The AI economist: Improving Equality and Productivity with AI-Driven Tax Policies

Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C. Parkes, and Richard Socher, 2020, mimeo.

Presenter: Yoji Tomita

RL-GT ゼミ June 17, 2021

Table of Contents

1. Introduction

・イントロダクション

2. Economic Simulations: Learning in Gather-and-Build Games

- Economic environment について.
- まずは税の無い設定 (”free-market”) で説明する.

2.1 Notation and Preliminaries

- Partial-observable multi-agent Markov Games(MGs): $(S, A, r, \mathscr{T}, \gamma, o, \mathscr{I})$
  - S : 状態空間 (state space)
  - A : 行動空間 (action space)
  - $r_{i,t}$: 報酬関数 (reward function)
  - $\mathscr{T}$: 遷移関数 (transition function) $s_{t+1} \sim \mathscr{T}(\cdot \mid s_t, \boldsymbol{a}_t)$
  - γ : 割引因子 (discount factor)
  - $o_{i,t}$: 観測 (observation)
  - time step $t = 0, 1, \ldots, H$.

- Agents' policy : $\pi_i(\cdot \mid o_{i,t}, h_{i,t}; \theta_i)$
  - $h_{i,t}$: hidden state（自分の私的情報と, 過去の history）
  - θ_i : policy の parameter
  - エージェント i は次の最大化問題を得く policy を求める:

$$\max_{\theta_i} \mathbb{E}_{a_i \sim \pi_i, \boldsymbol{a}_{-i} \sim \boldsymbol{\pi}_{-i}, s' \sim \mathscr{T}} \left[\sum_t \gamma^t r_{i,t} \right]. \tag{1}$$

- データ効率性のため, すべてのエージェントは training の間パラメータ θ を共有する.
- 彼らの行動 $\pi_i(a_i \mid o_i, h_i; \theta)$ は, agent-specific observations o_i と hidden-state h_i によって異なる.

t	time
i, j, k	agent indices
θ, ϕ	model weights
s	state
o	observation
a	action
r	reward
π	policy
γ	discount factor
$\mathcal{T}$	state-transition, world dynamics
h	hidden state

x	endowment
x^c	coin
x^s	stone
x^w	wood
z	income
l	labor
u	utility
T	tax
τ	tax-rate
π_p	planner policy
`swf`	social welfare
ω	social welfare weight
g	social marginal welfare weight
`gini`	Gini index
`eq`	Equality index

Table 1: Notation. Subscripts are indices. Superscripts are labels.

Gather-and-Build game

- 2 次元の grid (25×25) からなる世界が舞台.
- エージェントはフィールドを歩き回り, 資源 (石と木) を集め, それらを 1 つずつ使って家を建て, また資源を coin を介してトレードする.
- 資源は空タイルに確率的に産み出される.
- エージェントは家を建てると coin が得られるが, 得られる coin は agent の skill ごとに異なる.

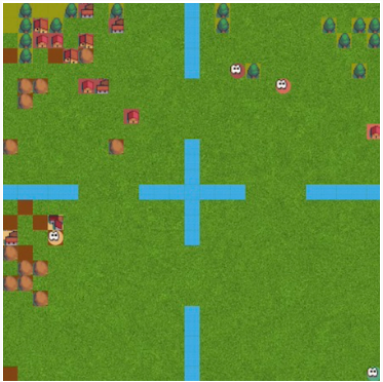
Labor and Skill.

- Agent の action(moving, gathering, trading, building) にはそれぞれ labor cost が設定されている.
- 各 time に agent がどれか 1 つ行動をとると, その labor cost がかかる.

- building skill (1 以上 3 以下) が各 agent に設定されていて, 家を建てると agent は $10\times$ skill 分の coin を得る.
- collection skill (1 以上 2 以下) もあり, 資源を拾うとこの skill 分の資源を得る (skill 1.2 の場合, 確定で 1 つ資源を得て, さらに確率 0.2 でもう 1 つ資源を得る)

Environment Scenario.

- field は水により 4 つの区域に別れている（水部分は通れない）
- 資源は空間的に集まって発生する.
- 4 agents
- buildng skills は 1.13, 1.33, 1.65, 22.2（Pareto 分布 w/ exponent $a = 4$, scale $m = 1$ の quartiles を元に設定）
- 1 episode は $H = 1000$ time steps からなる.



- Agent の utility function:

$$u_i(x_{i,t}, l_{i,t}) = \text{crra}\left(x^c_{i,t}\right) - l_{i,t}, \quad \text{where } \text{crra}(z) = \frac{z^{1-\eta} - 1}{1-\eta}, \ \eta > 0. \tag{2}$$

  - $x_{i,t} = (x^w_{i,t}, x^s_{i,t}, x^c_{i,t})$: i の保有する木・石・コイン.
  - $l_{i,t}$: 蓄積労働量.
  - η: エージェントの utility function の non-linearity をコントロールするパラメータ.
- Rational economic agent は以下の最大化を行う.

$$\forall i : \max_{\pi_i} \mathbb{E}_{a_i \sim \pi_i,\ \boldsymbol{a}_{-i} \sim \boldsymbol{\pi}_{-i}, s' \sim \mathscr{T}} \left[u_i(x_{i,0}, l_{i,0}) + \sum_{t=1}^{H} \gamma^t \underbrace{\left(u_i(x_{i,t}, l_{i,t}) - u_i(x_{i,t-1}, l_{i,t-1})\right)}_{=r_{i,t}} \right]. \tag{3}$$

Deep RL agents

- deep neural network を用いる agent policy を modelling する:

$$a_{i,t} \sim \pi(o_{i,t}^{\text{world}}, o_{i,t}^{\text{agent}}, o_{i,t}^{\text{market}}, o_{i,t}^{\text{tax}}, h_{i,t-1}; \theta)$$

  - $o_{i,t}^{\text{world}}$: 近くの状況に関する観測.
  - $o_{i,t}^{\text{agent}}$: public な agent の状況 (資源・コイン保有) と, private agent states(skill 値と labor performed)
  - $o_{i,t}^{\text{market}}$: transfer market の状況（bid, ask offer）
  - $o_{i,t}^{\text{tax}}$: tax rates

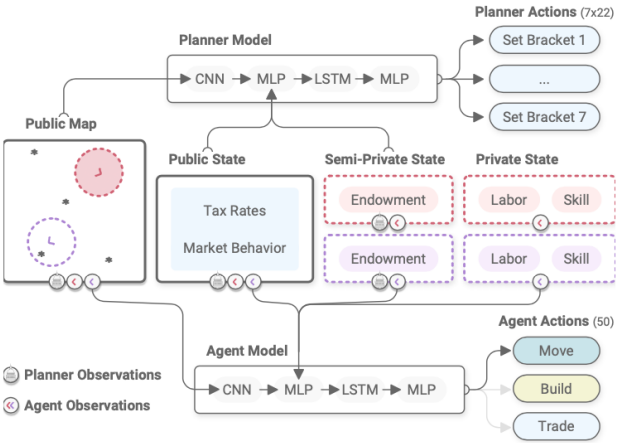
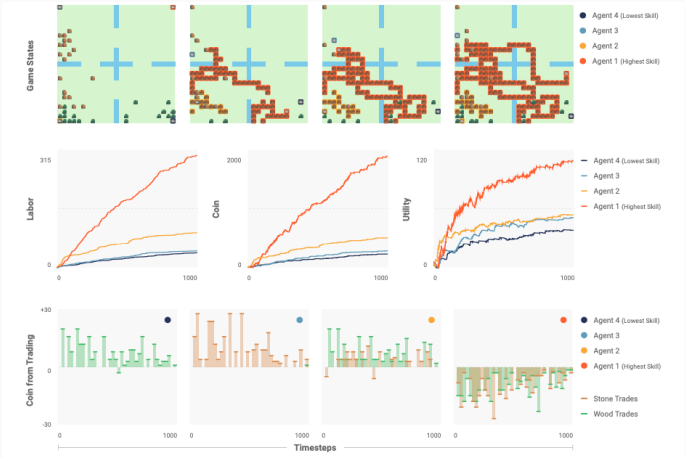


Figure 3: Schematic overview of the general network architecture used in our work. Spatial observations are processed by a stack of two convolutional layers (CNN) and flattened into a fixed-length feature vector. This feature vector is concatenated with the remaining observation inputs and the result is processed by a stack of two fully connected layers (MLP). The output is then used to update the hidden state of an LSTM and action logits are computed via a linear projection of the updated hidden state. Finally, the network computes a softmax probability layer for each action head. For the agent policy, there is a single action space and action head. For the tax policy, there is a separate action space and action head for each tax rate the tax policy controls (described below).

Emergent Behavior of AI Agents



- 左図は no-tax 下で train 後の AI agents の 1 episode の行動の例.
- low-skill agents（紺, 水）は資源を集めて market で売ることに徹している.
- high-skill agent（オレンジ）は market で資源を買って家を建てている.
- 黄色は最初は家を建ててるが, のちに資源を得る方にスイッチしている.

3. Machine Learning for Optimal Tax Policies

- 課税と再分配を行う social planner を導入する.
- Social planner は, 生産性と平等性の trade-off に直面している.
  - 無課税（free-market）では生産性は最大化されるが, 不平等.
  - 課税・再分配を行うと平等性が増すが, 生産性が落ちる.

- ここでは, free-market, US-federal, Saez framework, AI economist による social planner を試す.

Income Taxes.

- Tax period は M steps 続く（実験では $M = H/10$ とし, 1 episode に 10 tax periods があるものとする）
- ピリオド p の税は, time step t から $t + M$ までの収入 z_i^p に課される.

- Tax period の初めに, social planner は tax schedule $T(z)$ を決めて公表する.
  - 各 agent i は, 収入 z_i^p に応じて $T(z)$ を支払う.
  - 集められた税は, 全 agent に平等に分配される.
  - よって, 分配後の agent i の収入は,

$$\tilde{z}_i^p = z_i^p - T(z_i^p) + \frac{1}{N} \sum_{j=1}^{N} T\left(z_j^p\right). \tag{5}$$

Bracketed Tax Schedules.

- Scheme 間の比較を可能にするため, tax schedule は次のように ”bracketed” されたもののみを考える.
- Cut-off income levels $\{m_b\}_{b=0}^{B}$ s.t. $0 = m_0 \leq m_1 \leq \cdots \leq m_{B-1} \leq m_B = +\infty$ が先に与えられている.
- Social planner は, 各 bracket b に含まれる収入に対して適用される marginal tax rate $\tau_b \in [0, 1]$ を選ぶことで, tax schedule $T(\cdot)$ を決定する.

$$T(z) = \sum_{b=0}^{B-1} \tau_b \cdot ((m_{b+1} - m_b) \cdot 1[z > m_{b+1}] + (z - m_b) \cdot 1[m_b < z \leq m_{b+1}]) .$$

Social Welfare Functions

- Social planner の目的関数である social welfare function は, 生産性と平等性の trade-off を組み込めるように次のように決める.
- エージェントのコイン保有 $\boldsymbol{x}^c = (x_1^c, \ldots, x_N^c)$ に対し, equality を次で定義:

$$\mathbf{eq}(\boldsymbol{x}^c) = 1 - \mathbf{gini}(\boldsymbol{x}^c) \cdot \frac{N}{N-1}, \quad 0 \leq \mathbf{eq}(\boldsymbol{x}^c) \leq 1. \tag{7}$$

where

$$\mathbf{gini}(\boldsymbol{x}^c) = \frac{\sum_{i=1}^N \sum_{j=1}^N |x_i^c - x_j^c|}{2N \sum_{i=1}^N x_i^c}, \quad 0 \leq \mathbf{gini}(\boldsymbol{x}^c) \leq \frac{N-1}{N} \tag{8}$$

  - $\mathbf{eq}$ は, 1 で完全に平等 (全員同じ収入), 0 で完全に不平等 (1 人が全コインを独占).

- 生産性は,

$$\mathbf{prod}(\boldsymbol{x}^c) = \sum_{i=1}^{N} x_i^c. \tag{9}$$

- この $\mathbf{eq}$ と $\mathbf{prod}$ を social welfare function とする.[1]

$$\mathbf{swf}_t(\boldsymbol{x}_t^c) = \mathbf{eq}_t(\boldsymbol{x}_t^c) \cdot \mathbf{prod}_t(\boldsymbol{x}_t^c). \tag{10}$$

[1] Social welfare function として, weight $\omega_i \geq 0$ を用いて

$$\mathbf{swf}_t(\boldsymbol{x}_t^c, \boldsymbol{l}_t) = \sum_{i=1}^{N} \omega_i \cdot u_i \left(x_{i,t}^c, l_{i,t}\right). \tag{11}$$

を用いることも可能.

The Planner's Problem.

- Social Planner は,
  - agent の保有資源・コイン $\boldsymbol{x}_{i,t}$, フィールドの状態（agent・資源の位置）と market の状況は観測可能,
  - 各 agent の skill は直接には観測できない.
- Planner の最大化問題は,

$$\max_{\pi_p} \mathbb{E}_{\tau\sim\pi_p, \boldsymbol{a}\sim\boldsymbol{\pi}, s'\sim\mathscr{T}} \left[\mathbf{swf}_0 + \sum_{t=1}^{H} \gamma^t \underbrace{(\mathbf{swf}_t - \mathbf{swf}_{t-1})}_{=r_{p,t}} \right]. \tag{12}$$

3.3 Inner-Outer-Loop Reinforcement Learning

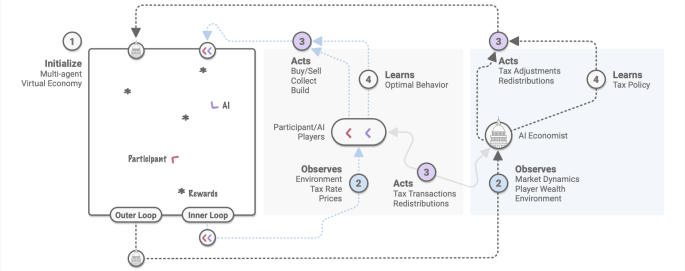


Figure 6: Two-level RL. In the inner loop, RL agents gain experience by performing labor, receiving income, and paying taxes, and learn through balancing exploration and exploitation how to adapt their behavior to maximize their utility. In the outer loop, the social planner adapts tax policies to optimize its social objective.

- この状況では, Inner loop で agents が行動を選んで学習し, Outer loop で planner が税制を選んで学習することになる.
- Inner loop での Agent の学習行動と, outer loop で planner の選ぶ税制が相互に影響し合うため, 報酬が agent と planner 双方にとって不安定になる.

Algorithm 1 Inner-Outer Loop Reinforcement Learning. Economic agents and social planner learn simultaneously. Bold-faced symbols indicate quantities for multiple agents. Note that agents share weights.

Require: Sampling horizon $\hbar$, tax period length M
Require: On-policy learning algorithm $\mathbb{A}$ (for instance, A3C, PPO)
Require: Stopping criterion C (for instance, agent and planner rewards have not improved)
Ensure: Trained agent and planner policy weights θ, ϕ

$s, \boldsymbol{o}, o_p, \boldsymbol{h}, h_p \Leftarrow s_0, \boldsymbol{o}_0, o_{p,0}, \boldsymbol{h}_0, h_{p,0}$ ▷ Reset episode
$\theta, \phi \Leftarrow \theta_0, \phi_0$ ▷ Initial agent and planner policy weights
$D, D_p \Leftarrow \{\}, \{\}$ ▷ Reset agent and planner transition buffers
while training **do**
  for $t = 1, \ldots, \hbar$ **do**
    $\boldsymbol{a}, \boldsymbol{h} \Leftarrow \boldsymbol{\pi}(\cdot|\boldsymbol{o}, \boldsymbol{h}, \theta)$ ▷ Sample agent actions; update hidden state
    if t mod M = 0 **then** ▷ First timestep of tax period
      $\tau, h_p \Leftarrow \pi_p(\cdot|o_p, h_p, \phi)$ ▷ Sample marginal tax rates; update planner hidden state
    else
      no-op, $h_p \Leftarrow \pi_p(\cdot|o_p, h_p, \phi)$ ▷ Only update planner hidden state
    end if
    $s', \boldsymbol{o}', o'_p, \boldsymbol{r}, r_p \Leftarrow$ `Env.step`$(s, \boldsymbol{a}, \tau)$ ▷ Next state / observations, pre-tax reward, planner reward
    if t mod M = M-1 **then** ▷ Last timestep of tax period
      $s', \boldsymbol{o}', o'_p, \boldsymbol{r}, r_p \Leftarrow$ `Env.tax`(s', τ) ▷ Apply taxes; compute post-tax rewards
    end if
    $D \Leftarrow D \cup \{(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{r}, \boldsymbol{o}')\}$ ▷ Update agent transition buffer
    $D_p \Leftarrow D_p \cup \{(o_p, \tau, r_p, o'_p)\}$ ▷ Update planner transition buffer
    $s, \boldsymbol{o}, o_p \Leftarrow s', \boldsymbol{o}', o'_p$
  end for
  Update θ, ϕ using data in D, D_p and $\mathbb{A}$.
  $D, D_p \Leftarrow \{\}, \{\}$ ▷ Reset agent and planner transition buffers
  if episode is completed **then**
    $s, \boldsymbol{o}, o_p, \boldsymbol{h}, h_p \Leftarrow s_0, \boldsymbol{o}_0, o_{p,0}, \boldsymbol{h}_0, h_{p,0}$ ▷ Reset episode
  end if
  if criterion C is met **then return** θ, ϕ
  end if
end while

4.1 Baseline Methods

- 次の 4 つの tax model を比較する.
  - free-market (no taxes)
  - US federal single-filer 2018 tax schedule
  - Saez tax formula (adapted for a multi-period setting)
  - AI Economist planner
- Tax bracket は, 全ての tax model に共通で, 2018 US federal incom tax をもとに 1/1000 にスケーリングした以下を用いる.

$$\boldsymbol{m} = [0, 9.7, 39.475, 84.2, 160.725, 204.100, 510.3, \infty]. \tag{15}$$

US Federal Income Tax Rates (Single-filer, 2018).

- 2018 US federal income tax をもとに, bracket tax rates は,

$$\boldsymbol{\tau} = [0.1, 0.12, 0.22, 0.24, 0.32, 0.35, 0.37] \tag{16}$$

  とする.

Saez Tax Formula (single-period)

- Saez [2001] をもとに, まず single-period economy での optimal tax rates を求める.
- f, F は (pre-tax) 収入の分布の probability density と cummulative distribution function とし, planner はそれらを観測できるとする.
- Saez [2001] では, まず linear-weighted social welfare functions (11) に対し, social marginal welfare weights

$$g_i = \frac{\mathrm{d}\mathbf{swf}}{\mathrm{d}u_i}\frac{\mathrm{d}u_i}{\mathrm{d}x_i^c} = \omega_i \frac{\mathrm{d}u_i}{\mathrm{d}x_i^c}.$$

  を求める.
- これを我々のモデルに当てはめるために $g_i = \frac{1}{z_i}$ とし, さらにこれを normalize してよい $\sum_{i\in\mathscr{T}} g_i = 1$.

- $\alpha(z)$ を, the marginal average income at income z, normalized by the fraction of incomes above z, つまり,

$$\alpha(z) := \frac{z \cdot f(z)}{1 - F(z)}. \tag{18}$$

- $G(z)$ は, normalized, reverse cumulative Pareto weight over incomes above a threshold z

$$G(z) := \frac{1}{1 - F(z)} \int_{z'=z}^{\infty} f(z')g(z')\mathrm{d}z'. \tag{19}$$

- また, 弾力性 (elasticity) $e(z)$ を, average sensitivity of an agent's income to changes in the tax rate, つまり

$$e(z) = \frac{\mathrm{d}z/z}{\mathrm{d}(1 - \tau(z))/(1 - \tau(z))}. \tag{20}$$

- Saez [2001] によると, optimal marginal tax-rate は

$$\tau(z) = \frac{1 - G(z)}{1 - G(z) + \alpha(z)e(z)} \tag{21}$$

となり, これは income distribution と elasticity に依存して決まる.

Saez Tax Formula (multi-period)

- Saez formula (21) を使うには, income elasticity $e(z)$ の推定が必要.
- Gruber and Saez [2002] に従い, constant tax elasticity $\tilde{e}$ を仮定すると,

$$z_t = z^0 \cdot (1 - \tau_t)^{\tilde{e}}. \tag{22}$$

- したがって,

$$\log(z_t) = \tilde{e} \cdot \log(1 - \tau_t) + \log(z^0) \tag{23}$$

  を過去の tax period のデータから OLS で推定して用いる.

- AI Economist は, deep neural network により各 bucket の marginal tax rate を決める.

$$\tau \sim \pi_p \left(o_{p,t}^{\text{world}}, o_{p,t}^{\text{agent}}, o_{p,t}^{\text{market}}, o_{p,t}^{\text{tax}}, h_{p,t-1}; \phi \right). \tag{24}$$

  - Basic network architecture は agent と同様 (Figure 3).
  - agent と planner は持つ observation が異なる（planner はフィールド全体が見れるが, agent の private skill は見れない）.

4.2 Training Strategy: Two-phase Training and Tax Curricula

- Section 3.3 で述べたように, この joint optimization は不安定性をもたらす.
- 学習を安定化させるため, ここでは次の two-pahse training approach を行った.
  - First phaze: agent models の集合に対し, 無課税 (no-tax scenario) で学習させる.
  - Second phaze: 税モデルを融合にし, エージェント・プランナーの学習を継続する. また突然の税導入による不安定性を回避するため, 限界税率の上限を 10% から 100% に徐々に引き上げる.
- また, planner policy に entropy regularization を行うことも効果的だった.
  - Entropy regularization は, policy gradient objective に次の additional, weighted term

    $$\mathbf{entropy}(\pi) = -\mathbb{E}_{a\sim\pi(\cdot|s)}[\log \pi(a \mid s)].$$

    を加える（Williams and Peng [1991], Mnih et al. [2016]）.

- 実験は RLlib flamework [Liang et al., 2018] を用いて実施.
- また Policy gradients の計算に, proximal policy gradients [Schulman et al., 2017] と Adam optimizer [Kingma and Ba, 2014] を使う.
- サンプルは 60 の environment replicas から平行して集め, sampling horizon は 200 timesteps とする.
- 全ての実験で, 400 million sample の phase two training を実施し, これは agent と planner model は stable policy への収束に十分であった.

4.3 Equality, Productivity, and Social Welfare Metrics

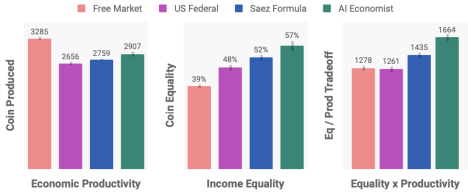


Figure 8: Comparison of overall economic outcomes (error bars show the variance during the final 20 million training steps, across 10 random seeds). The AI Economist achieves significantly better equality-productivity trade-offs compared to the baseline models. All baseline models have converged.

- Figure 8 は episode 終了時の各 tax model での economic outcomes の比較.
- Tax は常に productivity を下げるが, その下り幅は AI Economist が最も少ない.
- Income equality（$1 - \text{Gini index}$）は, AI economist で最も高い.
- Equality と Productivity の積も AI economist が最も高く, 次に高い Saez formula と比べても 16% 高い.

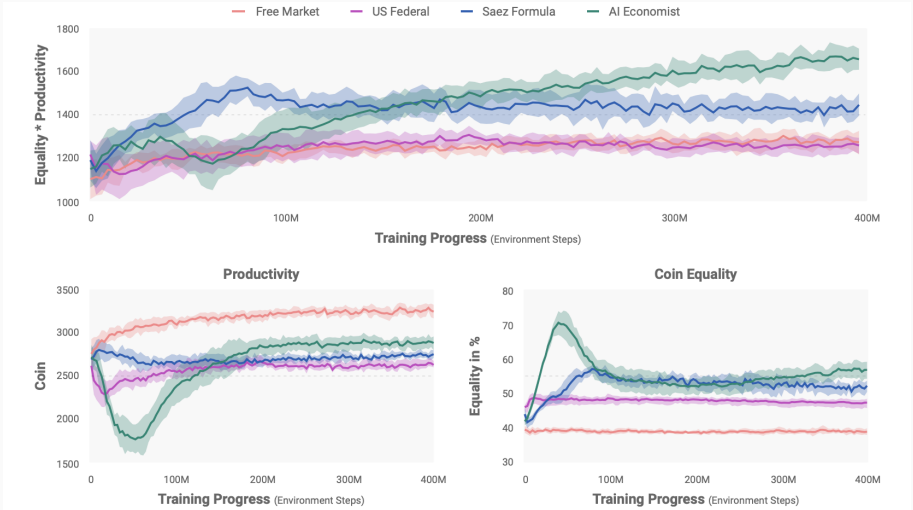


Figure 7: Empirical training progress for all models. The AI Economist (Green) achieves significantly better social outcomes than the baseline models. All baseline models have converged.

Comparing Tax Schedule

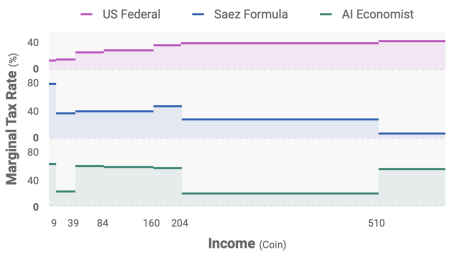


Figure 9: Comparison of average tax rates per episode. Variances within the Saez and AI Economist schedules are not shown. On average, the AI Economist sets a higher top tax rate than both of the US Federal and Saez tax schedules. The free-market collects zero taxes.

- Figure 9 は, 各 tax model での marginal tax rate の比較.
- US federal は, higher income に対して marginal tax rate は上がっていく.
- Saez formula では概ね逆.
- AI Economist はその blend になってる.

Effective Tax After Redistribution.

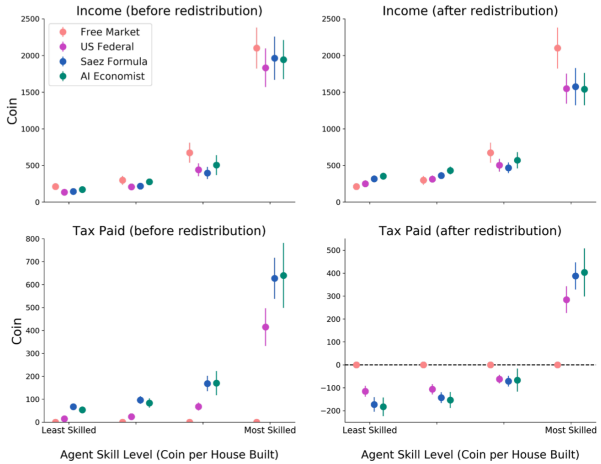


Figure 11: Agent-by-agent averages after sorting by skill. Income before redistribution (top-left) shows the average pre-tax income earned by each kind of agent. The amount of tax paid before distribution is shown in the bottom left. The amount of tax paid after redistribution is shown in the bottom right (the lower skill agents receive a net subsidy). The income after redistribution (top-right) shows the net average coin per agent at the

The Impact of Tax on Economic Activity

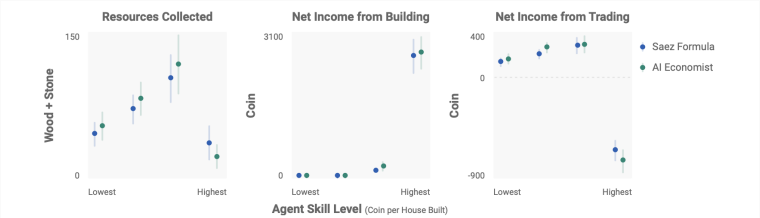


Figure 12: Comparing the impact of tax on economic activity for the Saez formula and the AI Economist. Left: Average number of resources (both wood and stone) collected per episode for each of the four agents. Middle: Average (per episode) total income earned from building. Right: Average (per episode) total income earned from trading. Negative income means the agent spends more than it earned. Lines indicate the standard deviation.

- Figure 12 は, Saez framework と AI economist での agent の行動の比較.
- AI economist では, high-skill worker はより資源を集めることより買って building することに特化し, low-skill workers はより資源を集めるようになっている.

4.5 Tax-Gaming Strategies

Figure 13: Left: Income of the highest-skilled agent for each tax period in an example episode with the AI Economist (green line). The dashed grey line shows the agent's average income. Right: Comparison of the total amount of tax the agent owed based on its actual income (green) and the tax it would have owed if it reported its average income in each period (gray). Each box in the column denotes the tax obligation in a single period.

- Figure 13 は AI economist での high-skilled worker の 1 episode の income の推移.
- 高収入の tax period と低収入の tax period が交互に起きている.
- Agent は収入をばらつかせることで課税額を抑えるように行動している (Saez でも)