Technical Appendix

This appendix provides a detailed explanation of the mathematical formulation used in the reinforcement learning environment, particularly how **Principal Component Analysis (PCA)** and **Normalization** are applied to define the state dynamics.

State Dynamics using PCA and Normalization

The state $s_t \in \mathcal{S}$ at any time step t represents the status of each group's **knowledge**, **health**, **innovation**, and **potential private consumption**. These features are extracted from a set of sample data, as described earlier, and reduced in dimensionality through **Principal Component Analysis (PCA)**.

The process begins with generating a dataset where each feature (e.g., **Sampled Knowledge**, **Health Status**, **Innovation Capacity**, etc.) is normalized using **Min-Max Normalization** to scale the data between 0 and 1:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Where x represents the feature values, and $x_{\min}$, $x_{\max}$ are the minimum and maximum values for that feature. After normalization, the features are combined into a matrix X, where each row corresponds to an individual's features.

Next, **Principal Component Analysis (PCA)** is applied to reduce the high-dimensional feature space to three principal components:

$$X' = P \cdot X$$

Where P is the projection matrix obtained from the PCA transformation, and X' represents the projected data in three-dimensional space. These three principal components capture the most significant variance in the data and serve as the basis for constructing the **state space** for the groups in the environment.

The state s_t for group i at time t consists of three components:

$$s_t^i = (k_t^i, h_t^i, in_t^i, p_t^i)$$

Where: - k_t^i represents **knowledge** (derived from the first PCA component), - h_t^i represents **health** (derived from the second PCA component), - in_t^i represents **innovation** (derived from the third PCA component), - p_t^i represents **potential private consumption** (calculated as a function of public investment consumption and taxes).

Action Space and State Transitions

The actions $a_t \in \mathcal{A}$ represent the investments in **education**, **health**, and **innovation** for each group, constrained between -1 and 1. At each time step, the agent chooses an action vector:

$$a_t = [a_t^{\textbf{education}}, a_t^{\textbf{health}}, a_t^{\textbf{innovation}}]$$

The state transitions are determined by the actions applied to the current state. The update rules for each group's **knowledge**, **health**, and **innovation** values are as follows:

$$k_{t+1}^i = k_t^i + 0.5 \cdot a_t^{\textbf{education}} \cdot (3 - k_t^i)$$
$$h_{t+1}^i = h_t^i + 0.5 \cdot a_t^{\textbf{health}} \cdot (3 - h_t^i)$$
$$in_{t+1}^i = in_t^i + 0.5 \cdot a_t^{\textbf{innovation}} \cdot (3 - in_t^i)$$

Here, the factor 0.5 ensures that the growth is scaled down, and the term $(3 - x)$ introduces diminishing returns as the feature value x approaches its maximum of 3.

The **potential private consumption** p_t^i is updated based on the total investment in **education**, **health**, and **innovation**:

$$p_{t+1}^i = p_t^i + 0.5 \cdot (a_t^{\textbf{education}} + a_t^{\textbf{health}} + a_t^{\textbf{innovation}}) \cdot (3 - p_t^i)$$

All state values are clipped to remain within the range [0, 5].

Reward Function

The reward function incentivizes **balanced growth** across all dimensions, penalizing imbalance and extreme actions. The total reward r_t at time t is:

$$r_t = \textbf{Improvement} + \textbf{Balanced Growth Bonus} - \textbf{Imbalance Penalty} - 0.05 \cdot \sum_{d \in \{a^{\textbf{edu}}, a^{\textbf{hea}}, a^{\textbf{inn}}\}} a_t^d$$

Where: - **Improvement** is the sum of improvements across all dimensions:

$$\textbf{Improvement} = \sum_{i=1}^{3} \sum_{d \in \{k,h,in\}} (s_{t+1}^i(d) - s_t^i(d))$$

- **Balanced Growth Bonus** rewards equal growth across dimensions:

$$\textbf{Balanced Growth Bonus} = 0.05 \cdot \textbf{mean}(s_{t+1} - s_t)$$

- **Imbalance Penalty** discourages uneven growth:

$$\textbf{Imbalance Penalty} = 0.1 \sum_{i=1}^{3} \sum_{d \in \{k,h,in\}} \left| s_{t+1}^i(d) - \frac{1}{3} \sum_{j \in \{k,h,in\}} s_{t+1}^i(j) \right|$$

- The final term $0.05 \cdot \sum a_t^d$ applies a small penalty for extreme actions to prevent rapid changes in policy.

Policy Optimization: Actor-Critic Approach

The **Actor-Critic** reinforcement learning framework is used to optimize the policy:

Critic Network (Value Estimation)

The critic network estimates the value of state-action pairs using a value function $Q(s_t, a_t)$, which predicts the expected future reward:

$$Q(s_t, a_t) = E\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

Actor Network (Policy Optimization)

The actor network optimizes the policy by maximizing the expected return. The policy gradient is computed using:

$$\nabla_\theta J(\theta) = E\left[\nabla_\theta \log \pi_\theta(a_t|s_t) Q(s_t, a_t)\right]$$

Where $\pi_\theta(a_t|s_t)$ is the policy parameterized by θ.

Soft Updates

Soft updates are used to gradually update the target networks:

$$\theta_{\text{target}} = \tau\theta_{\text{local}} + (1 - \tau)\theta_{\text{target}}$$

Where τ is a small constant (e.g., 0.005) that ensures smooth updates to the target network.