# 1 Original Mathematical Equations

Based on the excerpts from the original paper, below is a list of each mathematical equation, definition, or formula explicitly mentioned in the text and supplements describing the simulation framework.

## 1.1 Definitions and Calculated Probabilities

These formulas define the probabilities of success for individual and social learning, as well as equilibrium population fitness in the baseline model:

| Description | Equation |
|---|---|
| Individual Learning Success Probability (Probability of becoming adapted when individual learning) | $\mathbf{p_i^{OK} := (1 - c_I)z_i}$ |
| Calculated Value (Using $c_I = 0.05$ and $z_i = 0.66$) | $0.95 \times 0.66 = 0.627$ |
| Baseline Equilibrium Fitness (Long-run equilibrium fitness when only individual learners exist) | $\mathbf{E[q^{OK}] = p_i^{OK} s_{OK}}$ |
| Social Learning Success Probability (Conditional probability of success when learning from an adapted agent, assuming no environment change $u$) | $p_s^{OK \to OK} := P(\text{unchanged})P(\text{copied successfully}) = (1 - u) \times 1 = 0.99$ |
| Social Learning Adaptation Probability (Probability of becoming adapted when social learning) | $\mathbf{p_s^{OK} := (1 - c_S)q^{OK} p_s^{OK \to OK}}$ |
| AI Social Learning Adaptation Probability (The AI's adaptation level is set to the mean adaptation status of the population) | $\mathbf{p_{AI}^{OK} := q^{OK}}$ |
| Critical Social Learning Success Probability (When an agent decides to switch to individual learning if social learning fails) | $\mathbf{p_{cs}^{OK} = 1 - (1 - p_s^{OK})(1 - p_i^{OK})}$ |
| AI Individual Learning Success Probability (The probability that the AI successfully learns individually, where $c_{\lambda i}$ is the cost and $z_{AI}$ is the success rate) | $\mathbf{p_{x \to OK}^{AI} := (1 - c_{\lambda i})z_{AI}}$ |

Table 1: Summary of Learning Success Probabilities and Fitness Equations.

## 1.2 Numbered Equations from the Main Text and Supplement

The paper includes formal derivations of the expected fitness measures, presented both in the main body (Equations 1, 2, 3) and reiterated/derived in the supplement (Equations 4, 5).

### 1.2.1 Expected Mean Fitness Equations (Original Rogers' Paradox)

These equations describe the expected fitness measures for a network containing only individual and social learners.

- **Equation (1) / Equation (4): Expected mean fitness across all agents** → Presented as Equation (1) in Section 2.1 and derived as Equation (4) in Supplement 6)

$$E[q^{OK}] = \frac{p_i^{OK} s_{OK} E[q_i]}{1 - (1 - c_S)p^{OK \to OK} ss_{OK} E[1 - q_i]} \quad (1, 4)$$

  - Intermediate steps:

$$\begin{aligned} E[q^{OK}] &= p_i^{OK} s_{OK} E[q_i] + E[p_s^{OK}] s_{OK} E[q_s] \\ &= p_i^{OK} s_{OK} E[q_i] + (1 - c_S)p^{OK \to OK} ss_{OK} E[q^{OK}] E[1 - q_i] \end{aligned}$$

- **Equation (2) / Equation (5): Expected mean fitness of just social learners** → Presented as Equation (2) in Section 2.1 and derived as Equation (5) in Supplement 6

$$E[q_s^{OK}] = \frac{(1 - c_S)p^{OK \to OK} ss_{OK} p_i^{OK} s_{OK} E[q_i]}{1 - (1 - c_S)p^{OK \to OK} ss_{OK} E[1 - q_i]} \quad (2, 5)$$

  - Intermediate steps:

$$\begin{aligned} E[q_s^{OK}] &= E[p_s^{OK}] s_{OK} \\ &= (1 - c_S)p^{OK \to OK} ss_{OK} E[q^{OK}] \end{aligned}$$

### 1.2.2 Expected Mean Fitness Equation (AI Rogers' Paradox)

This equation describes the expected mean fitness when individual learners, social learners (from humans), and AI social learners are present:

$$E[q^{OK}] = p_i^{OK} s_{OK} E[q_i] + E[p_s^{OK}] s_{OK} E[q_s] + E[p_{AI}^{OK}] s_{OK} E[q_{AI}] \quad (3)$$

## 1.3 Equations for Model-Centric Strategies

These formulas define the mechanics of how the AI system updates its knowledge and how individual learning is impacted by negative feedback.

### 1.3.1 AI Update Schedule Probabilities

(Presented in Section 3.2.1, describing the expected rate of AI engaging in social learning where $c_{\lambda s}$ is the cost of updating socially.)

$$p_{x \to q^{OK}}^{AI} := 1 - c_{\lambda s}$$
$$p_{x \to x}^{AI} := c_{\lambda s}$$

### 1.3.2 AI Adaptation Level Update (Social Learning Cadence)

Describes the AI's adaptation level in the next timestep $t + 1$:

$$p_{AI}^{OK,(t+1)} = 1 - (1 - p_{x \to q^{OK}}^{AI} q^{OK})(1 - p_{AI}^{OK,t} p_{x \to x}^{AI} (1 - u))$$

## 1.4 Equations for Negative Feedback (Deskilling)

These equations define the individual learning penalty ($\kappa$) applied to human agents who learn socially from the AI system (Section 4):

$$\kappa_j^0 = 1$$

$$\kappa_j^{t+1} = 0.9 \kappa_j^t$$

$$p_{ij}^{OK} = (1 - c_i) z_i \kappa_j$$

## 1.5 Step-by-Step Mathematical Explanations

The equations presented in the original paper define the core probabilities, fitness measures, and strategic rules governing agent behavior within the dynamic simulation environment. Below is a step-by-step explanation of each mathematical equation, formula, and definition.

| Equation | Meaning and Step-by-Step Explanation |
| --- | --- |
| **1. Individual Learning and Baseline Adaptation** | |
| $\mathbf{p_i^{OK} := (1 - c_I) z_i}$ | **Individual Learning Success Probability:** Defines the probability ($p_i^{OK}$) that an agent becomes adapted when attempting individual learning. $c_I$ is the cost of individual learning (e.g., $c_I = 0.05$). $(1 - c_I)$ represents the likelihood of a successful attempt, and $z_i$ is the intrinsic success rate (e.g., $z_i = 0.66$). The product gives the overall adaptation probability via IL. |
| $\mathbf{E[q^{OK}] = p_i^{OK} s_{OK}}$ | **Baseline Equilibrium Fitness:** Gives the long-run expected mean fitness ($E[q^{OK}]$) of a pure individual-learning population. It is the product of individual learning success ($p_i^{OK}$) and the survival probability for adapted agents ($s_{OK}$), e.g., $s_{OK} = 0.93$. |
| $0.95 \times 0.66 = 0.627$ | **Calculated Value for $p_i^{OK}$:** Numerical result using $c_I = 0.05$ and $z_i = 0.66$, showing that the individual learning success probability equals 0.627. |
| **2. Social Learning Probabilities** | |

| | |
|---|---|
| $p_s^{OK\to OK} := P(\text{unchanged})P(\text{copied successfully}) = (1-u)\times 1 = 0.99$ | **Social Learning Success Probability (Conditional):** Probability of success when learning from an already adapted agent. Requires (1) the environment remains unchanged ($1-u$, where $u = 0.01$), and (2) successful copying (1). |
| $\mathbf{p_s^{OK} := (1 - c_S)q^{OK}p_s^{OK\to OK}}$ | **Social Learning Adaptation Probability (Human-Human):** Defines the probability of becoming adapted through social learning. $c_S$ is the cost (often 0), $q^{OK}$ is the fraction of adapted agents, and $p_s^{OK\to OK}$ is the conditional copying success. |
| $\mathbf{p_{AI}^{OK} := q^{OK}}$ | **AI Social Learning Adaptation Probability:** The AI's adaptation level equals the mean adaptation status of the population from the previous timestep ($q^{OK}$). |
| **3. Expected Mean Fitness Equations (Rogers' Paradox Derivations)** | |
| $\mathbf{E[q^{OK}] = \dfrac{p_i^{OK} s_{OK} E[q_i]}{1 - (1 - c_S)\, p_s^{OK\to OK}\, s_{OK}\, E[1 - q_i]}}$ | **Expected Mean Fitness (Baseline Rogers' Paradox):** Computes the expected mean fitness $E[q^{OK}]$ when both individual and social learners exist. The numerator captures success of individual learning; the denominator reflects the dependency of social learners on individual learners generating new knowledge. |
| $\mathbf{E[q_s^{OK}] = \dfrac{(1 - c_S)\, p_s^{OK\to OK}\, s_{OK}\, p_i^{OK}\, s_{OK}\, E[q_i]}{1 - (1 - c_S)\, p_s^{OK\to OK}\, s_{OK}\, E[1 - q_i]}}$ | **Expected Mean Fitness of Social Learners:** Shows that social learners' fitness depends on individual learners' success. It equals $E[q^{OK}]$ multiplied by the probability of social learning success. |
| $\mathbf{E[q^{OK}] = p_i^{OK} s_{OK} E[q_i] + E[p_s^{OK}] s_{OK} E[q_s] + E[p_{AI}^{OK}] s_{OK} E[q_{AI}]}$ | **Mean Population Fitness (AI Rogers' Paradox):** The total expected fitness in equilibrium across individual, human-social, and AI-social learners, each weighted by their proportion and respective success rates. |
| **4. Human-Centric Strategy: Critical Social Learning** | |
| $\mathbf{p_{cs}^{OK} = 1 - (1 - p_s^{OK})(1 - p_i^{OK})}$ | **Critical Social Learning Success Probability:** Defines the success probability for agents who first attempt social learning and switch to individual learning if social learning fails. Subtracts the joint failure probability of both methods from 1. |
| **5. Model-Centric Strategies: AI Update Schedule and Individual Learning** | |
| $p_{x\to q^{OK}}^{AI} := 1 - c_{\lambda s}$ | **AI Social Update Probability:** Probability that the AI successfully updates its adaptation state through social learning, where $c_{\lambda s}$ is the associated cost. |
| $p_{x\to x}^{AI} := c_{\lambda s}$ | **AI Self-Maintenance Probability:** Probability that the AI fails to update socially (retains its current state $x$) due to cost $c_{\lambda s}$. |
| $\mathbf{p_{AI}^{OK,(t+1)} = 1 - (1 - p_{x\to q^{OK}}^{AI}\, q^{OK})(1 - p_{AI}^{OK,t}\, p_{x\to x}^{AI}\, (1 - u))}$ | **AI Adaptation Level Update (Social Learning Cadence):** Computes the AI's adaptation at timestep $t+1$ based on its previous success, update probabilities, and environmental change rate $u$. The equation subtracts the total failure probability from 1. |
| $\mathbf{p_{x\to OK}^{AI} := (1 - c_{\lambda i})z_{AI}}$ | **AI Individual Learning Success Probability:** Defines the probability of successful adaptation when the AI performs individual learning, with cost $c_{\lambda i}$ and intrinsic success rate $z_{AI}$. |
| **6. Negative Feedback Environment Model (Deskilling)** | |
| $\kappa_{\mathbf{j}}^{\mathbf{0}} = \mathbf{1}$ | **Initial Penalty Value:** Sets the initial value of the learning penalty ($\kappa$) for any agent $j$ to 1, meaning no penalty at the start. |
| $\kappa_{\mathbf{j}}^{\mathbf{t+1}} = \mathbf{0.9}\kappa_{\mathbf{j}}^{\mathbf{t}}$ | **Penalty Update Rule:** After each instance of social learning from the AI, the agent's $\kappa_j$ value is multiplied by 0.9, modeling skill degradation over time. |
| $\mathbf{p_{ij}^{OK} = (1 - c_i)z_i}\kappa_{\mathbf{j}}$ | **Individual Learning Success Probability with Penalty:** Modifies the individual learning success probability by incorporating $\kappa_j$, showing that reliance on AI reduces future individual learning capability. |

## 2 Updated Mathematical Definitions and Simulation Framework

The project proposal outlines significant architectural changes to the simulation environment, replacing the simple binary state with a complex, multi-state "Knowledge Map" or "Tech Tree." Because of this shift, the original algebraic equations (which derived equilibrium fitness based on a binary adapted/unadapted state) are conceptually replaced by new metrics and simulation rules.

Below is a full list of updated mathematical definitions, equations, and simulation rules that fit the proposed project framework.

### 2.1 New Agent State and Fitness Definitions

The fundamental state representation of an agent is changed from a single binary value (adapted or not adapted, $OK$ or $\neg OK$) to a multi-dimensional state vector.

- **Agent State Vector:** The agent's adapted status is replaced by a vector corresponding to discovered nodes on a "tech tree." The adaptation status is thus represented by an $n$-dimensional vector:

  $$\mathbf{s_j}(t) = [x_1, x_2, \ldots, x_n]$$

  where $x_i = 1$ if node $i$ is discovered by agent $j$, and 0 otherwise.

- **Survival Mapping ($s$):** The original survival probability ($s_{OK}$) is replaced by a function mapping the agent's current position on the Knowledge Map to a survival probability:

  $$s(\mathbf{s_j}) : \mathbb{R}^n \to [0, 1]$$

  - Different nodes (states) correspond to varying survival probabilities, defining a fitness landscape with local and global optima.

### 2.2 New Learning Mechanisms and Cost Structure

The original cost structure ($c_I$ and $c_S$) remains, but the result of successful learning is movement along the Knowledge Map rather than reaching a single "OK" state.

- **Individual Learning (IL):** Individual learning remains costly and risky.
  - Success corresponds to progression along the same branch of the Knowledge Map (incremental progress).
    - The success probability follows the original conceptual form:

      $$p_i = (1 - c_I)z_i$$

- **Social Learning (SL):** Social learning is inexpensive ($c_S \approx 0$).
  - Success corresponds to movement across branches in the Knowledge Map $\to$ cross-pollination of knowledge from another agent.

- **AI Social Learning (AI SL):** The AI's adaptation level is defined as the modal or mean population state:

  $$\mathbf{s_{AI}}(t) = \text{mode}(\mathbf{s_1}(t), \ldots, \mathbf{s_N}(t))$$

  replacing the previous scalar definition $p_{AI}^{OK} := q^{OK}$.

### 2.3 Replacement for Expected Mean Fitness $E[q^{OK}]$

The original algebraic derivations (Equations 1–5) are invalid in an $n$-dimensional state environment. They are replaced by simulation-derived metrics measuring mastery, innovation, and convergence.

- **Mastery / Fitness:** *Mastery Score (Population Fitness)* — the collective understanding level, representing the mean proportion of discovered nodes across all agents:

  $$\text{Mastery}(t) = \frac{1}{Nn} \sum_{j=1}^{N} \sum_{i=1}^{n} x_{ij}(t)$$

- **Innovation / Discovery:** *Discovery Rate* — the frequency with which agents reach previously undiscovered nodes, particularly in high-risk branches:

  $$D(t) = \frac{\#\text{ of new nodes discovered at } t}{n}$$

- **Speed / Convergence:** *Convergence Rate ($T_c$)* — number of iterations required for the Mastery Score to stabilize or reach a critical competence threshold $\theta$:

$$T_c = \min\{t \mid \text{Mastery}(t) \geq \theta\}$$

- **Threshold Definition:**

$$\theta \in (0, 1)$$

  defines the population's target competence level used to calculate $T_c$.

## 2.4 Strategy Equations

While most algebraic fitness equations are replaced by simulation metrics, key strategy rules are retained to test learning and AI update dynamics.

$$\mathbf{p_{cs}^{OK}} = 1 - (1 - p_s^{OK})(1 - p_i^{OK}) \qquad \text{(Critical Social Learning Success)} \tag{1}$$

$$\mathbf{p_{x \to q^{OK}}^{AI}} = 1 - c_{\lambda s} \qquad \text{(AI Social Update Probability)} \tag{2}$$

$$\mathbf{p_{x \to OK}^{AI}} = (1 - c_{\lambda i}) z_{AI} \qquad \text{(AI Individual Learning Success)} \tag{3}$$

$$\mathbf{p_{AI}^{OK,(t+1)}} = 1 - (1 - p_{x \to q^{OK}}^{AI} q^{OK})(1 - p_{AI}^{OK,t} p_{x \to x}^{AI}(1 - u)) \qquad \text{(AI Adaptation Update Rule)} \tag{4}$$

These equations are used to explore trade-offs between update frequency, discovery rate, and collective mastery.

## 2.5 Deskilling / Negative Feedback Mechanism

The explicit penalty factor $\kappa$ from the original model is not directly included. Instead, negative feedback is implemented structurally through agent behavior on the Knowledge Map.

- **Original Mechanism:**
  - Deskilling through $\kappa$:

$$p_{ij}^{OK} = (1 - c_i) z_i \kappa_j$$

- **Proposed Analogy:**
  - Incentive-based loss of innovation:
    * Over-reliance on the AI (which converges toward the average or modal state) pushes the population toward low-risk, low-reward branches, reducing incentive for high-risk exploration and thereby simulating a deskilling effect.

# 3 Formal Definitions and Equations

## 3.1 Agent State and Fitness Definitions

The fundamental definitions for agents and fitness are formalized as follows:

### 3.1.1 Agent State Vector

Each agent $j$'s knowledge is defined by a state vector $K_j(t) \in \mathbb{Z}^n$.

### 3.1.2 General Fitness Function

Survival probability is defined as a function of the agent's knowledge state:

$$P(\text{survival} \mid K_j(t)) = V(K_j). \tag{5}$$

### 3.1.3 Low-Risk Fitness Landscape

Defined by linear growth, modeling a local optimum:

$$V_{\text{Low-Risk}}(d) = s_0 + \delta \cdot d \tag{6}$$

where $d$ is the depth along the branch.

#### 3.1.4 High-Risk Fitness Landscape (Simplified)

Defined using a sigmoid function for stable, interpretable dynamics (slow initial growth, eventual saturation at $V_{\max}$):

$$V_{\text{High-Risk}}(d) = \frac{V_{\max}}{1 + e^{-\gamma(d-d_0)}}. \tag{7}$$

The original design used a logarithmic function, $V_{\text{High-Risk}}(d) = s_0 + \beta \cdot \log(1 + d)$.

### 3.2 Learning Rules

The learning rules define how agents update their state vector $K_j(t)$.

#### 3.2.1 Incremental Individual Learning (IL)

Advances the agent one step along its current branch ($e_i$):

$$K_j(t + 1) = K_j(t) + e_i. \tag{8}$$

#### 3.2.2 Exploratory Individual Learning (IL)

Allows agents to switch branches with probability $p_e$ at any depth, resetting progress in the new branch:

$$K_j(t + 1) = \begin{cases} \text{reset to High-Risk node 0}, & \text{if exploration occurs,} \\ K_j(t) + e_i, & \text{otherwise.} \end{cases} \tag{9}$$

#### 3.2.3 Social Learning (SL) Bias

Agents copy a peer's knowledge state (subgraph) with a prestige/success bias, originally formalized using exponential weighting on fitness:

$$P(\text{copy } K_{\text{peer}}) \propto e^{\alpha \cdot V(K_{\text{peer}})}. \tag{10}$$

The proposal suggests simplifying this to a rank-based rule.

#### 3.2.4 AI Learning (AI)

The AI aggregates population states. The original plan was to compute the modal state; a simplified implementation suggests reinforcing the dominant branch by comparing mean depths ($\bar{d}_{\text{Low}}$ vs. $\bar{d}_{\text{High}}$):

$$K_{AI}(t) = \text{mode}\{K_1(t), K_2(t), \ldots, K_N(t)\}. \tag{11}$$

### 3.3 Simulation Metrics

The new metrics (which replace the original algebraic fitness derivations) are formalized as follows:

#### 3.3.1 Mastery Score ($Q_M$)

The mean population fitness, replacing $E[q^{OK}]$:

$$Q_M(t) = \frac{1}{N} \sum_{j=1}^{N} V(K_j(t)). \tag{12}$$

#### 3.3.2 Convergence Time ($T_c$)

The time required for the Mastery Score to exceed a threshold ($\theta$):

$$T_c = \min\{t : Q_M(t) \geq \theta\}. \tag{13}$$

#### 3.3.3 Discovery Rate ($R_D$) (Streamlined)

The number of new node visits per timestep (to avoid costly set operations):

$$R_D(t) = \frac{1}{\Delta t} \sum_{j=1}^{N} \mathbf{1}[K_j(t) \notin \mathcal{D}_{t-1}] \tag{14}$$

where $\mathcal{D}_{t-1}$ is the set of all previously visited nodes. The original definition was

$$R_D = \frac{\Delta \left| \bigcup_j K_j(t) \right|}{\Delta t}.$$

### 3.4 Deskilling (Mechanistic Implementation)

The mechanistic implementation of deskilling, Incentive-Based Loss of Innovation (the population is channeled away from the necessary exploration, leading to stagnation at a local minimum), is confirmed by the design choices, particularly the AI's focus on the majority state combined with the existence of the Low-Risk plateau. The design ensures that if the average depth ($\bar{d}$) favors the Low-Risk branch, the AI will reinforce that choice, which is the core structural mechanism causing the stagnation (analogous to deskilling).