Active Inference – A Synthetic Overview

An executable introduction to Active Inference and synthetic intelligences across equilibrium and non-equilibrium settings

Daniel Ari Friedman

Active Inference Institute

daniel@activeinference.institute

ORCID: 0000-0001-6232-9096

Alexandra Mikhailova

Active Inference Institute

alexandra@activeinference.institute

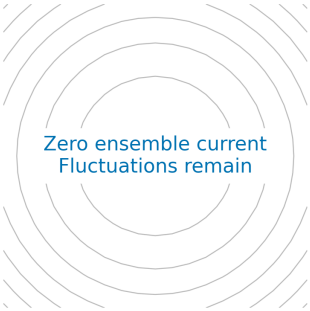
ORCID: not-provided

DOI: Revised manuscript; archived edition: 10.5281/zenodo.23321281

2026-10-12

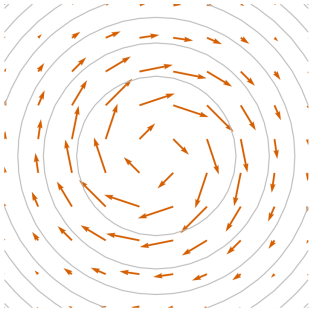
ONE DENSITY, TWO STATIONARY REGIMES

EQUILIBRIUM



$j_* = 0, \sigma = 0$ | detailed balance

NON-EQUILIBRIUM STEADY STATE



$\nabla \cdot j_* = 0, \sigma > 0$ | circulation

$\rho_* \propto e^{-\Phi}$ | $D = \Gamma > 0$ | both drift paths relax

Even coordinates under time reversal; current concerns a stochastic ensemble.

ACTIVE INFERENCE: A SEPARATE GENERATIVE-MODEL LAYER

Observe → infer beliefs (VFE) → evaluate policies (EFE) → act

Blankets: conditional independence | agency: explicit sensing and control

GNN

Specify variables, parameters and equations

SYNTHETIC

Compute examples, controls and figures

FEP_FORMAL

Import Lean results with stated hypotheses

Specification ≠ execution ≠ proof ≠ physical interpretation

Contents

1 Abstract **2**

2 Introduction and Extended Abstract **3**

3 Perception as Inference **5**

4 Methods and Reproducibility **6**

5 Equilibrium, Relaxation and Non-Equilibrium Steady States **8**
5.1 A common density does not imply a common process . . . 8
5.2 Relaxation, circulation and entropy production . . . 8
5.3 From stationary physics to inference . . . 9

6 Worked Regimes: Equilibrium and Non-Equilibrium **10**

7 Action: Variational vs Expected Free Energy **12**

8 The Markov Blanket **13**
8.1 A matched native Gaussian example . . . 14

9 The Steady State, Quantified **15**

10 Formal Reference and Verification **17**
10.1 Selected results and hypotheses . . . 17
10.2 Numerical and formal evidence . . . 17
10.3 Qualified blanket and planning adapters . . . 17

11 Thermodynamic Proof Imports and GNN Specifications **18**
11.1 What is actually imported from FEP_Formal . . . 18
11.2 A concrete GNN representation of both settings . . . 18

12 Planning: Policies as Action Sequences **19**
12.1 An explicit upstream carrier bridge . . . 19

13 A Closed-Loop Synthetic Agent **20**

14 When Information Improves a Decision **21**

15 A Two-Level Generative Hierarchy **23**

16 Discussion: Informational and Thermodynamic Interpretation **24**
16.1 Equilibrium and stationary circulation . . . 24
16.2 Statistical boundaries and synthetic agents . . . 24
16.3 Reading and reusing the Institute repositories . . . 24

17 A Reader's Route Beyond Synthetic **26**

1 Abstract

This tutorial connects Active Inference and synthetic intelligences to explicitly specified informational and dynamical models. Finite categorical examples make Bayesian conditioning, variational free energy, expected free energy, policy planning, and observation-contingent commitments executable. A decision-value example separates information gain in nats from improved expected utility and observation cost. Gaussian examples distinguish deterministic relaxing paths from stationary ensemble probability current, and marginal dependence from conditional independence. A two-level model illustrates prediction and joint conditioning without assuming a neural implementation. The standalone Python package supplies reproducible figures, independent analytical checks, and counterexamples; a separate Lean layer imports selected real-valued results from a pinned revision of the broader FEP_Formal library. Generalized Notation Notation and other Active Inference Institute resources provide further routes for model specification and learning. The paper distinguishes computational correctness, theorem hypotheses, thermodynamic units, and scientific interpretation. These controlled demonstrations explain particular models; they neither establish general intelligence nor infer life from circulation.

2 Introduction and Extended Abstract

This is a worked introduction to Active Inference and synthetic intelligences across equilibrium and non-equilibrium thermodynamic and informational settings. The examples connect the Free Energy Principle and Bayesian mechanics to inference, action, and self-organization. The broader `fep_formal` library provides the mathematical reference [Friedman, 2026a,b]; this paper supplies selected executable models and their exposition. Variational, expected, and thermodynamic free energy have distinct definitions and units. Persistent stationary current alone is not evidence of life or intelligence.

Active Inference connects inference and action within a specified generative model [Friston, 2010, Parr et al., 2022]. Updating a candidate belief changes its variational free energy after an observation. Choosing a future policy instead requires an objective over predicted outcomes, with explicit preferences and information terms. Those objectives are related by their model assumptions; they are not interchangeable quantities. Whether a biological or engineered system realizes such computations is a further scientific question.

On the admitted support domain, variational free energy bounds **surprise**, the negative log evidence $-\ln p(s)$. The finite demonstration can compute both the exact posterior and that evidence independently, so the identity has a direct numerical check. Relating this inference calculation to physical dynamics requires additional assumptions. Likewise, a **Markov blanket** states conditional independence between *external* states η and *internal* states μ given *sensory* and *active* blanket states. That screening condition alone does not establish causal direction or a representational mapping. **Bayesian mechanics** investigates such mappings under specified dynamical and statistical hypotheses [Da Costa et al., 2021, Ramstead et al., 2023].

fig. 1 shows the move that this overview is built around, without a single equation of derivation. Both panels live on *the same* free-energy landscape — a Gaussian bowl whose floor is the system's expected state. At **equilibrium** (panel a) the flow is pure descent, $f = -\nabla F$: every trajectory falls straight to the mode and stops. Detailed balance holds, the stationary probability current vanishes. This does not mean that stochastic microscopic motion ceases. Panel (b) supplies a contrasting non-equilibrium example. There the flow gains a *solenoidal* term, $f = (Q - \Gamma)\nabla F$, that runs along the contours of F rather than down them. The deterministic drift still lowers the quadratic potential, while stationary probability mass **circulates** indefinitely in a **non-equilibrium steady state**. That persistent current diagnoses broken detailed balance in the stated diffusion model. The gradient part illustrates the Free Energy Principle's descent; the solenoidal part supplies conservative circulation in this example [Friston et al., 2023].

Read the two panels together and the headline of Active Inference is visible at a glance: the *same* imperative — minimize free energy — produces relaxation in one regime and persistent stationary current in the other. The rest of this overview unpacks that picture in the spirit of *show, not tell*: each section adds one runnable, minimal demonstration rather than another layer of formalism. sec. 3 makes the *inference* concrete — that minimizing variational free energy over a generative model **is** Bayesian updating, and that free energy is a tight upper bound on surprise. sec. 6 makes the *mechanics* concrete — integrating the flow above into worked equilibrium and non-equilibrium trajectories and quantifying the persistent current that separates these two model regimes. sec. 7 turns to *action* — evaluating expected free energy over policies, whose pragmatic and epistemic terms produce the explore/exploit balance from one rule. sec. 8 exhibits conditional screening in a specified Gaussian model; a representational interpretation requires further hypotheses. sec. 9 quantifies the non-equilibrium regime outright — the analytic probability current and the entropy-production rate that is zero exactly at detailed balance and positive out of it. Finally sec. 10 closes the loop on rigour: the load-bearing theorems — Gibbs' inequality, the free-energy bound, the conservative solenoidal current, the sign of entropy production — are **machine-checked in Lean 4**, proved from the kernel's axioms with no `sorry`: the real-valued exposition imports the `FepSketches` library maintained in `fep_formal`.

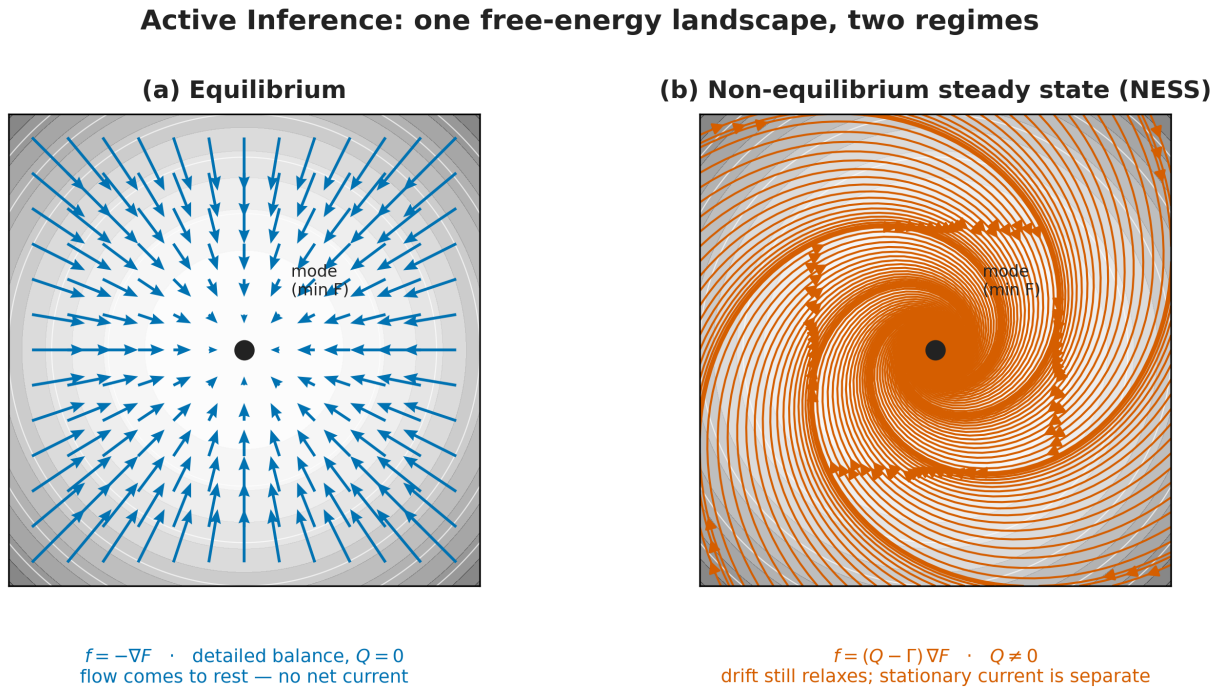


Figure 1: **One landscape, two drift regimes.** Grey contours show the same Gaussian dimensionless potential; the black dot is its mode. **(a)** Zero drive gives radial deterministic relaxation. **(b)** An antisymmetric operator adds tangential motion to the same dissipative drift, producing a relaxing spiral. With matched diffusion, these drifts correspond to equilibrium and a non-equilibrium steady state, respectively; stationary current is computed separately in sec. 9. The diagram does not establish a universal identification of FEP with gradient descent, or a belief interpretation from a statistical partition alone. Rendered from the tested numerical flow.

3 Perception as Inference

The extended abstract makes a strong claim in one sentence: *minimizing variational free energy **is** approximate Bayesian inference, carried out by the physical dynamics of the thing itself.* This section makes that claim runnable. It is the smallest generative model that still says something true, and everything below is checked against closed-form Bayes in `src/generative_model.py`.

A system holds a **generative model** of a hidden cause x and a sensation s: a prior $p(x)$ over states and a likelihood $p(s \mid x)$ saying which sensations each state tends to produce. Together they fix the joint $p(s, x) = p(s \mid x)\, p(x)$, the evidence $p(s) = \sum_x p(s, x)$, and — by Bayes' rule — the posterior $p(x \mid s) = p(s, x)/p(s)$. The system never sees x; it only ever has s and its own model. Inference is the problem of recovering $p(x \mid s)$ from those.

Active Inference solves it without ever computing $p(s)$ directly, through a belief $q(x)$ and a single scalar, the **variational free energy**:

$$F[q] \;=\; \underbrace{\mathrm{KL}[q(x) \,\|\, p(x)]}_{\text{complexity}} \;-\; \underbrace{\mathbb{E}_q[\ln p(s \mid x)]}_{\text{accuracy}}. \tag{1}$$

The decomposition in eq. 1 separates complexity from accuracy. Two facts, both asserted to machine precision in the tests, make F the whole game. First, F **is an upper bound on surprise**: $F[q] = -\ln p(s) + \mathrm{KL}[q \| p(x \mid s)] \geq -\ln p(s)$, because a KL divergence is never negative. Surprise $-\ln p(s)$ is exactly what an organism "wants" to keep low — the improbability of its own sensory state — but it cannot compute it. It *can* compute F, from quantities it already has. Second, **the bound is tight at the posterior**: F is least, and equals the surprise, precisely when $q = p(x \mid s)$. So the belief that minimizes free energy is the Bayesian posterior, and the minimized value is the surprise it bounds. Inference falls out of minimization.

fig. 2 shows this happening. Take a two-state model with prior $p(x) = (0.60, 0.40)$ and a likelihood under which state x_0 makes the observed sensation s far more probable than x_1 does. Panel (a) starts from a maximally uncommitted uniform belief and runs gradient descent on F over the belief simplex; the belief mass migrates toward the exact posterior $p(x_0 \mid s) = 0.54/0.62 \approx 0.87$. Panel (b) tracks F across the same descent: it falls **monotonically** to the surprise $-\ln p(s)$ (dashed line), and the shrinking shaded gap is exactly $\mathrm{KL}[q_t \| p(x \mid s)]$ — the belief's remaining error. Perception, here, is nothing but this descent: the system lowers free energy by updating beliefs, and in doing so recovers the posterior it could not compute directly.

The complexity–accuracy split names the trade-off. *Accuracy* rewards beliefs that explain the sensation; *complexity* penalizes moving away from the prior. Perception moves beliefs only as far as the evidence warrants — the same regularized fit that makes the free-energy principle a principle rather than a slogan. Action, treated in the next demonstrations, minimizes the *same* F in the other direction: by changing s rather than q.

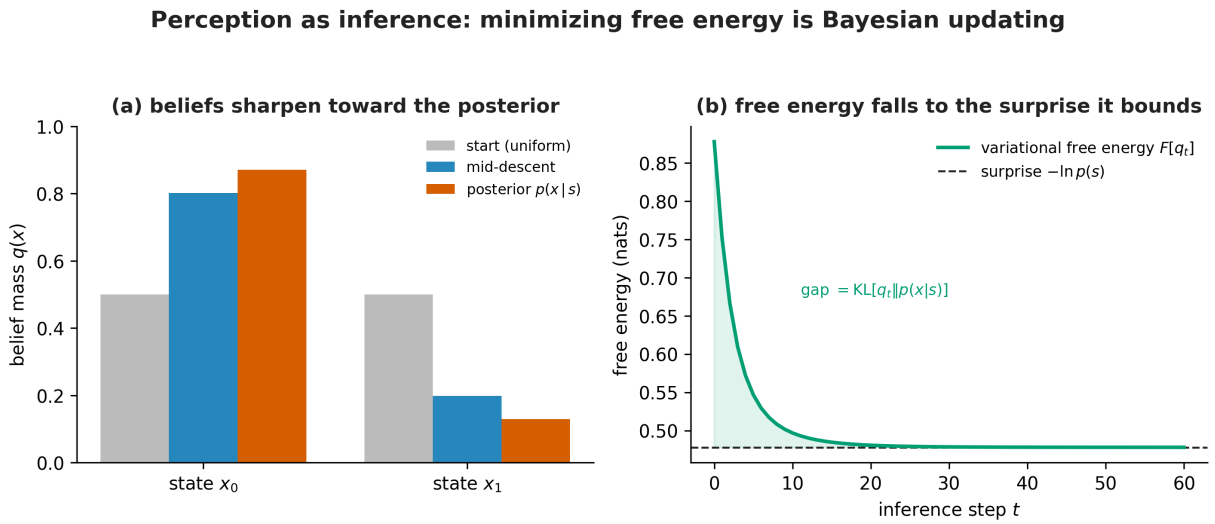


Figure 2: **Perception as free-energy minimization.** A two-state generative model (prior $p(x) = (0.60, 0.40)$; likelihood favouring x_0 for the observed s). **(a)** From a uniform starting belief, gradient descent on variational free energy drives the belief over hidden states to the exact Bayesian posterior $p(x \mid s)$ (orange), passing through intermediate beliefs (blue) on the way. **(b)** Along the same descent the free energy $F[q_t]$ falls monotonically to the surprise $-\ln p(s)$ (dashed) that it bounds; the shaded gap is exactly $\mathrm{KL}[q_t \| p(x \mid s)]$, the belief's error, which closes as inference completes. Minimizing free energy is Bayesian updating. Rendered deterministically by `scripts/generate_belief_update_figure.py` over the tested core in `src/generative_model.py`.

4 Methods and Reproducibility

The examples are deliberately small models with declared inputs, rather than empirical estimates of biological or artificial-system performance. Numerical parameters are shared between figure builders and tests in `src/demos.py`. Manuscript tokens are computed by the same numerical cores and hydrated before the template renders the paper. Each figure caption identifies the corresponding model or script; the figure registry binds labels, filenames and builders.

Finite probability channels store outcomes in rows and hidden states in columns. Their columns are normalized; beliefs are probability vectors. Conditioning enumerates reachable observations or observation histories. The decision-value calculation compares the best prior commitment with optimal observation-contingent commitments under the same state, actions, utilities and acquisition cost. Impossible outcomes do not acquire invented posteriors. These finite enumerations cover their specified branch spaces, while their Python arithmetic remains floating point with explicit validation and test tolerances. They do not establish an optimum outside the declared model class.

Independent references are chosen for each model: direct Bayes conditioning for belief updates, alternative free-energy decompositions, exhaustive deterministic feedback policies for decision value, the linear closed form for RK4 paths, and Gaussian covariance and precision identities for blanket screening. The OU stationary calculations additionally require matched diffusion and positive dissipation. A deterministic path is not substituted for stationary current. Information is reported in nats; utility and observation cost share declared utility units; physical thermodynamic interpretations require extra scaling.

Counterexamples exercise interpretation boundaries. Independent cues provide no context information. Informative cues with state-independent utilities give no instrumental gain. Excessive observation cost can make net gain negative. Partial blanket conditioning and altered precision couplings need not preserve screening. Invalid probabilities, support mismatches and nonfinite quantities are rejected. The formal tests also check that false Lean statements fail, rather than relying only on successful proof builds.

The package's installed namespace is `active_inference_synthetic`. Reproduction from the standalone project uses:

```
uv sync --locked --extra dev
uv run pytest tests/ --cov=src --cov-fail-under=90
uv run python scripts/generate_all_figures.py
```

```
uv run python scripts/run_teaching_experiments.py
```

The last command records inputs, computed decision values and source hashes for the informative, uninformative and irrelevant-payoff conditions. A changed or missing source changes or blocks that receipt. Such a receipt establishes the inputs of a particular execution; it is not certification of a scientific interpretation.

The independent formal layer builds with `lake build` from `formal/` and pins FEP_Formal through Lake. The publication renderer belongs to the template repository. Python execution, Lean checking, rendered-document inspection and publication-file integrity are separate verification steps. The accompanying source archive supplies the authored numerical, test, formal and manuscript files; it excludes local environments, credentials and intermediate runtime state. The resource guide identifies further mathematical and educational materials sec. 17.

5 Equilibrium, Relaxation and Non-Equilibrium Steady States

Three questions organize the dynamical examples: **what distribution is stationary, which probability currents persist, and what changes under time reversal?** They distinguish a snapshot of a density, a deterministic drift path, and a stochastic process. Equilibrium is a property of the process under specified reversal and boundary conditions, not merely a minimum in a plot. This treatment follows stationary Bayesian mechanics and stochastic thermodynamics [Da Costa et al., 2021, Seifert, 2012, Da Costa and Pavliotis, 2023].

5.1 A common density does not imply a common process

Write $\Phi(x) = -\ln \rho_*(x) + \text{constant}$ for the dimensionless stationary potential. This notation separates a function of physical/model coordinates from the variational functional $F[q]$ of a recognition distribution. The code uses `free_energy` for this Gaussian potential; that name does not establish an identity between the two mathematical objects. For constant positive-definite diffusion D, the Fokker–Planck equation and current are

$$\partial_t \rho_t = -\nabla \cdot j_t, \qquad j_t = b\rho_t - D\nabla\rho_t. \tag{2}$$

Stationarity requires $\nabla \cdot j_* = 0$, not $j_* = 0$. In this paper's overdamped, even-coordinate model, with no boundary flux, detailed balance means that stationary forward and reversed transitions agree; its current vanishes. A NESS instead maintains a nonzero divergence-free current. A nonstationary distribution relaxing toward either regime is a **transient**, even when its drift has time-independent coefficients.

Object	Equilibrium stationary process	Non-equilibrium stationary process	Relaxation transient
Density	Time independent	Time independent	Changes with time
Stationary current	Zero in this even-coordinate setting	Nonzero, divergence free	Current need not be divergence free
Detailed balance	Holds	Fails for the stated reversal	A stationary criterion cannot be inferred from a transient snapshot
A noise-free path	May relax to a fixed point	May also relax to that fixed point	Describes drift only

For $P = \Pi I$, $\Gamma = \gamma I$ and the antisymmetric Q, the two OU regimes share $\rho_* \propto \exp(-x^\mathsf{T} P x/2)$ because $D = \Gamma$ and $\gamma, \Pi > 0$. Their currents differ by $j_* = QPx\rho_*$. Rotating the drift does not change this isotropic density. It changes temporal statistics and forward/reverse path probabilities. The cover graphical abstract therefore compares **currents over the same contours**, rather than using a different density as a visual proxy for non-equilibrium.

5.2 Relaxation, circulation and entropy production

Along the deterministic drift, antisymmetry gives $\dot{\Phi} = -\nabla\Phi^\mathsf{T}\Gamma\nabla\Phi \leq 0$. With stochastic forcing, individual paths fluctuate and need not decrease Φ monotonically. At stationarity, fluctuations balance dissipation in the density while the solenoidal current may remain. Constant Q, Γ make this decomposition particularly transparent; state-dependent coefficients require additional derivative terms and regularity conditions [Da Costa and Pavliotis, 2023].

For the stated even-coordinate reversal, path irreversibility and the elliptic stationary current expression agree:

$$\sigma = \lim_{T\to\infty} \frac{1}{T} D_{\mathrm{KL}}(\mathbb{P}_{[0,T]} \| \mathbb{P}^{\mathrm{R}}_{[0,T]}) = \int \frac{j_*^\mathsf{T} D^{-1} j_*}{\rho_*}\, dx. \tag{3}$$

Here $\mathbb{P}^{\mathrm{R}}$ is the stationary time-reversed path law aligned to forward coordinates. The OU result is $\sigma = 2\omega^2\Pi/\gamma$. Positive diffusion makes the inverse well defined. Momentum coordinates, magnetic-field reversal, degenerate diffusion or a different reverse protocol require a separate analysis; a circulating Hamiltonian phase-space current is not automatically dissipative. This qualification prevents the word "solenoidal" from being mistaken for a universal thermodynamic diagnosis [Da Costa and Pavliotis, 2023, Seifert, 2012].

Nonzero ω establishes non-equilibrium in this model, but does not by itself establish **far** from equilibrium. The ratio $|\omega|/\gamma$ compares rotational and dissipative strengths; σ depends additionally on precision and the model time unit. Neither is an empirical measure of a living system's metabolic cost without a physical calibration. Near-equilibrium linear response and a strongly driven regime belong to a continuum of parameter choices, not two different kinds of Gaussian density.

5.3 From stationary physics to inference

Bayesian mechanics relates suitably partitioned stochastic systems to belief parameterizations under explicit assumptions [Ramstead et al., 2023, Da Costa et al., 2021]. A stationary density alone supplies neither a generative model nor an agent's action set. Synthetic therefore adds finite inference, policy selection and feedback as separate demonstrations. A blanket's conditional independence, an inference algorithm's VFE descent and an OU process's stationary current answer different questions. Establishing their relationship requires a specified parameter map, sensing and control channels, and hypotheses that survive the transfer; it cannot be inferred from similar looking gradients [Friston, 2010, Friston et al., 2023].

6 Worked Regimes: Equilibrium and Non-Equilibrium

The overview figure (fig. 1) contrasts two *flows* — one that comes to rest, one that circulates — as arrow fields. A flow field says where a system would go next; it does not, on its own, show where a system *ends up*. This section integrates the flow into actual **trajectories**, so the equilibrium↔non-equilibrium distinction becomes a path you can watch, and the rotation diagnostic becomes a number rather than an adjective. The integrator and its diagnostics live in `src/trajectories.py`, built directly on the tested flow of `src/free_energy_flow.py`.

Both regimes obey the Bayesian-mechanics flow $f = (Q - \Gamma)\nabla F$ on the *same* isotropic Gaussian free-energy landscape. The only difference is the antisymmetric solenoidal operator Q: zero at equilibrium, nonzero in a non-equilibrium steady state (NESS). A fixed-step fourth-order Runge–Kutta scheme (no adaptive stepping, no randomness) advances a starting point under this flow. That the integrator can be trusted is not assumed: at equilibrium the flow reduces to the linear system $\dot{x} = -\gamma\,\Pi\,x$ (with Π the precision), and the tests confirm the RK4 path matches the closed form $x(t) = x_0\,e^{-\gamma\,\Pi\,t}$ — i.e. $x_0 e^{-\gamma t}$ at unit precision — to better than 10^{-6}.

fig. 3 shows the two fates. In panel (a), the **equilibrium** trajectory ($Q = 0$) falls straight down the free-energy gradient to the mode and stops. Free energy decreases monotonically along the path, the radius collapses to under 1% of its starting value, and the net angle swept about the mode is zero to numerical precision. Detailed balance and zero stationary current concern the corresponding matched-noise diffusion, not this single noise-free path.

Panel (b) is the **NESS** ($Q \neq 0$). The dissipative term still pulls the system down F — free energy again decreases monotonically, so the system is *still* minimizing surprise — yet the solenoidal term drives it *around* the level sets of F at the same time. The result is a spiral that sweeps many full turns before relaxing inward.

How much it winds is worth stating carefully. The *total* net rotation — the figure's ≈ -4.58 turns over the demonstration window $T = 12$ — is not an intrinsic property of the NESS; it grows with T and would diverge as $T \to \infty$. The intrinsic, initial-condition-free quantity is the **rotation rate**. For this isotropic-Gaussian flow the angular velocity is analytically constant, $\dot{\theta} = -\omega\,\Pi$ radians per unit time (with Π the precision), so the rate is ≈ -0.38 turns per unit time ($-\omega\,\Pi/(2\pi)$ at $\omega = 2.4$, $\Pi = 1$) for *any* starting point off the mode — and it is exactly 0 at equilibrium for every window. That zero-versus-nonzero rate diagnoses the antisymmetric drive in this isotropic model; stationary current establishes the detailed-balance consequence. The tests pin the rate directly: net rotation equals $-\omega\,\Pi\,T$ across several starting points to a tenth of a percent.

One more distinction keeps the claim precise. Taken on its own, the deterministic trajectory here *relaxes* — it spirals into the mode and would sit there. A genuine non-equilibrium steady state is a property of the *stochastic* system: a stationary density $\rho \propto e^{-F}$ carrying a divergence-free probability current $j = Q\nabla F\,\rho$ that never dies. What the figure shows is that system's **deterministic drift**, and the spiral is the mechanistic *origin* of that current, not a measurement of it. So the honest reading of "NESS" here is precise: the nonzero winding **witnesses** the solenoidal $Q \neq 0$ that breaks detailed balance and thereby renders the corresponding stochastic stationary state a non-equilibrium one. Reverse the sign of Q and the circulation reverses with it, as the tests confirm. (sec. 9 supplies the measurement this section withholds — the actual probability current and its entropy-production rate.)

Read the two panels against the paper's teaching aim. Descent is common to both: the free-energy landscape decreases along each deterministic trajectory. Nonzero solenoidal drive additionally produces rotation. Persistent stationary probability current belongs to the corresponding stochastic model and is quantified in the next steady-state section. These are distinct demonstrations of relaxation and non-equilibrium circulation; they do not establish life, intelligence, or perpetual motion of an individual deterministic trajectory.

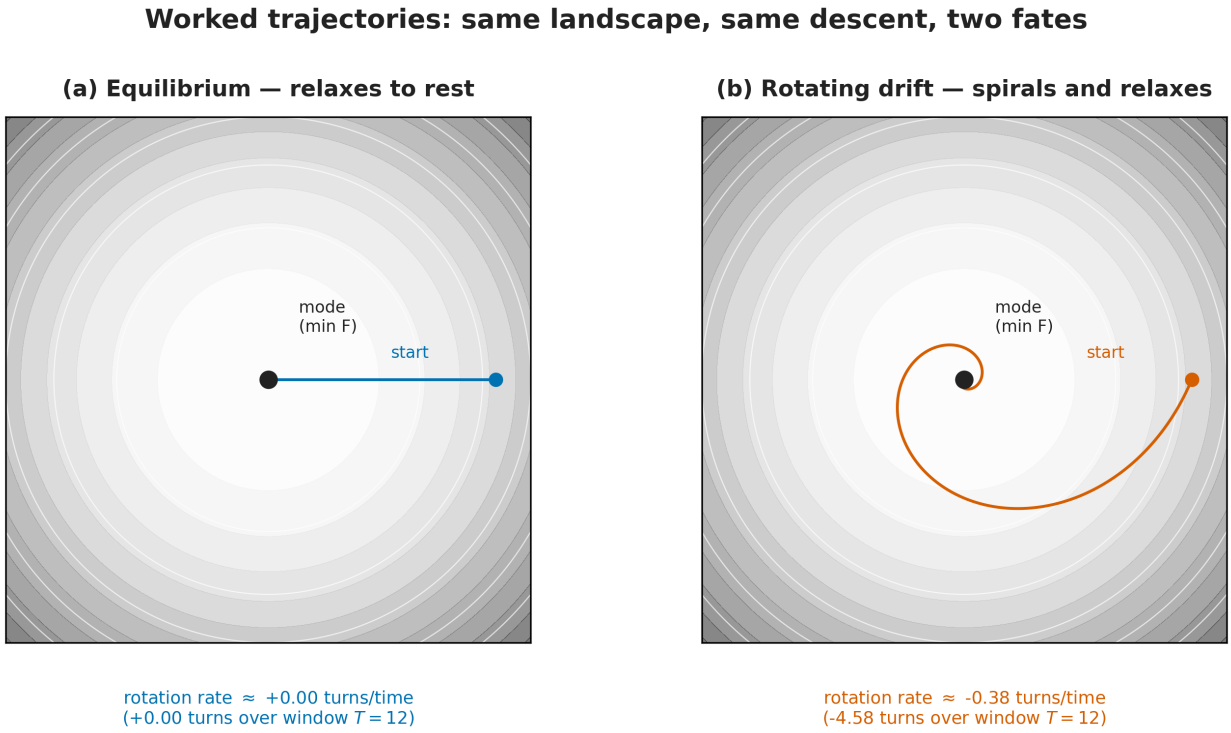


Figure 3: **Worked trajectories over one landscape.** Both panels integrate the Bayesian-mechanics flow $f = (Q - \Gamma)\nabla F$ from the same starting point (coloured dot) over the same Gaussian free-energy bowl (grey contours; black dot marks the mode). **(a) Equilibrium** ($Q = 0$): pure descent falls radially to the mode and rests — free energy monotonically decreasing, rotation rate ≈ 0 turns/time. **(b) NESS** ($Q \neq 0$): the same descent plus a solenoidal term spirals inward at a constant rate ≈ -0.38 turns/time (≈ -4.58 turns over the window $T = 12$) while free energy still decreases monotonically. The nonzero rotation rate witnesses the broken detailed balance in the corresponding stochastic model; sec. 9 computes its stationary probability current. Rendered deterministically by `scripts/generate_trajectory_figure.py` over the tested cores in `src/trajectories.py` and `src/free_energy_flow.py`.

7 Action: Variational vs Expected Free Energy

sec. 3 showed perception: minimizing **variational** free energy F over beliefs about what is happening *now*. But a thing that only perceived would be a spectator. Active Inference closes the loop by minimizing the *same* currency in the other direction — over **actions**, scored by the free energy they are *expected* to produce. That forward-looking quantity is **expected free energy** G, and it is what turns inference into behavior. The core is `src/expected_free_energy.py`; every value below is checked against a hand-worked reference.

The two free energies differ only in tense. $F[q]$ grades a belief against a sensation already in hand — it is retrospective, and its minimum is the posterior. $G(a)$ grades a *candidate action* against the sensations it *would* generate, averaged over what the agent currently believes. For an action a, belief $q(s)$, outcome model $p(o \mid s, a)$, and preferences $p_{\text{pref}}(o)$ — a distribution over the outcomes the agent wants to encounter — G splits into two terms the agent seeks to make large (so it minimizes their negatives):

$$G(a) \;=\; -\underbrace{\mathbb{E}_{q(o|a)}\big[\ln p_{\text{pref}}(o)\big]}_{\text{pragmatic value}} \;-\; \underbrace{\mathbb{E}_{q(o|a)}\big[\mathrm{KL}\big[\,q(s \mid o, a)\,\|\,q(s)\,\big]\big]}_{\text{epistemic value}}. \tag{4}$$

Pragmatic value is expected utility: how well an action's likely outcomes match what the agent prefers. **Epistemic value** is expected information gain — how much the action's outcomes would sharpen the belief about the hidden state, which is exactly the mutual information $I(s; o \mid a)$ and is always non-negative. One is the drive to *get what you want*; the other is the drive to *find out*. Both live inside a single number.

fig. 4 shows why that unification matters. An agent is unsure which of two contexts holds ($q(s) = (0.50, 0.50)$) and can take one of two actions: a **cue** that would reveal the context (highly informative, epistemic value $+0.37$ nats, but preference-neutral) or a **bet** that pays off regardless of context (uninformative, epistemic value 0, but potentially rewarding). The *only* thing that changes between the two panels is the preferences. Under **flat** preferences (panel a) the agent has nothing to chase, so the epistemic term decides: it takes the cue and **explores** ($G = 0.33$ vs 0.69). Under **sharp** preferences (panel b) the pull of the reward dominates, and the same minimization takes the bet and **exploits** ($G = 0.43$ vs 0.84). This is a deliberately constructed two-action example, chosen to isolate the mechanism; what it demonstrates is structural, not incidental to these numbers. The explore/exploit trade-off is not a separate rule bolted onto the agent — there is one rule, $\arg\min_a G$. Because G carries both a pragmatic and an epistemic term, exploration and exploitation are the two limits of a single balance, and which one wins is decided by the preferences alone.

This is the whole architecture in miniature. Perception (F) and action (G) are one imperative — minimize free energy — read in two tenses. Inference makes beliefs fit the world; action makes the world fit beliefs *and* preferences, gathering information where it is uncertain and reward where it is confident.

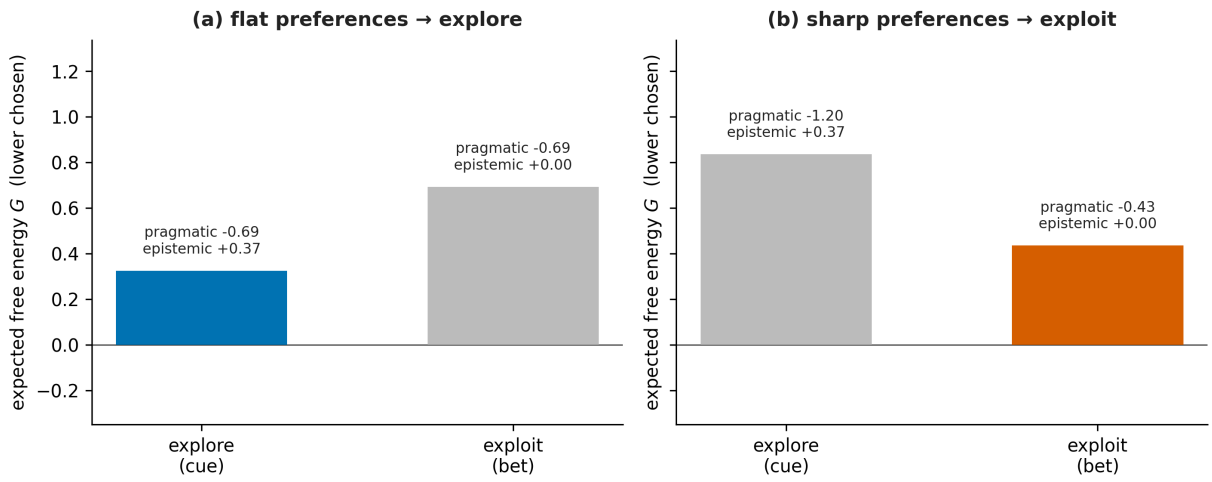


Figure 4: **One rule yields explore and exploit.** An agent uncertain between two contexts ($q(s) = (0.50, 0.50)$) chooses between an informative "cue" (epistemic value +0.37, preference-neutral) and a rewarding but uninformative "bet" (epistemic value 0). Bars are expected free energy G per action (lower is chosen); each is annotated with its pragmatic/epistemic split. **(a) Flat preferences:** the epistemic term dominates, so the agent explores (cue, $G = 0.33$). **(b) Sharp preferences:** the pragmatic term dominates, so the same $\arg\min G$ rule exploits (bet, $G = 0.43$). A single rule; the balance of G's two terms — not a hand-coded switch — is what turns over between the panels. Rendered deterministically by `scripts/generate_efe_figure.py` over the tested core in `src/expected_free_energy.py`.

8 The Markov Blanket

Everything so far assumed a clean split between a system and its world — a belief $q(s)$ about *hidden* states, sensations o that are not those states, actions that reach *out*. That split is not a modeling convenience; it is a structural claim, and it has a name: the **Markov blanket**. This section shows what the blanket actually asserts, using the one case where it is exactly computable — a jointly Gaussian world. The core is `src/markov_blanket.py`.

Partition the variables of a system-plus-environment into four roles: *external* states η (the world), *internal* states μ (the system's own states, which parameterize its beliefs), and the *blanket* that lies between them — *sensory* states s (world → system) and *active* states a (system → world). The blanket's defining property is a **conditional independence**:

$$p(\eta, \mu \mid s, a) \;=\; p(\eta \mid s, a)\, p(\mu \mid s, a). \tag{5}$$

In this example the endpoints are marginally dependent, but their coupling is screened by the blanket. Given sensory and active coordinates, knowing the external coordinate supplies no additional information about the internal coordinate under the stated Gaussian law. A causal interpretation would need a dynamical or interventional model. Conditional independence alone does not establish a posterior encoding or a cognitive interpretation.

fig. 5 makes both halves of the claim visible on a Gaussian built so that the η–μ block of its **precision matrix** (the inverse covariance) is exactly zero. Panel (a) is the ordinary correlation matrix: η and μ are plainly correlated (corr = 0.19): they are statistically coupled through the blanket. Panel (b) is the precision matrix, and the payoff is the highlighted cell: $\Pi_{\eta\mu} = 0$. In a Gaussian a zero precision entry is *exactly* a conditional independence given the remaining variables — so the zero at (η, μ) says $\eta \perp \mu \mid \{s, a\}$. Conditioning on the blanket collapses the partial correlation between internal and external states to zero (the tests confirm it vanishes to 10^{-10} via the Schur complement), even though conditioning on only *part* of the blanket does not — both the sensory and active states are needed to close the channel.

8.1 A matched native Gaussian example

A second fixture matches the pinned upstream canonical four-axis Gaussian carrier. Its ordered variables are external, sensory, active, and internal. The exact precision and covariance arrays live in `src/demos.py`; tests verify their rational inverse relation. External and internal marginal covariance is 0.0417, while conditioning on sensory and active yields a diagonal endpoint covariance. The native Lean adapter binds the canonical precision to a positive-definite Gaussian law and uses upstream conditional independence for every supplied center, including its almost-everywhere conditional-kernel interpretation.

The corresponding perturbed endpoint precision supplies a genuine boundary: adding a nonzero external-internal coupling produces conditional covariance $-1/15$ and native non-independence. The original numerical fixture and this canonical native fixture are distinct examples. Formal coverage is claimed for the matched canonical carrier, not inferred for arbitrary precision matrices from the appearance of a zero cell.

Conditional independence alone does not imply that internal states encode a posterior, that the partition is causal, or that an organism is intelligent. Those interpretations require a generative or dynamical model and additional assumptions. The finite perception-action example makes such a model explicit rather than inferring it from the Gaussian partition.

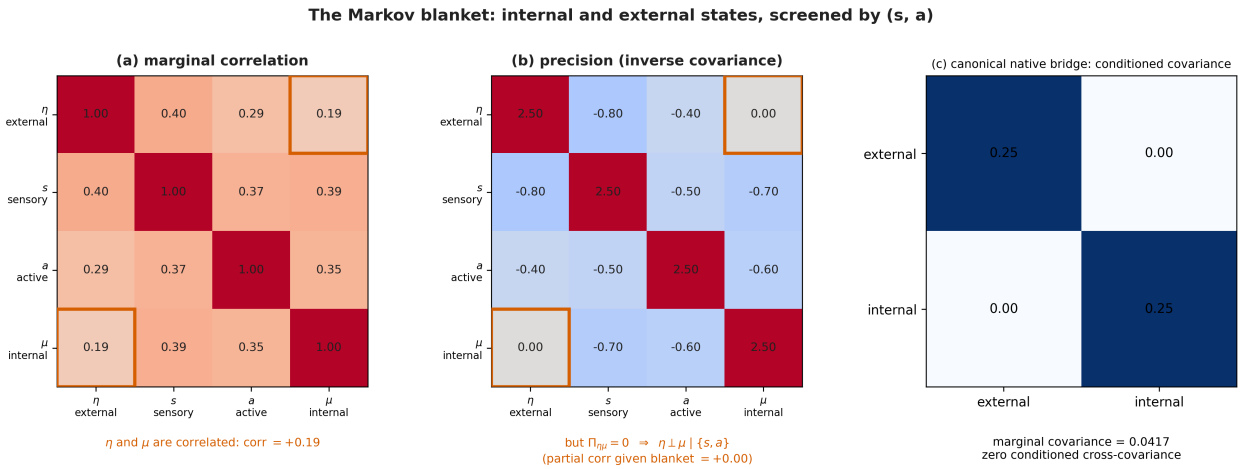


Figure 5: **Marginal dependence, conditional independence.** A jointly Gaussian world over external η, sensory s, active a, internal μ, built with a zero η–μ precision block. **(a) Marginal correlation:** η and μ are correlated (0.19, highlighted) — they are statistically coupled through the blanket. **(b) Precision (inverse covariance):** the η–μ entry is exactly 0, which in a Gaussian *is* conditional independence given the other variables — so $\eta \perp \mu \mid \{s, a\}$ and the partial correlation given the blanket is 0. **(c)** The matched upstream canonical fixture has a diagonal conditioned endpoint covariance despite nonzero marginal coupling. The sensory and active states screen the internal states from the external world. Rendered deterministically by `scripts/generate_markov_blanket_figure.py` over the tested core in `src/markov_blanket.py`.

9 The Steady State, Quantified

sec. 6 was deliberately careful about one thing. The deterministic spiral it drew is the *drift* of a non-equilibrium steady state, not the state itself — a single trajectory relaxes to the mode and stops, while a genuine steady state is a property of a whole *stochastic* ensemble, with a current that never dies. This closing section supplies that missing object, still analytically and with no randomness, in `src/steady_state.py`. It is where non-equilibrium is characterized by two model quantities: a **probability current** and an **entropy-production rate**.

Give the Bayesian-mechanics flow a matched noise term — the Ornstein–Uhlenbeck diffusion $dx = -Mx\,dt + \sqrt{2D}\,dW$ with drift $M = (\Gamma - Q)P$ and, by the fluctuation–dissipation relation, diffusion $D = \Gamma$. Three exact results follow, each checked against a closed form in the tests.

First, the ensemble settles into a **stationary density**. Its covariance solves the Lyapunov equation $M\Sigma + \Sigma M^{\mathsf{T}} = 2D$, and with $D = \Gamma$ the solution is exactly $\Sigma = P^{-1}$, the stationary ensemble covariance. The density is $\rho \propto e^{-\Phi}$, where Φ is the Gaussian potential called `free_energy` by the code. A stationary ensemble has nonzero variance; it is not the point reached by a deterministic path or automatically an agent's recognition distribution.

Second, that density carries a **probability current** $j(x) = b(x)\rho - D\nabla\rho = (QPx)\,\rho(x)$. When $Q = 0$ the current is zero *everywhere* (detailed balance — the density just sits), and when $Q \neq 0$ it is a divergence-free field that circulates around the mode without end. This is the sustained current the deterministic trajectory could only gesture at; here it is an explicit vector field, orthogonal to ∇F (the tests confirm both $\nabla\cdot j = 0$ and $j \perp \nabla F$ to machine precision).

Third, the current has a price. The **entropy-production rate** $\sigma = \mathrm{tr}(D^{-1}QP\,\Sigma\,P^{\mathsf{T}}Q^{\mathsf{T}})$ measures the system's irreversibility, and for the isotropic model it is exactly $\sigma = 2\,\omega^2\,\Pi/\gamma$ (with Π the precision). It is strictly positive out of equilibrium and **exactly zero if and only if** $\omega = 0$. Under even-coordinate time reversal this rate measures path irreversibility; with an isothermal physical model and local thermodynamic assumptions it can represent the cost of maintaining the current. It distinguishes the difference between the two regimes: not the shape of a path, but a non-negative number that vanishes precisely at detailed balance. (Two conventions, stated plainly: the choice $D = \Gamma$ is the fluctuation– dissipation relation with $k_{\mathrm{B}}T \equiv 1$ absorbed, so σ is a pure rate — the dimensional entropy-production rate is $k_{\mathrm{B}}\sigma$ and the dissipated power is $k_{\mathrm{B}}T\sigma$; and crucially the nonequilibrium character is certified by the *current* and σ, **not** by the density — the Gaussian ρ on its own looks equilibrium-like, so it is the circulation, not the shape of the steady state, that marks the regime as nonequilibrium.)

fig. 6 shows both panels over the same stationary density. At equilibrium (a) the current field is identically zero and $\sigma = 0$: there is no net stationary circulation or entropy production, although stochastic fluctuations and energy exchange remain. In the NESS (b) the current curls steadily around the mode and $\sigma > 0$: the same density, now maintained by a perpetual, entropy-producing flow. Read against the whole overview, this is the final turn of the one thesis — and the point at which the mathematics reconnects to the motivation. A system that minimizes free energy need not merely come to rest at its expected states; a *nonequilibrium* steady state holds those states while circulating through them, at a strictly positive entropy cost. The model makes the sustained current and its irreversible cost precise and measurable under the stated diffusion assumptions. This motivates connections to living and synthetic systems without proving biological adequacy or intelligence.

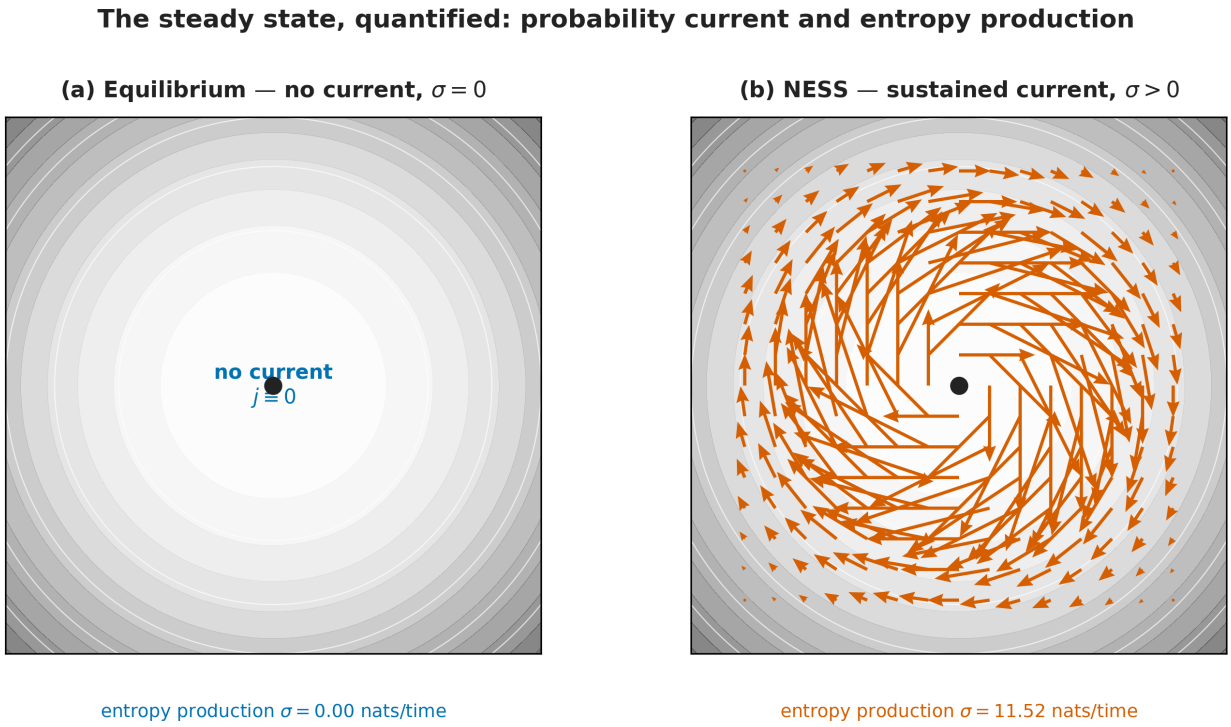


Figure 6: **Probability current and entropy production.** Both panels place the same stationary Gaussian density (grey contours) under the Bayesian-mechanics diffusion. **(a) Equilibrium** ($Q = 0$): the steady-state probability current is zero everywhere and the entropy-production rate is $\sigma = 0$ — detailed balance. **(b) NESS** ($Q \neq 0$): a divergence-free current $j = (QPx)\rho$ circulates around the mode (orange arrows), sustained at an entropy-production rate $\sigma = 2\omega^2\Pi/\gamma > 0$ (here $\sigma \approx 11.5$ nats/time). The current vanishes at the mode (where $QPx = 0$) and in the tails (where $\rho \to 0$), peaking on an annulus between. Rendered deterministically by `scripts/generate_steady_state_figure.py` over the tested core in `src/steady_state.py`.

10 Formal Reference and Verification

This paper supplies worked models and exposition. Its formal layer selects real-valued results from the Institute's fep_formal repository [Friedman, 2026a,b]. Lake imports its `FepSketches` Lean modules from the pinned revision in `formal/lakefile.lean`. Generic proofs remain upstream; Synthetic's Python cores do not import `fep_lean`.

10.1 Selected results and hypotheses

`ActiveInferenceFormal.Geometry` selects the flow and steady-state results; `ActiveInferenceFormal.Information` selects finite information, VFE, and EFE results. Their `ℝ_` theorem names correspond to the following claims:

- `ℝ_solenoidal_orthogonal` and `ℝ_flow_work_eq_dissipation`: the antisymmetric flow is orthogonal to the gradient, and only dissipation contributes work. Non-increasing work requires nonnegative dissipation.
- `ℝ_detailed_balance_iff`: zero solenoidal flow is equivalent to zero drive at a nonzero gradient. The gradient condition cannot be dropped.
- `ℝ_entropyProduction_nonneg` and `ℝ_entropyProduction_eq_zero_iff`: the sign and zero condition for the selected isotropic OU expression; the equivalence requires strictly positive precision and dissipation.
- `ℝ_finiteKL_nonneg`, `ℝ_finiteKL_self`, and `ℝ_finiteKL_eq_zero_iff`: finite information laws for the upstream normalized carriers and totalized functional.
- `ℝ_surprisal_le_VFE`, `ℝ_VFE_at_posterior`, and `ℝ_negative_VFE_le_logEvidence`: posterior-form VFE bounds and attainment for a finite generative model with positive predictive evidence.
- `ℝ_EFE_decomposition`, `ℝ_EFE_minimizer_exists`, and `ℝ_epistemicValue_nonneg`: the pragmatic/epistemic identity, finite-policy minimum, and information-gain sign for the stated upstream model.

The source signatures, rather than these summaries, own the full hypotheses. The upstream methods and formal-kernel guides describe the carriers, assumptions, and reviewed scope. The repository's `docs/positioning.md` provides their section-level navigation map.

10.2 Numerical and formal evidence

A Lean theorem checks its exact proposition under its hypotheses. A numerical test checks implementation behavior against its references and tolerances. Neither replaces the other. In particular, the Python KL helper rejects zero reference mass on positive belief support, while upstream finite KL has a real-valued totalization with different boundary behavior. Comparison requires a common support domain and a justified carrier match.

`Witnesses.lean` retains concrete controls: a permitted zero-work case, positive work when the nonnegative-dissipation hypothesis is violated, and the upstream disjoint-support totalized-KL boundary. `Audit.lean` prints the axiom footprint of every selected result and witness. The Python verification suite invokes the real build, checks the complete audit for `sorryAx`, and asks Lean to reject an explicit false real inequality.

10.3 Qualified blanket and planning adapters

`GaussianBlanket.lean` matches an additional canonical four-axis precision to the upstream native Gaussian carrier. It reuses positive definiteness, marginal endpoint covariance, conditional product rows, and endpoint conditional independence at every center. A supplied coupled endpoint perturbation has native non-independence. This covers the matched canonical example in sec. 8, not the separate original precision fixture or arbitrary Gaussian models.

`Planning.lean` uses Boolean observation histories as the outcome carrier. Its normalized terminal-history preference adds the explicit constant $\ln 2$ to the terminal-preference/history-information score. It proves predictive and posterior mass identities, a fair-coin terminal/cumulative mismatch, and a repeated-revealing-observation boundary. The Python implementation is compared numerically to that mathematical carrier in `src/planning_bridge.py`. Neither the identity nor the source-name correspondence certifies floating-point code or general observation-contingent policy-tree optimization.

Together these adapters make the paper's transfer boundaries constructive: state the carrier, bind an actual example, and test a condition that should break the conclusion. The closed-loop agent and hierarchical model remain independently tested numerical demonstrations. The paper does not establish empirical intelligence or universal formal coverage from these examples.

11 Thermodynamic Proof Imports and GNN Specifications

11.1 What is actually imported from FEP_Formal

The immutable `FepSketches` dependency remains revision `99276926` of `ActiveInferenceInstitute/fep_formal`. `Geometry.lean` imports `ness_flow`: solenoidal orthogonality, the dissipative work identity, a pointwise zero-drive equivalence away from a zero gradient, and the isotropic entropy-rate algebra. These are real-valued propositions. They do not constitute a proof of an SDE's invariant measure, a general detailed-balance theorem or a measured physical heat rate [Friedman, 2026a,b].

This revision also imports `FepSketches.path_thermodynamics` in `Thermodynamics.lean`. Its carrier is a pair of **normalized finite path laws** with an involutive reversal and a reverse law aligned to forward coordinates. Synthetic re-exports nonnegative path KL and, with strictly positive forward and reverse support, the integral fluctuation identity

$$\Sigma(\tau) = \ln \frac{P(\tau)}{P^{\mathrm{R}}(\tau)}, \qquad \mathbb{E}_P[\exp(-\Sigma)] = 1. \tag{6}$$

This identity follows from normalization and the supported likelihood ratio. Interpreting Σ as physical entropy production additionally requires the correct physical reverse protocol and local thermodynamic assumptions [Seifert, 2012]. The finite theorem is not asserted to be the continuum OU rate theorem: the path carrier, measure, support and time limit differ. Lean's totalized real logarithm at zero is not an infinite entropy production; the full-support hypotheses must remain visible. Upstream documentation and the paper's positioning guide distinguish these contracts.

11.2 A concrete GNN representation of both settings

Two versioned model inputs now accompany the paper: `data/gnn/ou_regimes.md` describes the matched-density continuous process, and `data/gnn/finite_agent.md` describes the finite cue/bet observation channels [Smékal and Friedman, 2023, Friedman, 2026c]. The continuous file declares dimension, precision, dissipation, the zero/driven alternatives, diffusion, density, current, time convention and equations. `Q_ou` denotes the antisymmetric drive; it must not be confused with the process-noise covariance called `Q` by some continuous-model interfaces. The finite file declares outcome-by-state likelihoods, prior, normalized preferences and their logarithmic conversion. It keeps the state static and does not introduce an adaptive payout.

GNN declaration	Synthetic carrier	Required interpretation
`x[2]`, `P_ou`, `Gamma_ou`, `Q_ou`, `D_ou`	`IsotropicOU` and stationary-current functions	Continuous even coordinates; $D = \Gamma$; $\gamma, \Pi > 0$ for diffusion
`A_cue`, `A_bet`, `D`	Action likelihood arrays and belief	Outcome rows, state columns; normalized columns and prior
`C_probs`, `C_log`	Preference law and expected log-preference	Positive normalized preference law; natural logs in nats
`Equations`, `Time`	Authored semantic specification	Syntax does not implement an SDE solver or an EFE backend

The optional `scripts/validate_gnn_specifications.py` invokes the upstream Markdown GNN parser, verifies exact demo parameters and selected dimensions, records parser source hashes and revision, and rejects altered dissipation or state dimension as real negative controls. The parser provenance is separate from the older cited GNN software snapshot. Run it in an environment with the GNN parser available, passing `--gnn-root` explicitly. The normal Synthetic Python package remains independent of that optional environment.

Parsing a specification, evaluating its declared equations, executing a target backend and establishing a theorem correspondence are four separate checks. This bridge establishes the first and its parameter/dimension correspondence to the tested examples. It does not claim that a generic GNN backend executes the OU diffusion or shares Synthetic's policy-scoring convention. A future execution bridge must compare matrices, time discretization, state parity, support, reversal and outputs before inheriting any conclusion.

12 Planning: Policies as Action Sequences

The one-step EFE example can be extended to a finite sequence of actions on a static hidden state. The local algorithm enumerates every observation branch, updates beliefs along that branch, and scores a declared terminal-preference objective. It does not optimize a general feedback policy. With $\pi = (a_1, \ldots, a_T)$, its branch law is

$$q(o_{1:T} \mid \pi) = \sum_x q(x) \prod_t p(o_t \mid x, a_t). \tag{7}$$

The score implemented in `src/policy_planning.py` is

$$G_{\text{terminal}}(\pi) = -\mathbb{E}[\ln C(o_T)] - I(x; o_{1:T} \mid \pi). \tag{8}$$

The score in eq. 8 uses the predictive history in eq. 7. Terminal pragmatic value is collected only at the last step. Epistemic value is the expected sequential posterior KL gain, checked independently against mutual information of the complete observation history. The two-step result selects cue-then-bet at 0.07, compared with 0.43 for bet-then-bet. Its information contribution is 0.37.

The bet channel here is state-independent. Observing the cue does not improve that fixed bet's reward distribution. Cue-then-bet wins through the objective's information term while retaining the preferred terminal outcome law. Genuine observation-adaptive commitment is demonstrated separately in sec. 13. Reading this open-loop score as an instrumental advantage or a closed-loop control theorem would exceed what the example computes.

12.1 An explicit upstream carrier bridge

For Boolean observations, define one history outcome (o_1, o_2) with product likelihood and normalized preference $C_H(o_1, o_2) = C(o_2)/2$. Upstream one-step EFE on that finite history carrier equals the local terminal/history objective plus the constant $\ln 2$. The computed normalization offset is 0.6931, and the cue-then-bet history-channel score is 0.7600. The offset does not change the ranking of the same policy set. `src/planning_bridge.py` tests the correspondence numerically; `formal/ActiveInferenceFormal/Planning.lean` proves the exact finite-carrier identity and predictive mass law.

This differs from repeatedly summing one-step EFE at every stage. A fair uninformative two-step model has terminal preference cost $\ln 2$ but cumulative preference cost $2 \ln 2$. A repeated perfect observation resolves the static state only once: history mutual information is $\ln 2$, not twice that value. The native controls preserve these distinctions. The Lean identity is about its explicit normalized carrier; it does not certify the floating-point branch implementation or general policy-tree optimization.

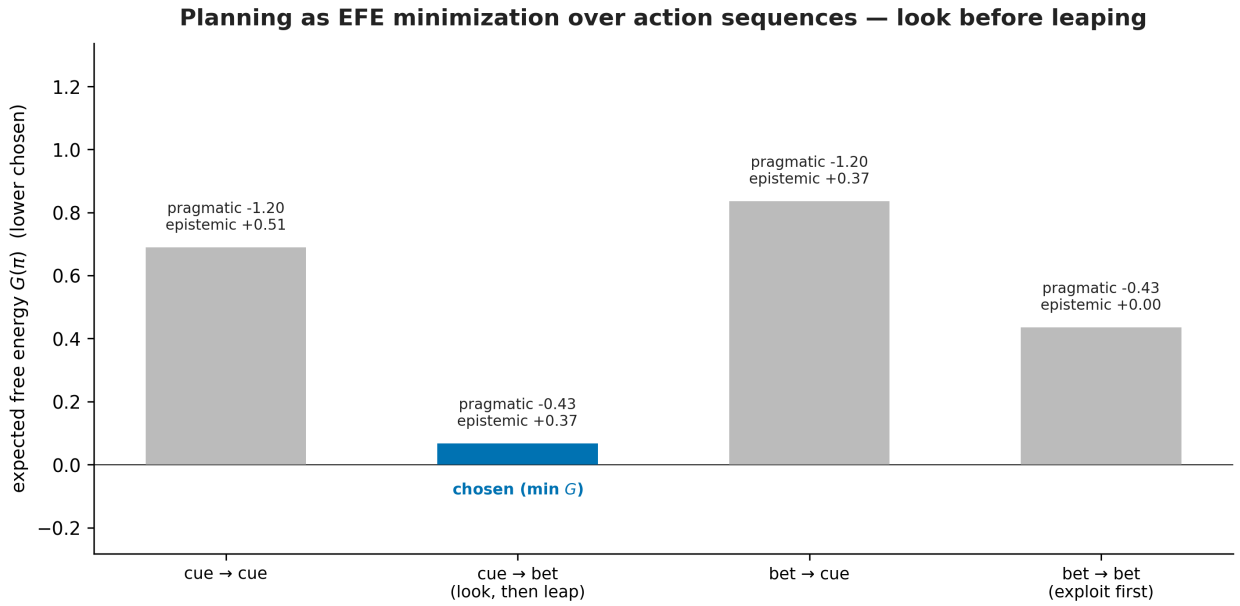


Figure 7: **Finite sequence scoring.** The four two-step sequences share a static hidden state. Cue-then-bet minimizes the declared terminal-preference minus history-information objective. The bet remains state-independent; the advantage is information scoring, not an observation-conditioned improvement in its payout.

13 A Closed-Loop Synthetic Agent

The previous examples separate inference and policy scoring. Here one finite agent connects them: predict observations, select an action by one-step EFE, receive a cue, update its belief by Bayes' rule, and choose a subsequent commitment. The hidden context is left or right and remains fixed during the episode. The outcome alphabet distinguishes cue-left, cue-right, reward, and loss. Action-conditioned likelihoods and positive outcome preferences are explicit in `src/synthetic_agent.py`.

The initial action set is cue, left, and right. The cue minimizes the initial EFE, with value 0.9163. A cue-right observation has predictive probability 0.5000 and gives posterior right-context mass 1.0000. The second-stage action set is explicitly restricted to left and right: information gathering has finished, and the agent must commit. Its right-action EFE is 0.5276, compared with 4.6052 for left. Cue-left gives the opposite decision. These are observation-contingent commitments, not a fixed action sequence.

The matched counterexample replaces the cue with a context-independent coin. An observed cue-right label then leaves right-context mass at 0.5000. The commitment scores tie, and the declared deterministic first-index tie rule selects left for either cue label. The label alone does not convey information: its state-dependent likelihood does. The tests check both real informative branches, the uninformative branch, impossible evidence, invalid indices, and normalization.

This is a deliberately small synthetic agent, with a supplied observation whose likelihood is checked against its environment model. The episode includes inference and an observation-sensitive next action. It does not learn its likelihood, discover its preferences, solve an arbitrary partially observed control problem, or establish empirical intelligence. EFE and policy support still depend on the stated model [Parr et al., 2022].

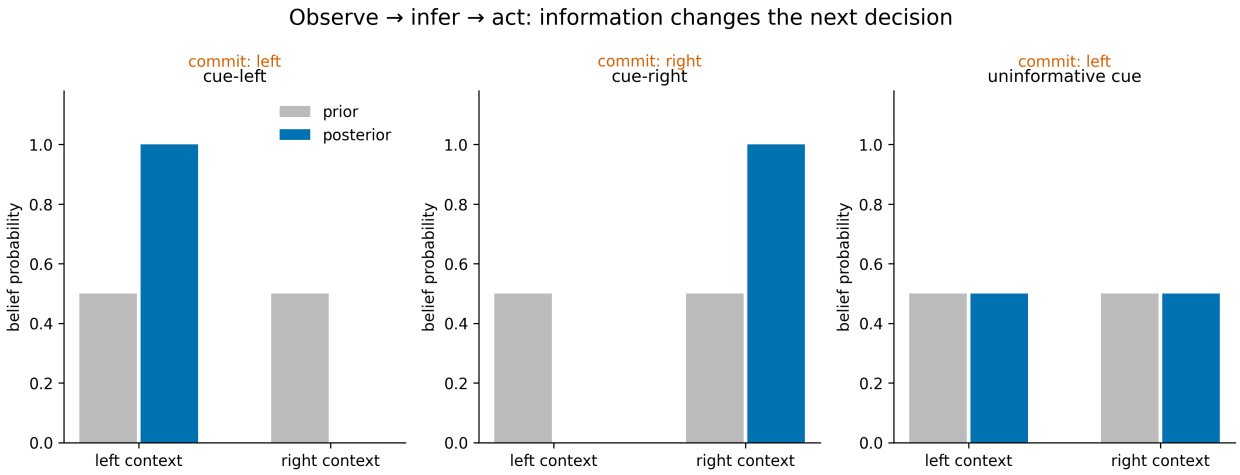


Figure 8: **Observation changes the next decision.** Informative cue-left and cue-right produce different posterior beliefs and opposite commitments. Replacing the cue with a state-independent coin leaves the prior unchanged and fails to produce an observation-sensitive commitment. Every panel comes from the same exact finite inference and action-selection core.

14 When Information Improves a Decision

Information gain and instrumental value answer different questions. Mutual information measures expected belief change in nats. Instrumental value asks whether an observation improves a subsequent choice under a specified utility. The open-loop bet in sec. 12 does not use hidden context. The closed-loop agent in sec. 13 does. We can isolate this difference with exact finite Bayesian decision analysis, without asserting that utility is EFE or thermodynamic free energy.

Let the static hidden state have prior $q(x)$, the cue channel be $A(o \mid x)$, and the available commitments have finite utilities $u(a, x)$. With no cue, the best expected utility is V_0. With a cue, a feedback policy can choose a different commitment for each reachable observation:

$$V_0 = \max_a \sum_x q(x)u(a,x), \qquad V_{\text{obs}} = \sum_o \max_a \sum_x q(x)A(o \mid x)u(a,x). \tag{9}$$

The joint-mass expression avoids defining a posterior on an impossible outcome. On reachable outcomes it equals expected optimal posterior utility. The policy that ignores the observation and always selects a prior-optimal action is among the admissible feedback policies. Therefore $V_{\text{obs}} \geq V_0$: for each outcome the maximum is at least the value of that constant action, and summing over outcomes recovers V_0 by channel normalization. This elementary finite argument is an exposition here, not a newly imported Lean theorem.

For a fixed acquisition cost c in the same utility units, define

$$\text{VOI} = V_{\text{obs}} - V_0, \qquad \text{VOI}_{\text{net}} = V_{\text{obs}} - V_0 - c. \tag{10}$$

The gross value is nonnegative under these assumptions; the net value can be negative. An optional acquisition is worthwhile under this utility objective only when its net value is positive. Unlike EFE's explicit epistemic reward, this calculation rewards information solely through its later consequences. Utility units, log-preference scores, and nats must not be silently substituted for one another. If utility is rescaled, its acquisition cost must be rescaled consistently.

The worked channel has known accuracy 0.9000 and a uniform prior. Correct commitment receives unit utility and incorrect commitment receives zero. Exact enumeration gives baseline utility 0.5000, informed utility 0.9000, and gross value 0.4000. With cost 0.1500, net value is 0.2500. The same cue carries 0.3681 nats of information when the utility matrix is replaced by state-independent outcomes, yet its gross instrumental value then vanishes. An independent cue gives both zero information and zero gross value.

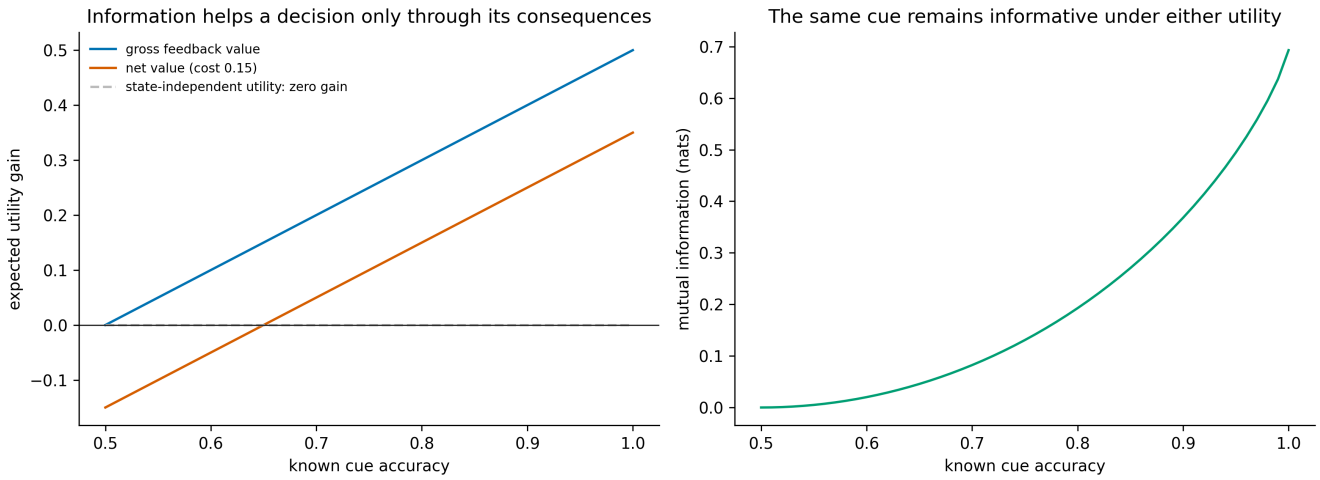


Figure 9: **Decision value and information have distinct units.** The left panel sweeps known cue accuracy for an observation-contingent commitment, its acquisition cost, and a state-independent utility counterexample. The right panel shows the same channel's mutual information. Positive information does not imply positive instrumental value; a beneficial cue can also be too costly. Both panels use exact finite computation, not sampled performance estimates.

`src/value_of_information.py` enumerates every reachable observation; tests independently enumerate every deterministic feedback policy. Unreachable outcomes have no invented posterior or action. Ties choose the lowest action index. The sweep holds the prior and utilities fixed and treats the channel as known. Miscalibrated models, changing hidden states, action-dependent sensing costs, limited computation, and learned preferences require additional models. This is a controlled bridge from learning to acting, not a claim of general adaptive intelligence.

Enumeration covers the complete finite policy space, but the Python arithmetic is floating point. Tests use independently calculated references and explicit tolerances; nonfinite inputs and unrepresentable utility results are rejected.

15 A Two-Level Generative Hierarchy

A hierarchy gives the model another hidden level without changing the logic of conditioning. Let context c generate state x, which generates observation o. The factorization is

$$p(c, x, o) = p(c)\, p(x \mid c)\, p(o \mid x). \tag{11}$$

Top-down prediction marginalizes the context law through the state channel. Bottom-up inference conditions the joint on the observed outcome, then marginalizes the posterior to each hidden level. This preserves the joint rather than updating the levels independently. The exact categorical core is `src/hierarchical_model.py`; matrix columns are conditional distributions.

For the worked observation, predictive state-zero mass is 0.5600 and observation evidence is 0.6360. The posterior state-zero mass becomes 0.7925 and the posterior context-zero mass becomes 0.7358. The observation therefore updates both levels while their posterior dependence remains represented in the joint. fig. 10 shows the downward prediction and upward update.

A real control makes the observation channel independent of state. Conditioning then leaves both hidden marginals unchanged. An impossible observation raises an error instead of manufacturing a posterior. Tests compare the full joint and each posterior with independently enumerated products, and reject malformed channels and out-of-range observations.

This is exact inference in a finite Bayesian hierarchy, not a neural network, a variational message-passing algorithm, or a learned deep model. Its purpose is to show how a higher-level model supplies a lower-level prior and how observations update that higher level through the shared generative structure [Parr et al., 2022].

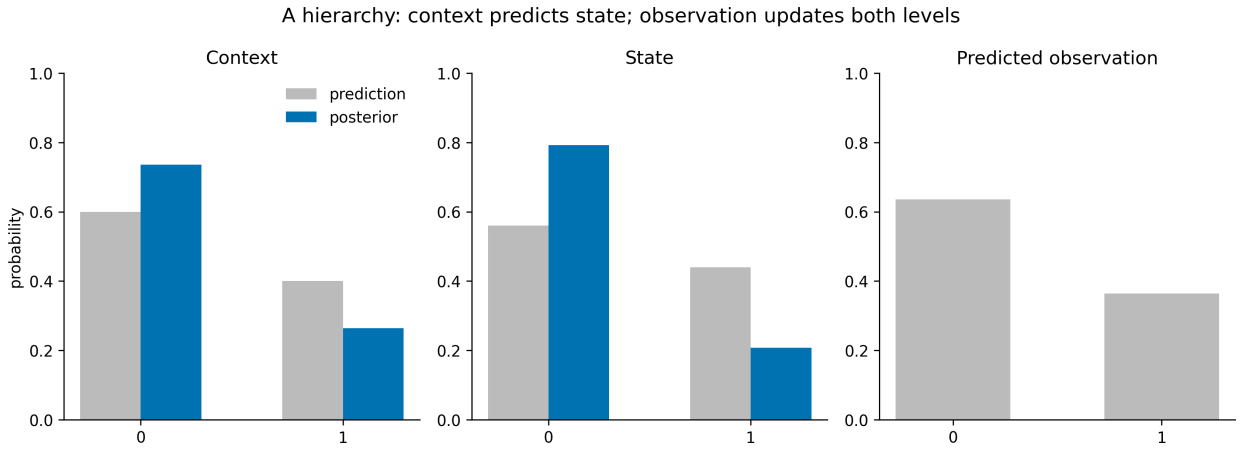


Figure 10: **Prediction downward, inference upward.** The context law induces a state prediction and an observation prediction. Conditioning on one observation updates both context and state using the same posterior joint, without discarding their dependence.

16 Discussion: Informational and Thermodynamic Interpretation

The demonstrations connect several mathematical settings without identifying their free energies merely by name. Variational free energy is a functional of a recognition distribution and a generative model; with natural logarithms it is measured in nats. Expected free energy scores predicted outcomes and states under a policy, with pragmatic and epistemic terms in the same informational units. Thermodynamic free energy is an energy quantity, measured in joules. A relation between these objects requires a stated physical model and scaling.

16.1 Equilibrium and stationary circulation

The OU example fixes a Gaussian stationary density and a matched-noise fluctuation-dissipation convention. Zero solenoidal drive gives detailed balance. Nonzero drive gives a divergence-free stationary probability current and positive entropy production for positive precision and dissipation. The deterministic drift trajectory still relaxes; it is not a perpetual orbit. Equilibrium's zero probability current also does not imply zero microscopic stochastic motion [Da Costa et al., 2021, Friston et al., 2023].

The Gaussian landscape used in the drift is dimensionless. Under an explicit Gibbs model it can be related to an energy landscape through $F(x) = U(x)/(k_{\mathrm{B}}T) + \text{constant}$, with temperature in kelvin. For an isothermal physical interpretation using the corresponding local thermodynamic assumptions, an informational entropy-production rate σ can be scaled to $k_{\mathrm{B}}\sigma$ and dissipated power to $k_{\mathrm{B}}T\sigma$. The code does not estimate temperature, friction, physical work, or metabolic expenditure from data. The conversion is conditional on a physical interpretation, not an empirical result of the numerical figure.

16.2 Statistical boundaries and synthetic agents

A Gaussian precision zero is a conditional-independence statement under the Gaussian law. It does not by itself specify causal channels, action effects, model learning, or agency. The finite closed-loop example supplies a separate explicit perception-action carrier, while the hierarchy supplies a separate multi-level inference carrier. A shared vocabulary is a guide to comparison; transporting a theorem requires compatible carriers and hypotheses.

Bayesian mechanics provides a framework for relating probabilistic dynamics, statistical boundaries, and belief interpretation [Ramstead et al., 2023]. The Free Energy Principle supplies the broader organizing perspective [Friston, 2010]. These worked examples introduce that perspective while keeping its assumptions visible. They do not establish that every persistent current is intelligent, every conditional-independence partition is an agent, or every informational objective is a measured physical free energy.

The intended next use is reproducible exploration: change a channel, a preference, a precision entry, or a drive parameter; rerun the tests and figures; and inspect which conclusion survives. For deeper mathematics, the pinned `fep_formal` methods and formal-kernel documentation provide the declarations, support conventions, and scope reviews used by this paper's exposition.

16.3 Reading and reusing the Institute repositories

Synthetic supplies the worked teaching models and paper. The Institute's fep_formal provides the pinned mathematical reference used in sec. 10 [Friedman, 2026a,b]. Its `FepSketches` Lean interface is the actual formal dependency; `fep_lean` and `fep-lean` are upstream Python and command-line interfaces for readers who want to inspect the broader catalogue.

Generalized Notation Notation (GNN) is a complementary language and toolchain for specifying generative models [Smékal and Friedman, 2023, Friedman, 2026c]. It is relevant when moving these explicit priors, likelihoods, preferences, and state variables into a shareable model specification. The software citation identifies the reviewed GNN source snapshot. Synthetic constructs its runtime arrays directly. The optional specification bridge now parses two authored GNN files with the upstream parser and checks their parameters and dimensions against those arrays and the OU model (sec. 11). It does not execute a GNN simulation backend. Execution would require further mapping of channel orientation, observations, preferences, time stepping and control assumptions, with output equivalence and boundary tests. A GNN representation does not itself certify an upstream Lean theorem or a thermodynamic interpretation.

The numerical distribution installs as `active_inference_synthetic`; checkout scripts retain their local `src` imports. Readers can reproduce the finite models independently of either upstream Python package. The repository's

positioning guide separates mathematical reuse, model specification, and template-based manuscript rendering, with revision-bound links and citation metadata for each upstream project.

17 A Reader's Route Beyond Synthetic

Reproduce a worked model first, then choose the resource that answers the next question. Mathematical proof, shared vocabulary, model specification, software implementation, and educational discussion are different forms of support. Links below are navigation resources; they do not add runtime dependencies or extend the formal guarantees of the selected Synthetic carriers.

Question	Resource	How to use it
Which mathematical hypotheses are required?	FEP_Formal	Inspect exact declarations and assumption reviews; Synthetic uses its pinned `FepSketches` Lean interface.
How can a model be written as a shareable specification?	Generalized Notation Notation	Compare state dimensions, channel orientation, preferences and control assumptions before implementing a mapping.
What does a term mean in the shared vocabulary?	Active Inference Ontology	Read definitions and typed relations; a vocabulary entry is not a theorem or empirical validation.
Where can I study the foundations systematically?	fundamentals and courses	Use reading-group companion materials and structured curricula; the fundamentals repository describes itself as unofficial textbook support.
Where can I explore a broader executable agent implementation?	cognitive	Inspect the discrete/continuous model interfaces and their validation commands; compare conventions before reusing code.
Where can I find talks and their transcripts?	ActiveInferenceJournal	Navigate the structured video/transcript archive and check the original recording and cited papers.

The bibliography distinguishes FEP_Formal and GNN software snapshots from their scholarly papers [Friedman, 2026a,b,c, Smékal and Friedman, 2023]. The additional learning resources are maintained independently. Repository availability or a passing software test is not evidence that a particular explanation is scientifically correct.

For a constructive next experiment, change one cue channel, utility, preference, precision entry or drive parameter. Predict the result before computing it, include a control that should falsify the intended conclusion, and compare the new result with the analytical reference. If the modification crosses into a different mathematical carrier, document the transfer assumptions before citing an upstream theorem. The repository's resource and methods guides provide direct links, runnable examples, and the limits of each execution surface.

The ontology and transcript archive have their own software/dataset citation records [Active Inference Institute, 2023, Active Inference Institute]. Their citation metadata is independent of the mathematical and scholarly sources used for the worked models.

References

Active Inference Institute. ActiveInferenceJournal. Video-library transcript and metadata dataset. URL https://github.com/ActiveInferenceInstitute/ActiveInferenceJournal. Dataset DOI: https://doi.org/10.5281/zenodo.21074926. Dataset identity from CITATION.cff; accessed 2026-10-11.

Active Inference Institute. Active inference ontology. Controlled vocabulary repository, 2023. URL https://github.com/ActiveInferenceInstitute/Active_Inference_Ontology. Release DOI: https://doi.org/10.5281/zenodo.7972289. Citation metadata from CITATION.cff, version v5, released 2023-05-25; accessed 2026-10-11.

Lancelot Da Costa and Grigorios A. Pavliotis. The entropy production of stationary diffusions. Journal of Physics A: Mathematical and Theoretical; author manuscript arXiv:2212.05125v3, 2023. URL https://arxiv.org/html/2212.05125v3.

Lancelot Da Costa, Karl Friston, Conor Heins, and Grigorios A. Pavliotis. Bayesian mechanics for stationary processes. *Proceedings of the Royal Society A*, 477(2256):20210518, 2021. doi: 10.1098/rspa.2021.0518.

Daniel Ari Friedman. fep_formal: Towards Lean 4 formalization of the Free Energy Principle. Active Inference Institute software repository, 2026a. URL https://github.com/ActiveInferenceInstitute/fep_formal/tree/99276926da1df4287a2e13e9aa7f6d1822cfdca3. Source revision 99276926; citation metadata from the pinned CITATION.cff, accessed 2026-10-11.

Daniel Ari Friedman. Towards Lean 4 formalization of the Free Energy Principle: AI-driven theorem sketching and verification for Active Inference and Bayesian Mechanics. Scholarly paper, Active Inference Journal, 2026b. URL https://doi.org/10.5281/zenodo.23220027.

Daniel Ari Friedman. Generalized Notation Notation (GNN). Active Inference Institute software repository, 2026c. URL https://github.com/ActiveInferenceInstitute/Generalized_Notation_Notation/tree/9255e19f951d4c9c1461514464d420bd3fc8b912. Source revision 9255e19f; project concept DOI, not a version-specific archive; citation metadata accessed 2026-10-11.

Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010. doi: 10.1038/nrn2787.

Karl Friston, Lancelot Da Costa, Noor Sajid, Conor Heins, Kai Ueltzhöffer, Grigorios A. Pavliotis, and Thomas Parr. The free energy principle made simpler but not too simple. *Physics Reports*, 1024:1–29, 2023. doi: 10.1016/j.physrep.2023.07.001.

Thomas Parr, Giovanni Pezzulo, and Karl J. Friston. *Active Inference: The Free Energy Principle in Mind, Brain, and Behavior*. MIT Press, Cambridge, MA, USA, 2022. ISBN 978-0-262-04535-3. doi: 10.7551/mitpress/12441.001.0001.

Maxwell J. D. Ramstead, Dalton A. R. Sakthivadivel, Conor Heins, Magnus Koudahl, Beren Millidge, Lancelot Da Costa, Brennan Klein, and Karl J. Friston. On bayesian mechanics: a physics of and by beliefs. *Interface Focus*, 13(3):20220029, 2023. doi: 10.1098/rsfs.2022.0029.

Udo Seifert. Stochastic thermodynamics, fluctuation theorems and molecular machines. *Reports on Progress in Physics*, 75:126001, 2012. doi: 10.1088/0034-4885/75/12/126001. URL https://arxiv.org/abs/1205.4176.

Jakub Smékal and Daniel Ari Friedman. Generalized notation notation for Active Inference models. Scholarly paper, Active Inference Journal, 2023. URL https://zenodo.org/records/7803328.