a_t.

Causal skeleton and chronological decomposition. It is convenient to describe causal dependencies at the *token* level. When π or μ is a mixture (as in the universal prior), we can regard a mixture component index as chosen first, after which tokens are generated in time order. This yields the chronological causal skeleton:

$$\forall t, \qquad p \to a_t, \qquad p \to o_t, \qquad \underline{ao}_{<t} \to a_t, \qquad \underline{ao}_{<t} a_t \to o_t. \tag{2}$$

Equivalently, under a fixed component p, interaction is generated by two one-step channels

$$a_t \sim \nu_p(\cdot \mid \underline{ao}_{<t}), \qquad o_t \sim \nu_p(\cdot \mid \underline{ao}_{<t}, a_t),$$

where the same ν_p induces both an action channel and an observation channel.

Primitive vs. mixture generators. The bookkeeping distinction between hats and unhatted tokens becomes substantive when a generator is a mixture.

- If π (or μ) is *primitive* (not a mixture), then hats are pure bookkeeping: they do not change the local chronological one-step rules. In particular we have the "action/observation exchange" identity from do-calculus [32]:

  $$\pi(\cdot \mid \underline{\hat{a}o}_{<t}) = \pi(\cdot \mid \underline{ao}_{<t}), \qquad \pi(\cdot \mid \underline{\hat{a}o}_{<t}, \hat{a}_t) = \pi(\cdot \mid \underline{ao}_{<t}, a_t), \tag{3}$$

  and analogously for μ when it is primitive. Intuitively: in a primitive generator, there is nothing "inside" for the intervention/evidence marks to act on.

- If π (or μ) is a *mixture*, e.g. $\pi = \sum_p w(p)\, \nu_p$, then hats do real work: interventions are computed by the usual "surgery" rule inside the mixture [32] (see next subsection). Action probabilities may guide generation, but they are excluded from evidence when updating mixture weights. This is exactly the mechanism that prevents "learning from one's own actions" in joint interaction models [28, 30].

3.1. Interventions in mixtures

Everything that follows is the explicit computation rule for pushing the intervention/evidence discipline through a mixture. Under a fixed component ν_p, we write down the on-path chronological factorization; to impose actions we replace action mechanisms by point masses; in a mixture we then update weights using only the remaining world-response factors; prediction and action are obtained by posterior averaging (or posterior sampling) of the corresponding component channels [28].

Formal intervention rule in a joint generator. Under a fixed component ν_p, the on-path joint likelihood of a length-t interaction history factorizes chronologically as

$$\nu_p(\underline{ao}_{\le t}) = \prod_{k=1}^{t} \nu_p(a_k \mid \underline{ao}_{<k})\, \nu_p(o_k \mid \underline{ao}_{<k} a_k). \tag{4}$$

When the agent imposes interventions $\hat{a}_1, \ldots, \hat{a}_t$, we replace each action mechanism by a point mass at the imposed value:

$$\nu_p(a_k \mid \underline{ao}_{<k}) \quad \longrightarrow \quad \delta(a_k = \hat{a}_k),$$

so the interventional likelihood retains only the empirically tested world-response factors:

$$\nu_p(\underline{\hat{a}o}_{\le t}) = \prod_{k=1}^{t} \nu_p(o_k \mid \underline{\hat{a}o}_{<k}\hat{a}_k). \tag{5}$$

(For primitive ν_p, we can drop the hats on the right hand side; the distinction matters only when we must update over p.)

Intervention posterior. Suppose the agent uses a mixture $\pi = \sum_p w(p)\nu_p$ (in particular, $\pi := M$ later). Given a realized first-person history $\underline{\hat{a}o}_{\le t}$, Bayes' rule with the interventional likelihood (5) yields the intervention posterior

$$w(p \mid \underline{\hat{a}o}_{\le t}) = \frac{w(p)\prod_{k=1}^{t}\nu_p(o_k \mid \underline{\hat{a}o}_{<k}\hat{a}_k)}{\sum_q w(q)\prod_{k=1}^{t}\nu_q(o_k \mid \underline{\hat{a}o}_{<k}\hat{a}_k)}. \tag{6}$$

Equivalently, the one-step recursion after observing o_t is

$$w(p \mid \underline{\hat{a}o}_{\le t}) = \frac{\nu_p(o_t \mid \underline{\hat{a}o}_{<t}\hat{a}_t)\, w(p \mid \underline{\hat{a}o}_{<t})}{\sum_q \nu_q(o_t \mid \underline{\hat{a}o}_{<t}\hat{a}_t)\, w(q \mid \underline{\hat{a}o}_{<t})}.$$

Since interventions are not evidence, appending an action alone leaves weights unchanged:

$$w(p \mid \underline{\hat{a}o}_{\le t}\hat{a}_{t+1}) = w(p \mid \underline{\hat{a}o}_{\le t}). \tag{7}$$

Prediction. Given an intervention prefix $\underline{\hat{a}o}_{<t}\hat{a}_t$, the predictive mixture for the next observation is the posterior average of hypotheses' interventional observation rules:

$$\pi(o_t \mid \underline{\hat{a}o}_{<t}\hat{a}_t) = \sum_p \nu_p(o_t \mid \underline{\hat{a}o}_{<t}\hat{a}_t)\, w(p \mid \underline{\hat{a}o}_{<t}). \tag{8}$$

This is the first-person correction: we condition on the imposed action $\hat{a}_t$ but do not score the action as evidence.

Action. The agent's action distribution is defined by mixing the hypotheses' action channels under the intervention posterior:

$$\pi(a_{t+1} \mid \underline{\hat{a}o}_{\le t}) = \sum_p \nu_p(a_{t+1} \mid \underline{\hat{a}o}_{\le t})\, w(p \mid \underline{\hat{a}o}_{\le t}). \tag{9}$$

This looks formally parallel to (8); the difference is semantic: (8) is used for belief updating and prediction, while (9) is used to *generate* the next intervention.

Operational view: random beliefs / posterior sampling. Sampling an action from the mixture (9) can be implemented by first sampling a single hypothesis from the intervention posterior and then acting according to its action channel:

$$a_{t+1} \sim \pi(\cdot \mid \underline{\hat{a}o}_{\le t}) \qquad \Longleftrightarrow \qquad \bar{p} \sim w(p \mid \underline{\hat{a}o}_{\le t}), \qquad a_{t+1} \sim \nu_{\bar{p}}(\cdot \mid \underline{\hat{a}o}_{\le t}).$$

The timing is as in (1): the agent emits substrate symbols until the action token completes; only then is the completed action retroactively recorded as $\hat{a}_{t+1}$ to mark it as "self-generated".

3.2. Counterfactual and third-party actions

When the agent writes the output at time t, we need a notion of the "intended continuation" to compare to. To do so, we introduce a counterfactual output variable $\dot{a}_t$ (and a simulation-based operational definition) to denote the output the world *would have written* at that substrate position. Conversely, if the agent watches the world writing a long output o_t, it can identify part of that output as containing an embedded action. This induces a decomposition of the realized observation token o_t into prefix-action-suffix subtokens.

Definition 2 (Potential action index)**.** Fix an interaction system $(\Sigma, \Gamma, \pi, \mu)$. A *potential action index* is a substrate position k such that, immediately before writing x_k, the already-written transcript (substrate plus gate history up to $k-1$) decodes uniquely as

$$\underline{\hat{a}o}_{<t}\hat{a}_t\, w,$$

where w is the non-empty world-written prefix since the end of $\hat{a}_t$ (i.e. a prefix of the currently written observation token). That is, if the gate were given to the agent next, it would write a_{t+1} and $o_t = w$ would conclude.

Definition 3 (Counterfactual action)**.** Let k be a potential action index with associated transcript $\underline{\hat{a}o}_{<t}\hat{a}_t\, w$, where w is the non-empty world-written prefix since the end of $\hat{a}_t$. Assume $\gamma_k = 1$, so the agent writes next.

To define the world's $\mathcal{A}$-continuation at k, run the following *separate* counterfactual simulation branch, initialized from the already-written on-path transcript up to $k-1$. We will use the gate Γ to generate a second sequence $\dot{\gamma}_k$, used solely for tokenization purposes. Let $(\dot{\gamma}_j, \dot{x}_j)_{j\geq k}$ be generated as follows:

- *Shared prefix.* Set $\dot{\gamma}_{\leq k-1} := \gamma_{\leq k-1}$ and $\dot{x}_{\leq k-1} := x_{\leq k-1}$.
- *Force an $\mathcal{A}$-block start.* Set $\dot{\gamma}_k := 1$.
- *Evolve branch chronologically.* For $j \geq k$, first sample the next substrate symbol by

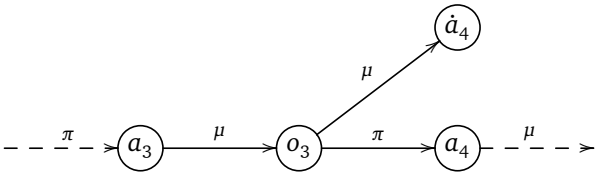
$$\dot{x}_j \sim \mu(\cdot \mid \underline{a\hat{o}}_{<t} a_t\, w\, \dot{x}_{k:j-1}),$$

  i.e. μ emits the content of the forced $\mathcal{A}$-block in the branch, conditioned on the shared past and on the already-emitted branch block prefix. Then sample the next gate value by

$$\dot{\gamma}_{j+1} \sim \Gamma(\cdot \mid \dot{\gamma}_{\leq j}, \dot{x}_{\leq j}).$$

Let $k' > k$ be the first position such that $\dot{\gamma}_{k'} = 0$; equivalently, $[k, k')$ is the maximal block in the branch with $\dot{\gamma} = 1$.

The *counterfactual action* $\dot{a}_{t+1} \in \mathcal{A}$ is defined to be the (random) block content written over positions $k, \ldots, k'-1$ in that branch, i.e. $\dot{a}_{t+1} := \dot{x}_{k:k'-1}$. This defines $\dot{a}_{t+1}$ precisely. Note that k' is determined inside the branch and therefore the length of $\dot{a}_{t+1}$ need not match the length of the on-path $\mathcal{A}$-token written by the agent starting at k. The diagram below illustrates the counterfactual action $\dot{a}_4$ within the generated transcript.



Definition 4 (Third-party action)**.** Let k be a potential action index with associated transcript $\underline{\hat{a}o}_{<t}\hat{a}_t\, w$ with w non-empty, and assume $\gamma_k = 0$, so the world *continues* writing the current observation token o_t.

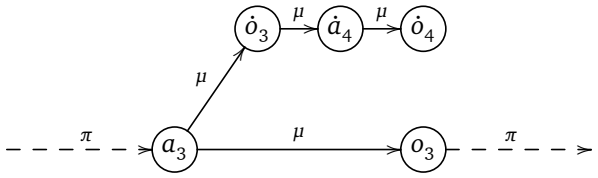
To define the world's $\mathcal{A}$-continuation at k, run the following *tokenization procedure*, initialized from the already-written on-path transcript up to $k-1$. Let $(\dot{\gamma}_j)_{j\geq k}$ be generated as follows:

- *Shared prefix.* Set $\dot{\gamma}_{\leq k-1} := \gamma_{\leq k-1}$.
- *Force an $\mathcal{A}$-block start.* Set $\dot{\gamma}_k := 1$.
- *Read transcript chronologically.* For $j \geq k$, let x_j be the next substrate symbol generated by the world on-path. Then sample the next gate value by

$$\dot{\gamma}_{j+1} \sim \Gamma(\cdot \mid \dot{\gamma}_{\leq j}, x_{\leq j}).$$

Let $k' > k$ be the first position such that $\dot{\gamma}_{k'} = 0$; equivalently, $[k, k')$ is the maximal block in the branch with $\dot{\gamma} = 1$.

The *third-party action* $\dot{a}_{t+1} \in \mathcal{A}$ is defined to be the (random) block content written in the transcript over positions $k, \ldots, k'-1$, that is, $\dot{a}_{t+1} := x_{k:k'-1}$. This defines $\dot{a}_{t+1}$ precisely. If $\dot{a}_{t+1}$ ends before its embedding O-token, we can write v for the remaining non-empty suffix of the same world-written token after position $k'-1$ and thus $o_t = w\, \dot{a}_{t+1}\, v$. The diagram below illustrates the third-party action $\dot{a}_4$ within the generated transcript:



It is not hard to see that, for a given potential index k, the counterfactual action $\dot{a}_{t+1}$ and the third-party action $\dot{a}_{t+1}$ are the same random block: the difference is only whether the gate sampled $\gamma_k = 1$ (counterfactual, not observed) or $\gamma_k = 0$ (third-party, observed) at position k. The precise distinction between the different types of $\mathcal{A}$-tokens is important; we will also refer to them as *factual* (i.e. first-person, agent-generated), *counterfactual*, and *third-party* $\mathcal{A}$-tokens.

3.3. Observations on the model

Why actions cannot be evidence. Hypotheses may assign probabilities to actions. If one naively updated on realized actions as if they were evidence, the posterior would spuriously reward hypotheses that happened to predict the agent's own sampled choices [28, 30]. Concretely, suppose two programs p, q have identical tested world-response rules,

$$\nu_p(o \mid h, a) = \nu_q(o \mid h, a) \quad \forall (h, a, o),$$

but different action channels $\nu_p(a \mid h) \neq \nu_q(a \mid h)$. Then the world responses provide no evidence to distinguish them. Under the intervention posterior, (7) ensures that appending an action alone leaves odds unchanged, so neither program is self-reinforced by the agent's own behavior. If one instead conditioned on $\nu_p(a_{t+1} \mid h)$ as evidence, the posterior would drift toward whichever program better predicted the sampled action, even though no new information about world responses was observed.

The gate as a boundary-uncertainty device. The gate Γ is an information-structure device: it formalizes that, from the agent's perspective, the action/observation boundary is not itself a learnable regularity of the interaction history. If the agent could reliably detect a pattern in when it must write versus when the world will write, then $\mathcal{A}$-blocks would split into distinguishable regimes, and evidence obtained when the world writes a would-be $\mathcal{A}$-continuation need not constrain behavior when the agent is later required to write one (transfer can fail by selection effects). We therefore treat slot assignment as a Harsanyi-style move by Nature in a Bayesian game [9]: at each would-be $\mathcal{A}$-block start, the gate outcome is chosen *chronologically* from the already-written transcript (including past gate values) and is not revealed to the agent at decision time; hence the agent's continuation rule is defined invariantly—without conditioning on whether the next $\mathcal{A}$-block will be revealed as evidence or demanded as an intervention.

Meaning of tokens (experiments and reported outcomes). The tokens a_t and o_t are interface-level abstractions, not literal motor and sensor variables. An "action" a_t may denote a temporally extended procedure (issuing a tool call, running a controller, choosing an experimental protocol), and an "observation" o_t may be a summary of the resulting outcomes (tool output, success/failure, diagnostics, feedback). From the agent's first-person perspective, the atomic unit of evidence is therefore the intervention–outcome pair $(\hat{a}_t, o_t)$: an experiment together with its reported result.

Where are policies, predictors, and preferences? Our setup is unusual because we use a single-model semantics: each hypothesis is a complete chronological generator of substrate strings and therefore (together with the gate) induces *simultaneously* an action and an observation channel. That is, we do not factor hypotheses into "policy" and "environment" components. The agent–world distinction is imposed externally by the interface and, crucially, by the epistemic rule used for distinguishing actions from evidence.

We also do not take rewards and utilities as primitive. Rewards can be included as one kind of observation token, alongside demonstrations, language, tool outputs, and feedback. They could potentially be ignored by the agent. As we will see later, we let the patterns contained in the single-stream substrate plus the intervention/evidence split induce the behavioral schemas directly: from the agent's point of view, interaction becomes a "gap-filling" or "pattern completion" problem.

Sampling from a semimeasure (symbol-level oracle assumption). As a simplifying modeling assumption, we posit access to a symbol-level sampling procedure for any lower-semicomputable generator ν on Σ^*: given the current written prefix, we can draw the next substrate symbol according to $\nu(\cdot \mid \text{prefix})$. Together with the gate, this induces token-level sampling: whichever side holds the gate samples symbols until the gate switches and the block closes. If ν stops early (the semimeasure's missing mass), sampling may halt mid-block.

Counterfactual vs. third-party actions. We have shown how certain substrate positions admit an *interpretation* as an $\mathcal{A}$-token: namely, the $\mathcal{A}$-block that would be produced starting at k under a fixed counterfactual convention in which the world is taken to write through the next $\mathcal{A}$-block (Definitions 3 and 4). This choice of counterfactual is not meant to be unique; it is adopted for mathematical convenience, because it yields a single canonical continuation object that is sometimes observable (as a third-party action) and sometimes not (as a counterfactual action).

4. Universal Imitation

We now treat purposeful interaction as *intrinsic continuation*: the agent extends a rule that is already present in its experience. The key trick is *equating predicting with acting*. In passive prediction, the agent observes a prefix and predicts the next symbol. In interaction, the transcript contains alternating *gaps* (action slots) and *evidence* (world outputs). If the agent chooses its next action by sampling from its current predictive distribution for "what should occupy the next action slot", then the agent is literally a *pattern completer*: it extends the transcript according to the regularities compressed by its current beliefs.

4.1. Intrinsic continuation and the identifiability problem

A tempting picture of interaction is *intrinsic continuation*: given what has been written so far, the agent computes a "natural next piece" by predicting what should come next in the transcript. In many imitation-style situations this is exactly the computation we want: the agent extends a regularity that is already present in its experience.

To see the intuition in a minimal form, consider a supervised-learning style prompt embedded in a single observation token:

$$o_{t-1} = (u_1, v_1), (u_2, v_2), \ldots, (u_n, \cdot).$$

The last pair is incomplete: the label v_n is missing. Now imagine running the agent's internal continuation simulation on this token. After each prompt u_i, the agent runs the same computation to form a prediction for the next label v_i.[1] Nothing about this changes at the last prompt: after seeing u_n, the same internal computation could simply predict (or sample) v_n—which makes the task sound straightforward.

But there is an *identifiability* pitfall: the example assumes the agent is in the *same epistemic situation* at every label. If, before each v_i, an "act/observe" signal indicates whether the label will be *shown by the world* or *demanded of the agent*, then the first $n-1$ labels are typically learned under "observe," while the final gap may be the first time "act" appears. The signal itself could be explicit (a codeword) or implicit (a coding pattern). Regularities under "observe" need not transfer to the "act" regime. Worse, under a first-person treatment past "act" labels are not evidence and thus can't inform future behavior. In short, the problem is that different hypotheses can agree on all observed outcomes yet prescribe different actions. Acting is therefore unidentifiable even when "observe" prediction is well supported.

To avoid the identifiability problem, the agent must commit to its continuation *before* any "act/observe" assignment is revealed/identified. This is what the stochastic gate enforces: the agent does not observe the assignment, so it uses the same intrinsic continuation rule whether the next slot is revealed or demanded. If the slot is assigned to the agent, it simply outputs the completion.

To evaluate an action, we compare it to a *counterfactual action* [7, 32]: the token the world would have written in the same position if it had kept writing. In the labeled-pairs example $(u_1, v_1), \ldots, (u_n, \cdot)$, we can segment the stream and thereby identify each label location as a *potential action position*. For $i = 1, \ldots, n-1$, the world fills that potential action position, so each v_i is a *third-party action*: a token in an $\mathcal{A}$-position, but written by the world and therefore available as evidence. The same potential action position after u_n defines the final missing label: the *counterfactual action* is the label the world would have written there, while the agent's emitted label is the *factual action*. This makes the relation between the three types of actions explicit: third-party actions are the previously revealed labels, and

[1] The same simulation also predicts future prompts, but that part is typically much harder; the salient object here is the next label.

their shared slot structure defines the counterfactual target for the final gap against which the factual action can be compared.

The same intuition extends beyond labeled examples. The conditioning context is the entire experience, which may mix extensional content (examples, demonstrations, traces) with intensional content (instructions, rules, constraints, verifier descriptions, program sketches). As long as the agent focuses on predicting the next continuation from whatever structure is present, no special handling is required: the gate determines whether that continuation is scored as evidence (world writes) or realized as an intervention (agent writes), while the agent's intrinsic continuation computation stays the same.

4.2. From third-party evidence to behavioral convergence

We now formalize the setting in which third-party actions provide evidence about the continuation rule, and in which this evidence *transfers* to the agent's own actions. We then show that an agent driven by the universal mixture M converts this evidence into behavior: acting by intrinsic completion induces a policy whose divergence from the counterfactual targets is bounded in cumulative expectation. We begin by listing definitions and assumptions.

We need to specify *which* substrate positions count as "action-slot starts." This definition ensures (i) each chosen position is a valid potential action index (so $\dot{a}^{(k)}$ is well-defined) and (ii) slots do not overlap and are separated by at least some world-written material, so that evidence from one slot is part of the agent-visible history at later slots.

Definition 5 (Action schedule)**.** Fix an interaction system $(\Sigma, \Gamma, \pi, \mu)$. An *action-slot schedule* is an infinite random sequence of substrate positions $k_1 < k_2 < \cdots$ such that each k_i is a potential action index (Definition 2), and the $\mathcal{A}$-token beginning at k_i ends strictly before k_{i+1} begins. Moreover, between the end of the $\mathcal{A}$-token beginning at k_i and the start of the $\mathcal{A}$-token beginning at k_{i+1}, the world writes at least one nonempty substrate block w_i, so the agent-visible transcript is $h_i := \hat{a}_1 o_1 \cdots \hat{a}_{t(i)} w_i$, where $t(i)$ is the interaction time of the last completed action. Let $\dot{a}^{(k_i)}$ denote $\mathcal{A}$-token the world would write at position k_i (i.e. the counterfactual/third-party action when $\gamma(k_i) = 1$ / $\gamma(k_i) = 0$ respectively).

To make "revealed" slots informative about "demanded" slots, we need the gate assignment at each slot to behave like a randomized masking device: it should decide whether the $\mathcal{A}$-token becomes evidence or an agent intervention using only the agent-visible past, and without peeking at what the world continuation will be. We also restrict the world to primitive measures (i.e. no mixtures, no semimeasures).

Assumption 1 (Standard setup)**.** Assume $(\Sigma, \Gamma, \pi, \mu)$ is an interaction system where $\pi := M$ is the *universal semimeasure* and μ is a *primitive measure*. Let $(k_i)_{i \geq 1}$ be an action-slot schedule (Definition 5). The following conditions hold:

- *Action-slot is chosen by coin flip.* At each k_i, the gate draws

$$\gamma(k_i) \sim \text{Bernoulli}(\rho_i), \qquad \rho_i \in (0, 1),$$

  where ρ_i is a chronological function of the agent-visible history h_i. Conditional on h_i, the bit $\gamma(k_i)$ is independent of the world's $\mathcal{A}$-token $\dot{a}^{(k_i)}$ at k_i.

- *Gate held fixed through action-slot.* The gate holds the value of $\gamma(k_i)$ fixed throughout the $\mathcal{A}$-token beginning at k_i. If $\gamma(k_i) = 0$, the world writes the $\mathcal{A}$-token (so it is a third-party action); if $\gamma(k_i) = 1$, the agent writes the $\mathcal{A}$-token (so it becomes an intervention $\hat{a}$ from the agent's view).

- *Infinitely many agent-written slots.* With probability 1, $\gamma(k_i) = 1$ occurs for infinitely many i.

Induced agent interventions and world targets. Before we proceed, we need to clarify the indexing of action slots, and in particular, their substrate position versus agent-time. According to Assumption 1, the schedule specifies substrate positions $k_1 < k_2 < \cdots$. Then, $\dot{a}^{(k_i)} \in \mathcal{A}$ denotes the $\mathcal{A}$-token the world would write starting at k_i. If $\gamma(k_i) = 0$ this token is realized on-path as an embedded third-party action; if $\gamma(k_i) = 1$ it is only a counterfactual target. To index only the factual actions (the slots assigned to the agent), let $i_1 < i_2 < \cdots$ be the (random) indices with $\gamma(k_{i_t}) = 1$. For each $t \geq 1$, define $a_{t+1} \in \mathcal{A}$ as the $\mathcal{A}$-token the agent actually writes at k_{i_t}, and define the corresponding counterfactual target by $\dot{a}_{t+1} := \dot{a}^{(k_{i_t})}$. In addition, notice that in this case, the previous observation token was completed, and hence $h_i = \hat{a}_1 o_1 \cdots \hat{a}_{t(i)} w_i = \underline{\hat{a}o}_{\leq t}$.

Deviation measures. To quantify how closely intrinsic completion tracks the target continuation, we use D_{KL} and TV. Since $M(\cdot \mid \cdot)$ may have missing mass, we complete it by adding a stop outcome $\perp \notin \mathcal{A}$ and writing $\overline{\mathcal{A}} := \mathcal{A} \cup \{\perp\}$:

$$\overline{M}(a \mid \cdot) := M(a \mid \cdot) \ (a \in \mathcal{A}), \qquad \overline{M}(\perp \mid \cdot) := 1 - \sum_{a \in \mathcal{A}} M(a \mid \cdot),$$

while for the measure μ we set $\overline{\mu}(a \mid \cdot) := \mu(a \mid \cdot)$ for $a \in \mathcal{A}$ and $\overline{\mu}(\perp \mid \cdot) := 0$. For distributions P, Q on a countable set, define the *Kullback-Leibler divergence* and the *total variation* as

$$D_{\mathrm{KL}}(P\|Q) := \sum_x P(x) \log \frac{P(x)}{Q(x)} \qquad \text{and} \qquad \mathrm{TV}(P, Q) := \frac{1}{2} \sum_x |P(x) - Q(x)|$$

respectively. In this section, $D_{\mathrm{KL}}(\mu\|M)$ and $\mathrm{TV}(\mu, M)$ are shorthand for $D_{\mathrm{KL}}(\overline{\mu}\|\overline{M})$ and $\mathrm{TV}(\overline{\mu}, \overline{M})$ on $\overline{\mathcal{A}}$ (with log natural).

4.2.1. The transfer lemma

The next lemma is the basic bookkeeping step that links third-party and counterfactual actions. Whenever a nonnegative quantity of interest is determined by the agent-visible history immediately before an action slot begins, then the expected total of that quantity over agent-assigned slots can be rewritten exactly as a reweighted expected total over world-assigned slots. The only ingredient is that the slot assignment is a coin flip based on the agent-visible past.

Lemma 1 (Transfer)**.** *Under Assumption 1, for any nonnegative sequence $(\Delta_i)_{i \geq 1}$ such that each Δ_i is determined by the agent-visible history immediately before k_i, we have the exact identity*

$$\mathbb{E}\left[\sum_{i:\,\gamma(k_i)=1} \Delta_i\right] = \mathbb{E}\left[\sum_{i:\,\gamma(k_i)=0} \frac{\rho_i}{1-\rho_i} \Delta_i\right]. \tag{10}$$

In particular, if there exists $r < \infty$ such that $\frac{\rho_i}{1-\rho_i} \leq r$ with probability 1 for all i, then

$$\mathbb{E}\left[\sum_{i:\,\gamma(k_i)=1} \Delta_i\right] \leq r\, \mathbb{E}\left[\sum_{i:\,\gamma(k_i)=0} \Delta_i\right]. \tag{11}$$

Proof. Fix i and condition on the agent-visible history h_i available immediately before k_i. Under this conditioning, Δ_i and ρ_i are fixed and $\gamma(k_i) \sim \text{Bernoulli}(\rho_i)$, hence

$$\mathbb{E}[\mathbf{1}\{\gamma(k_i) = 1\}\Delta_i \mid h_i] = \rho_i \Delta_i = \mathbb{E}\left[\mathbf{1}\{\gamma(k_i) = 0\}\frac{\rho_i}{1-\rho_i}\Delta_i \;\middle|\; h_i\right].$$

Taking expectation gives

$$\mathbb{E}[\mathbf{1}\{\gamma(k_i) = 1\}\Delta_i] = \mathbb{E}\left[\mathbf{1}\{\gamma(k_i) = 0\}\frac{\rho_i}{1-\rho_i}\Delta_i\right].$$

Summing over $i = 1, \ldots, N$ and letting $N \to \infty$ (all terms are nonnegative) yields (10). If $\frac{\rho_i}{1-\rho_i} \le r$ a.s. for all i, then (11) follows by bounding the right-hand side termwise by $r\,\mathbf{1}\{\gamma(k_i) = 0\}\Delta_i$ and taking expectation. □

4.2.2. Universal bound on third-party actions

We now instantiate the abstract Δ_i with a quantity that measures how much *evidence about the continuation* the agent would obtain if the world writes it. The key design constraint is twofold. First, Δ_i must be determined from the agent-visible transcript available immediately before the action slot begins (so the Transfer lemma applies). Second, when the continuation is actually revealed as a third-party action, Δ_i should be chargeable to a single global "evidence budget" implied by universality: along the world-written stream, the universal mixture M cannot fall behind the true world μ by more than a fixed constant in cumulative log-loss [20, 38]. Since third-party actions are literal substrings of the world-written stream, their total contribution can be bounded by the same constant.

For each action slot k_i, define the *action evidence divergence* (at the agent's decision time) by

$$\Delta_i := \mathbb{E}\left[\log \frac{\mu(\dot{a}^{(k_i)} \mid h_i)}{M(\dot{a}^{(k_i)} \mid h_i)} \;\middle|\; h_i\right]. \tag{12}$$

On slots where $\gamma(k_i) = 0$, $\dot{a}^{(k_i)}$ is a third-party action, hence it contributes genuine evidence to the intervention posterior; on slots where $\gamma(k_i) = 1$, $\dot{a}^{(k_i)}$ remains counterfactual and is not observed on-path.

The next lemma says that the universal mixture has only a finite expected "log-evidence budget" to pay when comparing the true world to the mixture along the actually observed world-written stream. Since third-party actions are literally part of that observed stream, their total contribution is also bounded.

Lemma 2 (Universal bound on third-party actions)**.** *Assume (Σ, Γ, M, μ) is as in Assumption 1, with μ a measure, and consider any action-slot setup satisfying Assumption 1. Let Δ_i be defined by* (12)*. Then there exists a constant $C_\mu < \infty$ (depending only on μ and the chosen universal prior weights) such that*

$$\mathbb{E}\left[\sum_{i:\,\gamma(k_i)=0} \Delta_i\right] \le C_\mu. \tag{13}$$

Proof. Because μ is primitive, hats can be removed from the chronological conditionals:

$$\mu(a^{(k_i)} \mid h_i) = \mu(a^{(k_i)} \mid \underline{\hat{a}o}_{<t(i)}\, \hat{a}_{t(i)}\, w_i) = \mu(a^{(k_i)} \mid \underline{ao}_{<t(i)}\, a_{t(i)}\, w_i).$$

Fix i. Conditional on the decision-time transcript h_i available immediately before k_i, the slot assignment $\gamma(k_i)$ is a coin flip and (by Assumption 1) is independent of the $\mathcal{A}$-continuation $\dot{a}^{(k_i)}$. Moreover, on the event $\{\gamma(k_i) = 0\}$ the world writes through the $\mathcal{A}$-token beginning at k_i, so $\dot{a}^{(k_i)}$ is realized on-path as the embedded third-party action $a^{(k_i)}$(Definition 4). Therefore,

$$\mathbb{E}[\mathbf{1}\{\gamma(k_i) = 0\}\Delta_i] = \mathbb{E}\left[\mathbf{1}\{\gamma(k_i) = 0\} \log \frac{\mu(a^{(k_i)} \mid h_i)}{M(a^{(k_i)} \mid h_i)}\right]. \tag{14}$$

Fix $T \geq 1$. Consider the first T completed world tokens $o_1, \ldots, o_T$. Split each o_t using exactly the third-party segmentation of Definition 4: on each event $\{\gamma(k_i) = 0\}$ occurring during the generation of some o_t, we decompose that token as $o_t = w_i\, a^{(k_i)}\, v_i$, where w_i is the already-written world prefix at k_i and $a^{(k_i)}$ is the extracted $\mathcal{A}$-block. If multiple events $\{\gamma(k_i) = 0\}$ occur within the same token o_t, apply this decomposition iteratively (in chronological order) to obtain a sequence of sub-blocks whose concatenation equals o_t. Also split at the separating nonempty world-written blocks guaranteed by Definition 5. The interventional chain rule then expresses

$$\mathbb{E}\left[\log \frac{\mu(o_t \mid \underline{ao}_{<t} a_t)}{M(o_t \mid \underline{\hat{a}o}_{<t} \hat{a}_t)}\right]$$

as a sum of conditional expected log-ratios for these sub-blocks, each nonnegative. The auxiliary gate process used in Definition 4 is only a tokenization device: it reads the already-generated on-path symbols and is used solely to define the sub-block boundaries. Keeping only the terms corresponding to the world-written $\mathcal{A}$-blocks and using that the conditional history at the start of each $a^{(k_i)}$ is precisely h_i, together with (14), yields

$$\mathbb{E}\left[\sum_{i:\gamma(k_i)=0,\ t(i)\leq T} \Delta_i\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \log \frac{\mu(o_t \mid \underline{ao}_{<t} a_t)}{M(o_t \mid \underline{\hat{a}o}_{<t} \hat{a}_t)}\right].$$

By dominance, choose p_μ with $\mu = \nu_{p_\mu}$ and $w(p_\mu) > 0$, so for every realized first-person history $\underline{\hat{a}o}_{\leq T}$ we have $M(\underline{\hat{a}o}_{\leq T}) \geq w(p_\mu)\mu(\underline{\hat{a}o}_{\leq T})$, hence

$$\sum_{t=1}^{T} \log \frac{\mu(o_t \mid \underline{ao}_{<t} a_t)}{M(o_t \mid \underline{\hat{a}o}_{<t} \hat{a}_t)} = \log \frac{\mu(\underline{\hat{a}o}_{\leq T})}{M(\underline{\hat{a}o}_{\leq T})} \leq -\log w(p_\mu) =: C_\mu.$$

Taking expectation gives, for all T,

$$\mathbb{E}\left[\sum_{i:\gamma(k_i)=0,\ t(i)\leq T} \Delta_i\right] \leq C_\mu.$$

Letting $T \to \infty$ and using monotone convergence (nonnegative terms) yields (13). □

4.2.3. Universal bound on actions

We now combine the two ingredients: (1) the transfer lemma, which rewrites expected sums over agent-assigned slots in terms of reweighted expected sums over world-assigned slots, and (2) the universal bound on the total evidence accumulated on world-assigned slots. This yields the main statement: the agent's action distribution on the demanded slots approaches the world target in the cumulative divergence sense.

Theorem 3 (Universal bound on actions)**.** *Assume an action-slot setup in an interaction system* (Σ, Γ, M, μ) *as in Assumption 1. Let* $(a_t, \dot{a}_t)_{t\geq 1}$ *be the induced factual actions and their counterfactual targets, and let the agent act by intrinsic completion under* M*, i.e. at each agent-written slot it draws*

$$a_{t+1} \sim M(a_{t+1} \mid \underline{\hat{a}o}_{\leq t}),$$

and retroactively records the realized action as an intervention $\hat{a}_{t+1}$*. Define the gapwise action-target divergence*

$$D_t := D_{\mathrm{KL}}\big(\mu(\dot{a}_{t+1} \mid \underline{ao}_{\leq t}) \,\big\|\, M(a_{t+1} \mid \underline{\hat{a}o}_{\leq t})\big).$$

Then the cumulative action-target divergence on agent-written slots satisfies the exact identity

$$\sum_{t\geq 1} \mathbb{E}[D_t] = \mathbb{E}\left[\sum_{i:\gamma(k_i)=1} \Delta_i\right] = \mathbb{E}\left[\sum_{i:\gamma(k_i)=0} \frac{\rho_i}{1-\rho_i}\,\Delta_i\right], \tag{15}$$

where Δ_i *is defined in* (12)*. In particular, if there exists* $r < \infty$ *such that* $\frac{\rho_i}{1-\rho_i} \leq r$ *with probability* 1 *for all* i*, then*

$$\sum_{t\geq 1} \mathbb{E}[D_t] \leq r\, C_\mu, \tag{16}$$

where C_μ *is the constant from Lemma 2.*

Proof. By construction, indices i with $\gamma(k_i) = 1$ are in bijection with agent-written actions a_{t+1} via the map $t \mapsto i_t$. At agent-written action a_{t+1} we have $D_t = \Delta_{i_t}$. Hence, for each T,

$$\sum_{t=1}^{T} \mathbb{E}[D_t] = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{i_t}\right] = \mathbb{E}\left[\sum_{i:\gamma(k_i)=1,\ i\leq i_T} \Delta_i\right],$$

and letting $T \to \infty$ (nonnegative monotone increase) gives

$$\sum_{t\geq 1} \mathbb{E}[D_t] = \mathbb{E}\left[\sum_{i:\gamma(k_i)=1} \Delta_i\right].$$

Lemma 1 gives (15). If $\frac{\rho_i}{1-\rho_i} \leq r$ a.s., then

$$\sum_{t\geq 1} \mathbb{E}[D_t] = \mathbb{E}\left[\sum_{i:\gamma(k_i)=0} \frac{\rho_i}{1-\rho_i}\Delta_i\right] \leq r\,\mathbb{E}\left[\sum_{i:\gamma(k_i)=0} \Delta_i\right] \leq r\, C_\mu,$$

using Lemma 2. This is (16). □

4.2.4. Finite mistakes

The theorem above is a cumulative divergence statement. To read it as a "mistake" guarantee, we convert finite cumulative divergence into the claim that large deviations from the target continuation can occur only finitely often—either at the distribution level, or (in the deterministic case) at the level of literal action mismatches.

Corollary 4 (Finite mistakes)**.** *Suppose*

$$\sum_{t\geq 1} \mathbb{E}\big[D_{\mathrm{KL}}\big(\mu(\dot{a}_{t+1} \mid \underline{ao}_{\leq t}) \,\big\|\, M(a_{t+1} \mid \underline{\hat{a}o}_{\leq t})\big)\big] < \infty. \tag{17}$$

Then, for every $\varepsilon > 0$, the number N_ε of times $\mathrm{TV}\big(\mu(\dot{a}_{t+1} \mid \underline{ao}_{\le t}),\, M(a_{t+1} \mid \underline{\hat{a}o}_{\le t})\big) > \varepsilon$, satisfies

$$\mathbb{E}[N_\varepsilon] \le \frac{1}{2\varepsilon^2} \sum_{t \ge 1} \mathbb{E}\Big[D_{\mathrm{KL}}\big(\mu(\dot{a}_{t+1} \mid \underline{ao}_{\le t}) \,\big\|\, M(a_{t+1} \mid \underline{\hat{a}o}_{\le t})\big)\Big],$$

and $N_\varepsilon < \infty$ with probability 1. If moreover for each t the target $\mu(\dot{a}_t \mid \underline{ao}_{<t})$ is a point mass at some $\dot{a}_t \in \mathcal{A}$, then the total number of mismatches

$$N \;:=\; \sum_{t>1} \mathbf{1}\{a_t \neq \dot{a}_t\}$$

satisfies

$$\mathbb{E}[N] \le \sum_{t \ge 1} \mathbb{E}\Big[D_{\mathrm{KL}}\big(\mu(\dot{a}_{t+1} \mid \underline{ao}_{\le t}) \,\big\|\, M(a_{t+1} \mid \underline{\hat{a}o}_{\le t})\big)\Big],$$

and hence $N < \infty$ with probability 1.

Proof. Write $T_t := \mu(\dot{a}_{t+1} \mid \underline{ao}_{\le t})$, $P_t := M(a_{t+1} \mid \underline{\hat{a}o}_{\le t})$, and $D_t := D_{\mathrm{KL}}(T_t \| P_t)$. Pinsker gives $\mathrm{TV}(T_t, P_t)^2 \le \frac{1}{2} D_t$, hence

$$\Pr(\mathrm{TV}(T_t, P_t) > \varepsilon) \le \frac{\mathbb{E}[D_t]}{2\varepsilon^2}.$$

Summing over t yields the bound on $\mathbb{E}[N_\varepsilon]$ and implies $N_\varepsilon < \infty$ a.s. For the deterministic clause, let $\dot{a}_{t+1}$ denote the point-mass target and set $p_t := P_t(\dot{a}_{t+1}) = M(\dot{a}_{t+1} \mid \underline{\hat{a}o}_{\le t})$. Then $D_t = -\log p_t$ and

$$\Pr(a_{t+1} \neq \dot{a}_{t+1} \mid \underline{\hat{a}o}_{\le t}) = 1 - p_t \;\le\; -\log p_t.$$

Taking expectations and summing over t gives the bound on $\mathbb{E}[N]$, and finiteness of $\sum_t \Pr(a_t \neq \dot{a}_t) = \mathbb{E}[N]$ implies $N < \infty$ a.s. □

4.3. Universal imitation of computable stochastic functions

The previous results are stated for an abstract randomized interaction protocol. We now show this is not a vacuous idealization by constructing a computable interaction system in which the agent learns to implement *any* computable (possibly stochastic) function. Intuitively, this is the supervised example-learning setup from Section 4.1: after each prompt u_i, there is a designated next output position in $\mathcal{A}$. For instance, the world may present labeled examples followed by a new query,

$$o_{t-1} = (u_1, v_1), (u_2, v_2), \ldots, (u_n, \cdot),$$

so that the designated next output is the missing label and the agent's response is $a_t = v_n \in \mathcal{A}$. Crucially, this happens *after each prompt*: with a fixed nonzero probability, the protocol routes the next output position to the agent, so every prompt has a positive chance of requiring an on-path agent completion. To make generalization to unseen examples explicit, we choose the prompt schedule by dovetailing over $\mathcal{U}$ (i.e. a computable schedule in which every $u \in \mathcal{U}$ appears infinitely often), so any prompt not yet seen will still occur later and be tested/reused infinitely many times.

Corollary 5 (Universal imitation of computable stochastic functions)**.** *Let $f(\cdot \mid u)$ be any computable measure on $\mathcal{A}$ indexed by prompts $u \in \mathcal{U}$. Then there exists a computable interaction system (Σ, Γ, M, μ) with μ a primitive computable measure, such that the following holds on every slot assigned to the agent.*

At each decision time $t \ge 1$, the most recent world token o_t contains a computably decodable prompt $u_t \in \mathcal{U}$, and the counterfactual action $\dot{a}_{t+1} \in \mathcal{A}$ satisfies

$$\mu(\dot{a}_{t+1} = a \mid \underline{ao}_{\le t}) = f(a \mid u_t), \qquad a \in \mathcal{A}.$$

where each prompt $u \in \mathcal{U}$ appears an infinite number of times. There exists a constant $C < \infty$ such that

$$\sum_{t\geq 1} \mathbb{E}\Big[D_{\mathrm{KL}}\big(\mu(\dot{a}_{t+1} \mid \underline{ao}_{\leq t}) \,\Big\|\, M(a_{t+1} \mid \underline{\hat{a}o}_{\leq t})\big)\Big] \;\leq\; C.$$

Consequently, by Corollary 4, only finitely many large deviations from the counterfactual can occur (and if $f(\cdot \mid u)$ is deterministic, only finitely many literal mismatches occur).

Proof. Construction. Organize the world-written stream into an infinite sequence of prompt–gap pairs indexed by $i \geq 1$. Before the i-th gap, the world writes a nonempty block from which u_i is uniquely decodable at decision time. Choose the u_i according to a dovetailing strategy, so each prompt appears an infinite number of times. Let k_i be a potential action index at the start of that gap, and ensure gaps do not overlap and are separated by at least one further nonempty world-written block.

At each k_i, draw the gate coin $\gamma(k_i) \sim \mathrm{Bernoulli}(\rho)$ with a fixed bias $\rho \in (0, 1)$, and hold $\gamma(k_i)$ fixed throughout the ensuing $\mathcal{A}$-block. Define μ to be a primitive computable measure that, at the start of the i-th gap, generates the world continuation target according to $f(\cdot \mid u_i)$ (revealed on-path when $\gamma(k_i) = 0$, and otherwise remaining counterfactual as $\dot{a}$ for Definition 3). By construction, conditional on the agent-visible history immediately before k_i, the coin $\gamma(k_i)$ is independent of the ensuing world-generated $\mathcal{A}$-token, and the gate is held fixed through the slot. Then $\gamma(k_i) = 1$ occurs infinitely often with probability 1.

Apply the universal bound. The setup satisfies Assumption 1 with $\rho_i \equiv \rho$ and μ primitive, so Theorem 3 gives

$$\sum_{t\geq 1} \mathbb{E}\Big[D_{\mathrm{KL}}\big(\mu(\dot{a}_{t+1} \mid \underline{ao}_{\leq t}) \,\Big\|\, M(\cdot \mid \underline{\hat{a}o}_{\leq t})\big)\Big] \;\leq\; \frac{\rho}{1-\rho} C_\mu.$$

Taking $C := \frac{\rho}{1-\rho} C_\mu$ yields the claim, and the identity $\mu(\dot{a}_{t+1} = a \mid \underline{ao}_{\leq t}) = f(a \mid u_t)$ holds by construction on each agent-assigned slot. □

4.4. Preference pluralism

A useful way to read the preceding results is as *schema acquisition with context identification*. The interaction stream need not encode a single uniform behavioral regularity: it can contain many qualitatively different continuation principles—task procedures, tool protocols, dialogue norms, and choice rules—each applicable only in certain situations. Formally, we can view this as a computable partition of the prompt space $\mathcal{U} = \bigsqcup_{j=1}^{J} \mathcal{U}^{(j)}$ together with a family of computable continuation rules $f^{(j)}(\cdot \mid u)$ on $\mathcal{A}$, where the operative rule depends on which cell contains the current prompt. The substantive learning problem is therefore not merely to fit a single rule, but to infer *which* rule is in force by extracting the right situational cues from the transcript, i.e. to learn the partitioning of contexts and the corresponding schema within each cell.

Corollary 6 (Preference pluralism)**.** *Let $\mathcal{U}$ be a countably infinite prompt set with a computable partition $\mathcal{U} = \bigsqcup_{j=1}^{J} \mathcal{U}^{(j)}$, where each $\mathcal{U}^{(j)}$ is infinite. For each $j \in \{1, \ldots, J\}$ let $f^{(j)}(\cdot \mid u)$ be a computable measure on $\mathcal{A}$ indexed by $u \in \mathcal{U}^{(j)}$. Define the combined rule*

$$f(a \mid u) \;:=\; f^{(j)}(a \mid u) \qquad \text{for the unique } j \text{ with } u \in \mathcal{U}^{(j)}.$$

Then there exists a computable interaction system (Σ, Γ, M, μ) with μ a primitive computable measure, and a computable prompt schedule $(u_t)_{t\geq 1}$, such that on every slot assigned to the agent the most recent world token computably decodes a prompt $u_t \in \mathcal{U}$ and the counterfactual target satisfies

$$\mu(\dot{a}_{t+1} = a \mid \underline{ao}_{\leq t}) \;=\; f(a \mid u_t) \qquad \forall a \in \mathcal{A}.$$

Moreover, each $u \in \mathcal{U}$ occurs infinitely often in (u_t), and new prompts appear arbitrarily late, i.e. for every N there exists $t > N$ such that $u_t \notin \{u_1, \dots, u_{t-1}\}$. There exists $C < \infty$ such that

$$\sum_{t \geq 1} \mathbb{E}\Big[D_{\mathrm{KL}}\Big(\mu(\dot{a}_{t+1} \mid \underline{ao}_{\leq t}) \,\Big\|\, M(\cdot \mid \underline{\hat{a}o}_{\leq t}) \Big) \Big] \;\leq\; C.$$

Consequently, the finite-deviation (and deterministic mismatch) conclusions of Corollary 4 apply to these agent-assigned slots.

Proof. By computability of the partition and of each $f^{(j)}$, the combined rule $f(\cdot \mid u)$ is a computable measure on $\mathcal{A}$ indexed by $u \in \mathcal{U}$.

Because $\mathcal{U}$ is countably infinite and each cell $\mathcal{U}^{(j)}$ is infinite, we can fix a computable enumeration $\phi : \{1, \dots, J\} \times \mathbb{N} \to \mathcal{U}$ with $\phi(j, n) \in \mathcal{U}^{(j)}$ and $\{\phi(j, n) : n \in \mathbb{N}\} = \mathcal{U}^{(j)}$ for each j. Choose a computable prompt schedule (u_t) by dovetailing over pairs (j, n) through ϕ in a way that revisits each pair infinitely often. Then each $u \in \mathcal{U}$ occurs infinitely often, and since $\mathcal{U}$ is infinite, first occurrences are unbounded, so new prompts appear arbitrarily late.

Apply Corollary 5 to f using this schedule to obtain a computable interaction system (Σ, Γ, M, μ) with μ primitive and the stated target identity on agent-assigned slots, together with the finite cumulative divergence bound. The final sentence follows by invoking Corollary 4. □

The point is not merely eventual accuracy on previously seen prompts. Because the prompt schedule dovetails over $\mathcal{U}$, genuinely new prompts appear at arbitrarily late times while every prompt is revisited infinitely often. Together with the finite-deviation (and, for deterministic targets, finite-mismatch) guarantee, this implies that only finitely many demanded slots can be substantially wrong along the realized schedule—hence after a finite transient the agent tracks the target counterfactual even when a prompt has just appeared for the first time. This cannot be explained by finite memorization of prompt–completion pairs; it requires learning reusable *situational structure* on $\mathcal{U}$ (a partition into context classes) together with the corresponding per-class continuation rules. In this sense, "learning preferences" is one instance of a more general capability: acquiring multiple heterogeneous schemas and applying each one in the situations where it governs the next $\mathcal{A}$-token.

Notice that we can combine a variety of schema rules that instantiate well-known decision principles and other preference structures, such as:

- *Bayes-optimal finite-horizon POMDP control:* u encodes an executable finite POMDP, horizon/discount, and reward parameters; f outputs a Bayes-optimal adaptive controller.
- *Safety-first / constrained control:* u specifies dynamics plus a hard safety predicate (or budget constraint) and a secondary objective; f outputs a controller that enforces the constraint when feasible and otherwise follows the specified fallback rule.
- *Multi-objective tradeoffs:* u provides multiple reward components and weights (or a specified scalarization); f outputs the optimal controller under that tradeoff.
- *Choice from comparisons:* u contains a finite set of candidates plus computable pairwise comparisons or rankings; f outputs the candidate (or action program) selected by a computable revealed-preference rule.
- *Rule-/constitution-following:* u encodes a finite set of rules (hard constraints) plus a computable tie-breaker; f outputs an action/program satisfying the rules when feasible, and otherwise follows an explicitly encoded fallback.

- *Program synthesis / tool protocol:* u encodes a specification together with a computable evaluator or tool interface; f outputs a program (or macro-action) that passes the evaluator, or an explicit next repair step in an iterative protocol.
- *Norms / dialogue acts:* u encodes an interaction context together with a computable norm taxonomy; f outputs an appropriate dialogue act (e.g. apologize, clarify, refuse, defer) consistent with the norms and the stated context.

On prompts of the corresponding type, the agent will behave "as if" following the schema.

5. Discussion

Adaptive compression under interventions. A useful way to read the construction is as *adaptive compression* of the *realized* interaction history. In the passive case, a Bayesian mixture is an optimal adaptive code in expected log-loss [5, 8, 35]. In interaction, the same coding interpretation is recovered only after the first-person correction: evidence is the sequence of world responses under the agent's interventions, so mixture weights update by the interventional likelihood and the atomic unit of evidence is the completed pair $(\hat{a}_t, o_t)$ [6, 29].

The crucial additional point is that this compression view already fixes how to *act* at a potential action index. Along an action-slot schedule (k_i), the decision-time history h_i determines a distribution for the next $\mathcal{A}$-token. The gate bit $\gamma(k_i)$ decides whether the next $\mathcal{A}$-token is realized as a third-party action (when $\gamma(k_i) = 0$) or must be produced as a factual action (when $\gamma(k_i) = 1$), but the agent's decision-time information is h_i in either case. Hence the same mixture conditional used to predict (and therefore code) the next $\mathcal{A}$-token from h_i must also be used to generate it when the agent writes: the action rule draws the next factual action from that conditional. Operationally, this is exactly sampling from the mixture's conditional at decision time (equivalently, posterior sampling as an implementation detail) [19].

Demonstrations are observations; why actions are not evidence. Demonstration data enters the agent only as part of the observation tokens o_t: in particular, what were "their actions" for a demonstrator appear for the agent as substrings of some o_t and therefore count as evidence only insofar as they are predictable as world output. By contrast, the agent's own completed outputs $\hat{a}_t$ are interventions and therefore cannot be used as evidence to update mixture weights. This distinction is essential for joint interaction generators that assign probabilities to both action and observation tokens: if the agent were to update on realized a_{t+1} using the factor $\nu_p(a_{t+1} \mid \underline{\hat{a}o}_{\leq t})$ as if it were evidence, then hypotheses would be spuriously reinforced merely for assigning higher probability to the action that the agent itself sampled, even when all hypotheses agree on the tested world-response terms $\nu_p(o_t \mid \underline{\hat{a}o}_{<t}\hat{a}_t)$. The intervention posterior corrects this: weights are updated only through observation likelihoods under imposed interventions, i.e. via the factors $\nu_p(o_t \mid \underline{\hat{a}o}_{<t}\hat{a}_t)$, while appending an intervention alone leaves weights unchanged as in (7). This is exactly what prevents "learning from one's own actions" in mixtures: the agent conditions on $\hat{a}_t$ for prediction, but treats $\hat{a}_t$ as chosen input rather than as evidence about which generator is true.

Main guarantee and how it is proved. The main statement is a guarantee about the agent's *actions*. The action-slot schedule fixes positions $k_1 < k_2 < \cdots$, and at each k_i the gate bit $\gamma(k_i)$ decides whether the $\mathcal{A}$-token there is realized as a third-party action (world writes) or as a factual action (agent writes). The comparison target is always the counterfactual action at the same k_i, i.e. what the world would have written there under the counterfactual convention. Theorem 3 bounds the agent's

cumulative expected divergence from these counterfactual targets over the sequence of factual actions, and Corollary 4 turns this into a finite-deviation (and, for deterministic targets, finite-mismatch) conclusion about the agent's actions. The proof uses two ingredients: Lemma 2 shows that the total expected evidence contributed by third-party actions is bounded by the universal log-loss budget along the realized world output, and Lemma 1 transfers this bound to the factual-action indices.

When transfer fails. Transfer can fail if the gate violates the masking condition at the scheduled positions k_i. Two representative failure modes are: (i) *selection bias*, where the gate bit $\gamma(k_i)$ is correlated with the (unseen at decision time) continuation $\dot{a}^{(k_i)}$ even after conditioning on the decision-time history; and (ii) *regime detectability*, where the decision-time history contains cues that let the agent distinguish whether $\gamma(k_i) = 0$ or $\gamma(k_i) = 1$ before producing a factual action. In either case, third-party actions no longer provide representative evidence about the counterfactual actions that define the targets on agent-assigned positions, so good predictive fit on the realized transcript need not imply good behavior when the agent must act.

The gate as an interface, not a new object. To an RL reader, the gate Γ may look like an extra component alongside agent and world. It is better read as an abstraction of the I/O protocol (an interface or scheduler): turn-taking in dialogue, event-triggered control/interrupts, or "act-until-termination" macro-actions followed by an outcome report. Standard RL hard-codes a fixed action–observation rhythm; the gate generalizes this to variable-length, event-driven interaction where the next action slot need not be predictable from the agent-visible transcript [9, 24]. The coin-flip formalization is one convenient way to express this epistemic situation at the interface level.

Purpose as stabilized imitation. Under the intervention posterior, the mixture compresses the realized interventional stream by shifting mass toward those joint generators that continue to predict the world's responses under the interventions actually taken. Since actions are generated by posterior mixing (or posterior sampling) of those same generators' induced action channels, this reweighting is simultaneously a reorganization of behavior: action tendencies stabilize as posterior mass concentrates on a survivor class (not necessarily a single hypothesis). In particular, prompt-indexed continuation schemas (including the preference constructions of Section 4) are absorbed as reusable modes of action under the same posterior dynamics, with deviations on agent-assigned occurrences controlled by the universal imitation bounds (Theorem 3, Corollary 6).

Experiment design under interventions. Once actions are treated as interventions, what the agent can learn is controlled by which factual actions $\hat{a}_t$ it actually executes [32]. Along the realized transcript $\underline{\hat{a}o}_{\leq t}$, distinct joint generators can remain compatible with all observed outcomes o_t while disagreeing elsewhere; in that case the realized evidence does not determine their induced action channels at future decision times. The only way to break such equivalence classes is to choose factual actions under which the plausible generators predict measurably different outcomes, so that the resulting likelihood ratios shift the intervention posterior. This is the common core behind "exploration," "active querying," and "experimental design": select interventions that force disagreements in the world-response terms $\nu_p(o_t \mid \underline{\hat{a}o}_{<t}\hat{a}_t)$, rather than repeatedly acting in regimes where the remaining hypotheses make near-identical predictions [16, 42]. In this sense, the interface does not merely record behavior; it controls which intervention–outcome pairs $(\hat{a}_t, o_t)$ are ever realized, and therefore which regularities can become identifiable from interaction.

Interface-defined horizons. In standard reinforcement learning, "purpose" is fixed by an analysis horizon: one commits to a finite-horizon, discounted, or average-reward objective and then derives individual actions by solving (explicitly or implicitly) a Bellman-style recursion [1, 33, 41]. The horizon or discount is therefore not discovered from interaction; it is a parameter of the agent that determines what counts as long-run consequence and what gets treated as negligible. In our setting, there is no separate horizon knob: the atomic unit of evidence is the completed intervention–outcome pair $(\hat{a}_t, o_t)$, so the interface itself determines what one "step" means. An action token a_t may denote a short probe, a tool protocol, or a temporally extended procedure whose completion is followed by an outcome report o_t; changing what is bundled into a_t changes which consequences are immediately scored by the intervention posterior and therefore which hypotheses can be separated on-path. In this sense, the effective commitment length is neither fixed nor universal: it is induced by the gate-defined tokenization, and it can vary over time as different interventions expose different discriminative outcomes. This reframes the classical question "which discount is right?" into an interface question: "which experiments do we execute as single actions, and which results do we demand as outcomes?".

Language, tool use, and program acquisition. Language and tools fit the same interaction template without adding any special machinery: an utterance or tool invocation is just an action token a_t, and the reply, tool output, or feedback is an observation token o_t [25, 37, 44]. In that view, "language" is not a separate module but whatever compressive structure helps a short program predict how the stream of outcomes will change when the agent emits certain strings or runs certain procedures. Because the hypothesis space already ranges over complete interaction generators, it includes programs that implement parsing, protocol-following, and tool-use routines; repeated outcomes then shift posterior mass toward generators that reuse the fragments that compressed earlier I/O best. This is the practical content of Solomonoff's training-sequence picture in first-person form: earlier successful fragments become the easiest-to-sample continuations later, so the agent increasingly reuses verbal rules, templates, and tool protocols that have repeatedly predicted the world's replies to its interventions.

Reasons, schemas, and prompting. "Reasons" and "schemas" can be treated as reusable subprogram structure: latent components that reduce description length while preserving high interventional likelihood across contexts. Prompting can be read as providing side information that changes which programs are effectively short in the agent's reference language, thereby shifting posterior mass rapidly toward a compatible subset [2]; subsequent tool outputs and feedback are then the evidential stream that continues to refine the intervention posterior. On this view, *decision principles are subsumed*: a utility function and its associated choice rule can be part of a hypothesis in exactly the same way as any other reusable subroutine, and will be favored when it yields a shorter, better-predicting explanation of the observed intervention–outcome sequence (e.g., when observations include evaluative feedback). Because evidence is on-path, distinct internal decompositions can remain underdetermined unless interaction supplies outcomes that separate them; this is the same identifiability constraint expressed at the level of internal modular structure.

What this does not claim. The results are not reward-optimality statements: the paper does not define purposeful behavior by maximizing a specified objective, and the guarantees do not assert optimal long-run return under any reward interpretation. They are also not safety guarantees: they control divergence from counterfactual continuation targets under a particular interface protocol, not the downstream consequences of executing those continuations in arbitrary worlds. Finally, the transfer-based imitation guarantee is conditional: it relies on the masking-style gate assignment at

action slots; if the interface selects which continuations are revealed using information unavailable to the agent at decision time, the evidential anchoring that drives the bound need not apply.

Idealizations used in the model. The presentation makes explicit idealizations to keep the interface accounting clean. In particular, we assume access to symbol-level sampling from lower-semicomputable generators, and we allow semimeasure stopping (missing mass), which can halt generation mid-block. These assumptions isolate the epistemic point: the update rule must treat completed actions as interventions and completed world outputs as evidence, regardless of how the underlying sampling is approximated.

Limits of computable universality. Even with the intervention posterior enforcing first-person discipline, the construction remains an idealization: M is compact to *describe* (a small universal interpreter can run any program), but the inverse problem—inferring which short program explains the realized interaction outcomes and using it for control—is not captured by any fixed computable procedure. In particular, for any computable predictor/control rule there exists a computable interaction that defeats it by construction, so no single computable agent can dominate the full computable class in the same sense as the universal mixture [17, 18]. What is achievable computably is always *bounded universality*: if we restrict attention to generators of description length at most n, then there are procedures that succeed uniformly on that restricted class, but any procedure with such a guarantee must itself have description length $\Omega(n)$ [17]. Thus scaling competence forces expanding the represented program class (or search resources), which in turn forces the agent itself to grow; AI-as-induction is structurally open-ended. Finally, beyond some scale, broad claims of the form "this agent succeeds on all generators up to size n" can be true yet unprovable within a fixed consistent formal system, so even *certifying* wide competence can hit an epistemic wall [17].

6. Related work

Algorithmic probability and universal induction. Solomonoff induction constructs a universal predictor by mixing over all computable generators with description-length weights, yielding a universal semimeasure on strings [20, 22, 38, 39]. We adopt the standard mixture-over-enumerable-semimeasures viewpoint, but place it directly on *action–observation transcripts* rather than passive strings. This shift in the base object is what later lets us state *gapwise* imitation guarantees on agent-written slots, not only prediction guarantees on world-written substrings.

Compression, MDL, and prequential viewpoints. Bayesian mixtures admit a classical interpretation as optimal sequential codes in expected log-loss [5, 6, 8, 35]. Our contribution is to show how this coding/forecasting interpretation lifts to interaction only after the first-person correction: the evidential stream is the sequence of world outcomes under interventions, i.e. completed pairs $(\hat{a}_t, o_t)$, and mixture weights update via the interventional likelihood [29].

AIXI and an alternative foundation for universal AI. AIXI-style universal RL takes Solomonoff induction as the epistemic engine but defines agency by an additional semantic primitive: maximize a specified utility (reward) functional [10]. This makes "purpose" exogenous: the utility functional fixes what the agent is for, and the rest of cognition is whatever machinery best serves that scalar objective. We propose a different characterization: agency is universal inference in interaction, with compression applied end-to-end to complete action–observation transcripts. Hypotheses are joint generators that

compress regularities spanning both actions and outcomes; the agent–world asymmetry enters only through the first-person learning rule (factual actions are interventions, outcomes are evidence), yielding an intervention posterior over generators. There is no additional decision-theoretic layer: behavior is obtained directly by posterior mixing (equivalently, posterior sampling) of the hypotheses' induced action channels, so acting is the generative counterpart of predicting what belongs next in the transcript under the same compressed model. On this view, utility maximization is not the semantic core of AI but one optional schema that can be acquired when the realized transcript contains the corresponding regularities (e.g. demonstrations, language, tool protocols, feedback), bringing the universal limit picture closer to modern continuation-trained systems.

Bayesian adaptive control under interventions and posterior sampling. The closest methodological ancestor is the Bayesian control rule, which is already formulated as a Bayesian mixture over joint action–observation generators together with the first-person rule [28, 29]. What those formulations do *not* include is an interface ingredient like our gate: without a randomized gate assignment that sometimes produces a third-party action and sometimes requires a factual action, the induced action channel is generally not identifiable. As a result, earlier work typically couples hypotheses so that learning about outcomes also fixes behavior—for example, by pairing an environment model with its optimal policy under a specified objective. With such coupling in place, sampling from the posterior predictive admits the familiar two-step implementation (sample a hypothesis from the intervention posterior, then sample an action from its induced action channel), which is the operational core shared with Thompson sampling [19]. Our contribution is to combine a Solomonoff-style universal mixture over computable joint generators with a gate-defined interface that enables transfer from third-party actions to factual actions, yielding an imitation-style guarantee without hard-wiring a particular objective into the hypothesis coupling.

Decision theory and causal semantics for "actions are not evidence". The first-person update rule is also the Bayesian analogue of the causal-decision-theoretic stance that one's own choice should be treated as an intervention when forming beliefs about consequences [14, 21]. The contrasting evidential perspective—treating the probability of one's action as informative about the world—is a useful foil for explaining why joint generators require explicit surgery on action mechanisms rather than naive conditioning [13]. Our hat bookkeeping mirrors the do-operator intuition that $P(Y \mid \mathrm{do}(X))$ is the relevant quantity for predicting outcomes of chosen actions [32]. This framing helps isolate why joint action-likelihood terms must be excluded from evidence in mixtures, even though action probabilities remain meaningful for generation.

Imitation, inverse RL, and ambiguity. Inverse reinforcement learning explains demonstrations by inferring a reward function that rationalizes expert behavior, with additional principles often used to resolve ambiguity among explanations [26, 45]. We take a different stance: hypotheses are generative programs for interactive continuation, and behavior is induced directly by posterior-weighted action channels under the intervention posterior, without positing reward as the semantic primitive. Interactive imitation methods such as DAgger are motivated by the distribution shift induced when the learner's actions change the visited states [36]; our intervention/evidence split isolates the underlying epistemic discipline and makes explicit that what can be learned depends on which intervention contexts are actually tested. The universal imitation protocol in Section 4 makes this dependence concrete by introducing counterfactual continuation targets that are sometimes revealed (as third-party actions) and sometimes demanded (as agent interventions).

RL as sequence modeling and action-conditioning pathologies. Trajectory and sequence-modeling approaches treat control as inference in a learned generative model of interaction histories [3, 12]. This is close in representation to our joint model on interactions, but it also highlights a known pitfall: if one conditions on self-generated actions as if they were evidence, joint models can exhibit spurious certainty and "self-delusion" [30]. The intervention posterior makes the required correction explicit.

Preference learning, feedback, and prompt-indexed schemas. Modern preference-learning pipelines infer a training signal from feedback (often comparisons) and then optimize behavior to satisfy it [4, 31, 40, 46]. Our framing treats feedback as just another kind of observation token in an interaction transcript: the agent's outputs are interventions that select which feedback will be revealed, and only that revealed feedback updates beliefs. This supports the manuscript's "rewards are one observation among many" stance while keeping the same first-person discipline. The preference-following corollaries in Section 4 can be read as an idealized limit statement for prompt-indexed choice rules: preferences are absorbed as computable continuation schemas and then incorporated into the agent's behavior. Related alternatives such as direct preference optimization emphasize log-probability ratios relative to a reference model for fitting preferences [34]; here, log-ratios arise as evidential increments in a posterior over interaction generators, not as an optimization objective.

LLM agents and tool-use interaction. Interactive language-model agents instantiate the same intervention/evidence split in practice: tool calls and protocol steps function as interventions (action tokens), while tool outputs, environment responses, and labels function as observations [25, 37, 44]. Our single-stream + gate formalism is designed to model such variable-length, protocol-driven turns without hard-coding a fixed action–observation rhythm, while the intervention posterior provides the explicit guardrail against treating the agent's own emitted tokens as evidence about the world.

7. Conclusion

This paper offers a different starting point for universal AI. Instead of defining agency by the optimization of an exogenously specified objective, we treat agency as *first-person inference in interaction*: the agent forms compact generative explanations of how the world's replies depend on what the agent does, and behavior follows from those explanations. The reason this matters is simple: much of real intelligence is acquired second-hand—through demonstrations, language, tools, and feedback—and those patterns are not naturally or reliably reducible to a single reward channel without losing what makes them action-guiding. The resulting type of agent is thus not reward-optimal, nor does it come with safety guarantees.

Technically, the construction is deliberately minimal. We extend Solomonoff's mixture from passive strings to interaction by placing a universal mixture over computable generators on a single shared symbol stream and reading that stream through an interface. Hypotheses have no intrinsic action/observation ontology: each is simply a program that generates symbol continuations. The action–observation distinction is supplied by the interface: a computable gate partitions the stream into alternating blocks, defining who writes next and where token boundaries occur. In this sense, the universal mixture is a universal *pattern generator* for transcripts, and "actions" are the read-outs demanded when the gate assigns the next block to the agent. This separation is reminiscent of *reservoir computing* [11, 23]: a generic dynamical substrate provides rich temporal structure, while semantics enter only through a readout—what is read out, and when.

The agent/world asymmetry is imposed *epistemically* by one rule: completed agent-written tokens are *choices* and therefore not evidence, while completed world-written tokens are evidence. This

yields an *intervention posterior* whose weights update only through the world's responses to the agent's interventions. This first-person discipline is the principled fix for the characteristic pathology of joint sequence models, where naive conditioning would spuriously reward hypotheses that happen to predict the agent's own sampled behavior. Once that first-person correction is in place, acting becomes the generative counterpart of predicting "what belongs next" in the transcript. This account fits modern autoregressive learning (e.g. large language models) unusually well: the same continuation structure that explains outcomes also produces behavior, without requiring a separate policy–environment factorization to make "agency" meaningful.

The payoff is an account of behavioral schema acquisition that does not treat reward, utility, or costs as the semantic primitive. "What to do next" is grounded in a counterfactual target: what the world would have written next if the world had continued instead of handing control to the agent. In ordinary interaction, the world sometimes carries a pattern forward (so the continuation arrives as evidence) and sometimes the agent must supply it (as a choice), without either side needing to represent a teacher–student protocol. Because the agent must use the same continuation rule either way, the world's continuations anchor the agent's own, yielding a finite cumulative divergence guarantee to these counterfactual targets. Over time, many such regularities compress into reusable schemas and are recombined at decision time, so behavior need not look like imitation at all—it can look like purpose emerging as stabilized continuation under interventions.

This perspective also suggests a developmental reading. Early in life, a learner cannot be expected to carve the stream into useful regularities on its own: many different programs can remain compatible with what has been observed while implying very different next completions when the learner is asked to act. The practical role of early training is therefore to teach basic schemas first—stable interface conventions, reusable templates, and simple action–outcome regularities—so later interaction is not underdetermined. After this bootstrapping phase, training becomes naturally curricular: the world supplies progressively richer regularities that build on the installed schemas. This is close in spirit to Alan Turing's "child machine" idea: begin with a simple initial mechanism and let competence arise through education and experience [43].

It is also useful to separate two coupled sources of evidence in the world-written stream. One is *external*: tasks, tools, other agents, feedback, demonstrations, and environmental consequences. The other is *internal*: regularities tied to embodiment and implementation—timing constraints, controllability, action costs, systematic side effects, and (more broadly) the stable normative and procedural conventions that make actions mean something operationally. Crucially, this internal channel need not be passive: it can be arranged to provide part of the curriculum itself—for example, by functioning as a teacher coupled to developmental phases, exposing and shaping those internal regularities in a staged way (as in early childhood). Early exposure to such internal structure can anchor basic competent behavior and make later external learning coherent.

Acknowledgements

The author thanks Daniel A. Braun for joint foundational work on the Bayesian control rule during the author's PhD. The author also thanks Nando de Freitas, Greg Wayne, and Joel Veness for feedback after reading a short addendum to [30].

References

[1] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss,

G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[3] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, pages 15084–15097, 2021.

[4] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *CoRR*, abs/1706.03741, 2017.

[5] A. P. Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society. Series A (General)*, 147(2):278–292, 1984.

[6] A. P. Dawid and V. G. Vovk. Prequential probability: principles and properties. *Bernoulli*, 5(1):125–162, 1999.

[7] A. Gibbard and W. L. Harper. Counterfactuals and two kinds of expected utility. In C. A. Hooker, J. J. Leach, and E. F. McClennen, editors, *Foundations and Applications of Decision Theory*, volume II, pages 125–162. D. Reidel, 1978.

[8] P. D. Grünwald. *The Minimum Description Length Principle*. Adaptive Computation and Machine Learning. MIT Press, 2007.

[9] J. C. Harsanyi. Games with incomplete information played by "bayesian" players, part I. the basic model. *Management Science*, 14(3):159–182, 1967.

[10] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Springer, 2005.

[11] H. Jäger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, Apr. 2004.

[12] M. Janner, Q. Li, and S. Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2021.

[13] R. C. Jeffrey. *The Logic of Decision*. McGraw-Hill, New York, 1965.

[14] J. M. Joyce. *The Foundations of Causal Decision Theory*. Cambridge University Press, 1999.

[15] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[16] F. Lattimore, T. Lattimore, and M. D. Reid. Causal bandits: Learning good interventions via causal inference. In *Advances in Neural Information Processing Systems*, 2016.

[17] S. Legg. Is there an elegant universal theory of prediction? In *International Conference on Algorithmic Learning Theory*, pages 274–287. Springer, 2006.

[18] J. Leike and M. Hutter. On the computability of Solomonoff induction and knowledge-seeking. In K. Chaudhuri, C. Gentile, and S. Zilles, editors, *Algorithmic Learning Theory*, volume 9355 of *Lecture Notes in Computer Science*, pages 364–378. Springer, Cham, 2015.

[19] J. Leike, T. Lattimore, L. Orseau, and M. Hutter. Thompson sampling is asymptotically optimal in general environments. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.

[20] L. A. Levin. Laws of information conservation (non-growth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974.

[21] D. Lewis. Causal decision theory. *Australasian Journal of Philosophy*, 59(1):5–30, 1981.

[22] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 4 edition, 2019.

[23] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, Nov. 2002.

[24] R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.

[25] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, M. Chen, P. Jain, V. Yu, J. Hutchinson, P. Mishkin, et al. WebGPT: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021.

[26] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, 2000.

[27] P. A. Ortega and D. A. Braun. A Bayesian rule for adaptive control based on causal interventions. *CoRR*, abs/0911.5104, 2009.

[28] P. A. Ortega and D. A. Braun. A Bayesian rule for adaptive control based on causal interventions. In *Proceedings of the Third Conference on Artificial General Intelligence (AGI 2010)*, pages 121–126. Atlantis Press, 2010.

[29] P. A. Ortega and D. A. Braun. A minimum relative entropy principle for learning and acting. *Journal of Artificial Intelligence Research*, 38:475–511, 2010.

[30] P. A. Ortega, M. Kunesch, G. Delétang, T. Genewein, J. Grau-Moya, J. Veness, J. Buchli, J. Degrave, B. Piot, J. Perolat, T. Everitt, C. Tallec, E. Parisotto, T. Erez, Y. Chen, S. Reed, M. Hutter, N. de Freitas, and S. Legg. Shaking the foundations: delusions in sequence models for interaction and control, Oct. 2021.

[31] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

[32] J. Pearl. *Causality*. Cambridge University Press, 2009.

[33] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.

[34] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *CoRR*, abs/2305.18290, 2023.

[35] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[36] S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[37] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, L. Lombardo, L. Zettlemoyer, and T. Scialom. Toolformer: Language models can teach themselves to use tools. *CoRR*, abs/2302.04761, 2023.

[38] R. J. Solomonoff. A formal theory of inductive inference. part I. *Information and Control*, 7(1):1–22, 1964.

[39] R. J. Solomonoff. A formal theory of inductive inference. part II. *Information and Control*, 7(2):224–254, 1964.

[40] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize from human feedback. *CoRR*, abs/2009.01325, 2020.

[41] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

[42] C. Toth, L. Lorch, C. Knoll, A. Krause, F. Pernkopf, R. Peharz, and J. von Kügelgen. Active Bayesian causal inference. In *Advances in Neural Information Processing Systems*, volume 35, pages 16261–16275, 2022.

[43] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[44] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. ReAct: Synergizing reasoning and acting in language models. *CoRR*, abs/2210.03629, 2022.

[45] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 2008.

[46] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. F. Christiano, and G. Irving. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593, 2019.