Market-based Architectures in RL and Beyond

Blue Sky Ideas Track

Abhimanyu Pallavi Sudhir
University of Warwick, UK
abhimanyu.pallavi-sudhir@warwick.ac.uk

Long Tran-Thanh
University of Warwick, UK
long.tran-thanh@warwick.ac.uk

ABSTRACT

Market-based agents refer to reinforcement learning agents which determine their actions based on an internal market of sub-agents. We introduce a new type of market-based algorithm where the state itself is factored into several axes called "goods", which allows for greater specialization and parallelism than existing market-based RL algorithms. Furthermore, we argue that market-based algorithms have the potential to address many current challenges in AI, such as *search*, *dynamic scaling* and *complete feedback*, and demonstrate that they may be seen to generalize neural networks; finally, we list some novel ways that market algorithms may be applied in conjunction with Large Language Models for immediate practical applicability.

KEYWORDS

markets; prediction markets; alignment; RL

ACM Reference Format:
Abhimanyu Pallavi Sudhir and Long Tran-Thanh. 2025. Market-based Architectures in RL and Beyond: Blue Sky Ideas Track. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 5 pages.

1 INTRODUCTION

Before neural networks won the mandate of heaven, an AI research paradigm that had shown considerable promise was that of *market-based architectures*, i.e. AI agents that determine their output based on some internal market-based mechanism [3, 18].

There are general intuitive arguments that motivate such a line of research. Philosophers and psychologists have long pondered multi-agent models of the mind [23]; moreover, one may imagine that "any" machine learning task could in principle be solved by a market of agents solving sub-tasks with their individual reward set by the sale value of their output. There is work suggesting that markets can capture some notion of *bounded rationality*, e.g. the *Boundedly Rational Inductive Agent* (BRIA) [25] and *Algorithmic Bayesian Epistemology* [24]. In some sense, markets "aggregate" the intelligence or capacities of their individual participants.[1]

[1]See e.g. Hayek on the role of markets in aggregating information [32] – or the famous parable "I, Pencil" [19]: "... no one person, no matter how smart, could create from scratch a small, everyday pencil [yet the market makes over a billion of them each year] ...". There is also some empirical work on emergent intelligent behaviour in markets comprised of zero-intelligence traders [11, 17, 31]

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

"Market-based architectures" can be made concrete in the case of reinforcement learning (RL), where the majority of work in this area lies (see 1.1 for a brief summary). For example in the *Hayek machine* [2] and its derivatives, the setting is a Markov Decision Problem (MDP) and there is a single resource, the "right to act and collect reward", that is traded between sub-agents. At each time step, this resource is sold to the highest-bidding sub-agent, who performs some action that modifies the state and collects reward.

In this paper, we argue that market-based agents represent an underexplored and promising niche, *especially* in context of recent advancements in language models (LLMs), and have potential to address a range of present challenges in contemporary AI research. Specifically, we make the following claims and contributions:

Theoretical framework for market-based agents. We present two general frameworks for market-based RL agents: (1) the "deep market" (Def 2.1), where a single good, the *state*, is passed through a sequence of transacting agents, and (2) the "wide market" (Def 2.2), in which the state space itself is partitioned into factors called *goods*. The deep framework is not much of a departure from existing algorithms, and can be applied to any Partially Observed Markov Decision Process (POMDP); to our knowledge the wide framework is original to us, and is a generalization of the deep framework which better mirrors the success of real-world markets allowing for greater specialization and parallelism. A Python library for creating market-based algorithms will be released upon publication.

Markets, neural networks and backpropagation. We demonstrate that these market-based agents can in principle be applied even to basic supervised learning tasks such as classification, and that neural networks (though not backpropagation or gradient descent) emerge as a special case of them. Furthermore we generalize the result in [33] to wide markets, demonstrating a suggestive relationship between backpropagation and markets at equilibrium.

Search, complete feedback and alignment. We claim that markets can address several present problems in AI research, specifically: they are a natural framework for *search*, their scale or depth can be *dynamic* rather than fixed, and they allow *complete feedback* [8], a property widely regarded as valuable in AI alignment.

Markets and LLMs. We present novel ways in which market algorithms might be applied in conjunction with LLMs to address their limitations: they can be used for developing reasoning models like o1, and LLMs can facilitate "information markets" that can in turn improve human feedback mechanisms in AI training.

1.1 Related Work

Market-based RL. The majority of early work in this area has focused on market-based *reinforcement learning* (RL) algorithms. The pioneering work in this domain consists of Holland's *Learning Classifier Systems* or "bucket brigade" [15] in rule-based systems, where

condition-action agents ("classifiers") bid to post messages (which can be actions described in some language) onto a global message board. Improvements to this paradigm were made by Schmidhuber [29, 30] who allowed agents to determine their own bids and imposed credit conservation, and by Baum [2] who further strictly enforced property rights, resulting in a much more familiar set-up called the "Hayek Machine". The Hayek machine, which can be applied to any Markov Decision Process (MDP), was subsequently extended to POMDPs by [18] by adding external memory, and more recently [3] modified the framework to use Vickrey auctions and prove that a Nash equilibrium of the market produces a globally optimal policy.

Bounded rationality and markets. Other, more recent work in this area includes: *Logical Induction* [9], an algorithm that assigns probabilities to mathematical sentences based on their prices in a prediction market that (roughly speaking) pays off when a sentence is proven; and the *Boundedly Rational Inductive Agent* (BRIA) [25] which solves finite decision problems by assigning it to the highest-bidding trader, similar to in market-based RL. The key insight of these works is that markets are useful for modeling *boundedly rational agents*. The main results of each work – the fact that the logical inductor cannot by dominated by any polynomial-time trader, and the "boundedly rational inductive agent criterion" in the latter paper – are specific and precise formulations of the Efficient Market Hypothesis [24].

Markets and neural networks. A specific equivalence between classifier systems and neural networks has been studied in the *Neural bucket brigade* [7, 30], although this does not consider backpropagation. A suggestive analogy between markets and backpropagation is discussed in [33], though only for the case of a strictly sequential, unit-width market like that in Def 2.1. In our work we make this more precise and generalize it to 2.2

2 MARKET ALGORITHMS

Setting (POMDP). We assume a ususal POMDP setting, with a state space $\mathcal{S}$, action space $\mathcal{X}$, transition probability $\mathbf{P}(s' \mid s, x)$ (which allows us to treat actions as stochastic functions i.e. $x(s) \sim \mathbf{P}(s' \mid s, x)$), reward function $\mathbf{R}(s, x, s')$, and observation distribution $\omega(s) \sim \mathbf{O}(\omega \mid s)$ over a set of observations Ω. A policy is a map $\alpha : \Omega \to \mathcal{X}$, and the process proceeds as per usual.

Mirroring [18] and similar to Belief-MDP formulations, we can extend the state and message spaces by taking the cartesian product space with a message space str which is always preserved by ω; this gives the policy a "memory", or in terms of markets, creates informational goods.

The first algorithm we describe is Def 2.1: here, agents bid at each time step for the *right to act and collect reward*, and the highest-bidding agent is chosen to act. As in previous work, e.g. [2, 3], these agents are not utility-maximizers but programs out of a possibly infinite collection of agents $\mathcal{A}$ (which we leave abstract). The parameters of this algorithm are the wealths of each agent $w[\alpha]$. trained by the training loop Capitalism: at each step, the agent pays its bid to the previous agent, collects the reward generated by its actions and receives the bid of the next. This means that at equilibrium, each agent is incentivized to bid the value function, and perform the action with maximum Q-value.

Algorithm 1 Deep market

procedure Market ▷ Forward pass
  parameters: $w[\alpha] \in \mathbb{R}$ ▷ wealths of each $\alpha \in \mathcal{A}$
  input: $\omega \in \Omega$
  $b[\alpha] \leftarrow \min(\alpha_b(\omega), w[\alpha])$ for $\alpha \in \mathcal{A}$ ▷ Cap bids by wealth
  $\alpha^* \leftarrow \arg\max b[\alpha]$ ▷ Choose winning agent
  $x \leftarrow \hat{\alpha}^*(\omega)$ ▷ Determine action
  return x, α^*
end procedure
procedure Capitalism ▷ Training loop
  Initialize agent wealths $w[\alpha] \in \mathbb{R}$ for each $\alpha \in \mathcal{A}$
  Initialize original owner of the world α^*
  Initialize state $s \in \mathcal{S}$
  while $t \in \mathbb{N}$ **do**
    $\omega \leftarrow \omega(s)$ ▷ Generate observation
    $\alpha^*_{\text{prev}} \leftarrow \alpha^*$
    $x, \alpha^* \leftarrow$ Market(ω)
    $w[\alpha^*] \leftarrow w[\alpha^*] - b[\alpha^*]$ ▷ Pay bid
    $w[\alpha^*_{\text{prev}}] \leftarrow w[\alpha^*_{\text{prev}}] + b[\alpha^*]$ ▷ to previous owner
    $s_{\text{prev}} \leftarrow s$
    $s \leftarrow x(s)$ ▷ Transition state
    $w[\alpha^*] \leftarrow w[\alpha^*] + \mathbf{R}(s_{\text{prev}}, x, s)$ ▷ add reward to wealth
  end while
end procedure

Definition 2.1 (Deep market)**.** Assume a POMDP setup, and let $\mathcal{A}$ be a collection of "agents", which are (stochastic) maps $\alpha : \Omega \to \mathcal{X} \times \mathbb{R}$. The first component $\hat{\alpha} : \Omega \to \mathcal{X}$ of an agent is called its *action*, the second component $\alpha_b : \Omega \to \mathbb{R}$ is called its *bid*. The market algorithm then proceeds as in Algorithm 1.[2]

Some details have been ignored. $\mathcal{A}$ will usually be infinite and so tables like $w[\alpha]$ cannot simply be indexed on it: instead, $\mathcal{A}$ must be countably enumerated and added to the economy one-by-one in the training loop with each agent being endowed with some allowance. To prevent holdout problems, one may impose a small fixed "rent" on α^* at each training step i.e. $w[\alpha^*] \leftarrow (1 - \varepsilon)w[\alpha^*]$. A simple first-price auction is shown for simplicity, and may be replaced with a Vickrey auction in line with [3]. One may also replace the explicit reward function $\mathbf{R}$ with a class of "consumers" C who place bids upon desirable states, which may be a useful formulation for reinforcement learning from diverse human feedback[3].

Definition 2.1, which subsumes existing market-based RL, already illustrates one of the key defining features of markets recognizable to any student of economics: markets serve not only to *select* (via market competition) the best process to achieve a task, but also to *distribute* a complex task among agents which are individually much too weak or uninformed to complete the entire task. This means that the collection of actions $\mathcal{A}$ can be a class of "simple" agents, so that enumerating $\mathcal{A}$ can quickly find many valuable agents.

[2] For training, this may be executed in multiple episodes with different initial state s, either with finite episodes or in parallel (with shared wealth variables across running instances).
[3] see [6, 10] for a primer on this area

Specifically, Def 2.1 exploits modularity of *action*, where the state can be transformed one step at a time. There is however another form of modularity, missed by all existing market-based RL algorithms, which we may call modularity of *state*, and is crucial to the success of real-world markets: here, agents are not constantly transacting the whole "state of the world": instead, the state of the world is decomposed into several components, called *goods*[4]: $\mathcal{S} = \mathcal{S}_1 \oplus \cdots \oplus \mathcal{S}_n$. For instance, $s_1 \in \mathcal{S}_1$ might represent the quantity of iron ore in the world. Agents bid for small quantities of each good; no agent owns the whole world, and does not have to bother performing a valuation of the whole world. The "state of the world" may be recovered as the vector sum of all agents' holdings.

At least two new difficulties are introduced by considering markets of multiple divisible goods:

General equilibrium theory. Allocating goods is no longer as easy as an auction, because agents might have joint demand schedules for goods that are complementary or substitute to each other. The problem of matching buyers and sellers in this setting is the domain of "General Equilibrium Theory" in economics, where there are models such as the Fisher market and the Arrow-Debreu exchange market [1, 20]. Computing the equilibrium in these models is non-trivial and often intractable [4, 5]

Property rights in POMDPs. A more subtle difficulty lies in the fact that we want to divide the state $s \in \mathcal{S}$, which is not directly observed, among bidding agents (so that each of their actions only transform their respective portions of the state, i.e. their properties), but the agents only submit demand schedules over $\omega \in \Omega$. It is not obvious how to map a decomposition of a vector $\omega(s)$ back onto s.

Both of these have to do with specific questions of how buyers and sellers meet and match in real markets, i.e. having to do with *institutions* such as property rights and mechanism design. These questions are out of scope for us, and we abstract them away by postulating some effective equilibrium computation algorithm[5] $\text{Equ}(\omega, \alpha_b^1, \ldots \alpha_b^m) = (\mathbf{p}, \omega[\alpha^1], \ldots \omega[\alpha^m])$ i.e. which takes the total perceived quantity of goods in the world Ω and each agent's valuation function $\alpha_b^i : \Omega \to \mathbb{R}$, and returns a price vector $\mathbf{p} \in \Omega$ and allocations to each agent $\omega[\alpha^i] \in \Omega$, such that (in line with a Walrasian equilibrium with quasilinear utilities [22]):

- $\omega = \sum \omega[\alpha^i]$ (the full quantity is allocated)
- $\mathbf{p} \cdot \omega[\alpha^i] \leq \alpha_b^i(\omega[\alpha^i])$ for all α^i (no agent pays for what it doesn't value), and
- $\omega[\alpha^i] = \arg\max_{\omega' \in \Omega} \alpha_b^i(\omega') - \mathbf{p} \cdot \omega'$ for all α^i (each agent gets a utility-maximizing bundle at the given price).

Definition 2.2 (Wide market)**.** Everything from the POMDP setup and the agent type in Def 2.1 remains the same; except that $\mathcal{S}$ and Ω are now vector spaces with each vector called a *goods bundle*. Further, we have action spaces $\mathcal{X}_\omega$ indexed by $\omega \in \Omega$ such that (1) for any $x \in \mathcal{X}_\omega$, there is an "exercised property right" denoted $s_x(\omega) \in \mathcal{S}$ such that $\omega(s_x(\omega)) = \omega$ and $x(s) = x(s_x(\omega)) + (s - s_x(\omega))$ (i.e. each agent's actions transform only the goods they own) and (2) there is an injective map $\xi : \mathcal{X}_{\omega_1} \times \mathcal{X}_{\omega_2} \to \mathcal{X}_{\omega_1+\omega_2}$ such that $\xi(x_{\omega_1}, x_{\omega_2}) = x_{\omega_1}(s_{x_{\omega_1}}(\omega_1)) + x_{\omega_2}(s_{x_{\omega_2}}(\omega_2)) + (s - s_{x_{\omega_1}}(\omega_1) - s_{x_{\omega_2}}(\omega_2))$ (this is used to combine actions by different agents). The agents now have dependent type signatures $\alpha : (\omega : \Omega) \to \mathcal{X}_\omega \times \mathbb{R}$, and the transition probability $x(s) \sim \mathbf{P}(s' \mid s, x)$, reward function $\mathbf{R}(s, x, s')$ and observation distribution $\mathbf{O}(\omega \mid s)$ are now interpreted as applying to "private property", i.e. to any goods bundle in their respective domains, rather than to the whole state, e.g. each action $x(s)$ defines a *production function* that transforms one goods bundle into another, and α_b is an agent's *valuation function* over all possible bundles, i.e. how much it is willing to pay for a particular perceived bundle (if it's differentiable, then $\nabla\alpha_b(\omega)$ can be interpreted as the price vector it offers). The market algorithm proceeds as in Algorithm 2 .

Algorithm 2 Wide market

procedure MARKET ▷ Forward pass
  parameters: $w[\alpha] \in \mathbb{R}$ ▷ wealths of each $\alpha \in \mathcal{A}$
  input: $\omega \in \Omega$
  ▷ Cap bids by budget
  $b[\alpha] \leftarrow \lambda s : \min(\alpha_b(s), w[\alpha])$ for all $\alpha \in \mathcal{A}$
  ▷ Compute equilibrium prices and allocations
  $\mathbf{p}, \omega'[\alpha^1], \ldots \omega'[\alpha^n] \leftarrow \text{Equ}(\sum \omega[\alpha], b[\ldots])$
  $x[\alpha] \leftarrow \hat{\alpha}(\omega'[\alpha])$ for all $\alpha \in \mathcal{A}$ ▷ Determine actions
  return $x[\alpha^1], \ldots x[\alpha^n], \mathbf{p}, \omega'[\alpha^1], \ldots \omega'[\alpha^n]$
end procedure
procedure CAPITALISM ▷ Training loop
  Initialize agent wealths $w[\alpha] \in \mathbb{R}$ for each $\alpha \in \mathcal{A}$
  Initialize agent properties $\omega[\alpha] \in \Omega$ for each $\alpha \in \mathcal{A}$
  Initialize state $s \in \mathcal{S}$
  while $t \in \mathbb{N}$ **do**
    $\omega \leftarrow \omega(s)$ ▷ Generate observation
    $\cdots \leftarrow \text{MARKET}(\omega)$ ▷ get all outputs
    $w[\alpha] \leftarrow w[\alpha] - \mathbf{p} \cdot \omega'[\alpha]$ for all α ▷ Charge buyers
    $w[\alpha] \leftarrow w[\alpha] + \mathbf{p} \cdot \omega[\alpha]$ for all α ▷ Pay sellers
    $s[\alpha] \leftarrow s_{x[\alpha]}(\omega[\alpha])$ for all α ▷ Calculate property rights
    $s'[\alpha] \leftarrow x[\alpha](s[\alpha])$ for all α ▷ Transform goods
    $w[\alpha] \leftarrow w[\alpha] + \mathbf{R}(s[\alpha], x[\alpha], s'[\alpha])$ for all α
    $s \leftarrow \sum s'[\alpha]$ ▷ Update state
  end while
end procedure

Computing prices via backpropagation. Though it remains to be seen how standard RL problems might be cast in this setting, we expect implementations of this algorithm to be much more effective than of Def 2.1, as it allows us to use simpler and more specialized agents in the collection $\mathcal{A}$. In particular, these agents do not need to estimate the valuations of the whole world, but only of their particular input goods.

This last point can be illustrated particularly nicely when the setup is an MDP, and rewards are replaced by consumers – here, $\Omega = \mathcal{S}$ and $\omega(s) = s$, so $\hat{\alpha} : \mathcal{S} \to \mathcal{X}$ can directly be interpreted as a production function $\hat{\alpha} : \mathcal{S} \to \mathcal{S} := \hat{\alpha}(s)(s)$. Then if the agent can estimate what the market prices of its output goods will be (e.g. if prices are sufficiently stable that it makes sense to speak of a "prevailing price" $\mathbf{p}$), then it can compute its offered prices via the chain rule – where $D\hat{\alpha}$ denotes the Jacobian:

[4] $\oplus$ denotes the direct sum of vector spaces, which is a Cartesian product equipped with a pointwise vector addition operator

[5] e.g. there are results demonstrating that simple tâtonnement converges to a Walrasian equilibrium when the agents' valuations are gross subtitutes [13].

$$\nabla \alpha_b = D\hat{\alpha} \cdot \mathbf{p} \tag{1}$$

i.e. once the market "graph" is fixed, prices can be computed by simply backpropagating consumer bids through the graph. This generalizes the result in [33], which demonstrated this relationship for deep markets only.

3 MOTIVATION FOR MARKET-BASED AI

In this section, we describe how markets could potentially generalize neural networks and provide a more "flexible training mechanism". Although we have presented our algorithms in an RL setting, they can even be applied to supervised learning tasks by treating internal representations as "states". To see this, it is illustrative to see how a simple neural network can be recast as a market.

Theorem 3.1 (Neural networks as markets)**.** *Consider a fully-connected neural network $f : \mathbb{X} \to \mathbb{Y} := f_n \circ \dots f_1$ where each $f_i : \mathbb{R}^{m_{i-1}} \to \mathbb{R}^{m_i}$ is a layer, i.e. a function of the form $f_i(\mathbf{x}) = \sigma(W_i\mathbf{x}+\mathbf{b}_i)$ where σ is a ReLU activation. Then there is a deep market whose forward pass performs the same operation as f.*

Proof. The construction is straightforward. Define the state space $\mathcal{S} := \left[\bigoplus_{1\le i\le n} \mathcal{S}_i\right] \oplus \mathbb{Y}$ (with each $\mathcal{S}_i := \mathbb{R}^{m_i}$), with $\omega : \mathcal{S} \to \Omega$ discarding only the last component $\mathbb{Y}$ which represents the true label which is unchanged under all actions. Each $\mathcal{X}_\omega = \{(W_i, \mathbf{b}_i) : W_i \in \mathbb{R}^{m_i \times m_{i-1}}, \mathbf{b}_i \in \mathbb{R}^{m_i}\}$ if $\omega \in \mathcal{S}_{i-1}$ and empty if no such i exists, and an action $x = (W_i, \mathbf{i})$ acts on $s \in \mathcal{S}_{i-i}$ as $x(s) = \sigma(W_i s+\mathbf{b}_i)$. The reward $\mathbf{R}(s, x, s') = -\ell(s', s'_{\mathbb{Y}})$ for some loss function ℓ if $s' \in \mathcal{S}_n$ and 0 otherwise. Finally, let $\mathcal{A}$ consist of all constant maps to $\mathcal{X}_\omega$ and endow non-zero wealth to only those agents whose actions' parameters are the same as some f_i. □

While the market model, i.e. the forward pass, in Theorem 3.1 is the same as the neural network, the training mechanism is Capitalism (as defined in Algorithm 1) rather than backpropagation. Detailed below are some strengths of this we anticipate:

Search and dynamic scale. *Reasoning* is widely touted as a key limitation of current-day LLMs [16, 21]. A view held by some researchers including Yann LeCun [34], is that this is due to the fact that "[neural networks] produce their answers with a constant number of computational steps between input and output", independent of the complexity required by the problem. Some proposed architectures that avoid this limitation include dynamic neural networks [14], adaptive computation time [12] as well as chain-of-thought based methods such as o1 [26]. Markets provide a principled alternative, as here the structure of the computational graph is itself learned, and different agents and structures may be active for different inputs.

Complete feedback. Informally speaking, markets allow *any* aspect of the system to be optimized. Formal results are needed to make this statement precise, but intuitively: any aspect of a learner, such as any hyperparameter, or meta-learning, can be changed by adding a trader to the market who will profit if his changes are beneficial and the incentives are correctly designed. This is suggestive of the notion of "complete feedback" in AI alignment research, which refers to the property that "the trainer can enact *any* modification they'd like to make to the system" [8], and is viewed as a desirable characteristic of an AI system for alignment.

4 PRACTICALITY AND FUTURE WORK

We have presented two general frameworks for market-based RL agents, and illustrated that they may be seen to generalize neural networks in a supervised learning setting, albeit with a more flexible training mechanism that holds promise to address the limitations of current-day AIs with respect to reasoning and alignment properties.

Despite these theoretical strengths, our algorithm as described faces practical challenges to implement in real-world machine learning tasks: blindly enumerating large classes of even simple agents is inefficient (compared to backpropagation, where the search is guided by gradients), and we have to store many more agents in memory than the "size" of the network (the exact number depending on the rule we use to prune low-wealth agents). Some potentially promising approaches include:

- "integrated" models which perform backpropagation by default but intelligently resort to markets when it expects changing the network structure to be worthwhile
- having each agent simultaneously learn its parameters via backpropagation
- decentralized set-ups, perhaps using frameworks such as BitTensor [28], allowing traders to be shared across machine learning applications.

Markets of LLMs. A more immediately feasible application is to develop *markets comprised of LLMs*, i.e. where $\mathcal{A}$ is a collection of LLM agents. For instance, one may let $\mathcal{S} = \Omega$ be a message space, and let actions act on s by appending some "chain-of-thought item" to the current message. The final reward is determined by human feedback, and intermediate rewards by bids. Such a market would function as a "reasoning model" analogous to o1.

The extension to a wide market is also immediate: agents may bid for the right to read only a portion of the message space[6] – this allows for more precise credit assignment to contributions by different agents, and may be understood as to *trees-of-thought* [35] what o1 is to *chain-of-thought*.

Theoretical work. The most pressing need at present is for *precise theoretical results* on the effectiveness of market-based algorithms. An immediate research agenda includes the following:

- Determining **convergence and optimality conditions** of market algorithms; in particular, generalizing the "coverage" results of BRIA [25] and logical induction [9], i.e. demonstrating that the market will give a fair chance to the best policy, conditional on some suitable wealth endowments.
- A **Learning Theory** perspective on markets and the wealth update mechanism. In particular, (real-world) markets appear to have many useful features from an alignment standpoint, such as their inherent capacity for online learning and generalization even from imperfect reward signals.
- A thorough translation of **economic terminology** into our model – especially concepts like perfect competition, economies of scale, growth and welfare.

Finally, to accelerate empirical work with market-based algorithms, we plan to release a Python library for efficiently creating and applying market-based algorithms.

[6] As for how to enable the agent to "inspect" the message to make an informed bid without it stealing the entire message, [27] is relevant: the agent can subcontract another LLM to inspect the message and place the bid, then have its context deleted.

REFERENCES

[1] Kenneth J. Arrow and Gerard Debreu. 1954. Existence of an Equilibrium for a Competitive Economy. *Econometrica* 22, 3 (1954), 265–290. https://doi.org/10.2307/1907353 arXiv:1907353

[2] Eric B. Baum. 1999. Toward a Model of Intelligence as an Economy of Agents. *Machine Learning* 35, 2 (May 1999), 155–185. https://doi.org/10.1023/A:1007593124513

[3] Michael Chang, Sid Kaushik, S. Matthew Weinberg, Tom Griffiths, and Sergey Levine. 2020. Decentralized Reinforcement Learning: Global Decision-Making via Local Economic Transactions. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 1437–1447.

[4] Xi Chen, Decheng Dai, Ye Du, and Shang-Hua Teng. 2009. Settling the Complexity of Arrow-Debreu Equilibria in Markets with Additively Separable Utilities. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. 273–282. https://doi.org/10.1109/FOCS.2009.29

[5] Xi Chen and Shang-Hua Teng. 2009. Spending Is Not Easier Than Trading: On the Computational Equivalence of Fisher and Arrow-Debreu Equilibria. In *Algorithms and Computation*, Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 647–656.

[6] Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley H. Holliday, Bob M. Jacobs, Nathan Lambert, Milan Mosse, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, Emanuel Tewolde, and William S. Zwicker. 2024. Position: Social Choice Should Guide AI Alignment in Dealing with Diverse Human Feedback. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 9346–9360.

[7] Lawrence Davis. 1988. Mapping Classifier Systems into Neural Networks. In *Proceedings of the 1st International Conference on Neural Information Processing Systems (NIPS'88)*. MIT Press, Cambridge, MA, USA, 49–56.

[8] Abram Demski. 2024. Complete Feedback. https://www.alignmentforum.org/posts/3ag99iJEgFFwyj64Z/complete-feedback.

[9] Scott Garrabrant, Tsvi Benson-Tilsen, Andrew Critch, Nate Soares, and Jessica Taylor. 2020. Logical Induction. https://doi.org/10.48550/arXiv.1609.03543 arXiv:1609.03543 [cs, math]

[10] Luise Ge, Daniel Halpern, Evi Micha, Ariel D. Procaccia, Itai Shapira, Yevgeniy Vorobeychik, and Junlin Wu. 2024. Axioms for AI Alignment from Human Feedback. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[11] Dhananjay K. Gode and Shyam Sunder. 1993. Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy* 101, 1 (February 1993), 119–137. https://doi.org/10.1086/261868

[12] Alex Graves. 2017. Adaptive Computation Time for Recurrent Neural Networks. https://doi.org/10.48550/arXiv.1603.08983 arXiv:1603.08983 [cs]

[13] Faruk Gul and Ennio Stacchetti. 1999. Walrasian Equilibrium with Gross Substitutes. *Journal of Economic Theory* 87, 1 (1999), 95–124. https://doi.org/10.1006/jeth.1999.2531

[14] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. 2021. Dynamic Neural Networks: A Survey. https://doi.org/10.48550/arXiv.2102.04906 arXiv:2102.04906

[15] John H. Holland. 1985. Properties of the Bucket Brigade. In *Proceedings of the 1st International Conference on Genetic Algorithms*. L. Erlbaum Associates Inc., USA, 1–7.

[16] Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards Reasoning in Large Language Models: A Survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1049–1065. https://doi.org/10.18653/v1/2023.findings-acl.67

[17] Karim Jamal, Michael S. Maier, and Shyam Sunder. 2015. Simple Agents, Intelligent Markets. *SSRN Electronic Journal* (2015). https://doi.org/10.2139/ssrn.2478665

[18] Ivo Kwee, Marcus Hutter, and Juergen Schmidhuber. 2001. Market-Based Reinforcement Learning in Partially Observable Worlds. In *Proceedings of the International Conference on Artificial Neural Networks*. arXiv, 865–873. https://doi.org/10.48550/arXiv.cs/0105025 arXiv:cs/0105025

[19] Leonard E Read. 1958. I, Pencil: My Family Tree as Told to Leonard E. Read. *The Freeman* 8 (December 1958), 32–37.

[20] Lionel W. McKenzie. 1959. On the Existence of General Equilibrium for a Competitive Market. *Econometrica* 27, 1 (1959), 54–71. https://doi.org/10.2307/1907777 arXiv:1907777

[21] Aidan McLau. 2024. AI Search: The Bitter-er Lesson. https://news.ycombinator.com/item?id=40683697.

[22] Nolan Miller. 2006. Notes on Microeconomic Theory. (August 2006). (Lecture Notes).

[23] Marvin Minsky. 1988. *Society Of Mind*. Simon and Schuster.

[24] Eric Neyman. 2024. Algorithmic Bayesian Epistemology. https://doi.org/10.48550/arXiv.2403.07949 arXiv:2403.07949 [cs]

[25] Caspar Oesterheld, Abram Demski, and Vincent Conitzer. 2023. A Theory of Bounded Inductive Rationality. *Electronic Proceedings in Theoretical Computer Science* 379 (July 2023), 421–440. https://doi.org/10.4204/EPTCS.379.33 arXiv:2307.05068 [cs]

[26] OpenAI. 2024. Learning to Reason with LLMs.

[27] Nasim Rahaman, Martin Weiss, Manuel Wüthrich, Yoshua Bengio, Li Erran Li, Chris Pal, and Bernhard Schölkopf. 2024. Language Models Can Reduce Asymmetry in Information Markets. https://doi.org/10.48550/arXiv.2403.14443 arXiv:2403.14443 [cs]

[28] Yuma Rao, Jacob Steeves, Ala Shaabana, Daniel Attevelt, and Matthew McAteer. 2021. BitTensor: A Peer-to-Peer Intelligence Market. https://doi.org/10.48550/arXiv.2003.03917 arXiv:2003.03917

[29] Jürgen Schmidhuber. 1987. Evolutionary Principles in Self-Referential Learning, or on Learning How to Learn: The Meta-Meta-. Hook.

[30] Jurgen Schmidhuber. 1989. A Local Learning Algorithm for Dynamic Feedforward and Recurrent Networks. *Connection Science* 1, 4 (January 1989), 403–412. https://doi.org/10.1080/09540098908915650

[31] Alan Schwartz. 2008. How Much Irrationality Does the Market Permit? *The Journal of Legal Studies* 37, 1 (January 2008), 131–159. https://doi.org/10.1086/519963

[32] F. A. von Hayek. 1937. Economics and Knowledge. *Economica* 4, 13 (1937), 33–54. arXiv:2548786

[33] John Wentworth. 2018. Competitive Markets as Distributed Backprop.

[34] Yann LeCun. 2023. Towards Machines That Can Learn, Reason, and Plan. In *AI and Barrier of Meaning Workshop*. Santa Fe Institute.

[35] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, 11809–11822.