Mathematical & Statistical Underpinnings

GP-Cognition Optimal Foraging for Archaeological Agent-Based Models

This document formalizes the model proposed in the position paper. It is written to be both mathematically explicit and implementation-ready.

---

1) High-level structure

We distinguish three coupled objects:

1. **Truth landscape** (latent, possibly time-varying):

2. A scalar field over space, e.g. resource payoff, suitability, safety-adjusted return.

3. **Belief landscape** (agent-specific, probabilistic):

4. A posterior distribution over the truth landscape, represented as a Gaussian Process (GP).

5. **Decision rule** (OFT-inspired, uncertainty-aware):

6. A utility maximization over feasible moves that trades off expected payoff vs uncertainty (exploration).

---

2) Notation

2.1 Space and time

- Discrete time steps: $t = 0, 1, \ldots, T$.
- Discrete spatial grid (raster) of size $N \times N$.
- Let s denote a cell location, typically $s = (i, j)$ with $i, j \in \{0, \ldots, N-1\}$.
- Let $\mathcal{N}(s)$ denote the neighbor set of s (Von Neumann or Moore neighborhood).

2.2 Agent state

For an agent a: - Position: s_t^a. - Observation set (bounded memory): $\mathcal{D}_t^a = \{(x_n, y_n)\}_{n=1}^{m_t}$, where $m_t \leq k$ if using a memory window of size k. - Belief model: posterior GP $f_t^a(\cdot)$ with mean $\mu_t^a(\cdot)$ and variance $(\sigma_t^a(\cdot))^2$.

2.3 Landscape components

- Truth/payoff surface: $Z_t(s)$.
- Optional movement cost: $C(s)$ (static or dynamic).
- Optional risk surface: $R(s)$ (static or dynamic).

---

3) Truth landscape model (environment)

3.1 Stationary truth (MVP equilibrium)

A stationary resource landscape is a fixed field:

$$Z_t(s) \equiv Z(s).$$

3.2 Dynamic truth (disequilibrium)

A minimal feedback form (depletion/regeneration):

$$Z_{t+1}(s) = \underbrace{Z_t(s)\,(1 - \delta\,\mathbb{I}[s \in V_t])}_{\text{local depletion}} + \underbrace{\gamma\,(1 - Z_t(s))}_{\text{regeneration}},$$

where: - $\delta \in [0, 1]$ is depletion rate, - $\gamma \in [0, 1]$ is regeneration rate, - V_t is the set of cells visited (or exploited) at time t.

This is deliberately simple but sufficient to create **belief–world mismatch** (belief lag) when Z_t changes faster than beliefs update.

---

4) Observation model (what an agent learns)

At time t, agent a at position s_t^a observes a noisy signal:

$$Y_t^a = Z_t(s_t^a) + \varepsilon_t^a,$$

where $\varepsilon_t^a \sim \mathcal{N}(0, \sigma_{\text{obs}}^2)$.

4.1 Optional observation constraints (extensions)

To incorporate visibility/knowledge constraints, modify either:

1) **Where** observations are available (observation operator):

$$\text{Agent observes } \{Z_t(s) : s \in \mathcal{O}(s_t^a)\},$$

where $\mathcal{O}(s)$ could be a radius, viewshed, or line-of-sight set.

2) **How reliable** observations are (heteroskedastic noise):

$$\varepsilon_t^a \sim \mathcal{N}(0, \sigma_{\mathrm{obs}}^2(s_t^a)).$$

For the MVP, we use local observation at the current cell with constant noise.

---

5) Belief model: Gaussian Process regression over space

We model a latent function $f(s)$ representing the agent's internal map of value. The GP prior is:

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)).$$

Typically, $m(s) = m_0$ is constant (e.g., 0.5 after normalization) or 0.

5.1 Input encoding

Because $s = (i, j)$ is a grid index, we map to continuous coordinates:

$$x = \big(i/(N-1),\ j/(N-1)\big) \in [0, 1]^2.$$

Denote this mapping by $x = \phi(s)$.

5.2 Kernel choice

A standard stationary kernel for spatial generalization:

$$k(x, x') = \sigma_f^2 \, \exp\Big(-\tfrac{1}{2\ell^2} \|x - x'\|^2 \Big).$$

Interpretation: - ℓ: cognitive generalization scale (how far knowledge transfers) - σ_f^2: prior variance (initial uncertainty about values)

5.3 Observation likelihood

With Gaussian observation noise:

$$y_n = f(x_n) + \eta_n, \quad \eta_n \sim \mathcal{N}(0, \sigma_n^2).$$

For constant noise, $\sigma_n^2 = \sigma_{\mathrm{obs}}^2$.

---

6) GP posterior: mean and uncertainty

Given training inputs $X = [x_1, \ldots, x_m]^\top$ and targets $y = [y_1, \ldots, y_m]^\top$, define: - Kernel matrix: $K = K(X, X)$ with $K_{ij} = k(x_i, x_j)$ - Cross-kernel: $k_* = K(X, x_*)$ - Prior variance at test: $k(x_*, x_*)$

Posterior predictive distribution at x_* is Gaussian:

$$f(x_*) \mid X, y \sim \mathcal{N}(\mu(x_*), \ \sigma^2(x_*)).$$

6.1 Posterior mean

$$\mu(x_*) = m(x_*) + k_*^\top \left(K + \sigma_{\text{obs}}^2 I\right)^{-1}(y - m(X)).$$

6.2 Posterior variance

$$\sigma^2(x_*) = k(x_*, x_*) - k_*^\top \left(K + \sigma_{\text{obs}}^2 I\right)^{-1} k_*.$$

This $\sigma(x_*)$ is the agent's **cognitive uncertainty** at x_*.

---

7) Computational approximations (bounded cognition and scalability)

Exact GP inference scales as $\mathcal{O}(m^3)$. Archaeological ABMs require bounded cognition, so approximation is both practical and theoretically defensible.

7.1 Sliding memory window (local/recency-limited belief)

Maintain only the most recent k observations:

$$\mathcal{D}_t^a = \{(x_n, y_n)\}_{n=m_t-k+1}^{m_t}.$$

This models bounded memory and ensures inference cost remains bounded.

7.2 Local neighborhood GP

Maintain observations within a radius r of current position:

$$\mathcal{D}_t^a(r) = \{(x_n, y_n) : \|x_n - \phi(s_t^a)\| \le r\}.$$

This is appropriate when spatial correlation decays with distance.

7.3 Sparse inducing point GP (high-performance variant)

Choose inducing locations $U = [u_1, \ldots, u_M]$ (e.g., a coarse grid). Approximate inference uses $M \ll m$. This supports larger runs and multi-agent scenarios.

7.4 Amortized updates

Belief updates need not happen every step. Update the GP every K steps (bounded attention), while decisions still occur every step using the last posterior.

8) Decision rule: OFT with an uncertainty term

Let the agent consider candidate moves $s \in \mathcal{N}(s_t^a)$. Define a utility score:

$$U_t^a(s) = \underbrace{\mu_t^a(\phi(s))}_{\text{expected return}} + \underbrace{\beta\,\sigma_t^a(\phi(s))}_{\text{exploration bonus}} - \underbrace{\lambda\,C(s)}_{\text{movement cost}} - \underbrace{\rho\,R(s)}_{\text{risk penalty}}.$$

Parameters: - $\beta \geq 0$: exploration weight (curiosity; institutional tolerance for uncertainty) - $\lambda \geq 0$: sensitivity to travel cost - $\rho \geq 0$: risk aversion

Action selection:

$$s_{t+1}^a = \arg \max_{s \in \mathcal{N}(s_t^a)} U_t^a(s).$$

Tie-breaking can be random.

8.1 Interpretation

- When $\beta = 0$, the model collapses toward exploitation (classic OFT on belief means).
- When $\beta > 0$, uncertainty becomes a driver of movement (exploration as motivated behavior).

8.2 Alternatives (optional variants)

- **Thompson sampling**: sample $\tilde{f} \sim \mathcal{GP}$ posterior and choose $\arg\max \tilde{f}(s)$ over candidates.
- **Information gain**: choose moves maximizing expected reduction in posterior entropy.

The MVP uses GP-UCB for simplicity and interpretability.

---

9) Step-by-step: one complete "turn" for an agent

Below is a concrete per-step procedure for a single agent. Multi-agent versions run the same logic per agent, either synchronously or asynchronously.

Inputs at time t

- Current position s_t
- Current belief state (μ_t, σ_t) (from previous update)
- Observation history $\mathcal{D}_t$
- Environment state Z_t (unknown to agent), optional C, R

Turn sequence

Step 1 — Observe (sample the world locally) 1. Compute true payoff at current cell: $Z_t(s_t)$. 2. Draw noisy observation:

$$y_t = Z_t(s_t) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma_{\mathrm{obs}}^2).$$

3. Append to memory:

$$x_t = \phi(s_t), \quad \mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_t, y_t)\}.$$

4. If using bounded memory, drop oldest points so $|\mathcal{D}_{t+1}| \leq k$.

Step 2 — Update belief (GP posterior refresh) 5. If $|\mathcal{D}_{t+1}| < m_{\min}$ (e.g., 3), either: - keep a prior belief, or - fall back to a baseline policy. 6. Otherwise, fit/update GP using $\mathcal{D}_{t+1}$ to obtain $\mu_{t+1}(\cdot)$ and $\sigma_{t+1}(\cdot)$. - With amortization, this step occurs only when $t \bmod K = 0$; otherwise $(\mu_{t+1}, \sigma_{t+1}) = (\mu_t, \sigma_t)$.

Step 3 — Evaluate candidate moves 7. Enumerate candidate neighbor cells:

$$\mathcal{S}_t = \mathcal{N}(s_t).$$

8. For each $s \in \mathcal{S}_t$, compute posterior predictions:

$$\mu_s = \mu_{t+1}(\phi(s)), \quad \sigma_s = \sigma_{t+1}(\phi(s)).$$

9. Compute utility score:

$$U(s) = \mu_s + \beta\sigma_s - \lambda C(s) - \rho R(s).$$

Step 4 — Choose and move 10. Select:

$$s_{t+1} = \arg\max_{s \in \mathcal{S}_t} U(s).$$

11. Move to s_{t+1}.

Step 5 — World update (optional dynamic equilibrium) 12. If depletion/regeneration is enabled, update Z_{t+1} according to the ecological dynamics (Section 3.2). In the simplest version, depletion is applied at the current or visited cell.

Step 6 — Record outputs 13. Record for analysis: - trajectory s_{t+1} - observed value y_t - (for diagnostics only) true value $Z_t(s_t)$ - uncertainty at chosen step $\sigma_{t+1}(\phi(s_{t+1}))$ - reward definition (e.g., $Z_t(s_t) - \lambda C(s_t) - \rho R(s_t)$)

This completes one turn.

---

10) Multi-agent extension (brief)

With multiple agents $a = 1, \ldots, A$, the same turn sequence applies. Two major design choices are:

1) **Beliefs are private**: each agent maintains $\mathcal{D}_t^a$ and a personal GP. 2) **Beliefs are shared**: agents exchange observations or fuse posteriors. A simple fusion mechanism is to share observations (data-level pooling) within social networks.

Synchronous scheduling: - all agents observe - all beliefs update - all agents move - world updates

Asynchronous scheduling: - agents take turns sequentially, updating the world after each move.

---

11) Key derived quantities (for analysis and falsification)

11.1 Uncertainty collapse

Track $\sigma_t(\phi(s_t))$ over time. In stationary worlds, it should decline in visited regions.

11.2 Belief–world mismatch (disequilibrium)

When truth is dynamic, track:

$$\Delta_t(s) = \mu_t(\phi(s)) - Z_t(s).$$

Large systematic $|\Delta_t|$ in exploited regions is an operational measure of disequilibrium.

11.3 Exploration–exploitation balance

A simple operational definition: - exploitation-dominated steps: chosen s has high μ relative to neighborhood - exploration-dominated steps: chosen s has high σ relative to neighborhood

---

12) Parameter interpretation (archaeological mapping)

- ℓ (kernel length-scale): how far experiences generalize; can represent cultural knowledge transfer or landscape legibility.
- β (exploration weight): curiosity, risk tolerance, innovation pressure, frontier expansion propensity.
- k (memory window): bounded memory / attention.
- σ_{obs}: perceptual noise, cue reliability, or environmental stochasticity.
- δ, γ: depletion and regeneration; minimal ecological feedback producing dynamic equilibrium.

---

13) Minimal pseudocode (single agent)

```
initialize landscape Z_0, cost C, risk R
initialize agent position s_0
initialize belief prior GP
D = empty

for t in 0..T-1:
  # observe
  y_t = Z_t(s_t) + Normal(0, sigma_obs)
```

```
   D.append( (phi(s_t), y_t) )
   D = keep_last_k(D)

   # belief update (optional amortization)
   if |D| >= m_min and (t mod K == 0):
       fit GP posterior using D

   # evaluate neighbors
   candidates = neighbors(s_t)
   for s in candidates:
       mu_s, sigma_s = GP.predict(phi(s))
       score_s = mu_s + beta*sigma_s - lambda*C(s) - rho*R(s)

   # move
   s_{t+1} = argmax(score)

   # world update (optional)
   Z_{t+1} = update(Z_t, visited={s_t})

   record metrics
```

14) Notes on implementation fidelity

- Standardize/normalize Z to [0,1] for stable GP priors.
- Avoid hyperparameter optimization at every step; treat kernel parameters as interpretable cognitive priors.
- Use bounded memory or sparse GP for tractability; these approximations align with bounded rationality.

End of document.